CHAPTER 1. INTRODUCTION

Speech coding or speech compression is one of the important aspects of speech communications nowadays. Some of the speech communication media that need speech coding are wireless communications and Internet telephony. By coding the speech, the speed to transmit the digitized speech, called the bit rate, can be reduced. This means that for a certain speech communications channel, the lower the bit rate of the speech coding, the more communicating parties can be carried on that channel. This research has as its main application the extraction of the parameters of human speech for speech coding purposes.

1.1. Literature Survey

Code-Excited Linear Prediction (CELP) is one of the most popular forms of speech coding. In CELP-based speech coding, the human vocal tract is modeled by an autoregressive (AR) process which is extracted by inverse filtering [1] [19]. The envelope of the power spectrum associated with the AR parameters is very important, not only for intelligibility but also for the perceived quality of the synthesized speech [61]. The AR parameters are usually extracted by using the autocorrelation method, and can be solved for efficiently by using the Levinson algorithm [43] [32] [19].

However, the AR parameters, which are usually called LPC coefficients, are not very suitable for direct quantization [42]. Therefore, the AR parameters are transformed into another representation that has better quantization properties. One of the popular alternative representations for the AR parameters is in terms of the line spectral frequencies (LSF) [37] [42]

[74]. The LSF are suitable for quantization because they are closely related to the formant frequencies.

To obtain the LSF, symmetric and anti-symmetric polynomials are derived from the AR parameters [42]. The LSF are then the roots of these polynomials, where the roots of the symmetric and anti-symmetric polynomials have the property of being on the unit circle and interlaced. Since root finding is not suitable for real time computation, the roots are obtained by using search and approximation techniques. A widely used approximation technique uses the Chebyshev approximation [42].

Another important parameter that needs to be extracted from the speech signal is the pitch [1] [39] [44]. The pitch represents the movement of the glottis during the production of the speech signal [19]. The most popular method used to extract the pitch is the autocorrelation method, where the pitch is the inverse of the lag at the highest autocorrelation peak within a specified lag range not including lag zero. Strictly speaking, the concept of pitch exists only during voiced speech, where the speech waveform clearly shows some periodicity. During unvoiced speech, the speech waveform does not show any periodicity. The most common errors in pitch estimation are pitch doubling and pitch halving. As the names indicate, pitch doubling is a condition where the estimated pitch is twice as high as the "true" pitch, and pitch halving is a condition where the estimated pitch is half the "true" pitch.

Since the human vocal tract and the pitch are non-stationary processes, the use of adaptive filters seems natural. By using an adaptive filter, the evolution of the vocal tract and the pitch can be estimated and tracked. The popular least mean squares (LMS) adaptive algorithm is not a good choice for the speech application since LMS converges too slowly (relative to the changes in the vocal tract) when the input is ill-conditioned, which it is for speech signals [32]

[23]. The Recursive Least Squares (RLS) algorithm is a better choice. However, the RLS algorithm is computationally complex [32]. The reason being that, for the RLS algorithm, we need to compute the inverse of the auto-correlation matrix of the input data. Even though, by using the matrix inversion lemma, the direct computation of the inverse of the auto-correlation matrix can be avoided, the required computational effort is still on the order of the square of the order of the filter (and thus presently too high for real time computation).

There have been a number of attempts to reduce the computational complexity of the RLS algorithm. The fast RLS (FRLS) algorithm utilizes the shift-invariance property of the data matrix to update the Kalman gain in a “fast” scheme [29] [32]. The term “fast” means that the computational effort of the algorithm is a linear function of the order of the filter. Unfortunately, the FRLS algorithm suffers from finite quantization instability [32], [29]. Furthermore, the forgetting factor for FRLS has to be kept close to 1 - the higher the order, the closer to 1 for the forgetting factor -, otherwise FRLS will diverge [29], [84].

The Fast Quasi Newton algorithm is a variant of FRLS, where the adaptations of the autocorrelation matrix are done on a block-by-block basis, even though the adaptive filter coefficients are adapted sample-by-sample [29]. Furthermore, the Kalman gain computation, which involves the computation of the inverse of the autocorrelation matrix, utilizes the Levinson algorithm. If the input signal is a speech signal, which can be modeled as an N-th order AR process where N is less then or equal to the total order of the filter M, the MxM autocorrelation matrix can be obtained by an extrapolation of the NxN autocorrelation matrix of the input signal. This is the approach used by the Fast Newton Transversal Filter (FNTF) algorithm [29], [54]. The inverse of the MxM autocorrelation matrix needed to compute the

Kalman gain can be calculated using the Levinson algorithm. For inverse filtering applications in speech coding, where N is equal to M, the FNTF algorithm becomes fast RLS, i.e. FRLS.

1.2. Purpose and New Contribution

In this dissertation we propose an RLS-based cascade adaptive filter structure that can significantly reduce the computational effort required by the RLS algorithm for inverse filtering types of applications. We named it the Cascade RLS with Subsection Adaptation (CRLS-SA) algorithm. The reduction in computational effort comes from the fact that, for inverse filtering applications, the gradients of each section in the cascade are almost uncorrelated with the gradients in other sections. Hence, the gradient autocorrelation matrix is assumed to be block diagonal. Since we use a second order filter for each section, the computation of the adaptation involves only the *2x2*-gradient autocorrelation matrix for that section, while still being based on a global minimization criterion. The gradient signal of a section itself is defined as the derivative of the overall output error with respect to the coefficients of the particular section, which can be computed efficiently by passing the overall output of the cascade to a filter with coefficients that are derived from the coefficients of that section. Here, the computational effort of the CRLS-SA algorithm is approximately $20*L*N/2$ where L is the data record length, and N is the order of the filter [85]. This research is thought to be original, as no publication about this topic has been found.

The CRLS-SA algorithm shows a faster convergence rate than the Direct Form RLS (DFRLS) algorithm. We show that the faster convergence of CRLS-SA corresponds to the fact that its convergence time constant [18] [3] is lower than that of the direct form. The convergence time constant is defined as the ratio of the condition number and the sensitivity. The convergence behavior is verified by looking at how fast the estimated system approaches the true system.

Here we use the Itakura distance as the measure of closeness between the estimated and the true system. We show that the Itakura distance associated with the CRLS-SA algorithm approaches zero faster than that associated with the direct form RLS algorithm. Even though the sensitivity of the CRLS-SA algorithm is sometimes larger than that of the direct form, perturbations to the CRLS-SA parameters yield less distortion of the overall system than when the direct form parameters are subject to the same amount of perturbation. This means that the CRLS-SA structure is more robust than the DF structure.

The CRLS-SA algorithm is applied in this dissertation to general linear prediction, to the direct adaptive computation of the LSF, named LSF_2, and to the detection and tracking of the pitch and harmonics. The LSF are also investigated in terms of their representation in quantized form using a vector quantization (VQ) approach.

The CRLS-SA algorithm is also used for estimating and tracking frequencies of a periodic signal. For estimating a low frequency, much lower than the sampling frequency, the down-sampling technique is used prior to the frequency estimation. This is especially useful to estimate the fundamental frequency, or pitch, of a speech signal, which can be very low. For estimating the fundamental frequency of a continuous speech signal, we can keep the previous fundamental frequency estimate as the initial value for the computation of the current fundamental frequency estimate. This way we can have a smooth evolution of the fundamental frequency over time.

1.3. Organization

The rest of the dissertation is organized as follows. Chapter 2 consists of the approach and computational aspects of the CRLS-SA algorithm. In this chapter, a quick review of the RLS algorithm is given, while the rest is dedicated to CRLS-SA. We show experimentally that the gradients of the different sections are nearly uncorrelated for an AR process signal, such as speech signals. Chapter 3 covers the convergence analysis of the CRLS-SA algorithm. Chapter 4 covers the applications of the CRLS-SA algorithm to frequency estimation. Chapter 5 covers the direct LSF computation, including the VQ representation of the LSF. Chapter 6 provides conclusions and suggestions for further research.