9.4 Below are the steps to follow for the implementation of the LPC coder. First, the encoder is constructed without the quantizers (parameter encoders); then, the decoder is constructed without the parameter decoders. Functionality of the encoder/decoder constructed in this way is tested; if the system is working properly, the parameter encoders and decoders are incorporated to evaluate the final system.

1. *Encoder.*
   - Pre-emphasis filter: This is a first-order difference equation (Chapter 4).
   - Voicing detector: Before exploring a multiparameter detector, a simple detector based on just one parameter, like the zero crossing rate, can easily be implemented. In this case, find out the threshold value that seems to do a reasonably good job in voiced/unvoiced classification by using some speech material.
   - LP analysis: See Chapter 4 for details regarding autocorrelation estimation and algorithms to solve the normal equation. Prediction order is equal to ten.
   - Prediction-error filter and power computation: The filter is a difference equation with coefficients given by LP analysis. Power computation makes use of the prediction error.
   - Pitch period estimation: The magnitude difference function (Chapter 2) can be used.
   - Integration: The blocks from previous steps are joined together following the diagram of Figure 9.6. However, no quantization or encoding of an individual parameter is considered yet. The decoder is first built to perform a system test.
2. *Decoder.*
   - Impulse train generator: Given a period value, a train of impulses is generated, with each impulse having unit amplitude. The instant when the first impulse appears can be adjusted for consecutive voiced frames so that variation of the pitch period across frame boundaries is minimized.
   - White noise generator: This is essentially a random number generator. Uniform distribution is appropriate. Most programming languages have a built-in random number generator. Many issues exist in the design of a random number generator; as a reference, see Banks and Carson [1984]. To reduce computational load, an array of random numbers can be used; that is, a long sequence of random numbers is stored in memory as an array. To generate the sequence of white noise, one single random number is created and is used to index the first element of the sequence from the array. That is, instead of generating N numbers, only one number is generated per frame. This method was found to work well in practice.
   - Synthesis filter: This is a tenth-order difference equation.

- De-emphasis filter: This is a first-order difference equation.
- Integration: From the diagram in Figure 9.8, join the different blocks to form the decoder.

3. *Test of encoder–decoder operation*.
   - Using the system developed so far, speech is input to the encoder with the resultant parameters processed by the decoder. Verify the intelligibility of the synthetic speech. Note that this step is done without any quantization of individual parameters. If the quality of synthetic speech is as expected, proceed to the next steps. Otherwise, review the different blocks to discover the problem.
4. *Incorporation of parameter-quantizers*.
   - LPC: See Chapter 8 for options and ideas. You can opt for the same technique as used by the FS1015 or methods adopted by other standards, as well as creating your own scheme.
   - Power: Uniform quantization can be used for a first test. However, nonuniform quantization will definitely provide higher quality. This can be designed using the Lloyd algorithm with training data collected from real speech signals (Chapter 5).
   - Pitch period: Many options are possible. For instance, with 7-bit encoding, 128 values of pitch period are covered; the interval of [20, 147] is reasonably good in practice. A scheme similar to FS1015 can also be applied.
   - Integration (encoder): Encoders for the various parameters can be integrated into the encoder according to Figure 9.6; outputs are packed together as the LPC bit-stream.
   - Integration (decoder): Decoders for the various parameters are integrated to the decoder according to Figure 9.8. The LPC bit-stream is unpacked with the indices directed to the right places.
5. *Overall test and system improvement*.
   - The system is tested again for correct operation and quality. For quality improvement, more bits can be allocated to a particular quantizer, at the expense of a higher bit-rate. After the basic system is working properly, the different blocks can be modified individually. For instance, a more sophisticated voicing detector using multiple parameters can be developed or a better LPC quantization scheme can be utilized.

9.5 A 130-sample block is selected for pitch-synchronous LP analysis within each 180-sample voiced frame, specified with $n \in [0, 179]$. The 130-sample block is indicated with a starting position $n_o \in [-20, 50]$; that is, the interval on which LP analysis is performed is given by $n \in [n_o, n_o + 129]$. The starting position is generally found in such a way that consecutive frames rely on approximately the same cycle of the waveform to perform LP analysis. In this way, synchronization is maintained. Design an algorithm to determine n_o based on peak location. Propose alternative realizations to boost robustness.

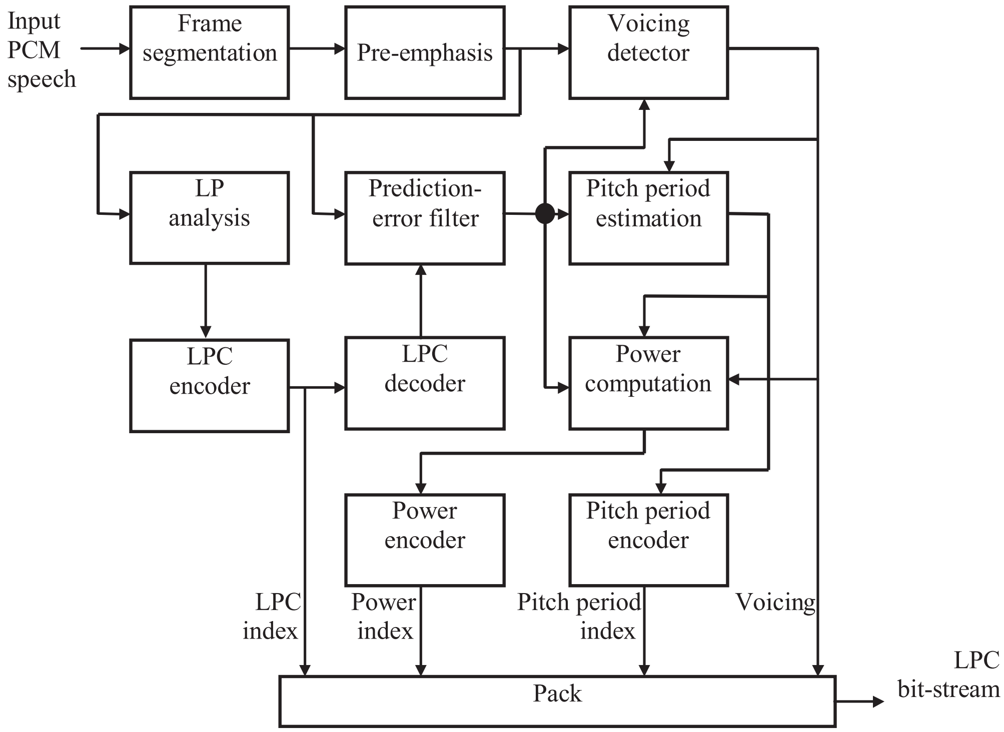

Figure 9.6 Block diagram of the LPC encoder.

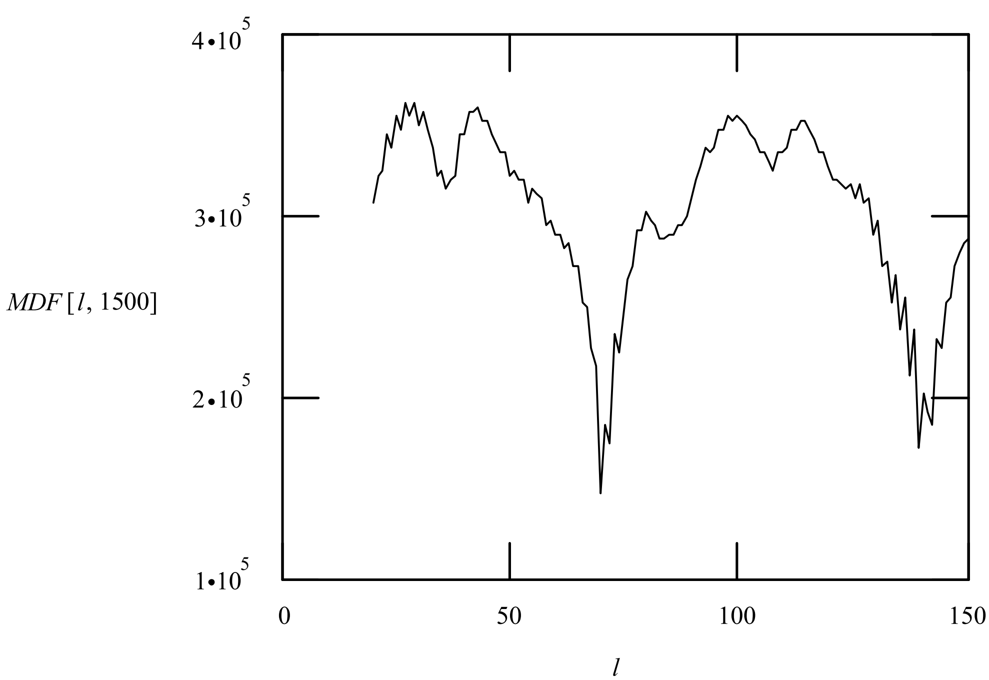

Figure 9.7 Magnitude difference values obtained from a voiced frame. Prediction error is used for its computation.