미비사항 : 이정배, 황영섭, 남지은 영문이름이 없습니다. 확인바랍니다.
그림 3, 4, 5, 6, 7 상태가 좋지 않습니다. 수정할 수 있는 그림 파일을 보내 주시기 바랍니다.

변형된 상태분할 알고리즘을 이용한 원격 HMI 시스템 제어

이종욱* · 이정배** · 황영섭** · 남지은**

요 약

일반적인 HMI system은 원격 감시제어를 네트워크를 통하여 하고 있으나 기능이 제한 적이다. 본 논문에서는 산업용 HMI 시스템을 변형된 상태분할 알고리즘을 적용 하였다. 이 방법은, 미리 예상되는 질의어에 대한 데이터들을 갖고 모델링을 하였다. 그 결과, 모델링하는데 많은 시간이 절약되었고, 시스템을 안정적이고 정밀하게 구성하여 98.15%의 높은 인식률을 나타냈다. 음성 HMI 시스템을 산업용에 적용하여 인간이 직접적으로 활동할 수 없는 작업 환경에서도 산업용 기기들을 안정적으로 구동시킬 수 있다. HMI 시스템 엔진의 성능을 최적화하였다.

The Remote HMI System Control Using the Transformed Successive State Splitting Algorithm

Jong Woock Lee* ·

ABSTRACT

Currently, The HMI system is being used on the network is limited in the ability. In this paper, an Industrial HMI applied the transformed state splitting algorithm. this study suggests by applying a transformed the Successive state splitting algorithm, for the modeling in the questions of the expected data. So, you can save time and reliable and precise as high as 98.15 percent repregented recognition rate.

HMI system applied to the voice of industrial equipment the man can not act directly in the industry environment was able to drive devices. Optimize the performance of the engine was the voice of HMI system.

Key words : Speech Recognition, HMI, Linear Predictive Coding, Contextual Domain

* 한국폴리텍 I 대학 제주캠퍼스 전자통신학과
** 선문대학교 컴퓨터공학부

1. 서 론

산업 분야의 기술 발달과 기업 전략의 변화는 산업 전반에 걸친 공정 자동화(Process Automation)의 활용 영역을 극대화 시키고 있다. 오늘날의 공정 자동화는 각 업무분야에서 적용되는 이기종 시스템 간의 연결과 통합은 물론 원활한 흐름을 포함한다. HMI 시스템은 사용자에게 각 업무 분야의 플랫폼을 연결하는 것 뿐만 아니라 각 플랫폼을 통합하여 최적의 생산성을 확보할 수 있어야 한다[14].

이러한 문제점들을 해결하기 위해서는 Robot, Conveyor, 자동창고 및 그 외 수많은 기계들을 움직일 수 있는 제어시스템과 이 시스템들을 통합하는 Network 및 Software가 반드시 필요하다. 또한 이러한 Network을 통하여 수집되는 공장 내의 수많은 정보들을 처리하는 MMI Software와 SCADA System 같은 소프트웨어에 의해 의존해 왔다. 이러한 시스템들은 원격으로 제어하는 기능은 있으나 아직까지 음성제어 부분 기능은 없다. 본 논문에서는 HMI system에 음성기능을 추가하여 유·무선 원격감시제어를 위한 Voice HMI(Human Machine Interface) 시스템'을 개발하고자 하였다. 특히, 기존 HMI 시스템은 산업현장 또는 전문분야에서의 주로 사용되었으나, Voice HMI 시스템은 음성과 문자 등 편리한 인터페이스 기능과 더불어 산업현장에서 검증된 신속 정확한 정보처리 능력을 바탕으로 기존분야의 확대활용은 물론 홈-네트워킹, 사이버 시큐리티, 모바일 원격감시제어 등 인간의 생활 전반에 걸쳐 다양한 모습으로 확대 적용될 수 있다.

(그림 1) Voice HMI 시스템의 구성도

2. HMI 시스템에서의 음성 인식

2.1 음성 신호의 개요

음성파에는 음운성, 개인성 등의 정보가 포함되어 있다. 음성 인식의 목표는 음운성에 관한 정보 즉 음성파가 담당한 언어적 내용을 추출하는 것이다. 음성 인식의 과정은 신호의 음성 구간을 구분하여 그 구간을 구분 분석하여 그 구간의 특징을 추출하여 인식하는 것이다.

2.2 음성의 전처리

음성 신호로부터 인식에 사용되는 특징 파라미터를 얻는 과정까지를 음성의 전처리라 한다. 예를 들어 어떤 단어의 신호가 입력되었을 때 음성 정보의 손실없이 음성 구간의 앞, 뒤를 명확하게 구분하여 인식에 사용될 수 있도록 특성 파라미터를 얻는 방법이다[11].

2.2.1 음성 구간의 끝점 검출

음성을 인식하기 위하여 가장 먼저 해야 할 일은 입력된 음성에서 음성 구간을 감지하는 일이다. 음성인식에서 음성구간을 감지하는 기술을 끝점검출(End Point Detection)이라 한다. 정확한 끝점 검출의 필요성은 두 가지로 요약할 수 있다. 첫 번째로 끝점의 정확성은 단어 인식의 정확도와 밀접한 관계를 갖고 있다. 따라서 끝점 검출이 제대로 되지 않으면 단어 인식 성능이 떨어지게 된다. 두 번째로 음성인식을 위해서는 음성의 특징을 추출해야 하는데 끝점 검출이 제대로 되지 않으면 특징 추출하는데 많은 시간이 걸린다. 정확히 끝점 검출을 할 경우 음성에 해당하는 구간만을 처리하므로 비음성 구간에 대한 계산 시간을 줄일 수 있다[3].

끝점 검출을 위한 알고리즘으로는 Rabiner와 Sambur의 에너지와 영교차율(ZCR : zero Cross Rate)

을 이용한 음성부 검출 알고리즘, Lamel의 Level Equalizer를 이용한 검출 알고리즘, 그리고, Teager 에너지를 이용하는 방법 등이 있다[4]. 일반적으로 영교차율과 에너지의 합을 이용하여 음성 구간을 검출하는 방법을 사용하고 있으나 신호대 잡음비가 충분히 큰 경우에는 에너지 함수만을 이용하여도 끝점 검출이 잘 이루어지므로 실시간 처리에 역점을 두고 처리 시간의 감소를 위해 에너지 함수만을 이용하였다[5].

일반적으로 단시간 에너지 함수는 식 (2-1)과 같이 정의된다.

$$E = \sum_{m=n-N+1}^{n} X^2(m) \qquad (2\text{-}1)$$

에너지함수가 가지는 한 가지 어려움은 큰 신호값에 매우 민감하다는 것이며, 이 문제를 해결하기 위하여 평균값에 절대치로 계산하였으며 이것의 표현은 식 (2-2)와 같다.

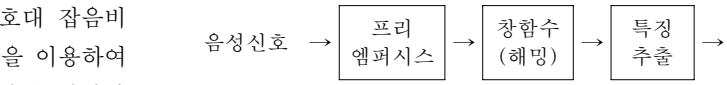
$$E = \sum_{m=n-N+1}^{n} |X(m)| \qquad (2\text{-}2)$$

즉, 단시간의 평균치는 n-N+1에서 n까지 N개의 Sample값에 절대치로 나타낼 수 있다.

2.2.2 특징 파라미터 추출

음성 인식 시스템은 LPF(Low-Pass Filter)를 통과한 데이터를 A/D(Analog/Digital) 변환하여 얻어지는 신호를 음성 부분을 검출하고 음성인식에 유효한 특징 파라미터를 추출하기 위해서 여러 신호처리 과정을 거친다. 이러한 음성 신호의 특성을 나타내는 특징 계수의 추출 방법은 에너지와 영교차율 등의 여러 가지 방법들이 있다[6].

단독어 인식시스템의 경우 특징 계수 추출 방법을 선택할 때 계산시간, 필요한 메모리 용량, 구현의 용이성 등을 고려해야 한다. 여기서는 여러 특징 파라미터들 중에서 가장 높은 인식율을 나타내는 선형계수법을 이용하여 얻어지는 LPC Cepstrum 계수를 사용하였다.



(그림 2) 신호분석 과정

이와 같이 수행함으로서 음성 신호의 Spectral slope를 더욱 정확히 표현할 수 있다. 다음 과정에서는 이 음성 신호를 16ms 단위로 나누어서 각 프레임(frame)에 창함수를 적용하여 평활화(soomthing)한다. 그 후 자기 상관함수(Autocorrelation)을 수행한 후 LPC 계수를 바탕으로 LPC 캡스트럼 계수를 추출하여 음성의 특징으로 한다[7].

- A/D 변환 : 10KHz Sampling, 10Bit A/D Converter
- Premphysis : $H(z) = 1-0.95 * z^{-1}$ (2-3)
- 프래임 : N-분석구간(16ms, 128samples)
  M-이동간격(16ms, 128samples)
- 해밍 창함수 :
  $W(n) = 0.54-0.46 * \cos(2/n-1)$ (2-4)
- 자기 상관 함수 :

$$R_{1(m)} = \sum_{n=0}^{N-m} x1(n) * x1(n+m) \quad 0 \le m \le p \qquad (2\text{-}5)$$

여기서, 1은 프레임 인덱스, p는 분석 차수이다.

- LPC 분석 :

자기 상관 방법인 Durbin 알고리즘을 사용하여 매 프래임에 대하여 p차의 계수 {a(m), 0 ≤ m ≤ p} 들을 구한다. 이때 분석 모델은 다음 식과 같다.[12]

$$A(z) = \sum_{i=0}^{p} a_1 \times z^{-1} \qquad (A_0 = 1) \qquad (2\text{-}6)$$

- LPC 캡스트럼 :

LPC 계수로부터 식 (2.7)에 의해 매 프레임에 대해 캡스트럼 계수{c(m), 1 ≤ m ≤ p}를 구한다.

$$c(m) = -a(m) - \sum_{k=1}^{m-1} \frac{k}{m} c(k)\, a(m-k) \qquad (2\text{-}7)$$

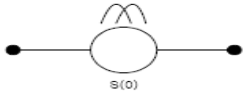
- 분석 차수 : 10차

2.3 연쇄상태분할(Successive State Splitting) 알고리즘

일반적인 음성인식 방법으로 잘 알려져 있는 HMM(Hidden Markov Model) 알고리즘은 Buam에 의해 처음 소개된 이후 1975년 CMU의 Barker와 IBM의 Jelinek 등에 의해 음성인식 분야에 도입되어 많이 쓰이고 있는 방법으로, 통계적 패턴인식에 기반을 둔 음향 모델로 시간영역의 순차적 사건들에 대해 과거와 현재의 사건이 주어졌을 때 현재 사건의 조건 확률 밀도는 가장 최근의 사건에 영향을 받는다는 마르코프 특성의 개념[13]을 가지고 시간에 따라 변화하는 신호의 분석으로 신호 모델링 방법이다. 여기에 SSS 알고리즘을 부가적으로 적용하여 변형된 알고리즘으로 인식 모델을 구성하고자 하며, 아래에 이러한 알고리즘에 대해 각 단계적으로 설명하고자 한다[8, 9].

단계 1 : 초기 모델의 학습

먼저 초기 모델로서, 1 상태 2 혼합수의 HMM을 준비하고 학습 데이터에 나타나는 모든 문맥 요소를 포함한 모든 학습 데이터를 사용하여 초기 모델을 학습한다. 초기 모델의 구조는 (그림 3.1)과 같다[9].



(그림 3) 초기 모델의 훈련

단계 2 : 분할할 상태의 결정

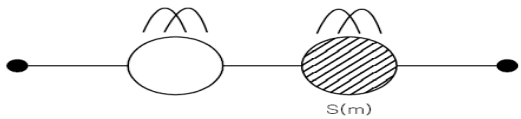
모든 상태 중에서 출력 분포가 가장 큰 상태를 선택하고, 분할해야 할 상태로 설정한다.

$$d_i = n_i \times \sum_{k}^{K} \frac{\sigma_{ik}^2}{\sigma_{Tk}^2} \qquad (2\text{-}8)$$

$$\sigma_{ik}^2 = \lambda_{i1}\sigma_{i1k}^2 + \lambda_{i2}\sigma_{i2k}^2 + \lambda_{i1}\lambda_{i2}(\mu_{i1k} - \mu_{i2k})^2 \qquad (2\text{-}9)$$

식 (2.8)를 사용하여 출력 분포를 계산하고 된다.

여기서, K는 파라미터의 차원수, λ_{i1}, λ_{i2}는 상태 i의 가중계수, μ_{i1k}, μ_{i2k}는 상태 i의 k번째 평균, σ^2_{i1k}, σ^2_{i2k}는 상태 i의 k번째 분산, n_i는 상태 i일 때의 학습샘플의 수, σ^2_{Tk}는 모든 학습샘플의 k번째 분산을 나타낸다.



(그림 4) 분포 크기의 계산

단계 3 : 상태의 분할

단계 2에서 선택된 상태를 2개로 분할한다. 이때 새로운 상태의 출력 확률 분포는 분할된 상태의 2혼합수를 각각 하나씩 할당받게 된다. 그 후 분할된 상태의 배치를 문맥방향(병렬)과 시간방향(직렬)으로 하고 각각에 대해 우도를 계산하게 된다, 문맥방향으로 분할한 경우의 우도를 P_c, 시간방향으로 분할한 경우의 우도를 P_t로 하여 최대 우도를 나타내는 방향으로 상태 분할을 결정한다.

2.3.1 문맥방향의 상태 분할

문맥방향의 상태 분할은 2가지 경로를 고려할 수 있는데 이때 각각의 학습샘플이 어느 쪽의 상태를 선택할 것인가를 결정할 필요가 있다. 각각의 학습 샘플에 대해 문맥환경 요소(선행음소와 후행음소)로 그룹을 나누고 이 그룹에서 우도가 높은 상태를 결정하게 된다. 이때 상태의 결정은 식 (2-10)를 사용한다.

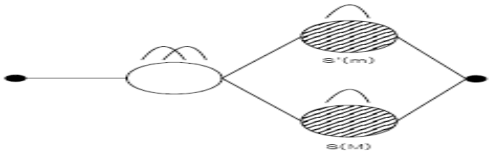
$$P_c = \max_{j} \sum_{l} \max\left(P_m(y_{jl}), P_M(y_{jl})\right) \qquad (2\text{-}10)$$

여기서, j는 현재 상태에서 문맥환경 요소를, y_{jl}은 요소 j의 값이 l번째 요소가 되는 학습 샘플의

부분집합을, $P_m(y_{jl})$는 y_{jl}을 상태 m에 할당할 때의 우도를, $P_M(y_{jl})$는 y_{jl}을 상태 M에 할당할 때의 우도를 각각 나타낸다. 문맥환경 요소 j를 결정한 후 식 (2.31)을 이용하여 분포 e_{jl}을 결정하게 된다.

$$\begin{cases} e_{jl} \in E_{m,j} & (P_m(y_{jl}) \geq P_M(y_{jl})) \\ e_{jl} \in E_{Mj} & (P_m(y_{jl}) < P_M(y_{jl})) \end{cases} \quad (2\text{-}11)$$

여기서, e_{jl}은 문맥환경 요소 j에 속하는 l번째 성분을, E_{mj}는 상태 m을 통과하는 성분들의 집합을, E_{Mj}는 상태 M을 통과하는 성분들의 집합을 각각 나타낸다. 문맥 방향의 상태 분할은 (그림 5)와 같다.



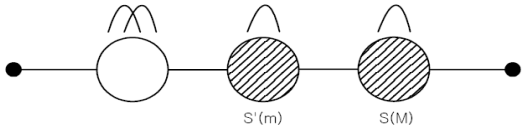
(그림 5) 문맥 방향 분할

2.3.2 시간방향의 상태 분할

시간방향의 상태 분할은 과 S'(m)을 직렬로 연결하여 학습한다. 이 경우 2가지의 상태 연결을 고려 할 수 있다. 하나는 상태 다음에 S(M)을 연결한 경우와, 다른 하나는 S(M) 상태 다음에 상태를 연결한 경우이다. 이 경우 식 (2-12)에서 최대 우도를 나타내는 연결방법을 선택한다.

$$Pt = \max(P_{mM}(Y), P_{Mm}(Y)) \quad (2\text{-}12)$$

여기서, $P_{mM}(Y)$은 S(M)을 차례대로 연결한 경우의 전체 우도를 나타내고, $P_{Mm}(Y)$은 S(M)과 S(m)을 차례대로 연결한 경우의 전체 우도를 나타낸다. 시간방향의 상태분할은 (그림 7)과 같다.

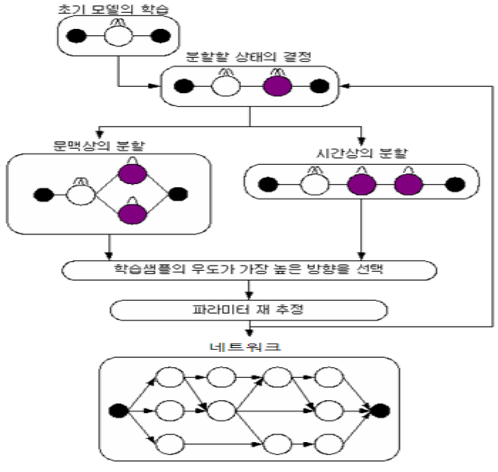


(그림 6) 시간 방향 분할

단계 4 : 재학습

S'(m)과 S(M)은 각각의 새로운 상태에 대한 2 혼합수에 대하여 가우시안 확률 분포를 가지고 있으며, 재구성할 수 있게 재학습한다. 분할동작에 영향을 받은 다른 상태들도 모델 파라미터들을 최적화 하도록 또한 재학습한다. 이러한 과정 후에, S'(m)은 S(m)와같이 새로운 이름을 부여하며, M은 1증가한다. 스텝 2에서 스텝 4를 M이 원하는 상태수에 다다를 때까지 반복한다.

지금까지의 단계를 통하여 네트워크 모델의 각 상태는 혼합수 2의 가우스 확률 분포를 가지게 된다. 최종적으로 네트워크 모델이 각 상태마다 혼합수 1의 가우스 출력 확률 분포를 가지도록 네트워크 전체 상태를 재학습 하게 된다. 이상의 단계를 수행한 후 최종적으로 네트워크 모델이 생성된다[2].



(그림 7) 연쇄 상태 분할 과정[10]

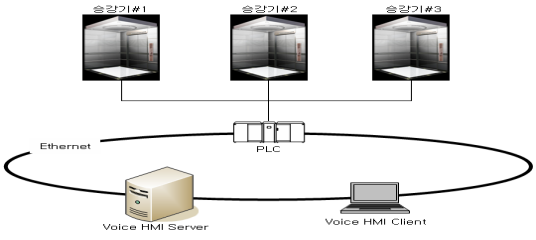
3. 인식 실험

3.1 실험 환경

3.1.1 HMI 시스템 구성

HMI 시스템은 음성엔진 서버에 모델링된 음성

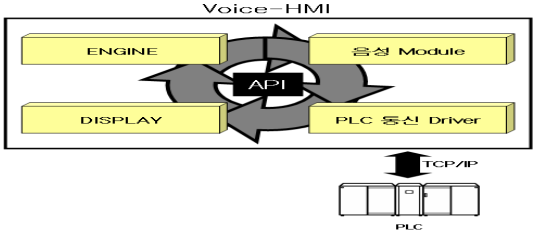
데이터를 저장하고 있으며, 모바일폰이나 일반전화를 통하여 발생된 음성 신호는 네트워크를 통하여 전달되며, 이러한 음성 신호를 서버에 장착된 음성엔진에서 인식이 이루어져 제어부로 전달, TCP/IP 모듈을 통하여 제어기로 데이터를 전달하여 산업용(엘리베이터 3대) 기기를 동작 시킬 수 있도록 구성하였다[1].



(그림 8) 음성 HMI 시스템 구성도

3.1.2 서버 내부 블록도

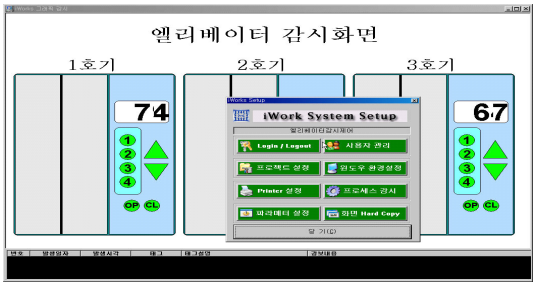
서버 내부에는 음성엔진과 음성을 외부로부터 입력받을 수 있는 입력부, 모니터링부, 제어기와 통신 제어가 가능한 통신 system 구성을 (그림 9)와 같이 구성하였다.



(그림 9) 음성 HMI 시스템 구성도

3.1.3 제어기 동작에 따른 모니터링 감시 화면

음성 서버에서 제어기에 명령을 주면 제어기에 연결된 기기들과 동시에 감시화면에 동작 상태가 나타나며, 서버 화면에 Payload 값들이 저장 및 표시되어 에러가 발생되면 바로 응급 조치를 취할 수 있게 구성하였다.



(그림 10) 실행 초기 화면

3.2. 음성 인식 구현

3.2.1 음성 모델링 및 인식 실험

- 본 시스템을 위한 인식 실험에는 대상어로 산업용 용어를 중심으로 20단어를 선정하여 남성 4명과 여성 4명으로 각각 3번씩 발음한 480개의 데이터 중에서 두 번씩 발음한 320개의 모델을 작성하여 사용하였고, 나머지 160개는 인식 실험에 사용하였다.

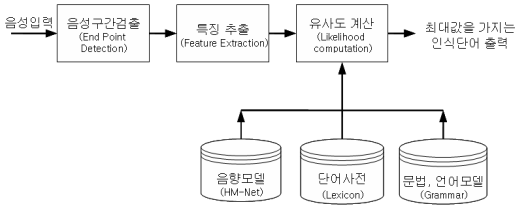
〈표 1〉 음성신호에 대한 코드북

no	word	sequence	no	word	sequence
1	일호기 일층	00000001	11	삼호기 일층	00001011
2	일호기 이층	00000010	12	삼호기 이층	00001110
3	일호기 삼층	00000011	13	삼호기 삼층	00001111
4	일호기 사층	00000100	14	삼호기 사층	00010000
5	일호기 오층	00000101	15	삼호기 오층	00010001
6	이호기 일층	00000110	16	엘리베이터	00010010
7	이호기 이층	00000110	17	맨 위층	00010011
8	이호기 삼층	00001000	18	맨 아래층	00010100
9	이호기 사층	00001001	19	문 닫으세요	00010101
10	이호기 오층	00001010	20	문 여세요	00010110

음성 데이터는 4.5KHz의 LPF(Low Pass Filter)를 통과한 음성을 Sampling하여 A/D Converter하여 얻은 데이터를 사용하였다.

- Sampling frequency : 10KHz
- A/D converter : 10Bit
- Frame length : 20ms

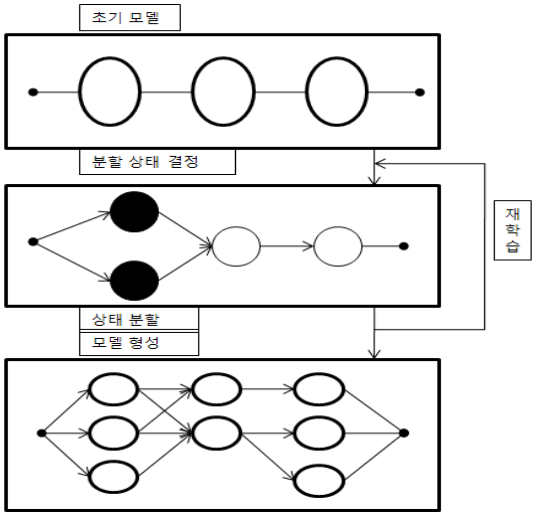
- Function : Hamming Windows
- Feature parameter : 10order,



(그림 11) 음성 모델링 과정

변형된 상태분할 알고리즘을 갖고 상태 분할을 할 때 일반적인 SSS에서는 문맥에 나타나는 특징만을 갖고 분할을 하였으나 본 논문에서는 문맥 클래스 분할법 즉 모델을 생성할 때 미리 예상되는 질의어에 대한 데이타들을 lexicon화하여 모델링하므로 시간적으로 많은 절약이 되었으며 보다 안정적이고 정밀한 모델링이 가능하게 되었다[10].

- step 1 : 초기모델 학습
- step 2 : 상태분할 결정
- step 3 : 상태 문맥 및 시간을 고려 분할
- step 4 : 안정될 때까지 반복(재학습)
- step 5 : 안정화된 모델 형성

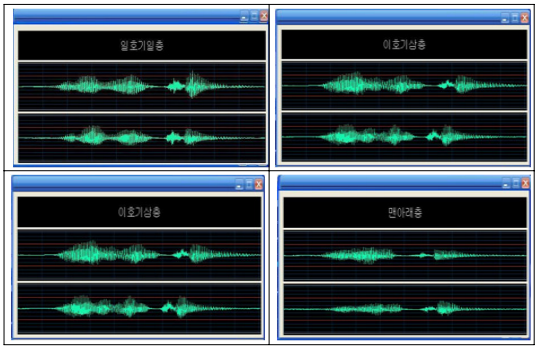


(그림 12) 음성 모델링 과정

3.3 음성 인식 실험 결과

3.3.1 음성 인식 화면

각 음성에 대한 모델링한 데이터로 코드북을 만들어 화자별로 남성이 2번씩 발음한 데이터와 여성 화자가 2번씩 발성한 데이터를 이용하여 인식 실험을 수행하여, 아래 표와 같은 인식률을 얻었다. 인식 결과를 Voice HMI system에 이식하여 적용한다면, 사람이 직접 작업할 수 없는 환경에서도 최대의 성능을 발휘할 것으로 기대된다.



(그림 13) 음성 인식 화면

〈표 2〉 코드북에 대한 인식률

남 성	인식률	여 성	인식률
화자 A	100%	화자 A	98.5%
화자 B	97.5%	화자 B	99.7%
화자 C	98.5%	화자 C	96.4%
화자 D	96.5%	화자 D	97.6%

4. 결 론

본 논문은 산업 현장에서 많이 사용되는 제어 시스템인 산업용 HMI 시스템이 TCP/IP 인터페이스에 의한 네트워크 제어 방식을 사용하고 있었으나 본 논문의 변형된 SSS 알고리즘을 이용하여 문맥 클래스 분할법 즉 모델을 생성할 때 미리 예

상되는 질의어에 대한 데이타들을 lexicon화하여 모델링된 데이터를 갖고 codebook을 생성하여 음성인식에 사용하였다.

본 논문에서는 변형된 상태분할 알고리즘을 이용하여 모델을 생성할 때 미리 예상되는 질의어에 대한 데이타들을 lexicon화하여 모델링하므로 시간적으로 많은 절약이 되었으며 보다 안정적이고 정밀한 모델링이 가능하게 되어 98.15%의 높은 인식율을 나타냈다. 그러나, 코드북에 데이터가 10000개 이상이 되면 인식율이 저하되고 인식 시간이 많이 걸리는 현상이 발생되어 코드북을 생성할 때 예상되는 질의어를 그룹으로 나타내 모듈식 코드북을 만들면 인식률이 향상될 것으로 생각 된다

산업용 HMI 시스템은 주로 미국 제품을 많이 사용하고 있었으나 Voice HMI 시스템의 엔진을 체계적으로 설계하면 국산화에도 기여할 것으로 생각되며, 특히 원격으로 사람들이 들어갈 수 없는 환경에서 Voice HMI 시스템은 성능을 발휘할 것으로 생각된다.

참 고 문 헌

[1] 송요면, 황진선, 신석현, “각종설비를 원격감시제어하기 위한 음성인터페이스 기반의 통합감시제어장치 및 중앙집중식 분산설비관리 시스템에 관한 중소기업기술혁신개발사업”, 서울중소기업청 연구보고서, 2005.

[2] 서준배, “대어록 연속음성인식을 위한 HM-Net 음성모델의 분석”, 석사학위논문, 2004.

[3] 변용규, “DMS 모델을 이용한 단독어 인식에 관한 연구”, 박사학위논문, 1991.

[4] L. R. Rabiner and B. H. Jung, “Fundementals of speech recognition”, Prentice hall, New Jersey, 초미. 6, 1993.

[5] L. R. Rabiner and R. W. Schafer, “Digital Processing of Speech Signals”, Prentice hall, Inc, 1978.

[6] J. D. Markel and A. H. Gray, “Linear Prediction of Speech”, Spring-Verlag Berline Heidelberg, 1976.

[7] S. Saito and K. Nakata, “Fundamentals of Speech Signal Processing”, Academic Publishers, Inc, 1981.

[8] J. Takami and S. Sagayama, “Asuccessive state splitting algorithm for efficient allophone modelling”, ICASSP-92, pp. 573-576, Mar. 1992.

[9] M. Ostendorf and H. Singer, “Hmm topology design using maximum likelihood successive state splitting”, Computer Speech and Language, Vol. 11. pp. 17-41, 1997.

[10] Junko TOKUNO, Nobuhito INAMI, “Context-dependent Substroke Model for HMM-based On-line Handwriting Recognition”, 8th International Workshop on Frontiers in Handwriting Recognition, pp. 78-83, 2002.

[11] 서진호, 박호종, “디지털 통신 시스템에서의 음성 인식 성능 향상을 위한 전처리 기술”, 한국음향학회지, 제14권, 제7호, pp. 416-422, 2005.

[12] Furi and Sondhi, “Advances in speech signal processing”, Marcel Dekker, Inc. 1992.

[13] 이윤근, 박준, 김상훈, “음성인터페이스 기술”, ETRI, 전자통신동향분석, 제20권, 제5호, 2005.

[14] 최경일,“HMI SCADA system의 기술동향”, 계장기술(계측제어자동화기술전문지), 1999.

이 종 욱

1984년 서울시립대학교 전자공학과(공학사)

1993년 광운대학교 컴퓨터공학과(공학석사)

2007년 선문대학교 컴퓨터정보학부(공학박사)

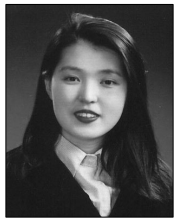
1995년~현재 한국폴리텍Ⅰ대학 제주캠퍼스 전자통신학과 부교수

이 정 배

1981년 경북대학교 전자공학과 (공학사)
1983년 경북대학교 전자공학과 (공학석사)
1995년 한양대학교 컴퓨터 공학부(공학박사)
2002년～현재 선문대학교 컴퓨터공학부 교수

황 영 섭

1989년 서울대학교 컴퓨터 공학과(공학사)
1991년 포항공과대학교 컴퓨터 공학과(공학석사)
1997년 포항공대대학교 컴퓨터 공학부(공학박사)
1997년～2002년 한국전자통신연구원 선임연구원
2003년～현재 선문대학교 컴퓨터공학부 교수



남 지 은

2000년 원광대학교 컴퓨터 공학과(공학석사)
2004년～현재 선문대학교 컴퓨터공학부 (박사과정)