

SMV와 AMR 음성부호화기를 위한 상호부호화 알고리즘

Transcoding Algorithm for SMV and AMR Speech Coder

이 덕 중*, 정 규 혁*, 이 인 성*
(Duck-Jong Lee*, Gyu-Hyeok Jeong*, In-Sung Lee*)

*충북대학교 전파공학과

(접수일자: 2008년 9월 12일; 수정일자: 2008년 10월 29일; 채택일자: 2008년 11월 17일)

본 논문에서는 SMV와 AMR 음성부호화기를 위한 상호부호화 알고리즘을 제안한다. 각기 다른 통신망에서 음성부호화기간의 효율적인 연동을 위해 가장 간단한 방법인 tandem 방식이 있지만 긴 지연시간과 많은 연산량, 그리고 음질 저하의 문제점들을 해결하기 위해 상호부호화 방법을 사용한다. 제안하는 상호부호화 알고리즘은 LSP (Line Spectral Pairs) 변환, 피치 변환, 그리고 고속 고정 코드북 탐색을 수행한다. 실험 결과, 20~50%의 적은 계산량과 5~10 ms의 지연 시간을 줄이면서도 동등한 음질을 확인할 수 있었다.

핵심용어: 상호부호화, AMR, SMV, tandem, 고속 고정 코드북 검색

투고분야: 음성처리 분야 (2.2)

In this paper, a transcoding algorithm for SMV and AMR speech coder is proposed. In the application requiring the interoperability of different networks, two speech coders must work together with the structure of cascaded connection, tandem. The tandem which is one of the simplest methods has several problems such as long delay, high complexity and the quality degradation due to twice complete encoding/decoding process. These problems can be solved by using transcoding algorithm. The proposed algorithm consists of LSP (Line Spectral Pair) conversion, pitch delay conversion, and fast fixed codebook search. The evaluation results show that the proposed algorithm achieves equivalent speech quality to that of tandem with reduced computational complexity and delay.

Keywords: Transcoding, AMR, SMV, Tandem, Fast fixed codebook search

ASK subject classification: Speech Signal Processing (2.2)

I. 서론

각 통신망에 따라 각기 다른 음성부호화기가 표준화되어 사용되고 있다. 예를 들어 CDMA 시스템에서는 EVRC [1]와 SMV [2]가 사용되고 있으며, WCDMA 시스템에서는 AMR [3]이 사용되고 있다. 최근 서로 다른 통신을 이용하는 데이터 통신망간의 통합과 효율적인 연동이 중요한 과제로 제기되고 있다. 그러나 서로 다른 음성부호화기 표준은 서로 호환이 되지 않기 때문에 A라는 음성부호화기로 부호화된 패킷을 B라는 음성부호화기의 복호화하는 과정이 필요하다.

이를 위한 방법에는 tandem 방식과 상호부호화 방식이 있다. tandem 방식은 연속적으로 부/복호화 과정을 반복하기 때문에 긴 지연시간과 많은 계산량, 그리고 음

질 저하라는 단점을 가지고 있다. 이런 문제점을 해결하기 위해 다른 음성부호화기 간의 상호부호화 방법이 연구되고 있다. 예를 들어, VoIP 방식을 이용하는 G.723.1과 G.729A간의 상호부호화 방법 [4], CDMA망인 EVRC와 VoIP 방식을 사용하는 G.729A간의 상호부호화 방법 [5], 그리고 CDMA망인 EVRC와 WCDMA망인 AMR간의 상호부호화 방법 [6]들과 같이 서로 다른 통신망의 효율적인 연동을 위해서 연구되었다. 위의 연구들에서 사용된 음성부호화기들은 선형예측분석 후 잔여신호를 CELP 방법으로 모델링한다. 따라서 LSP변환이나 피치와 코드북 검색과 관련한 방식들이 공통적으로 포함된다. 또한 위의 연구들의 공통적인 목적은 tandem 방식을 사용했을 때의 음질과 비슷하면서 계산량과 지연 시간을 감소시키는 데에 있다.

본 논문에서는 북미 CDMA 망에서의 SMV (Selective Mode Vocoder)와 GSM망에서의 AMR (Adaptive Multi-Rate)을 위한 상호부호화 알고리즘을 제안한다. SMV

[2]는 eX-CELP (eXtended Code Excited Linear Prediction) 를 기반으로 하고 있으며, AMR [3]은 ACELP (Algebraic Code-Excited Linear Prediction) 기반으로 하고 있다. 제안된 알고리즘은 LSP (Line Spectral Pair) 변환, 피치 지연 변환, 고속 고정 코드북 (Fixed Codebook) 검색, 그리고 전송률 결정으로 이루어져 있다. 기존의 연구와 달리 계산량을 크게 감소시킨 고속 고정 코드북 검색방법을 제안하고 적용하였다.

본 논문의 구성은 다음과 같다. II장에서는 SMV와 AMR 음성부호화기에 대해서 설명하며, III장과 IV장에서는 제안된 알고리즘에 대해서 자세히 설명한다. 그리고 V장에서는 성능 평가에 대해 알아보고, 마지막으로 VI장에서 결론으로 끝을 맺는다.

II. SMV와 AMR 음성부호화기

SMV 음성부호화기 [2]는 8.55 kbps (Rate 1), 4.0 kbps (Rate 1/2), 2.0 kbps (Rate 1/4) 그리고 0.8 kbps (Rate 1/8)의 전송률을 가지는 다중 전송률 모드의 부호화기이다. 이들 모드는 네트워크 상황에 의해 결정되며, 프레임마다 다른 모드를 선택할 수 있기 때문에 평균 전송률과 음질의 균형을 적절히 조절할 수 있다. 각 프레임에 대한 전송률은 전송률 결정 알고리즘 (Rate Decision Algorithm, RDA)에 의해 결정된다. SMV 음성부호화기는 결정된 전송률에 따라 서로 다른 방식으로 LPC 합성 필터를 위한 여기 신호를 계산한다. Rate 1/4의 여기 신호 (Excitation Signal)는 난수 발생기 (Random Number Generator)에 의해 생성된 후, 2 ms의 부프레임마다 이득을 곱하고 주파수 변형 필터 (Frequency Shaping Filter)를 통과시켜 얻어진다. Rate 1/8은 난수 발생기에 얻어진 신호에 하나의 이득 값을 곱하여 여기 신호가 생성된다. Rate 1과 Rate 1/2의 여기 신호 계산은 eX-CELP 방식으로 이루어진다.

AMR 음성부호화기 [3]는 4.75 kbps에서 12.2 kbps까지 8개의 모드를 지원한다. 모든 모드에서 공통적으로 선택형 예측 분석 후 여기 신호는 적응 코드북 벡터와 고정 코드북 벡터에 각 코드북의 이득을 곱하여 더한 형태로 모델링되며, ACELP 방식을 사용한다. 표 1에서는 음성부호화기들의 주요 특징들을 요약하였다.

표 1. SMV와 AMR 음성부호화기의 특징
Table 1. Characteristics of SMV and AMR.

	SMV	AMR
Input speech	8 kHz, 13 Bit PCM	8 kHz, 13 Bit PCM
Frame size	20 ms	20 ms
LPC order	10 th	10 th
Lookahead	10 ms	5 ms
Pitch delay	Update every frame (20 ms)	Update every subframe (5 ms)
Coding method	eX-CELP	ACELP

III. SMV에서 AMR 음성부호화기로의 상호부호화 알고리즘

3.1. 전체적인 구조

그림 1은 SMV에서 AMR로의 상호부호화 알고리즘에 대한 블록도이다. SMV의 부호화된 패킷은 먼저 파라미터 복호화 과정을 통해 상호부호화 과정에서 변환되어야 하는 파라미터들을 선택한다. 그 후, SMV의 여기 신호를 이용하여 AMR 코드북 검색 과정에서 사용될 목적 신호를 계산한다. SMV의 4개의 전송률에 따라 다른 상호부호화과정이 이루어진다. LSP 변환 과정은 전송률과 관계없이 동일하게 수행된다. 피치 지연 및 적응 코드북 변환은 SMV의 Rate 1과 1/2에서 이루어지며, Rate 1/4와 1/8은 피치에 대한 정보가 없기 때문에 피치 지연 및 적응 코드북 변환 과정이 실행되지 않는다.

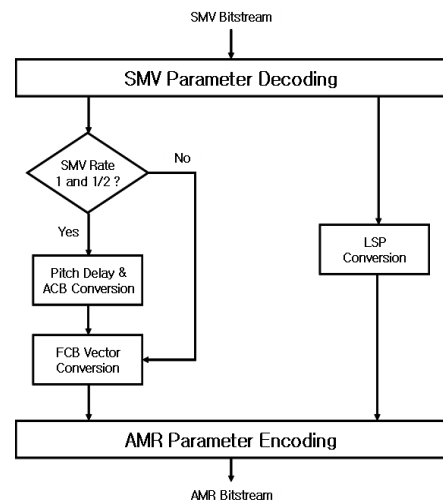


그림 1. 제안된 SMV에서 AMR로의 상호부호화 알고리즘의 블록도

Fig. 1. Block diagram of the proposed transcoding algorithm from SMV to AMR.

3.2. LSP 변환

그림 2는 LSP 변환 과정을 나타낸 것이다. 그림에서 진한 부분은 윈도우 가중치를 나타낸다. AMR과 SMV는 프레임 크기 및 부프레임 개수가 같다. 또한 AMR의 12.2 kbps 모드에서 2번째와 4번째 부프레임의 LSP를 전송하는 것만 다를 뿐, AMR의 12.2 kbps를 제외한 모드와 SMV에서 모든 4번째 부프레임에 해당하는 LSP만 양자화하여 전송한다. 따라서 AMR 음성부호화기에서 사용될 LSP는 SMV의 LSP를 그대로 사용하거나 간단한 선형 보간 (Linear Interpolation)을 통하여 변환될 수 있다. 식 (1)은 AMR의 12.2 kbps에만 적용되며, 식 (2)는 모든 전송률에서 적용된다.

$$LSP_{AMR}^{(n,2nd)} = 0.5(LSP_{SMV}^{(n-1,4th)} + LSP_{SMV}^{(n,4th)}) \quad (1)$$

$$LSP_{AMR}^{(n,4th)} = LSP_{SMV}^{(n,4th)} \quad (2)$$

3.3. 피치 지연 변환

피치 지연은 CELP방식에서 적응코드북 검색에 중요한 역할하기 때문에 작은 오차도 음질에 큰 영향을 미친다. 또한 SMV의 Rate 1/4와 Rate 1/8에서는 피치 파라미터를 사용하지 않는다. 따라서 SMV 패킷에서의 복호화된 피치 지연을 AMR 복호화 과정에서 그대로 사용할 경우 음질 저하는 당연한 결과이다.

그림 3은 SMV와 AMR의 피치 지연 값을 비교하기 위해 부프레임 단위로 나타낸 것이다. 그림 3의 (a)는 음성부호화기의 부프레임에 따른 피치 값이고, 그림 3의 (b)는 피치의 차이이다. SMV의 피치 지연을 계산하지 않는 영역 (Rate 1/4와 Rate 1/8)에서는 피치 지연을 0으로 고정하였다. 유성음에 가까운 53번째에서 506번째 부프레임 구간에서 피치 지연이 많은 부분이 일치함을 알 수 있었고, 그 외 구간 (무성음 구간)에서는 SMV의 부호화과정에서 피치 파라미터를 검출하지 않기 때문에 AMR의 피치 지연과 큰 차이를 보였다.

그림 4는 피치 지연 변환을 위해 제안된 알고리즘 블록도이다. 제안된 알고리즘은 피치 지연의 개회로 피치 지연 과거 값을 이용하여 예측하고, 예측된 피치 지연은 파라미터 복호화 과정을 통하여 복호화된 SMV의 피치 지연과 비교한다. 만약 그림 4에서 P_{SMV} 와 P_{AMR} 의 차가 문턱 값보다 작으면 SMV의 폐회로 피치 지연이 AMR의 개회로 피치 지연으로 선택되어 개회로 피치 지연 검색 과정을 생략한다. 하지만, 피치 지연의 차이가 문턱 값보다 크면 좀 더 정확한 피치 지연을 얻기 위해 AMR의 개회로 피치

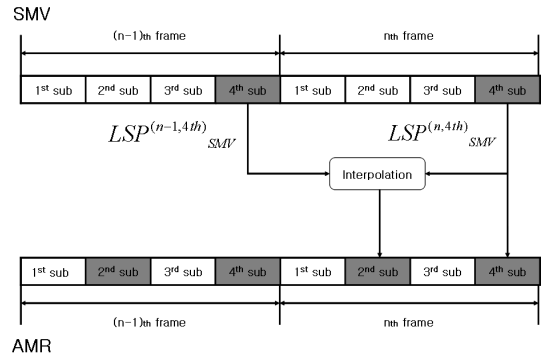


그림 2. SMV에서 AMR로의 LSP 변환
Fig. 2. LSP conversion from SMV to AMR via linear interpolation.

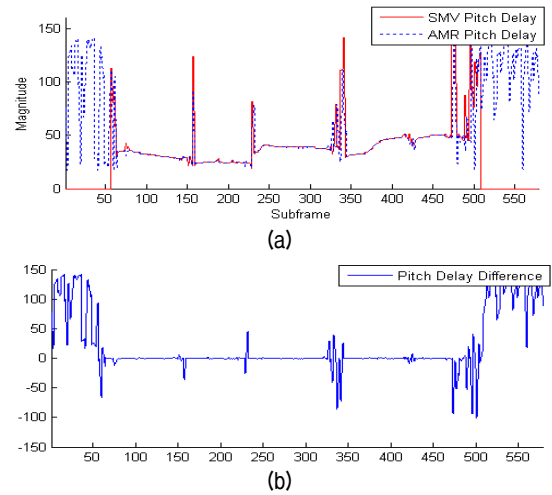


그림 3. 피치 지연 비교. (a) SMV의 피치 지연과 AMR의 피치 지연, (b) 피치 지연 차
Fig. 3. Comparison of the pitch delay, (a) The pitch delay of SMV and AMR, (b) Difference of the pitch delay.

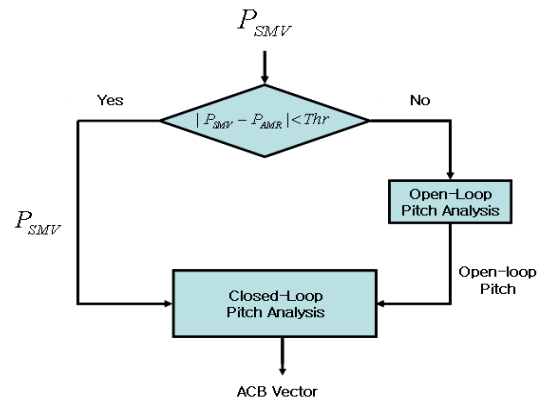


그림 4. SMV에서 AMR로의 피치 지연 변환 블록도
Fig. 4. Block diagram of the pitch delay conversion from SMV to AMR.

지연 검색이 수행된다. 유성음 부분에서는 피치 지연이 거의 일치하기 때문에 개회로 피치 지연 검색의 계산량만

큼 적어지고, 두 음성부호화기에서 얻어진 큰 피치 지연 차이로 발생하는 음질 저하 문제를 해결할 수 있다 [5][7].

3.4. 고정 코드북 벡터 변환

고정 코드북 벡터 검색은 두 음성부호화기의 부호화 과정에서 가장 큰 계산량이 요구되는 부분이고, 상호부호화 알고리즘에서 매우 중요하다. 따라서 계산량을 줄이기 위해서 고속 고정 코드북 검색 방법이 필요하다. 제안된 고속 고정 코드북 검색방법은 레퍼런스 벡터 (reference vector)를 이용하는 반복 펄스 위치 재검색 알고리즘이다. 제안된 방법은 레퍼런스 벡터로부터 추출해낸 펄스 배열을 사용하여 2 펄스 재검색을 먼저 수행한 뒤 단일 펄스 재검색을 수행한다.

우선, 제안된 고정 코드북 검색을 위해 관련 파라미터 및 수식을 알아본다. 고정 코드북 검색을 위한 목적을 신호를 먼저 구해야 된다. 고정 코드북 목적 신호는 식 (3)과 같다.

$$x_2(n) = x(n) - g_p y(n) \quad n = 0, \dots, 40 \quad (3)$$

$x(n)$ 는 적응 코드북 목적 신호이며, $y(n)$ 는 필터링된 적응 코드북 벡터이다. 그리고 g_p 는 양자화 되지 않은 적응 코드북 이득이다.

$$d(n) = \sum_{i=n}^{39} x_2(n) h(i-n), \quad n = 0, \dots, 39 \quad (4)$$

$$\Phi(i, j) = \sum_{n=j}^{39} h(n-i) h(n-j), \quad i = 0, \dots, 39 \quad (5)$$

$$j = i, \dots, 39$$

여기서 $d(n)$ 는 목적 신호인 $x_2(n)$ 와 임펄스 응답인 $h(n)$ 간의 상관도이며, $\Phi(i, j)$ 는 $h(n)$ 간의 자기상관행렬이다.

고정 코드북은 다음 검색 기준 식 (6)을 최대화하는 펄스 위치들을 선택한다.

$$Q_k = \frac{x_k^r H c_k}{c_k^t H^t H c_k} = \frac{(d^t c_k)^2}{c_k^t \Phi c_k} = \frac{(R_k)^2}{E_k} \quad (6)$$

검색의 목적은 최적화된 펄스 위치를 가지고 코드 벡터를 결정하는 것이다. 최적화된 펄스 위치를 구하기 위해서는 우선 레퍼런스 벡터 식 (7)을 이용하여 초기 검색을 위한 펄스들의 위치를 얻어낸다. 그리고 $res_{LTP}(n)$ 는 장구간 예측 잔여 신호이다.

$$b(n) = \frac{res_{LTP}(n)}{\sqrt{\sum_{i=0}^{39} (res_{LTP}(i))^2}} + \frac{d(n)}{\sqrt{\sum_{i=0}^{39} (d(i))^2}}, \quad n = 0, \dots, 39 \quad (7)$$

각 트랙에 위치할 수 있는 펄스들은 $b(n)$ 에서 큰 값부터 선택하여 위치 검색을 위한 초기 펄스 배열을 만들어낸다. 2펄스 재검색과 단일 펄스 재검색을 위해서는 초기화된 펄스들의 검색 기준 Q_k 가 계산되어 있어야 된다.

검색을 단순화하기 위해서 신호 $d(n)$ 과 행렬 $\Phi(i, j)$ 는 미리 선택된 sign을 포함하기 위해 수정된다. $s_b(n)$ 은 $b(n)$ 의 sign이며 수정된 신호 $d'(n)$ 과 $\Phi'(i, j)$ 는 다음과 같다.

$$d'(n) = s_b(n) d(n), \quad n = 0, \dots, N-1 \quad (8)$$

$$\Phi'(i, j) = s_b(i) s_b(j) \Phi(i, j), \quad i = 0, \dots, N-1 \quad (9)$$

$$j = 0, \dots, N-1$$

식 (8)과 (9)에 의해 Q_k 의 분자의 R과 E는 다음과 같다.

$$R = \sum_{i=0}^{N_p-1} d'(i) \quad (10)$$

$$E = \sum_{i=0}^{N_p-1} \Phi'(m_i, m_i) + 2 \sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} \Phi'(m_i, m_j) \quad (11)$$

먼저 2펄스 재검색이 수행된다. 인접한 두 트랙에서 하나의 펄스씩 두 개의 펄스를 제거한 후 펄스가 제거된 트랙에서 새로운 펄스 조합을 검색한다. 이는 검색 기준 Q_k 를 최대화하는 것을 선택함으로써 인접한 두 개의 트랙에서 기존에 검색된 펄스를 검색하게 된다. 두 펄스가 제거된 뒤 펄스가 제거된 후의 검색 기준 Q_k'' 의 분자와 분모는 다음 식 (12)과 (13)에 의해 다시 계산된다.

$$R' = R - d'(m_k) - d'(m_{k+1}) \quad (12)$$

$$E' = E - \Phi'(m_k, m_k) - \Phi'(m_{k+1}, m_{k+1}) - 2\Phi'(m_k, m_{k+1}) - 2R_{hv}(m_k) - 2R_{hv}(m_{k+1}) \quad (13)$$

다음 단계에서 분자와 분모는 새로 추가된 펄스의 변화량만큼 더해져 수정된다. 제거된 펄스가 속해 있는 트랙에서 새로운 Q_k 을 최대화하는 펄스를 선택한다. R과 E는 다음 식 (14), (15)와 같이 수정된다.

$$R = R' + d'(m_k) + d'(m_{k+1}) \quad (14)$$

$$E = E' + \Phi'(m_k, m_k) + \Phi'(m_{k+1}, m_{k+1}) + 2\Phi'(m_k, m_{k+1}) + 2R_{hv}(m_k) + 2R_{hv}(m_{k+1}) \quad (15)$$

$$v_h(n) = \sum_{i=0}^{k-1} h(n-m_i) \quad (16)$$

$$R_{hv}(m) = \sum_{n=m}^{N-1} h(n)v_k(n-m) \quad (17)$$

새로운 펄스를 검색하는데 있어서 $R_{hv}(m)$ 값은 해당 트랙에서 선택 가능한 모든 경우에 대하여 실행 중에 계산된다.

위의 2펄스 재검색을 통해 얻어진 펄스 배열을 가지고 단일 펄스 재검색을 수행한다. 각각의 반복에서 트랙내 각 펄스의 가장 적절한 위치는 트랙 내에서 이전에 검색되어 있는 다른 펄스들의 위치를 유지하면서 검색한다. 새로운 펄스를 검색하기 위해 먼저 한 개의 펄스 위치를 제거한 뒤 제거된 후의 Q_k 값을 계산한다. 한 개의 펄스가 제거된 후의 검색 기준 Q'_k 의 분자와 분모는 다음 식 (18), (19)에 의해 계산된다.

$$R' = R - d'(m) \quad (18)$$

$$E' = E - \Phi'(m, m) - 2R_{hv}(m) \quad (19)$$

다음 단계에서 분자와 분모는 새로 추가된 펄스의 변화량만큼 더해져 수정된다. 제거된 펄스가 속해 있는 트랙에서 새로운 Q_k 을 최대화하는 펄스를 선택한다. R과 E는 다음 식 (20), (21)에 의해 수정된다.

$$R = R' + d'(m) \quad (20)$$

$$E = E' + \Phi'(m, m) + 2R_{hv}(m) \quad (21)$$

표 2는 AMR의 12.2 kbps 모드에 적용한 결과이다. 계산량은 1024에서 224로 약 80%정도 줄어든 것을 알 수 있다. 남/녀 화자가 발음한 입력 음성 모두 원본의 음성과 거의 같은 음성 품질을 유지한 것을 알 수 있다. 다른 모드들에 대해서도 같은 방식으로 검색 공간을 제한하여 계산량을 감소시켰다.

표 2. 고속 고정 코드북 벡터 검색 적용 결과 (12.2 kbps)
Table 2. Performance of the fast fixed codebook vector search (12.2 kbps mode).

	Original		Proposed	
	Male	Female	Male	Female
SNR (dB)	20.13	18.25	20.15	18.38
PESQ	4.073	3.742	4.051	3.717
Complexity	8x8x4x4=1024		2x4x8+10x2x8=224	

IV. AMR에서 SMV 음성부호화기로의 상호부호화 알고리즘

4.1. 전체적인 구조

AMR에서 SMV 음성부호화기로의 상호부호화기의 전체적 구조는 그림 5와 같다. AMR에서 복호화된 LSP를 가지고 직접적인 변환을 통해서 얻을 수 있으며, SMV의 모든 전송률에서 변환된다. 그러나 AMR과 달리 SMV는 전송률과 타입 (type)을 결정하는 추가적인 과정이 필요하다. SMV는 각 프레임마다 전송률과 타입을 결정하고 이에 따른 여기 신호 및 양자화 과정이 필요하다. 전송률과 타입을 구하기 위해서 AMR의 여기 신호에 관한 정보와 AMR 복호화 과정으로 복원된 음성신호를 이용한다. Rate 1과 1/2는 피치 변환을 통해 얻어진 개회로 (open loop) 피치 지연을 바탕으로 SMV의 적응 코드북 검색 방식을 통하여 폐회로 피치 지연과 적응 코드북 이득을 구한 후 고정 코드북 검색을 수행한다. Rate 1/4과 1/8일 때는 난수 발생기를 사용하여 여기 신호를 생성시킨다.

4.2. LSP 변환

AMR에서 SMV로의 LSP 변환은 프레임 및 부프레임 크기가 동일하고, 부프레임 수도 같기 때문에, 식 (22)와 같이 AMR에서 복호화된 4번째 서브프레임을 그대로 적용한다.

$$LSP_{SMV}^{(n, 4th)} = LSP_{AMR}^{(n, 4th)} \quad (22)$$

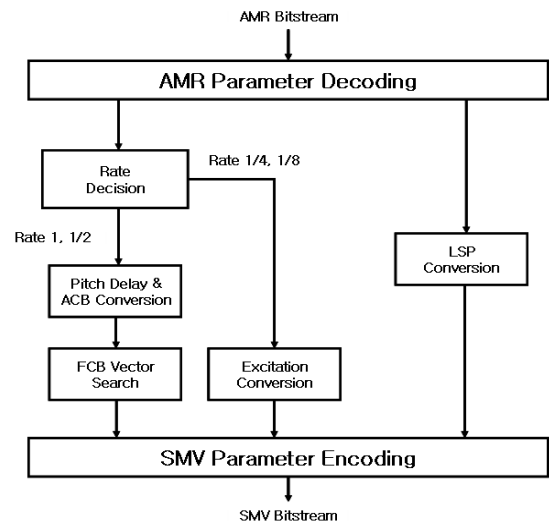


그림 5. 제안된 AMR에서 SMV로의 상호부호화 알고리즘의 블록도

Fig. 5. Block diagram of the proposed transcoding algorithm from AMR to SMV.

4.3. 전송률 결정

프레임 분류 과정은 그림 6과 같다. SMV의 전송률을 결정하기 위해 AMR 패킷에서 CELP 파라미터 (적응코드북 이득, 고정코드북 이득, NSR)와 AMR 패킷으로 복호화된 복원음성신호를 이용한다. SMV에서는 묵음(Silence), 잡음(Noise-like), 무성음(Unvoiced), 변화(Onset), 비정상적인 유성음(Unstationary Voiced), 정상적인 유성음(Stationary Voiced)의 6종류로 분류하지만 상호부호화기에서는 잡음을 제외한 5가지 프레임 클래스로 분류된다. ARM 패킷에서 필요한 정보 (피치 지연, 잡음 대 신호비 (NSR), 적응 코드북 이득, 고정 코드북 이득)를 이용하여 음성 (Speech)과 묵음 (Silence)을 1차 분류 (1st stage) 한다. 적절한 선형과정을 거친 적응 코드북 이득 (ACB gain)과 고정 코드북 이득을 참고하여 잡음 대 신호비 (NSR)가 일정 값 이상인 경우 묵음으로 분류한다. 그리고 피치 지연이 편차가 큰 경우도 묵음으로 분류한다. 이것은 보통 무성음 (Unvoiced)이나 묵음의 피치 지연이 큰 경향을 가지기 때문이다. 2차 분류 (2nd stage)에서는 음성 구간에서는 적응 코드북 이득의 크기를 이용하여 무성음과 유성음으로 분류한다. 유성음 구간에는 적응 코드북 이득 크기가 무성음 구간보다 크지만 대체적으로 값의 변화가 크다. 3차 분류 (3rd stage)에서는 유성음 구간에서 지난 프레임의 프레임 클래스가 무성음인 경우는 변화 (Onset)로 분류하고, 그렇지 않을 경우 유성음 (Voiced)로 분류한다. 4차 분류 (4th stage)에서는 유성음은 적응 코드북 이득과 피치 지연의 변화에 의해 정상상태의 유성음과 비정상상태의 유성음으로 분류된다. 프레임 안에서 분류된다. 최종적으로 평균 전송률 대 음성 품질이 가장 좋은 것으로 클래스 분류를 결정하였다 [7].

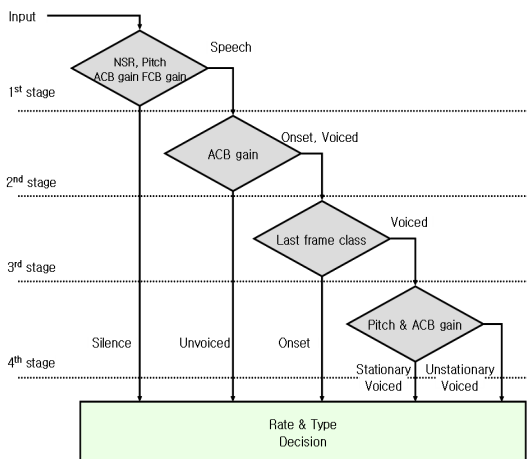


그림 6. 전송률 결정 알고리즘 블록도
Fig. 6. Block diagram of rate decision.

4.4. 피치 지연 변환

SMV의 개회로 피치값은 AMR 패킷에서 복호화된 피치 지연으로 대체하고, 그림 4와 비슷한 과정을 거친다. 즉 직전의 과거의 피치 지연 값과 비교를 통해 문턱값보다 작은 경우 개회로 분석을 생략하고 문턱값보다 큰 경우 좀 더 정확한 개회로 피치 지연 값을 얻기 위해 개회로 분석 과정이 수행된다. 결정된 개회로 피치 지연은 SMV의 폐회로 탐색에 이용된다.

V. 성능 평가

본 장에서는 제안된 방식의 성능을 확인하기 위해 계산량, 지연 시간, 음질 측면에서 tandem 방식과 비교된다.

제안된 알고리즘은 부동 소수점 형태의 C 언어로 구현되었다. 음성 샘플은 NTT에서 제공되는 8 kHz로 표본화된 8초 길이의 남성 음성과 여성 음성 각각 5개가 사용되었다. 그리고 전송 상의 오류는 없다고 가정하였다. AMR의 전송률 모드 중에 대표적인 4가지를 이용하여 실험값을 정리하였다.

5.1. 계산량

제안된 알고리즘의 계산량을 측정하기 위해 WMOPS (Weighted Million Operations Per Second)를 이용한다. 표 3에서 제안한 방식과 tandem 방식의 계산량이 비교 정리된다. 표 3의 (a)는 SMV에서 AMR로의 상호변환에서의 계산량을 정리한 것이며, 표 3의 (b)는 AMR에서 SMV로의 상호변환에서의 계산량을 정리한 것이다. 표에서 Average는 가로 축의 평균을 보인 것이다. 표 3의 (a)는 tandem 방식에 비해 전체적으로 50% 근접하게 감소됨을 알 수 있다. 이러한 이유는 선형 예측 분석의 생략, 폐회로 피치 검색 생략, 제안된 고속 고정 코드북 벡터 검색 등이 적용되었기 때문이다. 표 3의 (b)는 tandem 방식에 비해 약 20%의 계산량 감소를 확인할 수 있다. 그것은 고정 코드북 벡터 탐색 과정이 계산량에 있어서 많은 부분을 차지하는데, 이 부분에서 계산량을 줄이지 못하였기 때문에 감소율이 적은 계산량을 가지게 된 것이다.

5.2. 지연 시간

음성 통신 시스템에서 총 지연 시간은 알고리즘 지연, 처리 지연, 전송 지연을 합한 것을 말한다. 그러나 전송 지연은 네트워크의 구조와 상태에 따라 많이 달라지고, 처리 지연은 앞에서 본 계산량과 중복되는 결과이기 때문

표 3. 계산량 비교 (a)SMV→AMR (b)AMR→SMV
Table 3. Comparison of computational complexity. (a)SMV→AMR (b)AMR→SMV

(a)

		AMR		5.15 kbps	7.95 kbps	10.2 kbps	12.2 kbps	Average
		SMV						
S M V ↓ A M R	Mode 0	Proposed /Tandem	5.39 /9.81	6.63 /12.77	6.45 /12.40	6.53 /12.68	6.25 /11.92	
		Reduction Rate (%)	45.1	48.1	47.9	48.5	47.5	
	Mode 1	Proposed /Tandem	5.16 /9.82	6.67/12.81	6.54/12.45	6.93 /12.75	6.33/11.96	
		Reduction Rate (%)	47.4	47.8	47.4	45.6	47.1	
	Mode 2	Proposed /Tandem	5.18/9.87	7.04 /12.85	6.56 /12.47	6.55/12.82	6.33 /12.00	
		Reduction Rate (%)	47.5	45.2	47.4	48.8	47.2	
	Mode 3	Proposed /Tandem	5.21 /9.90	7.17 /12.86	6.59 /12.52	6.80 /13.00	6.45 /12.07	
		Reduction Rate (%)	47.0	44.2	47.3	47.6	46.5	

(b)

		SMV		Mode 0	Mode 1	Mode 2	Mode 3	Average
		AMR						
A M R ↓ S M V	5.15 kbps	Proposed /Tandem	22.82 /28.62	22.81 /28.66	22.88 /28.78	22.96 /28.70	22.87 /28.69	
		Reduction Rate (%)	20.2	20.4	20.4	20.0	20.2	
	7.95 kbps	Proposed /Tandem	22.01 /27.84	21.98 /27.86	21.94 /27.78	22.54 /27.90	22.12 /27.85	
		Reduction Rate (%)	20.9	21.1	21.0	19.2	20.5	
	10.2 kbps	Proposed /Tandem	21.37 /26.75	20.80 /26.66	21.23 /26.75	21.29 /26.66	21.17 /26.70	
		Reduction Rate (%)	21.6	21.7	22.1	22.3	21.9	
	12.2 kbps	Proposed /Tandem	20.26 /25.84	20.24 /25.84	20.23 /25.98	20.22 /26.01	20.24 /25.92	
		Reduction Rate (%)	21.6	21.5	22.1	22.2	21.8	

에 본 논문에서는 알고리즘 지연 시간만 다룬다. SMV에서 AMR로의 상호부호화에서 tandem 방식일 경우 지연 시간은 총 35 ms이다. 이것은 SMV의 한 프레임 길이가 20 ms, SMV의 예견 구간이 10 ms, 복호화된 신호부호화하기 위해서 사용된 AMR의 예견 구간 5 ms를 모두 더한 결과이다. 제안된 상호부호화 알고리즘일 경우, AMR의 예견 구간이 필요하지 않다. 선형 예측 분석 과정없이 LSP를 직접 변환하기 때문이다. 따라서 지연 시간은 30 ms로 짧아진다. AMR에서 SMV로의 상호부호화에서 tandem 방식일 경우 이처럼 총 35 ms의 지연 시간을 가진다. 하지만 제

안된 상호부호화 알고리즘일 경우, SMV의 예견 구간 10 ms가 필요하지 않다. 그래서 총 25 ms 지연 시간으로 짧아진다.

5.3. 객관적인 음질 평가

객관적인 음질 평가를 하기 위해 ITU-T의 표준인 PESQ (Perceptual Evaluation of Speech Quality) [8]를 사용하였다. 제안된 상호부호화 알고리즘과 tandem 방식을 비교한 결과를 표 4에 정리하였다. 표 4의 (a)는 SMV에서 AMR로의 상호부호화 음질 평가이고 표 4에서 (b)는 AMR에서 SMV로의 상호부호화 음질 평가이다. 표 4의 결과를 보면 대체적으로 tandem 방식과 비슷하거나 향상된 음질을 보였다. 하지만 표 4에서 (b)의 결과가 (a)의 결과보다 tandem 방식과 비교하여 좀 더 향상된 음질을 보였다. 왜냐하면 SMV에서 AMR로의 상호부호화에서는 고속 제안된 고정 코드북 벡터 탐색을 적용한 반면, AMR에서 SMV로의 상호부호화에선 제안된 고속 고정 코드북 벡터 탐색을 적용하지 않았기 때문이다. 제안된 고속 고정 코드북 벡터 검색은 음질에 영향을 미치지만 음질의 감소폭은 거의 미미한 것을 알 수 있다.

표 4. 객관적인 음질 비교 (a)SMV→AMR (b)AMR→SMV
Table 4. Comparison of PESQ score (a)SMV→AMR (b)AMR→SMV.

(a)

		AMR		5.15 kbps	7.95 kbps	10.2 kbps	12.2 kbps	Average
		SMV						
S M V ↓ A M R	Mode 0	Proposed /Tandem	3.10 /3.12	3.31 /3.30	3.38 /3.36	3.46 /3.43	3.31/3.30	
		Reduction Rate (%)	3.12 /3.11	3.30 /3.29	3.35 /3.34	3.44 /3.42	3.30 /3.29	
	Mode 2	Proposed /Tandem	3.08/3.09	3.27 /3.28	3.34 /3.32	3.41 /3.40	3.28 /3.27	
		Reduction Rate (%)	3.08/3.07	3.28 /3.27	3.32 /3.30	3.37 /3.36	3.26 /3.25	

(b)

		SMV		Mode 0	Mode 1	Mode 2	Mode 3	Average
		AMR						
A M R ↓ S M V	5.15 kbps	Proposed /Tandem	3.11/3.09	3.34 /3.31	3.41 /3.38	3.49 /3.47	3.34 /3.31	
		Reduction Rate (%)	3.11 /3.07	3.34 /3.30	3.43 /3.38	3.48 /3.44	3.32 /3.28	
	10.2 kbps	Proposed /Tandem	3.07 /3.04	3.31 /3.28	3.39 /3.36	3.45 /3.41	3.30 /3.27	
		Reduction Rate (%)	3.09 /3.04	3.28 /3.23	3.37 /3.34	3.47 /3.42	3.31 /3.26	

VI. 결론

본 논문에서는 기본 tandem 방식이 갖는 문제점을 해결하기 위해 서로 다른 디지털 셀룰러망에 사용되는 SMV와 AMR 음성부호화기를 위한 상호부호화 알고리즘을 제안하였다. LSP 변환, 피치 지연 변환, 펄스 재검색을 위한 고속 고정 코드북 벡터 검색, 전송률 결정 등으로 이루어진 알고리즘을 통해서 적은 계산량과 짧은 지연 시간으로도 동등한 음질을 보여주고 있으며, 이에 상응하는 음질을 제공하여 실시간 통신에 보다 효과적으로 이용될 수 있다.

감사의 글

“이 논문은 2008년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음 (This work was supported by the research grant of the Chungbuk National University in 2008).”

참고 문헌

1. TIA/EIA/IS-127, *Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital system*, (1997).
2. 3GPP2 C.S0030-0 V2.0, *Selectable Mode Vocoder Service Option for Wideband Spread Spectrum Communication System*, (2001).
3. 3GPP TS 26.071 V.5.0.0, *AMR Speech CODEC; General Description*, (Jun., 2002).
4. Sung Wan Yoon, Sung Kyo Jung, Young Cheol Park, and Dae Hee Youn, “An efficient transcoding algorithm for G.723.1 and G.729A speech coders”, In Proc. Eurospeech 2001, 4, 2499-2502, 2001.
5. Sunil Lee, Seongho Seo, Dalwon Jang, Chang D. Yoo, “A novel transcoding algorithm for AMR and EVRC speech coders via direct parameter transformation,” In Proc. ICASSP 2003.
6. Goo-Rak Kwon, Ji-Min Chung, Sang-Jae Nam, Sung-Jea Ko, “A Novel Transcoding Technique between EVRC and G.729A for Mobile Multimedia Devices,” *Consumer Electronics, IEEE Transactions on*, Vol.53, Issue 3, pp.885-890, Aug. 2007.
7. 서성호, 장달원, 이선일, 류창동, “SMV와 G.723.1 음성부호화기를 위한 파라미터 직접 변환 방식의 상호부호화 알고리즘,” 전자공학회논문지 제40권 SP편 제6호, 2003년 11월.
8. ITU-T Rec. P.862, *Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs*, (2000).

저자 약력

•이 덕 종 (Duck-Jong Lee)



2007년 2월: 충북대학교 정보통신공학과 (공학사)
2007년 3월~현재: 충북대학교 전파공학과 (공학석사)
※주관심분야: 음성/오디오 부호화, VoIP, 디지털신호처리

•정 규 혁 (Gyu-Hyeok Jeong)



2004년 2월: 충북대학교 전기전자공학 (공학사)
2006년 2월: 충북대학교 전파공학과 (공학석사)
2006년 3월~현재: 충북대학교 전파공학과 (공학박사)
※주관심분야: 음성/오디오 부호화, 디지털신호처리, VoIP

•이 인 성 (In-Sung Lee)



1983년 2월: 연세대학교 전자공학과 (공학사)
1985년 2월: 연세대학교 전자공학과 (공학석사)
1992년 2월: Texas A&M University 전기공학과 (공학박사)
1993년 2월~1995년 9월: 한국전자 통신연구원 이동통신 기술연구단 선임연구원
1995년 10월~현재: 충북대학교 전기전자공학부 정교수
※주관심분야: 음성/영상 신호 압축, 이동통신, 적응 필터