

음성인식 휠체어 시스템 설계 및 구현에 관한 연구

A study of speech recognition wheelchair system design & implementation

*강 성 인, *김 정 훈, *류 홍 석, *강 재 명 *이 상 배
Kang Sung In, Kim Jung Hoon, Ryu Hong Suk, Kang Jae Meung, Lee Sang Bae

*한국해양대학교 전자통신공학과

요 약

본 논문은 수족이 불편한 장애인의 편리성을 위해 휠체어에 음성인식 모듈을 개발하는데 목표로 하고 있다. 본 시스템의 주프로세서는 TMS320C32를 이용하였고, 전처리단계에서 잡음환경의 특성을 고려하여 Winer 필터를 적용해서 잡음을 제거하였고, 특징추출과정에서는 LPC Cepstrum을 이용하여 프레임당 12차의 특징패턴을 추출하였다. 그 후 인식부에서는 기존의 알고리즘 중 고립단어에서 흔히 사용하는 DTW(Dynamic Time Warping)과 오인식률 발생을 방지하기 위해 NN(Neural Network)를 결합한 Hybrid형태로 구현하였다. 본 연구에서는 DTW와 Hybrid형태를 각각 실험한 결과 고립단어 인식률이 평균 96%이상 나타났다.

ABSTRACT

This paper describes a speech recognition module in a wheelchair for the sake of convenience of the disabled persons. For this system, we used TMS320C32 as the main processor; and we eliminated noise by applying Winer filter while considering characteristics of noise environment in pre-processing stage, and then extracted 12 feature patterns per frame using LPC Cepstrum. Then, we implemented the hybrid form combining DTW (Dynamic Time Warping), which is generally used for isolated words in the conventional algorithms, and NN (Neural network) to minimize errors in recognition. In this research, we achieved a recognition rate of more than 96% on isolated words when DTW and Hybrid forms were individually experimented in noisy environment.

Key Words : 음성인식 알고리즘, Neural Network, DTW, DSP, 휠체어

1. 서 론

음성신호는 가장 보편적이고, 편리한 정보교환의 수단이다. 이 음성을 통해 기계 및 사용 장치에 인식시켜 동작시킴으로써 더욱 더 편리하게 응용 가능하게 할 수 있다. 이런 편리성으로 인해 최근 음성인식 분야는 가전제품, 자동차등 여러 분야에서 활발히 적용되고 있다.[1]

본 논문에서는 음성인식 기술을 이용하여 휠체어를 작동하여, 수족이 불편한 장애인에게 편리를 제공하고자 하는데 그 목적이 있다.[2]

음성 신호 처리는 대용량의 메모리를 요구하므로, 기존 연구는 PC에서 활발히 이루어졌지만, 여기서는 휠체어라는 점에 있어, PC를 직접 실어서 사용할 수 없기 때문에 소형이면서 수학적 처리속도가 빠른 TI사의 부동소수점 DSP인 TMS320C32를 사용해 설계 및 구현 중에 있다. DSP는 메모리 용량의 한계라는 단점은 있지만 소형화가 가능하기 때문에 임베디드 음성인식 시스템에 대부분을 차지한다.[3][4][5]

본 논문의 구성은 2장에서 전체적인 휠체어 시스템을 살펴보고, 3장에서는 음성인식 시스템의 하드웨어 구성과 소프트웨어 구조를 살펴 본 후 마지막으로 실험 및 고찰에 이어 결론을 맺는다.

2. 시스템 구성



그림 1. 휠체어 전체시스템 및 음성인식보드

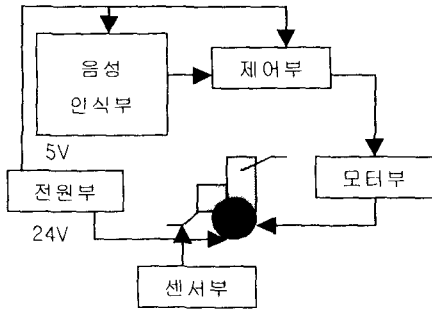


그림 2 전체시스템 블록도

그림1은 휠체어 시스템의 실제모습을 나타내었고, 그림 2에서는 전체 시스템 블록도로 이 휠체어는 제어부와 모터부, 전원부, 센서부, 음성인식부로 나누어져 있는 것을 확인 할 수 있다.

제어부는 80C196KC에 27C256과 62256의 메모리를 사용하여 조이스틱 제어에 사용하고 있으며, 전원부는 24V의 DC모터 두 개(230W@65rpm)를 80C196KC의 HSO 포트를 사용해서 H브릿지 회로방식을 통해 제어를 했다. 전원부는 80C196KC와 각종 주변 소자들을 위해 5V를 사용했고, 모터 제어를 위해 24V의 회로로 구성했다. 센서부에서는 초음파 센서를 통한 거리를 판별해서 세그먼트에 디스플레이 시킴으로써 안전거리를 판별하게 해준다 마지막으로 음성 인식부로서, TMS320C32 floating point DSP를 사용해서 음성입력 신호 즉 아날로그 신호를 디지털 신호로 처리, 제어부에 연결해서 모터를 제어 하는데 사용했다.[6]

3. 음성 신호 처리

3.1 하드웨어 구성

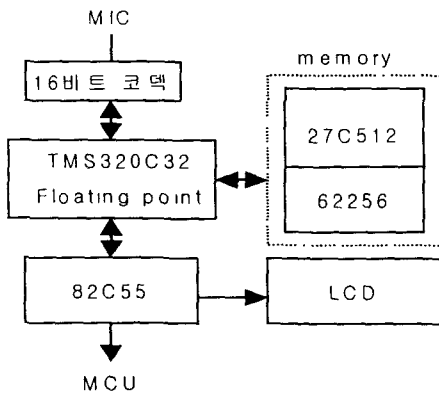


그림 3 음성인식 보드의 구성

실시간으로 받은 음성 신호를 16비트 코덱(A/D변환기)을 통해 디지털 신호로 바꾸어 준다. 이 때 한꺼번에 많은 데이터 들어오기 때문에 빠른 처리를 필요로 하는 프로세서가 필요하다. 여기에 TI사의 floating point DSP인 TMS320C32를 사용하여 빠른 데이터를 처리 후 82C55를 통해 병렬로 MCU에 인식된 값을 넘겨 준 후 이를 받은 제어부는 모터를 구동하게 한다.[3][4][5]

3.2 음성 분석(진처리과정)

3.2.1 Preemphasis, Frame Blocking, Windowing

식(1)의 Preemphasis는 음성신호의 저주파 성분을 약화시키고 고주파성분을 강조시켜 음성신호의 DC성분을 제거한다.

$$H(z) = 1 - a z^{-1}, 0.9 \leq a \leq 1.0 \quad (1)$$

다음에 Frame blocking을 통해 N개의 샘플 단위로 블록을 나누고, 윈도우를 통해 프레임별로 식(2)과 같이 계산을 실시한다. 여기에서 1프레임의 길이는 30msec 크기의 해밍(Hamming) 윈도우를 사용하고 있다.[7][8]

$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N-1 \quad (2)$$

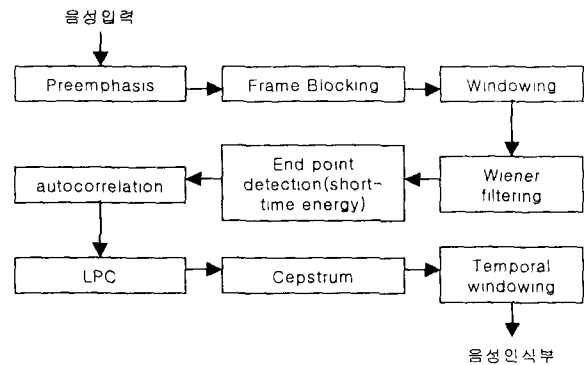


그림 4 음성 분석 과정

그림 4에서는 본 논문에서 적용한 음성분석단계를 블록도로 표현한 것이다.

3.2.2 Wiener Filtering 처리

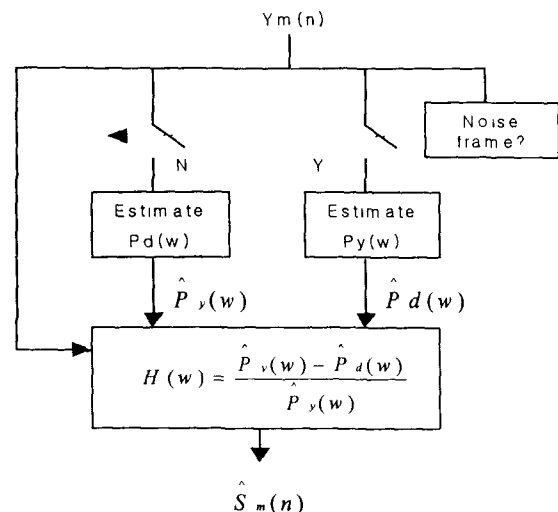


그림 5 Wiener filtering

해밍 윈도우 단위로 들어오는 음성신호는 잡음을 많이 포함하고 있다. 본 논문에서는 이러한 잡음을 제거하여

S/N비를 향상시키기 위해 필터링 처리를 하였으며, 여러 가지 필터링 중에서 응용분야에서 가장 많이 적용되고 있는 Wiener Filtering을 사용하였다. 필터링의 블록도는 그림 5와 같으며, 잡음이 포함된 $Y_m(n)$ 신호는 전달함수 $H(w)$ 를 통해 잡음이 제거된 원신호가 추출되게 된다.

3.2.3 음성 구간 검출

음성을 실시간으로 처리 시에는 음성의 구간을 검출해야 한다. 대부분 끝점구간을 많이 사용하는데, 끝점구간의 검출방법에는 절대에너지의 크기에 의해 음성을 검출하는 Short-time-energy와 음성 신호의 영교차율을 통해 검출하는 Zero Crossing rate방법이 있다.

두 방법을 적용해 본 결과 Short time energy가 시작점과 끝점 구간 검출에 유리하다는 것을 알았고, 본 논문에서는 Short time energy를 통해 끝점 구간을 검출했다. Short-time energy는 식(3)과 같다.[7][8]

$$E_i = \sum_{n=0}^{N-1} [W_i(n) S_i(n)]^2 \quad (3)$$

3.2.4 Autocorrelation & LPC (Linear Prediction Coefficient)

LPC는 과거의 음성 샘플을 가지고 현재의 음성을 샘플을 예측하는 방법이다. 이 방법은 LPC처리 전에 자기 상관 함수를 취해 주는데 그것은 LPC분석을 더욱더 안정적으로 해주기 위해서 사용한다.

자기 상관(Autocorrelation) 함수는 각 프레임에 취해준다. 식(4)은 자기 상관 함수이며, 여기서 P는 10으로 설정하였다.

$$r(m) = \sum_{n=0}^{(N-1)-m} x(n)x(n+m) \quad (4)$$

(m=0,1,2,3,...P)

자기 상관 계수를 추출한 다음 LPC계수를 추출한다.

LPC는 사람의 발성기관을 하나의 필터로 가정하고 그 필터의 계수를 음성의 특징 계수로 사용하는 방식이다. 흔히 사용하는 방법 중 Durbin method를 사용하였다.[7]

$$\begin{aligned} E^{(0)} &= r(0) \\ k_i &= \{r(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} r(i-j)\} / E^{(i-1)}, 1 \leq i \leq p \\ a_i^{(i)} &= k_i \\ a_j^{(i)} &= a_j^{(i-1)} - k_i a^{(i-1)}_{i-j}, \\ E^{(i)} &= (1 - k_i^2) E^{(i-1)} \end{aligned} \quad (5)$$

3.2.4 Cepstrum

LPC를 통과한 음성신호는 비선형적인 특성을 나타내는데, 이 신호들을 Cepstrum영역으로 변환하면 선형적 특성으로 변화된다. 식(6)은 durbin 알고리즘에 의해 생성된 LPC계수를 캡스트럼 계수로 변형 시키는 알고리즘이다. 본 논문에서는 12차 캡스트럼 계수를 사용했다.

$$\begin{aligned} \hat{C}_1 &= -a_1 \\ \hat{C}_n &= -a_n + \sum_{m=1}^{n-1} \frac{m}{n} a_m \hat{C}_{n-m} \quad (1 < n \leq p) \\ C_n &= \sum_{m=1}^{n-1} \frac{m}{n} a_m \hat{C}_{n-m} \quad (p < n) \end{aligned} \quad (6)$$

캡스트럼 계수의 민감도를 완화시켜주기 위해서 파라미터를 weighting 해준다. 구하여진 Cepstrum계수는 식(7)에서 표현한 tempered window를 통해 완화시킨다.

$$W_m = [1 + \frac{Q}{2} \sin \frac{\pi m}{Q}], \quad (1 \leq m \leq Q) \quad (7)$$

이렇게 처리된 Cepstrum계수는 DTW과 NN 인식 알고리즘에 입력되어진다.[7][8]

3.3 인식알고리즘

알고리즘은 그림6과 같이 하이브리드 방식을 취하고 있다. 주 인식 알고리즘으로 DTW를 사용하여 인식을 수행하였으며, 오 인식을 방지하기 위해 부 인식 알고리즘으로 신경망을 사용하였다. 여기서 Routine table은 오 인식이 날 경우를 테이블화 시킨 것이다.

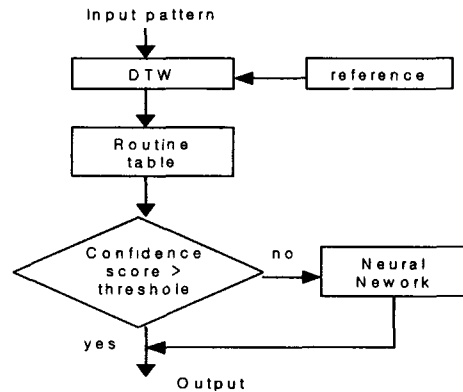


그림 6 인식알고리즘의 블록도

3.3.1 DTW

DTW는 입력 패턴과 참조 패턴사이의 거리를 계산해서 그 유사도를 측정하는 방식인데, 위의 그림을 보면 알 수 있듯이 시작점과 끝점을 결정하고 제한된 경로 내에서 단조증가를 해서 가장 가까운 거리를 판별해서 유사도를 측정한다. 이 DTW는 화자 종속에 많이 쓰인다.

$$\begin{aligned} ad(x_i, y_i) &= gd(x_i, y_i) + \min\{ad(x_{i-1}, y_i), \\ ad(x_{i-1}, y_{i-1}), ad(x_{i-1}, y_{i-2})\} \end{aligned} \quad (8)$$

위의 식을 보면 최종 누적거리는 현재의 거리와 이전 차원까지의 거리를 합으로 계산된다. 이 식에서 나온 최소거리를 통해 두 단어의 Distance값이 출력된다.

본 실험에서 인식 단어로 결정하기 위해서는 거리값(Distance Value)을 2.5로 설정하였고, 거리값이 2.5이하이면 인식단어로 결정한다.[1][7][8]

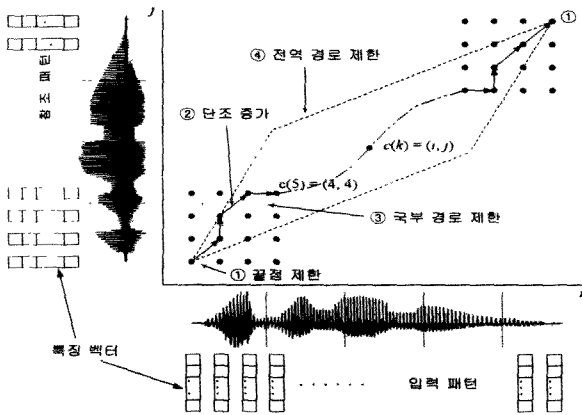


그림 7 DTW 계산법

3.3.2 Neural Network

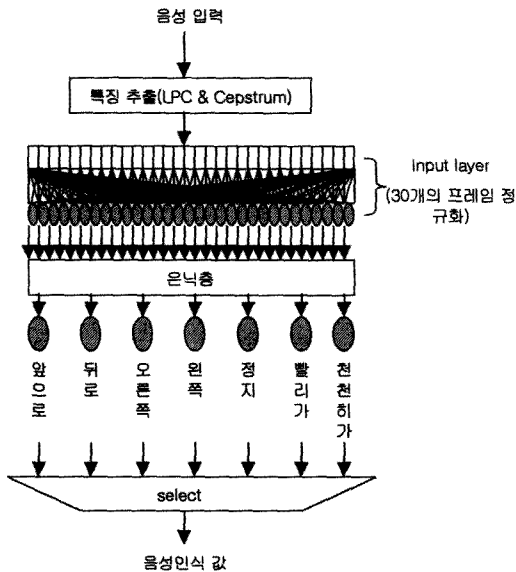


그림 8 신경망 계통도

학습된 입력패턴들과 전혀 다른 패턴이 입력 시 오인식이 발생 할 수 있는데, 이때 인식된 후보들 중에서 상위 두 후보의 Distance값의 차가 미소하다는 것을 실험에 통해서 알 수 있다. 따라서 본 연구에서는 인식된 두 단어의 Distace값의 차를 구하여 미리 결정된 threshold값과 비교해서 만약 작은 값이 나오면 신경망을 이용하여 재인식을 수행한다. threshold값은 0.3으로 실험을 통해서 결정했다. 신경망의 입력으로는 1프레임당 음성특징 추출 과정에서 얻어진 LPC Cepstrum 12차 계수를 각 뉴런에 입력된다. 자기 다른 음성신호의 길이를 30프레임으로 정규화시켜 입력층에 들어간다. 그림 8은 신경망 계통도를 나타냈으며, 총 7개의 단어를 학습시켰다.

본 연구에서는 비선형 문제를 해결하는데 우수한 특성을 가진 BP(Back Propagation) 알고리즘 사용하였다. 신경망의 구조는 그림 8에서 나타난 것처럼, 30개의 입력층과 45개의 은닉층, 7개의 출력층으로 구성하였고, 그림 9는 신경망의 학습과정을 도시하고 있다.[10]

본 논문에서는 미리 획득한 21개의 표본 음성에 대하여 위와 같은 과정으로 학습을 수행하였을 때 학습에 의해 설정된 가중치를 이용하여 임의의 입력음성에 대한

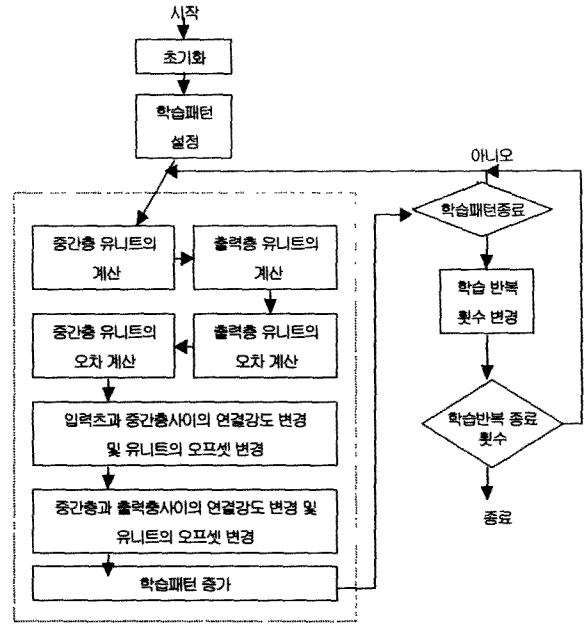


그림 9 신경망의 학습과정

음성인식이 이루어진다. 인식의 판별조건은 아래의 식(9)을 이용하여 판별한다.

$$(O_i / \sum_{i=1}^N O_i) * 100\% \geq 80\% \quad (9)$$

4. 실험 및 고찰

제안된 모델에 대하여 7개의 명령어에 대한 인식 실험을 행하였다. 인식종류는 화자중속으로 실험을 했으며, 1 사람의 목소리를 각 단어당 3번씩 발음하였으며, 이때 음성은 8Khz로 샘플링, 분해능 16bit로 저장했다.

	앞으로	뒤로	오른쪽	왼쪽	정지	빨라가	천천히가
앞으로	1.24	2.37	2.55	2.37	3.66	2.81	2.95
뒤로	1.86	2.72	2.5	2.62	4.08	3.03	3.28
오른쪽	2.01	2.64	2.44	2.52	3.94	2.98	2.91
왼쪽	2.37	1.45	3.01	2.12	4.18	2.89	10
정지	3.01	1.84	3.48	2.71	4.4	3.11	10
빨라가	2.62	1.95	3.17	2.63	3.78	3.16	10
천천히가	2.55	3.01	0.99	2.74	3.48	3.29	3.15
앞으로	2.65	3.11	2.21	2.73	3.2	3	2.86
뒤로	2.68	3.18	2.01	2.74	3.07	3.23	3.19
오른쪽	2.37	2.12	2.74	1.34	3.11	2.92	3.03
왼쪽	2.36	2.15	2.57	1.77	2.94	2.79	2.96
정지	2.74	2.25	2.61	1.7	3.63	3.05	2.71
빨라가	3.66	4.18	3.48	3.11	1.47	3.62	3.86
천천히가	3.7	4.24	3.49	3.09	1.68	3.71	3.93
앞으로	3.82	4.09	3.39	3.07	1.86	3.39	3.57
뒤로	2.81	2.89	3.29	2.92	3.36	1.08	2.91
오른쪽	2.94	3.45	3.32	3.1	3.64	1.79	2.77
왼쪽	2.99	3.3	3.19	3.1	3.78	1.83	2.73
정지	2.95	10	3.15	3.03	3.86	2.91	1.27
빨라가	2.93	10	3.19	3.12	3.53	2.67	1.78
천천히가	2.93	10	3.04	3.14	3.68	2.84	1.64

<표1>

표1에서는 7개 명령어가 들어갈 경우 21개의 레퍼런스와 비교한 distance값을 출력한 도표로 숫자가 가장 작은 부분이 인식된 부분이라고 할 수 있다. 그리고 출력값 중 10은 음성 입력이 레퍼런스 단어와 비교해서 전혀

맞지 않은 때 나타낸 값이다.

특성	내역
명령어	앞으로, 뒤로, 좌로, 우로, 정지, 빨리가, 천천히가
명령어당 단어 수	3개(상용자 점으로 변경 가능)
총 단어수	1명 X 7단어 X 3번 발성 = 21개
응답시간	1초이내 (단어의 발음이 끝나는 순간부터 호스트로 응답이 오는데 걸리는 시간)
유효 발음 길이	0.3 ~ 2초
잡음이 없는 환경에서의 인식률 (DTW)	98%
잡음이 15dB인 경우의 인식률 (DTW)	91%
잡음이 없는 환경에서의 인식률 (하이브리드)	98%
잡음이 15dB인 경우의 인식률 (하이브리드)	96%

<표2>

표2에서는 휠체어 명령어 7개로 정하였고, 이 결과 잡음이 없는 상황에서 DTW와 Hybrid형은 인식율이 거의 비슷하게 나왔지만, 잡음 환경이 약간 포함된 15dB에서는 인식률이 차이가 나타났다. 본 논문의 주목적인 휠체어는 잡음 환경을 고려해야 되기 때문에 Neural Network을 이용한 재학습은 오식률을 감소시키는데 필요한 절차로 간주된다.

5. 결론 및 향후과제

본 연구에서는 휠체어 시스템의 음성 인식부를 화자종속에서 가장 많이 사용되어지고 있는 DTW알고리즘에 후처리 방안이 고려된 Neural Network를 넣어 인식률 향상시킴에 초점을 맞추어서 설계 및 구현을 했다. 또한 실제 잡음환경을 고려하여 Wiener Filter가 사용되었으며 잡음의 상당부분을 제거시켜 주었다. 실험결과 실제환경에서 테스트한 결과 인식률이 향상된 것을 확인할 수 있다.

향후과제로는 DSP의 메모리 용량을 늘려서 보드를 설계한 뒤, 화자독립에서 많이 사용되는 HMM(Hidden Markov Model) 인식 알고리즘을 포팅 할 예정이고, 여기에서 발생하는 문제점을 신경망으로 해결할 수 있을 것이다. 아울러 좀 더 잡음에 강한 필터를 적용해야 할 것이다.

참고문헌

[1] 오영환, 음성언어정보처리, 홍릉과학출판사, 1998.
 [2] Satoru Nakanishi, Yoshinori Kuno, "Robotic Wheel chair Based on Observations of Both User and Environment", IEEE/RSJ International Conference on Intelligent Robots and System, 1999.
 [3] 이지홍, "DSP Chip의 활용", 서일 DSP기술연구서

공저, 2000.

[4] 김창근, 한학용, "TMS320C32를 이용한 실시간 음성 인식 무선자동차의 구현", 대한 시스템 학회, 2001
 [5] 정의주, 정훈 "TMS320C32 DSP를 이용한 실시간 화자 종속 음성인식하드웨어모듈(VR32)의 구현", 한국 음향 학회 Vol.17, No.4.14-22, 1998.
 [6] 박준혁, "음성인식이 가능한 이동 로봇에 관한 연구", 부산대학교 대학원 석사학위 논문, 1998.
 [7] Lawrence Rabiner, Biing-Hwang Juang, "Fundamentals of speech recognition", Prentice Hall International Inc. 1993
 [8] 손영선, 추명경, "DTW방식을 이용한 음성 명령에 의한 커서 조작", 퍼지 및 지능시스템학회 논문지 2001, Vol. 11, No 1, pp 3-8.
 [9] Steven L.Gay, "Acoustic signal processing for Telecommunication", Kluwer Academic Publishers, 2000.
 [10] A. Weibel, T. Hanazawa, G.Hinton, K.Shikano, and K.J.Lang, "Phoneme Recognition Using Time Delay Neural Network", IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-37: 328-339, 1989.