

Implementation of Instantaneous Frequency Estimation based on Time-Varying AR Modeling

Roshin Kadanna Pally

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Dr. A. A. (Louis) Beex, Chair

Dr. Michael Buehrer

Dr. A. Lynn Abbott

April 15, 2009

Blacksburg, Virginia

Keywords: Time-Varying Autoregressive Modeling, Instantaneous Frequency Estimation,
Block Toeplitz Inversion, Narrowband Interference Mitigation

Copyright © 2009 by Roshin Kadanna Pally. All rights reserved.

Implementation of Instantaneous Frequency Estimation based on Time-Varying AR Modeling

Roshin Kadanna Pally

Dr. A. A. (Louis) Beex, Chairman

(ABSTRACT)

Instantaneous Frequency (IF) estimation based on time-varying autoregressive (TVAR) modeling has been shown to perform well in practical scenarios when the IF variation is rapid and/or non-linear and only short data records are available for modeling. A challenging aspect of implementing IF estimation based on TVAR modeling is the efficient computation of the time-varying coefficients by solving a set of linear equations referred to as the generalized covariance equations. Conventional approaches such as Gaussian elimination or direct matrix inversion are computationally inefficient for solving such a system of equations especially when the covariance matrix has a high order.

We implement two recursive algorithms for efficiently inverting the covariance matrix. First, we implement the Akaike algorithm which exploits the block-Toeplitz structure of the covariance matrix for its recursive inversion. In the second approach, we implement the Wax-Kailath algorithm that achieves a factor of 2 reduction over the Akaike algorithm in the number of recursions involved and the computational effort required to form the inverse matrix.

Although a TVAR model works well for IF estimation of frequency modulated (FM) components in white noise, when the model is applied to a signal containing a finitely correlated signal in addition to the white noise, estimation performance degrades; especially when the correlated signal is not weak relative to the FM components. We propose a decorrelating TVAR (DTVAR) model based IF estimation and a DTVAR model based linear prediction error filter for FM interference rejection in a finitely correlated environment. Simulations show notable performance gains for a DTVAR model over the TVAR model for moderate to high SIRs.

*To my parents Raveendran Puthalath and Ranjini Kadanna Pally
for their unconditional love, care, support, and confidence in my endeavors*

Acknowledgements

I am profoundly indebted to my advisor Dr. A. A. (Louis) Beex for giving me the opportunity to be a part of the DSP Research Lab at Virginia Tech. The time spent under his supervision has been the most productive in my academic life. His expertise in the area of Signal Processing and diligence in research have motivated me to self-improve. I am grateful to him for extending me the opportunity to work on funded projects that exposed me to advanced topics and helped me to pursue my education. Finally, I wish to appreciate his outstanding guidance and insightful instructions that were invaluable in the completion of this thesis.

I would like to thank Dr. Michael Buehrer and Dr. A. Lynn Abbott for their time and effort in reviewing this work. Their comments are greatly appreciated. A special thanks to my friend Bijoy Antony Jose for being an excellent roommate and for making my graduate life enjoyable. In addition, I would like to thank my lab mates Areg Baghdasaryan and Yongfeng Zhi.

I am forever indebted to my parents Raveendran and Ranjini for their hardships and sacrifices in raising me and providing me the education that continue to enable me to strive toward intellectual excellence. Also, my brother Rijin Kadanna Pally. Your jokes and humor have cheered me in difficult times. Thanks for all the fun and adventures.

Contents

1	Introduction	1
1.1	Research Motivation and Goal	1
1.2	Literature Review	6
1.3	Research Outline	14
2	TVAR Modeling and IF Estimation	15
2.1	AR Signal Modeling	15
2.2	Time-Varying Autoregressive (TVAR) Modeling	19
2.3	IF Estimation	24
2.3.1	Procedure	25
2.3.2	Linear FM Signal	27
2.3.3	Non-linear FM Signal	34
2.3.4	Autocorrelation Method	38
2.4	Summary	41
3	Efficient Matrix Inversion for TVAR IF Estimation	42
3.1	Covariance Matrix Inversion	42
3.2	Akaike Algorithm	43
3.2.1	Properties of block-Toeplitz Matrix	44
3.2.2	Recursive Inversion Algorithm	45
3.2.3	Validity of Recursions and Computational Complexity	54
3.2.4	Experimental Verification	56
3.3	Wax-Kailath Algorithm	58
3.3.1	Properties of block-Toeplitz matrix with Toeplitz blocks	59
3.3.2	Wax-Kailath Recursions	65
3.3.3	Computational Complexity	71

3.3.4	Neumann Series	74
3.3.5	Experimental Verification	75
3.4	Summary	78
4	IF Estimation Application	79
4.1	Motivation for Decorrelating TVAR Model	79
4.2	Decorrelating TVAR Based IF Estimation	86
4.2.1	Decorrelating TVAR Model	86
4.2.2	IF Estimation in a Finitely Correlated Environment	88
4.2.3	Simulation Results	89
4.3	FM Interference Mitigation	93
4.3.1	Time-Varying PEF and Compensator	94
4.3.2	Simulation Results	96
4.4	Summary	101
5	Conclusion	102
5.1	Thesis Summary	102
5.2	Contributions	103
5.3	Suggestions for Future Work	104
	Bibliography	107

List of Figures

1.1	Case 1 – NBIM Applied to a Stationary Interference.	3
1.2	Nonstationary Interfering Frequency and its Estimate.	4
1.3	Case 2 – NBIM Applied to a Nonstationary Interference.	4
1.4	SER Performance for Stationary and Nonstationary Interference.	5
2.1	(a) Innovations Representation (b) Whitening Process.	16
2.2	The Basis Function Set $u_{m,n}$ as in (2.39).	28
2.3	Estimated TVAR Coefficients $a_{k,n}$ for a Linear FM Signal.	30
2.4	Trajectory of Time-Varying Poles for a Linear FM Signal.	30
2.5	True and Estimated IF of Linear FM components.	31
2.6	Comparison of Original Data and TVAR Prediction for Linear FM Signal.	32
2.7	Time Varying Power Spectrum of the TVAR Model.	33
2.8	Time-Frequency View of the TFD.	33
2.9	Estimated TVAR Coefficients $a_{k,n}$ for a Non-linear FM Signal.	35
2.10	Trajectory of Time-varying Poles for a Non-linear FM Signal.	35
2.11	True and Estimated IF of Non-linear FM component.	36
2.12	Comparison of Original Data and TVAR Prediction for Non-linear FM Signal.	37
2.13	Time-Frequency View of the TFD of Non-linear FM Signal.	37
2.14	Trajectory of Time-Varying Poles for a Linear FM Signal using Autocorrelation Method.	38
2.15	True and Estimated IF of FM Components using Autocorrelation Method.	39
2.16	Trajectory of Time-Varying Poles for a Non-linear FM Signal using Autocorrelation Method.	40
2.17	True and Estimated IF of Non-linear FM Component using Autocorrelation Method.	41

3.1	Solution Vector Comparison for Akaike Algorithm.	57
3.2	Magnitude of Solution Error for Gaussian Elimination and Akaike Algorithm.	57
3.3	Solution Vector Comparison for Wax-Kailath Algorithm.	76
3.4	Magnitude of Solution Error for Gaussian Elimination and Wax-Kailath Algorithm (9 Neumann iterations).	77
3.5	Magnitude of Solution Error for Wax-Kailath Algorithm (15 Neumann iterations).	77
4.1	CPM IF Estimation using $N = 8$ and $q = 0$	81
4.2	CPM IF Estimation using $N = 128$ and $q = 1$	82
4.3	TVAR and DTVAR Model Based IF Estimation Comparison.	83
4.4	TVAR and DTVAR Model Based IF Estimation Comparison: Zoomed-In.	83
4.5	DTVAR Parameters for $q = 1$	84
4.6	Linear IF law: True (solid, red), TVAR (Dashed, blue), and Decorrelating TVAR (dots, black).	90
4.7	1/MSE vs. SIR for IF Estimation of a linear FM signal: TVAR (solid, red) and Decorrelating TVAR (dashed, blue).	91
4.8	Non-linear IF law: Decorrelating TVAR based IF Estimation	92
4.9	1/MSE vs. SIR for IF Estimation of a Non-linear FM signal: TVAR (solid) and Decorrelating TVAR (dashed).	93
4.10	Block Diagram of a TVLPEF.	94
4.11	Block Diagram of a TVDAC.	96
4.12	FM Interference Mitigation in a CPM Signal.	98
4.13	SER vs. SIR for FM Interference Suppression using TVAR Model and DTVAR Model at SNR=20 dB.	99
4.14	Absolute value of Interference Prediction Error ($ \xi $).	100

Nomenclature

AR	Autoregressive
ARMA	Autoregressive Moving Average
CPM	Continuous Phase Modulation
DFT	Discrete Fourier Transform
DSSS	Direct Sequence Spread Spectrum
DTVAR	Decorrelating Time-Varying Autoregressive
ECG	Electrocardiogram
EEG	Electroencephalogram
FFT	Fast Fourier Transform
FM	Frequency Modulation
IF	Instantaneous Frequency
IM	Interference Mitigation
ISI	Inter Symbol Interference
NBI	Narrowband Interference
NBIM	Narrowband Interference Mitigation

OSD	One Symbol Decoder
PEF	Prediction Error Filter
PLL	Phase-Locked Loop
SER	Symbol Error Rate
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
STFT	Short-Time Fourier Transform
TFD	Time-Frequency Distribution
TVAR	Time-Varying Autoregressive
TVDAC	Time-Varying Decision Aided Compensator
TVLPEF	Time-Varying Linear Prediction Error Filter
TVNF	Time-Varying Notch Filter
TVWF	Time-Varying Wiener Filter
WVD	Wigner-Ville Distribution
YWE	Yule-Walker Equation

Chapter 1

Introduction

This thesis examines parametric time-varying autoregressive (TVAR) modeling for nonstationary signals and its use to implement instantaneous frequency (IF) estimation of time-varying frequency components in a signal. Efficient and practical ways of computing the time-varying parameters of the TVAR model are investigated. For illustration, the application of IF estimation to interference mitigation (IM) in a continuous phase modulation (CPM) communications system is considered. In this chapter, we present our research motivation and review some related works in the literature.

1.1 Research Motivation and Goal

Estimating the IF of a signal is a topic of significant research interest in the realm of wireless communications, radar, sonar, biomedical, speech, and seismic signal processing. The widespread applications of IF estimation stem from the underlying nonstationarity of signals encountered in these fields. A nonstationary signal has spectral characteristics that vary with time, the time variation being a result of the physical phenomena associated with its generation or methods and mediums involved in its communication. For instance, in a frequency modulated (FM) signal used in wireless communications, the frequency of the carrier is varied according to a modulating signal representing information – such as voice – that is inherently nonstationary. As a result, the information resides in the continuously and

instantaneously changing frequency (the IF) of the FM signal and its accurate estimation is of primary interest at the FM receiver. On a different note, in satellite communications systems using a CPM signal, an FM signal can appear as a strong narrowband interference (NBI) and degrade the performance of the CPM receiver. In this situation, IF estimation can be used to design filters that mitigate and/or extract the NBI. In biomedical applications, IF estimation provides useful information for the analysis of electroencephalogram (EEG) and electrocardiogram (ECG) signals.

In spite of the nonstationarity inherent in many signals of practical interest, most of the initial work on IF estimation and modeling of nonstationary signals is based on the assumption that the signal is stationary or only slowly time-varying over a short data record. Under such assumptions, it is possible to use stationary sinusoidal frequency estimation techniques or adaptive algorithms to estimate the IF. Some of these techniques include the phase difference estimators, zero-crossing IF estimators, adaptive IF estimation using a phase-locked loop (PLL), IF estimation based on the peak of discrete Fourier transforms (DFT), and high resolution IF estimation using stationary autoregressive AR modeling [1]. The most direct and fastest approach is estimating the IF from the peak of a DFT spectral estimate. The speed in this approach is a result of the Fast Fourier Transform (FFT) algorithm available for computing the DFT. In addition, for a small computational cost, the frequency estimate can be significantly improved by the three and five sample interpolators proposed by Malcolm [2] in 1998.

However, when the signal has frequency components that vary rapidly even in small data records, the methods based on stationarity assumptions give poor results. To illustrate this, we show an example simulation of narrowband interference mitigation (NBIM) in a CPM signal when the interference is a fixed frequency sinusoid of 600 Hz (stationary) in case 1 and a frequency sweep from 600 to 615 Hz at the rate of 0.0018 Hz/sample (nonstationary) in case 2. In both cases, we assume local stationarity within a data record of 256 samples and use a DFT-based approach to estimate the frequency of the interference. Furthermore,

a five sample interpolator is used to improve the frequency estimates. The signal-to-noise ratio (SNR) used is 25 dB. For case 1, a section of clean CPM signal, unfiltered CPM signal containing a stationary NBI, and the signal resulting after NBIM are shown in Figure 1.1 for a Signal-to-Interference Ratio (SIR) of -30 dB (all composite signals are scaled to have unit power).

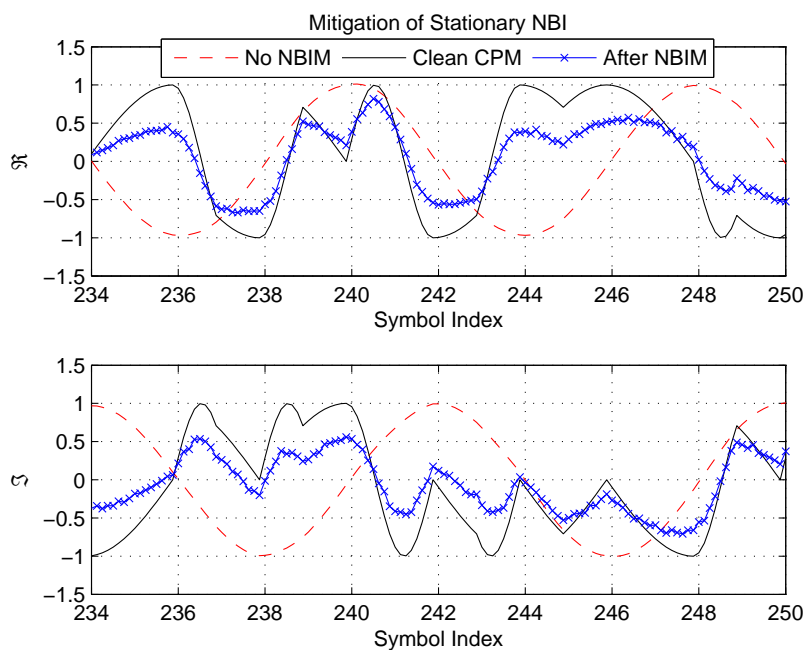


Figure 1.1: Case 1 – NBIM Applied to a Stationary Interference.

From Figure 1.1, we see that the stationary NBI has been effectively mitigated from the CPM signal with minor signal distortion. The effectiveness of the NBIM is further demonstrated in the Symbol Error Rate (SER) performance at different SIRs shown in Figure 1.4.

When the NBI is nonstationary, the interfering frequency can vary rapidly over short data records and the NBIM based on stationary frequency estimation cannot be used to effectively mitigate the interference. Also, an adaptive algorithm to do the same may not converge fast enough to track the rapidly varying frequency. To illustrate, in Figure 1.2 we show the frequency sweep of case 2 and its estimate obtained using the DFT-based approach at 25 dB SNR and -30 dB SIR. A section of clean CPM signal and the signal resulting after applying the DFT-based NBIM are shown in Figure 1.3.

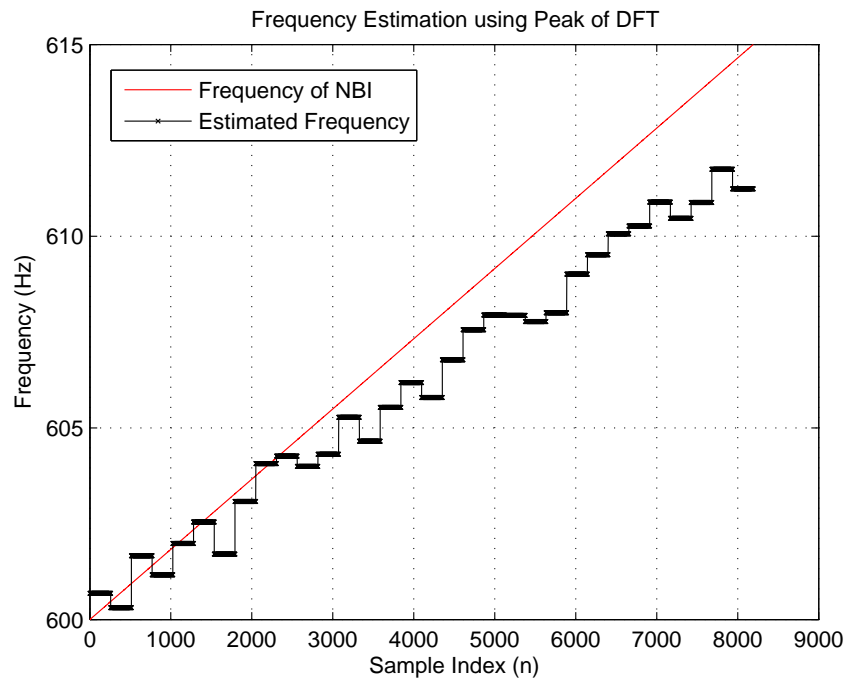


Figure 1.2: Nonstationary Interfering Frequency and its Estimate.

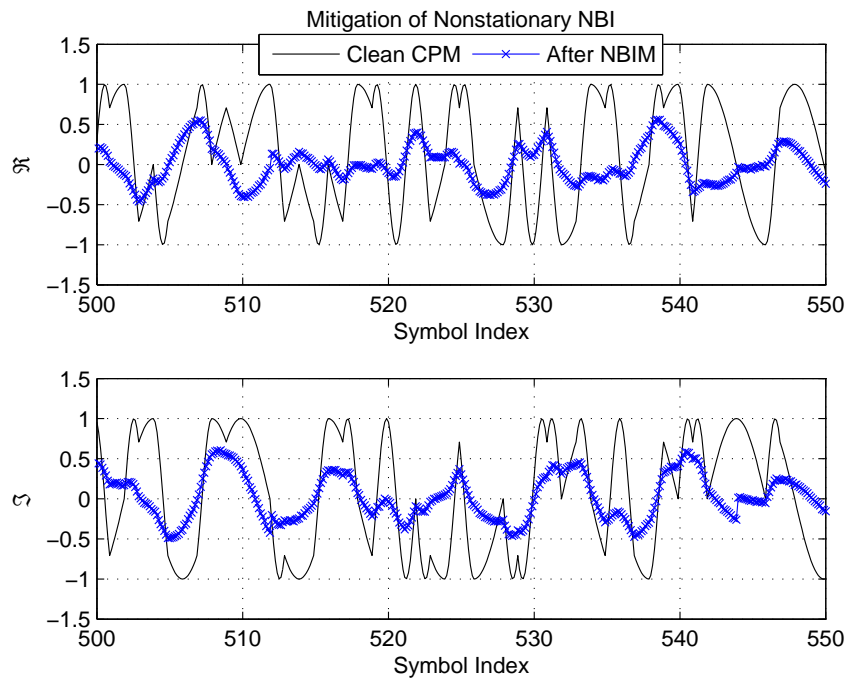


Figure 1.3: Case 2 – NBIM Applied to a Nonstationary Interference.

From Figure 1.2, we see that the DFT-based approach results in poor estimation of the frequency sweep. The resulting frequency sweep estimate has a step-like behavior because only a single estimate is available for a data record of 256 samples. We also note that using a DFT-based approach results in frequency quantization errors because for a given number of FFT points and sampling frequency, only a finite number of frequencies can be represented exactly. From Figure 1.3, we see that the signal resulting after NBIM is severely distorted. This happens because of the stationarity assumptions made about a signal that is actually nonstationary. Consequently, a large number of symbol errors results and the SER performance is degraded. The SER performance for both cases is shown in Figure 1.4. From Figure 1.4, we see that the NBIM has improved the SER performance when the interfering signal is stationary while the performance improvement is very little to none (for low SIRs) when the interfering signal is nonstationary.

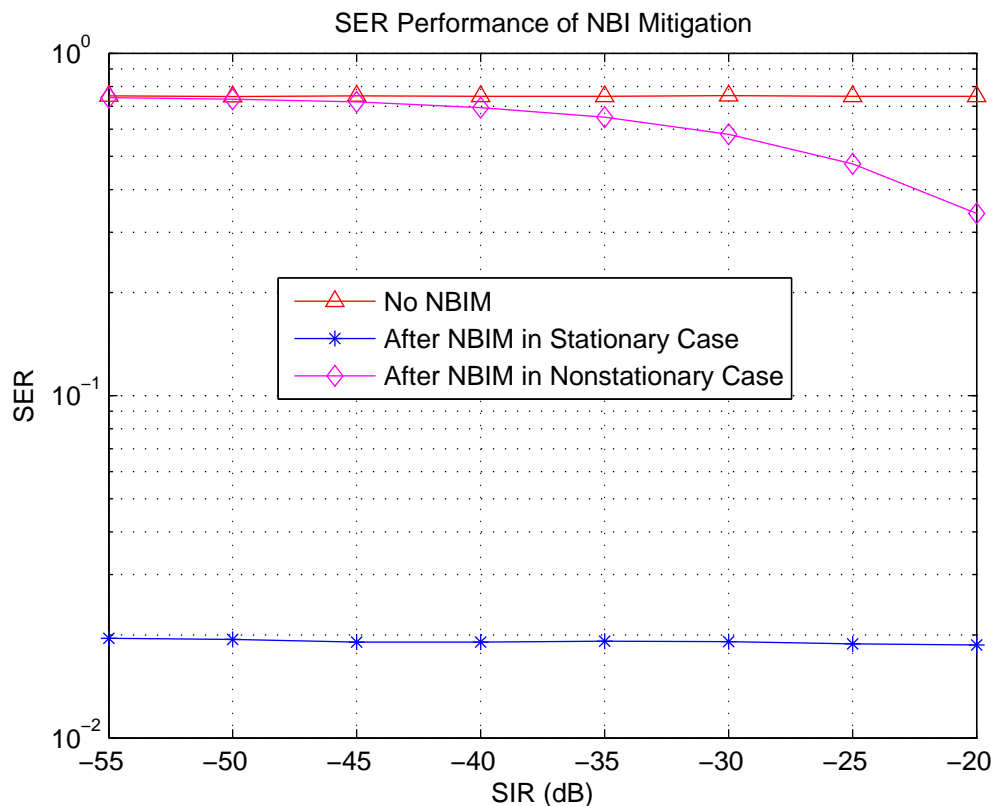


Figure 1.4: SER Performance for Stationary and Nonstationary Interference.

The performance degradation we have seen in the presented simulation is a result of applying a constant model to a time-varying signal. Furthermore, if the time-variation in the IF of a signal is rapid and non-linear, the application of a constant model is not meaningful anymore. More recently, various time-varying signal processing methods have surfaced to account for the nonstationarity in signals. An extensive review and comparison of some of these methods applied to IF estimation of a FM signal was contributed by Boashash [1] in 1992. The objective in IF estimation is to obtain an optimal or nearly optimal estimate of the IF of a noisy nonstationary signal using a criterion such as the least mean square error. The optimal filter that achieves this objective is referred to as the Time-Varying Wiener Filter (TVWF). Various methods for time-varying Wiener filtering have been proposed in the literature – such as – filtering based on Priestley’s evolutionary spectrum theory, Multi-resolution Parametric Spectral Estimation (MPSE), spectral representation of the nonstationary Wiener filter based on the Wigner-Ville Distribution (WVD) and Weyl Symbol (WS) of linear operators, and time-varying autoregressive (TVAR) modeling [3]. The TVAR modeling approach is particularly attractive because it is parametric and provides high resolution.

In this thesis, we consider the implementation of IF estimation based on TVAR modeling. The major contribution of this thesis is an investigation of efficient and practical ways of computing the time-varying coefficients in the TVAR model. Additionally, TVAR IF modeling is extended to the case of a finitely correlated environment.

1.2 Literature Review

A variety of methods for estimating the IF of a nonstationary signal exists in the literature. They can be broadly classified as non-parametric and parametric methods. The non-parametric methods do not require any *a priori* knowledge about the characteristics of a signal and many of these are based on the time-frequency distribution (TFD) of the signal. In the methods based on TFD, the IF is estimated from the peak of the TFD or its

first moment [1]. Two well-known TFDs are the short-time Fourier transform (STFT) and the WVD. The STFT is computationally efficient but has limited time-frequency resolution resulting in poor IF estimation. The WVD is optimal for linear FM signals at moderate to high SNRs [1]. However, a non-linear FM signal, low SNR, or insufficient frequency interpolation is found to saturate the performance of the WVD [4].

Recently, various improvements and/or extensions to these classical non-parametric methods of IF estimation have been proposed in the literature. Kwok and Jones proposed an improved instantaneous frequency estimation method based on the adaptive short-time Fourier transform (ASTFT) with peak tracking for nonstationary signals [5]. Significant performance improvement over using the STFT for IF estimation was reported with this method at the expense of increased computational and memory costs. The peak of the polynomial Wigner-Ville Distribution (PWVD) was proposed as an IF estimator for non-linear FM signals corrupted by additive and multiplicative white Gaussian noise by Barkat in 2001 [6]. The spectrogram (a quadratic version of the STFT) and WVD among many others belong to a class of TFDs referred to as the Quadratic TFDs. Numerous IF estimators based on quadratic TFDs have also been proposed in the literature [7]. For low SNRs, an IF curve estimation based on a new class of TFDs referred to as joint distributions was contributed by Shui, Shang, and Zhao in 2007 [8]. The joint distributions are constructed from a bank of directionally smoothed pseudo-Wigner-Ville distributions and were shown to outperform quadratic TFD based IF estimation methods at low SNRs. Many other non-parametric methods as well as high-resolution TFDs for robust IF estimation continue to be actively pursued by researchers around the world.

Apart from low time-frequency resolution and/or lack of robustness to noise, a major limitation of the non-parametric methods is that these methods cannot incorporate any *a priori* information about the signal into the IF estimation process. However, there are applications where some knowledge, usually on how a signal is generated, is available beforehand. Parametric methods for IF estimation are designed to exploit such information

about the signal and therefore provide robustness to noise, high-resolution, and a more accurate IF estimate since it is free of frequency quantization errors. These methods are based on modeling the non-stationary signal using some parameters (coefficients) and the IF estimates are derived from these parameters. An important step in parametric IF estimation is to select an appropriate model for the signal from *a priori* knowledge on how the signal is produced or from experimental results that justify the selection of a particular model. Some widely used models are autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA). The model we use in this thesis is a non-stationary or time-varying version of the AR model. After selecting a suitable model, the next step is estimating the model parameters. For a non-stationary signal, the parameters are time-varying and their accurate and computationally efficient estimation is vital for accurate IF estimation and its efficient implementation. One approach to account for the time variation in the estimation of parameters is to use an adaptive algorithm to update the parameters at each sample instant. Another approach, which we use in this thesis, is to represent the time-evolution of the coefficients using a small number of basis functions. A number of basis functions which can be used for representing the time-evolution have been proposed in the literature [9]. In this thesis, we use basis functions that are powers of time. Some works in the literature pertinent to time-varying signal modeling and IF estimation are discussed next.

Modeling a nonstationary discrete time signal as an autoregressive process with time-varying coefficients was proposed by Rao [10] in 1970. Rao introduced the idea of approximating the time-varying coefficients using a weighted linear combination of a small set of known time functions referred to as the basis. The method was soon used for linear estimation of time-varying signals by Liporace [11] in 1975 and applied to time-varying linear predictive coding (LPC) of speech by Hall, Oppenheim, and Willsky [12] in 1977. A heuristic treatment of time-dependent ARMA modeling of nonstationary stochastic processes and their application to time-varying spectral estimation was provided by Grenier [9] in 1983. Soon, time-varying autoregressive (TVAR) modeling based IF estimation of nonstationary signals was proposed by Sharman and Friedlander [13] in 1984. In this method, the angles

of the roots (poles) of the time-varying autoregressive polynomial at each sample instant are used as the IF estimates of the FM components in the nonstationary signal. Polynomial root finding and subsequent angle computation is the approach we use in this thesis for IF estimation. However, we make a note that this method is not necessarily the most efficient. First, it involves additional computations to arrive at the IF estimate. Second, if there are multiple IF components and/or more poles than IF components, a grouping algorithm must be used to associate the poles with the components and/or to distinguish between signal and noise poles. Third, if the order of the TVAR polynomial is high, root-finding may not be a good approach because small rounding errors in coefficients can lead to large variations in the roots. An alternative is to estimate the IF from peaks of the TFDs obtained from the signal model. Although additional computational burden is not avoided, the FFT algorithm can be used to efficiently obtain the TFD from the model parameters at each sample instant. Another approach is to directly use the pole-angle as a parameter in the model [14].

The TVAR based approach was considered a poor estimator initially and no significant investigations were reported in the literature until the work by Shan and Beex [4] in 1998. Their investigations show that the TVAR based IF estimator is acceptably good and performs well in practical scenarios where the IF variation can be rapid and/or non-linear and only short data records are available for modeling. However, the performance of the TVAR based IF estimator was found to degrade fast below some low SNR threshold. To improve the performance of the estimator at low SNRs, Beex and Shan proposed a time-varying Prony method [15] in 1999. Here, the authors use a higher model order than what is required to model pure tones and let the additional poles capture the noise. A subset selection algorithm is then used to distinguish between poles that correspond to the signal and those that correspond to the noise. IF estimates are now derived from the angles of the “signal poles.” This approach was found to improve the performance of the TVAR based IF estimator at low SNRs.

An IF estimation approach at low SNRs using a time-varying notch filter (TVNF) was contributed by Johansson and White in 2008 [16]. Two types of TVNFs are given in their work. The first is based on an adaptive notch filter with the same parameters as that of a TVAR model. In the second approach, the pole-angle is used directly in the model so that the time-varying coefficients are actually frequencies and their time-evolution is described by a set of basis functions. Therefore, the IF can be computed as a linear combination of weighted basis functions and there is no need for polynomial rooting. Also, each IF component can be determined separately as a linear combination and there is no need for grouping algorithms. The TVNF based IF estimation was shown to be superior to the TVAR based approach at low SNRs. However, at high SNRs, the TVAR approach gives lower frequency MSE compared to the TVNF approach. The TVNF IF estimation approach is more involved than the TVAR based approach because it uses a regularized Gauss-Newton iterative algorithm requiring convergence. As a result, the size of the data block required is also large compared to the TVAR approach. Therefore, it may not be suitable in applications where only short data records are available and/or used. The computational times for an example simulation in the work suggests that the TVAR model is predominantly faster than TVNF in computing the time-varying AR coefficients.

As we have seen, IF estimates can be derived from the time-varying autoregressive parameters in different ways. Therefore, the centerpiece of implementing IF estimation based on TVAR modeling is the efficient computation of the time-varying coefficients. It is found that to arrive at the time-varying coefficients, a linear system of equations involving a block-Toeplitz matrix needs to be solved when the basis functions used are powers of time. Conventional methods based on Gaussian eliminations are computationally inefficient to solve this linear system especially when the system matrix has high order. The primary objective in this thesis is to solve the block-Toeplitz linear system recursively and facilitate a more efficient and practical implementation of IF estimation. For the simplest first order TVAR model, the system matrix has a Toeplitz structure. An efficient recursive procedure to solve a Toeplitz system of equations was first developed by Trench and is referred to as

the *Trench Algorithm* [17].

For TVAR models of order higher than one, the linear system has a block-Toeplitz structure. An iterative procedure for inverting a general block-Toeplitz matrix was derived by Akaike in 1973 [18]. Triangular decomposition algorithms for block-Toeplitz matrices were proposed by Rissanen in 1973 [19]. If the covariance method is used for signal modeling, the block-Toeplitz matrix we obtain has individual blocks that are close to being Hermitian-Toeplitz. Thus, by adding an appropriate error matrix, we can convert the block-Toeplitz matrix into a block-Toeplitz matrix with Hermitian-Toeplitz blocks. Although Akaike's algorithm can be used to invert this matrix, a more efficient inversion algorithm derived by Wax and Kailath in 1983 [20] could be used. This algorithm exploits the Toeplitz structure of the individual blocks in addition to the block-Toeplitz structure of the matrix and is found to outperform the Akaike algorithm by at least a factor of two. Since the error matrix is known, the true solution can then be derived using the Neumann series expansion around the computed inverse. Some results in the Wax-Kailath algorithm were also independently discovered by Kalopitsidis, Carayannis, and Manolakis in 1982 [21].

A detailed study of fast and super-fast algorithms for block-Toeplitz matrix inversion was presented by Wang and Krishna in 1989 [22]. A fast algorithm for solving block-Toeplitz systems with Toeplitz blocks was contributed by Yagle in 2001 [23]. Here, the solution of the linear system with the aforementioned structure is treated as a 2-D deconvolution problem and efficiency is attained by using the FFT to perform fast convolution. Computationally efficient inversion of a block-Toeplitz or nearly block-Toeplitz matrix is a topic of significant research interest because it is used in a variety of other applications such as multichannel filtering, equalization of time-varying multipath fading of wireless channels, and cochannel interference mitigation in cellular communication systems. In July 2008, a patent for an algorithm to invert a nearly block-Toeplitz matrix having a specific form was issued [24]. In this thesis, we implement the Akaike Algorithm and the Wax-Kailath algorithm for inverting a block-Toeplitz matrix and a block-Toeplitz matrix with Hermitian-Toeplitz blocks

respectively to compute the TVAR parameters.

As an illustration, we have considered the application of IF estimation for FM interference mitigation in a CPM communication system. Mitigating time-varying (nonstationary) narrowband FM interference is a problem of significant interest in the area of Wireless Communications. For instance, in direct sequence spread spectrum communications (DSSS), an FM signal can appear as strong narrowband interference and degrade the performance of the DSSS receiver [25]. Various methods based on the time-frequency distribution (TFD) of a signal have been proposed in the literature for excising nonstationary interference. Amin considered interference excision using FIR notch filters formed from Instantaneous Frequency (IF) estimates of the interference obtained from its TFD [26]. Chaparro and Suleesathira considered a combined discrete evolutionary and Hough transform based IF estimation and subsequent interference mitigation (IM) in a direct sequence spread spectrum communications system (DSSS) [27]. Zhu et al. proposed an adaptive hierarchical lapped orthogonal transform based interference excision in DSSS communications [28]. Recently, Francos and Porat presented an approach to design time-frequency (TF) filter banks for nonstationary signals and showed its application in canceling interference from a CPM signal [29].

Instead of using a non-parametric TFD or transform based method for IF estimation and IM, Shan and Beex suggested a parametric TVAR model based IF estimator for designing a time-varying FIR notch filter to mitigate FM interference in DSSS communications [25]. Although an FIR notch filter performs well in the presence of a strong interference, it causes significant distortion to the desired signal when the interference is weaker. To overcome this problem, a TVAR model-based Wiener filter was proposed for canceling FM interference with minimum signal distortion [30]. In this approach, it is not required to estimate the IF of the interfering signal explicitly. Instead, the time-varying coefficients of the TVAR model are used directly to obtain an interference canceling filter based on the Wiener solution.

In yet another related approach, the time-varying coefficients of the TVAR model were used to form a time-varying linear prediction error filter (TVLPEF) for mitigating the FM interference without explicit IF estimation [3]. Simulations performed on a DSSS communication system with BPSK modulation and spreading showed performance gains for either of the approaches over the FIR notch filter. Here, the DSSS signal used is minimally correlated. In fact, most works in the literature have considered the use of TVAR modeling for IF estimation of FM components and its mitigation in noise that has minimal or no correlation. However, when conventional TVAR modeling is applied to a signal consisting of FM components, white noise, and a signal of interest - such as a CPM signal - that is finitely correlated, IF estimation of the FM components is found to deteriorate and the TVLPEF formed from the TVAR coefficients heavily distorts the CPM signal while mitigating the FM interference.

We propose a decorrelating TVAR (DTVAR) model to address this issue [31]. In a DTVAR model, a decorrelation delay (gap) is introduced between the time-varying coefficients of the TVAR model. The decorrelation delay reduces the correlation between the current sample of the finitely correlated signal and the samples that are operated on by the filter to minimize its predictability and its effect on the IF estimate of the FM components. The time-varying coefficients of the DTVAR model form a generalized TVLPEF in which the delay between the predictor taps can be more than one. This TVLPEF can mitigate an FM interference without causing excessive cancellation of the finitely correlated signal of interest. At a given sample instant, the spectrum of a TVLPEF consists of extraneous notches in addition to a notch at the IF of the FM interference. This causes the separated signal of interest to be distorted. We propose a time-varying decision aided compensator (TVDAC) to undo this distortion and obtain a minimally distorted signal of interest [32].

1.3 Research Outline

The layout of the research is as follows. In Chapter 2, we introduce TVAR modeling and IF estimation using the time-varying coefficients of the model by polynomial rooting. In addition, we demonstrate the effectiveness of TVAR model-based IF estimation by applying it to estimate the IF of two linear FM components in white noise and a single non-linear FM component in white noise using short data records. A detailed derivation and implementation of the Akaike algorithm and the Wax-Kailath algorithm for practical realization of the generalized covariance system solution to compute the TVAR coefficients is presented in Chapter 3. In our derivation of the Wax-Kailath algorithm, we corrected errors in some expressions and ambiguity in a definition found in the original work by Wax and Kailath [20]. Finally, in Chapter 4, we present an application of IF estimation for FM interference mitigation in a CPM communication system. Realizing that a TVAR based IF estimation and IM perform poorly in a finitely correlated environment, we propose a decorrelating TVAR (DTVAR) model for IF estimation and a DTVAR model based linear prediction error filter for FM interference rejection.

Chapter 2

TVAR Modeling and IF Estimation

In this chapter, we first introduce autoregressive (AR) signal modeling and show how it can be extended to model time-varying signals. Next, instantaneous frequency (IF) estimation based on time-varying autoregressive (TVAR) modeling is presented. Finally, some example simulations are shown to demonstrate its workings.

2.1 AR Signal Modeling

In this section, we introduce autoregressive (AR) modeling for discrete-time stochastic processes. One of the reasons for the widespread use of AR modeling is that in many applications, the physical phenomena that generate the signal cause it to be autoregressive. Even when the signals themselves are not autoregressive, AR models have been found to represent many different types of signals with reasonable accuracy [33]. Another attractive feature of AR models is the Hermitian-Toeplitz structure found in the normal equations solved for obtaining the AR coefficients. This structure enables fast and efficient computation of the coefficients.

Consider a discrete-time wide sense stationary (WSS) and regular stochastic process x_n . A stochastic process x_n is called a regular process if its power spectrum $P_x(z)$ can be factored as shown in (2.1),

$$P_x(z) = \sigma_0^2 H(z) H^*(z^{-*}) \tag{2.1}$$

The regular process x_n possesses some interesting properties. For instance, the regular process can be generated by filtering white noise v_n of variance σ_0^2 using a linear causal and stable filter $H(z)$. This is referred to as the *innovations representation* of the process. Furthermore, filtering the process x_n using the inverse filter $1/H(z)$ generates the white noise v_n with the same variance. Here, the white noise v_n is referred to as the innovation. The filter $1/H(z)$ is called the whitening filter and generation of the white noise is referred to as the *whitening process*. These are illustrated in Figure 2.1.

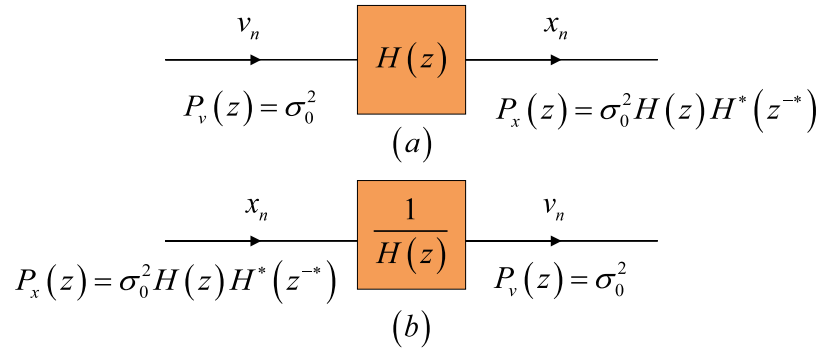


Figure 2.1: (a) Innovations Representation (b) Whitening Process.

In addition, if the process x_n has a rational power spectrum, then according to the Wold decomposition theorem [33] it can be modeled arbitrarily well as an AR process of increasing order p and generated by filtering a zero mean white noise v_n of variance σ_0^2 using an all-pole filter of the form given in (2.2).

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (2.2)$$

The rational power spectrum ensures that $P_x(z)$ can be factored as shown in (2.3) and makes AR modeling of the process feasible.

$$P_x(z) = \sigma_0^2 \frac{1}{A(z)} \frac{1}{A^*(z^{-*})} \quad (2.3)$$

Since the filter $H(z)$ in the innovations representation must be stable, the AR model parameters a_k must be such that $A(z)$ has all its roots inside the unit circle in the z -domain.

Now we show how the AR model parameters are computed. The input white noise v_n and the output stochastic process x_n are related by the linear constant coefficient difference equation (LCCDE) as shown below,

$$x_n + \sum_{k=1}^p a_k x_{n-k} = v_n \quad (2.4)$$

Multiplying both sides of (2.4) by x_{n-l}^* and applying the expected value operator ($E[\cdot]$), we have,

$$E[x_n x_{n-l}^*] + \sum_{k=1}^p a_k E[x_{n-k} x_{n-l}^*] = E[v_n x_{n-l}^*] \quad (2.5)$$

Since x_n is WSS we have,

$$E[x_{n-k} x_{n-l}^*] = r_{x,l-k} \quad (2.6)$$

Here, $r_{x,l}$ is the statistical autocorrelation of x_n . Denoting h_n as the causal impulse response of $H(z)$, the output process x_n can be expressed as,

$$x_n = h_n * v_n = \sum_{m=0}^{\infty} v_m h_{n-m} \quad (2.7)$$

Since the white noise v_n is WSS, x_n and v_n are jointly WSS. Therefore,

$$\begin{aligned} E[v_n x_{n-l}^*] &= r_{vx,l} = E \left[v_n \sum_{m=0}^{\infty} v_m^* h_{n-l-m}^* \right] && \text{(using (2.7))} \\ &= \sum_{m=0}^{\infty} E[v_n v_m^*] h_{n-l-m}^* \\ &= \sigma_0^2 \delta_{n-m} h_{n-l-m}^* \\ &= \sigma_0^2 h_{-l}^* \end{aligned} \quad (2.8)$$

Here $r_{vx,l}$ is the statistical cross-correlation between v_n and x_n . Using (2.6), (2.8), and the fact that $h_0^* = 1$, we rewrite (2.5) as,

$$r_{x,l} + \sum_{k=1}^p a_k r_{x,l-k} = r_{vx,l} = \begin{cases} \sigma_0^2 & l = 0 \\ 0 & l > 0 \end{cases} \quad (2.9)$$

The above equation is referred to as the *Yule-Walker Equation*. The YWE (Yule-Walker Equation) connects the statistical autocorrelation $r_{x,l}$ of the process x_n to the AR model

coefficients a_k . Thus, if the statistical autocorrelation of the process is known, the AR model parameters can be computed exactly using (2.9). However, in most practical scenarios, the statistical autocorrelation is unknown and must be estimated from a single realization of the process. If we assume that the process is mean and autocorrelation ergodic, we can estimate the statistical autocorrelation using a time average as shown below,

$$\hat{r}_{x,l} = \frac{1}{L} \sum_{\tau} x_n x_{n-l}^* \quad (2.10)$$

Here, L denotes the extent of the duration τ . Usually, a sample realization of the process is available only over a finite interval $[0, N - 1]$. In many applications, modeling has to be applied to overlapping or non-overlapping blocks of the signal having finite length due to computational limitations or because the signal is slowly time varying. Two of the approaches of signal modeling that arise when only finite data records are available or used are the *autocorrelation method* and the *covariance method*. In the autocorrelation method, it is assumed that the samples of the process x_n outside $[0, N - 1]$ are zero and the duration τ in (2.10) is chosen as $[0, N - 1 + p]$. On the other hand, the covariance method makes no assumptions about the samples outside the interval $[0, N - 1]$ and the duration τ in (2.10) is chosen as $[p, N - 1]$. The covariance method generally results in a less biased parameter estimate than the autocorrelation method; however, the resulting model is not guaranteed to be stable.

Now we look at the computational aspects of (2.9). Using the conjugate symmetry of the autocorrelation function, $r_{x,l} = r_{x,-l}^*$, we can write (2.9) in matrix form for $l = 1, 2, \dots, p$ as shown in (2.11).

$$\underbrace{\begin{bmatrix} r_{x,0} & r_{x,1}^* & r_{x,2}^* & \cdots & r_{x,p-1}^* \\ r_{x,1} & r_{x,0} & r_{x,1}^* & \cdots & r_{x,p-2}^* \\ r_{x,2} & r_{x,1} & r_{x,0} & \cdots & r_{x,p-3}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{x,p-1} & r_{x,p-2} & r_{x,p-3} & \cdots & r_{x,0} \end{bmatrix}}_{\mathbf{R}_x} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix}}_{\mathbf{a}_p} = - \underbrace{\begin{bmatrix} r_{x,1} \\ r_{x,2} \\ r_{x,3} \\ \vdots \\ r_{x,p} \end{bmatrix}}_{\mathbf{r}_x} \quad (2.11)$$

Note that the autocorrelation matrix \mathbf{R}_x is Hermitian-Toeplitz. It can be shown that if we replace the statistical autocorrelation $r_{x,l}$ with estimates $\hat{r}_{x,l}$ from (2.10) derived using the autocorrelation method, the Hermitian-Toeplitz structure is preserved in the resulting autocorrelation matrix estimate $\hat{\mathbf{R}}_x$. This structure allows for a fast $O(p^2)$ arithmetic operations) and recursive computation of the AR model parameters in (2.11) using the Levinson recursions [34]. Another approach to computing the AR model parameters is to invert the Hermitian-Toeplitz matrix. This can be efficiently realized using algorithms for Toeplitz matrix inversion contributed by Trench [17] and refined by Zohar [35]. However, if the statistical autocorrelation is estimated using the covariance method, the estimate $\hat{\mathbf{R}}_x$ is only Hermitian and close to Toeplitz. Although Levinson recursions or the Trench algorithm cannot be applied in this case, a fast covariance algorithm exists for computing the AR parameters [36]; however, it is found to be less efficient and more complicated when compared to the former. In the next section, we show how to extend the concepts introduced here for AR modeling to time-varying AR modeling.

2.2 Time-Varying Autoregressive (TVAR) Modeling

In the previous section, we considered the AR modeling of a wide sense stationary discrete-time stochastic process. Here, we present the AR modeling of nonstationary discrete-time stochastic processes. The resulting model is referred to as the TVAR model. Consider a discrete-time nonstationary stochastic process x_n . Similar to the innovations representation shown in Figure 2.1 for a stationary stochastic process, Cramer [37] showed that a nonstationary stochastic process can be generated by filtering an innovation v_n using a linear time-varying filter $H(z; n)$. In general, the impulse response of this system $h_{k,n}$ is time-varying and infinitely long. The existence of a finite-order AR model representation of this nonstationary process x_n is established by theorems contributed by Huang and Aggarwal [38] in 1980 and Grenier [39] in 1981. According to the theorems by Huang and Aggarwal, the impulse response $h_{k,n}$ can be equivalently represented by a distinct p^{th} order difference equation with unique time-varying coefficients $a_{k,n}$ and $b_{l,n}$ (moving average

part) as shown below,

$$\sum_{k=0}^p a_{k,n} x_{n-k} = \sum_{l=0}^{p-1} b_{l,n} v_{n-l} \quad (2.12)$$

Grenier looked at the conditions under which $H(z; n)$ can be expressed as a rational model and developed an alternative theorem that shows that the impulse response $h_{k,n}$ can be represented by an autoregressive moving average (ARMA) type recursion [9]. The above theorems ensure the existence and uniqueness of a p^{th} order AR representation for the nonstationary stochastic process x_n .

Now we show how the time-varying AR parameters $a_{k,n}$ can be estimated. In Section 2.1, we showed how the AR parameters of a stationary stochastic process can be computed from the autocorrelation function estimated using a time average (assuming that the process is ergodic). When the process is nonstationary, ergodicity assumptions cannot be applied and therefore a new assumption must be made to estimate $a_{k,n}$. Many assumptions exist in the literature [9] for tackling this problem. In this thesis, we assume that the time-varying coefficients $a_{k,n}$ can be approximated reasonably well as a weighted linear combination of a small set of known time functions referred to as the basis, an idea first introduced by Rao [10]. For instance, if we consider $\{1, n, n^2/2\}$ as the set of basis time functions, we can express the time-varying parameters as,

$$a_{k,n} = a_{k0} + a_{k1}n + a_{k2}\frac{n^2}{2} \quad (2.13)$$

Apart from time basis functions, various other bases such as a Fourier basis, Legendre Polynomials, etc. can be used to describe the time evolution of the coefficients [9]. The purpose of the basis is to allow time variation of the coefficients in a desired fashion. If we denote $u_{m,n}$ as the time basis function and consider a set of $(q + 1)$ functions for a given model, we can express the TVAR coefficients in general as,

$$a_{k,n} = \sum_{m=0}^q a_{km} u_{m,n} \quad (2.14)$$

From (2.14), we see that in order to compute the TVAR coefficients $a_{k,n}$, we need to compute the set of parameters a_{km} for $\{k = 1, 2, \dots, p; m = 0, 1, \dots, q; a_{0,n} \equiv 1\}$, and the TVAR

model is completely specified by this set.

Now we present the estimation of the parameters a_{km} . The time-varying filter $H(z; n)$ used for generating an autoregressive nonstationary stochastic process x_n by filtering a stationary zero mean white noise process v_n of variance σ_v^2 is shown below,

$$H(z; n) = \frac{1}{A(z; n)} = \frac{1}{1 + \sum_{k=0}^p a_{k,n} z^{-k}} \quad (2.15)$$

For the innovations representation using the above system, a p^{th} order TVAR representation of the process x_n is shown below,

$$x_n = -\sum_{k=1}^p a_{k,n} x_{n-k} + v_n \quad (2.16)$$

As we did for a stationary AR model, it is possible to formulate a Yule-Walker Equation for the time-varying AR model [9] and then derive a set of linear equations to be solved to obtain a_{km} . However, here we use a different approach to solving for the parameters a_{km} . We consider only a single realization of the process x_n and derive the results directly in terms of correlation estimates instead of statistical correlations. For a given realization of x_n , we can view (2.16) as a linear prediction error filter and consider the innovation to be the prediction error, $v_n = x_n - \hat{x}_n$, where,

$$\hat{x}_n \triangleq -\sum_{k=1}^p a_{k,n} x_{n-k} \quad (2.17)$$

Substituting for $a_{k,n}$ from (2.14) in (2.17), the total squared prediction error – which is also the error in modeling x_n – is now given by,

$$\epsilon_p = \sum_{\tau} \left| \underbrace{x_n + \sum_{k=1}^p \sum_{m=0}^q a_{km} u_{m,n} x_{n-k}}_{|v_n|} \right|^2 \quad (2.18)$$

The parameters a_{km} that minimize this modeling error can be found by setting the gradient of ϵ_p with respect to a_{lg}^* to zero [33],

$$\frac{\partial \epsilon_p}{\partial a_{lg}^*} = \sum_{\tau} \frac{\partial v_n v_n^*}{\partial a_{lg}^*} = \sum_{\tau} v_n \frac{\partial v_n^*}{\partial a_{lg}^*} = 0, \quad \{l = 1, 2, \dots, p; g = 0, 1, \dots, q\} \quad (2.19)$$

Now, $v_n^* = x_n^* + \sum_{l=1}^p \sum_{g=0}^q a_{lg}^* u_{g,n}^* x_{n-l}^*$ and the derivative of v_n^* with respect to a_{lg}^* is $u_{g,n}^* x_{n-l}^*$.

Therefore, (2.19) becomes,

$$\sum_{\tau} v_n u_{g,n}^* x_{n-l}^* = 0 \quad (2.20)$$

The above condition is equivalent to the *orthogonality principle* encountered in stationary signal modeling which states that the optimal vector \mathbf{v} of error samples is orthogonal to any vector \mathbf{x} of underlying data samples. Substituting for v_n in (2.20), from (2.18), we have,

$$\begin{aligned} \sum_{\tau} \left(x_n + \sum_{k=1}^p \sum_{m=0}^q a_{km} u_{m,n} x_{n-k} \right) u_{g,n}^* x_{n-l}^* &= 0 \\ \sum_{\tau} x_n u_{g,n}^* x_{n-l}^* + \sum_{k=1}^p \sum_{m=0}^q a_{km} \sum_{\tau} u_{m,n} x_{n-k} u_{g,n}^* x_{n-l}^* &= 0 \end{aligned} \quad (2.21)$$

Now we define a function $c_{mg}(l, k)$ as shown below,

$$c_{mg}(l, k) \triangleq \sum_{\tau} u_{m,n} x_{n-k} u_{g,n}^* x_{n-l}^* \quad (2.22)$$

Using the above definition in (2.21) we have,

$$\sum_{k=1}^p \sum_{m=0}^q a_{km} c_{mg}(l, k) = -c_{0g}(l, 0) \quad (2.23)$$

The above equation represents a system of $p(q+1)$ linear equations. Next, we show how this linear system of equations can be expressed compactly in matrix form. To facilitate this, we first define a column vector \mathbf{a}_m as follows,

$$\mathbf{a}_m = \begin{bmatrix} a_{1m} & a_{2m} & \cdots & a_{pm} \end{bmatrix}^T, \quad m = 0, 1, \dots, q \quad (2.24)$$

We now express the function in (2.22) in matrix form to obtain the following matrix,

$$\mathbf{C}_{mg} = \begin{bmatrix} c_{mg}(1, 1) & c_{mg}(1, 2) & \cdots & c_{mg}(1, p) \\ c_{mg}(2, 1) & c_{mg}(2, 2) & \cdots & c_{mg}(2, p) \\ \vdots & \vdots & \ddots & \vdots \\ c_{mg}(p, 1) & c_{mg}(p, 2) & \cdots & c_{mg}(p, p) \end{bmatrix}, \quad 0 \leq (m, g) \leq q \quad (2.25)$$

The above matrix is $p \times p$ and all the different values for m and g result in $(q + 1) \times (q + 1)$ such matrices. Using these matrices, we can now define a block matrix as shown below,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{00} & \mathbf{C}_{01} & \cdots & \mathbf{C}_{0q} \\ \mathbf{C}_{10} & \mathbf{C}_{11} & \cdots & \mathbf{C}_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{q0} & \mathbf{C}_{q1} & \cdots & \mathbf{C}_{qq} \end{bmatrix} \quad (2.26)$$

Matrix \mathbf{C} has $(q + 1) \times (q + 1)$ elements and each element is a $p \times p$ matrix resulting in a \mathbf{C} of size $p(q + 1) \times p(q + 1)$. Now we define a column vector \mathbf{d}_m as shown below,

$$\mathbf{d}_m = \left[c_{0m}(1, 0) \quad c_{0m}(2, 0) \quad \cdots \quad c_{0m}(p, 0) \right]^T, \quad m = 0, 1, \dots, q \quad (2.27)$$

Using the definitions in (2.24)-(2.27) we can write the linear system of equations of (2.23) in matrix form as,

$$\begin{bmatrix} \mathbf{C}_{00} & \mathbf{C}_{01} & \cdots & \mathbf{C}_{0q} \\ \mathbf{C}_{10} & \mathbf{C}_{11} & \cdots & \mathbf{C}_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{q0} & \mathbf{C}_{q1} & \cdots & \mathbf{C}_{qq} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_q \end{bmatrix}}_{\mathbf{a}} = - \underbrace{\begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_q \end{bmatrix}}_{\mathbf{d}} \quad (2.28)$$

Note that when $q = 0$, the above equation reduces to the YWE for a stationary AR model. The set of TVAR parameters a_{km} are elements of \mathbf{a} and can be computed by solving the above matrix equation.

In the analysis presented above, we have left the interval τ unspecified. If we consider a finite data record of length N and assume that the data can be extended by zero-padding outside $[0, N]$, we can set $\tau = [0, N - 1 + p]$ in the summation of (2.22) and get a correlation method for modeling the nonstationary stochastic process. In this case, equation (2.23) is referred to as the *generalized correlation equation* and $c_{mg}(l, k)$ in (2.22) is referred to as the *generalized correlation function*. The matrix \mathbf{C} in the correlation method is Hermitian and referred to as the *correlation matrix*.

On the other hand, if we make no assumptions on the data outside $[0, N]$ and set $\tau = [p, N - 1]$, we get a covariance method for modeling the process. In this case, equation (2.23) is referred to as the *generalized covariance equation* and $c_{mg}(l, k)$ in (2.22) is referred to as the *generalized covariance function*. The matrix \mathbf{C} in the covariance method is Hermitian block-Hankel and referred to as the *covariance matrix*. It is also possible to rewrite equation (2.28) and express the matrix \mathbf{C} as a block-Toeplitz matrix. In Section 2.1, we mentioned that a matrix with Hermitian-Toeplitz structure can be inverted efficiently using the Trench algorithm. Improved derivations for the formulas in the Trench algorithm were provided by Zohar. An extended version of Zohar's Toeplitz matrix inversion algorithm was proposed by Akaike [18] and can be used to invert the block-Toeplitz matrix recursively. We defer further discussion on efficient computation of the TVAR parameters a_{km} to Chapter 3 where we look at practical implementation aspects of IF estimation using TVAR modeling.

2.3 IF Estimation

In this section, we show how the instantaneous frequency (IF) of time-varying frequency components in a signal can be estimated using TVAR modeling. We consider a complex sinusoidal (cisoid) signal with a single linearly time-varying frequency modulated (FM) component as shown below,

$$x_n = e^{j2\pi(f_0 + \alpha n)n} \quad (2.29)$$

Here, the instantaneous frequency of the signal obtained by differentiating the angle is given by,

$$f_{i,n} = f_0 + 2\alpha n \quad (2.30)$$

where f_0 is the initial frequency and α is the rate of change of frequency. In general, an expression for IF $f_{i,n}$ as shown above is referred to as the IF law. It can be shown by substitution that the cisoid in (2.29) satisfies the following recursion,

$$x_n + a_{1,n}x_{n-1} = 0 \quad (2.31)$$

Here, the time-varying parameter $a_{1,n}$ is found to be,

$$a_{1,n} = -e^{j2\pi(f_0+2\alpha n-\alpha)} \quad (2.32)$$

Therefore, we have,

$$a_{1,n} \approx -e^{j2\pi(f_0+2\alpha n)} \quad (2.33)$$

If we consider the time-varying polynomial $A(z; n) = 1 + a_{1,n}z^{-1}$, we see that for (2.33) it has a root on the unit circle at an angle $2\pi(f_0 + 2\alpha n)$ for a given time instant n . Therefore, the instantaneous frequency $f_{i,n}$ can be approximated by dividing the angle of the root at each time instant by 2π . This analysis can be generalized to multiple cisoids as well as to signals that are not exactly cisoids but are narrowband processes with roots very close to the unit circle. In general, if we define r as the magnitude of the root, for r close to unity, we can approximately represent a narrowband process with a single FM component in white noise v_n using the following TVAR model,

$$x_n = re^{j2\pi(f_0+2\alpha n)}x_{n-1} + v_n \quad (2.34)$$

and approximate its instantaneous bandwidth from the angle of the time-varying root [13]. We now use the above analysis to formulate a method for IF estimation using the TVAR model.

2.3.1 Procedure

From the analysis presented above, we note that a complex exponential signal containing a single FM component can be modeled approximately as a first order TVAR process. In general, a complex exponential signal with M FM components in white noise can be modeled using order $p = M$ and a real sinusoidal signal using order $p = 2M$. The denominator polynomial $A(z; n)$ of the time-varying system function in (2.15) can be rooted to obtain p poles at each time instant n . The poles (P) close to or on the unit circle correspond to the FM components and the IF of these components can be approximated from the angles of these poles [4]. Using i to index the FM components we have,

$$\hat{f}_{i,n} = \angle P_{i,n} \quad \text{for } |P_{i,n}| \approx 1 \quad (2.35)$$

The steps for IF estimation based on TVAR modeling are outlined below [4],

1. Select a set consisting of $q + 1$ basis functions $u_{m,n}$ and decide on the order p of the TVAR model.
2. Choose between correlation and covariance methods to decide on the interval τ in (2.22) and compute $c_{mg}(l, k)$ to obtain the matrix \mathbf{C}_{mg} in (2.25). Next, set up the matrix \mathbf{C} in (2.26). Also, use $c_{mg}(l, k)$ to compute \mathbf{d}_m in (2.27).
3. Now compute the TVAR parameters a_{km} by solving (2.28) and form the coefficients $a_{k,n}$ using (2.14).
4. Root the denominator polynomial $A(z; n)$ to obtain the time-varying poles $P_{i,n}$.
5. For every sample instant n , find the angles of the poles $P_{i,n}$ close to the unit circle and divide by 2π to obtain IF estimates of the FM components.

In this thesis, we use basis functions that are powers of the time variable n as given below,

$$u_{m,n} = \left(\frac{n}{N}\right)^m \quad (2.36)$$

Here, N denotes the length of the interval τ and is used to normalize the $u_{m,n}$ amplitudes so that they don't exceed unity. This particular choice of basis functions results in a covariance matrix \mathbf{C} that has a block-Hankel structure which can be exploited for computationally more efficient solutions. It is found that for a complex exponential signal with M FM components, a model order $p > M$ results in good estimates and the choice of basis function order q depends on the rate of change of frequency α as well as on the order of the IF law [4, 13]. Furthermore, the performance of the above IF estimation method was found to degrade at low SNRs.

For moderate to low SNRs, a time-varying Prony method contributed by Beex and Shan [15] can be used to improve the IF estimate. In this method, we set the order used to model a complex signal with M FM components as $p > M$. Thus, M of the p poles resulting during a sample instant correspond to the FM components and the remaining $(p - M)$ poles

correspond to noise. For the different sample instants n , we get a trajectory for the FM component poles and noise poles. The FM component poles will have a trajectory close to the unit circle whereas the noise pole trajectory can be random and be inside or outside the unit circle. To estimate the IF, we now need to know which poles correspond to the FM components. To distinguish between poles of FM components and poles of noise, a subset selection method can be used. In the most trivial case, when there are exactly M poles close to the unit circle and the remaining $(p - M)$ poles are far away, the desired FM component subset at each time instant can be formed by choosing the M poles closest to the unit circle. In other cases a criterion based on the instantaneous squared prediction error can be used [15]. Next, we show some example simulations to demonstrate the workings of IF estimation using TVAR modeling.

2.3.2 Linear FM Signal

In this section, we show TVAR based IF estimation of a signal containing two linear FM components in white noise. The IF laws of the FM components in the signal are,

$$\begin{aligned} f_{1,n} &= 0.05 + 2(0.0015)n \\ f_{2,n} &= 0.2 + 2(0.002)n \end{aligned} \quad (2.37)$$

We consider a data record of $N = 64$ samples and a sample rate of 1 Hz. Therefore, $n = 0, 1, 2, \dots, 63$ and the two FM components chirp in frequency from 0.05 Hz to 0.239 Hz and from 0.2 Hz to 0.452 Hz respectively. The FM signal generated is given below,

$$x_n = \sqrt{0.5\sigma_s^2} \left(e^{j2\pi[(0.05+(0.0015)n]n+\phi_1]} + e^{j2\pi[(0.2+(0.002)n]n+\phi_2]} \right) \quad (2.38)$$

Here, σ_s^2 is the signal power and $2\pi\phi_1$ and $2\pi\phi_2$ are random initial phases. The above signal is corrupted by a complex additive white gaussian noise (AWGN) at SNR = 20 dB. The data record is modeled using a TVAR model of order $p = 2$ and $q = 3$. Also, we use the covariance method of signal modeling and set $\tau = [p, N - 1]$ resulting in $n = 2, 3, \dots, 63$. For the covariance method, we make a slight modification in the basis function definition of

(2.36) as shown below,

$$u_{m,n} = \left(\frac{n-p}{N} \right)^m \quad m = 0, 1, \dots, q; \quad n = p, p+1, \dots, N-1 \quad (2.39)$$

For $p = 2$, $q = 3$, and $N = 64$ plot of the above basis functions is shown in Figure 2.2.

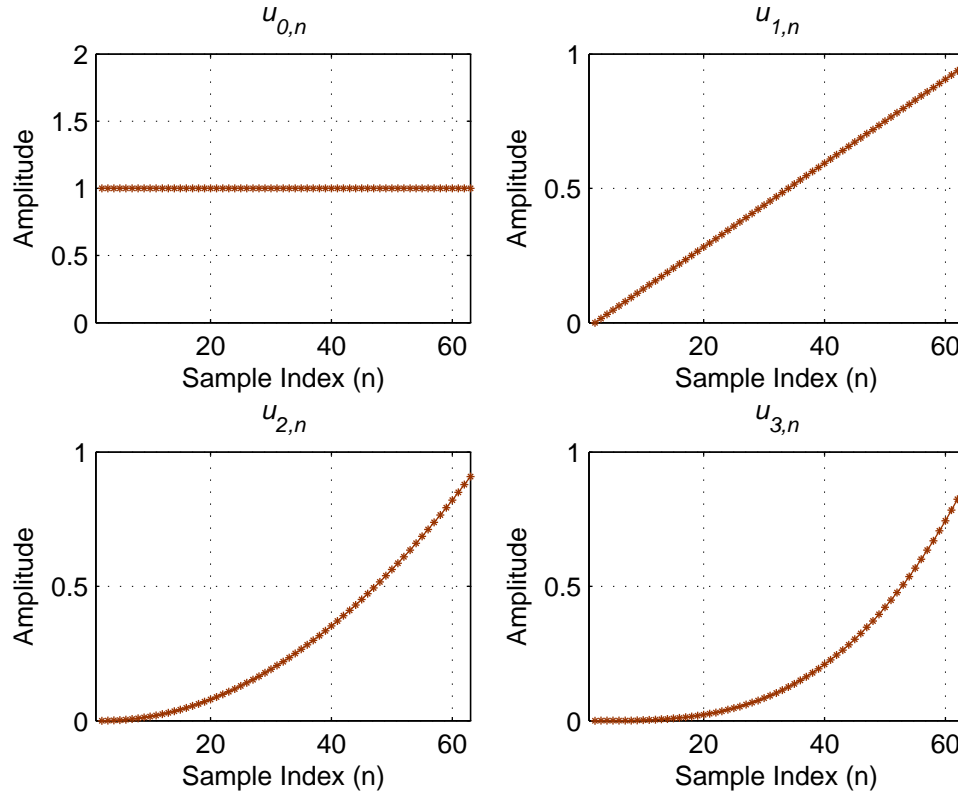


Figure 2.2: The Basis Function Set $u_{m,n}$ as in (2.39).

Next, we compute $c_{mg}(l, k)$ to obtain the matrix \mathbf{C}_{mg} in (2.25) and use it to set up the matrix \mathbf{C} in (2.28). The matrix \mathbf{C} that is obtained is given below.

$$\begin{bmatrix} 61.3 + 0.00j & 4.29 - 47.5j & 29.76 + 0.00j & -3.3 - 23.1j & 19.07 + 0.00j & -3.9 - 14.7j & 13.75 + 0.00j & -3.62 - 10.5j \\ 4.29 + 47.5j & 63.03 + 0.00j & -3.3 + 23.13j & 30.6 + 0.00j & -3.92 + 14.71j & 19.92 + 0.00j & -3.62 + 10.5j & 14.57 + 0.00j \\ 29.76 + 0.00j & -3.3 - 23.1j & 19.07 + 0.00j & -3.9 - 14.7j & 13.75 + 0.00j & -3.62 - 10.5j & 10.56 + 0.00j & -3.187 - 8.05j \\ -3.3 + 23.1j & 30.6 + 0.00j & -3.92 + 14.71j & 19.9 + 0.00j & -3.62 + 10.5j & 14.57 + 0.00j & -3.18 + 8.05j & 11.36 + 0.00j \\ 19.07 + 0.00j & -3.9 - 14.7j & 13.75 + 0.00j & -3.62 - 10.5j & 10.56 + 0.00j & -3.18 - 8.05j & 8.449 + 0.00j & -2.78 - 6.429j \\ -3.9 + 14.7j & 19.9 + 0.00j & -3.62 + 10.5j & 14.57 + 0.00j & -3.18 + 8.05j & 11.363 + 0.00j & -2.78 + 6.429j & 9.223 + 0.00j \\ 13.75 + 0.00j & -3.62 - 10.5j & 10.56 + 0.00j & -3.18 - 8.05j & 8.449 + 0.00j & -2.78 - 6.429j & 6.948 + 0.00j & -2.44 - 5.28j \\ -3.62 + 10.5j & 14.57 + 0.00j & -3.18 + 8.053j & 11.36 + 0.00j & -2.784 + 6.429j & 9.223 + 0.00j & -2.44 + 5.28j & 7.697 + 0.00j \end{bmatrix}$$

Also, matrix \mathbf{d} in (2.28) is then as follows,

$$\mathbf{d} = \begin{bmatrix} 3.5675 + 46.5914j \\ -17.2904 + 6.3899j \\ -3.2677 + 22.2642j \\ -7.9718 - 1.0323j \\ -3.7151 + 13.8708j \\ -4.5875 - 1.9548j \\ -3.3364 + 9.7274j \\ -2.9881 - 1.8858j \end{bmatrix}$$

Now we compute the TVAR parameters a_{km} by solving (2.28) and form the coefficients $a_{k,n}$ using (2.14). The matrix \mathbf{a} containing elements a_{km} is given below.

$$\mathbf{a} = \begin{bmatrix} -1.1543 - 1.2532j \\ -0.0038 + 0.9057j \\ 1.9778 - 1.2749j \\ -3.1212 + 0.0407j \\ 0.8098 + 0.8450j \\ 2.0908 - 4.9609j \\ -0.6912 + 0.5728j \\ 0.8997 + 3.0929j \end{bmatrix}$$

A plot of the TVAR coefficients $a_{k,n}$ is shown in Figure 2.3. The denominator polynomial $A(z; n)$ is now rooted to obtain the time-varying poles $P_{i,n}$. The trajectory of the poles is shown in Figure 2.4.

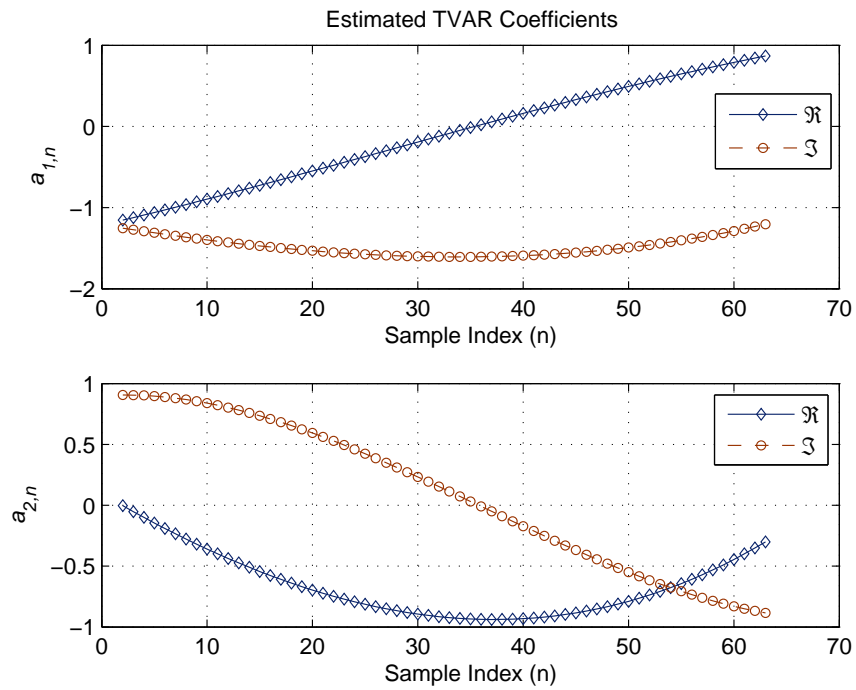


Figure 2.3: Estimated TVAR Coefficients $a_{k,n}$ for a Linear FM Signal.

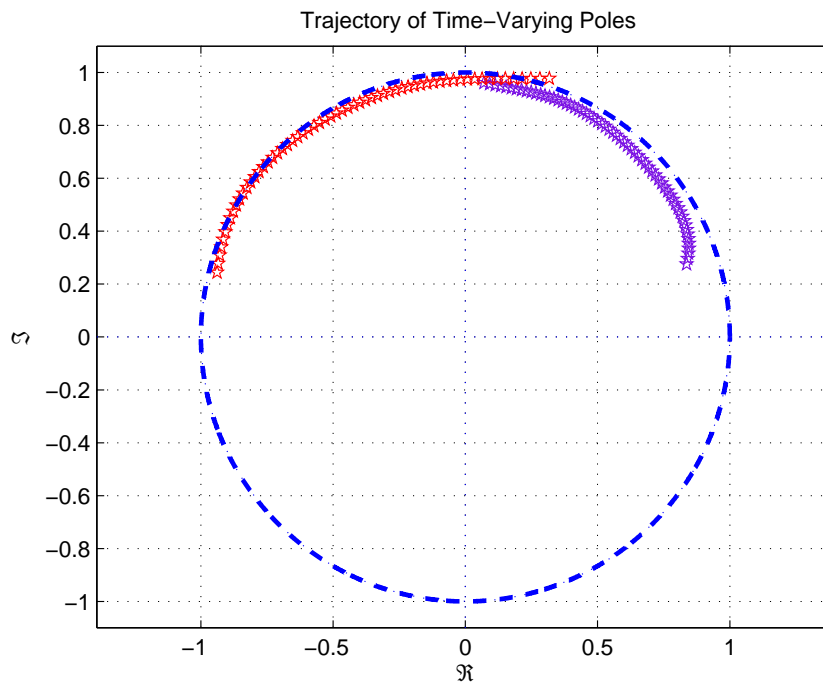


Figure 2.4: Trajectory of Time-Varying Poles for a Linear FM Signal.

From Figure 2.4, we see that the poles are close to the unit circle as expected. For each sample instant n , we now find the angles of the poles and divide by 2π to obtain the IF estimates of the FM components. The true IF and estimated IF of the FM components are shown in Figure 2.5. From Figure 2.5, we see that the TVAR based method has resulted in very good IF estimation. The mean square error (MSE) between the true IF $f_{i,n}$ and estimated IF $\hat{f}_{i,n}$ for $n = 2, 3, \dots, 63$ is computed to be $1.3230 \times 10^{-5} = -48.7844$ dB for $i = 1$ and $6.4758 \times 10^{-6} = -51.8871$ dB for $i = 2$. We note here that the estimates are not available for the interval $[0, p - 1] = [0, 1]$ in the covariance method.

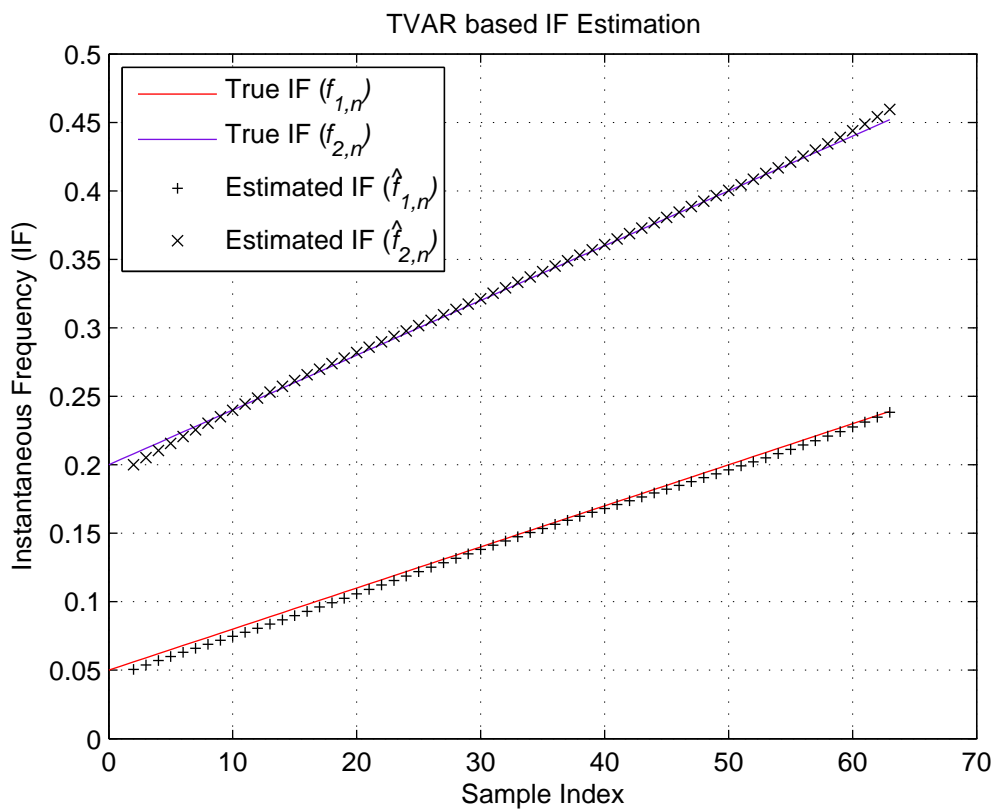


Figure 2.5: True and Estimated IF of Linear FM components.

The TVAR coefficients $a_{k,n}$ can also be used to predict the nonstationary process x_n using (2.17). For the covariance method, the parameters $a_{k,n}$ can be computed only for $n = 2, 3, \dots, 63$. Therefore, the prediction \hat{x}_n is also available only for this interval. The samples x_0 and x_1 in this simulation example are used as initial conditions for the

time-varying prediction error filter. The original data record x_n and the TVAR prediction are shown in Figure 2.6.

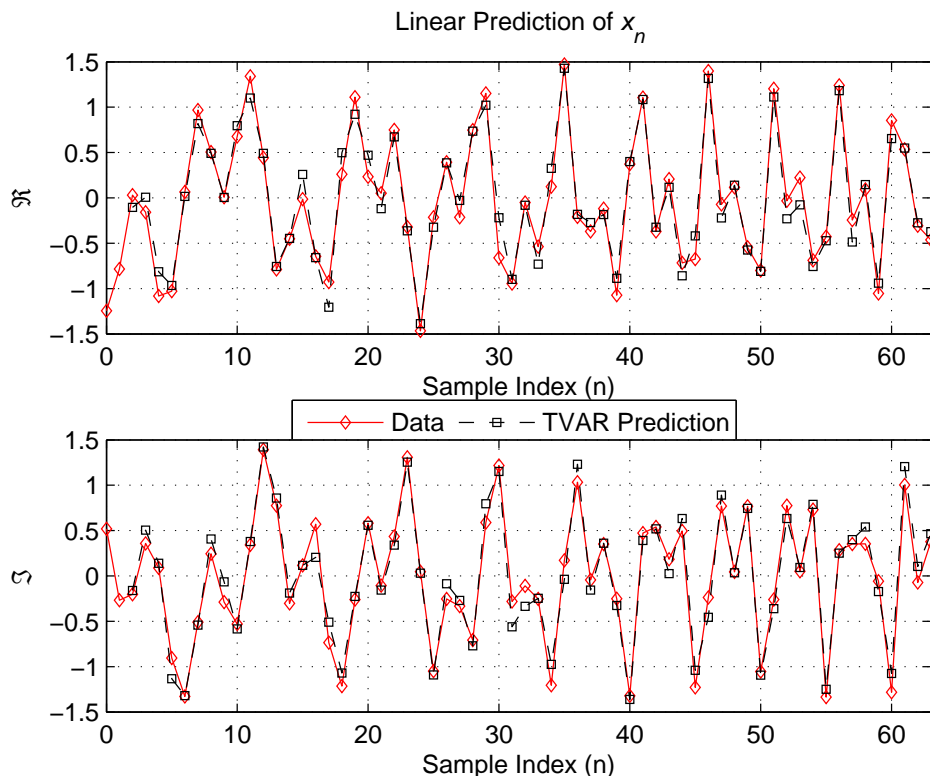


Figure 2.6: Comparison of Original Data and TVAR Prediction for Linear FM Signal.

From Figure 2.6, we see that the TVAR model has successfully predicted x_n . The average squared prediction error (ASPE) is 0.0403 which is close to the noise variance of 0.0098. Although TVAR based IF estimation does not use a TFD or the time-varying power spectrum, it is useful to look at the TFD obtained from the TVAR model. The time-varying power spectral density is given by,

$$P_x(z; n) = \frac{1}{|A(z; n)|^2} \quad (2.40)$$

A plot of the time-frequency distribution for the TVAR model obtained is shown in Figure 2.7. At each sample instant, the TFD is expected to have peaks at the IF estimates at that instant. To illustrate this, we also show the corresponding flat time-frequency view of the TFD in Figure 2.8.

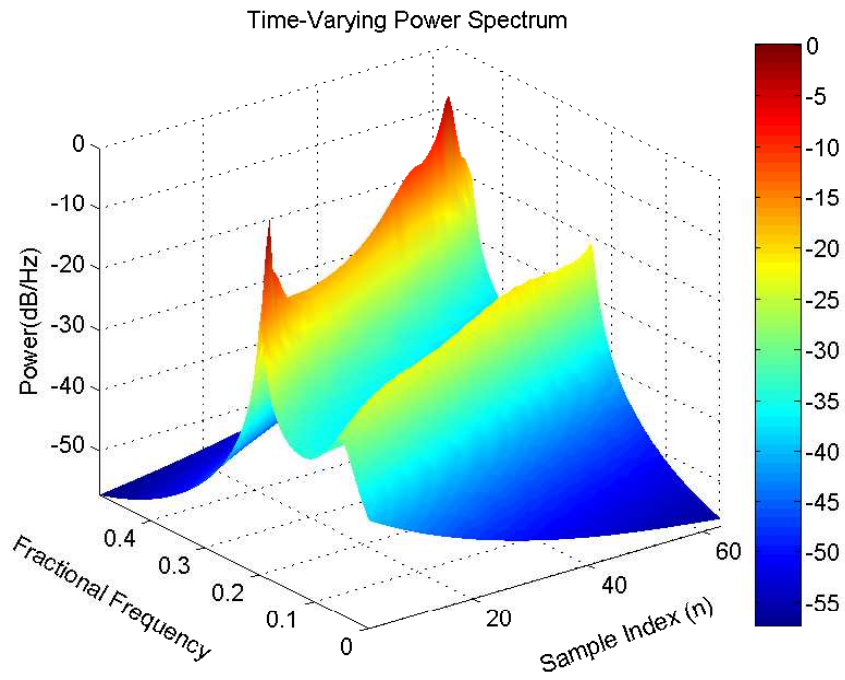


Figure 2.7: Time Varying Power Spectrum of the TVAR Model.

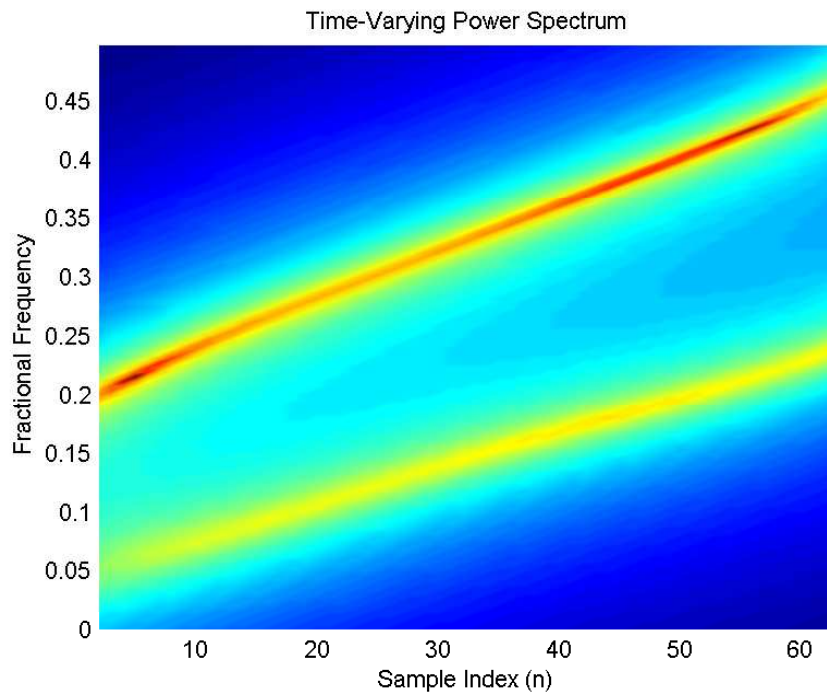


Figure 2.8: Time-Frequency View of the TFD.

In Figure 2.7, the colorbar indicates the power in dB/Hz with yellow-red representing peaks or high values and cyan-blue representing low values. Here, we see peaks corresponding to the two dominant FM components. The time-frequency view of the TFD in Figure 2.8 clearly shows that the TFD peaks are consistent with the IF estimate and the true IF shown in Figure 2.5. The presented example shows that a TVAR based method works well for signals containing multiple FM components obeying a linear IF law. Next, we show an example simulation for TVAR based IF estimation for a signal containing a non-linear FM component.

2.3.3 Non-linear FM Signal

Many of the IF estimation techniques such as the PLL and adaptive filtering approaches try to track the IF. Generally, tracking approaches take time to converge and cannot respond to rapid and/or non-linear frequency variations. In this section, we demonstrate that the TVAR based IF estimation is very effective for practical situations where the IF law is non-linear. We consider a signal having a single FM component with a sinusoidally varying frequency. The non-linear IF law is given by,

$$f_n = 0.15 + 0.02(0.1n\cos(0.1n) + \sin(0.1n)) \quad (2.41)$$

We consider a data record of $N = 64$ samples and a sample rate of 1 Hz. The FM signal generated is given below,

$$x_n = \sqrt{\sigma_s^2} \left(e^{j2\pi[(0.15+0.02\sin(0.1n))n+\phi]} \right) \quad (2.42)$$

Here, σ_s^2 is the signal power and $2\pi\phi$ is some random initial phase. The above signal is corrupted by a complex additive white gaussian noise (AWGN) at SNR = 20 dB. Since the IF law is non-linear and has a higher order than in the previous example, we use a TVAR model of order $p = 2$ and $q = 5$. Although we use a high SNR, the extra pole is used to capture some noise and improve the IF estimate as well as the linear prediction of x_n . The basis functions used in this simulation have the same form as in (2.39) except that here we

have a set of 6. A plot of the TVAR coefficients $a_{k,n}$ obtained for this signal is shown in Figure 2.9. A plot of the trajectory of time-varying poles is shown in Figure 2.10.

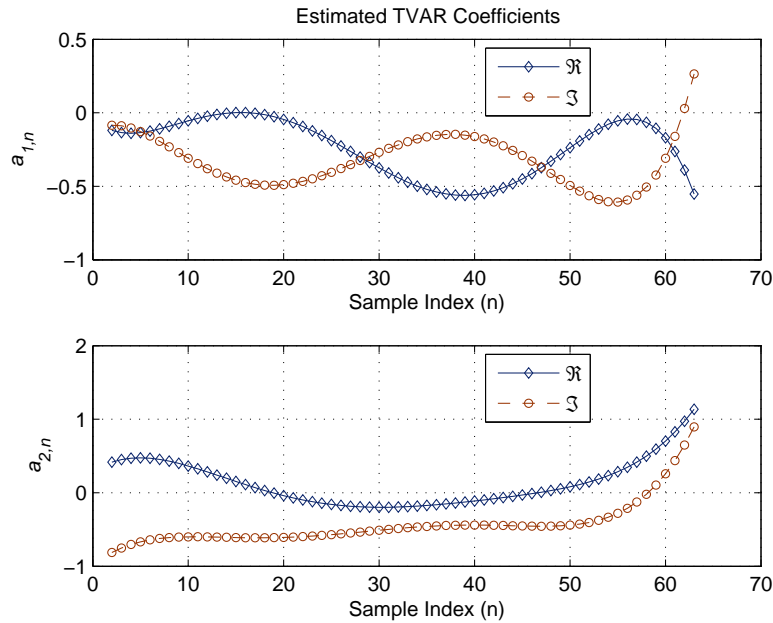


Figure 2.9: Estimated TVAR Coefficients $a_{k,n}$ for a Non-linear FM Signal.

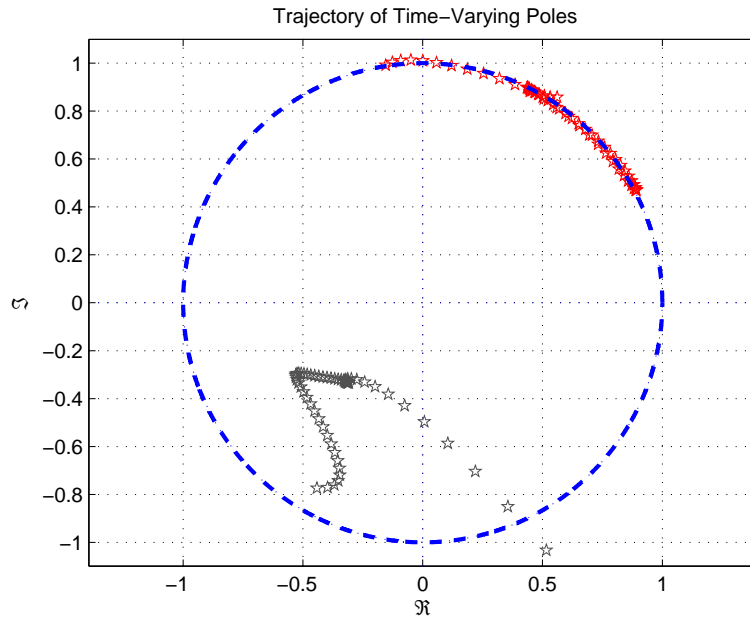


Figure 2.10: Trajectory of Time-varying Poles for a Non-linear FM Signal.

From the trajectory of the time-varying poles shown in Figure 2.10, we see that poles corresponding to the FM components (red) are close to the unit circle whereas poles that correspond to noise (grey) are randomly distributed, mostly away from the unit circle. The true and estimated IF of the FM component is shown in Figure 2.11.

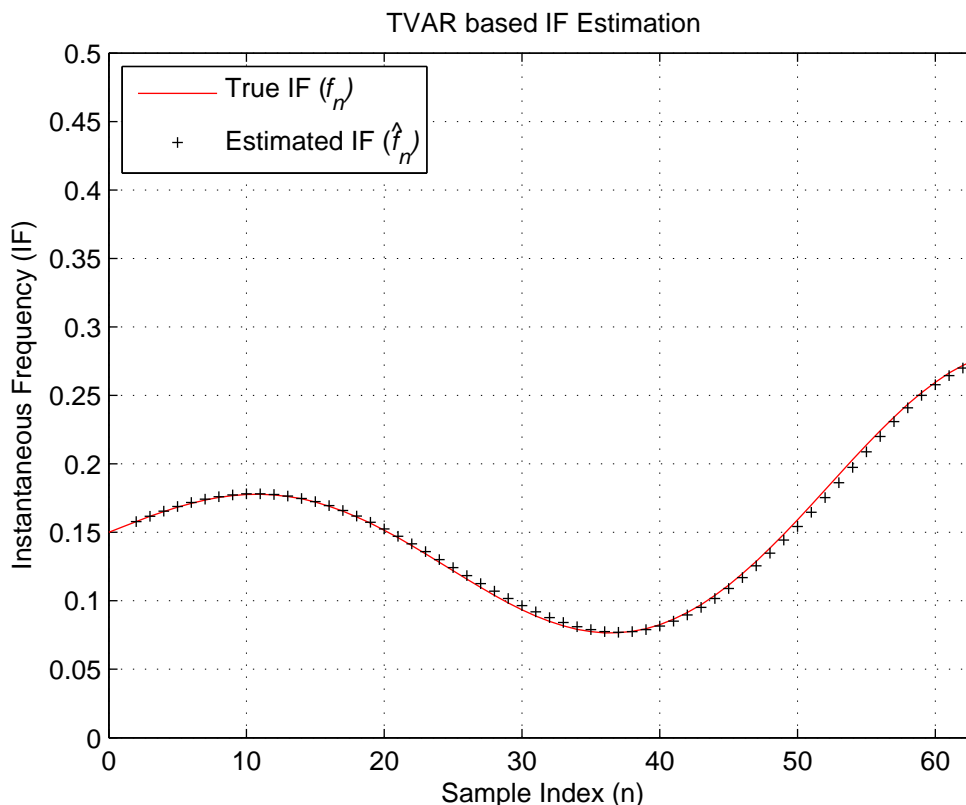


Figure 2.11: True and Estimated IF of Non-linear FM component.

From Figure 2.11, we see that the TVAR based method has resulted in very good IF estimation. The mean square error (MSE) between the true IF f_n and estimated IF \hat{f}_n for $n = 2, 3, \dots, 63$ is computed to be $6.3851 \times 10^{-6} = -51.9483$ dB. The original data record x_n and the TVAR prediction are shown in Figure 2.12. The figure shows successful prediction of x_n with an ASPE= 0.0126 which is close to the noise variance of 0.0098. A time-frequency view of the TFD is shown in Figure 2.13 to illustrate that the TFD peaks are consistent with the estimated IF and the true IF shown in Figure 2.11.

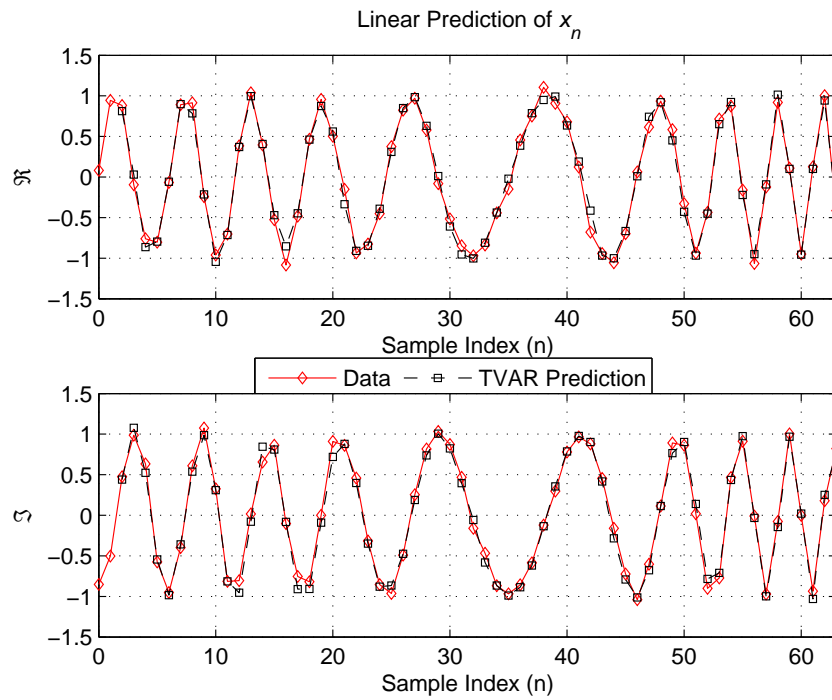


Figure 2.12: Comparison of Original Data and TVAR Prediction for Non-linear FM Signal.

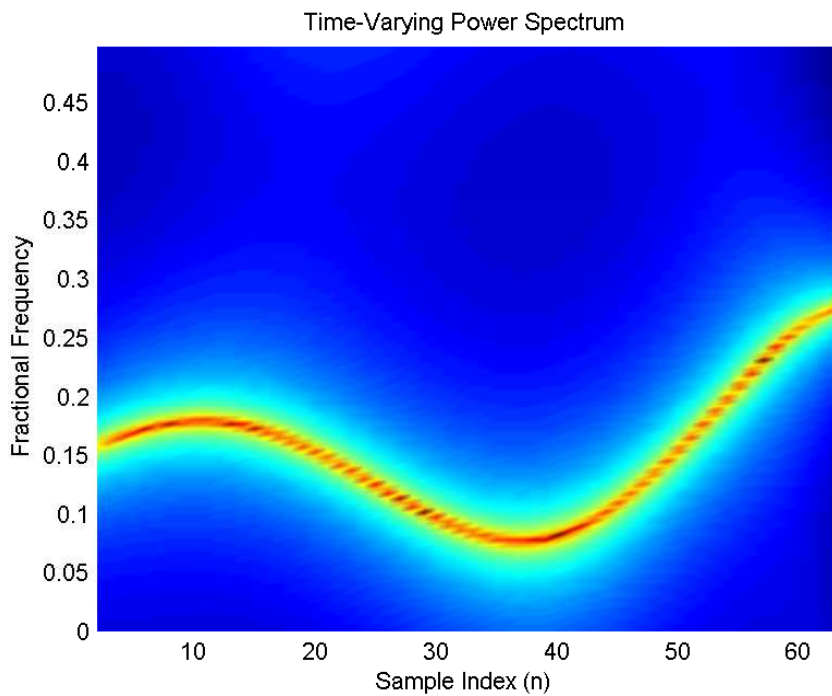


Figure 2.13: Time-Frequency View of the TFD of Non-linear FM Signal.

2.3.4 Autocorrelation Method

In the simulation examples presented in Sections 2.3.2 and 2.3.3, we have used the covariance method of signal modeling. Here, we present the corresponding simulation results obtained using the autocorrelation method. For the autocorrelation method of signal modeling, we set $\tau = [0, N - 1]$ resulting in $n = 0, 1, \dots, 63$ for a data record of $N = 64$ samples. The basis functions defined in (2.36) are used for the autocorrelation method. For the linear FM signal described in Section 2.3.2, the trajectory of the poles obtained using the autocorrelation method is shown in Figure 2.14.

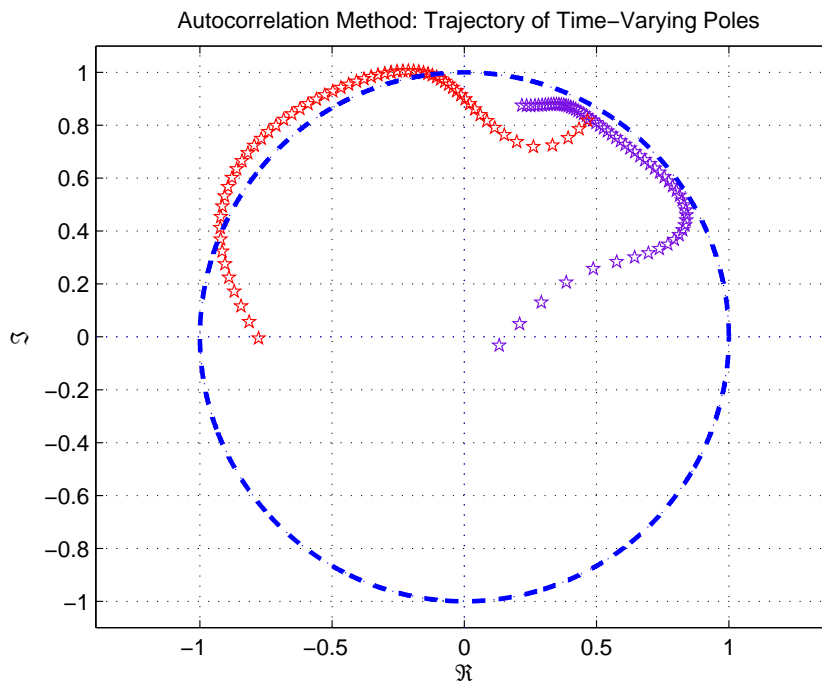


Figure 2.14: Trajectory of Time-Varying Poles for a Linear FM Signal using Autocorrelation Method.

From Figure 2.14, we see that the poles move often outside the unit circle. In the covariance method as well, it is possible for the poles to wander outside the unit circle. Although this would indicate instability in the case of a time-invariant filter, this is not necessarily true for a time-varying filter. In cases where the poles temporarily move outside the unit circle, the impulse response often remains bounded but excessively large [12]. Detailed

analysis regarding stability of time-varying filters is beyond the scope of this thesis and will be omitted. We see from Figure 2.4 that for the covariance method, even if there are poles outside the unit circle some of the time, they remain very close to it unlike for the autocorrelation method where they are further away from the unit circle. Therefore, in the remainder of this thesis, we consider only the covariance method of signal modeling and assume that the poles of the resulting time-varying filter will remain inside the unit circle or very close to it.

Another disadvantage of the autocorrelation method that justifies the use of the covariance method of signal modeling is that the autocorrelation method results in a more biased IF estimate than the covariance method. The true IF and estimated IF of the linear FM components using the autocorrelation method are shown in Figure 2.15.

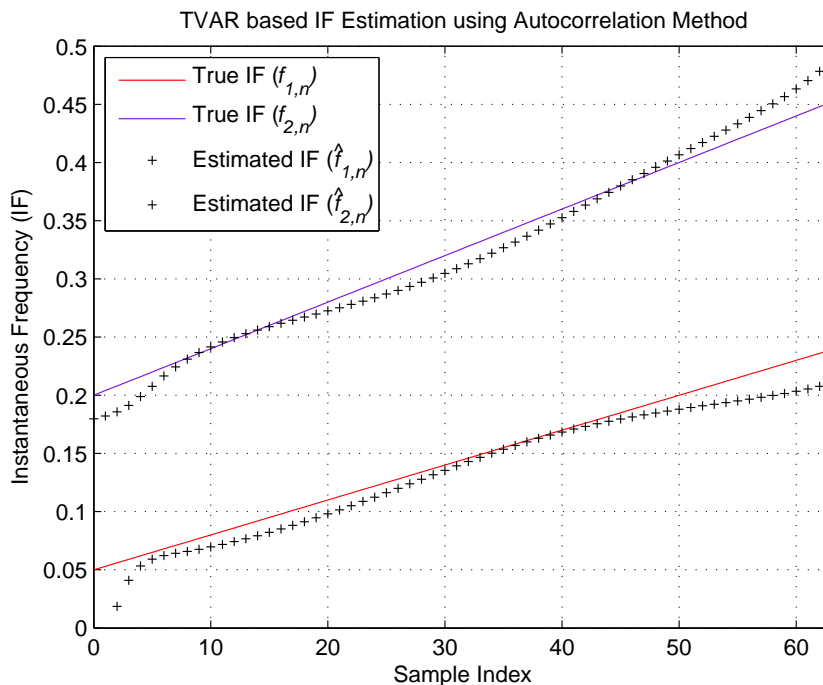


Figure 2.15: True and Estimated IF of FM Components using Autocorrelation Method.

From Figure 2.15, we see that the IF estimates are poor compared to the results obtained for the covariance method as shown in Figure 2.5. In particular, we observe that the estimates toward the beginning and end of the data record are worse and the best estimates are found

for the middle of the block. This is due to the assumption in the autocorrelation method that the data samples outside the range $\tau = [0, N - 1]$ are zeros. As a result, the modeling error (also the prediction error) v_n will be larger for $n < p$ and $n > N - p$ because it will be computed using fewer than p samples [33].

For the non-linear FM signal in Section 2.3.3, the trajectory of the time-varying poles is shown in Figure 2.16, and the true and estimated IF of the FM component are shown in Figure 2.17. From Figure 2.16, we see that some of the poles corresponding to the FM components (red) move outside the unit circle and the poles corresponding to the noise (grey) are randomly distributed and most lie far away from the unit circle. However, in the pole trajectory shown in Figure 2.10 using the covariance method of signal modeling, we see that the poles mostly remain within the unit circle and when they move outside the unit circle remain very close to it. Comparing the IF estimates in Figure 2.17 for the autocorrelation method to the IF estimates shown in Figure 2.11 for the covariance method, we see that the covariance method performs much better.

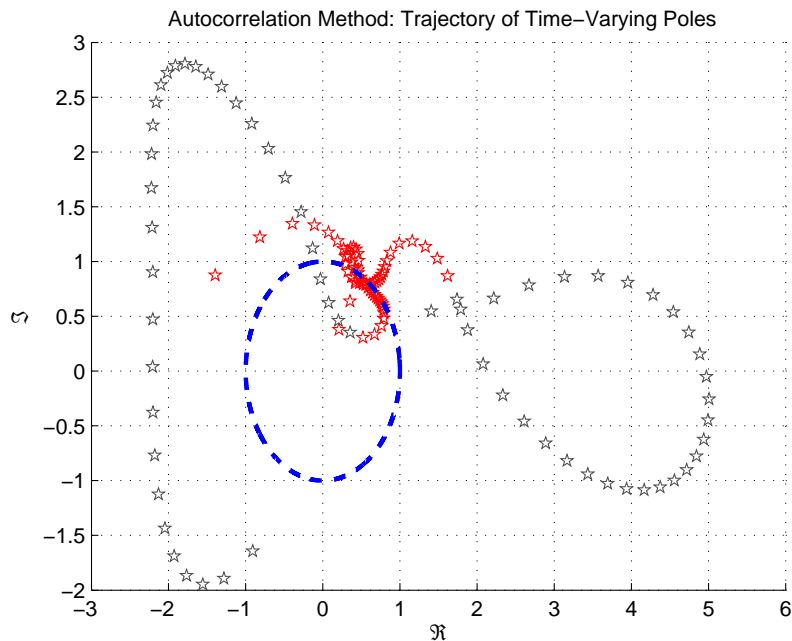


Figure 2.16: Trajectory of Time-Varying Poles for a Non-linear FM Signal using Autocorrelation Method.

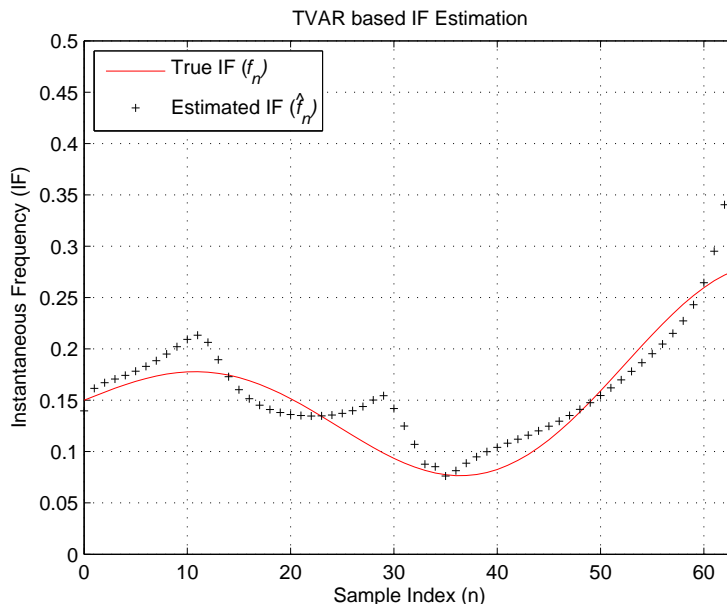


Figure 2.17: True and Estimated IF of Non-linear FM Component using Autocorrelation Method.

2.4 Summary

In this chapter, we first introduced AR modeling for stationary signals and then showed how the approach can be extended to model nonstationary signals by making the model time-varying. Next, we looked at the IF estimation procedure based on TVAR modeling and presented some example simulations to illustrate that the method works well for linear and non-linear IF laws as well as for short data records. We justified the selection of the covariance method of signal modeling over the autocorrelation method for further analysis. One of the challenging aspects in implementing IF estimation based on TVAR modeling is the computation of the TVAR parameters a_{km} by solving $\mathbf{C}\mathbf{a} = -\mathbf{d}$. In our simulations, we solved this linear system using the MATLAB command $\mathbf{a} = -\mathbf{C} \backslash \mathbf{d}$. For $\mathbf{C}_{p(q+1) \times p(q+1)}$, the \backslash operator in MATLAB solves the linear system using Gaussian elimination which requires $O(p^3(q+1)^3)$ computations and is not an efficient way to solve the linear system. From an implementation point of view, it is desirable to have recursive solutions to this linear system. We address the practical implementation aspects of IF estimation - which is the core of this research - in the next chapter.

Chapter 3

Efficient Matrix Inversion for TVAR

IF Estimation

In this chapter, we consider efficient and practical methods to solve the system of equations needed for IF estimation based on TVAR modeling. The Akaike algorithm and the Wax-Kailath algorithm for efficient inversion of a covariance matrix are presented which can lead to efficient estimation of the time-varying coefficients of a TVAR model.

3.1 Covariance Matrix Inversion

We have seen that estimating the time-varying parameters of the TVAR model is the essence of implementing IF estimation. To estimate the TVAR parameters, the system of equations $\mathbf{C}\mathbf{a} = -\mathbf{d}$ needs to be solved. Therefore, recursive and efficient inversion of \mathbf{C} is required for practical implementation of IF estimation. Interestingly, the covariance matrix \mathbf{C} is found to have a Hermitian block-Hankel structure. Rewriting the system of equations, we can convert this block-Hankel structure to a block-Toeplitz structure. Furthermore, we can express a block-Toeplitz covariance matrix \mathbf{C} – with individual blocks that are Hermitian but not Toeplitz in general – as the difference between a block-Toeplitz matrix with Hermitian-Toeplitz blocks and an error matrix. Noticing these special structures, we propose two approaches to efficiently invert the covariance matrix. In the first approach,

we use the Akaike algorithm to recursively compute the inverse of a general block-Toeplitz matrix \mathbf{C} . In the second approach, we first devise an appropriate error matrix \mathbf{E} and add it to \mathbf{C} to obtain a block-Toeplitz matrix \mathbf{T} with Hermitian-Toeplitz blocks. Next, we use the Wax-Kailath algorithm to efficiently invert the matrix \mathbf{T} . Using the inverse of \mathbf{T} and the known error matrix \mathbf{E} , we can use the Neumann series presented in Section 3.3.4 to compute the inverse of \mathbf{C} . The algorithms are presented in the next two sections.

3.2 Akaike Algorithm

In this section, we introduce the Akaike algorithm for inverting a general block-Toeplitz matrix. Consider a $(q + 1) \times (q + 1)$ block matrix \mathbf{C} with individual blocks of size $p \times p$. Let the elements (blocks) of the matrix \mathbf{C} be denoted as \mathbf{C}_{mg} . Then, the matrix \mathbf{C} is called a general block-Toeplitz matrix if all block matrices along each of the diagonals parallel to the main diagonal are equal, i.e.,

$$\mathbf{C}_{mg} = \mathbf{C}_{m+1,g+1} \quad 0 \leq (m, g) \leq q \quad (3.1)$$

The covariance matrix \mathbf{C} we obtain has a Hermitian block-Hankel structure. To convert \mathbf{C} to a block-Toeplitz structure, we rewrite equation (2.28) as shown below,

$$\underbrace{\begin{bmatrix} \mathbf{C}_{0q} & \mathbf{C}_{0q-1} & \cdots & \mathbf{C}_{00} \\ \mathbf{C}_{1q} & \mathbf{C}_{1q-1} & \cdots & \mathbf{C}_{10} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{qq} & \mathbf{C}_{qq-1} & \cdots & \mathbf{C}_{q0} \end{bmatrix}}_{\mathbf{R}_{q+1}} \underbrace{\begin{bmatrix} \mathbf{a}_q \\ \mathbf{a}_{q-1} \\ \vdots \\ \mathbf{a}_0 \end{bmatrix}}_{\check{\mathbf{a}}} = - \underbrace{\begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_q \end{bmatrix}}_{\mathbf{d}} \quad (3.2)$$

The subscript $(q + 1)$ indicates the order of the matrix. Since \mathbf{R}_{q+1} has a block-Toeplitz structure, we rewrite it as shown below.

$$\mathbf{R}_{q+1} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{-1} & \mathbf{C}_{-2} & \cdots & \mathbf{C}_{-q} \\ \mathbf{C}_1 & \mathbf{C}_0 & \mathbf{C}_{-1} & \cdots & \mathbf{C}_{-q+1} \\ \mathbf{C}_2 & \mathbf{C}_1 & \mathbf{C}_0 & \cdots & \mathbf{C}_{-q+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_q & \mathbf{C}_{q-1} & \mathbf{C}_{q-2} & \cdots & \mathbf{C}_0 \end{bmatrix} \quad (3.3)$$

For the system of equations in (3.2), the block-Toeplitz matrix is not block-Hermitian, i.e., \mathbf{C}_{-i} and \mathbf{C}_i for $i = 1, 2, \dots, q$ are not each other's Hermitian conjugate. In other applications, the matrix \mathbf{R}_{q+1} can be Hermitian and positive definite. For this case, if the sub-blocks are of order $p = 1$, the Levinson recursions can be used to efficiently invert the matrix \mathbf{R}_{q+1} . If the sub-blocks are of order $p > 1$, the extension to the Levinson recursions proposed by Wiggins and Robinson can be used [40]. In general, the matrix \mathbf{R}_{q+1} is not block-Hermitian and in such cases when the order of the block is 1, efficient inversion can be achieved using the Trench algorithm which requires the matrix to be only Toeplitz and not necessarily Hermitian. A refined version of the Trench algorithm proposed by Zohar can also be used for this purpose. For blocks of order $p > 1$, an extension of Zohar's algorithm was proposed by Akaike to invert a general block-Toeplitz matrix [18] and is referred to as the Akaike algorithm. Before going into the details of the Akaike inversion algorithm, we present some properties of a block-Toeplitz matrix.

3.2.1 Properties of block-Toeplitz Matrix

To facilitate the derivation of the Akaike inversion algorithm, some properties of a block-Toeplitz matrix are exploited. In order to perform operations such as flipping rows or columns by block, we define a $(q + 1) \times (q + 1)$ block matrix \mathbf{F}_{q+1} of block dimension $p \times p$

as shown below.

$$\mathbf{F}_{q+1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_p \\ \mathbf{0} & \cdots & \mathbf{I}_p & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{I}_p & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.4)$$

Here, \mathbf{I}_p is the $p \times p$ identity matrix and $\mathbf{0}$ is the $p \times p$ zero matrix. Premultiplying a block matrix by \mathbf{F}_{q+1} causes reversal of block rows and postmultiplying causes reversal of block columns. Switching the (m, g) block in a block matrix with the (g, m) block and vice versa is referred to as block transposition. We use the tilde symbol ($\tilde{\cdot}$) to denote a block transposition operation. Using the above definition, we obtain the following fundamental relationship,

$$\mathbf{F}_{q+1} \tilde{\mathbf{R}}_{q+1} \mathbf{F}_{q+1} = \mathbf{R}_{q+1} \quad (3.5)$$

Denoting $[\cdot]^{-1}$ as the inverse of a matrix, we have $\mathbf{F}_{q+1}^{-1} = \mathbf{F}_{q+1}$. Therefore, taking the inverse of equation (3.5) we obtain the following relation,

$$\mathbf{R}_{q+1}^{-1} = \mathbf{F}_{q+1} \tilde{\mathbf{R}}_{q+1}^{-1} \mathbf{F}_{q+1} \quad (3.6)$$

The block matrix \mathbf{F}_{q+1} has the property that $\mathbf{F}_{q+1} \mathbf{F}_{q+1} = \mathbf{I}_{q+1}$. Using this property, we premultiply and postmultiply equation (3.6) by \mathbf{F}_{q+1} to obtain the following relation,

$$\tilde{\mathbf{R}}_{q+1}^{-1} = \mathbf{F}_{q+1} \mathbf{R}_{q+1}^{-1} \mathbf{F}_{q+1} \quad (3.7)$$

In the Akaike algorithm, $\tilde{\mathbf{R}}_{q+1}^{-1}$ and \mathbf{R}_{q+1}^{-1} are considered simultaneously and the properties given above play a significant role in arriving at a recursive procedure for the block-Toeplitz matrix inversion.

3.2.2 Recursive Inversion Algorithm

In this section, we present the Akaike algorithm to recursively invert a block-Toeplitz matrix. To develop a recursive procedure, we represent the block-Toeplitz matrix \mathbf{R}_{q+1} as shown below,

$$\mathbf{R}_{q+1} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{-1} & \mathbf{C}_{-2} & \cdots & \mathbf{C}_{-q} \\ \mathbf{C}_1 & \mathbf{C}_0 & \mathbf{C}_{-1} & \cdots & \mathbf{C}_{-q+1} \\ \mathbf{C}_2 & \mathbf{C}_1 & \mathbf{C}_0 & \cdots & \mathbf{C}_{-q+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_q & \mathbf{C}_{q-1} & \mathbf{C}_{q-2} & \cdots & \mathbf{C}_0 \end{bmatrix}$$

$$\mathbf{R}_{q+1} = \begin{bmatrix} \mathbf{C}_0 & \tilde{\mathbf{S}}_q \\ \mathbf{P}_q & \mathbf{R}_q \end{bmatrix} \quad (3.8)$$

Here, the matrix \mathbf{R}_q is a block-Toeplitz matrix of order q and is given below,

$$\mathbf{R}_q = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{-1} & \cdots & \mathbf{C}_{-q+1} \\ \mathbf{C}_1 & \mathbf{C}_0 & \cdots & \mathbf{C}_{-q+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{q-1} & \mathbf{C}_{q-2} & \cdots & \mathbf{C}_0 \end{bmatrix} \quad (3.9)$$

The matrices \mathbf{S}_q and \mathbf{P}_q are as follows,

$$\mathbf{S}_q = \begin{bmatrix} \mathbf{C}_{-1} \\ \mathbf{C}_{-2} \\ \vdots \\ \mathbf{C}_{-q} \end{bmatrix} \quad (3.10)$$

$$\mathbf{P}_q = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_q \end{bmatrix} \quad (3.11)$$

We can alternatively represent \mathbf{R}_{q+1} as given below,

$$\mathbf{R}_{q+1} = \begin{bmatrix} \mathbf{R}_q & \check{\mathbf{S}}_q \\ \check{\mathbf{P}}_q & \mathbf{C}_0 \end{bmatrix} \quad (3.12)$$

Denoting reversal of block rows using the breve symbol ($\check{\cdot}$), in the above representation,

$$\check{\mathbf{S}}_q \triangleq \mathbf{F}_q \mathbf{S}_q = \begin{bmatrix} \mathbf{C}_{-q} \\ \mathbf{C}_{-q+1} \\ \mathbf{C}_{-q+2} \\ \vdots \\ \mathbf{C}_{-1} \end{bmatrix} \quad (3.13)$$

Also,

$$\check{\mathbf{P}}_q = \begin{bmatrix} \mathbf{C}_q & \mathbf{C}_{q-1} & \mathbf{C}_{q-2} & \cdots & \mathbf{C}_1 \end{bmatrix} \quad (3.14)$$

Applying block transposition to equation (3.3), we have,

$$\tilde{\mathbf{R}}_{q+1} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_q \\ \mathbf{C}_{-1} & \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{q-1} \\ \mathbf{C}_{-2} & \mathbf{C}_{-1} & \mathbf{C}_0 & \cdots & \mathbf{C}_{q-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{-q} & \mathbf{C}_{-q+1} & \mathbf{C}_{-q+2} & \cdots & \mathbf{C}_0 \end{bmatrix} \quad (3.15)$$

Therefore, using equations (3.10) and (3.11) we can represent $\tilde{\mathbf{R}}_{q+1}$ in the following form,

$$\tilde{\mathbf{R}}_{q+1} = \begin{bmatrix} \mathbf{C}_0 & \check{\mathbf{P}}_q \\ \mathbf{S}_q & \check{\mathbf{R}}_q \end{bmatrix} \quad (3.16)$$

Also, using equations (3.13) and (3.14), we have,

$$\tilde{\mathbf{R}}_{q+1} = \begin{bmatrix} \check{\mathbf{R}}_q & \check{\mathbf{P}}_q \\ \check{\mathbf{S}}_q & \mathbf{C}_0 \end{bmatrix} \quad (3.17)$$

Now let us represent the inverse of \mathbf{R}_{q+1} as,

$$\mathbf{R}_{q+1}^{-1} = \begin{bmatrix} \mathbf{Q}_q & \tilde{\mathbf{G}}_q \\ \mathbf{L}_q & \mathbf{M}_q \end{bmatrix} \quad (3.18)$$

and the inverse of $\tilde{\mathbf{R}}_{q+1}$ as,

$$\tilde{\mathbf{R}}_{q+1}^{-1} = \begin{bmatrix} \mathbf{K}_q & \tilde{\mathbf{U}}_q \\ \mathbf{O}_q & \mathbf{N}_q \end{bmatrix} \quad (3.19)$$

Here, \mathbf{Q}_q and \mathbf{K}_q are $p \times p$ matrices and \mathbf{M}_q and \mathbf{N}_q are $q \times q$ block matrices. The matrices \mathbf{U}_q , \mathbf{O}_q , \mathbf{G}_q , and \mathbf{L}_q are $q \times 1$ block matrices. The dimension of individual blocks in all the block matrices is $p \times p$. Using equation (3.6), we have the following alternative representation for \mathbf{R}_{q+1}^{-1} ,

$$\mathbf{R}_{q+1}^{-1} = \begin{bmatrix} \overset{\circ}{\mathbf{N}}_q & \overset{\circ}{\mathbf{O}}_q \\ \overset{\sim}{\mathbf{U}}_q & \mathbf{K}_q \end{bmatrix} \quad (3.20)$$

Here, $\overset{\circ}{\mathbf{N}}_q = \mathbf{F}_q \mathbf{N}_q \mathbf{F}_q$. In addition, using equation (3.7), we have the following alternative representation for $\tilde{\mathbf{R}}_{q+1}^{-1}$,

$$\tilde{\mathbf{R}}_{q+1}^{-1} = \begin{bmatrix} \overset{\circ}{\mathbf{M}}_q & \overset{\sim}{\mathbf{L}}_q \\ \overset{\sim}{\mathbf{G}}_q & \mathbf{Q}_q \end{bmatrix} \quad (3.21)$$

Here, $\overset{\circ}{\mathbf{M}}_q = \mathbf{F}_q \mathbf{M}_q \mathbf{F}_q$. Now we use the relation $\tilde{\mathbf{R}}_{q+1}^{-1} \tilde{\mathbf{R}}_{q+1} = \mathbf{I}_{q+1}$ and equations (3.19) and (3.16) to obtain the following,

$$\begin{bmatrix} \mathbf{K}_q & \tilde{\mathbf{U}}_q \\ \mathbf{O}_q & \mathbf{N}_q \end{bmatrix} \begin{bmatrix} \mathbf{C}_0 & \tilde{\mathbf{P}}_q \\ \mathbf{S}_q & \tilde{\mathbf{R}}_q \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{0}_{p \times pq} \\ \mathbf{0}_{pq \times p} & \mathbf{I}_{pq \times pq} \end{bmatrix} \quad (3.22)$$

From the North-East equation in the above equation, we derive the following relationship,

$$\mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q = -\tilde{\mathbf{P}}_q \tilde{\mathbf{R}}_q^{-1} \quad (3.23)$$

Similarly, from equations (3.21) and (3.17) we obtain the following,

$$\begin{bmatrix} \overset{\circ}{\mathbf{M}}_q & \overset{\sim}{\mathbf{L}}_q \\ \overset{\sim}{\mathbf{G}}_q & \mathbf{Q}_q \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{R}}_q & \overset{\sim}{\mathbf{P}}_q \\ \tilde{\mathbf{S}}_q & \mathbf{C}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{pq \times pq} & \mathbf{0}_{pq \times p} \\ \mathbf{0}_{p \times pq} & \mathbf{I}_{p \times p} \end{bmatrix} \quad (3.24)$$

From the South-West equation in the above matrix equation, we derive the following relationship,

$$\mathbf{Q}_q^{-1} \tilde{\mathbf{G}}_q = -\tilde{\mathbf{S}}_q \tilde{\mathbf{R}}_q^{-1} \quad (3.25)$$

In a similar fashion, we consider $\tilde{\mathbf{R}}_{q+1} \tilde{\mathbf{R}}_{q+1}^{-1} = \mathbf{I}_{q+1}$ and derive the following two relationships using equations (3.19, 3.16) and (3.21, 3.17) respectively.

$$\mathbf{O}_q \mathbf{K}_q^{-1} = -\tilde{\mathbf{R}}_q^{-1} \mathbf{S}_q \quad (3.26)$$

$$\check{\mathbf{L}}_q \mathbf{Q}_q^{-1} = -\check{\mathbf{R}}_q^{-1} \check{\mathbf{P}}_q \quad (3.27)$$

Also, from equations (3.22), (3.23), and (3.26), we derive the following connection.

$$\begin{aligned} \mathbf{K}_q^{-1} &= \mathbf{C}_0 + \left(\mathbf{K}_q^{-1} \check{\mathbf{U}}_q \right) \mathbf{S}_q \\ &= \mathbf{C}_0 - \check{\mathbf{P}}_q \left(\check{\mathbf{R}}_q^{-1} \mathbf{S}_q \right) \\ &= \mathbf{C}_0 + \check{\mathbf{P}}_q \left(\mathbf{O}_q \mathbf{K}_q^{-1} \right) \end{aligned} \quad (3.28)$$

Similarly, using equations (3.24), (3.25), and (3.27), we obtain,

$$\begin{aligned} \mathbf{Q}_q^{-1} &= \mathbf{C}_0 + \left(\mathbf{Q}_q^{-1} \check{\mathbf{G}}_q \right) \check{\mathbf{P}}_q \\ &= \mathbf{C}_0 - \check{\mathbf{S}}_q \left(\check{\mathbf{R}}_q^{-1} \check{\mathbf{P}}_q \right) \\ &= \mathbf{C}_0 + \check{\mathbf{S}}_q \left(\check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \right) \end{aligned} \quad (3.29)$$

An underlying assumption in the above derivations is the existence of the necessary inverses. Now we look at \mathbf{R}_{q+1}^{-1} and $\check{\mathbf{R}}_{q+1}^{-1}$ in equations (3.18) and (3.19) respectively. It can be shown that if the first block rows and columns of these inverses, \mathbf{Q}_q , $\check{\mathbf{G}}_q$, \mathbf{L}_q , \mathbf{K}_q , $\check{\mathbf{U}}_q$, and \mathbf{O}_q , are computed recursively, they can be used to obtain the rest (\mathbf{M}_q and \mathbf{N}_q) of the inverse matrices. We note that if the left-hand side matrix products in equations (3.23), (3.25), (3.26), and (3.27) can be expressed recursively, then using equations (3.28) and (3.29) we can recursively obtain the first block rows and columns of \mathbf{R}_{q+1}^{-1} and $\check{\mathbf{R}}_{q+1}^{-1}$. Next, we derive recursive relations for the aforementioned matrix products. Using equations (3.24) and (3.25), we derive the following expression for $\mathring{\mathbf{M}}_q$,

$$\begin{aligned} \mathring{\mathbf{M}}_q &= \check{\mathbf{R}}_q^{-1} - \check{\mathbf{L}}_q \left(\check{\mathbf{S}}_q \check{\mathbf{R}}_q^{-1} \right) \\ &= \check{\mathbf{R}}_q^{-1} + \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \check{\mathbf{G}}_q \end{aligned} \quad (3.30)$$

Similarly, using equations (3.22) and (3.23), we have,

$$\begin{aligned} \mathbf{N}_q &= \check{\mathbf{R}}_q^{-1} - \mathbf{O}_q \left(\check{\mathbf{P}}_q \check{\mathbf{R}}_q^{-1} \right) \\ &= \check{\mathbf{R}}_q^{-1} + \mathbf{O}_q \mathbf{K}_q^{-1} \check{\mathbf{U}}_q \end{aligned} \quad (3.31)$$

We derive the first recursive relation as shown below. The equations used to arrive at intermediate results are shown in parentheses while performing derivations.

$$\begin{aligned}
\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} &= -\tilde{\mathbf{P}}_{q+1} \tilde{\mathbf{R}}_{q+1}^{-1} \quad (3.23) \\
&= -\tilde{\mathbf{P}}_{q+1} \begin{bmatrix} \mathring{\mathbf{M}}_q & \check{\mathbf{L}}_q \\ \check{\mathbf{G}}_q & \mathbf{Q}_q \end{bmatrix} \quad (3.21) \\
&= -\begin{bmatrix} \tilde{\mathbf{P}}_q & \mathbf{C}_{q+1} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{R}}_q^{-1} + \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \check{\mathbf{G}}_q & \check{\mathbf{L}}_q \\ \check{\mathbf{G}}_q & \mathbf{Q}_q \end{bmatrix} \quad (3.11, 3.30) \\
&= -\begin{bmatrix} \tilde{\mathbf{P}}_q & \mathbf{C}_{q+1} \end{bmatrix} \left(\begin{bmatrix} \tilde{\mathbf{R}}_q^{-1} & \mathbf{0}_{pq \times p} \\ \mathbf{0}_{p \times pq} & \mathbf{0}_{p \times p} \end{bmatrix} + \begin{bmatrix} \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \\ \mathbf{I}_{p \times p} \end{bmatrix} \begin{bmatrix} \check{\mathbf{G}}_q & \mathbf{Q}_q \end{bmatrix} \right) \\
&= \begin{bmatrix} \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q & \mathbf{0}_{p \times p} \end{bmatrix} - \left(\tilde{\mathbf{P}}_q \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} + \mathbf{C}_{q+1} \right) \mathbf{Q}_q \begin{bmatrix} \mathbf{Q}_q^{-1} \check{\mathbf{G}}_q & \mathbf{I}_{p \times p} \end{bmatrix} \quad (3.23)
\end{aligned}$$

Substituting for $\check{\mathbf{L}}_q \mathbf{Q}_q^{-1}$ from equation (3.27) and subsequently replacing $\tilde{\mathbf{P}}_q \tilde{\mathbf{R}}_q^{-1}$ with $-\mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q$ using equation (3.23), we have the following final expression for the above recursion,

$$\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} = \begin{bmatrix} \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q & \mathbf{0}_{p \times p} \end{bmatrix} - \left(\left(\mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \right) \check{\mathbf{P}}_q + \mathbf{C}_{q+1} \right) \mathbf{Q}_q \begin{bmatrix} \mathbf{Q}_q^{-1} \check{\mathbf{G}}_q & \mathbf{I}_{p \times p} \end{bmatrix} \quad (3.32)$$

Let us just use the notation $\left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1}$ for $-\left(\left(\mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \right) \check{\mathbf{P}}_q + \mathbf{C}_{q+1} \right) \mathbf{Q}_q$ in the above equation. We then have,

$$\begin{aligned}
\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} &= \begin{bmatrix} \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q & \mathbf{0}_{p \times p} \end{bmatrix} + \left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1} \begin{bmatrix} \mathbf{Q}_q^{-1} \check{\mathbf{G}}_q & \mathbf{I}_{p \times p} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q + \left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1} \left(\mathbf{Q}_q^{-1} \check{\mathbf{G}}_q \right) & \left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1} \end{bmatrix} \quad (3.33)
\end{aligned}$$

We now observe that $\left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1}$ is notation for the last block of $\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1}$.

Proceeding in a similar fashion, we use equations (3.25), (3.19), (3.10), and (3.31) to obtain the second recursive relation as shown below,

$$\mathbf{Q}_{q+1}^{-1} \check{\mathbf{G}}_{q+1} = \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{Q}_q^{-1} \check{\mathbf{G}}_q \end{bmatrix} - \left(\mathbf{C}_{-q-1} + \left(\mathbf{Q}_q^{-1} \check{\mathbf{G}}_q \right) \mathbf{S}_q \right) \mathbf{K}_q \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \end{bmatrix} \quad (3.34)$$

Furthermore, using $\left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)_1$ to notate $-\left(\mathbf{C}_{-q-1} + \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\mathbf{S}_q\right)\mathbf{K}_q$, we have,

$$\begin{aligned}\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1} &= \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q \end{bmatrix} + \left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)_1 \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q \end{bmatrix} \\ &= \left[\left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)_1 \quad \mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q + \left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)_1 \left(\mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q\right)\right] \end{aligned} \quad (3.35)$$

In the above equation, $\left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)_1$ turns out to be notation for the first block of $\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}$. Additionally, using equations (3.21) and (3.19) we derive recursive relations for $\mathbf{O}_{q+1}\mathbf{K}_{q+1}^{-1}$ and $\check{\mathbf{L}}_{q+1}\mathbf{Q}_{q+1}^{-1}$ respectively as given below,

$$\mathbf{O}_{q+1}\mathbf{K}_{q+1}^{-1} = \begin{bmatrix} \mathbf{O}_q\mathbf{K}_q^{-1} \\ \mathbf{0}_{p \times p} \end{bmatrix} - \begin{bmatrix} \check{\mathbf{L}}_q\mathbf{Q}_q^{-1} \\ \mathbf{I}_{p \times p} \end{bmatrix} \mathbf{Q}_q \left(\mathbf{C}_{-q-1} + \check{\mathbf{S}}_q \left(\mathbf{O}_q\mathbf{K}_q^{-1}\right)\right) \quad (3.36)$$

$$\check{\mathbf{L}}_{q+1}\mathbf{Q}_{q+1}^{-1} = \begin{bmatrix} \mathbf{0}_{p \times p} \\ \check{\mathbf{L}}_q\mathbf{Q}_q^{-1} \end{bmatrix} - \begin{bmatrix} \mathbf{I}_{p \times p} \\ \mathbf{O}_q\mathbf{K}_q^{-1} \end{bmatrix} \mathbf{K}_q \left(\check{\mathbf{P}}_q \left(\check{\mathbf{L}}_q\mathbf{Q}_q^{-1}\right) + \mathbf{C}_{q+1}\right) \quad (3.37)$$

Now we show how to obtain recursive equations for \mathbf{Q}_{q+1}^{-1} and \mathbf{K}_{q+1}^{-1} . From equation (3.29), we have,

$$\mathbf{Q}_{q+1}^{-1} = \mathbf{C}_0 + \left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)\check{\mathbf{P}}_{q+1} \quad (3.38)$$

Subtracting $\mathbf{Q}_q^{-1} = \mathbf{C}_0 + \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\check{\mathbf{P}}_q$ from the above equation and using equation (3.34) we derive the following relation,

$$\begin{aligned}\mathbf{Q}_{q+1}^{-1} &= \mathbf{Q}_q^{-1} + \left(\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}\right)\check{\mathbf{P}}_{q+1} - \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\check{\mathbf{P}}_q \\ &= \mathbf{Q}_q^{-1} + \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q \end{bmatrix} \check{\mathbf{P}}_{q+1} - \left(\mathbf{C}_{-q-1} + \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\mathbf{S}_q\right)\mathbf{K}_q \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q \end{bmatrix} \check{\mathbf{P}}_{q+1} \\ &\quad - \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\check{\mathbf{P}}_q \\ &= \mathbf{Q}_q^{-1} + \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q \end{bmatrix} \begin{bmatrix} \mathbf{C}_{q+1} \\ \check{\mathbf{P}}_q \end{bmatrix} - \left(\mathbf{C}_{-q-1} + \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\mathbf{S}_q\right)\mathbf{K}_q \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q \end{bmatrix} \begin{bmatrix} \mathbf{C}_{q+1} \\ \check{\mathbf{P}}_q \end{bmatrix} \\ &\quad - \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\check{\mathbf{P}}_q \quad (3.14) \\ &= \mathbf{Q}_q^{-1} - \left(\mathbf{C}_{-q-1} + \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\mathbf{S}_q\right)\mathbf{K}_q \left(\mathbf{C}_{q+1} + \mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q\check{\mathbf{P}}_q\right) \end{aligned}$$

The block matrices $\left(\mathbf{C}_{-q-1} + \left(\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q\right)\mathbf{S}_q\right)\mathbf{K}_q$ and $\left(\mathbf{C}_{q+1} + \mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q\check{\mathbf{P}}_q\right)$ in the above equation are already available from the first and last blocks of $\mathbf{Q}_{q+1}^{-1}\tilde{\mathbf{G}}_{q+1}$ and $\mathbf{K}_{q+1}^{-1}\tilde{\mathbf{U}}_{q+1}$

in equations (3.35) and (3.33) respectively. Therefore, we can express the above recursive relation as shown below,

$$\mathbf{Q}_{q+1}^{-1} = \mathbf{Q}_q^{-1} - \left(\mathbf{Q}_{q+1}^{-1} \tilde{\mathbf{G}}_{q+1} \right)_1 \left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1} \mathbf{Q}_q^{-1} \quad (3.39)$$

Analogously, we can use equation (3.28) to derive a recursive relation for \mathbf{K}_{q+1}^{-1} and represent it using the last and first blocks of already computed matrix products in equations (3.35) and (3.33) respectively as given below.

$$\mathbf{K}_{q+1}^{-1} = \mathbf{K}_q^{-1} - \left(\mathbf{K}_{q+1}^{-1} \tilde{\mathbf{U}}_{q+1} \right)_{q+1} \left(\mathbf{Q}_{q+1}^{-1} \tilde{\mathbf{G}}_{q+1} \right)_1 \mathbf{K}_q^{-1} \quad (3.40)$$

The equations (3.32–3.37), (3.39), and (3.40) are the required recursive relations which we can use to compute the first block rows and columns of the matrices \mathbf{R}_{q+1}^{-1} and $\tilde{\mathbf{R}}_{q+1}^{-1}$. The recursions are started at $q = 0$ with \mathbf{Q}_0^{-1} and \mathbf{K}_0^{-1} set to \mathbf{C}_0 .

The next step is to use the first block rows and columns to compute the rest of the inverse matrix. Replacing $\mathring{\mathbf{M}}_q$ in equation (3.21) with the expression in equation (3.30) we have the following,

$$\tilde{\mathbf{R}}_{q+1}^{-1} = \begin{bmatrix} \tilde{\mathbf{R}}_q^{-1} + \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \tilde{\mathbf{G}}_q & \check{\mathbf{L}}_q \\ \tilde{\mathbf{G}}_q & \mathbf{Q}_q \end{bmatrix}$$

Using m and g to index the blocks of $\tilde{\mathbf{R}}_{q+1}^{-1}$, we obtain the following block-wise connection from the above equation,

$$\left(\tilde{\mathbf{R}}_{q+1}^{-1} \right)_{m,g} = \left(\tilde{\mathbf{R}}_q^{-1} \right)_{m,g} + \left(\check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \tilde{\mathbf{G}}_q \right)_{m,g} \quad 1 \leq (m, g) \leq q \quad (3.41)$$

Similarly, replacing \mathbf{N}_q in equation (3.19) with the expression in equation (3.31), we obtain the following block-wise connection,

$$\left(\tilde{\mathbf{R}}_{q+1}^{-1} \right)_{m+1,g+1} = \left(\tilde{\mathbf{R}}_q^{-1} \right)_{m,g} + \left(\mathbf{O}_q \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \right)_{m,g} \quad 1 \leq (m, g) \leq q \quad (3.42)$$

Subtracting equation (3.41) from equation (3.42), we arrive at the following recursive relation for computing the remaining blocks of the matrix $\tilde{\mathbf{R}}_{q+1}^{-1}$,

$$\left(\tilde{\mathbf{R}}_{q+1}^{-1} \right)_{m+1,g+1} = \left(\tilde{\mathbf{R}}_{q+1}^{-1} \right)_{m,g} + \left(\mathbf{O}_q \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q - \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \tilde{\mathbf{G}}_q \right)_{m,g} \quad (3.43)$$

The desired block-Toeplitz matrix inverse \mathbf{R}_{q+1}^{-1} can now be obtained using equation (3.6).

Using i as the iteration index, the steps for implementing the Akaike algorithm for inverting a block-Toeplitz matrix recursively are outlined below,

1. Initialization ($i = 0$): Set $\mathbf{Q}_i^{-1} = \mathbf{K}_i^{-1} = \mathbf{C}_0$. Now compute the following,

- $\mathbf{K}_{i+1}^{-1} \tilde{\mathbf{U}}_{i+1} = -\mathbf{C}_1 \mathbf{C}_0^{-1}$
- $\mathbf{Q}_{i+1}^{-1} \tilde{\mathbf{G}}_{i+1} = -\mathbf{C}_{-1} \mathbf{C}_0^{-1}$
- $\mathbf{O}_{i+1} \mathbf{K}_{i+1}^{-1} = -\mathbf{C}_0^{-1} \mathbf{C}_{-1}$
- $\check{\mathbf{L}}_q \mathbf{Q}_q^{-1} = -\mathbf{C}_0^{-1} \mathbf{C}_1$
- $\mathbf{Q}_{i+1}^{-1} = \mathbf{Q}_i^{-1} - \left(\mathbf{Q}_{i+1}^{-1} \tilde{\mathbf{G}}_{i+1} \right) \left(\mathbf{K}_{i+1}^{-1} \tilde{\mathbf{U}}_{i+1} \right) \mathbf{Q}_i^{-1}$ and $\mathbf{Q}_{i+1} = \left(\mathbf{Q}_{i+1}^{-1} \right)^{-1}$
- $\mathbf{K}_{i+1}^{-1} = \mathbf{K}_i^{-1} - \left(\mathbf{K}_{i+1}^{-1} \tilde{\mathbf{U}}_{i+1} \right) \left(\mathbf{Q}_{i+1}^{-1} \tilde{\mathbf{G}}_{i+1} \right) \mathbf{K}_i^{-1}$ and $\mathbf{K}_{i+1} = \left(\mathbf{K}_{i+1}^{-1} \right)^{-1}$
- $\mathbf{P}_{i+1} = \mathbf{C}_1$
- $\mathbf{S}_{i+1} = \mathbf{C}_{-1}$

2. Recursion:

For $i = 1$ to $q - 1$

- $\check{\mathbf{P}}_i = \mathbf{F}_i \mathbf{P}_i$
 - Compute $\mathbf{K}_{i+1}^{-1} \tilde{\mathbf{U}}_{i+1}$ using equation (3.32) and $\mathbf{Q}_{i+1}^{-1} \tilde{\mathbf{G}}_{i+1}$ using equation (3.34).
 - $\check{\mathbf{S}}_i = \mathbf{F}_i \mathbf{S}_i$
 - Form $\mathbf{S}_{i+1} = \begin{bmatrix} \mathbf{S}_i \\ \mathbf{C}_{-i-1} \end{bmatrix}$ for next iteration.
 - Compute $\mathbf{O}_{i+1} \mathbf{K}_{i+1}^{-1}$ using equation (3.36).
 - Block transpose \mathbf{P}_i to obtain $\check{\mathbf{P}}_i$ and form $\mathbf{P}_{i+1} = \begin{bmatrix} \mathbf{P}_i \\ \mathbf{C}_{i+1} \end{bmatrix}$ for next iteration.
 - Compute $\check{\mathbf{L}}_{i+1} \mathbf{Q}_{i+1}^{-1}$ using equation (3.37).
 - Compute \mathbf{Q}_{i+1}^{-1} and \mathbf{K}_{i+1}^{-1} using equations (3.39) and (3.40) respectively and invert them to obtain \mathbf{Q}_{i+1} and \mathbf{K}_{i+1} for the next iteration.
-

end

3. After computing the recursions, we have obtained for $i = q - 1$ the following: $\mathbf{K}_q^{-1}\tilde{\mathbf{U}}_q$, $\mathbf{Q}_q^{-1}\tilde{\mathbf{G}}_q$, $\check{\mathbf{L}}_q\mathbf{Q}_q^{-1}$, and $\mathbf{O}_q\mathbf{K}_q^{-1}$. We also obtain \mathbf{K}_q and \mathbf{Q}_q . Using these we form the inverse block-Toeplitz matrix \mathbf{R}_{q+1}^{-1} as given below.

- Compute $\tilde{\mathbf{U}}_q = \mathbf{K}_q \left(\mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \right)$. Similarly, compute $\tilde{\mathbf{G}}_q$ and \mathbf{O}_q .
- Define $\mathbf{X} = \mathbf{O}_q \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q - \check{\mathbf{L}}_q \mathbf{Q}_q^{-1} \tilde{\mathbf{G}}_q$.
- Initialize $\tilde{\mathbf{R}}_{q+1}^{-1} = \begin{bmatrix} \mathbf{K}_q & \tilde{\mathbf{U}}_q \\ \mathbf{O}_q & \mathbf{0}_{pq \times pq} \end{bmatrix}$ consistent with the definition in (3.19).
- Using m and g to index the blocks, we have from equation (3.43),

For $m = 1$ to q

For $g = 1$ to q

$$\left(\tilde{\mathbf{R}}_{q+1}^{-1} \right)_{m+1, g+1} = \left(\tilde{\mathbf{R}}_{q+1}^{-1} \right)_{m, g} + (\mathbf{X})_{m, g}$$

end

end

- Finally, compute $\mathbf{R}_{q+1}^{-1} = \mathbf{F}_{q+1} \tilde{\mathbf{R}}_{q+1}^{-1} \mathbf{F}_{q+1}$. (3.6)

3.2.3 Validity of Recursions and Computational Complexity

In this section, we consider the necessary assumptions for the validity of the Akaike algorithm and look at its computational complexity. From equations (3.19) and (3.31) we have,

$$\tilde{\mathbf{R}}_{q+1}^{-1} = \begin{bmatrix} \mathbf{K}_q & \tilde{\mathbf{U}}_q \\ \mathbf{O}_q & \tilde{\mathbf{R}}_q^{-1} + \mathbf{O}_q \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \end{bmatrix} \quad (3.44)$$

For general square matrices \mathbf{B}_1 and \mathbf{B}_4 , we have the following formula for the determinant of a block matrix [41],

$$\left| \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{bmatrix} \right| = |\mathbf{B}_1| |\mathbf{B}_4 - \mathbf{B}_3 \mathbf{B}_1^{-1} \mathbf{B}_2| \quad (3.45)$$

Here, $|\cdot|$ denotes the determinant of a matrix. Using the above formula, we have,

$$\begin{aligned} \left| \tilde{\mathbf{R}}_{q+1}^{-1} \right| &= |\mathbf{K}_q| \left| \tilde{\mathbf{R}}_q^{-1} + \mathbf{O}_q \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q - \mathbf{O}_q \mathbf{K}_q^{-1} \tilde{\mathbf{U}}_q \right| \\ &= |\mathbf{K}_q| \left| \tilde{\mathbf{R}}_q^{-1} \right| \end{aligned} \quad (3.46)$$

Similarly, using equations (3.21) and (3.30) we obtain,

$$\left| \tilde{\mathbf{R}}_{q+1}^{-1} \right| = |\mathbf{Q}_q| \left| \tilde{\mathbf{R}}_q^{-1} \right| \quad (3.47)$$

From the above equation and (3.46) we get,

$$|\mathbf{Q}_q| = |\mathbf{K}_q| \quad (3.48)$$

Therefore, nonsingularity of \mathbf{K}_q^{-1} or \mathbf{Q}_q^{-1} is the only assumption necessary for the validity of Akaike's recursive algorithm.

The Gaussian elimination approach to solve a system of n equations for n unknowns requires $n(n+1)/2$ divisions and $(2n^3 + 3n^2 - 5n)/6$ multiplications and additions [42]. Modern computers and Digital Signal Processors (DSPs) contain a dedicated multiply-accumulate (MAC) unit and can perform fast multiply adds. So we consider only the multiplications and divisions which adds to $(n^3 + 3n^2 - n)/3$ operations. Therefore, the computational complexity of Gaussian elimination is $O(n^3)$. We are considering a linear system of $p(q+1)$ equations and unknowns. Unlike the $O(p^3(q+1)^3)$ computations required for the Gaussian elimination approach, the Akaike algorithm involves $O(p^3(q+1)^2)$ operations [20]. While the computational complexity of the latter is lower, the computational effort required may or may not be less depending on the linear combination terms. We note that the recursion itself involves inversion of the $p \times p$ matrices \mathbf{Q}_q^{-1} and \mathbf{K}_q^{-1} . These matrices do not have a structure that we could exploit to perform further recursive and efficient inversion. In IF estimation based on TVAR modeling, p is the number of time-varying coefficients in the TVAR model. When $p = 2$, we can perform the inversion efficiently using the following well known relationship.

$$\text{If nonsingular } \mathbf{B} = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} \quad \text{then } \mathbf{B}^{-1} = \frac{1}{b_1 b_4 - b_2 b_3} \begin{bmatrix} b_4 & -b_2 \\ -b_3 & b_1 \end{bmatrix} \quad (3.49)$$

For $p = 3$ the following analytical expression can be used for inversion,

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} \Rightarrow \mathbf{B}^{-1} = \frac{1}{|\mathbf{B}|} \begin{bmatrix} b_5b_9 - b_6b_8 & -(b_9b_2 - b_8b_3) & b_2b_6 - b_5b_3 \\ -(b_9b_4 - b_7b_6) & b_9b_1 - b_7b_3 & -(b_6b_1 - b_4b_3) \\ b_8b_4 - b_7b_5 & -(b_8b_1 - b_7b_2) & b_5b_1 - b_4b_2 \end{bmatrix},$$

$$|\mathbf{B}| = b_1(b_5b_9 - b_6b_8) - b_4(b_9b_2 - b_8b_3) + b_7(b_2b_6 - b_5b_3) \quad (3.50)$$

When $p > 3$, we resort to Gaussian elimination. Still, the recursive approach avoids direct inversion of the relatively large $p(q+1) \times p(q+1)$ covariance matrix and provides significant computational and memory efficiency.

3.2.4 Experimental Verification

In this section, we show an example simulation to demonstrate the workings of a MATLAB implementation of the Akaike algorithm. We compare the solution vector $\hat{\mathbf{a}} = \mathbf{F}_{q+1}\check{\mathbf{a}}$ obtained by solving the linear system of equations $\mathbf{R}_{q+1}\check{\mathbf{a}} = -\mathbf{d}$ using the Akaike algorithm for inverting \mathbf{R}_{q+1} and the solution \mathbf{a} of the linear system of equations $\mathbf{C}\mathbf{a} = -\mathbf{d}$ obtained using Gaussian elimination. We use the matrix \mathbf{C} obtained in the linear FM example in Section 2.3.2 to form \mathbf{R}_{q+1} . In Figure 3.1 we show the real and imaginary parts of the Gaussian elimination solution vector \mathbf{a} and the Akaike algorithm solution vector $\hat{\mathbf{a}}$. We see that the Akaike algorithm results in a solution approximately equal to the Gaussian elimination solution. The magnitude of solution errors $\mathbf{e} = \mathbf{C}\mathbf{a} + \mathbf{d}$ and $\hat{\mathbf{e}} = \mathbf{C}\hat{\mathbf{a}} + \mathbf{d}$ are shown in Figure 3.2. From Figure 3.2, we see that magnitude of error is negligible in either approach and it is smaller for Gaussian elimination compared to the Akaike algorithm. The Akaike algorithm works well for a general block-Toeplitz matrix. However, it is not designed to take into account the properties of individual blocks. This is disadvantageous when the individual blocks have a Toeplitz structure so that the number of computations can be reduced further. In the next section, we present the Wax-Kailath algorithm that exploits the Toeplitz structure of individual blocks.

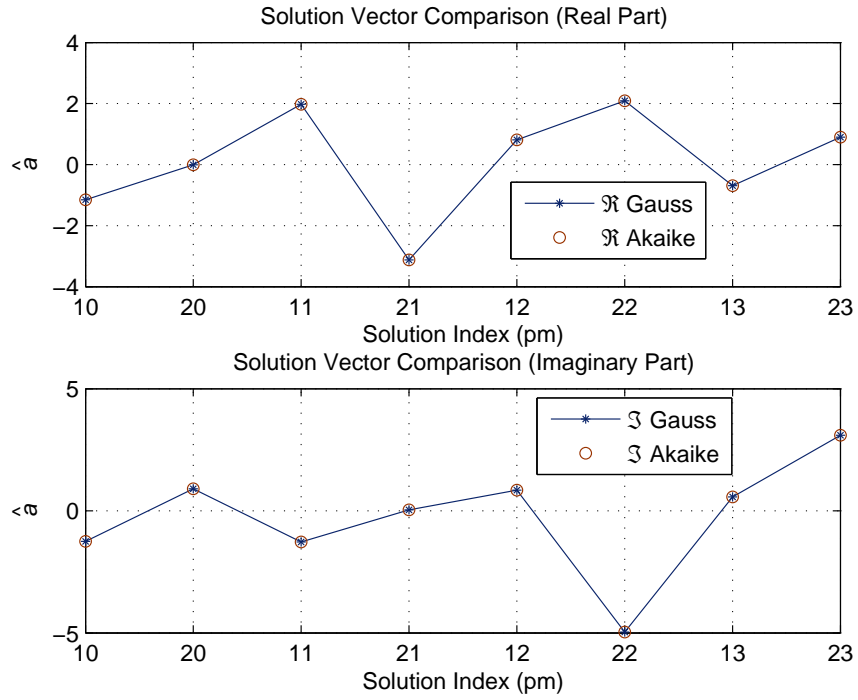


Figure 3.1: Solution Vector Comparison for Akaike Algorithm.

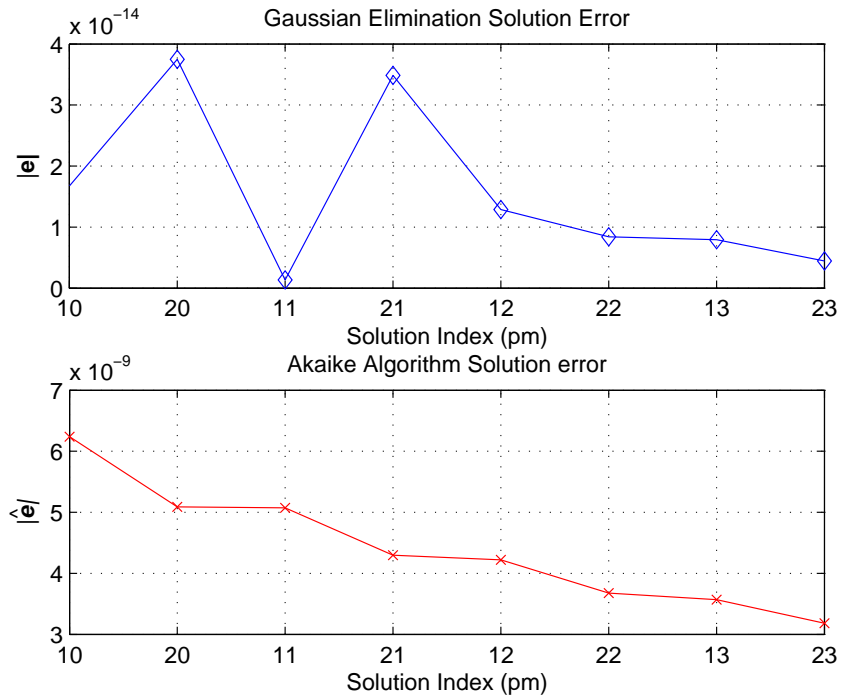


Figure 3.2: Magnitude of Solution Error for Gaussian Elimination and Akaike Algorithm.

3.3 Wax-Kailath Algorithm

In this section, we introduce the Wax-Kailath algorithm for inverting a block-Toeplitz matrix $\mathbf{T}_{p,q+1}$ whose individual blocks are Toeplitz. The feasibility of using this algorithm for solving the linear system of equations $\mathbf{R}_{q+1}\check{\mathbf{a}} = -\mathbf{d}$ stems from our observation that when the covariance method with power series basis functions is used for TVAR modeling, the block-Toeplitz matrix \mathbf{R}_{q+1} that we obtain has individual blocks that are Hermitian as well as being close to Toeplitz. Therefore, we add an appropriate error matrix \mathbf{E} to \mathbf{R}_{q+1} to obtain a block-Toeplitz matrix $\mathbf{T}_{p,q+1}$ with Hermitian-Toeplitz blocks as shown below.

$$\mathbf{R}_{q+1} = \mathbf{T}_{p,q+1} - \mathbf{E} \quad (3.51)$$

Using m and g to index the blocks, we have the following formula for the blocks of \mathbf{E} ,

$$\mathbf{E}_{mg} = \mathbf{T}_{mg} - \mathbf{C}_{mg} \quad 0 \leq (m, g) \leq q \quad (3.52)$$

Here \mathbf{C}_{mg} denote the blocks in \mathbf{R}_{q+1} and \mathbf{T}_{mg} is the corresponding Toeplitz block. We form the Hermitian-Toeplitz blocks as shown below,

$$\mathbf{T}_{mg} = \text{toeplitz}(\text{mean}(\text{diag}(\mathbf{C}_{mg}, i))) \quad i = 0, 1, \dots, p-1 \quad (3.53)$$

Here, the function *diag* extracts the i^{th} diagonal of the matrix \mathbf{C}_{mg} . The function *mean* computes the mean of the i^{th} diagonal. Finally, the function *toeplitz* forms a Hermitian-Toeplitz matrix whose first row elements equal the mean of the diagonals. Using the above equation, we obtain $\mathbf{T}_{p,q+1}$ and compute \mathbf{E}_{mg} to form the error matrix \mathbf{E} .

3.3.1 Properties of block-Toeplitz matrix with Toeplitz blocks

We consider a $(q + 1) \times (q + 1)$ block-Toeplitz matrix $\mathbf{T}_{p,q+1}$ with Hermitian-Toeplitz blocks of dimension $p \times p$ as shown below,

$$\mathbf{T}_{p,q+1} = \begin{bmatrix} \mathbf{T}_0 & \mathbf{T}_1 & \mathbf{T}_2 & \cdots & \mathbf{T}_q \\ \mathbf{T}_{-1} & \mathbf{T}_0 & \mathbf{T}_1 & \cdots & \mathbf{T}_{q-1} \\ \mathbf{T}_{-2} & \mathbf{T}_{-1} & \mathbf{T}_0 & \cdots & \mathbf{T}_{q-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_{-q} & \mathbf{T}_{-q+1} & \mathbf{T}_{-q+2} & \cdots & \mathbf{T}_0 \end{bmatrix} \quad (3.54)$$

Here, \mathbf{T}_i and \mathbf{T}_{-i} for $i = 0, 1, \dots, q$ are $p \times p$ matrices having Hermitian-Toeplitz structure as shown below,

$$\mathbf{T}_i = \begin{bmatrix} t_{0,i} & t_{1,i}^* & t_{2,i}^* & \cdots & t_{p-1,i}^* \\ t_{1,i} & t_{0,i} & t_{1,i}^* & \cdots & t_{p-2,i}^* \\ t_{2,i} & t_{1,i} & t_{0,i} & \cdots & t_{p-3,i}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{p-1,i} & t_{p-2,i} & t_{p-3,i} & \cdots & t_{0,i} \end{bmatrix} \quad (3.55)$$

Now we define a $p(q + 1) \times p(q + 1)$ exchange matrix $\mathbf{J}_{p,q+1}$ as shown below,

$$\mathbf{J}_{p,q+1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_p \\ \mathbf{0} & \cdots & \mathbf{J}_p & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{J}_p & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.56)$$

Here, \mathbf{J}_p is a $p \times p$ matrix obtained by column reversal of the identity matrix,

$$\mathbf{J}_p = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \quad (3.57)$$

We have the following property for the exchange matrix $\mathbf{J}_{p,q+1}$ [43],

$$\mathbf{J}_{p,q+1} = \mathbf{J}_{p,q+1}^H = \mathbf{J}_{p,q+1}^{-1} \quad (3.58)$$

Here, $[\cdot]^H$ denotes complex conjugate (or Hermitian) transpose. Toeplitz matrices belong to a class of matrices referred to as *persymmetric matrices* [44]. In general, a $p \times p$ matrix \mathbf{B} is referred to as persymmetric if its elements b_{kl} satisfy the following condition,

$$b_{kl} = b_{p-l+1, p-k+1} \quad \forall i, j \quad (3.59)$$

The above condition implies that the matrix is symmetric about its cross (northeast-southwest) diagonal. Equivalently, a persymmetric matrix satisfies the condition $\mathbf{B}^* = \mathbf{J}_p \mathbf{B}^H \mathbf{J}_p$ [43]. The following is an example of a persymmetric matrix.

$$\mathbf{B} = \begin{bmatrix} 1+j & 3-2j & 5-j & 7-3j \\ 2-5j & 2+5j & 4-2j & 5-j \\ 4-j & 4+j & 2+5j & 3-2j \\ 6+4j & 4-j & 2-5j & 1+j \end{bmatrix} \quad (3.60)$$

Using the above definitions, we now present the following properties for a block-Toeplitz matrix with Toeplitz blocks. We use $[\cdot]^T$ to denote matrix transposition.

Property 1

A block-Toeplitz matrix with Toeplitz blocks and its inverse satisfy the following relations [20] and is ‘‘persymmetric’’ in this Wax-Kailath sense.

$$\mathbf{J}_{p,q+1} \mathbf{T}_{p,q+1} \mathbf{J}_{p,q+1} = \mathbf{T}_{p,q+1}^T \quad (3.61a)$$

$$\mathbf{J}_{p,q+1} \mathbf{T}_{p,q+1}^{-1} \mathbf{J}_{p,q+1} = \mathbf{T}_{p,q+1}^{-T} \quad (3.61b)$$

Proof:

Performing the direct multiplication of the left-hand side of equation (3.61a) and using the relation $\mathbf{J}_p \mathbf{T}_i \mathbf{J}_p = \mathbf{T}_i^T$ we have,

$$\mathbf{J}_{p,q+1} \mathbf{T}_{p,q+1} \mathbf{J}_{p,q+1} = \begin{bmatrix} \mathbf{J}_p \mathbf{T}_{-q} & \mathbf{J}_p \mathbf{T}_{-q+1} & \mathbf{J}_p \mathbf{T}_{-q+2} & \cdots & \mathbf{J}_p \mathbf{T}_0 \\ \mathbf{J}_p \mathbf{T}_{-q+1} & \mathbf{J}_p \mathbf{T}_{-q+2} & \mathbf{J}_p \mathbf{T}_{-q+3} & \cdots & \mathbf{J}_p \mathbf{T}_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{J}_p \mathbf{T}_0 & \mathbf{J}_p \mathbf{T}_1 & \mathbf{J}_p \mathbf{T}_2 & \cdots & \mathbf{J}_p \mathbf{T}_q \end{bmatrix} \mathbf{J}_{p,q+1}$$

$$\begin{aligned}
\Rightarrow \mathbf{J}_{p,q+1} \mathbf{T}_{p,q+1} \mathbf{J}_{p,q+1} &= \begin{bmatrix} \mathbf{J}_p \mathbf{T}_0 \mathbf{J}_p & \mathbf{J}_p \mathbf{T}_{-1} \mathbf{J}_p & \mathbf{J}_p \mathbf{T}_{-2} \mathbf{J}_p & \cdots & \mathbf{J}_p \mathbf{T}_{-q} \mathbf{J}_p \\ \mathbf{J}_p \mathbf{T}_1 \mathbf{J}_p & \mathbf{J}_p \mathbf{T}_0 \mathbf{J}_p & \mathbf{J}_p \mathbf{T}_{-1} \mathbf{J}_p & \cdots & \mathbf{J}_p \mathbf{T}_{-q+1} \mathbf{J}_p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{J}_p \mathbf{T}_q \mathbf{J}_p & \mathbf{J}_p \mathbf{T}_{q-1} \mathbf{J}_p & \mathbf{J}_p \mathbf{T}_{q-2} \mathbf{J}_p & \cdots & \mathbf{J}_p \mathbf{T}_0 \mathbf{J}_p \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{T}_0^T & \mathbf{T}_{-1}^T & \mathbf{T}_{-2}^T & \cdots & \mathbf{T}_{-q}^T \\ \mathbf{T}_1^T & \mathbf{T}_0^T & \mathbf{T}_{-1}^T & \cdots & \mathbf{T}_{-q+1}^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_q^T & \mathbf{T}_{q-1}^T & \mathbf{T}_{q-2}^T & \cdots & \mathbf{T}_0^T \end{bmatrix} \\
&= \mathbf{T}_{p,q+1}^T \quad \square
\end{aligned} \tag{3.62}$$

Property (3.61b) is proved by inverting (3.61a) and using the relation in equation (3.58). However, to gain some insight into the structure of $\mathbf{T}_{p,q+1}^{-1}$ when its blocks are Hermitian-Toeplitz, we present a direct proof for this special case. First, we note that the inverse of a block-Toeplitz matrix with Hermitian-Toeplitz blocks is not block-Toeplitz in general and its individual blocks are also no longer Toeplitz. Instead, the inverse matrix has a block-wise persymmetry-like structure with the blocks symmetrically across the cross diagonal being conjugate-transpose (complex) or transpose (real) of each other instead of being equal. Also, the cross diagonal blocks ($\mathbf{B}_2, \mathbf{B}_3$ and \mathbf{B}_{-2}) are Hermitian. For $q = 2$, this structure is as shown below,

$$\mathbf{T}_{p,3}^{-1} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_{-1} & \mathbf{B}_3 & \mathbf{B}_1^H \\ \mathbf{B}_{-2} & \mathbf{B}_{-1}^H & \mathbf{B}_0^H \end{bmatrix} \tag{3.63}$$

Furthermore, the individual blocks were found to have a special structure. For an odd p , for instance $p = 5$, this structure is as shown below,

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 \\ b_6 & b_7 & b_8 & b_9 & b_{10} \\ b_{11} & b_{12} & b_{13} & b_{12}^* & b_{11}^* \\ b_{10}^* & b_9^* & b_8^* & b_7^* & b_6^* \\ b_5^* & b_4^* & b_3^* & b_2^* & b_1^* \end{bmatrix} \quad (3.64)$$

For an even p , say $p = 4$, the structure is shown below,

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_8 \\ b_8^* & b_7^* & b_6^* & b_5^* \\ b_4^* & b_3^* & b_2^* & b_1^* \end{bmatrix} \quad (3.65)$$

For matrices having the above structure, the following relation is found to be satisfied,

$$\mathbf{J}_p \mathbf{B} \mathbf{J}_p = \mathbf{B}^* \quad (3.66)$$

Using the above definitions, we develop the proof for (3.61b) for the case when $q = 2$. Directly multiplying the left-hand side of equation (3.61b) and utilizing property (3.66) as well as the fact that the cross diagonal blocks $\mathbf{B}_2, \mathbf{B}_3$ and \mathbf{B}_{-2} are Hermitian, we have,

$$\begin{aligned} \mathbf{J}_{p,3} \mathbf{T}_{p,3}^{-1} \mathbf{J}_{p,3} &= \begin{bmatrix} \mathbf{J}_p \mathbf{B}_{-2} & \mathbf{J}_p \mathbf{B}_{-1}^H & \mathbf{J}_p \mathbf{B}_0^H \\ \mathbf{J}_p \mathbf{B}_{-1} & \mathbf{J}_p \mathbf{B}_3 & \mathbf{J}_p \mathbf{B}_1^H \\ \mathbf{J}_p \mathbf{B}_0 & \mathbf{J}_p \mathbf{B}_1 & \mathbf{J}_p \mathbf{B}_2 \end{bmatrix} \mathbf{J}_{p,3} \\ &= \begin{bmatrix} \mathbf{J}_p \mathbf{B}_0^H \mathbf{J}_p & \mathbf{J}_p \mathbf{B}_{-1}^H \mathbf{J}_p & \mathbf{J}_p \mathbf{B}_{-2}^H \mathbf{J}_p \\ \mathbf{J}_p \mathbf{B}_1^H \mathbf{J}_p & \mathbf{J}_p \mathbf{B}_3^H \mathbf{J}_p & \mathbf{J}_p \mathbf{B}_{-1} \mathbf{J}_p \\ \mathbf{J}_p \mathbf{B}_2^H \mathbf{J}_p & \mathbf{J}_p \mathbf{B}_1 \mathbf{J}_p & \mathbf{J}_p \mathbf{B}_0 \mathbf{J}_p \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{B}_0^T & \mathbf{B}_{-1}^T & \mathbf{B}_{-2}^T \\ \mathbf{B}_1^T & \mathbf{B}_3^T & \mathbf{B}_{-1}^* \\ \mathbf{B}_2^T & \mathbf{B}_1^* & \mathbf{B}_0^* \end{bmatrix} \\ &= \mathbf{T}_{p,3}^{-T} \quad \square \end{aligned} \quad (3.67)$$

In a similar fashion, we can prove the property (3.61b) for any q .

When the blocks in $\mathbf{T}_{p,q+1}$ are only Toeplitz and not Hermitian, $\mathbf{T}_{p,q+1}^{-1}$ as well as its blocks have a different structure but the Wax-Kailath persymmetry property can be proved in a similar manner. This structure for $q = 2$ is shown below,

$$\mathbf{T}_{p,3}^{-1} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \\ \mathbf{B}_4 & \mathbf{B}_5 & \mathbf{B}_4^* \\ \mathbf{B}_3^* & \mathbf{B}_2^* & \mathbf{B}_1^* \end{bmatrix} \quad (3.68)$$

Moreover, the blocks in the inverse matrix are found to have the following structure (shown for $p = 4$),

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_3^* \\ b_8 & b_9 & b_6^* & b_2^* \\ b_{10} & b_8^* & b_5^* & b_1^* \end{bmatrix} \quad (3.69)$$

Matrices having the above structure satisfy the relation $\mathbf{J}_p \mathbf{B} \mathbf{J}_p = \mathbf{B}^H$.

Property 2

It is possible to convert the $(q+1) \times (q+1)$ block-Toeplitz matrix $\mathbf{T}_{p,q+1}$ with $p \times p$ Toeplitz blocks to another block-Toeplitz matrix $\mathbf{T}_{q+1,p}$ which is $p \times p$ and consists of Toeplitz blocks of size $(q+1) \times (q+1)$. Consider a $p(q+1) \times p(q+1)$ permutation matrix Ψ as defined below,

$$\Psi = \begin{bmatrix} \Psi_0 \\ \Psi_1 \\ \vdots \\ \Psi_{p-1} \end{bmatrix} \quad (3.70)$$

Here, Ψ_i is a $(q+1) \times (q+1)p$ matrix as given below,

$$\Psi_i = \begin{bmatrix} \Phi_i & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Phi_i & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Phi_i \end{bmatrix} \quad (3.71)$$

In the above equation, $\mathbf{0}$ is a $1 \times p$ vector of zeros and Φ_i is a $1 \times p$ vector consisting of zeros except for a 1 at the i^{th} position as shown,

$$\Phi_i = [0 \ 0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0] \quad (3.72)$$

Using the above definitions, we have the following property for a block-Toeplitz matrix with Toeplitz blocks.

$$\Psi \mathbf{T}_{p,q+1} \Psi^T = \check{\mathbf{T}}_{q+1,p} \quad (3.73)$$

Here $\check{\mathbf{T}}_{q+1,p}$ is a $p \times p$ block-Toeplitz matrix with Toeplitz blocks of size $(q+1) \times (q+1)$ as shown below,

$$\check{\mathbf{T}}_{q+1,p} = \begin{bmatrix} \check{\mathbf{T}}_0 & \check{\mathbf{T}}_1 & \check{\mathbf{T}}_2 & \cdots & \check{\mathbf{T}}_{p-1} \\ \check{\mathbf{T}}_{-1} & \check{\mathbf{T}}_0 & \check{\mathbf{T}}_1 & \cdots & \check{\mathbf{T}}_{p-2} \\ \check{\mathbf{T}}_{-2} & \check{\mathbf{T}}_{-1} & \check{\mathbf{T}}_0 & \cdots & \check{\mathbf{T}}_{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \check{\mathbf{T}}_{-p+1} & \check{\mathbf{T}}_{-p+2} & \check{\mathbf{T}}_{-p+3} & \cdots & \check{\mathbf{T}}_0 \end{bmatrix} \quad (3.74)$$

$\check{\mathbf{T}}_i$ are $(q+1) \times (q+1)$ matrices as shown below,

$$\check{\mathbf{T}}_i = \begin{bmatrix} t_{i,0} & t_{i,1}^* & t_{i,2}^* & \cdots & t_{i,q}^* \\ t_{i,1} & t_{i,0} & t_{i,1}^* & \cdots & t_{i,q-1}^* \\ t_{i,2} & t_{i,1} & t_{i,0} & \cdots & t_{i,q-2}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{i,q} & t_{i,q-1} & t_{i,q-2} & \cdots & t_{i,0} \end{bmatrix} \quad (3.75)$$

Proof:

Direct multiplication of the left-hand side of equation (3.73) yields the following,

$$\begin{aligned} \Psi \mathbf{T}_{p,q+1} \Psi^T &= \begin{bmatrix} \Psi_0 \\ \Psi_1 \\ \vdots \\ \Psi_{p-1} \end{bmatrix} \mathbf{T}_{p,q+1} \begin{bmatrix} \Psi_0^T & \Psi_1^T & \cdots & \Psi_{p-1}^T \end{bmatrix} \\ \Rightarrow \Psi \mathbf{T}_{p,q+1} \Psi^T &= \begin{bmatrix} \Psi_0 \mathbf{T}_{p,q+1} \Psi_0^T & \Psi_0 \mathbf{T}_{p,q+1} \Psi_1^T & \cdots & \Psi_0 \mathbf{T}_{p,q+1} \Psi_{p-1}^T \\ \Psi_1 \mathbf{T}_{p,q+1} \Psi_0^T & \Psi_1 \mathbf{T}_{p,q+1} \Psi_1^T & \cdots & \Psi_1 \mathbf{T}_{p,q+1} \Psi_{p-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{p-1} \mathbf{T}_{p,q+1} \Psi_0^T & \Psi_{p-1} \mathbf{T}_{p,q+1} \Psi_1^T & \cdots & \Psi_{p-1} \mathbf{T}_{p,q+1} \Psi_{p-1}^T \end{bmatrix} \end{aligned}$$

From the above, we see that the (m, g) block of $\Psi \mathbf{T}_{p,q+1} \Psi^T$ is given by $\Psi_m \mathbf{T}_{p,q+1} \Psi_g^T$. Denoting this block as $\check{\mathbf{T}}_i$, the (k, l) th element of this block is given by $\Phi_m \mathbf{T}_{l-k} \Phi_g = t_{g-m, l-k}$ which is a function of $(l - k)$ only and is therefore Toeplitz. In addition, the elements are a function of $(g - m)$ only which makes $\check{\mathbf{T}}_{q+1,p}$ a block-Toeplitz matrix with Toeplitz blocks. This proves the second property and relation (3.73). An important utility of this property will be presented in a later section on the computational complexity of the Wax-Kailath algorithm. In the next section, we present the Wax-Kailath recursions to efficiently invert the matrix $\mathbf{T}_{p,q+1}$. Later, we show how to use the error matrix \mathbf{E} and $\mathbf{T}_{p,q+1}^{-1}$ to obtain \mathbf{R}_{q+1}^{-1} in a given number of iterations using the Neumann series.

3.3.2 Wax-Kailath Recursions

The matrices $\mathbf{T}_{p,i}$ for $i = 1$ to $q + 1$ have a nested structure and we can represent these as,

$$\mathbf{T}_{p,i+1} = \begin{bmatrix} \mathbf{T}_{p,i} & \Upsilon_i \\ \Upsilon_{-i}^T & \mathbf{T}_0 \end{bmatrix} \quad (3.76)$$

Here,

$$\Upsilon_i \triangleq \begin{bmatrix} \mathbf{T}_i \\ \mathbf{T}_{i-1} \\ \vdots \\ \mathbf{T}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_i \\ \Upsilon_{i-1} \end{bmatrix} \quad (3.77)$$

and

$$\Upsilon_{-i}^T = \begin{bmatrix} \mathbf{T}_{-i} & \mathbf{T}_{-i+1} & \cdots & \mathbf{T}_{-1} \end{bmatrix} \quad (3.78)$$

Consider the following formula to invert a partitioned block matrix [41],

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} \quad (3.79)$$

Here \mathbf{A} and \mathbf{D} must be invertible. Using the above formula, we have,

$$\mathbf{T}_{p,i+1}^{-1} = \begin{bmatrix} \mathbf{T}_{p,i}^{-1} + \mathbf{T}_{p,i}^{-1}\Upsilon_i\Omega_i^{-1}\Upsilon_{-i}^T\mathbf{T}_{p,i}^{-1} & -\mathbf{T}_{p,i}^{-1}\Upsilon_i\Omega_i^{-1} \\ -\Omega_i^{-1}\Upsilon_{-i}^T\mathbf{T}_{p,i}^{-1} & \Omega_i^{-1} \end{bmatrix} \quad (3.80)$$

Here,

$$\Omega_i = \mathbf{T}_0 - \Upsilon_{-i}^T\mathbf{T}_{p,i}^{-1}\Upsilon_i \quad (3.81)$$

Now we define the following,

$$\mathbf{W}_i = -\mathbf{T}_{p,i}^{-1}\Upsilon_i \quad (3.82a)$$

$$\mathbf{V}_i^T = -\Upsilon_{-i}^T\mathbf{T}_{p,i}^{-1} \quad (3.82b)$$

Therefore,

$$\mathbf{T}_{p,i+1}^{-1} = \begin{bmatrix} \mathbf{T}_{p,i}^{-1} + \mathbf{W}_i\Omega_i^{-1}\mathbf{V}_i^T & \mathbf{W}_i\Omega_i^{-1} \\ \Omega_i^{-1}\mathbf{V}_i^T & \Omega_i^{-1} \end{bmatrix} \quad (3.83)$$

To compute the above inverse recursively, we need to develop recursive relations for \mathbf{W}_i , \mathbf{V}_i^T , and $\mathbf{\Omega}_i$. To derive the recursive relations, we first introduce the following notations,

$$\check{\mathbf{Y}}_i \triangleq \mathbf{J}_{p,i} \mathbf{Y}_i = \begin{bmatrix} \check{\mathbf{T}}_1 \\ \check{\mathbf{T}}_2 \\ \vdots \\ \check{\mathbf{T}}_i \end{bmatrix} \quad \text{where } \check{\mathbf{T}}_i = \mathbf{J}_p \mathbf{T}_i \quad (3.84a)$$

$$\check{\mathbf{Y}}_{-i}^T \triangleq \mathbf{Y}_{-i}^T \mathbf{J}_{p,i} = \begin{bmatrix} \check{\mathbf{T}}_{-1} & \check{\mathbf{T}}_{-2} & \cdots & \check{\mathbf{T}}_{-i} \end{bmatrix} \quad \text{where } \check{\mathbf{T}}_{-i} = \mathbf{T}_{-i} \mathbf{J}_p \quad (3.84b)$$

The notational equivalent of 3.84a in [20] is missing an underline causing it to be misinterpreted as $\check{\mathbf{T}}_i$. Applying matrix transposition to equation (3.83) we obtain,

$$\mathbf{T}_{p,i+1}^{-T} = \begin{bmatrix} \mathbf{T}_{p,i}^{-T} + \mathbf{V}_i \mathbf{\Omega}_i^{-T} \mathbf{W}_i^T & \mathbf{V}_i \mathbf{\Omega}_i^{-T} \\ \mathbf{\Omega}_i^{-T} \mathbf{W}_i^T & \mathbf{\Omega}_i^{-T} \end{bmatrix} \quad (3.85)$$

Utilizing the fact that $\mathbf{J}_{p,i+1} \mathbf{J}_{p,i+1} = \mathbf{I}_{p,i+1}$ we pre- and post- multiply equation (3.61b) by $\mathbf{J}_{p,i+1}$ to obtain the following.

$$\mathbf{T}_{p,i+1}^{-1} = \mathbf{J}_{p,i+1} \mathbf{T}_{p,i+1}^{-T} \mathbf{J}_{p,i+1} \quad (3.86)$$

Using the notations in (3.84) and equations (3.82a), (3.84a), (3.86), and (3.85) we have,

$$\begin{aligned} \mathbf{W}_{i+1} &= -\mathbf{T}_{p,i+1}^{-1} \mathbf{Y}_{i+1} \quad (3.82a) \\ &= -\mathbf{T}_{p,i+1}^{-1} \mathbf{J}_{p,i+1} \mathbf{J}_{p,i+1} \mathbf{Y}_{i+1} \\ &= -\mathbf{T}_{p,i+1}^{-1} \mathbf{J}_{p,i+1} \check{\mathbf{Y}}_{i+1} \quad (3.84a) \\ &= -\mathbf{J}_{p,i+1} \mathbf{T}_{p,i+1}^{-T} \mathbf{J}_{p,i+1} \mathbf{J}_{p,i+1} \check{\mathbf{Y}}_{i+1} \quad (3.86) \\ &= -\mathbf{J}_{p,i+1} \mathbf{T}_{p,i+1}^{-T} \check{\mathbf{Y}}_{i+1} \\ &= -\mathbf{J}_{p,i+1} \begin{bmatrix} \mathbf{T}_{p,i}^{-T} + \mathbf{V}_i \mathbf{\Omega}_i^{-T} \mathbf{W}_i^T & \mathbf{V}_i \mathbf{\Omega}_i^{-T} \\ \mathbf{\Omega}_i^{-T} \mathbf{W}_i^T & \mathbf{\Omega}_i^{-T} \end{bmatrix} \begin{bmatrix} \check{\mathbf{Y}}_i \\ \check{\mathbf{T}}_{i+1} \end{bmatrix} \quad (3.85) \\ &= -\mathbf{J}_{p,i+1} \begin{bmatrix} \mathbf{T}_{p,i}^{-T} \check{\mathbf{Y}}_i + \mathbf{V}_i \mathbf{\Omega}_i^{-T} (\mathbf{W}_i^T \check{\mathbf{Y}}_i + \check{\mathbf{T}}_{i+1}) \\ \mathbf{\Omega}_i^{-T} (\mathbf{W}_i^T \check{\mathbf{Y}}_i + \check{\mathbf{T}}_{i+1}) \end{bmatrix} \quad (3.87) \end{aligned}$$

Now we define $\mathbf{\Gamma}_i$ as,

$$\mathbf{\Gamma}_i = \mathbf{W}_i^T \check{\mathbf{\Upsilon}}_i + \check{\mathbf{T}}_{i+1} \quad (3.88)$$

Using property (3.86) and equation (3.82a) we derive the following relation for $\check{\mathbf{W}}_i$,

$$\begin{aligned} \check{\mathbf{W}}_i &= -\mathbf{J}_{p,i} \mathbf{T}_{p,i}^{-1} \check{\mathbf{\Upsilon}}_i \\ &= -\mathbf{J}_{p,i} \mathbf{J}_{p,i} \mathbf{T}_{p,i}^{-T} \mathbf{J}_{p,i} \check{\mathbf{\Upsilon}}_i \\ &= -\mathbf{T}_{p,i}^{-T} \check{\mathbf{\Upsilon}}_i \end{aligned} \quad (3.89)$$

Using equations (3.87–3.89) we have,

$$\begin{aligned} \check{\mathbf{W}}_{i+1} &= - \begin{bmatrix} \mathbf{T}_{p,i}^{-T} \check{\mathbf{\Upsilon}}_i + \mathbf{V}_i \mathbf{\Omega}_i^{-T} \mathbf{\Gamma}_i \\ \mathbf{\Omega}_i^{-T} \mathbf{\Gamma}_i \end{bmatrix} \\ \Rightarrow \check{\mathbf{W}}_{i+1} &= \begin{bmatrix} \check{\mathbf{W}}_i \\ \mathbf{0}_{p \times p} \end{bmatrix} - \begin{bmatrix} \mathbf{V}_i \\ \mathbf{I}_{p \times p} \end{bmatrix} \mathbf{\Omega}_i^{-T} \mathbf{\Gamma}_i \end{aligned} \quad (3.90)$$

Now we derive a recursive relation for $\check{\mathbf{V}}_{i+1}$. Applying matrix transposition to equation (3.82b) we obtain $\mathbf{V}_i = -\mathbf{T}_{p,i}^{-T} \check{\mathbf{\Upsilon}}_{-i}$. Therefore, from equations (3.86), (3.84), and (3.83) we have,

$$\begin{aligned} \mathbf{V}_{i+1} &= -\mathbf{T}_{p,i+1}^{-T} \check{\mathbf{\Upsilon}}_{-i-1} \\ &= -\mathbf{J}_{p,i+1} \mathbf{T}_{p,i+1}^{-1} \check{\mathbf{\Upsilon}}_{-i-1} \\ &= -\mathbf{J}_{p,i+1} \begin{bmatrix} \mathbf{T}_{p,i}^{-1} + \mathbf{W}_i \mathbf{\Omega}_i^{-1} \mathbf{V}_i^T & \mathbf{W}_i \mathbf{\Omega}_i^{-1} \\ \mathbf{\Omega}_i^{-1} \mathbf{V}_i^T & \mathbf{\Omega}_i^{-1} \end{bmatrix} \begin{bmatrix} \check{\mathbf{\Upsilon}}_{-i} \\ \check{\mathbf{T}}_{-i-1} \end{bmatrix} \\ &= -\mathbf{J}_{p,i+1} \begin{bmatrix} \mathbf{T}_{p,i}^{-1} \check{\mathbf{\Upsilon}}_{-i} + \mathbf{W}_i \mathbf{\Omega}_i^{-1} (\mathbf{V}_i^T \check{\mathbf{\Upsilon}}_{-i} + \check{\mathbf{T}}_{-i-1}) \\ \mathbf{\Omega}_i^{-1} (\mathbf{V}_i^T \check{\mathbf{\Upsilon}}_{-i} + \check{\mathbf{T}}_{-i-1}) \end{bmatrix} \end{aligned} \quad (3.91)$$

We define $\mathbf{\Delta}_i$ as shown below,

$$\mathbf{\Delta}_i = \mathbf{V}_i^T \check{\mathbf{\Upsilon}}_{-i} + \check{\mathbf{T}}_{-i-1} \quad (3.92)$$

The equivalent of $\mathbf{\Delta}_i$ in [20] is notated wrongly using the equivalent of $\mathbf{\Gamma}_i$ and its definition should not have a transpose in the second term. Using (3.61b) and (3.84) we have,

$$\check{\mathbf{V}}_i = -\mathbf{T}_{p,i}^{-1} \check{\mathbf{\Upsilon}}_{-i} \quad (3.93)$$

Now using equations (3.91–3.93) we have,

$$\begin{aligned}\check{\mathbf{V}}_{i+1} &= - \begin{bmatrix} \mathbf{T}_{p,i}^{-1} \check{\mathbf{\Upsilon}}_{-i} + \mathbf{W}_i \Omega_i^{-1} \Delta_i \\ \Omega_i^{-1} \Delta_i \end{bmatrix} \\ \Rightarrow \check{\mathbf{V}}_{i+1} &= \begin{bmatrix} \check{\mathbf{V}}_i \\ \mathbf{0}_{p \times p} \end{bmatrix} - \begin{bmatrix} \mathbf{W}_i \\ \mathbf{I}_{p \times p} \end{bmatrix} \Omega_i^{-1} \Delta_i\end{aligned}\quad (3.94)$$

Using the above recursive relation and equations (3.81), (3.82b), and (3.88) we have,

$$\begin{aligned}\Omega_{i+1} - \Omega_i &= \mathbf{\Upsilon}_{-i}^T \mathbf{T}_{p,i}^{-1} \mathbf{\Upsilon}_i - \mathbf{\Upsilon}_{-i-1}^T \mathbf{T}_{p,i+1}^{-1} \mathbf{\Upsilon}_{i+1} \quad (3.81) \\ &= -\mathbf{V}_i^T \mathbf{\Upsilon}_i + \mathbf{V}_{i+1}^T \mathbf{\Upsilon}_{i+1} \quad (3.82b) \\ &= \mathbf{V}_{i+1}^T \mathbf{J}_{p,i+1} \mathbf{J}_{p,i+1} \mathbf{\Upsilon}_{i+1} - \mathbf{V}_i^T \mathbf{J}_{p,i} \mathbf{J}_{p,i} \mathbf{\Upsilon}_i \\ &= \check{\mathbf{V}}_{i+1}^T \check{\mathbf{\Upsilon}}_{i+1} - \check{\mathbf{V}}_i^T \check{\mathbf{\Upsilon}}_i \\ &= \left(\begin{bmatrix} \check{\mathbf{V}}_i^T & \mathbf{0}_{p \times p} \end{bmatrix} - \Delta_i^T \Omega_i^{-T} \begin{bmatrix} \mathbf{W}_i^T & \mathbf{I}_{p \times p} \end{bmatrix} \right) \begin{bmatrix} \check{\mathbf{\Upsilon}}_i \\ \check{\mathbf{T}}_{i+1} \end{bmatrix} - \check{\mathbf{V}}_i^T \check{\mathbf{\Upsilon}}_i \quad (3.94) \\ &= -\Delta_i^T \Omega_i^{-T} \left(\mathbf{W}_i^T \check{\mathbf{\Upsilon}}_i + \check{\mathbf{T}}_{i+1} \right) \\ \Rightarrow \Omega_{i+1} &= \Omega_i - \Delta_i^T \Omega_i^{-T} \Gamma_i \quad (3.88)\end{aligned}\quad (3.95)$$

Using the recursive relations in (3.90), (3.94), and (3.95) we can compute $\mathbf{T}_{p,i+1}^{-1}$ recursively.

Using equation (3.86) and (3.85), we have the following alternative expression for $\mathbf{T}_{p,q+1}^{-1}$,

$$\mathbf{T}_{p,q+1}^{-1} = \begin{bmatrix} \mathbf{J}_p \Omega_q^{-T} \mathbf{J}_p & \mathbf{J}_p \Omega_q^{-T} \check{\mathbf{W}}_q^T \\ \check{\mathbf{V}}_q \Omega_q^{-T} \mathbf{J}_p & \mathbf{T}_{p,q}^{-1} + \check{\mathbf{V}}_q \Omega_q^{-T} \check{\mathbf{W}}_q^T \end{bmatrix}\quad (3.96)$$

Here, the first block row and column can be formed once $\check{\mathbf{W}}_q$, $\check{\mathbf{V}}_q$, and Ω_q have been determined recursively. The rest of the inverse matrix can be computed from the first block row and column as discussed next. From the original expression for $\mathbf{T}_{p,q+1}^{-1}$ in equation (3.83) we can represent its $(m, g)^{\text{th}}$ block of as follows,

$$\left(\mathbf{T}_{p,q+1}^{-1} \right)_{m,g} = \left(\mathbf{T}_{p,q}^{-1} + \mathbf{W}_q \Omega_q^{-1} \mathbf{V}_q^T \right)_{m,g}\quad (3.97)$$

Using the alternative expression in (3.96) we have the $(m+1, g+1)^{\text{th}}$ block of $\mathbf{T}_{p,q+1}^{-1}$ as,

$$\left(\mathbf{T}_{p,q+1}^{-1} \right)_{m+1,g+1} = \left(\mathbf{T}_{p,q}^{-1} + \check{\mathbf{V}}_q \Omega_q^{-T} \check{\mathbf{W}}_q^T \right)_{m,g}\quad (3.98)$$

Subtracting equation (3.97) from equation (3.98) we have,

$$(\mathbf{T}_{p,q+1}^{-1})_{m+1,g+1} = (\mathbf{T}_{p,q+1}^{-1})_{m,g} + \left(\check{\mathbf{V}}_q \boldsymbol{\Omega}_q^{-T} \check{\mathbf{W}}_q^T - \mathbf{W}_q \boldsymbol{\Omega}_q^{-1} \mathbf{V}_q^T \right)_{m,g} \quad (3.99)$$

The above recursion for computing the blocks combined with the recursive relations in equations (3.90), (3.94), and (3.95) constitute the Wax-Kailath algorithm for inverting a block-Toeplitz matrix with Toeplitz blocks. We note that the nonsingularity of $\boldsymbol{\Omega}_i$ is the only necessary assumption required for the validity of the recursions. The steps for implementing the Wax-Kailath Algorithm are outlined below.

1. Initialization ($i = 0$): Set $\mathbf{T}_{p,i+1}^{-1} = \mathbf{T}_0^{-1}$. Compute the following,

- $\mathbf{W}_{i+1} = -\mathbf{T}_0^{-1} \mathbf{T}_1$
- $\mathbf{V}_{i+1} = -\mathbf{T}_0^{-T} \mathbf{T}_{-1}^T$
- $\boldsymbol{\Omega}_{i+1} = \mathbf{T}_0 - \mathbf{T}_{-1} \mathbf{T}_0^{-1} \mathbf{T}_1$
- $\check{\mathbf{W}}_{i+1} = \mathbf{J}_p \mathbf{W}_{i+1}$ and $\check{\mathbf{V}}_{i+1} = \mathbf{J}_p \mathbf{V}_{i+1}$
- $\boldsymbol{\Upsilon}_{i+1} = \mathbf{T}_1$
- $\boldsymbol{\Upsilon}_{-i-1}^T = \mathbf{T}_{-1}$

2. Recursion:

For $i = 1$ to $q - 1$

- $\check{\boldsymbol{\Upsilon}}_i = \mathbf{J}_{p,i} \boldsymbol{\Upsilon}_i$
- $\check{\mathbf{T}}_{i+1} = \mathbf{J}_p \mathbf{T}_{i+1}$
- Use equation (3.88) to compute $\boldsymbol{\Gamma}_i$
- Then use equation (3.90) to compute $\check{\mathbf{W}}_{i+1}$
- $\check{\boldsymbol{\Upsilon}}_{-i} = (\boldsymbol{\Upsilon}_{-i}^T \mathbf{J}_{p,i})^T$
- $\check{\mathbf{T}}_{-i-1} = \mathbf{T}_{-i-1} \mathbf{J}_p$
- Use equation (3.92) to compute $\boldsymbol{\Delta}_i$

- Then use equation (3.94) to compute $\check{\check{\mathbf{V}}}_{i+1}$
- Compute $\check{\check{\mathbf{\Omega}}}_{i+1}$ using equation (3.95)
- Compute $\check{\check{\mathbf{W}}}_{i+1} = \mathbf{J}_{p,i+1} \check{\check{\mathbf{W}}}_{i+1}$ and $\check{\check{\mathbf{V}}}_{i+1} = \mathbf{J}_{p,i+1} \check{\check{\mathbf{V}}}_{i+1}$ for the next iteration.
- Update $\check{\check{\mathbf{T}}}_{i+1} = \begin{bmatrix} \check{\check{\mathbf{T}}}_{i+1} \\ \check{\check{\mathbf{T}}}_i \end{bmatrix}$ and $\check{\check{\mathbf{T}}}_{-i-1}^T = \begin{bmatrix} \check{\check{\mathbf{T}}}_{-i-1} & \check{\check{\mathbf{T}}}_{-i}^T \end{bmatrix}$

end

3. After computing the recursions, we have the following: $\check{\check{\mathbf{\Omega}}}_q$, $\check{\check{\mathbf{W}}}_q$, $\check{\check{\mathbf{V}}}_q$, $\check{\check{\mathbf{W}}}_q$, and $\check{\check{\mathbf{V}}}_q$.

Using these, we form $\check{\check{\mathbf{T}}}_{p,q+1}^{-1}$ as follows,

- Initialize $\check{\check{\mathbf{T}}}_{p,q+1}^{-1} = \begin{bmatrix} \mathbf{J}_p \check{\check{\mathbf{\Omega}}}_q^{-T} \mathbf{J}_p & \mathbf{J}_p \check{\check{\mathbf{\Omega}}}_q^{-T} \check{\check{\mathbf{W}}}_q^T \\ \check{\check{\mathbf{V}}}_q \check{\check{\mathbf{\Omega}}}_q^{-T} \mathbf{J}_p & \mathbf{0}_{pq \times pq} \end{bmatrix}$
- Compute $\check{\check{\mathbf{B}}} = \check{\check{\mathbf{V}}}_q \check{\check{\mathbf{\Omega}}}_q^{-T} \check{\check{\mathbf{W}}}_q^T - \check{\check{\mathbf{W}}}_q \check{\check{\mathbf{\Omega}}}_q^{-1} \check{\check{\mathbf{V}}}_q^T$
- Using m and g to index the blocks, we use equation (3.99) to compute the remaining blocks of the inverse matrix as follows, For $m = 1$ to q

For $g = 1$ to q

$$\left(\check{\check{\mathbf{T}}}_{p,q+1}^{-1}\right)_{m+1,g+1} = \left(\check{\check{\mathbf{T}}}_{p,q+1}^{-1}\right)_{m,g} + (\check{\check{\mathbf{B}}})_{m,g}$$

end

end

3.3.3 Computational Complexity

In this section, we present the computational complexity of the Wax-Kailath algorithm. Compared to the Akaike algorithm which requires six recursive relations to compute the first block row and column of the inverse, the Wax-Kailath algorithm requires only three recursions and therefore achieves a factor of 2 reduction in the number of recursions. In addition, the persymmetry property of the inverse shown in equation (3.61b) can be used to reduce by a factor of 2 the number of computations involved in forming the inverse from the first block row and column using (3.99). For instance, when the block-Toeplitz matrix $\check{\check{\mathbf{T}}}_{p,q+1}$ has Hermitian-Toeplitz blocks, the blocks symmetric with respect to the cross diagonal in the

inverse matrix $\mathbf{T}_{p,q+1}^{-1}$ are complex conjugate transposes of each other as shown in equation (3.63) and therefore we need to compute only half of such blocks. When the blocks are only Toeplitz and not Hermitian, we can still achieve the factor of two reduction realizing that the inverse matrix $\mathbf{T}_{p,q+1}^{-1}$ has the structure as shown in equation (3.68). However, the order of magnitude of the number of operations in the Wax-Kailath algorithm remains at $O(p^3(q+1)^2)$ as in Akaike's algorithm.

When $p > (q+1)$, significant computational savings can be achieved by utilizing **Property 2** which states that we can interchange the dimensions p and $(q+1)$ of $\mathbf{T}_{p,q+1}$ by converting it into another $p \times p$ block-Toeplitz matrix $\check{\mathbf{T}}_{q+1,p}$ with Toeplitz blocks of size $(q+1) \times (q+1)$. With this transformation, we can compute $\check{\mathbf{T}}_{q+1,p}^{-1}$ using the Wax-Kailath algorithm in $O((q+1)^3 p^2)$ operations. Since $\mathbf{\Psi}\mathbf{\Psi}^T = \mathbf{I}$, we can then compute $\mathbf{T}_{p,q+1}^{-1} = \mathbf{\Psi}^T \check{\mathbf{T}}_{q+1,p}^{-1} \mathbf{\Psi}$. Effectively, the Wax-Kailath algorithm outperforms the Akaike algorithm by a factor of $\max(2(p/(q+1)), 2)$ which is significant when $p \gg (q+1)$ [20].

The Wax-Kailath recursion involves computing the inverse of the matrix $\mathbf{\Omega}_{p \times p}$. For $p = 2, 3$ we can efficiently invert this matrix using the analytical expressions given in equations (3.49) and (3.50). For $p > 3$ we have to resort to Gaussian elimination. In our simulations, we find that the matrix $\mathbf{\Omega}_{p \times p}$ and its inverse have the same structure as the blocks in the inverse matrix $\mathbf{T}_{p,q+1}^{-1}$. When the blocks in $\mathbf{T}_{p,q+1}$ are Hermitian-Toeplitz, $\mathbf{\Omega}_{p \times p}$ and its inverse inherit the structure shown in (3.64) for odd p and the structure in (3.65) for even p . When the blocks in $\mathbf{T}_{p,q+1}$ are only Toeplitz, the structure is as shown in (3.69). Therefore, it is desirable to have efficient schemes to invert matrices having these structures as this would enable us to reduce the computational requirements in the Wax-Kailath algorithm further. One such scheme is now presented and applies to the structure in (3.65). First we represent the matrix \mathbf{B} in equation (3.65) as shown below,

$$\mathbf{B} = \begin{bmatrix} \mathbf{S}_0 & \mathbf{S}_1 \\ (\mathbf{J}_p \mathbf{S}_1 \mathbf{J}_p)^* & (\mathbf{J}_p \mathbf{S}_0 \mathbf{J}_p)^* \end{bmatrix} \quad (3.100)$$

The matrix \mathbf{B}^{-1} has the same structure as \mathbf{B} and is therefore represented as,

$$\mathbf{B}^{-1} = \begin{bmatrix} \mathbf{Q}_0 & \mathbf{Q}_1 \\ (\mathbf{J}_p \mathbf{Q}_1 \mathbf{J}_p)^* & (\mathbf{J}_p \mathbf{Q}_0 \mathbf{J}_p)^* \end{bmatrix} \quad (3.101)$$

Using $\mathbf{B}\mathbf{B}^{-1} = \mathbf{I}$ we arrive at the following relations,

$$\mathbf{S}_0 \mathbf{Q}_0 + \mathbf{S}_1 (\mathbf{J}_p \mathbf{Q}_1 \mathbf{J}_p)^* = \mathbf{I} \quad (3.102a)$$

$$(\mathbf{J}_p \mathbf{S}_1 \mathbf{J}_p)^* \mathbf{Q}_1 + (\mathbf{J}_p \mathbf{S}_0 \mathbf{J}_p)^* (\mathbf{J}_p \mathbf{Q}_0 \mathbf{J}_p)^* = \mathbf{I} \quad (3.102b)$$

$$\mathbf{S}_0 \mathbf{Q}_1 + \mathbf{S}_1 (\mathbf{J}_p \mathbf{Q}_0 \mathbf{J}_p)^* = \mathbf{0} \quad (3.102c)$$

$$(\mathbf{J}_p \mathbf{S}_1 \mathbf{J}_p)^* \mathbf{Q}_0 + (\mathbf{J}_p \mathbf{S}_0 \mathbf{J}_p)^* (\mathbf{J}_p \mathbf{Q}_1 \mathbf{J}_p)^* = \mathbf{0} \quad (3.102d)$$

From equation (3.102a),

$$\mathbf{Q}_0 = \mathbf{S}_0^{-1} - \mathbf{S}_0^{-1} \mathbf{S}_1 \mathbf{J}_p \mathbf{Q}_1^* \mathbf{J}_p \quad (3.103)$$

Taking the complex conjugate of equation (3.102c) we have,

$$\mathbf{S}_1^* \mathbf{J}_p \mathbf{Q}_0 \mathbf{J}_p = -\mathbf{S}_0^* \mathbf{Q}_1^* \quad (3.104)$$

Substituting for \mathbf{Q}_0 from (3.103) in the above equation we have,

$$\begin{aligned} \mathbf{S}_1^* \mathbf{J}_p (\mathbf{S}_0^{-1} - \mathbf{S}_0^{-1} \mathbf{S}_1 \mathbf{J}_p \mathbf{Q}_1^* \mathbf{J}_p) \mathbf{J}_p &= -\mathbf{S}_0^* \mathbf{Q}_1^* \\ \mathbf{S}_1^* \mathbf{J}_p \mathbf{S}_0^{-1} \mathbf{J}_p &= \mathbf{S}_1^* \mathbf{J}_p \mathbf{S}_0^{-1} \mathbf{S}_1 \mathbf{J}_p \mathbf{Q}_1^* - \mathbf{S}_0^* \mathbf{Q}_1^* \\ \mathbf{S}_1 \mathbf{J}_p \mathbf{S}_0^{-*} \mathbf{J}_p &= (\mathbf{S}_1 \mathbf{J}_p \mathbf{S}_0^{-*} \mathbf{S}_1^* \mathbf{J}_p - \mathbf{S}_0) \mathbf{Q}_1 \\ \Rightarrow \mathbf{Q}_1 &= (\mathbf{S}_1 \mathbf{J}_p \mathbf{S}_0^{-*} \mathbf{S}_1^* \mathbf{J}_p - \mathbf{S}_0)^{-1} \mathbf{S}_1 \mathbf{J}_p \mathbf{S}_0^{-*} \mathbf{J}_p \end{aligned} \quad (3.105)$$

Once \mathbf{Q}_0 and \mathbf{Q}_1 are computed using equations (3.103) and (3.105) respectively, we can form \mathbf{B}^{-1} as shown in (3.101). This approach is more efficient than direct inversion of \mathbf{B} because it involves computing inverses of matrices with dimension smaller by a factor of 2 compared to the dimension of \mathbf{B} . Although the Wax-Kailath algorithm provides significant computational advantages over the Akaike algorithm, we have to perform additional computations in the form of iterations to obtain \mathbf{R}_{q+1}^{-1} – which is the inverse of the covariance matrix of our interest – from $\mathbf{T}_{p,q+1}^{-1}$ and the error matrix \mathbf{E} . This procedure is discussed next.

3.3.4 Neumann Series

In this section, we introduce the Neumann series [41] and show how it can be used to obtain the inverse of the block-Toeplitz matrix \mathbf{R}_{q+1} . Using the Wax-Kailath algorithm, we have obtained the inverse matrix of the block-Toeplitz matrix $\mathbf{T}_{p,q+1}$ with Hermitian-Toeplitz blocks. However, our interest is in computing the inverse of the block-Toeplitz covariance matrix \mathbf{R}_{q+1} with Hermitian blocks. Therefore, from (3.51),

$$\mathbf{R}_{q+1}^{-1} = (\mathbf{T}_{p,q+1} - \mathbf{E})^{-1} \quad (3.106)$$

Consider a matrix $\mathbf{A} \in \mathcal{C}^{n \times n}$. For this matrix, the following statements are equivalent [41],

- The Neumann series shown below converges.

$$\sum_{k=0}^{\infty} \mathbf{A}^k = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \dots$$

- The spectral radius of \mathbf{A} denoted as $\rho(\mathbf{A})$ must be less than one.
- The following holds,

$$\lim_{k \rightarrow \infty} \mathbf{A}^k = \mathbf{0} \quad (3.107)$$

- $\mathbf{I} - \mathbf{A}$ is nonsingular and,

$$(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \dots = \sum_{k=0}^{\infty} \mathbf{A}^k \quad (3.108)$$

The set of all eigenvalues (λ) of \mathbf{A} is referred to as its spectrum and denoted $\lambda(\mathbf{A})$. The spectral radius of a matrix is its maximum absolute eigenvalue and defined as shown below,

$$\rho(\mathbf{A}) = \max_{\lambda \in \lambda(\mathbf{A})} |\lambda| \quad (3.109)$$

Using the Neumann series defined in (3.3.4) we rewrite the inverse in equation (3.106) as follows,

$$\begin{aligned}
\mathbf{R}_{q+1}^{-1} &= (\mathbf{T}_{p,q+1} - \mathbf{E})^{-1} \\
&= (\mathbf{T}_{p,q+1} (\mathbf{I} - [\mathbf{T}_{p,q+1}^{-1} \mathbf{E}]))^{-1} \\
&= (\mathbf{I} - [\mathbf{T}_{p,q+1}^{-1} \mathbf{E}])^{-1} \mathbf{T}_{p,q+1}^{-1} \\
\Rightarrow \mathbf{R}_{q+1}^{-1} &= \left(\sum_{k=0}^{\infty} [\mathbf{T}_{p,q+1}^{-1} \mathbf{E}]^k \right) \mathbf{T}_{p,q+1}^{-1} \tag{3.110}
\end{aligned}$$

The above equation provides an explicit relation for \mathbf{R}_{q+1}^{-1} in terms of $\mathbf{T}_{p,q+1}^{-1}$ and \mathbf{E} both of which are available. The convergence of the above series is guaranteed if $\rho([\mathbf{T}_{p,q+1}^{-1} \mathbf{E}]) < 1$ which also implies,

$$\lim_{k \rightarrow \infty} [\mathbf{T}_{p,q+1}^{-1} \mathbf{E}]^k = \mathbf{0} \tag{3.111}$$

If the series converges, we can approximate \mathbf{R}_{q+1}^{-1} using a finite number of iterations and compute the solution vector estimate $\hat{\mathbf{a}} = -\hat{\mathbf{R}}_{q+1}^{-1} \mathbf{d}$. Using this estimate, we can compute an estimate of the solution of the linear system of equations $\mathbf{C} \mathbf{a} = -\mathbf{d}$ as follows,

$$\hat{\mathbf{a}}_{WK} = \mathbf{F}_{q+1} \hat{\mathbf{a}} \tag{3.112}$$

3.3.5 Experimental Verification

In this section, we show an example simulation to demonstrate the workings of a MATLAB implementation of the Wax-Kailath algorithm. We compare the solution vector \mathbf{a} obtained by solving the linear system of equations $\mathbf{C} \mathbf{a} = -\mathbf{d}$ using Gaussian elimination and the solution vector estimate $\hat{\mathbf{a}}_{WK}$ in equation (3.112) obtained using the Wax-Kailath algorithm. We use the matrix \mathbf{C} obtained in the linear FM example in Section 2.3.2 to form the block-Toeplitz matrix \mathbf{R}_{q+1} and subsequently the block-Toeplitz matrix $\mathbf{T}_{p,q+1}$ with Hermitian-Toeplitz blocks. The error matrix \mathbf{E} is formed as shown in equations (3.52) and (3.53). Experimentally, we find that at least 9 iterations have to be used to obtain \mathbf{R}_{q+1}^{-1} using the Neumann series in (3.110). For this simulation we obtain the spectral radius as $\rho([\mathbf{T}_{p,q+1}^{-1} \mathbf{E}]) = 0.2838$. In Figure 3.3 we show the real and imaginary parts of the

Gaussian elimination solution \mathbf{a} and the Wax-Kailath solution estimate $\hat{\mathbf{a}}_{WK}$. We see that the Wax-Kailath algorithm results in a solution approximately equal to the Gaussian elimination solution.

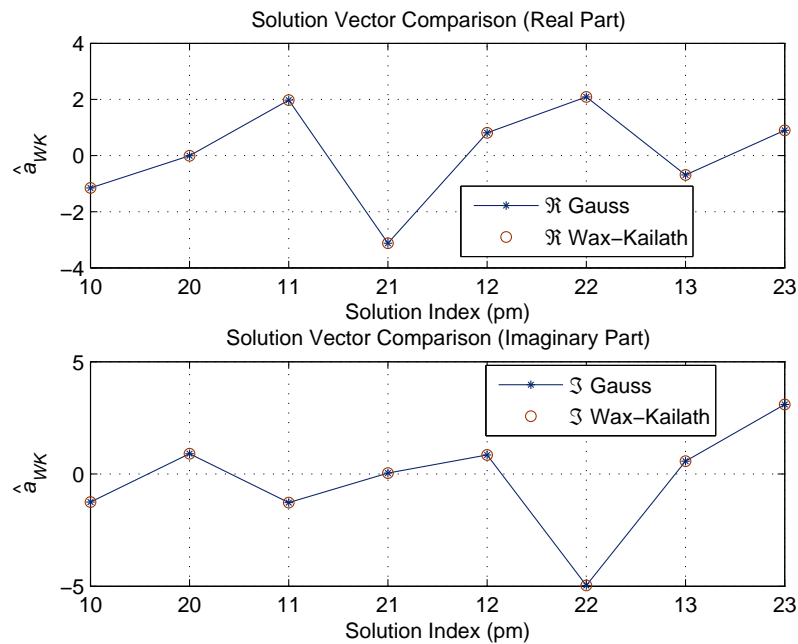


Figure 3.3: Solution Vector Comparison for Wax-Kailath Algorithm.

The magnitude of solution errors $\mathbf{e} = \mathbf{C}\mathbf{a} + \mathbf{d}$ and $\hat{\mathbf{e}} = \mathbf{C}\hat{\mathbf{a}}_{WK} + \mathbf{d}$ are shown in Figure 3.4. From Figure 3.4, we see that magnitude of error is negligible in either approach and it is smaller for Gaussian elimination compared to the Wax-Kailath algorithm. Comparing to the Akaike algorithm, we immediately notice that the magnitude of error is about 3 orders of magnitude bigger. However, by increasing the number of iterations to 15 in the Neumann series we can make the magnitude of solution error in Wax-Kailath algorithm comparable to that with the Akaike algorithm. The magnitude of the error in the solution for 15 iterations is shown in Figure 3.5. The Neumann series approximation approach using i iterations requires $i + 1$ matrix multiplications and i matrix additions. The computational complexity associated with the matrix multiplications is $O(p^3(q + 1)^3)$. Therefore, if i is large, the computational effort increases by a factor of i .

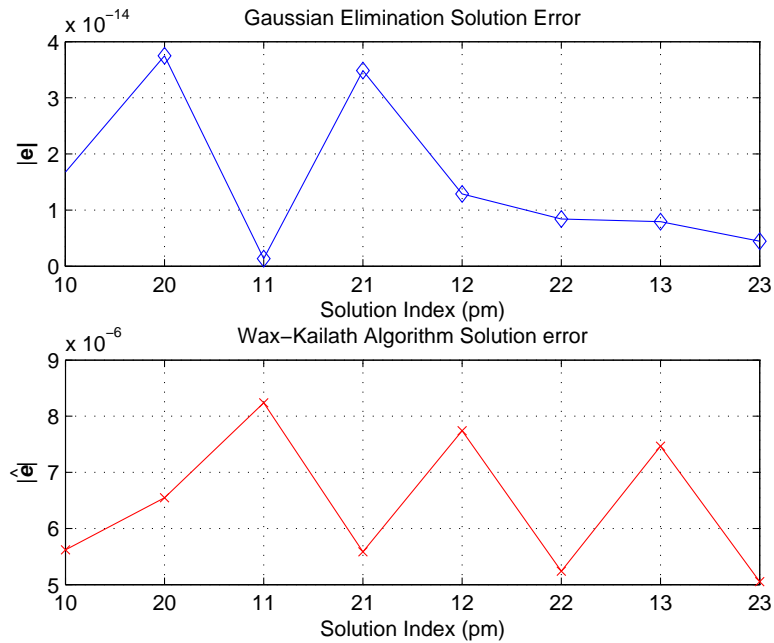


Figure 3.4: Magnitude of Solution Error for Gaussian Elimination and Wax-Kailath Algorithm (9 Neumann iterations).

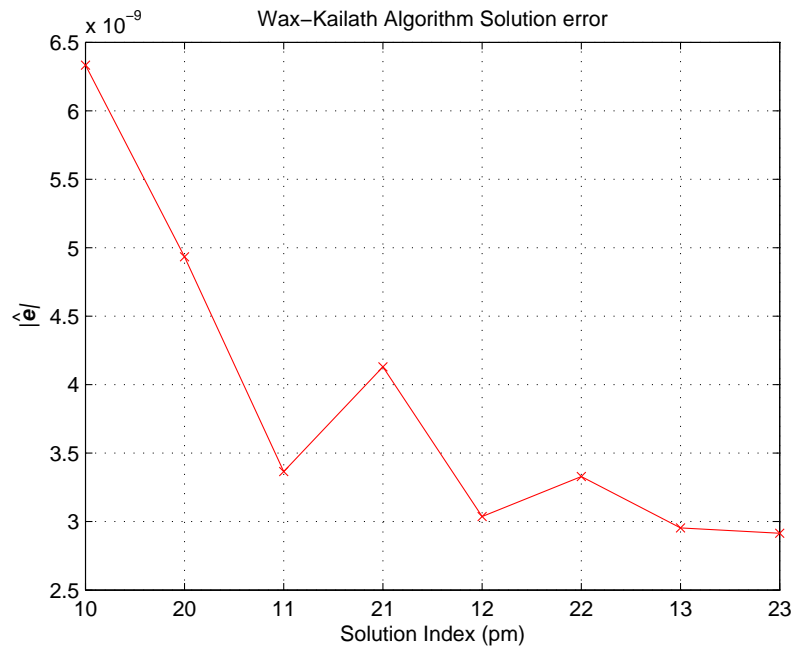


Figure 3.5: Magnitude of Solution Error for Wax-Kailath Algorithm (15 Neumann iterations).

3.4 Summary

In this chapter, we considered practical and efficient approaches to implement IF estimation based on TVAR modeling. Realizing that the covariance matrix obtained using TVAR modeling can be reduced to a block-Toeplitz structure, we proposed the Akaike algorithm to recursively compute its inverse. Compared to Gaussian elimination requiring $O(p^3(q+1)^3)$ computations, the Akaike algorithm requires only $O(p^3(q+1)^2)$ computations. Next, we introduced the Wax-Kailath algorithm to invert a block-Toeplitz matrix with Toeplitz blocks after observing that by adding a small error matrix we could convert our block-Toeplitz matrix to a block-Toeplitz matrix with Hermitian-Toeplitz blocks. The Wax-Kailath algorithm achieved a factor of 2 reduction in the number of recursions as well as the computational effort required to form the inverse matrix from the computed recursions relative to the Akaike algorithm. A scheme to efficiently invert the matrix $\mathbf{\Omega}_i$ required in the Wax-Kailath recursions having the structure in (3.65) was proposed to further reduce the computational effort.

Furthermore, we showed how **Property 2** can be exploited to interchange the dimension of the blocks and that of the block matrix so that significant computational savings can be achieved when $p \gg (q+1)$. For instance, when $p = 5$ and $q = 1$, the Wax-Kailath algorithm without any block dimension interchange requires $O(p^3(q+1)^2) = O(500)$ computations. However, by interchanging the block dimension the computational effort required is reduced to $O((q+1)^3p^2) = O(200)$. Finally, we introduced the Neumann series and showed how to obtain the inverse of the block-Toeplitz matrix from the available error matrix and the inverse of the block-Toeplitz matrix with Hermitian-Toeplitz blocks computed using the Wax-Kailath algorithm. Depending on the number of iterations required in the Neumann series, the overall computational effort in solving the linear system using the Wax-Kailath algorithm can become comparable or more than using the Akaike algorithm. In the next chapter, we look at an application example of IF estimation based on TVAR modeling.

Chapter 4

IF Estimation Application

In Chapter 2, we looked at TVAR based IF estimation for FM components in an uncorrelated (white noise) environment and in Chapter 3 its efficient implementation. In this chapter, we consider the application of IF estimation for FM interference mitigation in a CPM communications system. First, we present the motivation for a decorrelating TVAR (DTVAR) model and then show how it can be used for IF estimation of FM components in a finitely correlated environment. A linear prediction error filter (PEF) formed from the time-varying coefficients of a DTVAR model and a time-varying decision aided compensator are proposed for mitigating FM interference from a CPM signal causing only minor distortion.

4.1 Motivation for Decorrelating TVAR Model

In Chapter 2, we have shown that a TVAR model performs well when applied to short data records for Instantaneous Frequency (IF) estimation of frequency modulated (FM) components in white noise. However, when the model is applied to a signal containing a finitely correlated signal of interest – such as a continuous phase modulated (CPM) signal – in addition to the white noise, estimation performance degrades; especially when the correlated signal is not weak relative to the FM components. We extend time-varying autoregressive (TVAR) model-based IF estimation to a finitely correlated environment by introducing a decorrelation delay larger than one. The decorrelation delay minimizes the correlation

between the current sample of the correlated signal and the samples that are operated on by the filter in order to minimize its effect on the IF estimate of the FM component. The resulting model is referred to as the decorrelating TVAR (DTVAR) model and is presented in detail in Section 4.2.1. To demonstrate its effectiveness, we first apply the TVAR model to a CPM signal by itself. A CPM signal at complex baseband is described as,

$$s_n = e^{j\phi(n;I)} \quad (4.1)$$

Here the instantaneous phase ($\phi(n;I)$) of the CPM signal during the $(m+1)^{\text{st}}$ interval is given by,

$$\phi(n;I) = 2\pi \sum_{k=-\infty}^m I_k h_k G_{n-k\nu}, \quad (m\nu + 1) \leq n \leq (m+1)\nu \quad (4.2)$$

Here I_k denotes the k^{th} CPM symbol effective over ν samples per symbol, with the symbols being chosen from M-ary CPM alphabets $\pm 1, \pm 3, \dots, \pm(M-1)$ and M is restricted to be a power of 2. The constant h_k denotes the k^{th} modulation index, and G_n is called the phase smoothing response. G_n can be any function whose initial value is 0 and final value is 0.5 after some sample time $L\nu$, where L is an integer that determines how many symbols are reflected by G_n . G_n is generally expressed as the integral of some pulse shape. Some commonly used pulse shaping functions are: Raised Cosine (*LRC*), Rectangular (*LREC*), and Gaussian Shaped Minimum Shift Keying (*GMSK*). The parameter L determines the length of the pulse shaping function in terms of an integer number of samples per symbol. In our simulations, we use a *1REC* pulse with $M = 4$ and this CPM signal is referred to as Quaternary full response CPM. As a result, the phase smoothing response G_n is linear during a symbol duration. This implies that the instantaneous phase of the CPM varies linearly and the instantaneous frequency is a constant over a symbol duration.

The CPM signal considered here contains 8 samples per symbol and we use a block of size $N = 8$ for modeling with 50% overlap. The TVAR model parameters used are $p = 1$ and $q = 0$. In Figure 4.1, we compare the actual CPM IF and the IF estimated using TVAR modeling.

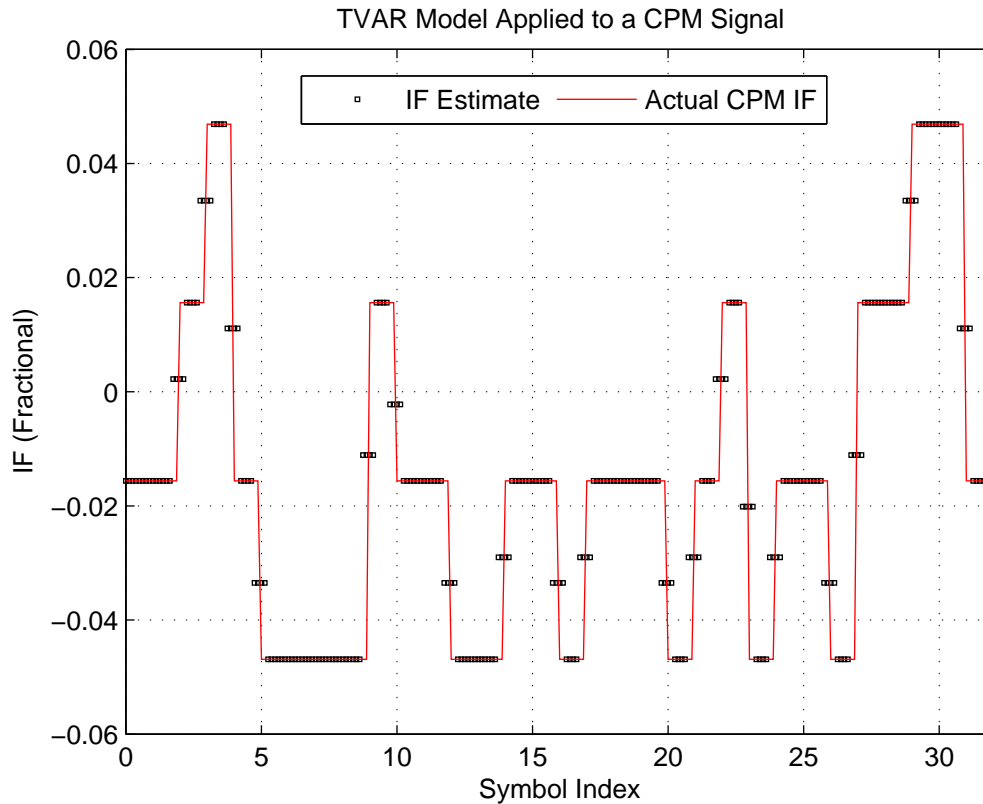


Figure 4.1: CPM IF Estimation using $N = 8$ and $q = 0$.

From Figure 4.1, we see that the TVAR model performs well in estimating the IF of the CPM signal. At the phase transitions, the IF estimates are off because of the block overlapping just a bit more with the next symbol frequency than the previous one. In Figure 4.2, we show the CPM IF estimation result using $N = 128$ and TVAR parameters $p = 1$ and $q = 1$.

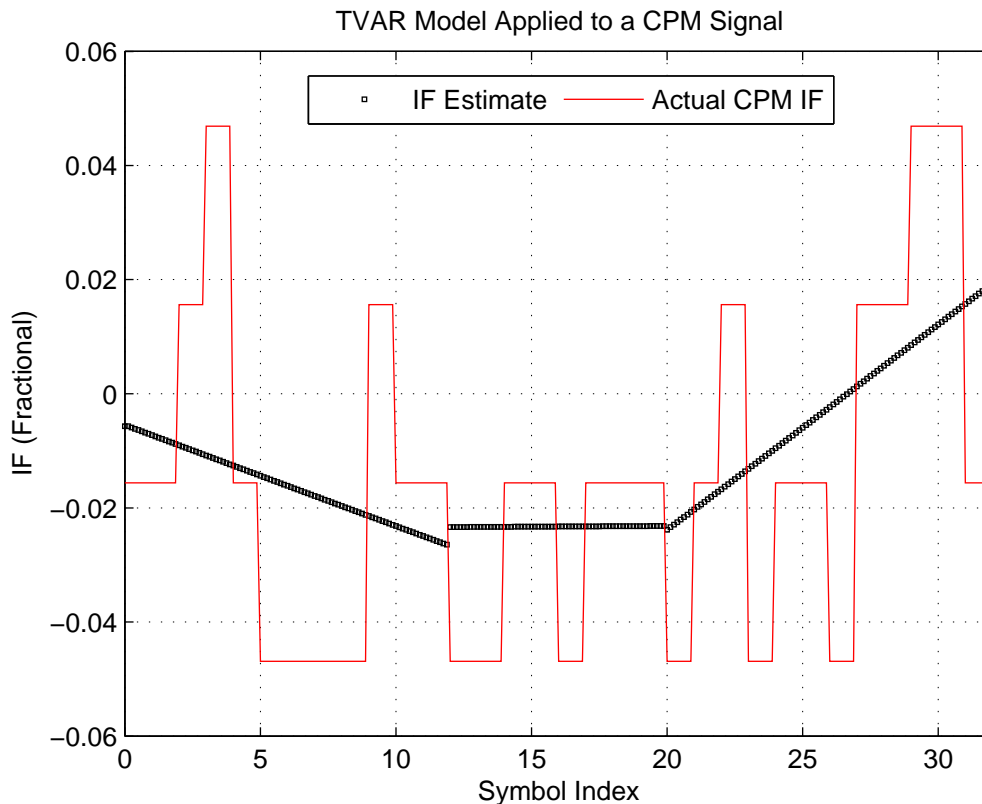


Figure 4.2: CPM IF Estimation using $N = 128$ and $q = 1$.

From Figure 4.2, we see that using a larger block size and an additional basis function – allowing a linear variation of the TVAR parameters – the IF estimation attempts to provide an average linear fit to the actual CPM IF. Note that here the information about the individual CPM symbols is lost. If the block is made even longer, the average representation will approach the average IF of zero for the CPM signal.

Next, we apply the TVAR model to a signal containing a single FM component (treated as an interference) and the CPM signal with the exact same IF as shown in Figures 4.1 and 4.2. The SIR used is -5 dB and therefore the CPM signal is not too weak compared to the FM interference. The TVAR parameters used are $p = 1$ and $q = 1$. Modeling is applied for blocks of size $N = 128$ with 50% overlap. In Figure 4.3, we show the IF estimation result. For comparison, the IF estimation result based on a DTVAR model is also shown. A zoomed-in view of Figure 4.3 is shown in Figure 4.4.

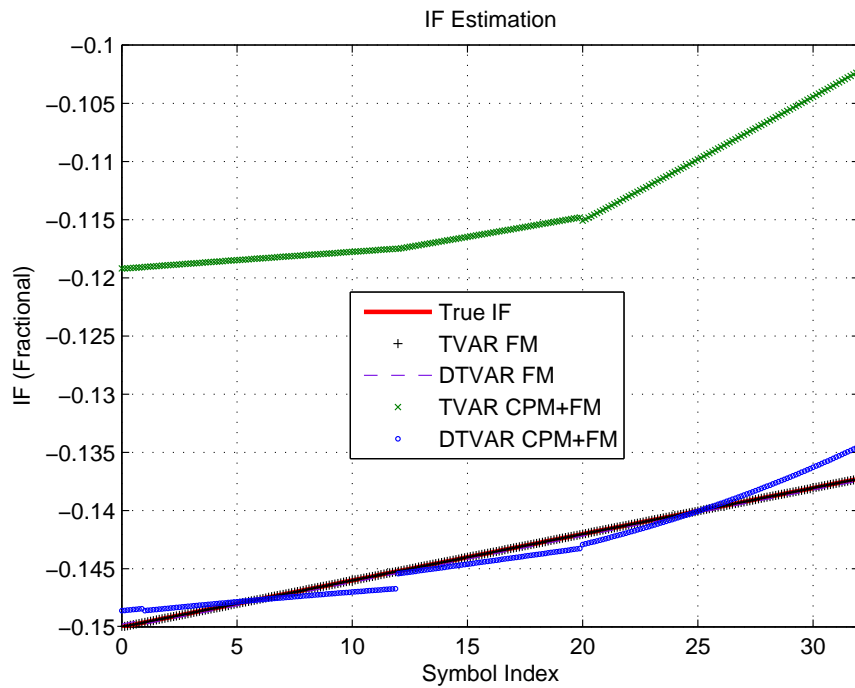


Figure 4.3: TVAR and DTVAR Model Based IF Estimation Comparison.

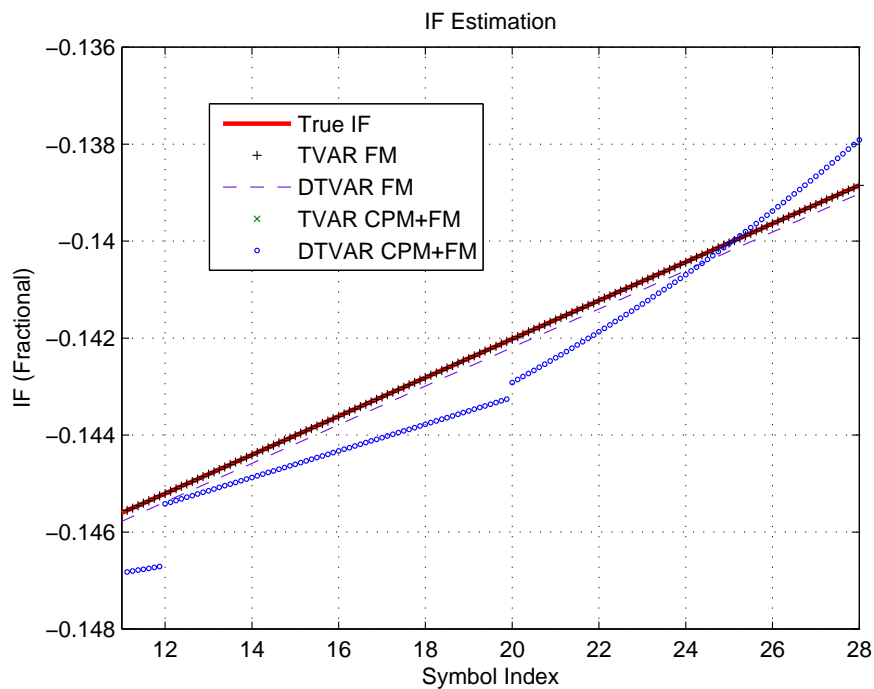


Figure 4.4: TVAR and DTVAR Model Based IF Estimation Comparison: Zoomed-In.

From Figure 4.3 and its zoomed-in view in Figure 4.4, we see that the TVAR model based IF estimate of the FM component (TVAR CPM+FM, green x) deviates significantly from the true IF of the FM component. Also, we note that instead of a single linear estimate, the estimated IF is piecewise linear with three pieces. In fact, comparing Figure 4.3 to Figure 4.2, we observe that the transitions from one linear piece to the next in the TVAR IF estimate are consistent with the transitions in the CPM IF estimate linear fit. From Figure 4.3, we see that the DTVAR performs better than TVAR in estimating the FM component IF. However, the DTVAR estimate also has a piecewise behavior and we see that it is not exactly linear all the time. However, we have used $q = 1$ for the DTVAR model and expect the DTVAR parameters to change linearly. A plot of the DTVAR parameters is shown in Figure 4.5.

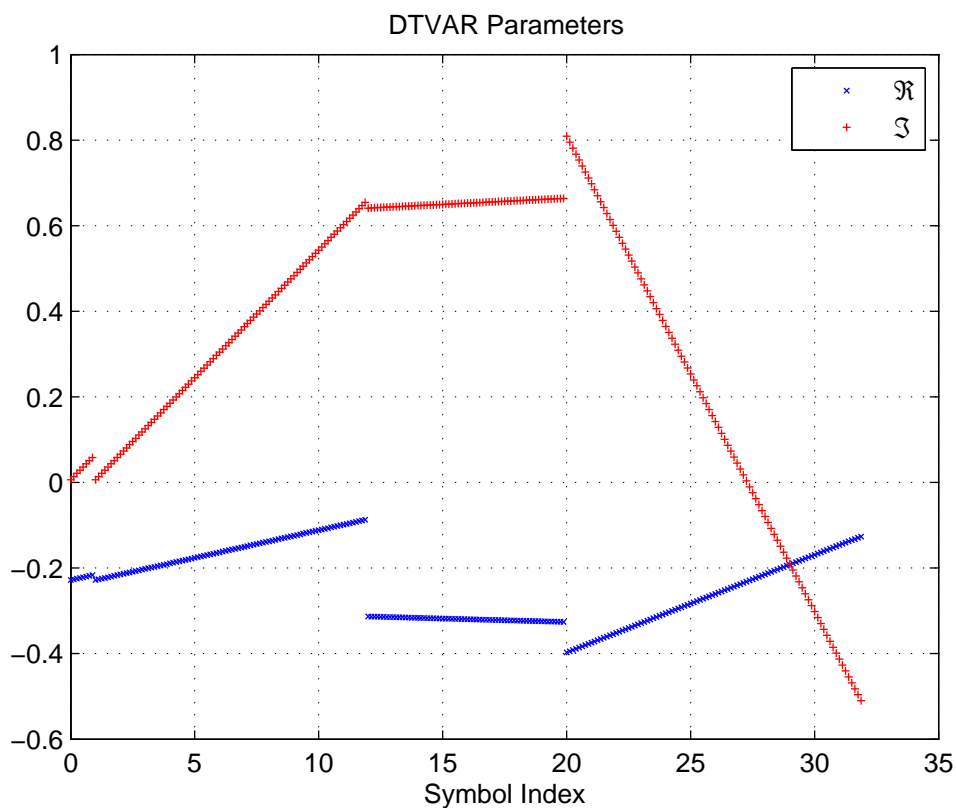


Figure 4.5: DTVAR Parameters for $q = 1$.

From Figure 4.5, we see that the parameters vary only linearly as expected. However, the DTVAR IF estimates are derived from the angles of the poles of an associated prediction error filter polynomial (to be introduced later) and it is no longer linear. Also, from Figure 4.4, we see that when the TVAR model is applied to the interference alone, the IF estimate is very good as expected. The corresponding DTVAR estimate is also very good but off by a small factor. This offset is a result of using a decorrelation delay greater than 1. To illustrate this, consider a complex sinusoidal (cisoid) signal with a single linearly time-varying frequency modulated (FM) component as shown below,

$$x_n = e^{j2\pi(f_0 + \alpha n)n} \quad (4.3)$$

Here, the instantaneous frequency of the signal obtained by differentiating the angle is given by,

$$f_{i,n} = f_0 + 2\alpha n \quad (4.4)$$

where f_0 is the initial frequency and α is the rate of change of frequency. It can be shown by substitution that the cisoid in (4.3) satisfies the following recursion,

$$x_n + a_{1,n}x_{n-\Omega} = 0 \quad (4.5)$$

Here, Ω is the decorrelation delay. The time-varying parameter $a_{1,n}$ is found to be,

$$a_{1,n} = -e^{j2\pi(f_0\Omega + 2\alpha\Omega n - \alpha\Omega^2)} \quad (4.6)$$

If we consider the time-varying polynomial $A(z; n) = 1 + a_{1,n}z^{-\Omega}$, we can show that for (4.6) it has one root on the unit circle at an angle $2\pi(f_0 + 2\alpha n - \alpha\Omega)$ for a given time instant n . Therefore, the instantaneous frequency estimate $\tilde{f}_{i,n}$ obtained by dividing the angle of the root at each time instant by 2π is off from the true IF shown in (4.4) by the factor $\alpha\Omega$. This difference can be significant if α or Ω is large.

In conclusion, the performance degradation seen in the TVAR IF estimate when applied to an FM component in the presence of a finitely correlated CPM signal is due to the IF of the CPM signal. Furthermore, in Section 4.2.3, we show through simulations that, as the FM

interference becomes stronger, the IF estimation performance of both TVAR and DTVAR modeling approaches improves with the DTVAR providing better results for moderate to high SIRs. One approach to improving the TVAR IF estimation result is to increase the TVAR order p from 1 to 2 and let the extra pole capture the CPM signal and noise. However, now 2 IF estimates are available for each sample and we are faced with the non-trivial problem of identifying the IF estimate corresponding to the FM component. This requires additional computational burden in implementing association algorithms. Even when we use $p > 1$, a decorrelating TVAR model is found to provide a slightly better result than the TVAR model. Our focus will be on using TVAR order $p = 1$ for estimating the IF of a single FM component in a finitely correlated environment. The DTVAR model and IF estimation based on it to accomplish this is introduced next.

4.2 Decorrelating TVAR Based IF Estimation

The time-varying parameters of the TVAR model derived with a decorrelation delay form a generalized time-varying linear prediction error filter (TVLPEF) in which the delay between the predictor weights can be more than one. Furthermore, the angles of the roots of the generalized PEF weights at each sample instant can be used to obtain an improved IF estimate at moderate to high SIRs. Comparison of the decorrelating TVAR based IF estimator to a conventional TVAR based IF estimator reveals performance gains at moderate to high signal to FM component power ratios.

4.2.1 Decorrelating TVAR Model

Here we outline the modifications to the TVAR modeling introduced in Section 2.2 to obtain time-varying coefficients of a decorrelating TVAR model. A p^{th} order DTVAR representation of the nonstationary discrete-time stochastic process x_n with decorrelation delay (Ω) between the time-varying coefficients is shown below,

$$x_n = -\sum_{k=1}^p a_{k,n} x_{n-k\Omega} + v_n \quad (4.7)$$

Here v_n is a stationary zero mean white noise process of variance σ_v^2 and the time-varying coefficients $a_{k,n}$ are modeled as a linear combination of a set of time basis functions. We can express the DTVAR coefficients in general as,

$$a_{k,n} = \sum_{m=0}^q a_{km} u_{m,n} \quad (4.8)$$

The basis functions ($u_{m,n}$) are powers of the time variable n as shown below,

$$u_{m,n} \triangleq \left(\frac{n - p\Omega}{N} \right)^m \quad (4.9)$$

The parameters a_{km} are estimated by minimizing the total squared prediction error, which is also the error in modeling x_n , as shown below,

$$\epsilon_p = \sum_{n=p\Omega}^{N-1} \underbrace{\left| x_n + \sum_{k=1}^p \sum_{m=0}^q a_{km} u_{m,n} x_{n-k\Omega} \right|}_{|v_n|}^2 \quad (4.10)$$

The generalized covariance function $c_{mg}(l, k)$ for $\{l = 1, 2, \dots, p; g = 0, 1, \dots, q\}$ is defined as,

$$c_{mg}(l, k) \triangleq \sum_{n=p\Omega}^{N-1} u_{m,n} x_{n-k\Omega} u_{g,n}^* x_{n-l\Omega}^* \quad (4.11)$$

The parameters a_{km} that minimize (4.10) are obtained by solving the following generalized covariance equations,

$$\sum_{k=1}^p \sum_{m=0}^q a_{km} c_{mg}(l, k) = -c_{0g}(l, 0) \quad (4.12)$$

The above equation represents a system of $p(q+1)$ linear equations and can be represented in a matrix form to obtain a covariance matrix $\mathbf{C}_{p(q+1) \times p(q+1)}$ that has a block-Hankel structure as obtained for a TVAR model. The introduction of the decorrelating delay does not change the structure of the covariance matrix and therefore the implementation algorithms given in Chapter 3 can be applied to efficiently solve the linear system of equations and compute the DTVAR coefficients.

4.2.2 IF Estimation in a Finitely Correlated Environment

In Section 2.3, we showed IF estimation based on the coefficients of a conventional TVAR model. In this section, we show how the IF estimates can be derived from a DTVAR model. The time-varying filter $H(z; n)$ corresponding to the DTVAR model can be expressed as,

$$H(z; n) = \frac{1}{A(z; n)} = \frac{1}{1 + \sum_{k=1}^p a_{k,n} z^{-k\Omega}} \quad (4.13)$$

The weights of the corresponding time-varying PEF at a given sample instant are given by,

$$\mathbf{a}_n = \left[1 \underbrace{0 \ 0 \ \cdots \ 0}_{\Omega-1} \ a_{1,n} \ \underbrace{0 \ 0 \ \cdots \ 0}_{\Omega-1} \ a_{2,n} \ \cdots \ a_{p,n} \right] \quad (4.14)$$

If there are p FM components in the modeled signal, the time-varying PEF frequency response at a given sample instant has $p\Omega$ notches with Ω periodic notches of period $1/\Omega$ for each IF component. Also, rooting the PEF polynomial corresponding to \mathbf{a}_n in (4.14) and taking the angles of the poles at each sample instant n , we can obtain the $p\Omega$ notch frequencies. Out of these, p notch frequencies are estimates of the IF of the p FM components. Therefore, we are now faced with the non-trivial problem of identifying the correct p IF estimates from a set of $p\Omega$ frequencies and classification algorithms need to be used to obtain the correct IF explicitly.

We propose a two step approach to obtain the correct IF. In the first step, we use a conventional TVAR model with a decorrelation delay $\Omega = 1$ to obtain p rough IF estimates $\hat{f}_{i,n}$ ($i = 1, 2, \dots, p$). In a finitely correlated environment, these estimates are generally poor but still relatively close to the true IF $f_{i,n}$. Next, we use the TVAR model with decorrelation delay $\Omega > 1$ to obtain $p\Omega$ frequency estimates of which p are expected to be better IF estimates than $\hat{f}_{i,n}$. From the set of $p\Omega$ frequencies, the ones that are closest to $\hat{f}_{i,n}$ for $i = 1, 2, \dots, p$ are chosen as the better IF estimates $\tilde{f}_{i,n}$.

4.2.3 Simulation Results

To illustrate the performance of the DTVAR based IF estimator, we compare it with a conventional TVAR based IF estimator. We consider a received signal r_n consisting of a CPM signal s_n having finite correlation, white noise v_n , and FM interference i_n as shown below,

$$r_n = s_n + v_n + i_n \quad (4.15)$$

CPM signal parameters we use for the simulation are given below,

$$\begin{aligned} \text{Symbol Rate } (R_s) &= 4800 \text{ symbols/sec}, \nu = 8, h = 0.25 \\ \text{Sampling Frequency } (f_s) &= \nu R_s = 38400 \text{ samples/sec} \end{aligned} \quad (4.16)$$

Here, ν is the number of samples per symbol and h is the modulation index. Simulations are performed at a signal-to-noise ratio (SNR) of 20 dB and moderate to high SIRs. IF estimation of a single component linear FM chirp interference and a non-linear FM interference are considered at complex baseband. We have used 6400 symbols equivalent to a CPM burst of length $D = 51200$ samples to perform the simulations. The reciprocal of the mean squared error (MSE) in the IF estimates as defined below is used as the performance measure,

$$\text{MSE}_i = \frac{1}{D} \sum_{n=0}^{D-1} (|f_{i,n} - \tilde{f}_{i,n}|^2) \quad (4.17)$$

Linear FM Interference

The interference signal i_n we use here is a single component linear FM that chirps in frequency from -4800 Hz to 4800 Hz over the entire length of the CPM signal burst. The linear IF law is given by,

$$f_n = f_0 + 2\alpha n = f_0 + 2 \frac{(f_e - f_0)}{2(D-1)} n \quad (4.18)$$

The FM signal generated is given below,

$$i_n = e^{j2\pi[(f_0 + \alpha n)n + \phi]} \quad (4.19)$$

Here, f_0 is the initial frequency, f_e is the end frequency, α is the rate of change of frequency, and ϕ is a random initial phase. In our example, $f_0 = -4800$ Hz, $f_e = 4800$ Hz, and $\alpha = \frac{|f_0|}{D-1} = 0.0938$ Hz/sample. The DTVAR model is applied to blocks of size $N = 128$ and in order to avoid the end effects in modeling we overlap the blocks by 50% and use the coefficient estimates for the middle of a block. The model parameters used are $p = 1$ and $q = 1$. Also, we use the covariance method of signal modeling and therefore $\tau = [p\Omega, \dots, N - 1]$.

The DTVAR parameters $a_{k,n}$ are derived as shown in Section 4.2.1 and the IF estimates are obtained using the two step procedure given in Section 4.2.2. In Figure 4.6, we show the true and estimated IF with and without a decorrelation delay at a SIR of -5 dB.

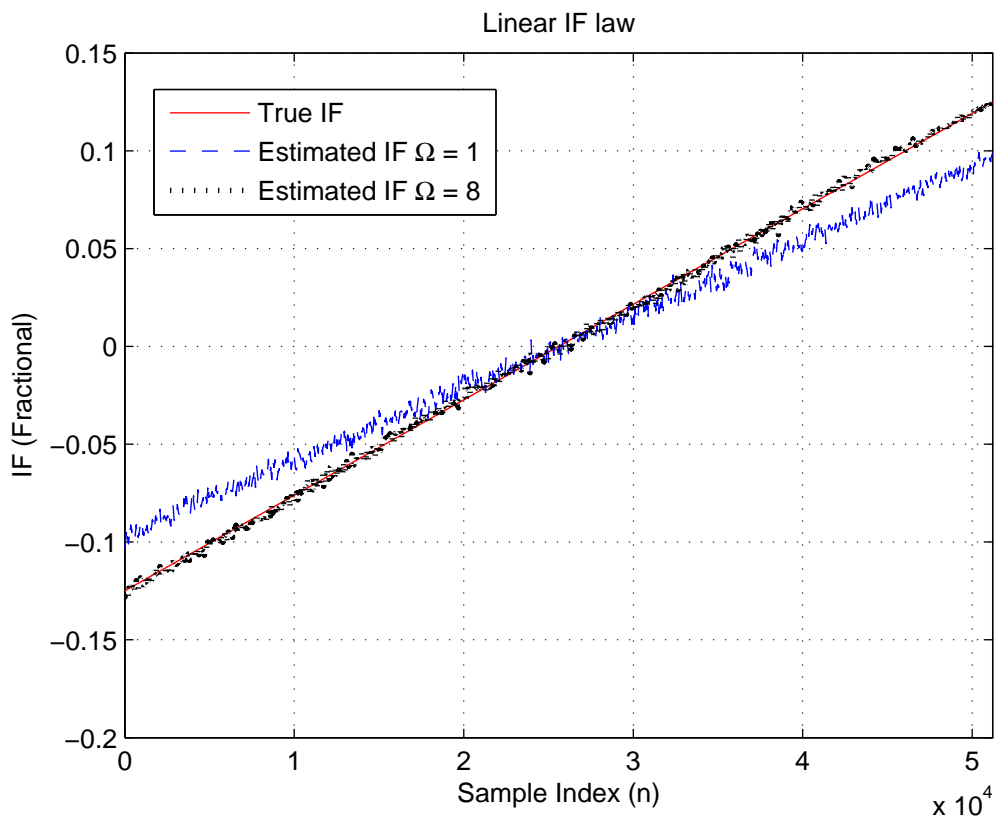


Figure 4.6: Linear IF law: True (solid, red), TVAR (Dashed, blue), and Decorrelating TVAR (dots, black).

From Figure 4.6, we see that the TVAR based IF estimates ($\Omega = 1$) deviate significantly from the true IF whereas the DTVAR based IF estimates ($\Omega = 8$) follow the true IF much more closely. In Figure 4.7, we show the reciprocal of the MSE plotted against SIR, for IF estimation with and without use of a decorrelation delay. The decorrelation delay we used here is $\Omega = 8$. For each SIR, the MSE is calculated using 51200 IF estimates.

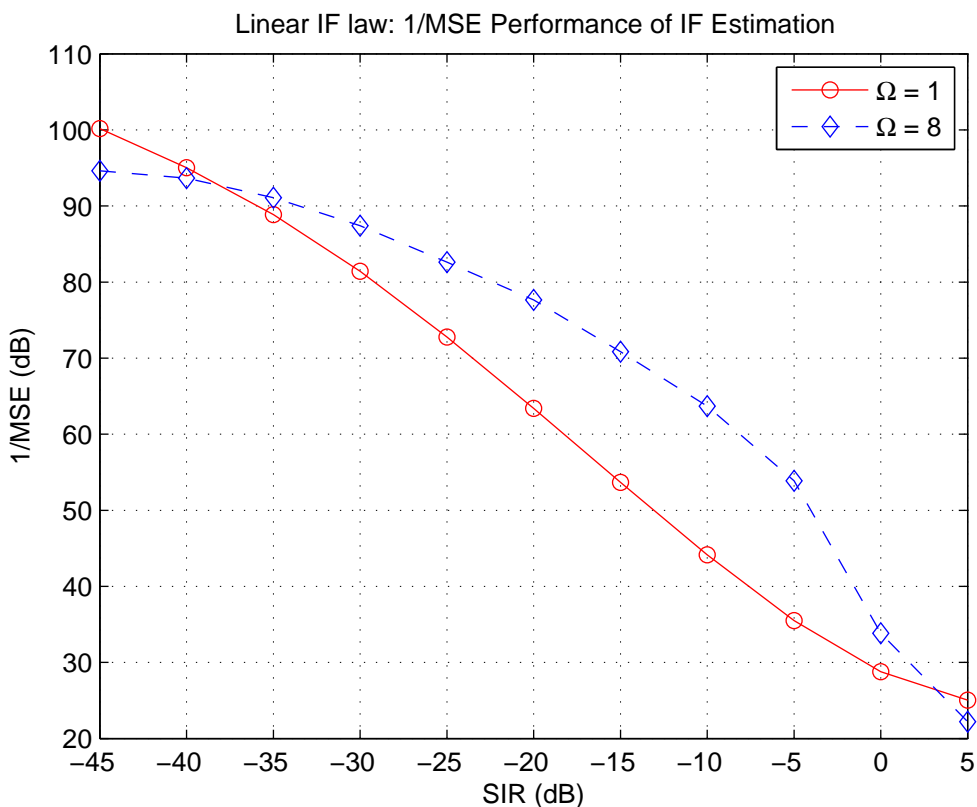


Figure 4.7: 1/MSE vs. SIR for IF Estimation of a linear FM signal: TVAR (solid, red) and Decorrelating TVAR (dashed, blue).

From Figure 4.7, we see that DTVAR IF estimation yields notable performance gains for SIR between -25 dB and -5 dB. When the SIR is more than 0 dB, the power of the correlated signal dominates and IF estimation results for either of the approaches deteriorates. When the SIR falls below -35 dB, the 1/MSE for either of the approaches is high, with the TVAR based IF estimator providing a somewhat better result.

Non-linear FM Interference

Now we show simulation results for a non-linear IF law. We consider a signal having a single FM component with a sinusoidally varying frequency. The DTVAR model parameters used are $p = 1$, $q = 1$, and $\Omega = 8$. The non-linear IF law used in this simulation is given by,

$$f_n = -0.01 + 0.004(0.0009n\cos(0.0009n) + \sin(0.0009n)) \quad (4.20)$$

The FM signal generated is given below,

$$i_n = e^{j2\pi[(-0.01+0.004\sin(0.0009n))n+\phi]} \quad (4.21)$$

In Figure 4.8, we show the true and estimated IF with and without a decorrelation delay at a SIR of -5 dB.

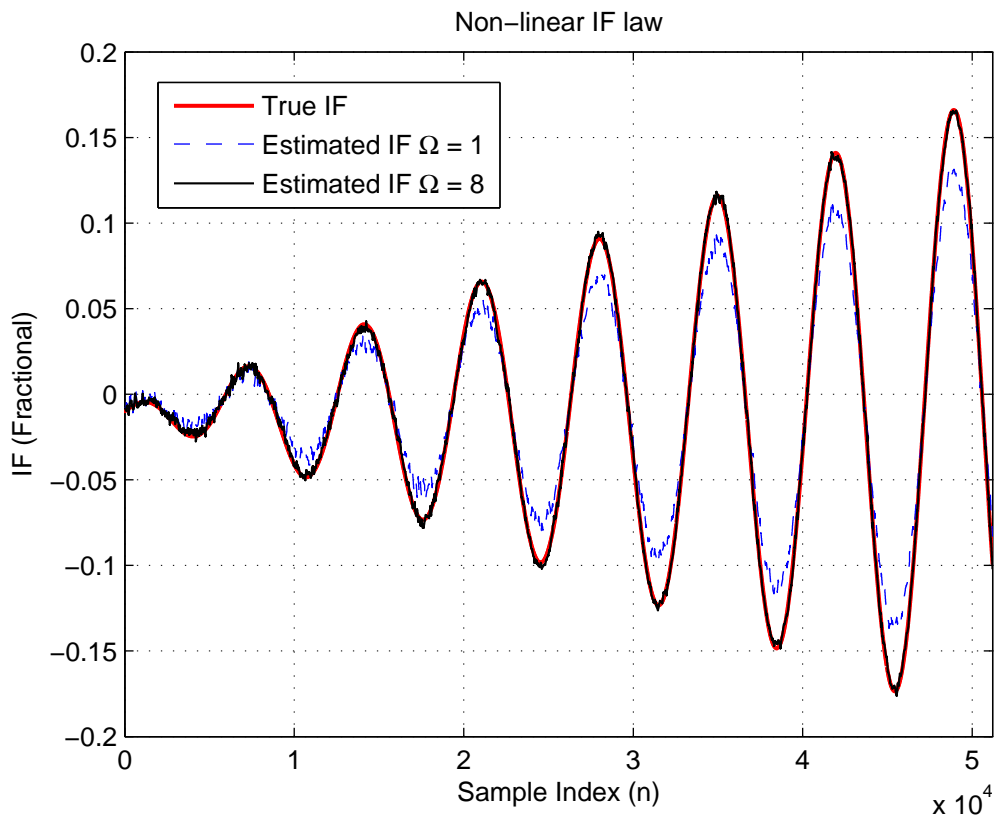


Figure 4.8: Non-linear IF law: Decorrelating TVAR based IF Estimation .

From Figure 4.8, we see that the DTVAR IF estimates follow the true IF much more closely than the conventional TVAR based IF estimates. The reciprocal of the MSE plotted against SIR is shown in Figure 4.9 for both of the approaches.

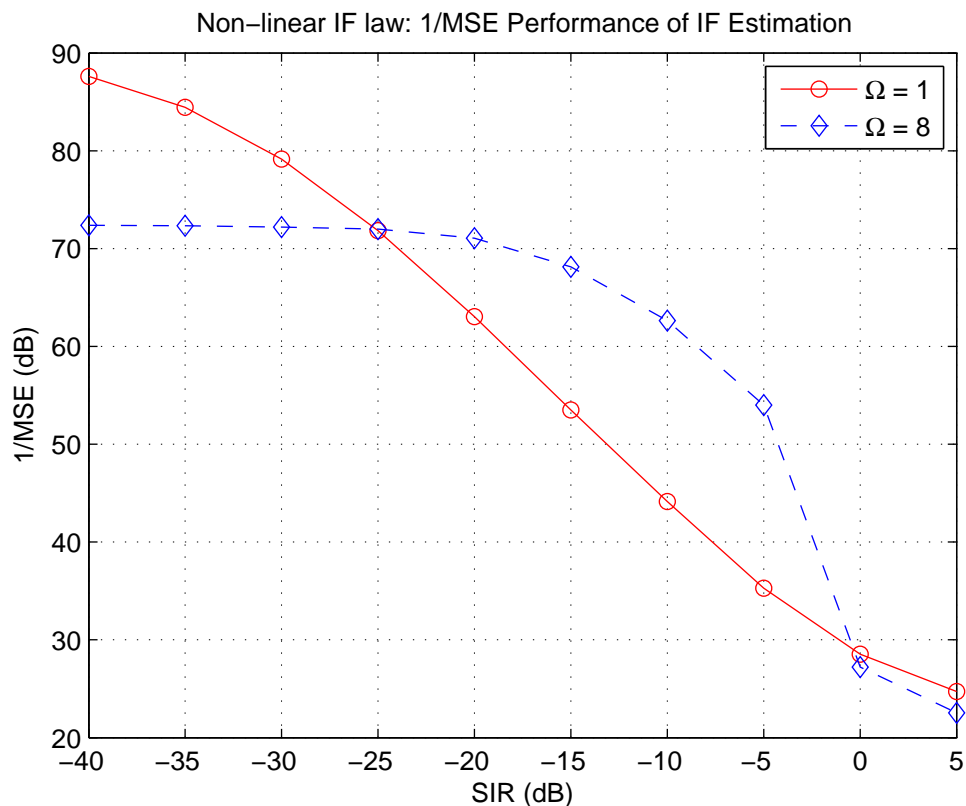


Figure 4.9: 1/MSE vs. SIR for IF Estimation of a Non-linear FM signal: TVAR (solid) and Decorrelating TVAR (dashed).

In Figure 4.9, notable performance gains are observed for the DTVAR for SIR between -20 dB and -5 dB. Although the DTVAR based IF estimator provides superior results at moderate to high SIRs, when the SIR falls below a threshold, the TVAR based approach starts providing better IF estimates than the DTVAR based IF estimator.

4.3 FM Interference Mitigation

A linear prediction error filter (PEF) formed from the coefficients of a time-varying autoregressive (TVAR) model can be used effectively for mitigating frequency modulated

(FM) components in a minimally correlated desired signal and white noise. However, when the desired signal is a finitely correlated signal – such as a continuous phase modulated (CPM) signal – the linear PEF formed from the TVAR coefficients results in mitigation of the desired signal along with the interference. We propose a linear PEF formed from the coefficients of a decorrelating TVAR (DTVAR) model to address this issue. A time-varying decision aided compensator (TVDAC) is introduced to compensate for the distortion in the desired signal produced by the proposed PEF. Comparison of a time-varying linear PEF (TVLPEF) formed from DTVAR and TVAR coefficients for interference mitigation in a CPM communications system reveals notable performance gains at most moderate to high signal-to-interference ratios (SIRs).

4.3.1 Time-Varying PEF and Compensator

In this section, we present a generalized TVLPEF formed from the time-varying coefficients $a_{k,n}$ of the DTVAR model for interference mitigation in a finitely correlated environment. To compensate for the PEF distortion, we introduce a time-varying decision aided compensator (TVDAC). We consider a received signal r_n consisting of a signal s_n having finite correlation, white noise v_n , and a FM interference i_n as shown in (4.15). The DTVAR model is applied to blocks of the received signal r_n and parameters $a_{k,n}$ are derived as shown in Section 4.2.1. Once the DTVAR coefficients are obtained, we form a time-varying PEF that subtracts the prediction \hat{i}_n of the interference from the received signal r_n . The block diagram of the TVLPEF is shown in Figure 4.10.

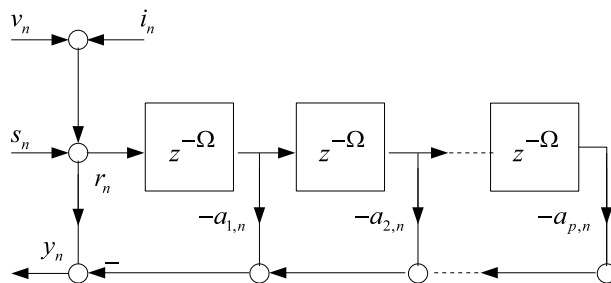


Figure 4.10: Block Diagram of a TVLPEF.

The output of the PEF is given by,

$$y_n = r_n + \sum_{k=1}^p a_{k,n} r_{n-k\Omega} \quad (4.22)$$

Substituting for r_n from (4.15) in (4.22) we have,

$$y_n = s_n + i_n + v'_n + \sum_{k=1}^p a_{k,n} (s_{n-k\Omega} + i_{n-k\Omega}) \quad (4.23)$$

Here, $v'_n = v_n + \sum_{k=1}^p a_{k,n} v_{n-k\Omega}$ is a filtered version of white noise v_n . When the decorrelation delay $\Omega = 1$, the parameters $a_{k,n}$ are of a conventional TVAR model and finite correlation in the desired signal s_n results in the approximate cancelation of s_n as well as i_n from the output y_n .

However, if we use a DTVAR model and set $\Omega > 1$, the predictability of the current sample of s_n from the samples operated on by the PEF is reduced so that it is not annihilated at the PEF output. Assuming the approximate cancelation of the interference, we have,

$$y_n \approx s_n + v'_n + \sum_{k=1}^p a_{k,n} s_{n-k\Omega} \quad (4.24)$$

While we are able to prevent the cancelation of s_n from the PEF output, the term $\sum_{k=1}^p a_{k,n} s_{n-k\Omega}$ appears as an inter symbol interference (ISI) and distorts the desired signal in the observed output y_n . A straight-forward approach to compensating for this distortion is to subtract the distortion from y_n as shown below,

$$z_n = y_n - \sum_{k=1}^p a_{k,n} s_{n-k\Omega} \approx s_n + v'_n \quad (4.25)$$

However, this approach requires past samples of the transmitted communication signal s_n to be known *a priori*. Nevertheless, assuming that the first $p\Omega$ samples of the signal s_n are available as a training sequence, we can compute z_n for $n > p\Omega$. When the $p\Omega$ training samples expire, we can replace $s_{n-k\Omega}$ in (4.25) with symbol decisions $\tilde{s}_{n-k\Omega}$ obtained by passing z_n for $n > p\Omega$ into a decision device. Assuming that the decision device produces correct symbol decisions ($\tilde{s}_n = s_n$), we can then maintain successful compensation of the distortion caused by the TVLPEF. A block diagram of this time-varying decision aided compensator (TVDAC) is shown in Figure 4.11.

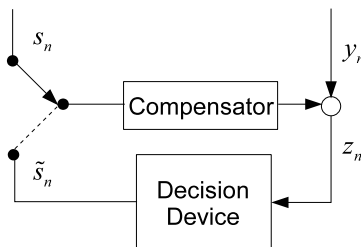


Figure 4.11: Block Diagram of a TVDAC.

4.3.2 Simulation Results

In this section, we consider time-varying narrowband FM interference mitigation in a CPM communications system to illustrate the workings and the desirable performance of a decorrelating TVAR model-based TVLPEF over the one based on a conventional TVAR model. The communication signal s_n we use is a finitely correlated quaternary full response CPM signal at complex baseband with the parameters given in (4.16). In the compensation approach shown in Figure 4.11, symbol decisions must be available immediately after samples of z_n are fed into the decision device. However, the optimum decision device for a CPM communication system is a Viterbi Decoder which looks at symbols that come after to determine what happened some time back - also called the trace back length. Therefore, symbol decisions are not available immediately. To overcome this, we use a decoder with a trace back length of 1 that is commonly used for acquisition and tracking of CPM signals [45]. We implement this decoder - referred to here as the one symbol decoder (OSD) - assuming ν and h are known at the receiver and that synchronization is perfect. A OSD is not robust and its symbol error rate (SER) performance deteriorates at low signal-to-noise ratios (SNRs). To obtain a more realistic SER for communication purposes, the compensated signal z_n is decoded using a Viterbi Decoder with larger trace back length.

We also note that a CPM symbol lasts multiple samples and for the decision device to provide a symbol decision, ν samples of z_n should be provided. Therefore, to make the compensation feasible, we set the decorrelation delay $\Omega \geq \nu$. The decorrelation delay we use in the example is $\Omega = 8$. The interference signal i_n used is a single component linear FM that chirps in frequency from -1920 Hz to 1920 Hz over the entire length of the CPM signal

burst of 1024 symbols. The linear IF law is given by,

$$f_n = f_0 + 2\alpha n = f_0 + 2\frac{(f_e - f_0)}{2(D - 1)}n \quad (4.26)$$

The FM signal generated is given below,

$$i_n = e^{j2\pi[(f_0 + \alpha n)n + \phi]} \quad (4.27)$$

Here, f_0 is the initial frequency, f_e is the end frequency, α is the rate of change of frequency, and ϕ is a random initial phase. D is the length of the CPM burst equal to 8192 samples. In our example, $f_0 = -1920$ Hz, $f_e = 1920$ Hz, and $\alpha = 0.2344$ Hz/sample. The TVAR model is applied to blocks of size $N = 128$; to avoid the end effects in modeling we overlap the blocks by 50% and use the coefficient estimates for the middle of a block for mitigation. The model parameters used are $p = 1$ and $q = 1$. The DTVAR parameters $a_{k,n}$ are derived as shown in Section 4.2.1 and the TVLPEF and TVDAC are formed as shown in Section 4.3.1. The performance measure used is symbol error rate (SER) computed for 512000 CPM symbols by performing 500 trials of 1024 symbols. In Figure 4.12, we show a section of the transmitted CPM signal s_n , and the output of the TVLPEFs using TVAR ($\Omega = 1$) and DTVAR ($\Omega = 8$) models. The signal z_n after compensation using the TVDAC approach is also shown. The simulations are performed at SNR of 20 dB and SIR of -30 dB.

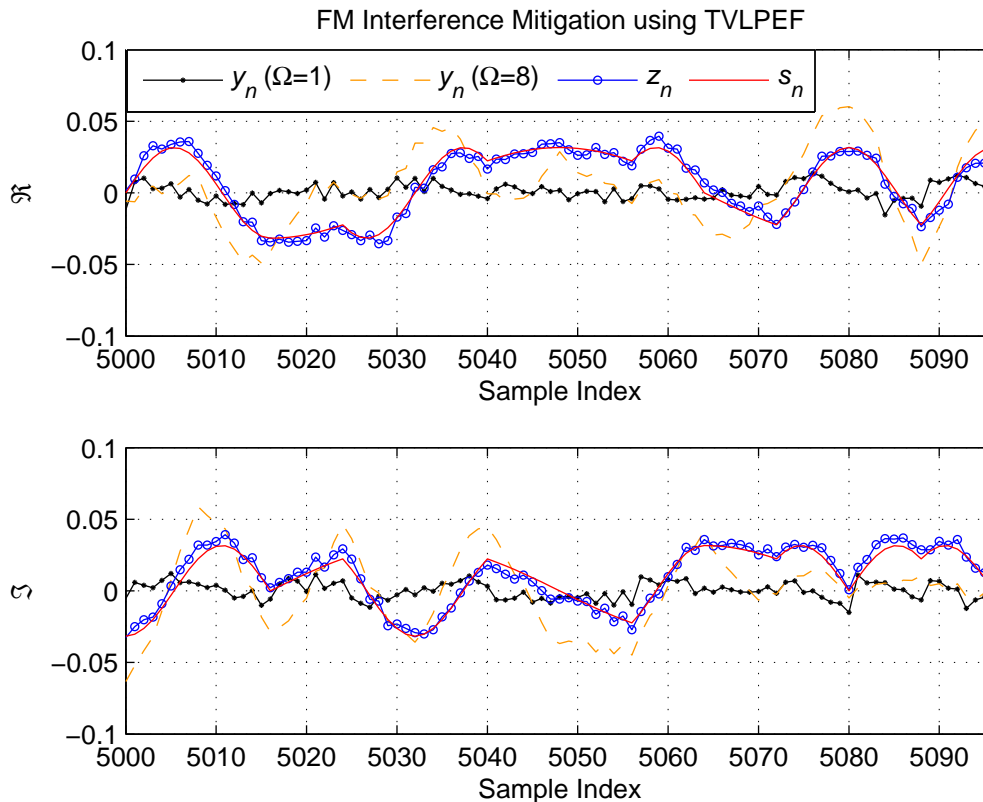


Figure 4.12: FM Interference Mitigation in a CPM Signal.

From Figure 4.12, we see that the TVLPEF output with $\Omega = 1$ is heavily distorted (no visible symbol information left). Using a decorrelation delay of $\Omega = 8$ reduces the excessive cancelation of the CPM signal although it results in ISI. After compensation using TVDAC based on a OSD, the signal z_n follows the transmitted signal closely.

In Figure 4.13, for SNR of 20 dB, we show the SER performance obtained by decoding y_n obtained for $\Omega = 1$ and decoding the compensated signal z_n obtained for $\Omega = 8$, both using a Viterbi decoder with trace back length of 10, as well as the SER of the OSD used in TVDAC. As a reference, the SER of decoding z_n obtained by ideal compensation, i.e. assuming perfect symbol decisions ($\tilde{s}_n = s_n$), is also shown.

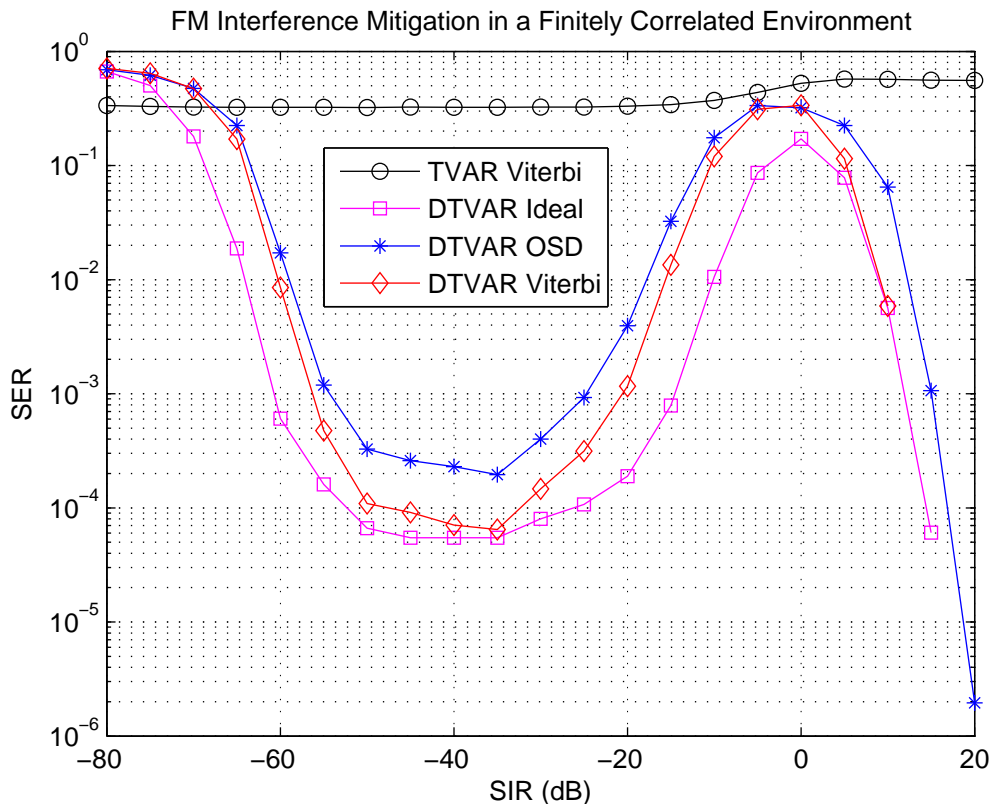


Figure 4.13: SER vs. SIR for FM Interference Suppression using TVAR Model and DTVAR Model at SNR=20 dB.

From Figure 4.13, we see that the DTVAR model-based TVLPEF for interference mitigation outperforms one based on a conventional TVAR model for the entire range of SIR considered. Notable performance gains are achieved at moderate to high SIR with the exception of SIRs from -5 dB to 5 dB. For very low SIRs, the prediction of the interference is good; however, the prediction error is large relative to the level of the desired signal s_n which explains the SER performance deterioration. As the SIR becomes -30 dB and more, we find that the SER performance starts deteriorating although it is much better than TVAR until a SIR of -5 dB. We attribute this behavior to the increased error in predicting the interference by the TVLPEF. The increasing prediction error increases the approximation error in equations (4.24) and (4.25) resulting in the compensation being less effective and correspondingly increased symbol errors. To illustrate this behavior, we show the plot of the absolute value

of the prediction error $\xi = i_n - \hat{i}_n$ at SIR of -35 dB, -15 dB, and -5 dB respectively in Figure 4.14.

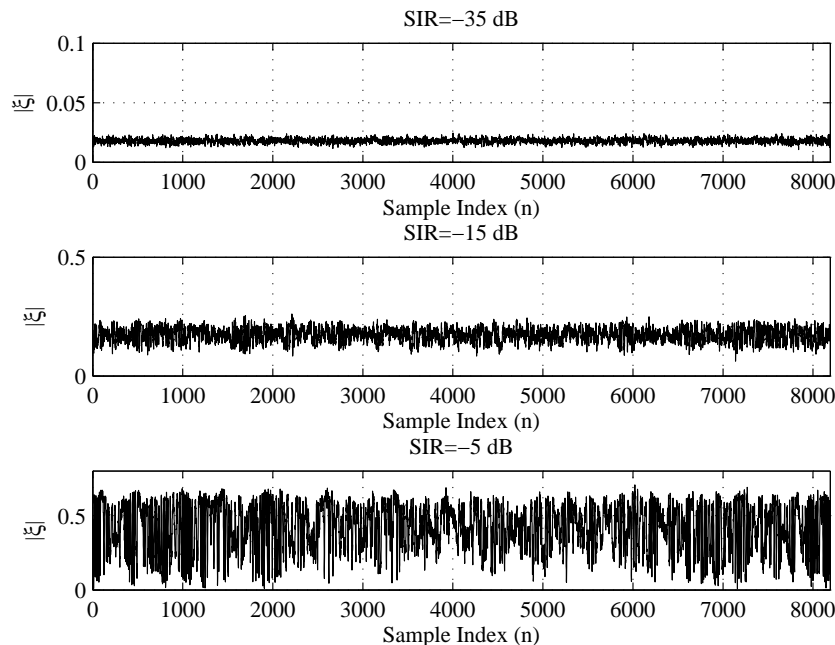


Figure 4.14: Absolute value of Interference Prediction Error ($|\xi|$).

From Figure 4.14, we see that the prediction error increases as SIR becomes higher and distorts the expected desired signal s_n in the PEF output y_n . Even using perfect symbol decisions in the proposed TVDAC does not help to achieve a notable SER performance gain. The OSD robustness is significantly affected by distortion in the PEF output and hence its performance deteriorates. Since the Viterbi decoder uses z_n obtained by OSD based compensation, it provides little to no improvement. As the SIR becomes larger than 5 dB, the signal starts dominating and the SER performance starts to improve for the DTVAR approach. However, the TVAR model starts predicting more of the finitely correlated CPM signal resulting in its cancellation from the PEF output. On the other hand, the decorrelating TVAR model keeps the predictability of the finitely correlated signal low and outperforms the TVAR in SER performance. The flatness of the SER curve for the TVAR model is due to the excessive cancellation of the finitely correlated CPM signal. Although the OSD SER

performance is good, the Viterbi decoder provides better results.

4.4 Summary

In this chapter, we considered the application of IF estimation in mitigating FM interference from a finitely correlated CPM signal. First we presented IF estimation in a finitely correlated environment. The TVAR based IF estimator is acceptably good and performs well when used in an uncorrelated environment. However, its performance degrades when TVAR modeling is applied to a signal containing a finitely correlated signal in addition to the white noise. The performance degradation worsens as the power of the correlated signal starts dominating. The proposed decorrelating TVAR model mitigates this performance degradation and helps restore the IF estimate to acceptable levels. Simulations for a linear and for a non-linear FM signal – containing a finitely correlated CPM signal – show that IF estimation using a decorrelating TVAR model outperforms IF estimation using the conventional TVAR model when SIR is moderate to high.

Next, we looked at FM interference mitigation in a finitely correlated environment. A time-varying linear prediction error filter (TVLPEF) derived from a TVAR model performs well for time-varying FM interference mitigation in a minimally correlated environment. However, its performance deteriorates when applied for mitigating interference from a finitely correlated signal. The proposed decorrelating TVAR (DTVAR) model-based TVLPEF and a time-varying decision aided compensator (TVDAC) achieve notable SER performance gain for FM interference mitigation in a finitely correlated environment. Simulations for mitigation of a single linear component of FM interference in a finitely correlated CPM signal show that the TVLPEF based on a DTVAR model achieves notable performance gains over one based on a TVAR model for most of the range of moderate to high SIRs.

Chapter 5

Conclusion

5.1 Thesis Summary

In this thesis, we implemented IF estimation based on TVAR modeling and showed its application in rejecting nonstationary FM interference from a CPM signal. A nonstationary narrowband signal is characterized by its instantaneously changing frequency whose estimation has widespread application in the realm of Wireless Communications, Biomedical Signal Processing, and Acoustic and Speech processing. Adaptive signal processing techniques and non-parametric techniques based on the time-frequency distribution of a signal can be used for IF estimation and/or tracking of slowly varying and linear IF law signals. However, when the nonstationary signal has fast variation and possibly obeys a non-linear IF law, parametric approaches for IF estimation such as the TVAR approach can achieve superior performance. Parametric approaches are also suitable in practical scenarios where only short data records are available and/or used for processing.

In Chapter 1, we reviewed the literature starting with some works related to parametric and non-parametric approaches for IF estimation. A historical overview of the development of TVAR modeling and its application to IF estimation was presented. Some works containing the algorithms for practical and efficient implementation of IF estimation based on TVAR modeling were discussed. Finally, we reviewed some works related to the application of

IF estimation in FM interference mitigation. Very recent works in the literature pertinent to IF estimation and its application in nonstationary interference mitigation shows its continued interest among scholars and practitioners around the world.

In Chapter 2, we introduced TVAR modeling and outlined the procedure to estimate the IF. Using a multi-component linear FM signal and a non-linear FM signal, we demonstrated IF estimation based on TVAR modeling. The most challenging aspect of implementing a TVAR based IF estimation is the computation of the time-varying coefficients of the model. In Chapter 3, we implemented the Akaike algorithm and the Wax-Kailath algorithm for efficient and practical computation of the TVAR model parameters. Finally, in Chapter 4, we proposed a decorrelating TVAR model for IF estimation in a finitely correlated environment. Comparison of the decorrelating TVAR based IF estimator to a conventional TVAR based IF estimator reveals performance gains at moderate to high signal to FM component power ratios. A time-varying linear prediction error filter (TVLPEF) and a time-varying decision aided compensator (TVDAC) were proposed for FM interference mitigation in a CPM communications system without explicit IF estimation. Comparison of a time-varying linear PEF (TVLPEF) formed from DTVAR and TVAR coefficients for interference mitigation in a CPM communications system reveals notable SER performance gains at most moderate to high signal-to-interference ratios (SIRs).

5.2 Contributions

The contributions of this thesis are outlined below.

1. Although a TVAR model works well for IF estimation of FM components in white noise, its performance degrades when a finitely correlated signal is present in addition to the white noise. We proposed a decorrelating TVAR (DTVAR) model-based IF estimator for improving IF estimation in a finitely correlated environment.
 2. One of the main contributions of this thesis is the efficient computation of (D)TVAR parameters facilitating practical realization of IF estimation. The Akaike algorithm
-

and the Wax-Kailath algorithm were implemented to efficiently solve the linear system of equations to obtain the (D)TVAR parameters.

3. Finally, to illustrate IF estimation in an application, we proposed a DTVAR model-based time-varying linear prediction error filter to mitigate nonstationary narrowband FM interference from a CPM communications system.

5.3 Suggestions for Future Work

In this work, the focus was on implementation of IF estimation based on TVAR modeling and its application to reject FM interference in a CPM communications system. Future research can be directed toward theoretical and practical aspects of using TVAR modeling for IF estimation and time-varying signal processing including filter design. Some future work suggestions are outlined below.

1. In the simulations, knowledge of the number of FM components present in the signal and the type of IF law were assumed. This made it possible to pick reasonable values for TVAR order p and order of the basis function q . In real world applications, this information may not be available beforehand and would have to be selected based on some criteria. Statistical classification criteria such as the Akaike Information Criterion (AIC) and Minimal Description Length (MDL) criterion can possibly be used for this purpose [33].
 2. The basis functions used herein to describe the evolution of the time-varying parameters of the TVAR model are powers of time. This choice enabled us to get a covariance matrix with a block-Toeplitz structure that leads to computationally efficient solutions of the time-varying model parameters. Although the basis set used worked well for the signals considered, it was not chosen based on any optimality criterion. It will be useful to develop some criteria for selecting optimal basis functions depending on the characteristics of the nonstationary signal under consideration [13]. Additionally, it
-

will be interesting to see if other basis functions can lead to a computationally efficient structure for the covariance matrix.

3. At low SNRs, over-modeling can be used to allow the resulting extra poles to capture some noise and thus improve IF estimation performance. This is an extension of the low-SNR Prony method to time-varying signals [15]. When polynomial rooting is used, we obtain multiple IF estimates and are faced with the problem of correctly associating the IF to the individual FM components. The same situation is encountered when multiple FM components are present. Further research is needed to develop efficient classification algorithms for correctly associating the frequencies. Instead of using polynomial rooting, it would also be useful to investigate other approaches that can provide IF estimates of the FM components directly without requiring any association algorithms.
 4. A similar situation arises when IF estimates are obtained from the coefficients of a time-varying linear prediction error filter (TVLPEF). A TVLPEF has multiple notch frequencies, one of which is the true IF. Identifying the IF from the set of notch frequencies is non-trivial and a challenging problem. In this thesis, that problem was solved using a two step approach in which a rough IF estimate from a TVAR model was obtained first and in the second step the notch frequency closest to this rough IF estimate was picked as the improved IF estimate. Although this approach worked well for the signals considered herein, it is computationally inefficient. Further work needs to be performed to enhance the IF estimation procedure based on the coefficients of a TVLPEF.
 5. In Chapter 3, a couple of structures for sub-matrices were encountered that require inversion to implement the Wax-Kailath algorithm. Inversion of these lower dimension matrices is still performed using Gaussian elimination. Research needs to be conducted to find efficient recursive schemes that exploit the inherent structure to compute the inverse. This would further reduce the computational complexity of Wax-Kailath
-

algorithm.

6. The decorrelating TVAR (DTVVAR) based TVLPEF that was introduced to mitigate FM interference causes ISI and distorts the finitely correlated CPM signal. A time-varying decision aided compensator (TVDAC) was proposed to minimize this distortion using a one symbol decoder (OSD). Although performance of the proposed approach was superior to using a TVAR model for most moderate to high SIRs, performance was degraded for SIR between -30 dB and 5 dB. Since ISI can also be mitigated using a decision feedback equalizer (DFE), it may be worth while to look at an adaptive compensation approach based on the DFE instead of using the TVDAC.
-

Bibliography

- [1] B. Boashash, “Estimating and interpreting the instantaneous frequency of a signal. II. Algorithms and applications,” *Proceedings of the IEEE*, vol. 80, no. 4, pp. 540–568, April 1992.
 - [2] M. D. Macleod, “Fast nearly ML estimation of the parameters of real or complex single tones or resolved multiple tones,” *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 141–148, January 1998.
 - [3] P. Shan, “Time-varying Autoregressive Model Based Signal Processing with Applications to Interference Rejection in Spread Spectrum Communications,” Ph.D. dissertation, Virginia Tech, July 1999.
 - [4] P. Shan and A. A. (Louis) Beex, “High-resolution instantaneous frequency estimation based on time-varying AR modeling,” in *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, October 1998, pp. 109–112.
 - [5] H. K. Kwok and D. L. Jones, “Improved instantaneous frequency estimation using an adaptive short-time Fourier transform,” *IEEE Transactions on Signal Processing*, vol. 48, no. 10, pp. 2964–2972, October 2000.
 - [6] B. Barkat, “Instantaneous frequency estimation of nonlinear frequency-modulated signals in the presence of multiplicative and additive noise,” *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2214–2222, October 2001.
-

-
- [7] V. N. Ivanović, M. Daković, and L. Stanković, "Performance of quadratic time-frequency distributions as instantaneous frequency estimators," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 77–89, January 2003.
- [8] P.-L. Shui, H.-Y. Shang, and Y.-B. Zhao, "Instantaneous frequency estimation based on directionally smoothed pseudo-Wigner-Ville distribution bank," *IET Radar, Sonar and Navigation*, vol. 1, no. 4, pp. 317–325, August 2007.
- [9] Y. Grenier, "Time-dependent ARMA modeling of nonstationary signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 899–911, August 1983.
- [10] T. S. Rao, "The fitting of non-stationary time-series models with time-dependent parameters," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 32, no. 2, pp. 312–322, 1970.
- [11] L. A. Liporace, "Linear estimation of nonstationary signals," *Journal of the Acoustical Society of America*, vol. 58, no. 6, pp. 1288–1295, December 1975.
- [12] M. Hall, A. V. Oppenheim, and A. Willsky, "Time-varying parametric modeling of speech," in *IEEE Conference on Decision and Control*, vol. 16, no. 1, December 1977, pp. 1085–1091.
- [13] K. C. Sharman and B. Friedlander, "Time-varying autoregressive modeling of a class of nonstationary signals," in *IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 9, no. 1, March 1984, pp. 227–230.
- [14] B.-S. Chen, T.-Y. Yang, and B.-H. Lin, "Adaptive notch filter by direct frequency estimation," *Signal Processing*, vol. 27, no. 2, pp. 161–176, May 1992.
- [15] A. A. (Louis) Beex and P. Shan, "A time-varying Prony method for instantaneous frequency estimation at low SNR," in *IEEE International Symposium on Circuits and Systems*, vol. 3, May 1999, pp. 3–8.
-

-
- [16] A. T. Johansson and P. R. White, “Instantaneous frequency estimation at low signal-to-noise ratios using time-varying notch filters,” *Signal Processing*, vol. 88, no. 5, pp. 1271–1288, May 2008.
- [17] W. F. Trench, “An algorithm for the inversion of finite Toeplitz matrices,” *SIAM Journal on Applied Mathematics*, vol. 12, no. 3, pp. 515–522, September 1964.
- [18] H. Akaike, “Block Toeplitz matrix inversion,” *SIAM Journal on Applied Mathematics*, vol. 24, no. 2, pp. 234–241, March 1973.
- [19] J. Rissanen, “Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring positive matrix polynomials,” *Mathematics of Computation*, vol. 27, pp. 147–154, January 1973.
- [20] M. Wax and T. Kailath, “Efficient inversion of Toeplitz-block Toeplitz matrix,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. ASSP-31, no. 5, pp. 1218–1747, October 1983.
- [21] N. Kalouptsidis, G. Carayannis, and D. Manolakis, “On block matrices with elements of special structure,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1982, pp. 1744–1747.
- [22] Y. Wang and H. Krishna, “On fast and superfast algorithms for solving block Toeplitz systems,” in *Twenty Third Asilomar Conference on Signals, Systems, and Computers*, vol. 2, October 1989, pp. 643–647.
- [23] A. E. Yagle, “A fast algorithm for Toeplitz-block-Toeplitz linear systems,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, May 2001, pp. 1929 – 1932.
- [24] D. Hui, “Method of inverting nearly Toeplitz or block Toeplitz matrices,” U.S. Patent US7388935, June 17, 2008.
-

-
- [25] P. Shan and A. A. (Louis) Beex, “FM interference suppression in spread spectrum communications using time-varying autoregressive model based instantaneous frequency estimation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, March 1999, pp. 2559–2562.
- [26] M. G. Amin, “Interference mitigation in spread spectrum communication systems using time-frequency distributions,” *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 90–101, January 1997.
- [27] L. F. Chaparro and R. Suleesathira, “Non-stationary jammer excision in spread spectrum communications via discrete evolutionary and Hough transforms,” *Signal Processing*, vol. 83, no. 5, pp. 1117–1133, May 2003.
- [28] L.-P. Zhu, G.-R. Hu, and Y.-S. Zhu, “Nonstationary interference excision in time-frequency domain using adaptive hierarchical lapped orthogonal transform for direct sequence spread spectrum communications,” *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 12, pp. 1210–1218, June 2003.
- [29] A. Francos and M. Porat, “Non-stationary signal processing using time-frequency filter banks with applications,” *Signal Processing*, vol. 86, no. 10, pp. 3021–3030, October 2006.
- [30] P. Shan and A. A. (Louis) Beex, “Time-varying filtering using full spectral information for soft-cancellation of FM interference in spread spectrum communications,” in *IEEE Workshop on Signal Processing Advances in Wireless Communications*, May 1999, pp. 321–324.
- [31] R. Kadanna Pally and A. A. (Louis) Beex, “Modeling of time-varying instantaneous frequency in a finitely correlated environment,” in *16th International Conference on Digital Signal Processing*, Santorini, Greece, July 2009, ppr T3C.3.
- [32] —, “FM interference mitigation in a finitely correlated environment using a decorrelating time-varying autoregressive model,” in *The 10th IEEE International*
-

- Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Perugia, Italy, June 2009.
- [33] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1996.
- [34] N. Levinson, "The Wiener RMS error criterion in filter design and prediction," *Journal of Mathematics and Physics*, vol. 25, pp. 261–278, 1947.
- [35] S. Zohar, "Toeplitz matrix inversion: The Algorithm of W. F. Trench," *Journal of the Association for Computing Machinery*, vol. 16, no. 4, pp. 592–601, October 1969.
- [36] M. Morf, B. Dickinson, T. Kailath, and A. Vieira, "Efficient solution of covariance equations for linear prediction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 429–433, October 1977.
- [37] H. Cramer, "On some classes of nonstationary stochastic processes," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 2, June 1960, pp. 57–78.
- [38] N. C. Huang and J. K. Aggarwal, "On linear shift-variant digital filters," *IEEE Transactions on Circuits and Systems*, vol. 27, no. 8, pp. 672–679, August 1980.
- [39] Y. Grenier, "Rational nonstationary spectra and their estimation," in *First ASSP Workshop on Spectral Estimation*, August 1981.
- [40] R. A. Wiggins and E. A. Robinson, "Recursive solution to the multichannel filtering problem," *Journal of Geophysical Research*, vol. 70, no. 8, pp. 1885–1891, April 1965.
- [41] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, 1st ed. Philadelphia, PA, USA: SIAM, 2000.
- [42] R. W. Farebrother, *Linear Least Squares Computations*. New York, NY, USA: Marcel Dekker Inc., 1988, vol. 91.
-

-
- [43] M. J. Goldstein, “Reduction of pseudoinverse of a Hermitian persymmetric matrix,” *Mathematics of Computation*, vol. 28, pp. 715–717, July 1974.
- [44] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins, 1996.
- [45] M. K. Nezami, “Techniques for acquiring and tracking MIL-STD-181B signals,” in *IEEE Military Communications Conference (MILCOM)*, vol. 1, October 2002, pp. 224–231.
-