APPENDIX 8

VOICE MODIFICATION AND SYNTHESIS

A8.1 INTRODUCTION

A speech synthesis procedure is outlined in Chapter 6 and discussed in Appendix 7. Chapter 7 describes the software that implements this system. This appendix provides some additional background material on the algorithms used in the software for the voice conversion and synthesis system.

A8.1.1 Pitch Detection and Glottal Closure Instants

A simple classification algorithm for voiced/unvoiced decision is the following. The energy of the prediction error and the first reflection coefficient are used to classify a segment as voiced. The first reflection coefficient is

$$r_1 = \frac{R_{SS}(1)}{R_{SS}(0)} \qquad \text{(A8.1.1.1)}$$

where

$$R_{SS}(0) = \frac{1}{N}\sum_{n=1}^{N} s(n)s(n)$$

$$R_{SS}(1) = \frac{1}{N}\sum_{n=1}^{N-1} s(n)s(n+1)$$

where N is the number of samples in the analysis frame and s(n) is the speech sample.

The decision rules are as follows.

- If the first reflection coefficient is greater than 0.2 and the prediction error energy is greater than twice the threshold, e.g., 10^7, then the current frame is classified as voiced.
- If the first reflection coefficient is greater than 0.3 and the prediction error energy is greater than the threshold used in rule 1 and the previous frame is also voiced, then the current frame is classified as voiced.
- If the above conditions are not valid, then the current frame is classified as unvoiced.

The above algorithm generates a sequence of 1s and 0s. Patterns of 101 and 010 seldom occur in real speech and are corrected to strings of 111 and 000, repectively, to reduce the classification error rate.

The two-pass glottal closure instant detection algorithm in Childers and Hu (1994) is outlined below. This method uses the results from both a voiced/unvoiced classification procedure and the prediction error waveform, e(n), to detect the pitch period and the glottal closure instants (GCIs). The detection algorithm consists of two stages: pitch period estimation and peak picking.

Pitch Period Estimation

- Low-pass filter one segment of the predicition error waveform, e(n). The filtered waveform is denoted as $e_{Lp}(n)$.

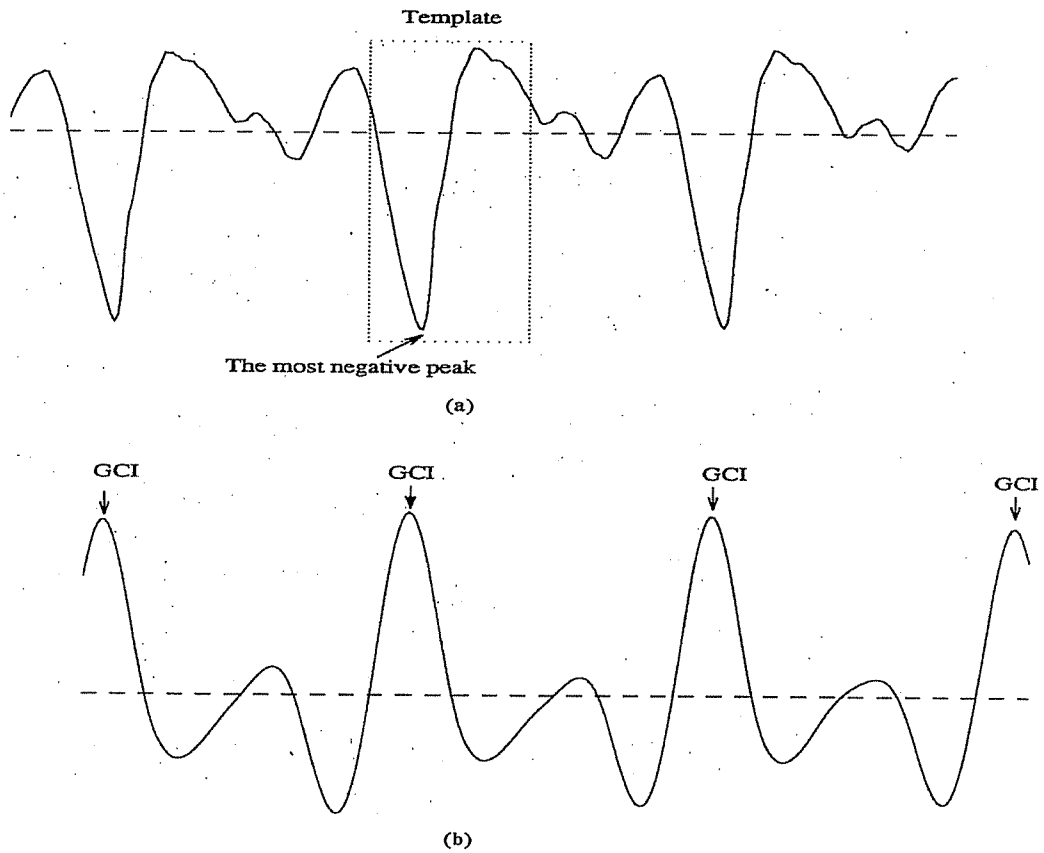

FIGURE A8.1 Illustration of the pitch period and GCI detection algorithm. (*a*) The filtered prediction error sequence, $e_{Lp}(n)$. (*b*) The correlation output sequence, $C_{te}(n)$.

- Calculate the cepstrum-like sequence, $C_e(n)$.

$$C_e(n) = \text{IFFT}(|\text{FFT}\, e_{LP}(n)|) \quad 1 \leq n \leq N \tag{A8.1.1.2}$$

  where N is the frame size, FFT is the fast Fourier transform, and IFFT is the inverse FFT.
- Search for the index m, where $C_e(m)$ is the maximum amplitude in the subset $\{C_e(i) \mid 25 \leq i \leq N\}$.
- Search for the index k, where $C_e(k)$ is the maximum amplitude in the subset $\{C_e(i) \mid 25 \leq i \leq m - 25\}$.
- If $C_e(k) > 0.7C_e(m)$, k is the estimated pitch period, otherwise m is the estimated pitch period.
- If an abrupt change in the pitch period is observed, compared to previous pitch periods, then low-pass filter (or median filter) to smooth the abrupt change.

Peak Picking

- In each analysis frame (256 samples), search for the most negative peak of the $e_{Lp}(n)$ waveform.
- Build a template as illustrated in Figure A8.1(*a*). This template is formed by the waveform around the most negative peak of $e_{Lp}(n)$. The length of the template is 46 samples, including 15 samples before the peak, 30 samples after the peak, and the peak itself.
- Correlate the template with the $e_{Lp}(n)$ waveform to generate a new sequence, $C_{te}(n)$, as shown in Figure A8.1(*b*).
- The positive peaks of the $C_{te}(n)$ sequence provide initial estimates for the GCIs. The estimated pitch period from the pitch period estimation (stage 1) can assist in correcting the erroneous peak detection.

- Adjust the position of each GCI under the criterion that no two GCIs are located within 25 samples of on another.

A8.2 PITCH CONTOUR MODIFICATION

The glottal closure instants (GCIs) are measured in a manner as described previously. Then the GCIs are sorted into a vector, which is a GCI sequence. The distance between the GCIs is the length of the glottal pulse or the pitch period. For pitch synchronous analysis and synthesis, these instants determine the timing for the generation of the glottal pulses as well as the times at which the vocal tract parameters are to be updated. The method for modifying the pitch contour is one that alters or modifies the GCI vector. This modification is one factor, if not the major factor, for creating or mimicking a voice type. For example, changing the length of the GCI sequence alters the fundamental frequency of voicing of the synthesized speech. Furthermore, if we alter segments of the GCI sequence, then we can alter the intonation pattern. For example, by sequentially decreasing the distance between successive GCIs in the sequence, one can synthesize speech that has a rising intonation. This is the method adopted here; that is, the pitch contour is altered by altering the GCI sequence.

A8.2.1 Pitch Contour Model

A plot of the GCI sequence is the pitch period contour. This plot is constructed with the horizontal axis being the GCI points, while the vertical axis is the pitch period; that is, the interval between successive GCIs. An example is shown in Figure A8.2. The top two panels show a segment of the sentence, "We were away a year ago," and the corresponding pitch contour for that segment. The third panel from the top shows the waveform for the sentence, while the fourth panel shows the pitch contour for the entire sentence. One can model the pitch contour with three waveforms. One waveform models the average value of the pitch contour, which is a constant over the sentence and is the fundamental pitch period. Another model waveform is the steady state or long time variation of the pitch contour, which is called the pitch wave. The pitch wave is related to the intonation. A third model waveform is the short time perturbation of the pitch contour, which is the jitter. The three waveform models are illustrated in Figure A8.3. One advantage of this model is that each waveform is related to certain perceptual features and each waveform can be independently controlled or modified. For example, the fundamental pitch period can be altered without affecting the pitch wave or the jitter.

The three pitch waveform models can be measured as follows. The fundamental pitch period is the mean value of the pitch contour. The pitch wave is estimated by a 5th-order median filter after subtracting the pitch contour from the fundamental pitch period. The jitter is the standard deviation of the difference between the pitch contour and the sum of the pitch wave and the fundamental pitch period.

Several approaches have been taken to model the pitch contour (Fujisaki, 1983; Maeda, 1974; Mattingly, 1966). One method models the contour as consisting of three parts or patterns: a falling waveform, a rising waveform, and a baseline trend. Each of these patterns can be fit with a second-order polynomial using a least squares approach to find the polynomial coefficients. Another approach is to fit a third-order spline function to the pitch contour. Both approaches are available in the software.

In summary, the pitch contour model is estimated as follows.

- Transform the GCI sequence to the pitch contour. Calculate the mean value, which is the fundamental pitch period.
- Smooth the pitch contour with a 5th-order median filter and subtract the result from the fundamental pitch period. This is the pitch wave.
- Subtract the original pitch contour from the sum of the pitch wave and the fundamental pitch period. Calculate the standard deviation of the resultant sequence. This is the jitter.
- Segment the pitch wave into three pitch patterns: rising, falling, and baseline trend. Approximate each pattern with a second-order polynomial or a third-order spline function.

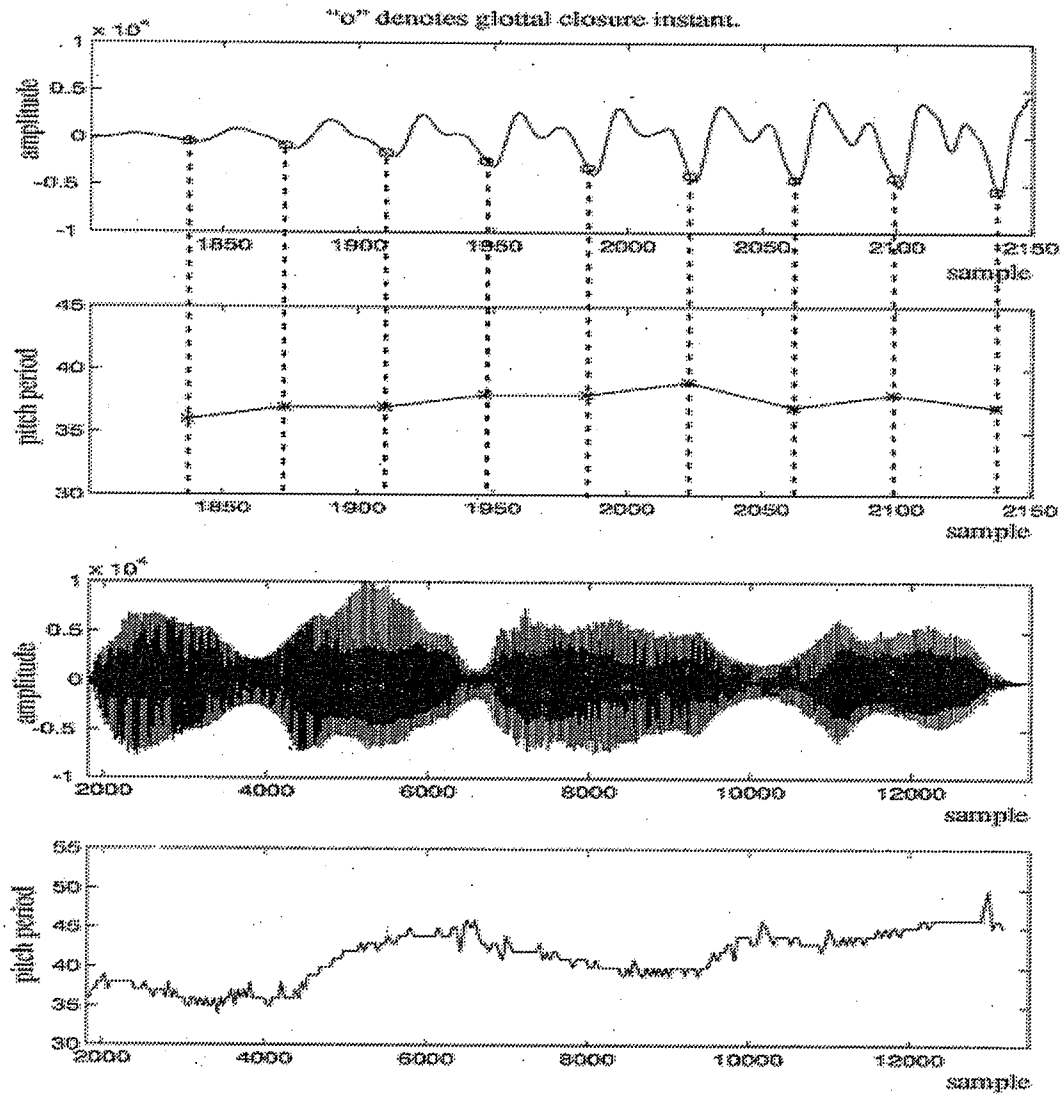

FIGURE A8.2 Speech signal and the corresponding pitch contour.

A8.2.2 Pitch Contour Modification

The pitch contour model is modified as follows.

- Fundamental pitch period. Scale this value upward or downward.
- Pitch wave. Segment the pitch wave manually into several pitch patterns, with each pattern modeled by a polynomial or a spline. Modify each pattern by altering the coefficients of the model.
- Jitter. Scale this factor upward or downward.

A8.2.3 Synthesis

The procedure for synthesis is as follows.

- Form the fundamental pitch period as a contour for a selected (voiced) segment or for the entire sentence.
- Add the jitter to the contour.
- Add the pitch wave to the contour. This forms the modeled pitch contour.

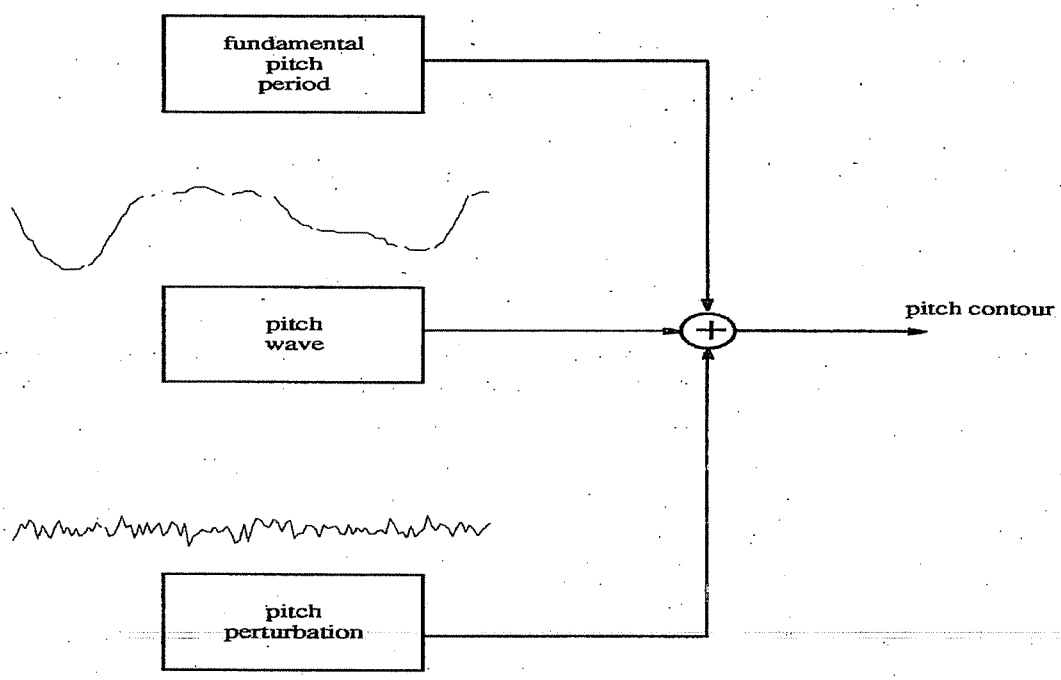

FIGURE A8.3 The pitch contour model.

- Construct a GCI sequence from the pitch contour model. Initialize the first GCI with its pitch period. This is done by sampling the pitch contour model at the first point (or a desired first point). Call this value GCI(1) = T_1. The next GCI is located at $(1 + T_1)$ with a value GCI$(1 + T_1) = T_2$, which is sampled from the pitch contour model. The next GCI is located at $(1 + T_1 + T_2)$ with a value GCI$(1 + T_1 + T_2) = T_3$, and so on. Continue this process until the new GCI sequence is constructed for the selected voiced speech segment. This new GCI sequence (vector) is used to synthesize a new voice or to convert one voice to another.

These algorithms are implemented in the software provided.

A8.3 GAIN CONTOUR MODIFICATION

The gain parameter is the average value of the speech energy for each pitch period. Its function is to control the energy transitions through out an utterance. This parameter is related to the intensity (or loudness) of the speech signal. As the loudness of the speech increases, the gain increases. The software provides a mechanism for controlling the gain to create or alter various voice types.

A8.3.1 Analysis

The gain parameter is pitch synchronous and has a contour, which is defined as the value of the gain versus its GCI location along the time axis. The gain contour is calculated using pre-emphasized speech. The gain contour can be divided into two factors or models, analogous to the pitch contour model. One model is the gain envelope, which is the smoothed envelope of the voiced speech segments. The second model is the gain perturbation, or the pitch period-to-pitch period variability of the gain. This model is related to shimmer. These two models, in combination, form the gain contour, and are estimated as follows.

- Construct the gain contour using the gain parameter.

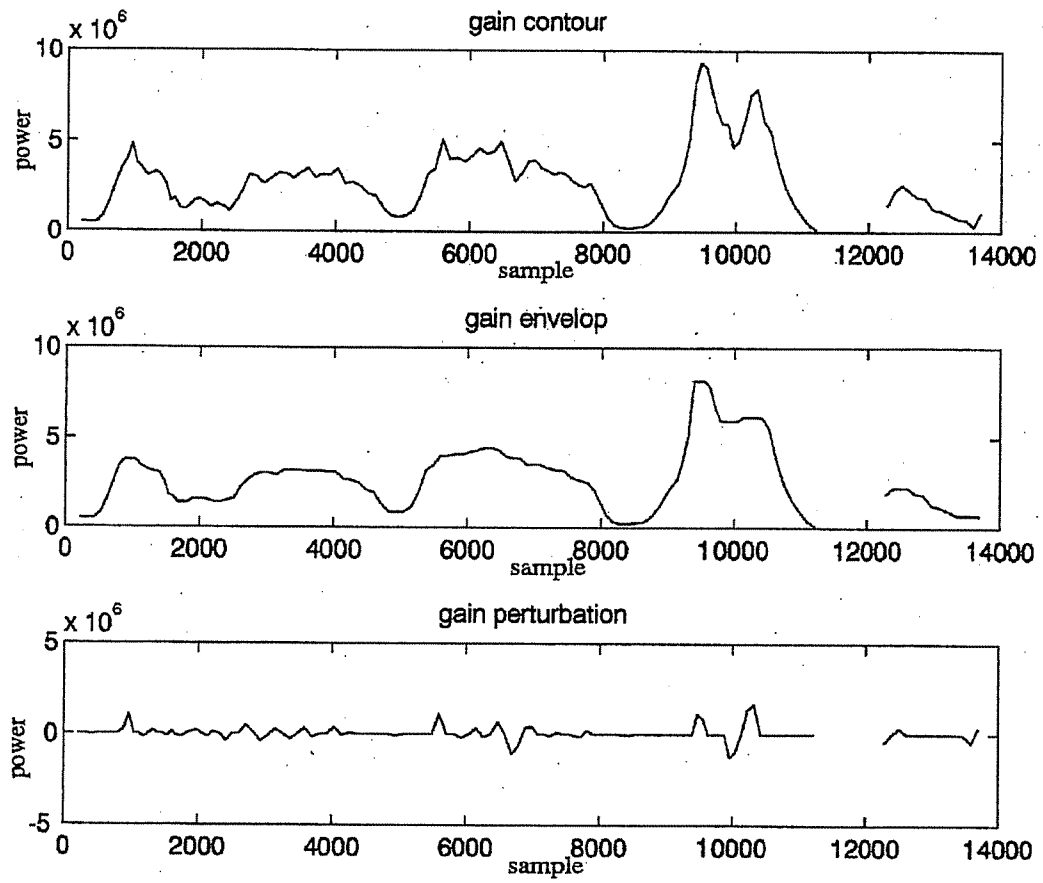

FIGURE A8.4 An illustration of the gain contour and its models (factors).

- Smooth the gain contour with a 5th-order median filter. The result is the gain envelope.
- Subtract the original gain contour from the gain envelope. The standard deviation of the resultant difference is the gain perturbation.

Figure A8.4 shows the gain contour for a sentence and its corresponding gain envelope and gain perturbation models.

A8.3.2 Modification of Gain Models

The gain models are modified independently as follows.

- The gain envelope. Segment the gain envelope manually into one to three gain patterns in an analogous manner as to that used for the pitch patterns. Model the patterns with a second-order polynomial or a third-order spline.
- The gain perturbation. Scale this model upward or downward.

A8.3.3 Synthesis

The gain contour is synthesized in a manner analogous to that used for the pitch contour. Interpolation of the gain contour may be required for the new GCI sequence. If the pitch contour is modified, then the gain contour must be modified to avoid discontinuities in the synthesized speech.

A8.4 VOCAL TRACT MODIFICATION

The formants and their bandwidths are found by computing the roots of the filter polynomial. This is done frame-by-frame to obtain the formant tracks. The formant tracks can be modified by a scale factor, or by using the mouse to draw a new track, or by loading a file that contains a desired track. The vocal tract filter is calculated from this information. However, this calculation can result in a filter design with improper pole positions. For example, the second formant may incorrectly merge with the first formant. This is called the pole interaction problem. This is solved using an algorithm presented in Hsiao and Childers (1996). This algorithm is included in the software.

A8.5 GLOTTAL PULSE MODIFICATION

There are two models for the glottal pulse: the polynomial model and the LF model. It is difficult to do glottal pulse waveform design using the coefficients of the polynomial model because there is little physical relationship between the volume–velocity waveform and the polynomial model. For voice conversion, the procedure recommended is to model the volume–velocity waveform of both the target and source speakers with separate polynomial models. Then, when converting the source speech to the target speech, use a linear mapping of the polynomial parameters of the source speaker to that of the target speaker using linear regression. In this way, the glottal parameters of the source speaker are mapped (modified) to match those of the target speaker. For the LF model, the source speaker model parameters can be modified either by using a linear regression approach like that described above or by using the graphic user interface provided in the software.

A8.6 VOICE CONVERSION

Voice conversion is the process of transforming the speech of one speaker to sound like that of another speaker. The objective is to develop methods for creating new synthetic voices, study factors responsible for synthetic voice quality, and to determine methods for speaker adaptation.

A8.6.1 Speaker Translation Models

The approach taken here is to model speaker characteristics with parametric models. To alter or convert the speech of one speaker to that of another, the parameters of one speaker (the source) are mapped to match the parameters of another speaker (the target). This is accomplished as follows.

A phoneme can be represented by an n-dimensional acoustic vector. The mean of this phoneme vector over various speakers is denoted as μ. The phoneme for speaker i is denoted as vector s^i and is given as

$$s^i = \mu + \delta \qquad \text{(A8.6.1.1)}$$

where the vector δ is a "bias" considered to be a characteristic of the speaker, and thus the name for this model.

Assuming that the acoustic features are independent and time invariant, then one can convert the acoustic parameters of one speaker to those of another provided the offset value between the two speakers is known, that is

$$X = Y + B \qquad \text{(A8.6.1.2)}$$

where X and Y are the n-dimensional acoustic parameter vectors for the target and source speakers, respectively. B is the offset vector and n is the number of measured acoustic features. This is called the translation model.

The task is to estimate B, which is the difference between the two speakers, including any channel effect such as a telephone line characteristic. Equation (A8.6.1.2) can be written as a set of

equations

$$\begin{aligned} x_1 &= y_1 + b_1 \\ x_2 &= y_2 + b_2 \\ &\vdots \quad\quad \vdots \\ x_n &= y_n + b_n \end{aligned} \tag{A8.6.1.3}$$

where x_i and y_i are the ith acoustic features for the target and the source speakers, respectively, and b_i is the offset scalar. If m samples are available for the two speakers, the estimate of the value of b_i is

$$\hat{b}_i = \frac{1}{m}\sum_{k=1}^{m}(x_{ik} - y_{ik}) \quad i = 1, \ldots, n \tag{A8.6.1.4}$$

where x_{ik} is the kth sample of the acoustic feature x_i, and similarly for y_{ik}.

A8.6.2 Affine Model

Although the translation model discussed in the previous section is simple, the assumption of the model is that a speaker's speech can be modeled as a single invariant transformation. This assumption may not be valid for some situations. A more detailed model is presented here.

As described previously, the acoustic features of one speaker may be modeled as a linear combination of another speaker's features, that is

$$X = AY + B \tag{A8.6.2.5}$$

where X and Y are n-dimensional acoustic feature vectors for the target and source speakers, respectively. A is an n by n matrix and B is an n-dimensional vector. This is known in modern algebra as an affine transformation between the vectors X and Y. If we hypothesize that the acoustic features are linearly independent and the transformation is time invariant, then A is a diagonal matrix, and of course, Equation (A8.6.1.2) is a special case of Equation (A8.6.2.1).

The process for determining the mapping function between X and Y is known as the training process. Due to variations in the speaking rate from one speaker to another, we use dynamic time warping (DTW) to adjust the parameters of the source to be in accord with those of the target along the time axis. A diagram illustrating the training process is shown in Figure A8.5. The DTW algorithm is discussed in Rabiner and Juang (1993).

The training process consists of the following procedures.

- The target and source speakers pronounce the same set of sentences.

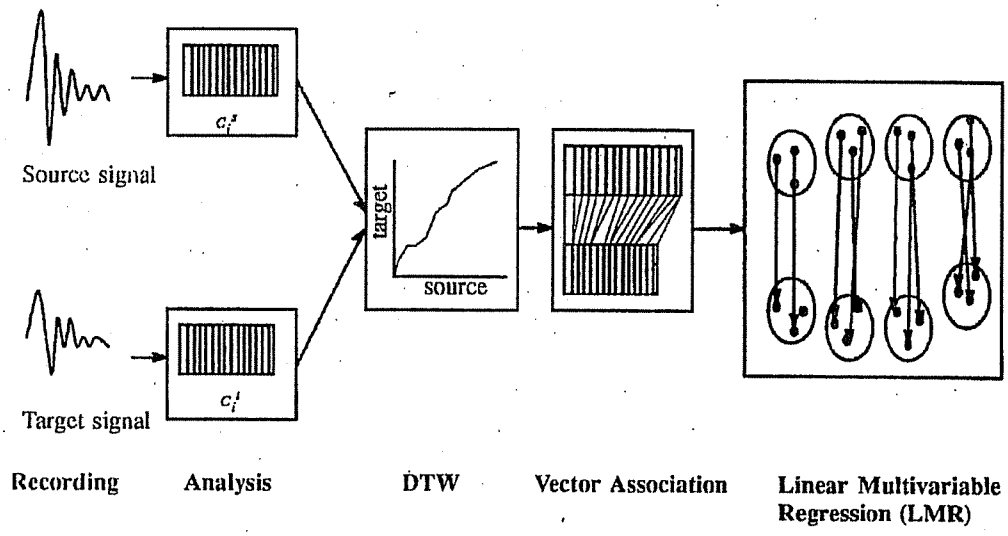

FIGURE A8.5 The training process.

- The acoustic features are measured, providing a set of framed-based vectors.
- The source vectors are time aligned with the corresponding target vectors by DTW.
- Linear multiple regression (LMR) is used to estimate the coefficients of the mapping function.

A8.7 VOICE CONVERSION ALGORITHMS

The algorithms described here focus on modification of the segmental parameters of the speech signal. We modify five measured acoustic features so that the voice conversion process is simulated with a parameter mapping process. This is illustrated in Figure A8.6.

We offer four types of mapping methods: the translation transformation, the affine transformation, the copy method, and the retain method. These mapping methods may be intermixed, that is, one method may be used for the pitch contour, another for gain contour, and so on. The copy method uses the acoustic features of the target speaker to synthesize the converted speech. This method can serve as a basis for examining the effectiveness of the translation and affine methods. The retain method retains the source parameters so that the synthesized speech contains certain features of the source speech. However, the source parameters are warped in accord with the target's speaking rate.

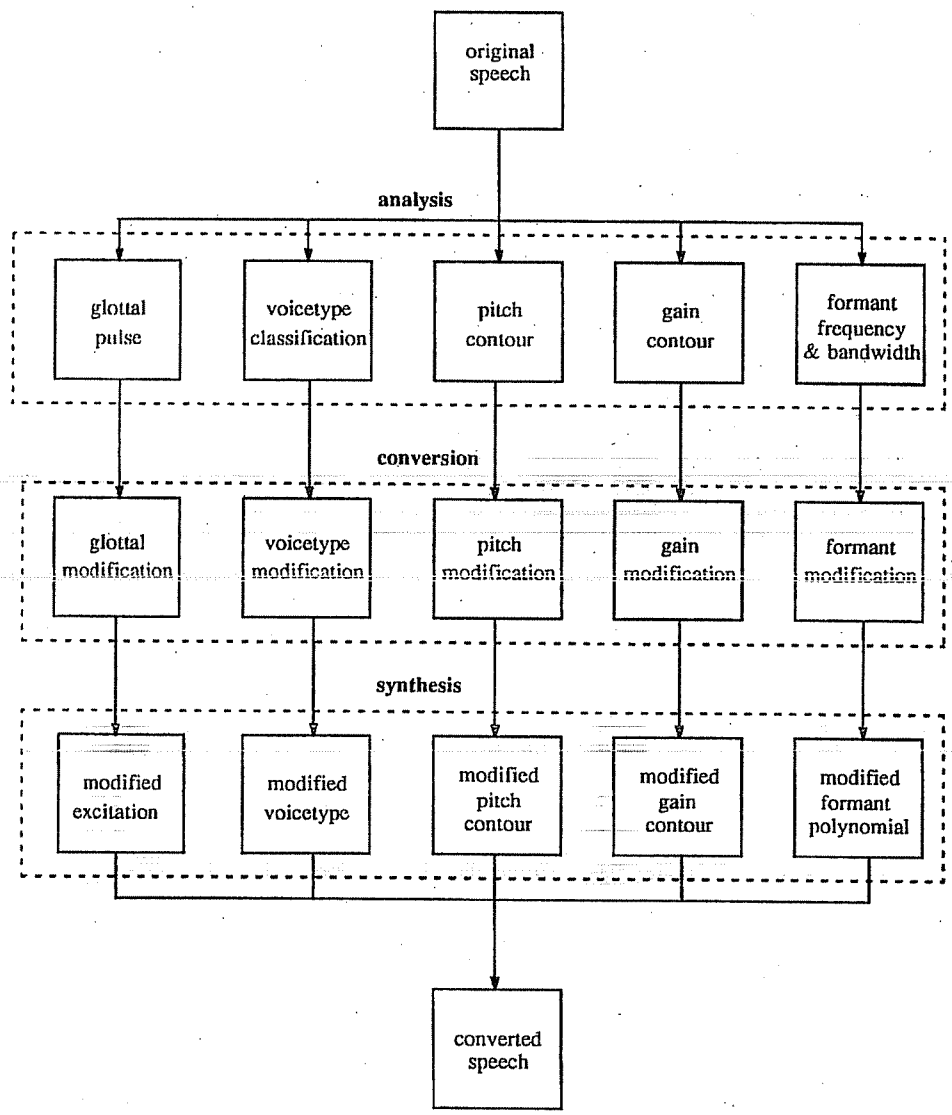

FIGURE A8.6 The voice conversion process.

A8.7.1 Overview of Conversion Algorithms for Each Acoustic Feature

- Voice type conversion: For this process, the voiced parameters (pitch, gain, glottal pulse, and formants) are transformed, while the unvoiced parameters (gain, stochastic codeword, and LP coefficients) are copied from the target to eliminate unwanted noise in the unvoiced segments.
- Pitch contour conversion. For transformation, the average value of the pitch period of each frame is used for the source and target. After training, the source vector is converted to a new value using a specified mapping function. Since the synthesizer is pitch synchronous, the pitch vector is transformed to the timing instants that correspond to the glottal closure instants (GCIs). To eliminate the unvoiced-to-voiced transition noise, the first GCI is fixed at the beginning of the voiced segment. For the voiced-to-unvoiced transitions, the last GCI is extended to the next unvoiced segment. In the voiced/unvoiced transition region, the voiced speech is overlapped with and added to the unvoiced speech.
- Gain contour conversion. The gain parameter controls the excitation energy for each pitch period and is pitch synchronous. Thus, the frame-based gain data must be interpolated for each GCI after transformation. Furthermore, once the pitch contour is modified, the gain contour must be modified, although the gain contour itself is not changed. To eliminate any discontinuities in the voiced-to-unvoiced transition, the gain value for the first voiced pulse is linearly interpolated between the last unvoiced gain and the second voiced gain; and vice versa.
- Glottal pulse conversion. The LF timing parameters (t_p, t_e, t_c, t_a) have constraints as described in Appendix 7. If the user tries to override these constraints, the synthesizer software uses either the previous LF timing parameters or the average values calculated for the entire speech record.
- Formant frequency transformation. Only the formant frequencies are converted by the linear mapping function. The bandwidths of the formants are determined by the algorithm we developed to counter the pole interaction problem (Hsiao and Childers, 1996). Unlike other transformation methods, our algorithm is independent of the pitch contour and the speaking rate (Childers et al., 1989).

A8.8 SUMMARY COMMENTS

From experimental results using the voice conversion system, it is concluded that the quality of the synthesis using the LP synthesizer is superior to that using the formant synthesizer. The implication is that the analysis algorithms for extracting the formants and their bandwidths is not as accurate as needed. The affine transformation method is superior to that using the simpler translation method. This is especially so for male-to-female and female-to-male voice conversion tasks. Generally, it is easier to convert words than it is to convert entire sentences. This is probably due to the fact that there are more dynamic changes over a sentence, than over a word. The gain contour conversion is nearly the same for both the translation and affine transformation methods. The glottal pulse transformation is a critical factor in achieving high-quality voice conversion. Generally, the quality of female synthesized speech is inferior to that for male synthesized speech. Possibly, this is due to the fact that female speech has a higher fundamental frequency than male speech, and therefore there may be more interaction of the fundamental frequency with the first formant in female speech than in male speech.

Consult Hsiao (1996) for additional details on the algorithms and experiments and Mizume and Abe (1995) for other voice conversion algorithms.