



Computer Science Department
University of Crete



Institute of Computer Science
FO.R.T.H.

Fast Least-Squares Solution for Harmonic and Sinusoidal Models

MSc Thesis

Georgios Tzedakis

Heraklion
November 2010

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING
UNIVERSITY OF CRETE

Fast Least-Squares Solution for Harmonic and Sinusoidal Models

Submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Master of Science

November 1, 2010

© 2010 University of Crete & ICS-FO.R.T.H. All rights reserved.

Author:

Georgios Tzedakis
Department of Computer Science

Committee:

Supervisor

Yannis Stylianou
Associate Professor

Member

Athanasios Mouchtaris
Assistant Professor

Member

Gerasimos Potamianos
Director of Research (Rank A)

Accepted by:

Chairman of the
Graduate Studies Committee

Angelos Bilas
Associate Professor

Heraklion, November 2010

Abstract

The sinusoidal model and its variants are commonly used in speech processing. In the literature, there are various methods for the estimation of the unknown parameters of the sinusoidal model. Among them, the most known methods are the ones based on the *Fast Fourier Transform* (FFT), on *Analysis-By-Synthesis* (ABS) approaches and through *Least Squares* (LS) methods.

The LS methods are more accurate and actually optimum for Gaussian noise, and thus, more appropriate for high quality estimations. In addition, LS methods prove to be able to cope with short analysis windows. On the contrary, the FFT and the ABS- based methods cannot handle overlapping frequency responses, in other words, they cannot handle short analysis windows. This is important since in the case of short analysis windows the stationary assumption for the signal is more valid. However, LS solutions are in general slower compared to FFT-based algorithms and optimized implementations of ABS schemes. In the present thesis, our goal is to alleviate the computational burden that the LS-based techniques bear, such that both the increased accuracy and the faster computational implementation can be achieved.

The four models of which the amplitude coefficients will be estimated, namely the Harmonic, Sinusoidal, Quasi-Harmonic and Generalized Quasi-Harmonic models, are re-introduced. Then, each model is studied individually and the straightforward LS solution for the amplitude estimation is presented.

The sources of computational load in the case of an LS solution are indicated and various computational improvements are introduced for each model in terms of its computational complexity and execution time. The first speed up process includes performing matrix multiplications manually, which yields a direct formula for every element of the result. For the next accelerating method, we show how we can calculate a certain matrix of exponentials using primarily multiplications. As a final acceleration, having realized that certain elements of a matrix, which is needed to be calculated and then inverted, play a less important role in the process of deriving the solution, we allow certain approximations of the matrix by omitting the calculation of the less important elements.

Finally, it is demonstrated that by following the suggested steps, the complexity of LS-based solution along with the execution time, are reduced. The methods are evaluated by analyzing and re-synthesizing randomly created synthetic signals and calculating the Mean Square Error, Signal-to-Reconstruction Error Ratio and CPU time improvement for each step. Next, in an effort to test the robustness of our hastening methods, we illustrate their competence in analyzing noisy synthetic signals. Furthermore, as a final test we check the ability of our amplitude estimation mechanisms to analyze and synthesize real-world voiced speech signals.

Περίληψη

Το Ημιτονοειδές μοντέλο και οι παραλλαγές του χρησιμοποιούνται ευρέως στην επεξεργασία του φωνής. Στη βιβλιογραφία, υπάρχουν διάφορες μέθοδοι για την εκτίμηση των άγνωστων παραμέτρων του ημιτονοειδούς μοντέλου. Μεταξύ των οποίων, οι πιο δημοφιλείς μέθοδοι είναι αυτές που βασίζονται στο *Γρήγορο Μετασχηματισμό Φουριέ* (ΓΜΦ), σε προσεγγίσεις *Ανάλυσης-Μέσω-Σύνθεσης* (ΑΜΣ) και μέσω μεθόδων επίλυσης *Ελαχίστων Τετραγώνων* (ΕΤ).

Οι μέθοδοι ΕΤ είναι πιο ακριβείς και στην πραγματικότητα βέλτιστες για Γκαουσιανό θόρυβο, και ως εκ τούτου, πιο κατάλληλες για υψηλής ποιότητας προσεγγίσεις. Επιπλέον, οι μέθοδοι ΕΤ αποδεικνύεται ότι είναι σε θέση να αντιμετωπίσουν τη χρήση βραχυπρόθεσμων παραθύρων ανάλυσης. Σε αντίθεση, οι μέθοδοι βασισμένες σε ΓΜΦ και το ΑΜΣ δεν μπορούν να χειριστούν επικαλυπτόμενες αποκρίσεις συχνότητας, με άλλα λόγια, δεν μπορούν να χειριστούν βραχυπρόθεσμα παράθυρα ανάλυσης, τα οποία είναι σημαντικά λόγω του γεγονότος ότι, όταν χρησιμοποιούνται η παραδοχή στασιμότητας για το σήμα είναι πιο ισχυρή. Ωστόσο, οι ΕΤ λύσεις είναι γενικά πιο αργές σε σύγκριση με αλγόριθμους βασισμένους σε ΓΜΦ και βελτιστοποιημένες εφαρμογές των συστημάτων ΑΜΣ. Στην παρούσα εργασία, στόχος μας είναι να μειωθεί η υπολογιστική επιβάρυνση που φέρουν οι τεχνικές βασισμένες σε ΕΤ, έτσι ώστε να επιτευχθούν τόσο η μεγαλύτερη ακρίβεια όσο και πιο γρήγοροι υπολογιστική υλοποίηση.

Τα τέσσερα μοντέλα στα οποία θα εφαρμοστεί η εκτίμηση των συντελεστών πλάτους, δηλαδή τα Αρμονικό, Ημιτονοειδές, Οιονεί-Αρμονικό και Γενικευμένο Οιονεί-Αρμονικό Μοντέλα, πρόκειται να επανεισαχθούν. Στη συνέχεια, κάθε μοντέλο μελετάται ξεχωριστά και η άμεση λύση ΕΤ για την εκτίμηση του εύρους παρουσιάζεται.

Οι πηγές του υπολογιστικού φορτίου για την περίπτωση της λύσης ΕΤ υποδεικνύονται και διάφορες βελτιώσεις όσον αφορά την υπολογιστική πολυπλοκότητα και τον χρόνο εκτέλεσης εισάγονται για κάθε μοντέλο. Η πρώτη επιταχυντική διαδικασία περιλαμβάνει την διεκπεραίωση πολλαπλασιασμού πινάκων με το χέρι, δίνοντας έναν άμεσο τύπο για κάθε στοιχείο του αποτελέσματος. Για την επόμενη επιτάχυνση της μεθόδου, θα δείξουμε πώς μπορούμε να υπολογίσουμε μια έναν συγκεκριμένο πίνακα εκθετικών χρησιμοποιώντας κυρίως μόνο πολλαπλασιασμούς. Ως τελική επιτάχυνση, έχοντας συνειδητοποιήσει ότι ορισμένα στοιχεία ενός πίνακα, που απαιτείται να υπολογιστεί και να αντιστραφεί, παίζουν λιγότερο σημαντικό ρόλο στη διαδικασία εύρεσης της λύσης, επιτρέπουμε προσεγγίσεις του πίνακα αποφεύγοντας τον υπολογισμό των λιγότερο σημαντικών στοιχείων. Τέλος, δείχνεται ότι, ακολουθώντας τις προτεινόμενες ενέργειες, η πολυπλοκότητα της λύσης βασισμένη σε ΕΤ, καθώς και ο χρόνος εκτέλεσης, μειώνονται. Οι μέθοδοι αξιολογούνται αναλύοντας και εκ νέου συνθέτοντας τυχαία συνθετικά σήματα και τον υπολογισμό του μέσου τετραγωνικού σφάλματος, του λόγου σήματος προς σφάλμα ανακατασκευής και της βελτίωση του χρόνου υπολογισμού για κάθε βήμα. Συνεχίζοντας, σε μια προσπάθεια να δοκιμαστεί η ευρωστία των επιταχυντικών μεθόδων μας δείχνουμε την ικανότητα τους όσον αφορά την ανάλυση θορυβωδώς συνθετικών σημάτων. Επιπλέον, ως τελική δοκιμή ελέγχουμε την ικανότητα των μηχανισμών μας για την εκτίμησης του πλάτους να αναλύουν και να συνθέτουν σήματα φωνημάτων.

Ευχαριστίες

Καταρχήν θα ήθελα να ευχαριστήσω τον επόπτη μου Αναπληρωτή Καθηγητή κ. Ιωάννη Στυλιανού καθώς και τον πλέον σήμερα κάτοχο διδακτορικού διπλώματος Ιωάννη Πανταζή για την πολύτιμη καθοδήγηση και συνεχή υποστήριξη τους στο πλαίσιο της συνεργασίας μας κατά την διάρκεια της δημιουργίας της παρούσης εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω θερμά τον Επίκουρο Καθηγητή κ. Αθανάσιο Μουχτάρη και τον Διευθυντή Έρευνας κ. Γεράσιμο Ποταμιάνο που συμπλήρωσαν ως μέλη την εξεταστική επιτροπή της μεταπτυχιακής μου εργασίας.

Πολλές ευχαριστίες οφείλω στον διδάκτορα Γεώργιο Τζαγκαράκη για τις πολύτιμες επισημάνσεις του όσον αφορά το κείμενο της εργασίας. Επίσης ευχαριστώ τους συνάδελφους και μέλη του εργαστηρίου για την συμπαράσταση και την συμβολή τους για την δημιουργία ευχάριστης ατμόσφαιρας στο χώρο εργασίας: Μαρία Κουτσογιαννάκη, Γιώργο Γρέκα, Χριστίνα Αλεξάνδρα Λιονουδάκη, Μαίρη Αστρινάκη, Γιώργο Καφετζή.

Επιπρόσθετα, ευχαριστώ όλους τους φίλους μου οι οποίοι στάθηκαν δίπλα μου σε αυτό το διάστημα και που επίσης βοήθησαν στο να περάσει πολύ πιο ευχάριστα.

Τέλος θέλω να ευχαριστήσω θερμά τους γονείς μου και την οικογένεια μου για την ανιδιοτελή στήριξη τους σε όλες τις αποφάσεις και τα βήματα μου.

Σας ευχαριστώ όλους!!!

Contents

List of Figures	xi
List of Tables	xiii
Abbreviations	xv
1 Introduction	1
1.1 Sinusoidal Modeling Applications	2
1.2 Amplitude Estimation	3
1.3 Frequency Estimation	4
1.4 Motivation	5
1.5 Thesis Contribution	6
1.6 Structure of the Thesis	6
2 Sinusoidal Models	7
2.1 Harmonic Model	7
2.2 Sinusoidal Model	11
2.3 Quasi-Harmonic Model	14
2.4 Generalized Quasi-Harmonic Model	17
3 Speeding Up the Computations	21
3.1 Window Parameterization	23
3.2 Harmonic Model	25

3.2.1	HM Step 1: Fast Matrix Multiplication	25
3.2.2	HM Step 2: Faster Computation of \mathbf{E}_{0h}	27
3.2.3	HM Step 3: \mathbf{R}_{0h} Matrix Approximation	28
3.3	Sinusoidal Model	30
3.3.1	SM Step 1: Fast Matrix Multiplication	30
3.3.2	SM Step 2: Faster Computation of \mathbf{E}_0	31
3.3.3	SM Step 3: \mathbf{R}_0 Matrix Approximation	32
3.4	Quasi-Harmonic Model	34
3.4.1	QHM Step 1: Fast Matrix Multiplication	34
3.4.2	QHM Step 2: Faster Computation of \mathbf{E}_{0h} and \mathbf{E}_{1h}	36
3.4.3	QHM Step 3: \mathbf{R}_{QHM} Matrix Approximation	37
3.5	Generalized Quasi-Harmonic Model	40
3.5.1	GQHM Step 1: Fast Matrix Multiplication	40
3.5.2	GQHM Step 2: Faster Computation of \mathbf{E}_0 and \mathbf{E}_1	42
3.5.3	GQHM Step 3: \mathbf{R}_{GQHM} Matrix Approximation	42
4	Evaluation	45
4.1	Computational Complexity	45
4.2	Synthetic signals	47
4.2.1	Mean Square Error	47
4.2.2	Signal-to-Reconstruction Error Ratio	48
4.2.3	CPU time	49
4.2.4	Noisy Signals	50
4.3	Voice Signals	54
5	Conclusions and Future Work	57
	References	60

List of Figures

3.1	Visual representation of different windows.	23
3.2	Visual representation of the square of different windows.	24
3.3	Average $ \mathbf{R}_{0h} $ values as a function of their distance from main diagonal. .	28
3.4	Average $ \mathbf{R}_0 $ values as a function of their distance from main diagonal. .	33
3.5	Average $ \mathbf{R}_{1h} $ values as a function of their distance from main diagonal. .	38
3.6	Average $ \mathbf{R}_{2h} $ values as a function of their distance from main diagonal. .	39
3.7	Average $ \mathbf{R}_1 $ values as a function of their distance from main diagonal. .	43
3.8	Average $ \mathbf{R}_2 $ values as a function of their distance from main diagonal. .	44

List of Tables

3.1	Different values of α_w give various windows.	23
4.1	Average MSE for each individual model and parameter.	48
4.2	Average SRER.	49
4.3	Average CPU times.	49
4.4	CPU time differences between every step and the previous one.	50
4.5	CPU time improvements in percentages.	50
4.6	Average SRER with additive noise (SNR=5).	51
4.7	Average SRER with additive noise (SNR=10).	51
4.8	Average SRER with additive noise (SNR=20).	51
4.9	Average SRER with additive noise (SNR=30).	51
4.10	Average SRER with additive noise (SNR=40).	52
4.11	Average SRER with additive noise (SNR=50).	52
4.12	Average SRER with additive noise (SNR=60).	52
4.13	Average SRER with additive noise (SNR=70).	52
4.14	Average SRER with additive noise (SNR=80).	53
4.15	Average SRER for voiced speech using HM and QHM.	54
4.16	Average SRER voiced speech using SM and GQHM.	55

Abbreviations

ABS:	Analysis-By-Synthesis
FFT:	Fast Fourier Transform
GQHM:	Generalized Quasi-Harmonic Model
HM:	Harmonic Model
LPC:	Linear Predictive Coding
LS:	Least Squares
MLE:	Maximum Likelihood Estimator
MSE:	Mean Square Error
OLA:	OverLap-Add
QHM:	Quasi-Harmonic Model
RMS:	Root Mean Square
SM:	Sinusoidal Model
SNR:	Signal-to-Noise Ratio
SRER:	Signal-to-Reconstruction Error Ratio

Hz	hertz
ms	millisecond
dB	decibel

Chapter 1

Introduction

Parametric modeling involves the reduction of a complicated process to a simpler one with a smaller number of parameters. It is commonly used in speech, as well as in other distinct fields, such as communication systems and economic models.

The parameter reduction implies approximation but even so, the signal is often decomposed into a form that is more easily or efficiently processed than the original. Moreover, if the parameters turn out to be psychically meaningful, then insight can be gained into the behavior, the nature and even the mechanism of the overall process by understanding the influence of each parameter. Thus, in order to produce parameters that can be physically interpreted, in speech most models are derived by an effort to replicate the human speech production mechanisms.

As an example, *Linear Predictive Coding* (LPC) models speech as the output of a linear, time-varying filter modeling the vocal tract. The filter is excited by either an excitation pulse train for voiced speech or a random noise source for unvoiced speech (purporting the role of the glottis). However, the implicit assumptions made about the excitation signal in this model are quite restrictive, and result in speech which may sound “unnatural” for input signals which do not match these assumptions exactly.

The desire for an alternative speech model, which is both more efficient and more general in representing speech, motivated development of the *Sinusoidal Model* (SM) of speech in the mid-1980’s. Sinusoidal modeling is usually modeled in a frame-based manner as a superposition of modulated sinusoids. One of the most widely known sinusoidal model implementations is the one introduced by McAulay and Quatieri in 1986 [9].

1.1 Sinusoidal Modeling Applications

Sinusoidal modeling of speech, musical and sound signals in general is widely recognized as a very powerful and flexible method. One of the applications in which the sinusoidal modeling plays a crucial role is feature extraction for classification of music and video. Many features have been proposed that are computed from the sinusoidal parameters, which are used in the context of non-parametric estimation of control parameters for physical models [2, 3], audio compression/coding, audio annotation [13] and instrument recognition [14].

Moreover, as one would expect, since the sinusoidal model should represent any periodic signal efficiently and most musical instruments produce periodic signals, sinusoidal models prove to be useful in music processing applications that usually follow the procedures of sinusoidal analysis of the original signal, perform modification of amplitude/frequency/time-scale of individual components and synthesis of the altered signal [23, 4, 5]. Another application can be found in acoustical oceanography and particularly in the study of sounds produced by marine mammals [24]. In addition, sinusoidal models are widely used in various ways for signal enhancement, such as: the enhancement for the hearing impaired [21, 22], the speech enhancement in noise [17], the peak-to-RMS reduction [16, 18] and the co-channel interference suppression (a problem which is similar to separation of sound sources) [15, 25, 8, 29, 30, 32, 33].

One of the main reasons for which sinusoidal modeling has been proven to be efficient in the area of speech processing is that, due to its simplicity and the implicit physical meaning of its parameters, it allows to vary the pitch and the duration of the sound independently [20] by allowing sound modifications of a very high quality [19, 5]. Combined with the ability to concatenate speech (and music) it result in an efficient implementation of text-to-speech systems [7]. Other applications apart from time and pitch scaling at the same area include speech analysis/synthesis [9], speech compression/coding [10] and speaker modification [27].

1.2 Amplitude Estimation

The sinusoidal models which will be employed in our work, as well as most variants of sinusoidal models, are described by two types of parameters to be estimated, namely the parameters concerning the amplitudes and the parameters related to the frequencies. The main focus of this thesis will be given on the estimation of the amplitude parameters. In this section, we will introduce the three main groups of methods used for amplitude estimation. The first main group utilizes the frequency response of the windowed signal segment, the second estimation method is based on an analysis-by-synthesis scheme and final group is known as the *Least Squares* (LS) type of method.

Early methods estimated the amplitude parameters from individual peaks using a parabolic interpolation over the main lobe of the log frequency response [9, 23]. One advantage of these methods is that they perform estimation of the frequencies and of the amplitudes simultaneously. However, these methods cannot handle frequency responses that are partially overlapping and therefore they require large windows. When using large windows, on the other hand, the local stationarity assumption becomes invalid. The accuracy can be increased by performing zero-padding (typically turning the length of the signal fragment to be analyzed to a power of two) before utilizing the *Fast Fourier Transform* (FFT) algorithm, since zero-padding in the time domain results in an increased sampling rate in the frequency domain. However, increasing the zero-padding factor, which is defined as the ratio between the zero-padded signal length and the original signal length, results in an increment of the computational load as well.

The second group of methods that concerns the amplitude estimation of sinusoidal components conscripts a technique based on analysis-by-synthesis, by using an iterative LS method [4, 6, 5]. More specifically, the so-called *Analysis-By-Synthesis/Overlap-Add* (ABS/OLA) method, detects the most dominant sinusoidal component, estimates the amplitude and subtracts the component from the spectrum. The process is repeated iteratively until a predetermined error criterion is no longer decreasing significantly enough or it has reached a predefined acceptable level. Like the previous methods, this group cannot handle frequencies frequencies that are close enough resulting in overlapping frequency responses and requiring the use of adequately large windows.

The last group of amplitude estimation methods, which is also the one that this work deals with, are based on the minimization of an error function which is usually the weighted sum of a squared error between the model and the actual analyzed frame. These techniques consist of Least Squares methods which estimate all amplitudes simultaneously [29, 30, 32, 33]. The main advantage of such methods is that they can handle close frequencies more effectively than the previous ones and therefore they can be also be applied to small analysis windows. For Gaussian noise is equivalent to the *Maximum Likelihood Estimator* (MLE). However their main drawback is their computational complexity scales at a power of three as a function of the number of the sinusoidal components.

1.3 Frequency Estimation

In the subsequent analysis, except if it is stated otherwise, we will assume that the frequencies of the sinusoids are known a priori. Though, the importance of having the correct, or close enough, frequency values should be stressed. Not only when the signal reconstruction is performed the frequencies themselves will be incorrect, but also by not using correct frequency values as input when performing amplitude estimation, then wrong amplitude parameter values will be yielded as well. That double impact that erroneous frequency values have, results in a significantly distorted signal which does not resemble the original. That is why, for completeness, the two main categories of methods for performing frequency estimation will be briefly discussed in this section.

The first category consists of methods which estimate the fundamental frequency, while the second one contains methods that estimate all the dominant frequencies. The problem solved by the methods belonging in the first class is often referred to as pitch estimation. The methods of the second class are employed more often in for non-harmonic models, in other words, for models of which one frequency value shares no mutual information about another frequency value. Pitch estimation is more meaningful when implementing harmonic models, that is, models of which the frequencies are assumed to be integer multiples of a fundamental frequency ($f_k = kf_0$), or routinely called pitch.

As far as pitch estimation is concerned, various methods exist. The majority of them follows the basic idea of searching for peaks in a short-time autocorrelation function. The autocorrelation function indicates “self-similarity” between the original signal $s[n]$ and the signal shifted in time by an offset P , which represents the under examination value for the pitch, $s[n+P]$. It can be shown that maximizing the autocorrelation is essentially the same with finding $\hat{P} > \epsilon$ such that $\hat{P} = \max_P \left(\sum_{n=-\infty}^{+\infty} s[n]s[n+P] \right)$. The main problem with such methods is that the peak at the pitch period is not always the one with the greatest amplitude. Although, longer windows usually help assuring that the peak close to the pitch becomes the largest, however, another usual problem introduced by larger windows is that peaks arise at pitch multiples as well.¹

For frequency estimation that yields all the frequencies, without any assumption for their values, the most widely used technique is known as peak picking. Individual local maxima are detected in the sampled spectrum obtained by an FFT. As with amplitude estimation techniques that take advantage of the spectrum of the signal, sufficiently large windows are required (with a frame length of at least four times the longest expected pitch period to obtain sufficient spectral resolution). Of course, frames are windowed and zero-padded to a fixed length, typically at a power of two. Windowing the analysis frames reduces the spurious peaks in the spectrum, while, zero-padding effectively interpolates the spectrum so that peaks may be located more accurately, as well as allowing efficient FFT algorithms.

¹A common problem in pitch estimation referred as pitch-halving and pitch doubling, whereby the pitch estimate is half or double the true pitch.

1.4 Motivation

In section 1.2, the three main amplitude estimation categories were presented. The first one, the FFT-based approaches, which try to balance between the accuracy of the estimated parameters, by increasing the window size and the zero-padding factor, the fast computation of the parameters, by decreasing zero padding-factor, and the non-stationarity of the analyzed signal, by applying a smaller window size to the signal.

The second category consists of methods applying a analysis-by-synthesis scheme by iteratively solving LS problems. Such algorithms can have time complexity as low as $\mathcal{O}(N \log N)$, with N being the signal length, when an efficient implementation is used (look-up tables for the frequency responses). On the other hand, these methods cannot resolve close frequencies, which result in the overlapping frequency responses.

The last category contains Least Squares methods which estimate all the amplitudes simultaneously. The main asset of these methods is that they can handle overlapping frequency responses better than the previous families of amplitude estimators. That fact can also be translated to an ability to cope with smaller windows. However, their drawback is that in general they have a high computational complexity of order $\mathcal{O}(K^2N)$ with K being the number of sinusoidal components.

The key point is that many applications require an amplitude estimator which can handle overlapping frequency responses. For instance, the separation of multiple harmonic sound sources, the monophonic recordings with strong reverberation and when the usage of small windows is mandatory. The choice of small windows can be important because the constant parameter assumption is more likely to hold. In addition, when small enough windows are used, then the phase interpolation is often made redundant, resulting in simpler, and therefore faster signal re-synthesis.

However, the high computational complexity (practically resulting in an increased processing time) restricts the usage of such LS methods more often than not. Many of the applications mentioned in section 1.1 would only be of interest if they could be implemented in real time. While for others real-time implementations would create more opportunities to be exploited. For instance, it is desirable that procedures, such as speech analysis, enhancement and synthesis in noisy environment could be implemented in real time even by mobile devices, which are characterized by limited computational and power resources.

Thus, in an effort to alleviate the disadvantages of the methods belonging to the latter category, the problem of reducing the computational load of the amplitude estimation using the Least Squares-based methods is addressed. In that way, both an increased accuracy and faster computations can be achieved.

1.5 Thesis Contribution

Most of the work presented in this thesis has been accepted and presented in Interspeech 2010 [31]. Additionally, an example where the methods presented in this thesis can be effectively applied, is a model called adaptive quasi-harmonic plus noise model presented in [12]. That model finds the amplitude components which are optimal, in a Least Squares sense and uses these components to correct the frequencies. These steps can be performed iteratively to further improve the frequency estimation. Since the LS solution is performed more than once, it is important that it is done fast. A feat which, as it will be demonstrated, is achieved by the techniques proposed in the present thesis.

1.6 Structure of the Thesis

This thesis deals with the problem of accelerating the computational procedures for estimating the amplitude parameters of certain sinusoidal models. For the estimation process the LS solution is used. The rest of the thesis is organized as follows: Chapter 2 introduces the Harmonic, Sinusoidal Quasi-Harmonic and Generalized Quasi-Harmonic models, of which the amplitude parameter estimation is our goal. In the same chapter, for each model separately, it is presented how the direct LS solution for finding the amplitudes is realized.

In Chapter 3, we develop the computational enhancements for each model individually. First, we show how the window function is expressed parametrically and we use the square of that expression to express analytically the elements of a matrix resulting from matrix multiplications. The next speed up consists of rewriting an array of exponentials in terms of trigonometric functions. Doing so we can take advantage of the trigonometric functions and the form of their arguments and express them as a solution of a differential equation, by replacing them mostly with multiplication operations. The last step consists of approximating the array that is to be inverted. We do so by noticing that sinusoids which are far apart from each other have small interference. By taking advantage of that fact we eliminate (set to zero) specific matrix elements depending on the model.

In Chapter 4, we present the results that stem from the improvements of the previous chapter. We begin by showing the reduction of the complexity that has been achieved. Then, with the help of synthetic signals, we find the *Mean Square Error* (MSE) and the *Signal-to-Reconstruction Error Ratio* (SRER) to illustrate that our methods are able to compute the amplitudes with accuracy equivalent to that achieved by the direct solution. Then, we show how well the actual speed up improvements perform by measuring the CPU time it takes to analyze the above synthetic signals. The last experimentation using synthetic signals consists of adding noise to the original signal in order to check whether or not our methods are robust to noise, in other words that they do not introduce further noise in noisy signals. As a final test, our techniques are tested in real voiced signals. Finally, we close with some remarks and propose future research directions in Chapter 5.

Chapter 2

Sinusoidal Models

The models we employ in the present work are introduced in this chapter. We will state each model and describe how its parameters can be yielded with the assistance of LS, in particular, how the amplitude parameters are estimated. The models at hand are the *Harmonic Model* (HM), the *Sinusoidal Model* (SM), the *Quasi-Harmonic Model* (QHM) and the *Generalized Quasi-Harmonic Model* (GQHM).

The main difference between the sinusoidal model and the harmonic model is the fact that the frequency values of the HM are integer multiples of a fundamental frequency, traditionally denoted by f_0 , whilst the SM has no such restriction. The same restriction is applied to the QHM. That is, it assumes as acceptable frequency values f_k only those for which $f_k = kf_0$, for $k = -K, \dots, K$ holds true. Thus, knowing the fundamental frequency yields all the other frequencies. On the other hand, as the name implies, the GQHM is the non harmonic counterpart of QHM. The discrimination between the general and the harmonic cases is done because by restricting the feasible frequency values certain calculations are greatly simplified, rendering that special case worth mentioning. Additionally, the human speech can be considered as a harmonic signal where f_0 denotes the fundamental pitch of the speaker and kf_0 are the harmonics. Also, all the frequency values may not be known in advance or be harder to be estimated in contrast to the fundamental frequency.

2.1 Harmonic Model

Consider that $\mathbf{s} = [s[-N] \ s[-N+1] \ \dots \ s[N-1] \ s[N]]^T$ is the original signal of duration $2N+1$ to be modeled. Assuming that \mathbf{s} will represent a relatively small portion of speech signal, it is reasonable to adopt that is stationary. Then, it may be decomposed into a harmonic and a noisy part denoted by $h_0[n]$ and $w_0[n]$, respectively:

$$s[n] = h_0[n] + w_0[n] . \quad (2.1)$$

Assuming a frame of two pitch periods, which is centered around zero, the harmonic model states that the harmonic part is modeled as follows:

$$h_0[n] = \sum_{k=-K}^K a_k e^{j2\pi nk f_0 / f_s}, \quad n = -N, \dots, N, \quad (2.2)$$

where f_0 is the fundamental frequency, a_k denotes the complex amplitude of the k th component, and K is equal to the number of the sinusoidal components. What f_s stands for is the sampling frequency.

By substituting (2.2) to (2.1) yields the following expression:

$$s[n] = \sum_{k=-K}^K a_k e^{j2\pi nk f_0 / f_s} + w_0[n], \quad n = -N, \dots, N. \quad (2.3)$$

Our interest lies in the estimation of the complex amplitudes a_k given the number of components K and the fundamental frequency f_0 . A Least Squares method minimizes the square of the error, which is given by:

$$\begin{aligned} \epsilon_{\mathbf{a}} &= \sum_{n=-N}^N (s[n] - h_0[n])^2 \\ &= \sum_{n=-N}^N \left(s[n] - \sum_{k=-K}^K a_k e^{j2\pi nk f_0 / f_s} \right)^2 \\ &= \sum_{n=-N}^N (s[n] - (\mathbf{E}_{Kh})^n \mathbf{a})^2 \\ &= (\mathbf{s} - \mathbf{E}_{0h} \mathbf{a})^H (\mathbf{s} - \mathbf{E}_{0h} \mathbf{a}), \end{aligned} \quad (2.4)$$

where $\mathbf{E}_{Kh} = [e^{j2\pi(-K)f_0/f_s} \quad e^{j2\pi(-K+1)f_0/f_s} \quad \dots \quad e^{j2\pi(K-1)f_0/f_s} \quad e^{j2\pi K f_0/f_s}]$ is the vector of exponentials and

$$\mathbf{a} = [a_{-K} \quad a_{-K+1} \quad \dots \quad a_{K-1} \quad a_K]^T \quad (2.5)$$

is the vector with the complex amplitudes (in our case the unknown parameter). With the symbolism \mathbf{A}^H denoting, henceforth, the Hermitian (conjugate) transpose of an array \mathbf{A} . We also mark that the symbolism $(\mathbf{E}_{Kh})^n$ indicates that every element of \mathbf{E}_{Kh} is to be raised to the power of n . Additionally, where the matrix \mathbf{E}_{0h} with dimensions

$(2N + 1) \times (2K + 1)$ is given by:

$$\mathbf{E}_{0h} = \begin{bmatrix} (\mathbf{E}_{Kh})^{-N} \\ (\mathbf{E}_{Kh})^{-N+1} \\ \vdots \\ (\mathbf{E}_{Kh})^{N-1} \\ (\mathbf{E}_{Kh})^N \end{bmatrix} = \begin{bmatrix} e^{j2\pi(-N)(-K)f_0/f_s} & e^{j2\pi(-N)(-K+1)f_0/f_s} & \dots & e^{j2\pi(-N)(K-1)f_0/f_s} & e^{j2\pi(-N)Kf_0/f_s} \\ e^{j2\pi(-N+1)(-K)f_0/f_s} & e^{j2\pi(-N+1)(-K+1)f_0/f_s} & \dots & e^{j2\pi(-N+1)(K-1)f_0/f_s} & e^{j2\pi(-N+1)Kf_0/f_s} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e^{j2\pi(N-1)(-K)f_0/f_s} & e^{j2\pi(N-1)(-K+1)f_0/f_s} & \dots & e^{j2\pi(N-1)(K-1)f_0/f_s} & e^{j2\pi(N-1)Kf_0/f_s} \\ e^{j2\pi N(-K)f_0/f_s} & e^{j2\pi N(-K+1)f_0/f_s} & \dots & e^{j2\pi N(K-1)f_0/f_s} & e^{j2\pi NKf_0/f_s} \end{bmatrix}. \quad (2.6)$$

Thus, the elements of \mathbf{E}_{0h} can be re-written as:

$$(\mathbf{E}_{0h})_{nk} = e^{j2\pi nk f_0/f_s}, \quad (2.7)$$

Assuming obviously, that $n = -N, \dots, N$ indicates the $(2N + 1)$ rows of the matrix, and $k = -K, \dots, K$ correspond to the $(2K + 1)$ columns.

From matrix calculus, we have:

$$\begin{aligned} \frac{\partial \epsilon_{\mathbf{a}}}{\partial \mathbf{a}} &= -\mathbf{E}_{0h}^H (\mathbf{s} - \mathbf{E}_{0h} \mathbf{a}) \\ &= \mathbf{E}_{0h}^H \mathbf{E}_{0h} \mathbf{a} - \mathbf{E}_{0h}^H \mathbf{s} \end{aligned}$$

To proceed, the derivative $\frac{\partial \epsilon_{\mathbf{a}}}{\partial \mathbf{a}}$ is set equal to zero and the existence of $(\mathbf{E}_{0h}^H \mathbf{E}_{0h})^{-1}$ is assumed, or equivalently, that \mathbf{E}_{0h} has full column rank. Then, it can be proved that the minimization of the error $\epsilon_{\mathbf{a}}$ with respect to \mathbf{a} results in the following solution¹:

$$\mathbf{a} = (\mathbf{E}_{0h}^H \mathbf{E}_{0h})^{-1} \mathbf{E}_{0h}^H \mathbf{s}, \quad (2.8)$$

In addition, if a proper window is used, then the signal, the least squares error and the solution take the following forms:

$$\mathbf{W}\mathbf{s} = [w[-N]s[-N] \quad w[-N+1]s[-N+1] \quad \dots \quad w[N-1]s[N-1] \quad w[N]s[N]]^T, \quad (2.9)$$

$$\epsilon_{\mathbf{a}} = \sum_{n=-N}^N w^2[n] \left(s[n] - \sum_{k=-K}^K a_k e^{j2\pi n f_k/f_s} \right)^2, \quad (2.10)$$

¹ The matrix $((\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H)$ is known as the Moore-Penrose generalized inverse (a special type of pseudo-inverse) of \mathbf{A} , it is often denoted by \mathbf{A}^+ and traditionally appears when solving Least Squares problems such as the one at hand.

$$\begin{aligned}\mathbf{a} &= (\mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{0h})^{-1} \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{s} \\ &= \mathbf{R}_{0h}^{-1} \mathbf{s}_{0h} ,\end{aligned}\tag{2.11}$$

where

$$\mathbf{R}_{0h} = \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{0h} ,\tag{2.12}$$

$$\mathbf{s}_{0h} = \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{s} ,\tag{2.13}$$

$w[n]$ is a proper window of length $2N + 1$ and \mathbf{W} is a diagonal matrix of size $(2N + 1) \times (2N + 1)$, with its elements being the values of the analysis window $w[n]$:

$$\begin{aligned}\mathbf{W} &= \begin{bmatrix} w[-N] & 0 & \dots & 0 & 0 \\ 0 & w[-N + 1] & \dots & 0 & 0 \\ \dots & \dots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & w[N - 1] & 0 \\ 0 & 0 & \dots & 0 & w[N] \end{bmatrix} \\ &= \text{diag} (w[-N], w[-N + 1], \dots, w[N - 1], w[N]) .\end{aligned}\tag{2.14}$$

It can be easily proved that \mathbf{R}_{0h} is a Hermitian matrix, that is \mathbf{R}_{0h} is equal to its own conjugate transpose ($\mathbf{R}_{0h} = \mathbf{R}_{0h}^H$). In Chapter 3, it will be established that it also has Toeplitz form, in other words, is a matrix of which each diagonal is a constant. One of the properties of the Hermitian Toeplitz matrices is that the matrix can be uniquely represented by a single row (customarily the first one is used).

Thus, by utilizing an LS method, the minimization of the sum of the squared and windowed difference, between the original signal \mathbf{s} and the sinusoidal model approximation of the harmonic part \mathbf{h}_0 , with respect to \mathbf{a} , has been achieved. The computational complexity of this computation is at the order of $\mathcal{O}(K^2(N + K))^2$, which is quite costly when compared with FFT-based algorithms, of which the computational cost is at the order of $\mathcal{O}(N \log(N))$.

² The term $\mathcal{O}(K^3)$ is presented by the inversion of \mathbf{R}_{0h} and the term $\mathcal{O}(K^2N)$ appears when performing the $\mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{0h}$ multiplications, considering the fact that \mathbf{W} is a diagonal matrix.

2.2 Sinusoidal Model

As with the HM, we start by decomposing \mathbf{s} into a harmonic and noisy a part denoted by $h_1[n]$ and $w_1[n]$ respectively:

$$s[n] = h_1[n] + w_1[n] . \quad (2.15)$$

By removing the frequency restrictions, and by assuming a frame of two pitch periods, centered around zero, the sinusoidal modeling states that the harmonic part is modeled as follows:

$$h_1[n] = \sum_{k=-K}^K a_k e^{j2\pi n f_k / f_s}, \quad n = -N, \dots, N, \quad (2.16)$$

where f_k are the frequencies in Hertz of the k th component. Thus, the substitution of (2.16) to (2.15) yields the following expression for the signal:

$$s[n] = \sum_{k=-K}^K a_k e^{j2\pi n f_k / f_s} + w_1[n], \quad n = -N, \dots, N . \quad (2.17)$$

Our goal is to estimate of the complex amplitudes a_k given the number of components K and the frequencies f_k . Following a Least Squares approach we minimize the square of the error, which is given by:

$$\begin{aligned} \epsilon_{\mathbf{a}} &= \sum_{n=-N}^N (s[n] - h_1[n])^2 \\ &= \sum_{n=-N}^N \left(s[n] - \sum_{k=-K}^K a_k e^{j2\pi n f_k / f_s} \right)^2 \\ &= \sum_{n=-N}^N (s[n] - (\mathbf{E}_K)^n \mathbf{a})^2, \end{aligned} \quad (2.18)$$

where $\mathbf{E}_K = [e^{j2\pi f_{-K}/f_s} \quad e^{j2\pi f_{(-K+1)}/f_s} \quad \dots \quad e^{j2\pi f_{K-1}/f_s} \quad e^{j2\pi f_K/f_s}]$. We also note that $(\mathbf{E}_K)^n$ denotes that each element of \mathbf{E}_K is raised to the power of n .

The minimization of $\epsilon_{\mathbf{a}}$ with respect to \mathbf{a} results in the following solution, assuming that the inverse of $\mathbf{E}_0^H \mathbf{E}_0$ exists:

$$\mathbf{a} = (\mathbf{E}_0^H \mathbf{E}_0)^{-1} \mathbf{E}_0^H \mathbf{s}, \quad (2.19)$$

where \mathbf{E}_0 is a $(2N + 1) \times (2K + 1)$ matrix, which is given by:

$$\begin{aligned} \mathbf{E}_0 &= \begin{bmatrix} (\mathbf{E}_K)^{-N} \\ (\mathbf{E}_K)^{-N+1} \\ \vdots \\ (\mathbf{E}_K)^{N-1} \\ (\mathbf{E}_K)^N \end{bmatrix} \\ &= \begin{bmatrix} e^{j2\pi-Nf_{-K}/f_s} & e^{j2\pi-Nf_{(-K+1)}/f_s} & \dots & e^{j2\pi-Nf_{K-1}/f_s} & e^{j2\pi-Nf_K/f_s} \\ e^{j2\pi(-N+1)f_{-K}/f_s} & e^{j2\pi(-N+1)f_{(-K+1)}/f_s} & \dots & e^{j2\pi(-N+1)f_{K-1}/f_s} & e^{j2\pi(-N+1)f_K/f_s} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e^{j2\pi(N-1)f_{-K}/f_s} & e^{j2\pi(N-1)f_{(-K+1)}/f_s} & \dots & e^{j2\pi(N-1)f_{K-1}/f_s} & e^{j2\pi(N-1)f_K/f_s} \\ e^{j2\pi Nf_{-K}/f_s} & e^{j2\pi Nf_{(-K+1)}/f_s} & \dots & e^{j2\pi Nf_{K-1}/f_s} & e^{j2\pi Nf_K/f_s} \end{bmatrix} \end{aligned} \quad (2.20)$$

In a more compact form, the elements of \mathbf{E}_0 are given by:

$$(\mathbf{E}_0)_{nk} = e^{j2\pi n f_k / f_s} , \quad (2.21)$$

for $n = -N, \dots, N$, indicating the $(2N + 1)$ rows of \mathbf{E}_0 and for $k = -K, \dots, K$ denoting the $(2K + 1)$ columns.

If we employ an appropriate window then the signal, the least squares error and the solution take the following forms:

$$\mathbf{W}\mathbf{s} = [w[-N]s[-N] \quad w[-N+1]s[-N+1] \quad \dots \quad w[N]s[N]]^T , \quad (2.22)$$

$$\epsilon_{\mathbf{a}} = \sum_{n=-N}^N w^2[n] \left(s[n] - \sum_{k=-K}^K a_k e^{j2\pi n f_k / f_s} \right)^2 , \quad (2.23)$$

$$\begin{aligned} \mathbf{a} &= (\mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_0)^{-1} \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{s} \\ &= \mathbf{R}_0^{-1} \mathbf{s}_0 , \end{aligned} \quad (2.24)$$

where

$$\mathbf{R}_0 = \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_0 \quad (2.25)$$

and

$$\mathbf{s}_0 = \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{s} . \quad (2.26)$$

It can be illustrated that \mathbf{R}_0 is Hermitian ($\mathbf{R}_0^H = \mathbf{R}_0$), a property used later for the accelerating steps.

Using an LS method, the minimization of the sum of the squared and windowed difference between the original signal \mathbf{s} , and the sinusoidal model approximation of the harmonic part \mathbf{h}_1 , with respect to \mathbf{a} , has a computational complexity at the order of $\mathcal{O}(K^2(N+K))$, which is higher when compared to the FFT-based algorithms, whose cost

is at the order of $\mathcal{O}(N \log(N))$. The complexity of this computation is $\mathcal{O}(K^2(N + K))^3$ and it is quite costly compared with FFT algorithms which cost $\mathcal{O}(N \log(N))$.

³ The term $\mathcal{O}(K^3)$ is presented when the inversion of \mathbf{R}_0 is taken into account and the term $\mathcal{O}(K^2N)$ derives from the $\mathbf{E}_0^H \mathbf{W} \mathbf{W} \mathbf{E}_0$ multiplication, which yields \mathbf{R}_0 , since \mathbf{W} is a diagonal matrix.

2.3 Quasi-Harmonic Model

The third approach for modeling the harmonic part, the so-called Quasi-Harmonic Model [11], which has an additional time-varying amplitude term for each component, is given by:

$$h_2[n] = \sum_{k=-K}^K (a_k + nb_k) e^{j2\pi nk f_0/f_s}, \quad n = -N, \dots, N, \quad (2.27)$$

where b_k denotes the complex slope for the k th component.

Working as before, we decompose the original signal \mathbf{s} into a harmonic and a noisy part denoted by $h_2[n]$ and $w_2[n]$, respectively, as follows:

$$\begin{aligned} s[n] &= h_2[n] + w_2[n] \\ &= \sum_{k=-K}^K (a_k + nb_k) e^{j2\pi nk f_0/f_s} + w_2[n]. \end{aligned} \quad (2.28)$$

In this case the LS error takes the following form:

$$\begin{aligned} \epsilon_{\mathbf{x}} &= \sum_{n=-N}^N (s[n] - h_2[n])^2 \\ &= \sum_{n=-N}^N \left(s[n] - \sum_{k=-K}^K (a_k + nb_k) e^{j2\pi nk f_0/f_s} \right)^2 \\ &= \sum_{n=-N}^N (s[n] - [(\mathbf{E}_{Kh})^n \quad n(\mathbf{E}_{Kh})^n] \mathbf{x})^2, \end{aligned} \quad (2.29)$$

where $\mathbf{E}_{Kh} = [e^{j2\pi(-K)f_0/f_s} \quad e^{j2\pi(-K+1)f_0/f_s} \quad \dots \quad e^{j2\pi(K-1)f_0/f_s} \quad e^{j2\pi K f_0/f_s}]$, as illustrated in HM and

$$\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = [a_{-K}, a_{-K+1}, \dots, a_{K-1}, a_K, b_{-K}, b_{-K+1}, \dots, b_{K-1}, b_K]^T, \quad (2.30)$$

with, $\mathbf{b} = [b_{-K}, \dots, b_K]^T$ denoting the vector which consists of the complex slopes. Working similarly as in the previous model, the error can be rewritten in the following matrix form:

$$\epsilon_{\mathbf{x}} = (\mathbf{s} - [\mathbf{E}_{0h} | \mathbf{E}_{1h}] \mathbf{x})^H (\mathbf{s} - [\mathbf{E}_{0h} | \mathbf{E}_{1h}] \mathbf{x}), \quad (2.31)$$

where \mathbf{E}_{1h} is a $(2N + 1) \times (2K + 1)$ matrix with its elements being exponentials multiplied

with time n . More specifically,

$$\begin{aligned}
 \mathbf{E}_{1h} &= \mathbf{N}_d \mathbf{E}_{0h} \\
 &= \begin{bmatrix} -N(\mathbf{E}_{Kh})^{-N} \\ (-N+1)(\mathbf{E}_{Kh})^{-N+1} \\ \vdots \\ (N-1)(\mathbf{E}_{Kh})^{N-1} \\ N(\mathbf{E}_{Kh})^N \end{bmatrix} \\
 &= \begin{bmatrix} -Ne^{j2\pi(-N)(-K)f_0/f_s} & -Ne^{j2\pi(-N)(-K+1)f_0/f_s} & \dots & -Ne^{j2\pi(-N)Kf_0/f_s} \\ (-N+1)e^{j2\pi(-N+1)(-K)f_0/f_s} & (-N+1)e^{j2\pi(-N+1)(-K+1)f_0/f_s} & \dots & (-N+1)e^{j2\pi(-N+1)Kf_0/f_s} \\ \vdots & \vdots & \ddots & \vdots \\ (N-1)e^{j2\pi(N-1)(-K)f_0/f_s} & (N-1)e^{j2\pi(N-1)(-K+1)f_0/f_s} & \dots & (N-1)e^{j2\pi(N-1)Kf_0/f_s} \\ Ne^{j2\pi N(-K)f_0/f_s} & Ne^{j2\pi N(-K+1)f_0/f_s} & \dots & Ne^{j2\pi NKf_0/f_s} \end{bmatrix},
 \end{aligned}$$

where \mathbf{N}_d is a $(2N+1) \times (2N+1)$ diagonal array given by:

$$\begin{aligned}
 \mathbf{N}_d &= \begin{bmatrix} -N & 0 & \dots & 0 & 0 \\ 0 & -N+1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & N-1 & 0 \\ 0 & 0 & \dots & 0 & N \end{bmatrix} \\
 &= \text{diag}(-N, -N+1, \dots, N-1, N).
 \end{aligned} \tag{2.32}$$

Thus, the elements of \mathbf{E}_{1h} can be re-written as: $(\mathbf{E}_{1h})_{nk} = ne^{j2\pi n f_k / f_s} = n(\mathbf{E}_{0h})_{nk}$.

By zeroing the derivative of the error with respect to \mathbf{x} , $\frac{\partial \epsilon_{\mathbf{x}}}{\partial \mathbf{x}} = 0$ and assuming the existence of the inverse $([\mathbf{E}_{0h} | \mathbf{E}_{1h}]^H [\mathbf{E}_{0h} | \mathbf{E}_{1h}])^{-1}$, we obtain the solution :

$$\begin{aligned}
 \mathbf{x} &= ([\mathbf{E}_{0h} | \mathbf{E}_{1h}]^H [\mathbf{E}_{0h} | \mathbf{E}_{1h}])^{-1} [\mathbf{E}_{0h} | \mathbf{E}_{1h}]^H \mathbf{s} \\
 &= (\mathbf{E}_{\text{QHM}}^H \mathbf{E}_{\text{QHM}})^{-1} \mathbf{E}_{\text{QHM}}^H \mathbf{s},
 \end{aligned} \tag{2.33}$$

where

$$\mathbf{E}_{\text{QHM}} = [\mathbf{E}_{0h} | \mathbf{E}_{1h}], \tag{2.34}$$

note that the symbol $|$ that appears above in $[\mathbf{E}_{0h} | \mathbf{E}_{1h}]$, represents the horizontal matrix concatenation of \mathbf{E}_{0h} and \mathbf{E}_{1h} .

If we apply a suitable window, then the solution takes the following form:

$$\begin{aligned}
 \mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} &= (\mathbf{E}_{\text{QHM}}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{\text{QHM}})^{-1} \mathbf{E}_{\text{QHM}}^H \mathbf{W}^H \mathbf{W} \mathbf{s} \\
 &= \begin{pmatrix} \mathbf{R}_{0h} & \mathbf{R}_{1h} \\ \mathbf{R}_{1h}^H & \mathbf{R}_{2h} \end{pmatrix}^{-1} \begin{bmatrix} \mathbf{s}_{0h} \\ \mathbf{s}_{1h} \end{bmatrix} \\
 &= \mathbf{R}_{\text{QHM}}^{-1} \mathbf{y}_{\text{QHM}},
 \end{aligned} \tag{2.35}$$

where the quantities \mathbf{a} , \mathbf{b} , \mathbf{W} , \mathbf{E}_{0h} , \mathbf{E}_{1h} , \mathbf{R}_{0h} , and \mathbf{s}_{0h} are given before. While,

$$\mathbf{R}_{\text{QHM}} = \begin{pmatrix} \mathbf{R}_{0h} & \mathbf{R}_{1h} \\ \mathbf{R}_{1h}^H & \mathbf{R}_{2h} \end{pmatrix}, \quad (2.36)$$

$$\mathbf{y}_{\text{QHM}} = \begin{bmatrix} \mathbf{s}_{0h} \\ \mathbf{s}_{1h} \end{bmatrix}, \quad (2.37)$$

moreover,

$$\mathbf{R}_{1h} = \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{1h}, \quad (2.38)$$

$$\mathbf{R}_{2h} = \mathbf{E}_{1h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{1h}, \quad (2.39)$$

and $\mathbf{s}_{1h} = \mathbf{E}_{1h}^H \mathbf{W}^H \mathbf{W} \mathbf{s}$. Like \mathbf{R}_{0h} it can be illustrated that \mathbf{R}_{1h} and \mathbf{R}_{2h} are both Hermitian matrices. In Chapter 3 it will be established that they also have Toeplitz form. Thus, they too can be uniquely described by one of their rows.

In this case, there is no efficient FFT-based algorithm which is able to compute the parameters of the model and thus, the LS approach is the only option.

2.4 Generalized Quasi-Harmonic Model

The final model for the harmonic part, namely the *Generalized Quasi-Harmonic Model* (GQHM), has an additional term for each component, like QHM, but without its feasible frequency values limitation. A GQHM is given by:

$$h_3[n] = \sum_{k=-K}^K (a_k + nb_k) e^{j2\pi n f_k / f_s}, \quad n = -N, \dots, N. \quad (2.40)$$

Following the previous approach by decomposing the original signal \mathbf{s} into a harmonic and a noisy part, denoted by $h_3[n]$ and $w_3[n]$, respectively, the signal model can be expressed as follows:

$$\begin{aligned} s[n] &= h_3[n] + w_3[n] \\ &= \sum_{k=-K}^K (a_k + nb_k) e^{j2\pi n f_k / f_s} + w_3[n]. \end{aligned} \quad (2.41)$$

The associated LS error is given by:

$$\begin{aligned} \epsilon_{\mathbf{x}} &= \sum_{n=-N}^N (s[n] - h_3[n])^2 \\ &= \sum_{n=-N}^N \left(s[n] - \sum_{k=-K}^K (a_k + nb_k) e^{j2\pi n f_k / f_s} \right)^2 \\ &= \sum_{n=-N}^N (s[n] - [(\mathbf{E}_K)^n \quad n(\mathbf{E}_K)^n] \mathbf{x})^2, \end{aligned} \quad (2.42)$$

where \mathbf{x} is defined in (2.30) and $\mathbf{E}_K = [e^{j2\pi f_{-K}/f_s} \quad e^{j2\pi f_{(-K+1)}/f_s} \quad \dots \quad e^{j2\pi f_{K-1}/f_s} \quad e^{j2\pi f_K/f_s}]$ from the Quasi-Harmonic and the Sinusoidal Model, respectively.

Working similarly to the previous models, the error can be written in a matrix form as follows:

$$\epsilon_{\mathbf{x}} = (\mathbf{s} - [\mathbf{E}_0 | \mathbf{E}_1] \mathbf{x})^H (\mathbf{s} - [\mathbf{E}_0 | \mathbf{E}_1] \mathbf{x}) \quad (2.43)$$

where \mathbf{E}_1 is a $(2N + 1) \times (2K + 1)$ matrix with its elements being the complex exponentials $(\mathbf{E}_K)^n$ multiplied with time. By employing (2.20), \mathbf{E}_1 can be expressed as follows:

$$\begin{aligned}
 \mathbf{E}_1 &= \mathbf{N}_d \mathbf{E}_0 \\
 &= \begin{bmatrix} -N(\mathbf{E}_K)^{-N} \\ (-N+1)(\mathbf{E}_K)^{-N+1} \\ \vdots \\ (N-1)(\mathbf{E}_K)^{N-1} \\ N(\mathbf{E}_K)^N \end{bmatrix} \\
 &= \begin{bmatrix} -Ne^{j2\pi-Nf_{-K}/f_s} & -Ne^{j2\pi-Nf_{(-K+1)}/f_s} & \dots & -Ne^{j2\pi-Nf_K/f_s} \\ (-N+1)e^{j2\pi(-N+1)f_{-K}/f_s} & (-N+1)e^{j2\pi(-N+1)f_{(-K+1)}/f_s} & \dots & (-N+1)e^{j2\pi(-N+1)f_K/f_s} \\ \vdots & \vdots & \ddots & \vdots \\ (N-1)e^{j2\pi(N-1)f_{-K}/f_s} & (N-1)e^{j2\pi(N-1)f_{(-K+1)}/f_s} & \dots & (N-1)e^{j2\pi(N-1)f_K/f_s} \\ Ne^{j2\pi Nf_{-K}/f_s} & Ne^{j2\pi Nf_{(-K+1)}/f_s} & \dots & Ne^{j2\pi Nf_K/f_s} \end{bmatrix}, \tag{2.44}
 \end{aligned}$$

where \mathbf{N}_d is a $(2N+1) \times (2N+1)$ diagonal matrix which is given by:

$$\begin{aligned}
 \mathbf{N}_d &= \begin{bmatrix} -N & 0 & \dots & 0 \\ 0 & -N+1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & N \end{bmatrix} \\
 &= \text{diag}([-N \quad -N+1 \quad \dots \quad N]) . \tag{2.45}
 \end{aligned}$$

The elements of \mathbf{E}_1 can be re-written in a more compact form, in relation to \mathbf{E}_0 , as follows: $(\mathbf{E}_1)_{nk} = ne^{j2\pi n f_k / f_s} = n(\mathbf{E}_0)_{nk}$.

By setting $\frac{\partial \epsilon_{\mathbf{x}}}{\partial \mathbf{x}} = 0$, we obtain the solution, which minimizes the error, assuming the existence of the inverse $([\mathbf{E}_0 | \mathbf{E}_1]^H [\mathbf{E}_0 | \mathbf{E}_1])^{-1}$:

$$\begin{aligned}
 \mathbf{x} &= ([\mathbf{E}_0 | \mathbf{E}_1]^H [\mathbf{E}_0 | \mathbf{E}_1])^{-1} [\mathbf{E}_0 | \mathbf{E}_1]^H \mathbf{s} \\
 &= (\mathbf{E}_{\text{GQHM}}^H \mathbf{E}_{\text{GQHM}})^{-1} \mathbf{E}_{\text{GQHM}}^H \mathbf{s}, \tag{2.46}
 \end{aligned}$$

where

$$\mathbf{E}_{\text{GQHM}} = [\mathbf{E}_0 | \mathbf{E}_1] \tag{2.47}$$

In addition, if we apply a window to the original signal as in (2.22), then it can be shown that the solution takes the following form:

$$\begin{aligned}
 \mathbf{x} &= \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = (\mathbf{E}_{\text{GQHM}}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{\text{GQHM}})^{-1} \mathbf{E}_{\text{GQHM}}^H \mathbf{W}^H \mathbf{W} \mathbf{s} \\
 &= \begin{pmatrix} \mathbf{R}_0 & \mathbf{R}_1 \\ \mathbf{R}_1^H & \mathbf{R}_2 \end{pmatrix}^{-1} \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \end{bmatrix} \\
 &= \mathbf{R}_{\text{GQHM}}^{-1} \mathbf{y}_{\text{GQHM}} \tag{2.48}
 \end{aligned}$$

where the terms \mathbf{a} , \mathbf{b} , \mathbf{W} , \mathbf{E}_0 , \mathbf{E}_1 , \mathbf{R}_0 , and \mathbf{s}_0 have been defined earlier, while

$$\mathbf{R}_{\text{GQHM}} = \begin{pmatrix} \mathbf{R}_0 & \mathbf{R}_1 \\ \mathbf{R}_1^H & \mathbf{R}_2 \end{pmatrix}, \tag{2.49}$$

$$\mathbf{y}_{\text{GQHM}} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \end{bmatrix}, \quad (2.50)$$

in addition,

$$\mathbf{R}_1 = \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_1, \quad (2.51)$$

$$\mathbf{R}_2 = \mathbf{E}_1^H \mathbf{W}^H \mathbf{W} \mathbf{E}_1, \quad (2.52)$$

and $\mathbf{s}_1 = \mathbf{E}_1^H \mathbf{W}^H \mathbf{W} \mathbf{s}$. Like \mathbf{R}_0 it can be easily proved that \mathbf{R}_1 and \mathbf{R}_2 are Hermitian matrices. A property that will be utilized in the speed-up steps.

As in the previous time-varying case, there is no FFT-based algorithm being able to effectively compute the parameters of the model, and thus, again an LS approach is the only alternative.

Chapter 3

Speeding Up the Computations

In this chapter, we will introduce novel techniques for decreasing the computational load of each individual model. We begin by describing a parameterization of the windows used as an aid in our work. Then, we start by presenting the hastening methods for the Harmonic Model, followed by the methods for the acceleration of the Sinusoidal Model and the Quasi-Harmonic Model, while we close with the techniques the Generalized Quasi-Harmonic.

In general, the solution for all the models presented can be summarized as follows:

$$\begin{aligned} \mathbf{x} &= \mathbf{R}^{-1}\mathbf{y} \\ &= (\mathbf{E}^H \mathbf{W}^H \mathbf{W} \mathbf{E})^{-1} \mathbf{E}^H \mathbf{W}^H \mathbf{W} \mathbf{s} , \end{aligned} \tag{3.1}$$

where \mathbf{x} , \mathbf{R} , \mathbf{y} and \mathbf{E} depend on the model. More specifically,

- For the *Harmonic Model* (HM): $\mathbf{x} = \mathbf{a}$, $\mathbf{R} = \mathbf{R}_{0h}$, $\mathbf{y} = \mathbf{s}_{0h}$ and $\mathbf{E} = \mathbf{E}_{0h}$. As presented in (2.5), (2.12), (2.13) and (2.6), respectively.
- For the *Sinusoidal Model* (SM): $\mathbf{x} = \mathbf{a}$, $\mathbf{R} = \mathbf{R}_0$, $\mathbf{y} = \mathbf{s}_0$ and $\mathbf{E} = \mathbf{E}_0$. First seen in the equations (2.5), (2.25), (2.26) and (2.20).
- For the *Quasi-Harmonic Model* (QHM): $\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$, $\mathbf{R} = \mathbf{R}_{\text{QHM}} = \begin{bmatrix} \mathbf{R}_{0h} & \mathbf{R}_{1h} \\ \mathbf{R}_{1h}^H & \mathbf{R}_{2h} \end{bmatrix}$, $\mathbf{y} = \mathbf{y}_{\text{QHM}} = \begin{bmatrix} \mathbf{s}_{0h} \\ \mathbf{s}_{1h} \end{bmatrix}$ and $\mathbf{E} = \mathbf{E}_{\text{QHM}} = [\mathbf{E}_{0h} | \mathbf{E}_{1h}]$, given by (2.30), (2.36), (2.37) and (2.34).
- For the *Generalized Quasi-Harmonic Model* (GQHM): $\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$, $\mathbf{R} = \mathbf{R}_{\text{GQHM}} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1 \\ \mathbf{R}_1^H & \mathbf{R}_2 \end{bmatrix}$, $\mathbf{y} = \mathbf{y}_{\text{GQHM}} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \end{bmatrix}$ and $\mathbf{E} = \mathbf{E}_{\text{GQHM}} = [\mathbf{E}_0 | \mathbf{E}_1]$, as they appear in the equations: (2.30), (2.49), (2.50) and (2.47).

Since our goal to decrease the computational time required to estimate \mathbf{x} , instinctively we begin by optimizing the part that acts as a bottleneck. From a computational point of view, it can be seen that the most expensive part is the computation of the elements of \mathbf{R} , the matrix to be inverted, which is defined as:

$$\mathbf{R} = \mathbf{E}^H \mathbf{W}^H \mathbf{W} \mathbf{E} . \quad (3.2)$$

It can be easily seen that the complexity for computing \mathbf{R} is at the order of $\mathcal{O}(NK^2)$, if we take into account the fact that \mathbf{W} is diagonal. In order to improve that, we will show that the elements of \mathbf{R} can be found analytically. As a result, the complexity is reduced to $\mathcal{O}(NK)$, since each element of the matrix is computed in constant time. As a second step, we accelerate the estimation of \mathbf{E} by taking advantage of the form of the trigonometric functions required to be computed. As a third and final improvement we perform approximations of the matrix \mathbf{R} in such a way that not only ease the load of its computation, but also the load of its inversion. We note that the latter acceleration should be performed while the error induced by the approximations is kept at bay.

3.1 Window Parameterization

In order to carry out certain computations such as summations that will be introduced in the next sections, we have to take under consideration what kind of window is applied to the signal \mathbf{s} . Typical windows employed are Hamming, Hann and rectangular. We consider the general class of windows, which are symmetric at the origin and they are parametrized by:

$$w_\alpha[n] = (1 - \alpha_w) + \alpha_w \cos(\pi n/N) \quad n = -N, \dots, N. \quad (3.3)$$

Table 3.1 shows the relationship between the various windows and the parameter α_w , while in Figure 3.1 we plot the windows for $N = 300$ (601 samples totally).

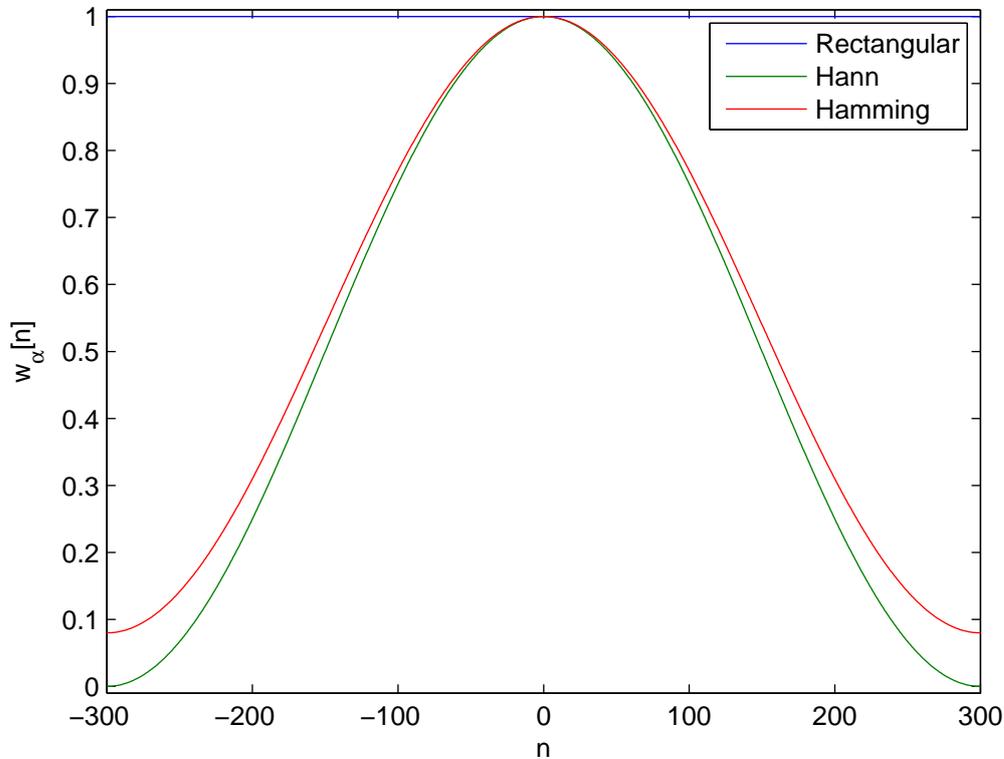


Figure 3.1: Visual representation of different windows.

$\alpha_w = 0$	Rectangular
$\alpha_w = 0.5$	Hann
$\alpha_w = 0.46$	Hamming

Table 3.1: Different values of α_w give various windows.

As the family of equations (3.2) assert, we will see that the squared windows will be

also necessary, and thus we will need the square of (3.3):

$$\begin{aligned} w_\alpha^2[n] &= ((1 - \alpha_w) + \alpha_w \cos(\pi n/N))^2 \\ &= d_0 + d_1 (e^{j\pi n/N} + e^{-j\pi n/N}) + d_2 (e^{j2\pi n/N} + e^{-j2\pi n/N}) , \end{aligned} \quad (3.4)$$

where the coefficients d_m , $m = 0, 1, 2$, are given by:

$$d_0 = (1 - \alpha_w)^2 + \alpha_w^2/2 , \quad (3.5)$$

$$d_1 = \alpha_w (1 - \alpha_w) , \quad (3.6)$$

$$d_2 = \alpha_w^2/4 . \quad (3.7)$$

In Figure 3.2 we can plot the square of each window shown in Table 3.1 for $N = 300$.

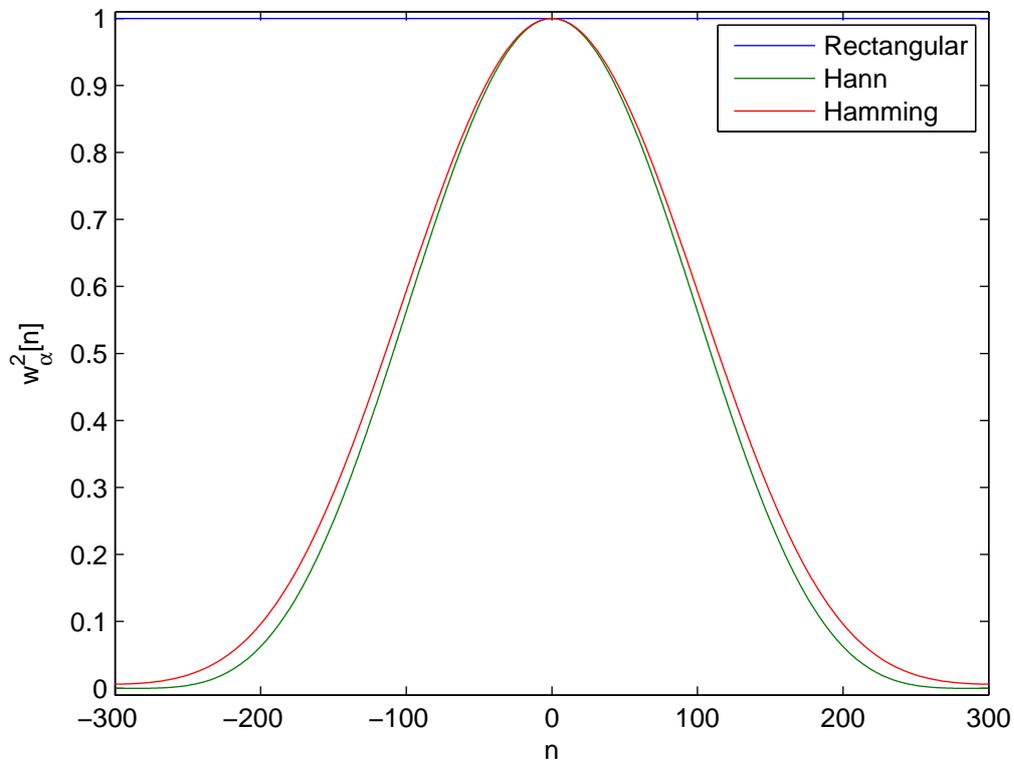


Figure 3.2: Visual representation of the square of different windows.

3.2 Harmonic Model

In this section, the methods for accelerating the computations of the Harmonic Model, which was introduced in section 2.1, will be described.

3.2.1 HM Step 1: Fast Matrix Multiplication

We start by calculating the terms of \mathbf{R}_{0h} analytically, which can be done by writing \mathbf{R}_{0h} as follows:

$$\mathbf{R}_{0h} = \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{0h} . \quad (3.8)$$

Since \mathbf{W} is a real diagonal matrix, the matrix multiplication $\mathbf{W}^H \mathbf{W}$ is simplified by taking the following form:

$$\begin{aligned} \mathbf{W}^H \mathbf{W} &= \mathbf{W}^2 \\ &= \begin{bmatrix} w^2[-N] & 0 & \dots & 0 & 0 \\ 0 & w^2[-N+1] & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & w^2[N-1] & 0 \\ 0 & 0 & \dots & 0 & w^2[N] \end{bmatrix} \\ &= \text{diag} (w^2[-N], w^2[-N+1], \dots, w^2[N-1], w^2[N]) . \end{aligned} \quad (3.9)$$

Thus, by combining (3.8) and (3.9) we can rewrite \mathbf{R}_{0h} using the following expression:

$$\begin{aligned} \mathbf{R}_{0h} &= \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{0h} \\ &= \mathbf{E}_{0h}^H \mathbf{W}^2 \mathbf{E}_{0h} \\ &= \mathbf{W}^2 \mathbf{E}_{0h}^H \mathbf{E}_{0h} . \end{aligned} \quad (3.10)$$

The elements of \mathbf{R}_{0h} are given by:

$$\begin{aligned} (\mathbf{R}_{0h})_{ik} &= \sum_{n=-N}^N w^2[n] e^{-j2\pi n i f_0 / f_s} e^{j2\pi n k f_0 / f_s} \\ &= \sum_{n=-N}^N w^2[n] e^{j2\pi n (k-i) f_0 / f_s} . \end{aligned} \quad (3.11)$$

By applying the squared window (3.4) to (3.11) the elements of \mathbf{R}_{0h} take the following form:

$$\begin{aligned}
 (\mathbf{R}_{0h})_{ik} &= d_0 \sum_{n=-N}^N \left[e^{j2\pi(k-i)f_0/f_s} \right] \\
 &+ d_1 \sum_{n=-N}^N \left[e^{j2\pi((k-i)f_0 + \frac{f_s}{2N})/f_s} \right] + d_1 \sum_{n=-N}^N \left[e^{j2\pi((k-i)f_0 - \frac{f_s}{2N})/f_s} \right] \\
 &+ d_2 \sum_{n=-N}^N \left[e^{j2\pi((k-i)f_0 + \frac{f_s}{N})/f_s} \right] + d_2 \sum_{n=-N}^N \left[e^{j2\pi((k-i)f_0 - \frac{f_s}{N})/f_s} \right],
 \end{aligned} \tag{3.12}$$

where the coefficients d_0 , d_1 and d_2 are given by (3.5), (3.6) and (3.7), respectively.

We proceed by employing a standard mathematical identity about the sum of geometric series, namely:

$$\sum_{n=-N}^N a^n = \frac{a^{N+1/2} - a^{-(N+1/2)}}{a^{1/2} - a^{-1/2}}. \tag{3.13}$$

Thus, the elements of \mathbf{R}_{0h} are given by:

$$\begin{aligned}
 (\mathbf{R}_{0h})_{ik} &= d_0 g_0(2\pi(k-i)f_0/f_s) \\
 &+ d_1 g_0\left(2\pi\left((k-i)f_0 + \frac{f_s}{2N}\right)/f_s\right) + d_1 g_0\left(2\pi\left((k-i)f_0 - \frac{f_s}{2N}\right)/f_s\right) \\
 &+ d_2 g_0\left(2\pi\left((k-i)f_0 + \frac{f_s}{N}\right)/f_s\right) + d_2 g_0\left(2\pi\left((k-i)f_0 - \frac{f_s}{N}\right)/f_s\right),
 \end{aligned} \tag{3.14}$$

where the auxiliary function $g_0(x)$ is given by¹:

$$g_0(x) = \begin{cases} \frac{\sin((2N+1)x/2)}{\sin(x/2)}, & x \neq 0 \\ 2N+1, & x = 0 \end{cases}. \tag{3.15}$$

Thus, we can calculate any element of \mathbf{R}_{0h} directly via (3.14), making the the matrix multiplication described in (3.8) obsolete, since in terms of computational load the direct computation is more efficient. Also, from (3.14) it can be easily observed that the matrix \mathbf{R}_{0H} is Toeplitz Hermitian, the values of its elements depend only from the result of $(k-i)$, or in other words, from the elements' distance from the main diagonal. Taking advantage of the Hermitian Toeplitz form of \mathbf{R}_{0h} (the knowledge of one line of the matrix results in its full form) we can form \mathbf{R}_{0h} by calculating only $(2N+1)$ of its elements instead of all $(2N+1)^2$.

¹ The function g_0 that appears here is known as the Dirichlet kernel. The importance of the Dirichlet kernel comes from its relation to Fourier series and the delta distribution. The convolution of the Dirichlet kernel with any function f of period 2π is the N th-degree Fourier series approximation.

3.2.2 HM Step 2: Faster Computation of \mathbf{E}_{0h}

In the following, we will introduce a way to accelerate the computation of \mathbf{E}_{0h} . Although the computational complexity in terms of N and K will not be altered, and thus, the same number of calculations will be required, the complexity of the calculations themselves will be reduced resulting in less computation time.

\mathbf{E}_{0h} can be written as:

$$\begin{aligned}
\mathbf{E}_{0h} &= \begin{bmatrix} (e^{j2\pi(-K)f_0/f_s})^{-N} & (e^{j2\pi(-K)f_0/f_s})^{-N+1} & \dots & (e^{j2\pi(-K)f_0/f_s})^N \\ (e^{j2\pi(-K+1)f_0/f_s})^{-N} & (e^{j2\pi(-K+1)f_0/f_s})^{-N+1} & \dots & (e^{j2\pi(-K+1)f_0/f_s})^N \\ & & \vdots & \\ (e^{j2\pi K f_0/f_s})^{-N} & (e^{j2\pi K f_0/f_s})^{-N+1} & \dots & (e^{j2\pi K f_0/f_s})^N \end{bmatrix} \\
&= \begin{bmatrix} e^{j2\pi(-N)(-K)f_0/f_s} & e^{j2\pi(-N+1)(-K)f_0/f_s} & \dots & e^{j2\pi N(-K)f_0/f_s} \\ e^{j2\pi(-N)(-K+1)f_0/f_s} & e^{j2\pi(-N+1)(-K+1)f_0/f_s} & \dots & e^{j2\pi N(-K+1)f_0/f_s} \\ & & \vdots & \\ e^{j2\pi(-N)K f_0/f_s} & e^{j2\pi(-N+1)K f_0/f_s} & \dots & e^{j2\pi N K f_0/f_s} \end{bmatrix} \quad (3.16) \\
&= \begin{bmatrix} \cos(2\pi(-N)(-K)f_0/f_s) & \dots & \cos(2\pi N(-K)f_0/f_s) \\ \cos(2\pi(-N)(-K+1)f_0/f_s) & \dots & \cos(2\pi N(-K+1)f_0/f_s) \\ & \vdots & \\ \cos(2\pi(-N)K f_0/f_s) & \dots & \cos(2\pi N K f_0/f_s) \end{bmatrix} + \\
&\quad j \begin{bmatrix} \sin(2\pi(-N)(-K)f_0/f_s) & \dots & \sin(2\pi N(-K)f_0/f_s) \\ \sin(2\pi(-N)(-K+1)f_0/f_s) & \dots & \sin(2\pi N(-K+1)f_0/f_s) \\ & \vdots & \\ \sin(2\pi(-N)K f_0/f_s) & \dots & \sin(2\pi N K f_0/f_s) \end{bmatrix}.
\end{aligned}$$

The most time-consuming part of the computation is the estimation of sines and cosines. The reduction of the computational cost stems from the fact that the solution $c_k(n)$ of the following second-order difference equation [26]:

$$c_k(n) - 2 \cos(2\pi k f_0/f_s) c_k(n-1) + c_k(n-2) = 0, \quad n = 3, 4, \dots, \quad (3.17)$$

with initial conditions

$$c_k(1) = \cos(2\pi k f_0/f_s), \quad c_k(2) = \cos(4\pi k f_0/f_s), \quad (3.18)$$

is given by

$$c_k(n) = \cos(2\pi n k f_0/f_s), \quad n = 1, 2, \dots. \quad (3.19)$$

On the other hand, by employing $c_k(1) = \sin(2\pi k f_0/f_s)$ and $c_k(2) = \sin(4\pi k f_0/f_s)$ as initial conditions, yields the solution $c_k(n) = \sin(2\pi n k f_0/f_s)$. Thus, by using the above difference equations, the computation of each trigonometric function is reduced to a multiplication.

3.2.3 HM Step 3: \mathbf{R}_{0h} Matrix Approximation

Up to now, no approximation or discretization error was performed and the LS solution has no additional error. However, if we allow for small errors, we can achieve a faster computation and inversion for the matrix \mathbf{R}_{0h} by discarding elements away from its diagonals, or in other words set them equal to zero. This approximation is valid because sinusoids which are away from each other have little or no interference.

In Figure 3.3 we plot in log scale the average (over 300 runs) magnitude of the $(\mathbf{R}_{0h})_{ik}$ elements as a function of their distance from the main diagonal (since the matrix is Toeplitz the elements of any given diagonal are equal). In the aforementioned plot it can be seen that as we move away from the main diagonal more than four matrix elements, the average magnitudes are always smaller than 10^{-4} , while the average main diagonal values are of order 10^2 . One could say that the values away from the diagonal are relatively small (when compared to the main diagonal). In other words, it can be assumed that in practice \mathbf{R}_{0h} is a strictly diagonally dominant matrix.

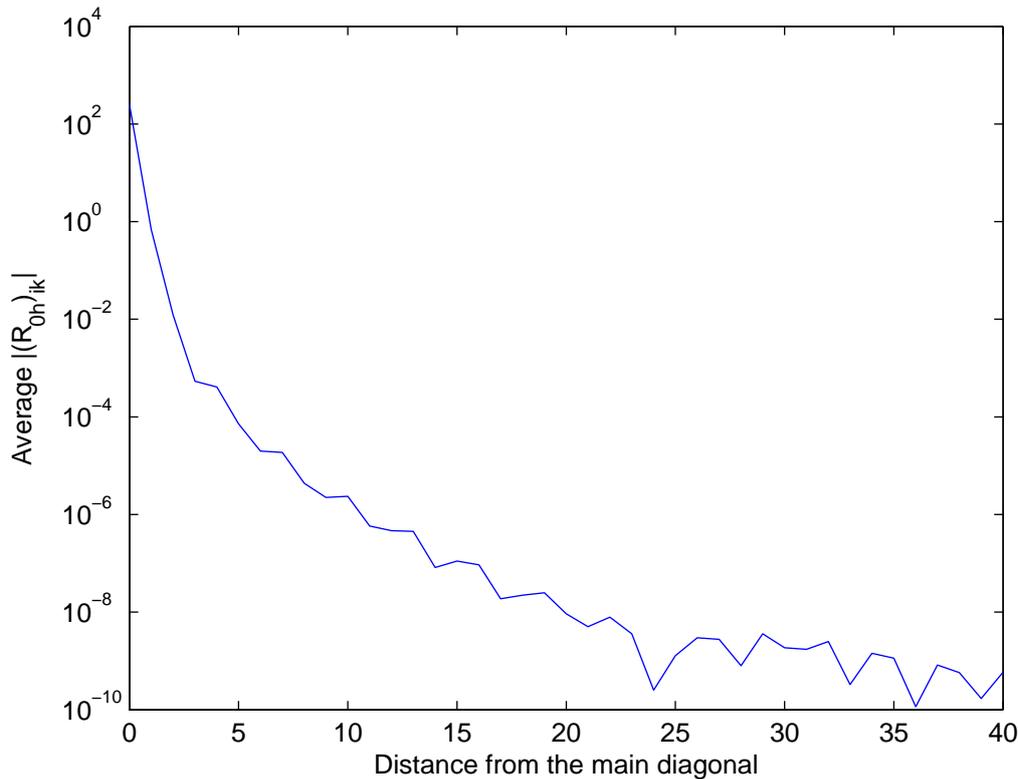


Figure 3.3: The average values of $|\mathbf{R}_{0h}|$ in log scale as a function of their distance from the main diagonal.

Besides, since \mathbf{R}_{0h} is Hermitian and strictly diagonally dominant, the Gershgorin circle theorem states that \mathbf{R}_{0h} is also non-singular. It is noteworthy that by setting the elements which are away from the main diagonal to be equal to zero preserves that beneficial property of \mathbf{R}_{0h} .

By keeping only K_0 diagonals of \mathbf{R}_{0h} , the computational cost for the inversion is expected to be decreased. In this Step we build \mathbf{R}_{0h} in a diagonal-by-diagonal way, which when combined with the fact that \mathbf{R}_{0h} is Toeplitz Hermitian allows us to find \mathbf{R}_{0h} by calculating only a single element from about half of the diagonals (one element from the main diagonal and one element for each one of the upper/lower diagonals). Obviously, even less computations are needed if we consider the fact that significantly less elements of \mathbf{R}_{0h} are actually computed. Note that if methods for storing sparse arrays are used then there is a decrease in the amount of memory utilized by the proposed algorithm too.

3.3 Sinusoidal Model

In this section, we propose methods for decreasing the computational cost of the Sinusoidal Model (introduced in section 2.2).

3.3.1 SM Step 1: Fast Matrix Multiplication

As a first step, we calculate the elements of \mathbf{R}_0 analytically. We begin by writing \mathbf{R}_0 in the following form:

$$\mathbf{R}_0 = \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_0 . \quad (3.20)$$

We substitute $\mathbf{W}^H \mathbf{W}$ given by (3.9) to (3.20) and rewrite \mathbf{R}_0 as follows:

$$\begin{aligned} \mathbf{R}_0 &= \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_0 \\ &= \mathbf{E}_0^H \mathbf{W}^2 \mathbf{E}_0 \\ &= \mathbf{W}^2 \mathbf{E}_0^H \mathbf{E}_0 . \end{aligned} \quad (3.21)$$

Then, the elements of \mathbf{R}_0 are given by the expression:

$$\begin{aligned} (\mathbf{R}_0)_{ik} &= \sum_{n=-N}^N w^2[n] e^{-j2\pi n f_i / f_s} e^{j2\pi n f_k / f_s} \\ &= \sum_{n=-N}^N w^2[n] e^{j2\pi n (f_k - f_i) / f_s} . \end{aligned} \quad (3.22)$$

By applying the squared window (3.4) to (3.22) the elements of \mathbf{R}_0 are written in an analytical way given by:

$$\begin{aligned} (\mathbf{R}_0)_{ik} &= d_0 \sum_{n=-N}^N [e^{j2\pi (f_k - f_i) / f_s}] \\ &+ d_1 \sum_{n=-N}^N [e^{j2\pi (f_k - f_i + \frac{f_s}{2N}) / f_s}] + d_1 \sum_{n=-N}^N [e^{j2\pi (f_k - f_i - \frac{f_s}{2N}) / f_s}] \\ &+ d_2 \sum_{n=-N}^N [e^{j2\pi (f_k - f_i + \frac{f_s}{N}) / f_s}] + d_2 \sum_{n=-N}^N [e^{j2\pi (f_k - f_i - \frac{f_s}{N}) / f_s}] , \end{aligned} \quad (3.23)$$

where the coefficients d_0 , d_1 and d_2 are defined by (3.5), (3.6) and (3.7), respectively.

As before, we proceed by employing a standard mathematical identity for the sum of

geometric series, seen in (3.13), which yields the following expression for \mathbf{R}_0 :

$$\begin{aligned}
(\mathbf{R}_0)_{ik} &= d_0 g_0 (2\pi (f_k - f_i) / f_s) \\
&+ d_1 g_0 \left(2\pi \left(f_k - f_i + \frac{f_s}{2N} \right) / f_s \right) + d_1 g_0 \left(2\pi \left(f_k - f_i - \frac{f_s}{2N} \right) / f_s \right) \\
&+ d_2 g_0 \left(2\pi \left(f_k - f_i + \frac{f_s}{N} \right) / f_s \right) + d_2 g_0 \left(2\pi \left(f_k - f_i - \frac{f_s}{N} \right) / f_s \right),
\end{aligned} \tag{3.24}$$

where the auxiliary function $g_0(x)$ is the one introduced in the Harmonic Model and defined by (3.15).

As with the HM, we now have a formula to calculate directly the elements of \mathbf{R}_0 and as a result we can refrain from doing the matrix multiplication in (3.20), which is more efficient. As noted earlier when introducing the SM, but also from (3.24) it can easily be deduced that the matrix \mathbf{R}_0 is Hermitian, thus, taking advantage that form of \mathbf{R}_0 we can find \mathbf{R}_0 by calculating only $(2N + 1)N$ of its elements instead of all $(2N + 1)^2$.

3.3.2 SM Step 2: Faster Computation of \mathbf{E}_0

In the second step, we propose a way to decrease the computational cost of the calculation of \mathbf{E}_0 . The computational complexity in terms of N and K will not change if after this step, however, even though the number of the calculations required does not change, the calculations required become less complex.

\mathbf{E}_0 can be written as:

$$\begin{aligned}
\mathbf{E}_0 &= \begin{bmatrix} (e^{j2\pi f_{-K}/f_s})^{-N} & (e^{j2\pi f_{-K}/f_s})^{-N+1} & \dots & (e^{j2\pi f_{-K}/f_s})^N \\ (e^{j2\pi f_{-K+1}/f_s})^{-N} & (e^{j2\pi f_{-K+1}/f_s})^{-N+1} & \dots & (e^{j2\pi f_{-K+1}/f_s})^N \\ & & \vdots & \\ (e^{j2\pi f_K/f_s})^{-N} & (e^{j2\pi f_K/f_s})^{-N+1} & \dots & (e^{j2\pi f_K/f_s})^N \end{bmatrix} \\
&= \begin{bmatrix} \cos(2\pi(-N)f_{-K}/f_s) & \dots & \cos(2\pi N f_{-K}/f_s) \\ \cos(2\pi(-N)f_{-K+1}/f_s) & \dots & \cos(2\pi N f_{-K+1}/f_s) \\ & \vdots & \\ \cos(2\pi(-N)f_K/f_s) & \dots & \cos(2\pi N f_K/f_s) \end{bmatrix} + \\
& j \begin{bmatrix} \sin(2\pi(-N)f_{-K}/f_s) & \dots & \sin(2\pi N f_{-K}/f_s) \\ \sin(2\pi(-N)f_{-K+1}/f_s) & \dots & \sin(2\pi N f_{-K+1}/f_s) \\ & \vdots & \\ \sin(2\pi(-N)f_K/f_s) & \dots & \sin(2\pi N f_K\omega_0/f_s) \end{bmatrix}.
\end{aligned} \tag{3.25}$$

From the above we can see that either way \mathbf{R}_0 is written, the most time-consuming part of the computation is the estimation of sines and cosines, or equivalently, of exponentials. The computational acceleration stems from the fact that the solution $c_k(n)$ of

the following second-order difference equation [26]:

$$c_k(n) - 2 \cos(2\pi f_k/f_s) c_k(n-1) + c_k(n-2) = 0, \quad n = 3, 4 \dots, \quad (3.26)$$

with initial conditions

$$c_k(1) = \cos(2\pi f_k/f_s), \quad c_k(2) = \cos(4\pi f_k/f_s), \quad (3.27)$$

is given by

$$c_k(n) = \cos(2\pi n f_k/f_s), \quad n = 1, 2, \dots. \quad (3.28)$$

Moreover, using $c_k(1) = \sin(2\pi f_k/f_s)$ and $c_k(2) = \sin(4\pi f_k/f_s)$ as initial conditions results in $c_k(n) = \sin(2\pi n f_k/f_s)$. Thus, using the above equations, the computation of each trigonometric function is reduced to a simple multiplication.

3.3.3 SM Step 3: \mathbf{R}_0 Matrix Approximation

In our previous calculations for the SM no approximations were performed. However, by considering small approximation errors can allow us to further decrease the computational load. Working similarly as in the HM case in Section 3.2.3, we take advantage of the fact that sinusoids which are away from each other have little interference.

The role of illustrating the effect of that realization is bestowed upon Figure 3.4. Which Figure shows the averaged (over 300 runs) values of $|(\mathbf{R}_0)_{ik}|$ in a log scale, as a function of their distance from the main diagonal. Since now the matrix is not Toeplitz, an additional average has been taken depending on how many elements with that distance exist. For instance, a $(2K+1) \times (2K+1)$ matrix has $2K+1$ elements in the main diagonal, $2 \cdot 2K$ elements with a distance of one from the main diagonal, $2 \cdot (2K-1)$ with a distance of two and so on, until we reach the two furthestmost elements whose distance from the main diagonal is $2K$. In this Figure, it can be also seen that the average magnitude of all the elements that have a distance greater than five elements from the main diagonal is lower than 10^{-4} , while the elements in the main diagonal have an average magnitude of order 10^4 . Thus, we can then safely assume that the values away from the main diagonal can be considered to negligible compared to the values closer to and on the main diagonal.

So in practice \mathbf{R}_0 has a strictly dominant main diagonal. If we also consider the fact that \mathbf{R}_0 is Hermitian then it is invertible, according to the the Gershgorin circle theorem, an important property of \mathbf{R}_0 that is kept intact if we remove elements that are away from the main diagonal.

Thus, if we keep only K_0 of the diagonals of \mathbf{R}_0 , the inversion is expected to be performed much more efficiently. In this Step we build \mathbf{R}_0 in a diagonal-by-diagonal fashion, which when combined with the fact that \mathbf{R}_0 is Hermitian it allows us to form \mathbf{R}_0 by calculating only half of the diagonals (the main diagonal and the upper/lower diagonals). Obviously, even less computations are performed if we consider the fact that significantly less elements of \mathbf{R}_0 are actually computed. Note that, if methods for storing sparse arrays are used then there is a decrease in the amount of memory utilized by the algorithm.

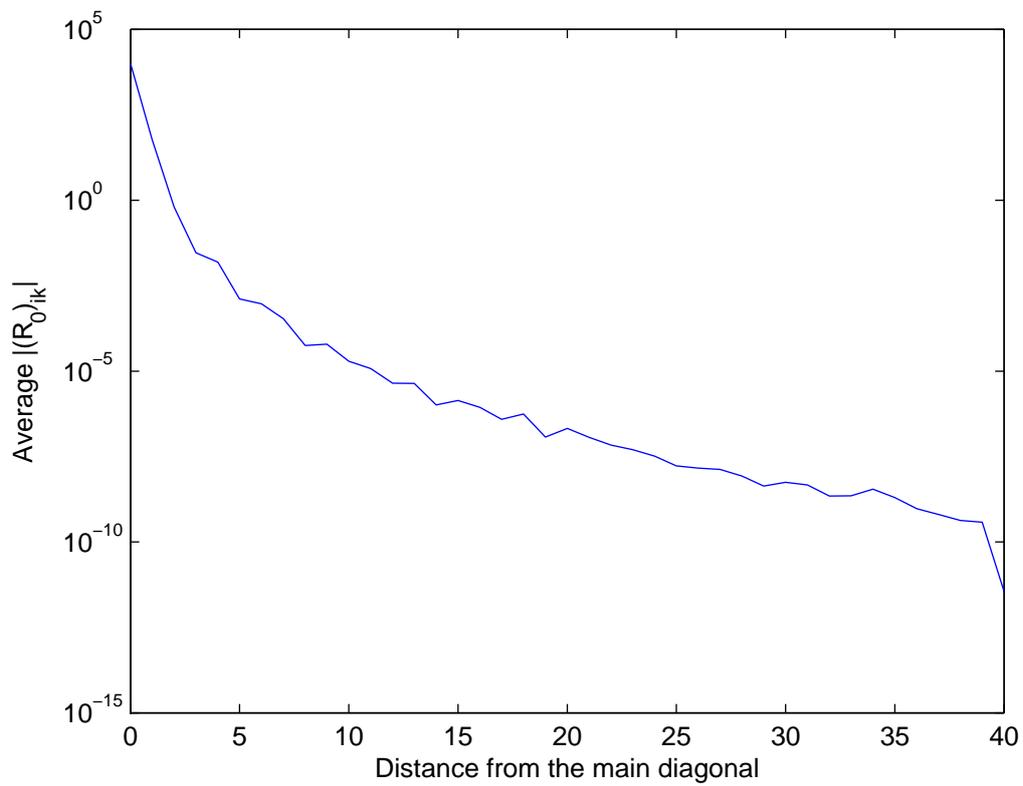


Figure 3.4: The average values of $|\mathbf{R}_0|$ in log scale as a function of their distance from the main diagonal.

3.4 Quasi-Harmonic Model

In this section, we will describe the process implemented to accelerate the computation of the Quasi-Harmonic Model coefficients (introduced in section 2.3).

3.4.1 QHM Step 1: Fast Matrix Multiplication

We start by calculating the elements of $\mathbf{R} = \mathbf{E}^H \mathbf{W}^H \mathbf{W} \mathbf{E}$ analytically. Considering the QHM case (the time-varying model) we have:

$$[\mathbf{E}_{0h} | \mathbf{E}_{1h}]^H \mathbf{W}^H \mathbf{W} [\mathbf{E}_{0h} | \mathbf{E}_{1h}] = \begin{bmatrix} \mathbf{R}_{0h} & \mathbf{R}_{1h} \\ \mathbf{R}_{1h}^H & \mathbf{R}_{2h} \end{bmatrix}, \quad (3.29)$$

where the sub-matrices \mathbf{R}_{0h} , \mathbf{R}_{1h} and \mathbf{R}_{2h} are given by:

$$\mathbf{R}_{0h} = \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{0h}, \quad (3.30)$$

$$\mathbf{R}_{1h} = \mathbf{E}_{0h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{1h}, \quad (3.31)$$

$$\mathbf{R}_{2h} = \mathbf{E}_{1h}^H \mathbf{W}^H \mathbf{W} \mathbf{E}_{1h}. \quad (3.32)$$

As far as (3.30) is concerned we have shown in section 3.2.1 that the matrix multiplication can be surpassed by calculating the elements of \mathbf{R}_{0h} directly via (3.14). Our purpose is to find similar expressions that will allow us to compute immediately the elements of \mathbf{R}_{1h} and \mathbf{R}_{2h} as well.

Working in similar fashion, we substitute (3.9) the squared window $\mathbf{W}^H \mathbf{W}$, to (3.31) and (3.32) and rewrite \mathbf{R}_{1h} and \mathbf{R}_{2h} as follows:

$$\mathbf{R}_{1h} = \mathbf{W}^2 \mathbf{E}_{0h}^H \mathbf{E}_{1h}, \quad (3.33)$$

$$\mathbf{R}_{2h} = \mathbf{W}^2 \mathbf{E}_{1h}^H \mathbf{E}_{1h}. \quad (3.34)$$

Doing this, the elements of \mathbf{R}_{1h} and \mathbf{R}_{2h} are given by:

$$\begin{aligned} (\mathbf{R}_{1h})_{ik} &= \sum_{n=-N}^N w^2[n] e^{-j2\pi n i f_0 / f_s} n e^{j2\pi n k f_0 / f_s} \\ &= \sum_{n=-N}^N w^2[n] n e^{j2\pi n (k-i) f_0 / f_s}, \end{aligned} \quad (3.35)$$

$$\begin{aligned} (\mathbf{R}_{2h})_{ik} &= \sum_{n=-N}^N w^2[n] n e^{-j2\pi n i f_0 / f_s} n e^{j2\pi n k f_0 / f_s} \\ &= \sum_{n=-N}^N w^2[n] n^2 e^{j2\pi n (k-i) f_0 / f_s}. \end{aligned} \quad (3.36)$$

By applying a parameterized squared window, as defined by equation (3.4) to (3.35) and (3.36) we obtain the following expressions:

$$\begin{aligned}
 (\mathbf{R}_{1h})_{ik} &= d_0 \sum_{n=-N}^N n \left[e^{j2\pi(k-i)f_0/f_s} \right] \\
 &+ d_1 \sum_{n=-N}^N n \left[e^{j2\pi((k-i)f_0 + \frac{f_s}{2N})/f_s} \right] + d_1 \sum_{n=-N}^N n \left[e^{j2\pi((k-i)f_0 - \frac{f_s}{2N})/f_s} \right] \\
 &+ d_2 \sum_{n=-N}^N n \left[e^{j2\pi((k-i)f_0 + \frac{f_s}{N})/f_s} \right] + d_2 \sum_{n=-N}^N n \left[e^{j2\pi((k-i)f_0 - \frac{f_s}{N})/f_s} \right],
 \end{aligned} \tag{3.37}$$

$$\begin{aligned}
 (\mathbf{R}_{2h})_{ik} &= d_0 \sum_{n=-N}^N n^2 \left[e^{j2\pi(k-i)f_0/f_s} \right] \\
 &+ d_1 \sum_{n=-N}^N n^2 \left[e^{j2\pi((k-i)f_0 + \frac{f_s}{2N})/f_s} \right] + d_1 \sum_{n=-N}^N n^2 \left[e^{j2\pi((k-i)f_0 - \frac{f_s}{2N})/f_s} \right] \\
 &+ d_2 \sum_{n=-N}^N n^2 \left[e^{j2\pi((k-i)f_0 + \frac{f_s}{N})/f_s} \right] + d_2 \sum_{n=-N}^N n^2 \left[e^{j2\pi((k-i)f_0 - \frac{f_s}{N})/f_s} \right].
 \end{aligned} \tag{3.38}$$

where the coefficients d_0 , d_1 and d_2 are given by equations (3.5), (3.6) and (3.7), respectively.

By employing the standard mathematical identity for the sum of geometric series, it can be proved that:

$$\sum_{n=-N}^N na^n = -\frac{a^N - a^{-N}}{(a^{1/2} - a^{-1/2})^2} + N \frac{a^{N+1/2} + a^{-(N+1/2)}}{a^{1/2} - a^{-1/2}}. \tag{3.39}$$

Using the above equation, the elements of \mathbf{R}_{1h} show up, thus they can be computed without performing the summation. Similarly, it can be proved:

$$\sum_{n=-N}^N n^2 a^n = N^2 \frac{a^{N+1} + a^{-(N+1)}}{(a^{1/2} - a^{-1/2})^2} - (N+1)^2 \frac{a^N + a^{-N}}{(a^{1/2} - a^{-1/2})^2} + 2 \frac{a^{N+1/2} - a^{-(N+1/2)}}{(a^{1/2} - a^{-1/2})^3} \tag{3.40}$$

Then, the elements of \mathbf{R}_{2h} are computed. Thus, it can be written in a more compact

form, that the elements of the sub-matrices \mathbf{R}_{mh} , $m = 0, 1, 2$ are given by²:

$$\begin{aligned} (\mathbf{R}_{mh})_{ik} &= d_0 g_m (2\pi (k-i) f_0 / f_s) \\ &+ d_1 g_m \left(2\pi \left((k-i) f_0 + \frac{f_s}{2N} \right) / f_s \right) + d_1 g_m \left(2\pi \left((k-i) f_0 - \frac{f_s}{2N} \right) / f_s \right) \\ &+ d_2 g_m \left(2\pi \left((k-i) f_0 + \frac{f_s}{N} \right) / f_s \right) + d_2 g_m \left(2\pi \left((k-i) f_0 - \frac{f_s}{N} \right) / f_s \right) , \end{aligned} \quad (3.41)$$

where $m = 0, 1, 2$ and the auxiliary function $g_0(x)$ is given by (3.15). While, the remaining auxiliary functions $g_1(x)$ and $g_2(x)$ are defined as follows:

$$g_1(x) = \begin{cases} j \frac{\sin(Nx)}{2\sin^2(x/2)} - jN \frac{\cos((2N+1)x/2)}{\sin(x/2)}, & x \neq 0 \\ 0, & x = 0 \end{cases}, \quad (3.42)$$

$$g_2(x) = \begin{cases} \frac{N^2 \cos((N+1)x) + (N+1)^2 \cos(Nx)}{2\sin^2(x/2)} - \frac{\sin((2N+1)x/2)}{2\sin^3(x/2)}, & x \neq 0 \\ N(N+1)(2N+1)/3, & x = 0 \end{cases}. \quad (3.43)$$

Finally, due to the fact that the computations of trigonometric functions are expensive, (both sines and cosines are required to be calculated) the computation of (3.41) can be accelerated by considering the following identities [28]:

$$\begin{aligned} \cos(\theta \pm \delta) &= \cos(\theta) - [\alpha \cos(\theta) \pm \beta \sin(\theta)] , \\ \sin(\theta \pm \delta) &= \sin(\theta) - [\alpha \sin(\theta) \mp \beta \cos(\theta)] , \end{aligned} \quad (3.44)$$

where α and β are precomputed coefficients defined in terms of δ ,

$$\begin{aligned} \alpha &= 2 \sin^2(\delta/2) , \\ \beta &= \sin(\delta) . \end{aligned} \quad (3.45)$$

Thus, the sines and cosines of only one of the five terms in (3.41) is required, while the remaining terms are computed using the above formulas.

3.4.2 QHM Step 2: Faster Computation of \mathbf{E}_{0h} and \mathbf{E}_{1h}

As it has been more thoroughly shown in section 3.2.2, we have already introduced a faster way of computing \mathbf{E}_{0h} by rewriting it as a sum of sines and cosines. That enables us to estimate quickly the matrix elements using the solution of a difference equation. Thus, instead of computing exponentials or trigonometric functions (except of course the initial conditions of the difference equations), the computation of \mathbf{E}_{0h} elements has been replaced by (faster to calculate) multiplications. Obviously, having computed \mathbf{E}_{0h} , the elements of \mathbf{E}_{1h} are given by $(\mathbf{E}_{1h})_{nk} = n(\mathbf{E}_{0h})_{nk}$ in $\mathcal{O}(NK)$ multiplications (or more precisely in $(2K+1)(2N+1)$ multiplications).

² An alternative way to find the elements of \mathbf{R}_{1h} and \mathbf{R}_{2h} is to set $\omega_{ik} = 2\pi n(k-i)f_0/f_s$. Then, we can see that $\frac{\partial(\mathbf{R}_{0h})_{ik}}{\partial\omega_{ik}} = nj(\mathbf{R}_{0h})_{ik} = j(\mathbf{R}_{1h})_{ik}$. Similarly, for the derivative of \mathbf{R}_{1h} : $\frac{\partial(\mathbf{R}_{1h})_{ik}}{\partial\omega_{ik}} = nj(\mathbf{R}_{1h})_{ik} = j(\mathbf{R}_{2h})_{ik}$. Thus, the respective relations can be used to find g_1 and g_2 by employing the Dirichlet kernel g_0 : $\frac{\partial g_0}{\partial\omega_{ki}} = jg_1$ and $\frac{\partial g_1}{\partial\omega_{ki}} = jg_2$. Thus g_1 is the second derivative of the Dirichlet kernel multiplied by j and $-g_2$ its second derivative.

3.4.3 QHM Step 3: \mathbf{R}_{QHM} Matrix Approximation

During the previous steps no approximation errors were introduced. The act of allowing small errors to the results enables us to accelerate even further the process of finding the amplitude coefficients for the Quasi-Harmonic Model. Working in the same way as for the Harmonic Model in section 3.2.3, we will discard certain elements of the matrix to be inverted ($\mathbf{R}_{\text{QHM}} = \begin{bmatrix} \mathbf{R}_{0h} & \mathbf{R}_{1h} \\ \mathbf{R}_{1h}^H & \mathbf{R}_{2h} \end{bmatrix}$ for the QHM) and replace them with zeros. In the HM we turned the matrix to be inverted (\mathbf{R}_{0h} in that case) into a K_0 -band matrix by discarding elements that were away from the main diagonal more than K_0 elements. Turning \mathbf{R}_{QHM} to a band matrix would seem a natural extension. However, in HM we reached that realization by taking advantage of the fact that sinusoids that are away from each other have little interference. Considering that, we see that another workaround should be implemented in this case.

In Figures 3.3, 3.5 and 3.6 we show in log a scale the average, over 300 executions, magnitude of the $(\mathbf{R}_{0h})_{ik}$, $(\mathbf{R}_{1h})_{ik}$ and $(\mathbf{R}_{2h})_{ik}$ matrix elements, respectively, as a function of the distance of the individual element from the main diagonal. Since all of the \mathbf{R}_{mh} , $m = 0, 1, 2$ matrices are Toeplitz, all the elements on a given diagonal are equal. Figure 3.3 shows that the elements in the main diagonal have an average magnitude at the order 10^2 , while by moving away from the main diagonal more than five elements results in a reduction of the magnitude at approximately 10^{-5} . We can easily see (by substituting $g_0(x)$ from (3.15) to (3.41)) that, by construction, \mathbf{R}_{1h} has zeros in the main diagonal. However, in Figure 3.5 we can see that the elements of the upper- and sub-diagonal have an average magnitude of order 10^2 , while by moving away from the diagonal more than five elements results in reduction of the magnitude value to 10^{-2} . Concerning the last Figure 3.6, it can be observed that the main diagonal has an average magnitude of about 10^6 , while by moving away from the main diagonal more than five elements results in reducing its value to 10^0 .

Therefore, it can be said that the average magnitude values decrease as we move away from the main diagonal. Following the same principle as in the HM would mean that the technique should be applied to each sub-matrix of \mathbf{R}_{QHM} individually. This is translated into keeping K_0 diagonals from \mathbf{R}_{0h} , \mathbf{R}_{1h} and \mathbf{R}_{2h} . The other elements which are not close enough to the main diagonal are not computed and set to zero in an effort to reduce the computational load. In this Step we build the \mathbf{R}_{mh} , $m = 0, 1, 2$ matrices in a diagonal-by-diagonal way, which when combined with the fact that \mathbf{R}_{mh} are Toeplitz Hermitians it allows us to form \mathbf{R}_{mh} by calculating only one element per diagonal for about half of the diagonals (one element for the main diagonal and one element per upper/lower diagonals).

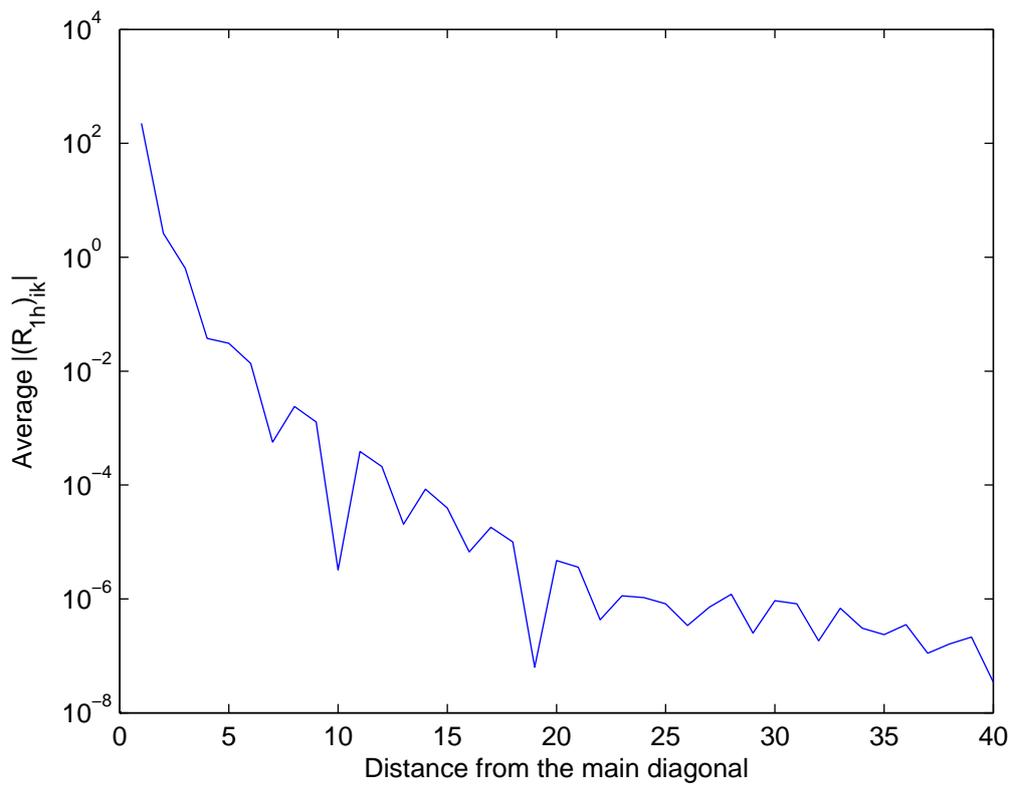


Figure 3.5: The average values of $|\mathbf{R}_{1h}|$ in log scale as a function of their distance from the main diagonal.

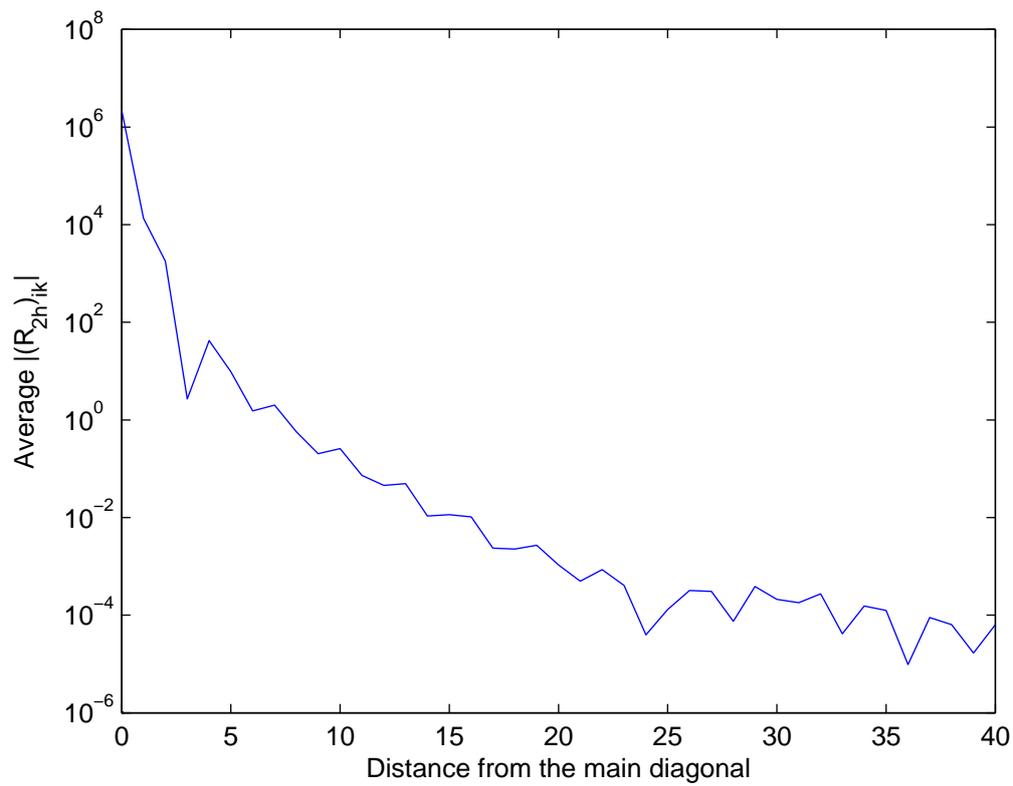


Figure 3.6: The average values of $|\mathbf{R}_{2h}|$ in log scale as a function of their distance from the main diagonal.

3.5 Generalized Quasi-Harmonic Model

In this section, we introduce the techniques implemented to accelerate the computation of the Generalized Quasi-Harmonic Model amplitude coefficients (introduced back in Section 2.4).

3.5.1 GQHM Step 1: Fast Matrix Multiplication

First, we calculate the terms of $\mathbf{R} = \mathbf{E}^H \mathbf{W}^H \mathbf{W} \mathbf{E}$ analytically. Considering the GQHM case (the aharmonic time-varying model) results in the following equation:

$$[\mathbf{E}_0 | \mathbf{E}_1]^H \mathbf{W}^H \mathbf{W} [\mathbf{E}_0 | \mathbf{E}_1] = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1 \\ \mathbf{R}_1^H & \mathbf{R}_2 \end{bmatrix}, \quad (3.46)$$

where the sub-matrices \mathbf{R}_0 , \mathbf{R}_1 and \mathbf{R}_2 are given by:

$$\mathbf{R}_0 = \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_0, \quad (3.47)$$

$$\mathbf{R}_1 = \mathbf{E}_0^H \mathbf{W}^H \mathbf{W} \mathbf{E}_1, \quad (3.48)$$

$$\mathbf{R}_2 = \mathbf{E}_1^H \mathbf{W}^H \mathbf{W} \mathbf{E}_1. \quad (3.49)$$

As far as (3.47) is concerned, we have shown in Section 3.3.1 that the matrix multiplication can be surpassed by calculating the elements of \mathbf{R}_0 directly via (3.24). Our purpose is to find similar expressions that will allow us to compute immediately the elements of \mathbf{R}_1 and \mathbf{R}_2 as well.

Working in a similar fashion, we substitute (3.9), which gives the squared window $\mathbf{W}^H \mathbf{W}$, to (3.48) and (3.49) and rewrite \mathbf{R}_1 and \mathbf{R}_2 , respectively, as follows:

$$\mathbf{R}_1 = \mathbf{W}^2 \mathbf{E}_0^H \mathbf{E}_1, \quad (3.50)$$

$$\mathbf{R}_2 = \mathbf{W}^2 \mathbf{E}_1^H \mathbf{E}_1. \quad (3.51)$$

Next, we compute to find the elements of \mathbf{R}_1 and \mathbf{R}_2 using the following expressions:

$$\begin{aligned} (\mathbf{R}_1)_{ik} &= \sum_{n=-N}^N w^2[n] e^{-j2\pi n f_i / f_s} n e^{j2\pi n f_k / f_s} \\ &= \sum_{n=-N}^N w^2[n] n e^{j2\pi n (f_k - f_i) / f_s}, \end{aligned} \quad (3.52)$$

$$\begin{aligned} (\mathbf{R}_2)_{ik} &= \sum_{n=-N}^N w^2[n] n e^{-j2\pi n f_i / f_s} n e^{j2\pi n f_k / f_s} \\ &= \sum_{n=-N}^N w^2[n] n^2 e^{j2\pi n (f_k - f_i) / f_s}. \end{aligned} \quad (3.53)$$

By employing the parameterized squared window from (3.4) in (3.52) and (3.53) we obtain:

$$\begin{aligned}
 (\mathbf{R}_1)_{ik} &= d_0 \sum_{n=-N}^N n \left[e^{j2\pi(f_k - f_i)/f_s} \right] \\
 &+ d_1 \sum_{n=-N}^N n \left[e^{j2\pi(f_k - f_i + \frac{f_s}{2N})/f_s} \right] + d_1 \sum_{n=-N}^N n \left[e^{j2\pi(f_k - f_i - \frac{f_s}{2N})/f_s} \right] \\
 &+ d_2 \sum_{n=-N}^N n \left[e^{j2\pi(f_k - f_i + \frac{f_s}{N})/f_s} \right] + d_2 \sum_{n=-N}^N n \left[e^{j2\pi(f_k - f_i - \frac{f_s}{N})/f_s} \right],
 \end{aligned} \tag{3.54}$$

$$\begin{aligned}
 (\mathbf{R}_2)_{ik} &= d_0 \sum_{n=-N}^N n^2 \left[e^{j2\pi(f_k - f_i)/f_s} \right] \\
 &+ d_1 \sum_{n=-N}^N n^2 \left[e^{j2\pi(f_k - f_i + \frac{f_s}{2N})/f_s} \right] + d_1 \sum_{n=-N}^N n^2 \left[e^{j2\pi(f_k - f_i - \frac{f_s}{2N})/f_s} \right] \\
 &+ d_2 \sum_{n=-N}^N n^2 \left[e^{j2\pi(f_k - f_i + \frac{f_s}{N})/f_s} \right] + d_2 \sum_{n=-N}^N n^2 \left[e^{j2\pi(f_k - f_i - \frac{f_s}{N})/f_s} \right].
 \end{aligned} \tag{3.55}$$

where the coefficients d_0 , d_1 and d_2 are given by equations (3.5), (3.6) and (3.7), respectively.

As before, the standard mathematical identity for the sum of geometric series is used. Employing (3.39), the elements of \mathbf{R}_1 show up, thus they can be computed without performing the summation. In a similar way, utilizing (3.40), the elements of \mathbf{R}_2 are computed.

Thus, in compact form, the elements of the sub-matrices \mathbf{R}_m , $m = 0, 1, 2$ are given by:

$$\begin{aligned}
 (\mathbf{R}_m)_{ik} &= d_0 g_m \left(2\pi (f_k - f_i) / f_s \right) \\
 &+ d_1 g_m \left(2\pi \left(f_k - f_i + \frac{f_s}{2N} \right) / f_s \right) + d_1 g_m \left(2\pi \left(f_k - f_i - \frac{f_s}{2N} \right) / f_s \right) \\
 &+ d_2 g_m \left(2\pi \left(f_k - f_i + \frac{f_s}{N} \right) / f_s \right) + d_2 g_m \left(2\pi \left(f_k - f_i - \frac{f_s}{N} \right) / f_s \right), m = 0, 1, 2
 \end{aligned} \tag{3.56}$$

where the auxiliary functions $g_0(x)$, $g_1(x)$ and $g_2(x)$ are given by (3.15), (3.42) and (3.43), respectively.

Finally, due to the fact that the computations of trigonometric functions are expensive, the fact that both sines and cosines are required to be calculated, enables us to decrease

the computational cost of (3.56) by exploiting the same identities utilized in the QHM model, given by (3.44). Thus, the sines and cosines of only one of the five terms in (3.56) are required, while the remaining terms are computed using the above formulas.

3.5.2 GQHM Step 2: Faster Computation of \mathbf{E}_0 and \mathbf{E}_1

As it has been thoroughly shown in section 3.3.2, we have already introduced a faster way of computing \mathbf{E}_0 by rewriting it as a sum of sines and cosines. Taking under consideration the form of arguments of these trigonometric functions we are able to estimate quickly the matrix elements using the solution of a difference equation. Thus, instead of computing exponentials or trigonometric functions, the computation of \mathbf{E}_0 has been replaced by (faster to calculate) multiplications. Obviously, having computed matrix \mathbf{E}_0 , the elements of \mathbf{E}_1 are given by $(\mathbf{E}_1)_{nk} = n(\mathbf{E}_0)_{nk}$ with its cost being one multiplication per array element. Thus we have replaced the $\mathcal{O}(NK)$ exponential or trigonometric calculations with an equal number of (computationally cheaper) multiplications, for both \mathbf{E}_{0h} and \mathbf{E}_{1h} .

3.5.3 GQHM Step 3: \mathbf{R}_{GQHM} Matrix Approximation

During the previous steps no approximation errors were introduced. Working in the same way as in the SM in 3.3.3, we discard certain elements of the matrix to be inverted ($\mathbf{R}_{\text{GQHM}} = \begin{bmatrix} \mathbf{R}_0 \mathbf{R}_1 \\ \mathbf{R}_1^H \mathbf{R}_2 \end{bmatrix}$ for the GQHM) and replace them with zeros. In the SM we turned the matrix to be inverted (\mathbf{R}_0 in that case) into a K_0 -band matrix by discarding elements that were far apart from the main diagonal more than K_0 elements. Turning \mathbf{R}_{GQHM} to a band matrix would be suitable. However, that realization in the HM was reached by considering the fact that sinusoids which are placed away from each other have a little interference. With that in mind, it is realized that a more complex variant of the same technique is required.

In Figures 3.4, 3.7 and 3.8 we can see in a log scale the average (over 300 runs) magnitude of the $(\mathbf{R}_0)_{ik}$, $(\mathbf{R}_1)_{ik}$ and $(\mathbf{R}_2)_{ik}$ matrix elements respectively as a function of the distance of the element from the main diagonal. Since the \mathbf{R}_m , $m = 0, 1, 2$ matrices are not in general Toeplitz and the elements of any given diagonal should have different values, a different average has been taken depending on how many elements at a given distance from the main diagonal exist. For instance, any $(2K + 1) \times (2K + 1)$ matrix \mathbf{R}_m has $2K + 1$ elements in the main diagonal, $2 \cdot 2K$ elements with a distance of one from the main diagonal, $2 \cdot (2K - 1)$ with a distance of two elements and so on, until we reach the two furthestmost elements that have a distance of $2K$ from the main diagonal. In Figure 3.4 it can be seen that the elements in the main diagonal have an average magnitude of 10^4 , while as we are moving away from it more than five elements it results in a magnitude reduction to values about 10^{-4} . We can easily see (by substituting $g_0(x)$ from (3.15) to (3.56)) that by construction, \mathbf{R}_1 has zeros in its main diagonal. However, Figure 3.7

shows that the elements of the upper- and sub-diagonal have average an measure of order 10^4 , while moving away from the diagonal more than five elements results in reducing its value to 10^{-1} . Regarding the Figure 3.6, it can be observed that the main diagonal has an average magnitude of about 10^8 , while by moving away from the diagonal more than five elements, results in reducing it at the order of 10^2 .

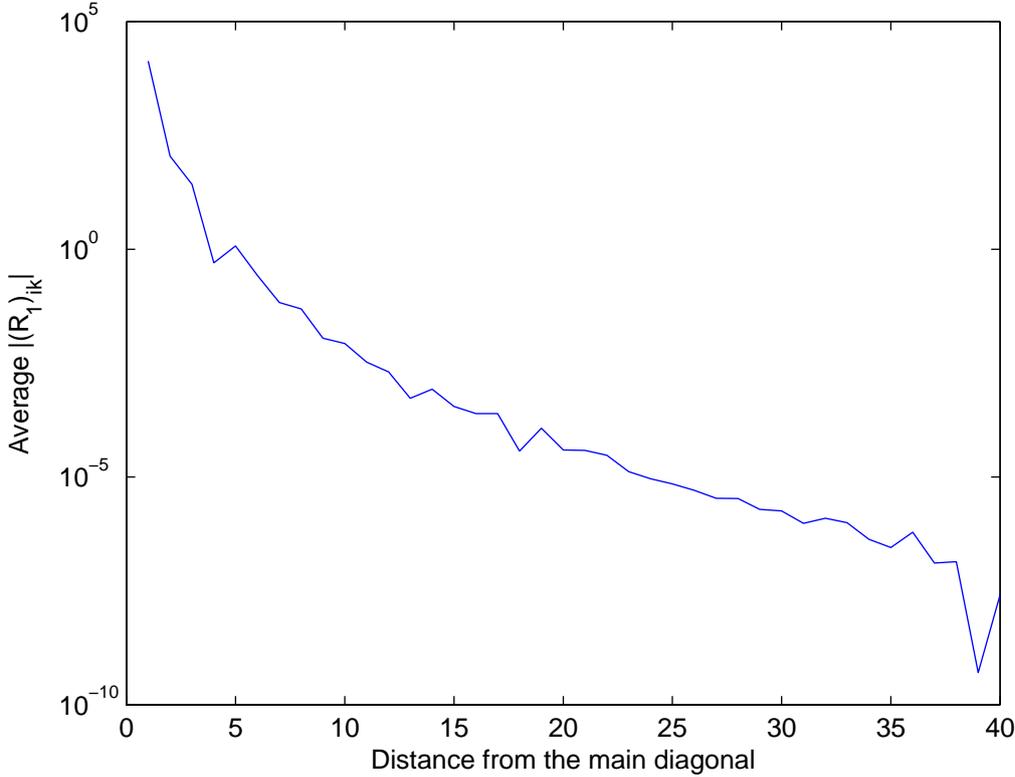


Figure 3.7: The average values of $|\mathbf{R}_1|$ in log scale as a function of their distance from the main diagonal.

Hence, from the above figures we can safely deduct that the average magnitude values decrease as we move away from the main diagonal. Following the same principle as in the SM would mean that the technique should be applied to each sub-matrix of \mathbf{R}_{GQHM} individually. This is equivalent to keeping K_0 diagonals from \mathbf{R}_0 , \mathbf{R}_1 and \mathbf{R}_2 . The other elements that are not close enough to the main diagonal are not computed and set to zero in an effort to reduce the computational load. In this step we build the \mathbf{R}_m , $m = 0, 1, 2$ matrices in a diagonal-by-diagonal manner which, when combined with the fact that \mathbf{R}_m are Hermitian, it allows us to construct \mathbf{R}_{mh} by calculating only about half of the diagonals (the main diagonal and the upper/lower diagonals).

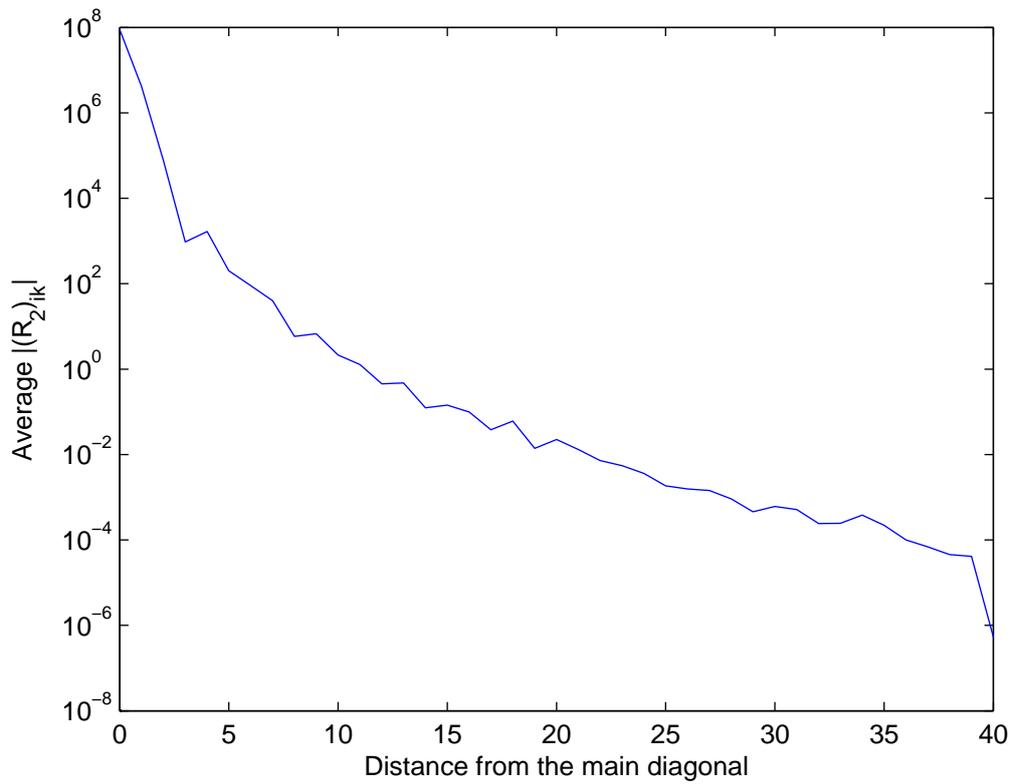


Figure 3.8: The average values of $|\mathbf{R}_2|$ in log scale as a function of their distance from the main diagonal.

Chapter 4

Evaluation

4.1 Computational Complexity

In this section, we start by analyzing the computational complexity of our proposed algorithms. Without any of our enhancing steps, which case is from now on denoted as Step 0, the total computational complexity of the LS solution is of order $\mathcal{O}(NK^2 + K^3 + NK)$, if we take into account that \mathbf{W} is diagonal. The first term, $\mathcal{O}(NK^2)$, arises from the multiplications from which the \mathbf{R}_m , $m = 0h, 1h, 2h, 0, 1, 2$ arrays are yielded, if computed directly from: (2.12), (2.38), (2.39), (2.25), (2.51) and (2.52). If Step 1 is used, each element of \mathbf{R}_m is computed in constant time ($\mathcal{O}(1)$), instead of $\mathcal{O}(N)$. Thus, the complexity of finding the \mathbf{R} matrices is reduced to the order of $\mathcal{O}(K^2)$ for the non-harmonic cases. While, for the harmonic cases taking advantage of the Hermitian Toeplitz forms of \mathbf{R}_m , $m = 0h, 1h, 2h$ reduces the order to $\mathcal{O}(K)$.

The term $\mathcal{O}(NK)$ is presented when calculating the \mathbf{E} matrix, which depending on the model can be \mathbf{E}_{0h} , \mathbf{E}_0 , $[\mathbf{E}_{0h}|\mathbf{E}_{1h}]$ or $[\mathbf{E}_0|\mathbf{E}_1]$. That which in practice burdens the CPU when computing the \mathbf{E} matrices, is the fact that each element calculation requires the usage of exponential or trigonometric functions. By exploiting Step 2, we achieve the computation of the sequences $\sin(2\pi n f_k/f_s)$ and $\cos(2\pi n f_k/f_s)$, for $k = -K, \dots, K$ and $n = -N, \dots, N$ (or $\sin(2\pi n k f_0/f_s)$ and $\cos(2\pi n k f_0/f_s)$ for the harmonic cases), by using approximately $8NK$ multiplications (2 multiplications per element of \mathbf{E}_0). It is noteworthy that, assuming that the multiplication has complexity $\mathcal{M}(d)$ ¹, where d the length of the numbers to be multiplied in digits, the complexity of calculating an exponential or a trigonometric function varies between $\mathcal{O}(d^{1/2}\mathcal{M}(d))$, $\mathcal{O}(d^{1/3}\log^2(d)\mathcal{M}(d))$ and $\mathcal{O}(\log^2(d)\mathcal{M}(d))$. Considering the asymptotically best case scenario ($\mathcal{O}(\log^2(d)\mathcal{M}(d))$) for the single call of a trigonometric function, the calculation of \mathbf{E}_{0h} and \mathbf{E}_0 without the usage of Step 2 requires $\mathcal{O}(KN\log^2(d)\mathcal{M}(d))$ operations, while after Step 2 it is reduced to $\mathcal{O}(KN\mathcal{M}(d))$. From the calculation \mathbf{E}_{1h} and \mathbf{E}_1 we have the same gain,

¹Depending on the multiplication algorithm used, the complexity $\mathcal{M}(d)$ represents at the worst case a complexity of order $\mathcal{O}(d^2)$ and at the best case $\mathcal{O}(d\log(d)\log(\log(d)))$.

since the direct method does not take advantage of the relation between \mathbf{E}_{0h} and \mathbf{E}_{1h} : $(\mathbf{E}_{1h})_{nk} = n(\mathbf{E}_{0h})_{nk}$ (and between \mathbf{E}_0 and \mathbf{E}_1 respectively: $(\mathbf{E}_1)_{nk} = n(\mathbf{E}_0)_{nk}$). Thus, while it takes at best $\mathcal{O}(KN \log^2(d)\mathcal{M}(d))$ operations (or $\mathcal{O}(KNd^{1/3} \log^2(d)\mathcal{M}(d))$ at worst) to find \mathbf{E} , after the implementation of Step 2 even for the time-varying models it reduces to $\mathcal{O}(KN\mathcal{M}(d))$ operations.

The remaining term $\mathcal{O}(K^3)$ stems from the inversion of the case dependent matrix \mathbf{R} , which has dimensions $(2K + 1) \times (2K + 1)$ for the HM and SM models and $(4K + 2) \times (4K + 2)$ for the time-varying QHM and GQHM. The act of allowing the approximation of the \mathbf{R} matrix by its K_0 -banded counterpart, reduces the cost of its inversion to $\mathcal{O}(K_0^2 K)$, if the positions of the zeroed elements are taken into account. Also, in Step 3 the fact that the $\mathbf{R}_m, m = 0h, 1h, 2h, 0, 1, 2$ are constructed in a diagonal-by-diagonal fashion, enabled us to take advantage of the Hermitian property of the matrices and calculate only $(K_0 - 1)/2 + 1$ diagonals. In total, the complexity was reduced from $\mathcal{O}(NK^2 + K^3 + NK)$ to $\mathcal{O}(K^2 + K_0^2 K + NK)$, for the general cases and to $\mathcal{O}(K + K_0^2 K + NK)$, for the harmonic cases.

4.2 Synthetic signals

In the following, we evaluate the performance of the proposed algorithm in a set of synthetic signals as follows:

$$\mathbf{s}[n] = \sum_{k=0}^K \cos(2\pi n f_k / f_s), \quad n = -N, \dots, N, \quad (4.1)$$

where $f_s = 16000$ Hz, which is a typical value for the sampling frequency. The parameters K and N vary in the following sets $K \in \{10, 20, \dots, 60\}$ and $N \in \{150, 175, \dots, 300\}$, respectively.

For the special case, of harmonic frequencies, $f_k = k f_0$, $k = -K, -K+1, \dots, K-1, K$, the fundamental frequency f_0 is chosen uniformly from the interval $[85, 255]$ Hz, which are typical values of fundamental frequency for the human speech.

For the general case, when the frequencies f_k have no relation to one another, we set $f_0 = 0$ and the rest of the frequencies (for $k = -K, -K+1, \dots, -1, 1, \dots, K-1, K$) are uniformly chosen from the interval $(85, \frac{f_s}{2})$ Hz under the conditions that every two frequencies should be at least 85 Hz apart and that $f_{k-1} < f_k$.

The experiments were performed on a computer equipped with an Intel Core 2 6600 processor at 2.4 GHz and a memory of 2GB. Note that only a single core of the CPU was used to ensure accuracy of the results. The operating system was Windows XP Professional (32 bit), while experiments were implemented using the MATLAB software.

4.2.1 Mean Square Error

In the following, we evaluate the estimation performance for the model parameters using the *Mean Square Error* (MSE) between the values estimated by employing the direct approach and the values we get when our accelerated algorithms are used. Assuming that \hat{x} is the estimated value of x , then the MSE of the estimator is defined as:

$$\text{MSE}(\hat{x}) = \text{E} [|\hat{x} - x|^2], \quad (4.2)$$

where $\text{E}[\cdot]$ indicates the mean value. The MSE is widely used in the design of algorithms and can assess the quality of an estimator in terms of its variation and unbiasedness.

In this experiment we analyzed 42000 arbitrarily created signals using the method described in the previous section (1000 per N and K case). Then, we calculated the average MSE for every modification and every individual model and model parameter. The results can be seen in Table 4.1. Note that Step 1 corresponds to the case where the elements of the matrices \mathbf{R}_m , $m = 0h, 1h, 2h, 0, 1, 2$ are computed quickly using the aforementioned auxiliary functions g_0, g_1, g_2 and (3.44). Step 2 refers to the additional improvement of accelerating the computation of \mathbf{E}_{0h} for the harmonic and \mathbf{E}_0 for the

general cases by using the difference equations (3.17) and (3.26), respectively, as well as Step 1 included. Finally, the set of Steps 3- K_0 for $K_0 = 3, 5, 7$ indicated that the methods described in Steps 1 and 2 were implemented and in addition that the \mathbf{R}_m matrices are K_0 -diagonalized (Step 3).

In Table 4.1 we can see that the error corresponding to the Steps 1 and 2 is very small and should be considered an unavoidable arithmetic error, especially when we consider the fact that all the calculations done in those steps were exact. Since approximations are performed at Steps 3-3, 3-5 and 3-7 a greater error is introduced, as expected. On the other hand, taking under consideration the fact that we approximated a matrix with its K_0 -diagonal counterpart the error is kept at acceptable levels.

	HM	SM	QHM		GQHM	
	a	a	a	b	a	b
Step 1	2.511e-015	2.511e-015	3.779e-012	3.553e-012	1.028e-013	6.658e-015
Step 2	2.402e-015	2.402e-015	3.779e-012	3.553e-012	1.664e-013	1.261e-014
Step 3-7	3.651e-006	3.651e-006	4.819e-003	1.010e-003	6.042e-004	3.653e-005
Step 3-5	1.485e-005	1.485e-005	1.868e-002	2.824e-003	1.186e-003	6.599e-005
Step 3-3	1.220e-004	1.220e-004	2.625e-002	2.030e-003	1.209e-002	6.207e-004

Table 4.1: Average MSE for each individual model and parameter compared to the direct LS solution.

4.2.2 Signal-to-Reconstruction Error Ratio

Through MSE we get an indication of how well the parameters are estimated. However, our main concern is how much the synthesized signal resembles the original signal. For this purpose, we used another 42000 signals generated as described in section 4.2. We analyzed and re-synthesized these signals and compared the re-synthesized with the original one using the *Signal-to-Reconstruction Error Ratio* (SRER) defined as:

$$\text{SRER} = 20 \log_{10} \frac{\sigma(\mathbf{s})}{\sigma(\mathbf{s} - \mathbf{y})}, \quad (4.3)$$

where \mathbf{s} is the original signal, \mathbf{y} is the reconstructed signal and $\sigma(\cdot)$ is the standard deviation which is defined by $\sigma(\mathbf{s}) = \sqrt{\mathbf{E}[(\mathbf{s} - \mathbf{E}[\mathbf{s}])^2]}$. In Table 4.2, we can see the resulting SRERs. Note that Step 0 is the case where no improvements are implemented. Therefore, Step 0 represents the highest value that can be reached. As we can see, the SRER values for Steps 1, 2 are hardly reduced as expected, since no approximations are performed. In steps 3- K_0 we can see that the SRER varies from 70 to 90 dB at the worst case and from 105 to 120 dB at the best case. Even considering the worst case scenario (72 dB) there is no significant loss of the SRER.

	HM	SM	QHM	GQHM
Step 0	277.43 dB	272.79 dB	279.44 dB	274.96 dB
Step 1	277.41 dB	272.64 dB	277.93 dB	271.38 dB
Step 2	277.10 dB	272.64 dB	274.37 dB	270.51 dB
Step 3-7	119.77 dB	125.94 dB	106.55 dB	112.89 dB
Step 3-5	107.85 dB	111.16 dB	94.73 dB	97.93 dB
Step 3-3	90.25 dB	86.20 dB	77.03 dB	72.06 dB

Table 4.2: Average SRER.

4.2.3 CPU time

Having studied the reliability of the parameter values and the reconstructed signals obtained by our proposed accelerations, in the following section we will focus on examining whether the actual process accelerated in terms of the required time. We randomly generated 42000 signals in the manner described in section 4.2 and analyzed these signals to find their amplitude coefficients. In Table 4.3, we observe the average execution time for each improvement. For convenience, Table 4.5 presents the relative CPU time improvements as a percentage. The improvement is computed with respect to the time it took to directly analyze the signals through the direct usage of the Least Squares method (Step 0).

As we can see in 4.5, from Step 0 to Step 1 we achieve an improvement of 45% both for the HM case, of 55% for the QHM and the improvements for the SM and GQHM cases are 6% and 26%, respectively. The reason for this difference in performance stems from the fact that sub-matrices \mathbf{R}_m are Toeplitz in the harmonic case. Step 2 offers almost the almost same improvement in performance in all the models as expected, since it is almost model independent. As it can be noticed in the second line of Table 4.4, step 2 saves about 2 ms in the GQHM and about 2.5 ms in all the other cases. The Steps 3- K_0 are always faster at about 80-85% faster.

	HM	SM	QHM	GQHM
Step 0	8.575 ms	8.650 ms	23.609 ms	23.736 ms
Step 1	4.666 ms	8.115 ms	10.569 ms	17.482 ms
Step 2	1.973 ms	5.454 ms	7.983 ms	15.455 ms
Step 3-7	1.474 ms	1.744 ms	4.155 ms	4.668 ms
Step 3-5	1.416 ms	1.620 ms	3.712 ms	4.092 ms
Step 3-3	1.397 ms	1.568 ms	3.424 ms	3.729 ms

Table 4.3: Average CPU times.

	HM	SM	QHM	GQHM
Step 0 - Step 1	3.909 ms	0.535 ms	13.040 ms	6.254 ms
Step 1 - Step 2	2.693 ms	2.661 ms	2.586 ms	2.026 ms
Step 2 - Step 3-7	0.499 ms	3.711 ms	19.455 ms	19.068 ms
Step 2 - Step 3-5	0.557 ms	3.834 ms	19.898 ms	19.643 ms
Step 2 - Step 3-3	0.576 ms	3.886 ms	20.185 ms	20.007 ms

Table 4.4: CPU time differences between every step and the previous one.

	HM	SM	QHM	GQHM
Step 1	45.583 %	6.184 %	55.232 %	26.349 %
Step 2	76.991 %	36.946 %	66.185 %	34.887 %
Step 3-7	82.810 %	79.841 %	82.402 %	80.335 %
Step 3-5	83.481 %	81.270 %	84.279 %	82.759 %
Step 3-3	83.705 %	81.871 %	85.496 %	84.291 %

Table 4.5: CPU time improvement of every step in percentage related to Step 0.

4.2.4 Noisy Signals

In the following, we test the robustness of the proposed algorithms in the case of noisy signals. In particular, we examine whether the approximation reduce the SRER when applied to noisy signals. Again, we employed 42000 signals randomly generated as described in section 4.2 corrupted by additive white Gaussian noise, such that the *Signal-to-Noise Ratio* (SNR) of the original signal decreased to 10, 20, ..., 80 dB. SNR is defined in similar a fashion as SRER:

$$\text{SNR} = 20 \log_{10} \frac{\sigma(\mathbf{s}_o)}{\sigma(\mathbf{s}_o - \mathbf{s})}, \quad (4.4)$$

where \mathbf{s}_o is the original synthetic signal and \mathbf{s} the noisy signal, which is also the signal to which the analysis is performed.

Tables 4.6 - 4.13 demonstrate the average SRER, the SNR varies from 5, 10, 20, ..., 80 dB. In most cases, the SRER is almost the same with the value given by Step 0 (our reference value, since it has no speed ups, thus no noise is induced by an approximation). Only, the tridiagonal case of Step 3 (Step 3-3) tends to deviate noticeably, especially for the SNR values of 70 and 80.

	HM	SM	QHM	GQHM
Step 0	8.20 dB	8.20 dB	9.56 dB	9.84 dB
Step 1	8.20 dB	8.20 dB	9.56 dB	9.84 dB
Step 2	8.20 dB	8.20 dB	9.56 dB	9.84 dB
Step 3-7	8.20 dB	8.20 dB	9.53 dB	9.84 dB
Step 3-5	8.20 dB	8.20 dB	9.56 dB	9.83 dB
Step 3-3	8.20 dB	8.20 dB	9.51 dB	9.54 dB

Table 4.6: Average SRER with additive noise (SNR=5).

	HM	SM	QHM	GQHM
Step 0	12.44 dB	12.49 dB	13.85 dB	14.13 dB
Step 1	12.44 dB	12.49 dB	13.85 dB	14.13 dB
Step 2	12.44 dB	12.49 dB	13.85 dB	14.13 dB
Step 3-7	12.44 dB	12.49 dB	13.83 dB	14.12 dB
Step 3-5	12.44 dB	12.49 dB	13.82 dB	14.10 dB
Step 3-3	12.44 dB	12.49 dB	13.71 dB	13.84 dB

Table 4.7: Average SRER with additive noise (SNR=10).

	HM	SM	QHM	GQHM
Step 0	22.09 dB	22.14 dB	23.49 dB	23.77 dB
Step 1	22.09 dB	22.14 dB	23.49 dB	23.77 dB
Step 2	22.09 dB	22.14 dB	23.49 dB	23.77 dB
Step 3-7	22.09 dB	22.14 dB	23.46 dB	23.77 dB
Step 3-5	22.09 dB	22.14 dB	23.44 dB	23.74 dB
Step 3-3	22.09 dB	22.14 dB	23.27 dB	23.29 dB

Table 4.8: Average SRER with additive noise (SNR=20).

	HM	SM	QHM	GQHM
Step 0	32.05 dB	32.10 dB	33.45 dB	33.74 dB
Step 1	32.05 dB	32.10 dB	33.45 dB	33.74 dB
Step 2	32.05 dB	32.10 dB	33.45 dB	33.74 dB
Step 3-7	32.05 dB	32.10 dB	33.39 dB	33.72 dB
Step 3-5	32.05 dB	32.10 dB	33.33 dB	33.66 dB
Step 3-3	32.05 dB	32.10 dB	33.00 dB	32.81 dB

Table 4.9: Average SRER with additive noise (SNR=30).

	HM	SM	QHM	GQHM
Step 0	42.05 dB	42.10 dB	43.45 dB	43.73 dB
Step 1	42.05 dB	42.10 dB	43.45 dB	43.73 dB
Step 2	42.05 dB	42.10 dB	43.45 dB	43.73 dB
Step 3-7	42.05 dB	42.10 dB	43.30 dB	43.68 dB
Step 3-5	42.05 dB	42.10 dB	43.17 dB	43.50 dB
Step 3-3	42.04 dB	42.07 dB	42.45 dB	41.98 dB

Table 4.10: Average SRER with additive noise (SNR=40).

	HM	SM	QHM	GQHM
Step 0	52.05 dB	52.10 dB	53.45 dB	53.73 dB
Step 1	52.05 dB	52.10 dB	53.45 dB	53.73 dB
Step 2	52.05 dB	52.10 dB	53.45 dB	53.73 dB
Step 3-7	52.05 dB	52.10 dB	53.15 dB	53.57 dB
Step 3-5	52.05 dB	52.10 dB	52.86 dB	53.18 dB
Step 3-3	51.98 dB	51.92 dB	51.41 dB	50.55 dB

Table 4.11: Average SRER with additive noise (SNR=50).

	HM	SM	QHM	GQHM
Step 0	62.05 dB	62.10 dB	63.45 dB	63.73 dB
Step 1	62.05 dB	62.10 dB	63.45 dB	63.73 dB
Step 2	62.05 dB	62.10 dB	63.45 dB	63.73 dB
Step 3-7	62.05 dB	62.10 dB	62.89 dB	63.32 dB
Step 3-5	62.04 dB	62.08 dB	62.28 dB	62.55 dB
Step 3-3	61.62 dB	61.32 dB	59.54 dB	58.07 dB

Table 4.12: Average SRER with additive noise (SNR=60).

	HM	SM	QHM	GQHM
Step 0	72.05 dB	72.10 dB	73.45 dB	73.73 dB
Step 1	72.05 dB	72.10 dB	73.45 dB	73.73 dB
Step 2	72.05 dB	72.10 dB	73.45 dB	73.73 dB
Step 3-7	72.04 dB	72.09 dB	72.40 dB	72.84 dB
Step 3-5	71.96 dB	71.99 dB	71.20 dB	71.36 dB
Step 3-3	70.37 dB	69.75 dB	66.23 dB	64.03 dB

Table 4.13: Average SRER with additive noise (SNR=70).

	HM	SM	QHM	GQHM
Step 0	82.05 dB	82.10 dB	83.45 dB	83.73 dB
Step 1	82.05 dB	82.10 dB	83.45 dB	83.73 dB
Step 2	82.05 dB	82.10 dB	83.45 dB	83.73 dB
Step 3-7	81.99 dB	82.05 dB	81.48 dB	81.95 dB
Step 3-5	81.49 dB	81.53 dB	79.13 dB	79.27 dB
Step 3-3	77.63 dB	76.57 dB	71.09 dB	68.03 dB

Table 4.14: Average SRER with additive noise (SNR=80).

4.3 Voice Signals

We illustrated the competence of the various Sinusoidal Models to analyze synthetic signals with and without the speed-up methods proposed. In this section we shall examine how well the models perform with real speech signals. Obviously, the time improvements have little difference with those of the synthetic signals and are not presented again. Our focus will be given in the reconstruction quality, which again is measured in terms of the SRER as previously by (4.3).

Contrary to the synthetic signals case, the frequency values of the sinusoids are unknown. Thus, the additional step of frequency estimation was required before estimating the amplitudes of the signal. For the Harmonic Case a variation of the YIN algorithm was used. The YIN algorithm estimates the fundamental frequency [1]. For the general case we performed a simple peak picking to the spectrum of the signal (found the K maximum values).

The recordings analyzed included various voiced utterations to ensure that the harmonic part of the signal was the dominant one, since no signal/noise separation was performed. For all the cases (both harmonic and non-harmonic) we set $K = 60$. The test set consists of signals with a sampling frequency $f_s = 44100$ Hz. For the harmonic models we set $N = 352$ and for the general models we set $N = 573$. This distinction is performed because (as we elaborated in 1.3) the frequency estimation methods used to detect the non-harmonic frequencies present approximation issues when using smaller windows. Erroneous frequency values have as an effect the significant reduction of the SRER, since the error affects both the process of the amplitude estimation (requires the frequencies as input) and the reconstruction of the signal.

	HM	QHM
Step 0	30.81 dB	49.23 dB
Step 1	30.81 dB	49.23 dB
Step 2	30.81 dB	49.23 dB
Step 3-7	30.81 dB	47.41 dB
Step 3-5	30.81 dB	42.37 dB
Step 3-3	30.80 dB	25.99 dB

Table 4.15: Average SRER for voiced speech using HM and QHM ($N = 352$)

In Tables 4.15 and 4.16 we can see the SRER for the harmonic and the general models. We can see that the time-varying models are always better as expected. But what is of great interest is that the SRER values do not vary significantly between step 0 (the reference values) and the next steps. With one exception being step 3-3 which in the harmonic case reduces the SNR from 42.3 dB (at step 3-5) to 25.9 dB. In general, it can be said that the speed up methods can be safely used in practice.

	SM	GQHM
Step 0	31.26 dB	36.30 dB
Step 1	31.26 dB	36.30 dB
Step 2	31.26 dB	36.30 dB
Step 3-7	31.26 dB	36.30 dB
Step 3-5	31.26 dB	36.09 dB
Step 3-3	31.25 dB	35.87 dB

Table 4.16: Average SRER voiced speech using SM and GQHM ($N = 573$)

Chapter 5

Conclusions and Future Work

In the present work our goal was to enhance the performance of the amplitude estimation for sinusoidal models. We began by doing some of the calculations manually, rendering a matrix multiplication redundant by finding a direct formula for the resulting elements of \mathbf{R} . For the Harmonic models, that also enabled us to exploit the Toeplitz form of the matrices to be calculated in order to avoid unnecessary calculations. Then, we rewrote the matrix \mathbf{E} comprised of exponentials, into a matrix consisting of trigonometric functions. Observing the pattern of the arguments of the trigonometric functions, allowed us to replace the calculations of these functions with simpler and faster multiplications by utilizing a differential equation. Following these steps we managed to reduce the computation times for the (simpler) Harmonic and Sinusoidal models by about 77% and 37% respectively. The time-varying Quasi Harmonic and Generalized Quasi-Harmonic Models had a reduction of 66% and 35% each.

In an effort to further reduce the computational cost of the Least Squares method for the computation of unknown complex parameters, we started by performing approximations of the matrices to be inverted (\mathbf{R}). Turning the matrices to be inverted into band matrices (or for the time-varying case into blocks of band matrices) was meaningful in more than one ways. First, it decreased both the computation time of the matrix (less elements to be computed) and the inversion. Second, the sinusoids that are away from each other have a little interaction which results in a matrix of which the elements away from the main diagonal have a smaller magnitude than the ones that are closer to the main diagonal. For the time-varying models we discussed that the equivalent action meant that we turn the sub-matrices of \mathbf{R} in band form. That step, when combined with the previous ones, decreased the computation time at about 80% - 85% depending the model and the length of the band allowed.

In addition, we showed that neither the calculations done by hand nor the approximations performed induced great distortions to the re-synthesized signal. The same comparison was fulfilled for real speech signals, which deducted that in practice the distortion induced by the frequency approximations was greater than the error introduced by the approximations.

As future research directions we are interested in designing a analysis/synthesis system using sinusoidal models, which would incorporate the proposed improvements. Then, one could testify whether these acceleration processes add artifacts to the re-synthesized signal, which are generally not detected by the SRER but easily heard by the human ear. This of course implies the implementation of fast frequency estimation techniques, which do not act as bottlenecks.

Undoubtedly, the possibility of additionally hastening of the estimation of the amplitude coefficients process is also worth investigating. Faster inversion of the case dependent matrix \mathbf{R} could be achieved if the special forms it obtains are exploited. The special categories seen in practice depend on the model. Additionally, if Step 3- K_0 is utilized, since it turns the matrix either in a K_0 -diagonal or in a block matrix consisting of K_0 -diagonal sub-matrices, another special form arises. Thus, the particular types that appear are Hermitian Toeplitz matrix, matrix consisting of 4 Hermitian Toeplitz sub-matrices, Hermitian K_0 -diagonal (banded) matrix, matrix consisting of 4 Hermitian K_0 -diagonal sub-matrices Hermitian Toeplitz K_0 -banded matrix and matrix consisting of 4 blocks of Hermitian Toeplitz K_0 -diagonal sub-matrices. The usage of lookup tables is a mean that could also help achieve even better results in terms of computational speed, however their usage is restricted to the harmonic models.

References

- [1] A. de Cheveign'e and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *Journal of Acoustical Society of America*, 2002.
- [2] W. D'haes and X. Roder. Discrete cepstrum coefficients as perceptual features. *International Computer Music Conference*, 2003.
- [3] W. D'haes and X. Roder. A new estimation technique for determining the control parameters of a physical model of a trumpet. *International Computer Music Conference*, 2003.
- [4] E. B. George and M. J. T. Smith. Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones. *Journal of the Audio Engineering Society*, 1992.
- [5] E. B. George and M. J. T. Smith. Speech analysis/synthesis and modification using an analysis-synthesis/overlap-add sinusoidal model. *IEEE Transactions on Speech and Audio Processing*, 1997.
- [6] S. M. Kay. *Fundamentals of Statistical Signal Processing, Estimation Theory*. Prentice Hall, 1993.
- [7] M. W. Macon and M. A. Clements. Speech concatenation and synthesis using an overlap-add sinusoidal model. *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996.
- [8] R. C. Maher. Evaluation of a method for separating digitized duet signals. *Journal of the Audio Engineering Society*, 1990.
- [9] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1986.
- [10] R. J. McAulay and T. F. Quatieri. The application of subband coding to improve quality and robustness of the sinusoidal transform coder. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1993.
- [11] Y. Pantazis, O. Rosenc, and Y. Stylianou. On the properties of a time-varying quasi-harmonic model of speech. *Interspeech*, 2008.

REFERENCES

- [12] Y. Pantazis, G. Tzedakis, O. Rosec, and Y. Stylianou. Analysis/synthesis of speech based on an adaptive quasi-harmonic plus noise model. *EEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.
- [13] G. Peeters, S. McAdams, and P. Herrera. Instrument sound description in the context of mpeg-7. *International Computer Music Conference*, 2000.
- [14] G. Peeters and X. Rodet. Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instrument databases. *6th International Conference on Digital Audio Effects*, 2003.
- [15] T. F. Quatieri and R. G. Danisewicz. An approach to co-channel talker interference suppression using a sinusoidal model for speech. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1988.
- [16] T. F. Quatieri and R. J. McAulay. Sinewave-based phase dispersion for audio pre-processing. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1988.
- [17] T. F. Quatieri and R. J. McAulay. Noise reduction using a soft-decision sinewave vector quantizer. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1990.
- [18] T. F. Quatieri and R. J. McAulay. Peak-to-rms reduction of speech based on a sinusoidal model. *IEEE Transactions on Signal Processing*, 1991.
- [19] T. F. Quatieri and R. J. McAulay. Shape invariant time-scale and pitch modification of speech. *IEEE Transactions on Signal Processing*, 1992.
- [20] X. Rodet. Musical sound signals analysis/synthesis: Sinusoidal + residual and elementary waveform models. *Proceedings of the IEEE Time-Frequency and Time-Scale Workshop (TFTS'97)*, 1997.
- [21] J. Rutledge. *Time-Varying, Frequency Dependent Compensation for Recruitment of Loudness*. PhD thesis, Georgia Institute of Technology, 1989.
- [22] J. Rutledge and M. Clements. Compensation for recruitment of loudness in sensorineural hearing impairments using a sinusoidal model of speech. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1991.
- [23] X. Serra and I. Julius Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 1990.
- [24] J. Severson. Modeling and frequency tracking of marine mammal whistle calls. Master's thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2009.

- [25] F. M. Silva and L. B. Almeida. Speech separation by means of stationary least-squares harmonic estimation. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1988.
- [26] P. Stoica, R. L. Moses, B. Friedlander, and T. Soderstrom. Maximum likelihood estimation of the parameters of multiple sinusoids from noisy measurements. *IEEE Transactions Acoustics, Speech and Signal Processing*, 1989.
- [27] I. Stylianou. *Harmonic plus Noise Models for Speech, combined with Statistical Methods for Speech and Speaker Modification*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1996.
- [28] Y. Stylianou. A simple and fast way of generating a harmonic signal. *Signal Processing Letters, IEEE*, 2000.
- [29] T. Tolonen. Methods for separation of harmonic sound sources using sinusoidal modeling. *Audio Engineering Society Convention*, 1999.
- [30] T. Tolonen. Object-based sound source modeling for musical signals. *Audio Engineering Society Convention*, 2000.
- [31] G. Tzedakis, Y. Pantazis, O. Rosec, and Y. Stylianou. Fast least-squares solution for sinusoidal, harmonic and quasi-harmonic models. *Interspeech*, 2010.
- [32] T. Virtanen. Separation of harmonic sounds using multipitch analysis and iterative parameter estimation. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.
- [33] T. Virtanen. Algorithm for the separation of harmonic sounds with time-frequency smoothness constraint. *International Conference on Digital Audio Effects*, 2003.