

Quantization process



Quantization is a process of mapping an infinite set of scalar or vector quantities by a finite set of scalar or vector quantities. Quantization has applications in the areas of signal processing, speech processing and Image processing. In speech coding, quantization is required to reduce the number of bits used for representing a sample of speech signal there by the bit-rate, complexity and memory requirement can be reduced. Quantization results in the loss in the quality of a speech signal, which is undesirable. So a compromise must be made between the reduction in bit-rate and the quality of speech signal.

Two types of quantization techniques exist they are: scalar quantization and vector quantization. " Scalar quantization deals with the quantization of samples on a sample by sample basis", while " vector quantization deals with quantizing the samples in groups called vectors". Vector quantization increases the optimality of a quantizer at the cost of increased computational complexity and memory requirements.

" Shannon theory states that quantizing a vector will be more effective than quantizing individual scalar values in terms of spectral distortion". According to Shannon the dimension of a vector chosen greatly affects the performance of quantization. Vectors of larger dimension produce better quality when compared to vectors of smaller dimension and in vectors of smaller dimension the transparency in the quantization is not good at a particular bit-rate chosen [8]. This is because in vectors of smaller dimension the correlation that exists between the samples will be lost and the scalar quantization itself destroys the correlation that exists between successive samples so the quality of the quantized speech signal will be lost. Therefore,

quantizing correlated data requires techniques that preserve the correlation between the samples, such a technique is the vector quantization technique (VQ). Vector quantization is the simplification of scalar quantization. Vectors of larger dimension produce transparency in quantization at a particular bit-rate chosen. In Vector quantization the data is quantized in the form of contiguous blocks called vectors rather than individual samples. But later with the development of better coding techniques, it is made possible that transparency in quantization can also be achieved even for vectors of smaller dimension. In this thesis quantization is performed on vectors of full length and on vectors of smaller dimensions for a given bit-rate [4, 50].

An example of 2-dimensional vector quantizer is shown in Fig 4. 1. The 2-dimensional region shown in Fig 4. 1 is called as the voronoi region, which in turn contains several numbers of small hexagonal regions. The hexagonal regions defined by the blue borders are called as the encoding regions. The green dots represent the vectors to be quantized which fall in different hexagonal regions and the red dots represent the codeword's (centroids). The vectors (green dots) falling in a particular hexagonal region can be best represented by the codeword (red dot) falling in that hexagonal region [51-54].

Vector quantization technique has become a great tool with the development of non variational design algorithms like the Linde, Buzo, Gray (LBG) algorithm. On the other hand besides spectral distortion the vector quantizer is having its own limitations like the computational complexity and memory requirements required for the searching and storing of the codebooks. For applications requiring higher bit-rates the computational complexity and

memory requirements increases exponentially. The block diagram of a vector quantizer is shown in Fig 4. 2.

Let x be an N dimensional vector with real valued samples in the range $[-1, 1]$. The superscript T in the vector denotes the transpose of the vector. In vector quantization, a real valued N dimensional input vector is matched with the real valued N dimensional codeword's of the codebook C_i , the codeword that best matches the input vector with lowest distortion is taken and the input vector is replaced by it. The codebook consists of a finite set of codeword's $C = \{C_1, C_2, \dots, C_L\}$, where C is the codebook, L is the length of the codebook and C_i denote the i th codeword in a codebook. In LPC coding the high bit-rate input vectors are replaced by the low bit-rate codeword's of the codebook. The parameters used for quantization are the line spectral frequencies (LSF).

The parameters used in the analysis and synthesis of the speech signals are the LPC coefficients. In speech coding the quantization is not performed directly on the LPC coefficients, the quantization is performed by transforming the LPC coefficients to other forms which ensure filter stability after quantization. Another reason for not using LPC coefficients is that, LPC coefficients have a wide dynamic range and so the LPC filter easily becomes unstable after quantization. So LPC coefficients are not used for quantization. The alternative to LPC coefficients is the use of line spectral frequency (LSF) parameters which ensure filter stability after quantization. The filter stability can be checked easily just by observing the order of the LSF samples in an LSF vector after quantization. If the LSF samples in a vector are in the ascending or descending order the filter stability can be ensured otherwise the filter stability cannot be ensured [54-58].

The angular positions of the roots of and gives us the line spectral frequencies and occurs in complex conjugate pairs. The line spectral frequencies range from.

The line spectral frequencies have the following properties:

Ø All the roots of and must lie on the unit circle which is the required condition for stability.

Ø The roots of and are arranged in an alternate manner on the unit circle i. e.,

The roots of equation (4. 6) can be obtained using the real root method [31] and is The coefficients of equations (4. 6) and (4. 7) are symmetrical and so the order p of equations (4. 6) and (4. 7) get reduces to $p/2$.

Vector quantization of speech signals requires the generation of codebooks. The codebooks are designed using an iterative algorithm called Linde, Buzo and Gray (LBG) algorithm. The input to the LBG algorithm is a training sequence. The training sequence is the concatenation of a set LSF vectors obtained from people of different groups and of different ages. The speech signals used to obtain training sequence must be free of background noise. The speech signals used for this purpose can be recorded in sound proof booths, computer rooms and open environments. In this work the speech signals are recorded in computer rooms. In practice speech data bases like TIMIT database, YAHOO data base are available for use in speech coding and speech recognition.

The codebook generation using LBG algorithm requires the generation of an initial codebook, which is the centroid or mean obtained from the training sequence. The centroid, so obtained is then splitted into two centroids or codeword's using the splitting method. The iterative LBG algorithm splits these two codeword's into four, four into eight and the process will be continued till the required numbers of codeword's in the codebook are obtained [59-61].

The flow chart of LBG algorithm is shown in Fig 4. 3. The LBG algorithm is properly implemented by a recursive procedure given below:

1. Initially the codebook generation requires a training sequence of LSF parameters which will be the input to LBG algorithm. The training sequence is obtained from a set of speech samples recorded from different groups of people in a computer room.
2. Let R be the region of the training sequence.
3. Obtain an initial codebook from the training sequence, which is the centroid or mean of the training sequence and let the initial codebook be C .
4. Split the initial codebook C into a set of codeword's and where is the minimum error to be obtained between old and new codeword's.
5. Compute the difference between the training sequence and each of the codeword's and and let the difference be D .
6. Split the training sequence into two regions R_1 and R_2 depending on the difference D between the training sequence and the codeword's and. The

training vectors closer to falls in the region R1 and the training vectors closer to falls in the region R2.

7. Let the training vectors falling in the region R1 be TV1 and the training sequence vectors falling in the region R2 be TV2.

8. Obtain the new centroid or mean for TV1 and TV2. Let the new centroids be CR1 and CR2.

9. Replace the old centroids and by the new centroids CR1 and CR2.

10. Compute the difference between the training sequence and the new centroids CR1 and CR2 and Let the difference be .

11. Repeat steps 5 to 10 until .

12. Repeat steps 4 to 11 till the required number of codeword's in the codebook are obtained.

Where $N = 2^b$ represents the number of codeword's in the codebook and b represents the number of bits used for codebook generation. represents the difference between the training sequence and the old codeword's, represents the difference between the training sequence and the new codeword's.

The quality of the speech signal is an important parameter in speech coders and is measured in terms of spectral distortion measured in decibels (dB). The spectral distortion is measured between the LPC power spectra of the quantized and unquantized speech signals. The spectral distortion is measured frame wise and the average or mean of the spectral distortion

calculated over all frames will be taken as the final value of the spectral distortion. For a quantizer to be transparent the mean of the spectral distortion must be less than 1 dB without any audible distortion in the reconstructed speech. But the mean of the spectral distortion is not a sufficient measure to find the performance of a quantizer, this is because the human ear is sensitive to large quantization errors that occur occasionally. So in addition to measuring the mean of the spectral distortion it is also necessary to have another measure of quality which is the percentage number of frames having a spectral distortion greater than 2dB and less than 4dB and the percentage number of frames having a spectral distortion greater than 4dB. The frames having spectral distortion between 2 to 4dB and greater than 4dB are called as outlier frames [54].

In order to measure objectively the distortion between the quantized and unquantized outputs, a method called the spectral distortion is often used in narrowband speech coding. For an i th frame the spectral distortion (in dB), is given by equation (4. 19).

(4. 19) Where P_u and P_q are the LPC power spectra of the unquantized and quantized i th frame respectively. The frequency f is in Hz and the frequency range is given by f_1 and f_2 . The frequency range used in practice for narrowband speech coding is 0-4000 Hz [12, 33].

The conditions for transparent speech coding are:

Ø The average or mean of the spectral distortion (SD) must be less than or equal to 1dB.

∅ There must be no outlier frames having a spectral distortion greater than 4dB.

∅ The number of outlier frames between 2 to 4dB must be less than 2%.

These three conditions are required to evaluate the performance of a quantizer. At a given bit-rate, an optimization process has to be carried out so as to obtain better performance i. e., accepting a large average spectral distortion for a few outliers.

In the design of a vector quantizer instead of using the mean squared error (MSE) distance measure the weighted LSF distance measurement is used. This is done to place emphasis on the low frequency LSFs and on the LSFs with higher power spectrum. The weights used can be of two types they are: static or dynamic [54].

∅ Fixed or Static weights : These are used to place emphasis on the low frequency LSFs in order to account for the sensitivity of human ear for low and high frequency LSFs.

∅ Varying or Dynamic weights : These are used to place emphasis on the LSFs with high power spectrum.

There exist a number of vector quantization techniques each one is having its own advantages and disadvantages. Each technique is developed to decrease the parameters like spectral distortion, computational complexity and memory requirements. The vector quantization techniques that exist are the Split Vector Quantization (SVQ) technique, Multistage Vector Quantization (MSVQ) technique, Split-Multistage Vector Quantization (S-

MSVQ) technique and Switched Split Vector Quantization (SSVQ) technique. As marketability and cost of a product depends on the complexity and memory requirements, the performance of the vector quantization techniques is measured in terms of spectral distortion in decibels, computational complexity in kilo flops per frame and memory requirements in floats. The performance of a vector quantization technique mainly depends on how efficiently the codebook is generated. The codebook can be generated efficiently using a large training set and using more number of bits for codebook generation. The goal involved in the design of each vector quantization technique is to make the technique to use more number of training vectors and less number of bits for codebook generation there by the spectral distortion, computational complexity and memory requirements can be reduced. It has been observed that as the number of bits used for codebook generation decreases the computational complexity and memory requirements decreases but the spectral distortion increases, this increase in spectral distortion can be reduced by increasing the number of training vectors used for codebook generation [62-71].

The block diagram of an Unconstrained Vector Quantizer (UVQ) is shown in Fig 4. 4.

Unconstrained Vector Quantization technique is the most awful vector quantization technique used for achieving lowest distortion at a given bit-rate and dimension. In LPC-10 the order of the filter chosen is 10 and so the length of each LSF vector will be 10. In Unconstrained Vector Quantization technique the quantization is done on vectors of full length i. e., using 10 samples of an LSF vector. From Fig 4. 4 $S_1, S_2, S_3, \dots, S_n$ are the input LSF vectors to be quantized using the Unconstrained Vector Quantizer.

The main advantage of this vector quantization technique is that it is expected to give lowest quantization distortion at a given bit-rate as the correlation that exists between the samples of a vector is preserved. But the disadvantage with this quantization technique is that as vectors of full length are used, at higher bit-rates the computational complexity and memory requirements increases in an exponential manner making it impractical for applications requiring higher bit-rates. Another problem with this quantization technique is that at higher bit-rates the size of the codebook will be large and the generation of the codebook for this type of quantization technique will be difficult on general purpose computers as the memory available with them is limited. So the number of training vectors used for codebook generation must be limited in number or the length of each vector must be reduced. In practice on general purpose computers the codebook cannot be generated even with training vectors less than the number of codeword's in a codebook at higher bit-rates. But the number of training vectors required to generate the codebook must be large than the number of codeword's in a codebook otherwise there will be too much over fitting of the training set [54].

The computational complexity and memory requirements of a b bit, n dimensional vector quantizer are calculated as follows [54]:

Ø To calculate the mean square error (MSE) between two vectors of n dimension, n subtractions, n multiplications and n-1 additions are required. So a total of $3n-1$ flops are required.

Ø To search a codebook of 2^b code vectors, $(3n-1)2^b$ flops are required in addition to the minimum distortion search requiring 2^{b-1} flops.

∅ So the number of computations made by a b bit, n dimensional vector quantizer is

$$\text{Total complexity} = (3n-1)2^b + 2^{b-1}$$

$$= 3n2^{b-1} \text{ flops per vector. (4. 24)}$$

In the computing the complexity each addition, multiplication and comparison is considered as one floating point operation. So a b bit n dimensional vector quantizer requires a codebook of 2^b code vectors, it needs to store $n2^b$ floating point values, it computes $3n2^b - 1$ flops per vector. Instead of the mean square error distance measure if weighted distance measure is used in the design of a vector quantizer the complexity increases from $3n2^b - 1$ to $4n2^b - 1$ flops per vector.

The computational complexity of an Unconstrained Vector Quantizer is given by equation (4. 25)

Where

n is the dimension of the vector

b is the number of bits allocated to the vector quantizer.

The Memory requirements of an Unconstrained Vector Quantizer is given by equation (4. 26)

Exhaustive search vector quantizers achieve lowest distortion at the expense of complexity and memory requirements at higher bit- rates. So to make the vector quantizers more practical for vectors of larger dimension and higher

bit-rates structural constraints are imposed on the design of a vector quantizer or codebook. One way of achieving this is to decompose the codebook into a Cartesian product of smaller codebooks i. e., $C = C_1 * C_2 * C_3 \dots \dots * C_m$. The advantage with smaller codebooks is that the computational complexity and memory requirements can be reduced to a very great extent. This is because the number of bits used for codebook generation will be divided among the sets of the decomposed codebook [12, 18]. Examples of product code vector quantization techniques are Split Vector Quantization (SVQ), Multistage Vector Quantization (MSVQ), Split-Multistage Vector Quantization (S-MSVQ), Switched Split Vector Quantization (SSVQ). In this thesis two product code vector quantization techniques are proposed they are: Switched Multistage Vector Quantization (SWMSVQ) and Multi Switched Split Vector Quantization (MSSVQ) techniques [54, 72].

The main disadvantage of Unconstrained Vector Quantizer is that the complexity, memory requirements are very high and the generation of codebook is a very difficult task as vectors of full length are used for quantization without any structural constraint. As a result more number of training vectors and bits cannot be used for codebook generation. With these constraints the quantizer cannot produce better quality quantized outputs. So to improve the performance of Unconstrained Vector Quantization technique a well known technique called Split Vector Quantization has been developed. The concept behind Split Vector Quantization is that, in it vectors of larger dimensions are splitted into vectors of smaller dimensions and the bits allocated to the quantizer are divided among the splits (parts). Due to splitting the dimension of a vector gets decreased hence more number of

training vectors and bits are used for codebook generation. As a result the performance of quantization is increased, the complexity and memory requirements are reduced. But the main disadvantage with this technique is that, due to splitting the linear and non linear dependencies that exist between the samples of a vector will be lost and the shape of the quantizer cells will be affected. As a result the spectral distortion increases slightly. This increase in spectral distortion can be compensated by increasing the number of training vectors and using more number of bits for codebook generation. The number of splits in this type of quantizer must be limited in number otherwise the vector quantizer will act as a scalar quantizer.

In Split Vector Quantization the training sequence used for codebook generation will also be splitted into vectors of smaller dimension and each split of the training sequence is used to generate separate sub codebooks, there by independent vector quantizers exist and the bits must be allocated to each of them. As a result less number of bits will be available at each quantizer, the computational complexity and memory requirements gets reduced as they depend on the number of bits allocated to the quantizer and on the dimension of the vector to be quantized. The block diagram of a three part Split vector quantizer is shown in Fig 4. 5.

From Fig 4. 5 it can be observed that a vector S_1 of dimension n is quantized by splitting it into sub-vectors S_{11} , S_{12} , S_{13} of smaller dimensions. Each of these sub-vectors are quantized using their respective codebooks. In this work the order of the filter is taken as 10 and so the LSF vector contain 10 samples and these 10 samples are splitted into three parts of 3, 3, 4 samples [54, 73-75].

From results of Split Vector Quantization technique it is proved that the computational complexity and memory requirements gets decreased when compared to Unconstrained Vector Quantization technique. So Split Vector Quantization technique is superior to Unconstrained Vector Quantization technique in terms of the computational complexity and memory requirements.

In a Split Vector Quantizer of n dimension, SP splits, operating at b bits per vector. The vector space R^n will be splitted into SP subspaces or splits or parts of lower dimension then the dimension of each subspace will be n_i . The number of independent quantizers will be equal to the number of splits and the bits used for quantization are divided among the splits and is b_i . Where b_i is the number of bits allocated to each vector quantizer.

The computational complexity of a Split Vector Quantizer is given by equation (4. 27)

Where

n_i is the dimension of a sub-vector in i th split

b_i is the number of bits allocated to the i th split of a quantizer

sp is the number of splits.

The Memory requirements of a Split Vector Quantizer is given by equation (4. 28)

Multistage Vector Quantization is a modification of Unconstrained Vector Quantization technique. It is also called as Multistep, Residual or Cascaded

<https://assignbuster.com/quantization-process/>

Vector Quantization. Multistage Vector Quantization (MSVQ) technique preserves all the features of Unconstrained Vector Quantization technique and decreases the computational complexity, memory requirements and spectral distortion when compared to it. When compared to Split Vector Quantization technique, Multistage Vector Quantization technique shows significant improvement in the quality of the speech signal, by decreasing the spectral distortion, but at the expense of increased computational complexity and memory requirements. This is because Split Vector Quantization technique deals with vectors of lower dimensions while Unconstrained and Multistage Vector Quantization techniques deal with vectors of larger dimensions. So the complexity and memory requirements are less for Split Vector Quantization technique.

Multistage Vector Quantizer is a cascaded connection of several vector quantizers, where the output of one stage is given as an input to the next stage and the bits used for quantization are divided among the stages connected in cascade [12, 14]. As a result the computational complexity and memory requirements get reduced when compared to Unconstrained Vector Quantizer. The generation of codebooks at different stages of a three stage MSVQ is shown in Fig 4. 6.

From Fig 4. 6 it can be observed that the codebook at the first stage is generated by taking the training sequence as an input. At the second stage the codebook is generated using the quantization errors of the first stage, likewise the codebook at the third stage is generated using the quantization errors of the second stage. This process is continued for the required number of stages [76-80].

The block diagram of a three stage Multistage Vector Quantizer is shown in Fig 4. 7.

Its implementation requires the design of vector quantizers at each stage. In Multistage Vector Quantization the input vector 's' to be quantized is passed through the first stage of the vector quantizer so as to obtain the quantized version of the input vector i. e., 's₁'. The quantization error or residual error at the first stage will be computed which is the difference of the input vector and the quantized version of the input vector. The quantization error at the first stage is given as an input to the vector quantizer of the second stage so as to obtain the quantized version of the error vector at the first stage i. e., 'e₁'. Likewise the quantization error at the second stage will be given as an input to the vector quantizer of the third stage so as to obtain the quantized version of the error vector at the second stage i. e., 'e₂', and this process can be continued for the required number of stages. Finally the decoder takes the indices l_i from each quantizer stage and adds the corresponding codeword's to obtain the quantized version of the input vector i. e., [54].

In a Multistage Vector Quantizer each stage acts as an independent vector quantizer and the total bits available for vector quantization will be divided among the stages. Then the complexity of a particular stage becomes, where b_j is the number of bits allocated to the jth stage. This is less than the complexity of Unconstrained Vector Quantizer. Likewise the memory requirements at each stage are , which is less than the memory requirements of Unconstrained Vector Quantizer.

The computational complexity of a Multistage Vector Quantizer is given by equation (4. 29)

Where

n is the dimension of the vector

b_j is the number of bits allocated to the j th stage

P is the number of stages

The Memory requirements of a Multistage Vector Quantizer is given by equation (4. 30)

In order to improve the performance of Multistage Vector Quantization and Split Vector Quantization techniques a hybrid product code vector quantization technique called Split-Multistage Vector Quantization technique has been developed. Split-Multistage Vector Quantization technique is a hybrid of Multistage Vector Quantization and Split Vector Quantization techniques. At reasonable improvement in the quality of the output speech signal, Split-Multistage Vector Quantization technique provides the lowest spectral distortion, computational complexity and memory requirements when compared to Unconstrained Vector Quantization, Multistage Vector Quantization and Split Vector Quantization techniques [73-80]. The decrease in spectral distortion is due to summing of the quantized errors at each stage. In Split-Multistage Vector Quantization the dimension of the vectors to be quantized has been reduced by means of splitting. Likewise the bits used for quantization are also divided among the stages and among the splits of each stage. As a result the computational complexity and memory requirements get decreased when compared to Unconstrained Vector Quantization, Multistage Vector Quantization and Split Vector Quantization techniques. This is due to the decrease in the dimension of vectors, number

of bits used for quantization at each stage and at each split of the vector quantizer.

The generation of the codebooks at each stage of the Split-Multistage Vector Quantizer is similar to the codebooks generation at each stage of the Multistage Vector Quantizer. But the difference is that each stage of the Split-Multistage Vector Quantizer involves the generation of several sub codebooks. The number of sub codebooks generated at each stage is equal to the number of splits at that stage. In this work, Split-Multistage Vector Quantizer with three parts (splits) and three stages have been developed. The performance of quantization depends on the number of stages and on the number of splits at each stage. As the number of stages increases the quality of the quantized output can be increased, but there must be a limit on the number of stages and on the number of splits at each stage as the number of bits at each stage is limited. The allocation of the bits at each stage is shown in Table 4. 1 and the allocation of bits to each split of a stage is shown in Table 4. 2. From Tables 4. 1 and 4. 2 it can be observed that the minimum number of bits at each stage with three parts must be at least three. So with three parts (splits) and three stages, in Split-Multistage Vector Quantizer the number of bits at a frame cannot be reduced below 9 bits.

The block diagram of a Split-Multistage Vector Quantizer with three parts and three stages is shown in Fig 4. 8. The block diagram is similar to three stage Multistage Vector Quantizer except for the splits at each stage. In Split-Multistage Vector Quantizer each split is treated as a separate vector quantizer and the vectors at each split are quantized independently. The quantization mechanism involved in Split-Multistage Vector Quantizer is

similar to the quantization mechanism involved in Multistage Vector Quantizer, except that in Split-Multistage Vector Quantizer at each stage the sub-vectors are quantized independently.

Split-Multistage Vector Quantizer is a hybrid of Split Vector Quantizer and Multistage Vector Quantizer. The equations for computational complexity and memory requirements are derived from the complexity and memory requirement equations of a Split Vector Quantizer and Multistage Vector Quantizer. Equations (4. 31) and (4. 32) below are obtained from the equations (4. 29) and (4. 30) by including the splits (SP) at each stage with a summation term having limits from 1 to SP.

The computational complexity of a Split-Multistage Vector Quantizer is given by equation (4. 31)

Where

n_{ji} is the dimension of a sub-vector in j th stage i th split

b_{ji} is the number of bits allocated to the j th stage and i th split of a quantizer

P is the number of stages

sp is the number of splits.

The Memory requirements of a Split-Multistage Vector Quantizer is given by equation (4. 32)

Switched Split Vector Quantization (SSVQ) is one of the latest vector quantization techniques and is developed to improve the performance of

Split Vector Quantization technique. Switched Split Vector Quantization technique is a hybrid of Switch Vector Quantization and Split Vector Quantization techniques and is used to exploit the linear and non linear dependencies that exist between the splits of a Split Vector Quantizer. In Switched Split Vector Quantizer initially the Switch Vector Quantizer partitions the entire vector space into voronoi regions and exploits the dependencies that exist across all dimensions of the vector space. Then a Split Vector Quantizer is designed for each of the voronoi regions. As a Split Vector Quantizer is adapted to the local statistics of the Voronoi region the sub optimality's of the Split Vector Quantizer will be localized. In a Switched Split Vector Quantizer a number of vector quantizers are connected in parallel and it can be implemented in two ways: hard decision scheme and soft decision scheme.

In hard decision scheme each vector to be quantized is quantized in only one of the codebooks connected in parallel, the selection of a codebook for quantization depends on the nearest codeword selected in the initial codebook. An initial codebook is one which is designed for the selection of a switch. The initial codebook is generated by the training vectors used for the generation of the codebooks at the vector quantizers connected in parallel. The number of codeword's or centroids in the initial codebook is equal to the number of switches chosen or number of codebooks connected in parallel and these centroids are used to form the Switch Vector Quantizer.

In soft decision sc