論文 98-23-6-5

TTS 적용을 위한 음성합성엔진

正會員 이 희 만*, 김 지 영*

Speech Synthesis Engine for TTS

Heeman Lee*, Ji-yeong Kim* *Regular Members*

요 약

본 논문은 컴퓨터에 입력된 문자정보를 음성정보로 변환하기 위한 음성합성엔진에 관한 것이며, 특히 명료성의 향상을 위해 파형처리 음성합성방식을 이용한다. 음성합성엔진은 컴맨드 스트림의 제어에 따라 자연성의 향상을 위한 피치조절, 길이 및 에너지 등을 제어하며 음성합성단위로서 반음절을 사용한다. 엔진에서 사용 가능한 컴맨드를 프로그램하여 음성합성엔진에 입력함으로서 음성을 합성하는 방식은 구문분석, 어휘분석 등의 하이레벨과 파형의 편집 가공 등의 로우레벨을 완전 분리하므로 시스템의 융통성과 확장성을 높인다. 또한 TTS시스템의 적용에 있어 각 모듈을 객체/컴포넌트(Object/Component)로 각 모듈이 상호 독립적으로 작동되도록 하여 쉽게 대체가 가능하다. 하이 레벨과 로우 레벨을 분리하는 소프트웨어 아키텍처는 음성합성 연구에 있어 각각 여러 분야별로 독립적으로 연구수행이 가능하여 연구의 효율성을 높이며 여러 소프트웨어의 조합사용(Mix-and-Match)이 가능하여 확장성과 이식성을 향상시킨다.

ABSTRACT

This paper presents the speech synthesis engine that converts the character strings kept in a computer memory into the synthesized speech sounds with enhancing the intelligibility and the naturalness by adapting the waveform processing method. The speech engine using demisyllable speech segments receives command streams for pitch modification, duration and energy control. The command based engine isolates the high level processing of text normalization, letter-to-sound and the lexical analysis and the low level processing of signal filtering and pitch processing. The TTS(Text-to-Speech) system implemented by using the speech synthesis engine has three independent object modules of the Text-Normalizer, the Commander and the said Speech Synthesis Engine those of which are easily replaced by other compatible modules. The architecture separating the high level and the low level processing has the advantage of the expandibility and the portability because of the mix-and-match nature.

I. 서 론

최근 인터넷의 열풍과 함께 멀티미디어의 중요성이 부각되면서 멀티미디어의 정보처리와 멀티미디어를 이용한 다양한 응용이 출현하고 있다. 멀티미디어 정보처리는 숫자와 문자를 주로 처리하던 기존의 정보처리방식에서 벗어나 인간에게 친숙한 시각, 청각

* 서원대학교 전자계산학과
論文番號:97291-0819
接受日字:1997年 8月 19日

형태의 정보를 처리하여 컴퓨터 사용자에게 편리함을 제공한다. 사람의 주요 의사 소통인 언어를 사용하여 컴퓨터 즉 기계와 인간이 대화할 수 있는 기능(Man-machine Interface)은 멀티미디어 정보시대를 맞아 그 필요성이 어느 때 보다도 요구되고 있다. 인간의 언어를 컴퓨터가 인식하는 음성인식기술과 필요한 음성을 합성하여 인간에게 들려주는 음성합성 기술은 그동안 많은 노력에도 불구하고 실생활에 직접 이용하기에는 아직 초보적 단계에 있다. 음성인식은 다양한 화자에 대한 인식 율의 향상이 목표로 현재 화자종속인 경우 많은 기술적 발전을 하였으나 불특정 다수에 대한 인식 율은 아직도 많은 연구의 필요성이 있다. 음성합성의 경우 합성음의 명료성과 자연성 향상이 목표로 현재 부분적 상용화 단계에 접어들고 있다.

본 논문은 컴퓨터에 입력된 문자정보를 음성정보로 변환하는 음성합성에 관한 것으로, 특히 명료성의 향상을 위해 파형처리 음성합성방식을 이용하였으며 아울러 자연성의 향상을 위한 피치조절, 길이 및 에너지 제어를 명령어(command)를 기반으로 쉽게 제어 가능하도록 하는 음성합성엔진에 관한 것이다. 음성합성엔진에 관한 내용을 설명을 하기 전에 먼저 음성합성에 대한 개요를 설명하고 파형처리 합성방식의 장점과 현안 문제점을 검토하기로 한다.

II. 음성합성 개요

기계적 구조를 가진 음성합성기는 18세기 후반에 Von Kemplean에 의해 기계식으로 제작되었으며 모음과 자음의 음을 벨로우로부터 발생되는 공기를 제어하여 공명장치에 유입시킴으로서 발생하였다[1]. 그러나 오늘날 여러 가지 용도에 사용되고 있는 전기적 구조의 음성합성기의 원형은 1939년에 H. Dudley에 의해 개발된 Voder라는 장치로, 발판 및 건반을 이용해 기본주파수 및 공진특성을 제어하여 음성을 합성했다. 이 장치는 인간의 음성기관의 동작을 시뮬레이션한 것으로 음성생성모델에 기초를 하였다. 이러한 음성생성모델에 기초하는 음성합성기를 생성원 처리에의한 음성합성기라 한다. 그러나 음성파형 그 자체에도 예를 들면 영교차(Zero Crossing) 간격처럼 음성의 여러 가지 특징이 포함되어 있다. 음성파형에 포함되어 있는 특징에 기초하는 음성합성법에 대해서도 여러 가지 방법이 고안되었는데 이것을 파형처리에 의한 음성합성기라 한다.

2.1 생성원 처리에 의한 음성합성방법

생성원 처리에 의한 음성합성방법의 하나로 조음합성방식이 있다. 조음합성방식은 조음기관의 움직임을 모델링한다. 즉 조음기관의 움직임, 조음기관의 위치, 성도의 모양을 X-선 등을 이용한 관측자료를 통해 혀의 위치, 턱의 높이, 입의 벌어진 정도, 후두의 높이 등을 파라메터로 하여 운율제어, 발성속도 음가변환 등의 복잡한 현상을 해결하고자 한다. 그러나 인간의 조음기관은 매우 자유도가 많은 조음기관의 복합적 조합에 따라 음을 생성되므로 실제 구현하는 데는 어려움이 많다.

포르만트형 음성합성기는 인간의 성도(Vocal Tract)를 여러 가지 공진회로 전달함수(Transfer function)의 직렬 및 병렬구조로 모델링하는데 이는 모음부분의 신호는 일정주기를 갖는 신호(파형)로 물리적인 면에서 보면 일정 주기로 진동하는 성대를 통과한 공기가 발성기관에서 공명될 때 생성되기 때문이다. 성도의 공진 주파수를 포르만트(Formant)라 한다. 병렬구조로 된 포만트 합성기의 음성합성이 직렬구조에 비해 비교적 간단하게 생성시킬 수 있으므로 보다 더 선호된다. 모음의 경우 성대가 일정주기로 진동하므로 여기신호를 일정주기로 갖는 펄스열로 하며 무성자음의 경우 성대의 진동이 없는 비주기의 파형이므로 백색신호(잡음)를 입력 여기 신호로 한다. 그러나 좀더 사람의 음성과 유사하도록 여러 형태의 입력 여기신호도 제안되고 있다[10, 11]. 그림 1은 인간의 성도를 3개의 공진회로를 병렬로 연결하여 모델링하였는데 구강에서 발생되는 모음과 비강에서 발생하는 비음, 성대가 진동하지 않는 무성자음을 위한 전달함수를 각

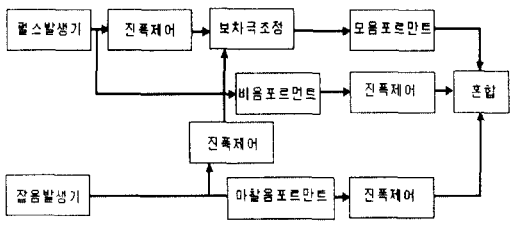

그림 1. 포르만트 음성합성기
Fig. 1 Formant Speech Synthesizer

각 병렬로 구성한다. 한국어 합성에 포만트방식을 적용한 연구가 있으나 우리말의 포만트에 대한 데이터가 적기 때문에 연구에 어려움이 많다.

선형예측(LPC) 합성법은, 음성신호는 데이터가 서서히 변화하는 성질을 이용한다. 즉 표본간의 상관성이 높으므로 현재 및 과거의 N개 데이터로 미래의 값을 예측할 수 있다. 이 예측 계수를 이용하여 all-pole 성도 모델 필터를 구성하여 음성신호를 코딩하며 이 계수를 저장 보관한 후 합성시에 저장된 계수를 필터로 불러와 복원하여 녹음된 음편을 연결 합성한다. LPC합성은 유성음의 경우 좋은 합성음을 만들어 낼 수 있으나 비음의 처리가 미흡하다. 선형 예측 계수를 구하는 방법으로는 Autocorrelation법, Covariance법, Lattice법 등이 있으며 각각의 방법은 연산량, 저장 데이터의 양, 안정성 등에서 장단점을 갖는다. 실제 선형 예측 계수는 저장 또는 전송할 때에는 일정한 비트로 양자화하는데 양자화 오차에 의해 안정성을 보장할 수 있는 선형 예측 계수의 범위가 명확하지 않다. 이 문제를 해결하기 위하여 실제 구현 시에는 PARCOR 계수를 많이 이용한다[9]. PARCOR 분석합성 시스템은 LPC 분석에 의해 얻어지는 스펙트럼의 포락이 불안정하기 때문에 편자기 상관함수(Partial Autocorrelation)를 이용하여 파라메터를 추정하는 방법이다.

2.2 파형처리에 의한 음성합성방법

단어나 음절단위, 또는 반음절이나 음소단위의 음성을 미리 녹음하여 필요시 연결 합성하는 방법으로 합성음질은 비교적 좋지만 자연성 확보가 어렵다. 또한 음성 데이터 베이스의 크기가 생성원 처리 방식보다 클 뿐만 아니라 합성처리 시간이 많이 소요되어 과거 컴퓨터환경에서는 현실성이 없었으나 컴퓨터 기술의 향상으로 메모리 용량이 커지고 프로세서의 처리속도가 매우 향상되어 파형처리 방식에 의한 음성합성은 80년대 후반부터 많이 연구되고 있다. 단순하게 글자를 각각 따로 녹음하여 필요시 이를 단순 연결 합성하는 방법은 매우 자연스럽고 실제적인 방법 같지만 자연스런 합성음을 얻을 수 없다. 그 이유는 한 글자의 발음 지속시간이 각 글자를 각각 한자씩 발음하는 지속시간의 1/2정도보다도 짧으므로 연결 합성시에 매우 느리고 답답함을 느끼게 되기 때문이다. 또한 같은 단어라도 문장에서의 역할에 따라 에너지의 변화, 길이의 변화 및 피치의 변화가 다양하게 변하므로 단순 연결만으로는 자연스런 합성음을 얻을 수 없어 연결합성 방식은 실용성이 없었다. 그러나 피치를 비교적 양질의 음을 유지하면서 변경할 수 있는 TD-PSOLA방식의 제안은 파형처리 방식의 문제점이었던 운율제어가 가능하도록 하며 아울러 명료성이 높은 이유로 파형 편집방식의 연결합성방식도

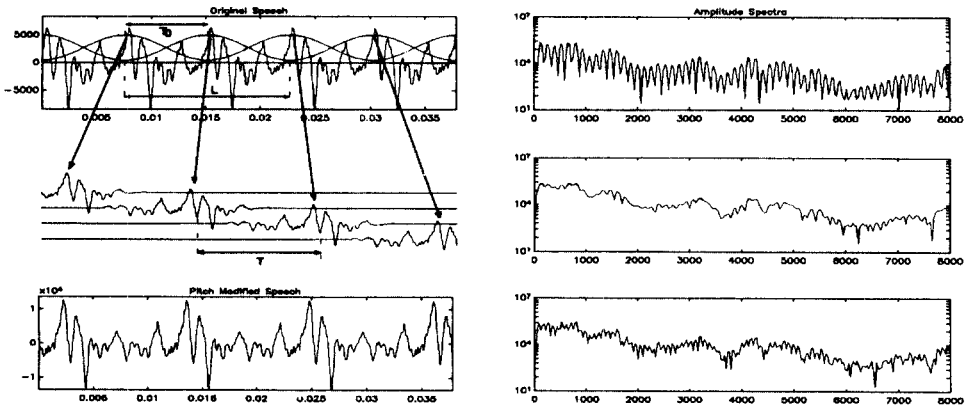

그림 2. TD-PSOLA 피치변경방법
Fig. 2 TD-PSOLA Pitch Modification Method

각광을 받게 되었다[2, 3]. TD-PSOLA의 운율제어를 위한 피치의 조절은 음성 파형에서 피치단위로 음성을 분해하고 분해된 피치의 위치를 원래의 순서에 따르되 재배치하여 조절한다(그림 2. 참조). 지속시간의 변경은 피치단위의 파형을 단순 복제하거나 절단함으로써 파라메터 처리방식의 음성합성보다 명료성과 자연성을 개선하여 유럽과 일본 그리고 한국에서 이를 이용한 음성합성 시스템이 개발되고 있다. 그러나 TD-PSOLA방식에는 다음과 같은 문제점이 있다[5]. 즉 각각 다른 단어에서 추출한 세그먼트를 연결합성하는 경우 모든 파형이 같은 조건에서 연결하는 것이 아니므로 위상(Phase), 피치(Pitch), 주파수 포락선(Spectral Envelope)의 불일치가 발생하며 에너지의 불균형과 피치를 매우 높일 경우 코러스현상이 나타난다. 위상의 불일치는 세그먼트 윈도우상의 파형의 위치가 각각 조금씩 다른 경우에 발생하여 왜곡된 파형으로 합성된다. 피치의 불일치는 성대의 긴장여부에 따라 피치가 달라지는데 음성 데이터베이스내의 피치를 일정하게 유지하는 것은 매우 어렵기 때문이다. 주파수 포락선의 불일치는 연결합성의 가장 결정적인 단점으로 지적되고 있다. 이 현상은 같은 화자가 같은 음편을 녹음하여도 조금씩 파형이 다르다는 점과 주변 음운환경에 의해 음가가 조금씩 변하기 때문이다. 본 논문에서는 연결합성방식을 사용하였고 피치의 조절은 TD-PSOLA방식을 이용하였다.

III. TTS 시스템에의 적용

3.1 시스템개요

TTS(Text-to-Speech) 시스템은 텍스트 문장을 자연스런 음성으로 출력하는 시스템이다. 텍스트문장을 음성합성으로 출력하기 위해서는 여러 단계의 전처리를 거쳐야 하는데 본 연구에서는 각 단계를 객체/컴포넌트(Object/Component:이하 객체라고 칭함)로 설계하여 상호간 독립성을 유지하도록 하여 소프트웨어의 유지보수를 쉽게 하였으며 또한 다른 연구자가 어느 특정모듈만을 개발하여 본 시스템과 용이하게 접목할 수 있도록 하였다.

그림 3은 본 연구에서 음성합성엔진을 TTS에 적용한 시스템의 구성도이다. 음성합성엔진은 컴맨드 스트림에 의해 파형을 처리하고 합성한다. 컴맨더 객체는 텍스트 문장을 분석하고 이를 음성으로 합성하기 위해 음성합성엔진을 기동하는 컴맨드를 발생한다. 컴맨더 객체는 텍스트 정규화 객체가 처리한 텍스트 스트림을 입력으로 한다. 텍스트 문장은 일반 문자외에도 약어, 축약어, 숫자, 시간, 특수문자 등이 내포되어 있는 데 이를 음성으로 합성하기 전에 일반 텍스트 문장으로 변환 할 필요가 있다. 예를 들면, 시간 표시로 "3:30"를 "세시 삼십분", 주소에서 "(우)370-470"를, "우편번호는 삼백칠십 대쉬 사백칠십" 등으로 변경하는 데 이를 텍스트정규화(Text Normalization)이라 하며 텍스트정규화 객체(Text Normalizer Object)에서 처리한다.

컴맨더 객체(Commander Object)는 텍스트정규화 객체에서 처리한 텍스트의 한글코드를 음운변화 등의 처리를 위해 완성형코드에서 조합형 코드로 변환하며 이를 다시 음성 데이터베이스 위치와 관련이 있는 내부코드로 변환한다(윈도우95 플랫폼은 KSC5601 완성형코드를 사용한다). 내부코드는 조합형코드의 일종이다. 컴맨더 객체 내의 문자소리처리모듈(Letter-to-Sound)에서는 정규 맞춤법에서 소리나는 데로 글자를 표기한다. 텍스트 문장은 글자 그대로 발음되지 않기 때문이다. 예를 들면 "가슴살"이 "가숨쌀", "가슴속"이 "가숨쏙" 등의 경음화나 또는 구개음화 현상이라든가 또는 음운 환경의 변화에서 오는 음가의 변동, 예를 들면, "부부"의 "ㅂ"이 각각 [p/b]로 변화하기 때문이다. 자연스런 합성음을 생성하기 위해서는 많은 규칙을 내포하고 있어야 한다. 다음단계는 구문분석(Syntactic Parser)으로 텍스트 문장의 문법을 분석하여 단어의 품사를 변별하고 의문문, 평서문 등에 따라 운율제어를 위한 정보를 분석한다. 분석한 정보는 운율제어(Prosody Rules)에 사용된다. 국어의 운율형태를 결정하는 주요자질은 액센트, 리듬, 억양, 휴지등이 있다[7]. 즉 운율제어라 함은 음높이(Pitch)의 고저, 소리의 크기(Amplitude), 소리의 장단, 리듬 등을 제어하는 것을 말한다. 액센트는 단어내의 상대적인 돋들림으로 음의 고저와 강약, 장단에 의해 결정된다. 리듬은 단어가 모여서 문장을 이룰 때 그 문장내에서 일정한 운율형태를 반복하는 것을 말하며 국어 연속 음성에서는 앞 뒤 단어들 사이에 피치의 높낮이가 변화하는 주기가 있으며 사람의 호흡시간과도 깊은 관계가 있다. 억양은 연속 음성 중 문장전체의 연속 피치곡선으로 국어의 경우 문장 끝의 억양이 문장전체의 의미적 정보를 나타내며 휴지 또한 중요한

운율요소로 휴지의 유무에 따라 의미가 달라진다. 예를 들면, "아버지가 방에 들어 가신다"와 "아버지 가방에 들어가신다."는 전혀 다른 의미가 된다.

음성합성의 명료성은 파형연결 처리시 파형의 연속변화성(Smotheness)을 유지하면 금속성음 등의 불쾌한 음을 쉽게 제거 할 수 있다. 즉 현재의 음성파형은 과거의 N개의 데이터로부터 예측가능하기 때문이다. 일반적으로 피치의 변경 및 연결합성 처리시 금속성음이 들리는 것은 파형이 부드럽게 공명되지 않고 고주파 성분을 많이 함유하기 때문이다. 금속성음은 저역필터 또는 제로 크로싱 연결 및 피치의 불연속변화방지 등의 방법에 의해 비교적 쉽게 해결 가능하다. 그러나 자연성 향상에는 방대한 데이터의 수집과 통계처리 및 모델링 등의 개발을 하여야 하므로 많은 노력과 시간이 필요하므로 팀웍단위의 협력이 요구된다. 컴맨드 발생기(Command Generator)에서는 운율정보에 따라 음성엔진을 가동하기 위한 컴맨드를 생성한다. 음성합성엔진 객체(Speech Synthesis Object)는 명령어에 따라 음성데이터 베이스에 있는 데이터를 가공 연결하여 음성으로 출력하는데 다음 장에서 논의한다.

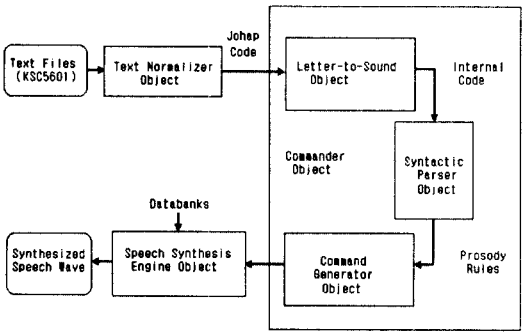

그림 3. TTS 시스템 구성도
Fig. 3 TTS System Structure

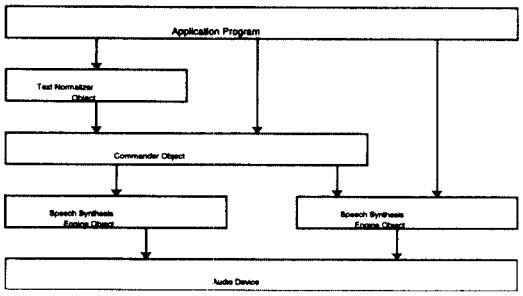

그림 4. TTS 시스템 구조
Fig 4. TTS System Archetecture

응용프로그램에서 음성합성엔진을 이용하여 TTS시스템을 구축하는 방법을 그림 4에 보였다. 음성합성엔진은 컴맨드 스트림에 의해 기동되는 프로그램어블 엔진으로 컴맨드를 발생할 수 있는 응용프로그램에서 직접 음성합성엔진을 구동하거나, 또는 컴맨더 객체 등의 전문 소프트웨어를 통해 구동할 수 있도록 함으로서 시스템 구축시 융통성을 향상시킨다. 음성합성엔진은 가상장치(Virtual Device)로 복수개의 음성합성엔진을 가동할 수 있다.

3.2 음성 데이테베이스의 구축

음성의 데이터 베이스를 구축하기 위해서는 합성단위를 선정하여야 한다. 우선 음소를 합성단위로 선정할 경우 국어는 초성 18개, 중성 21개 종성 7개로 46개의 기본 음편이 필요하며, 그외 변이음을 고려한다해도 60개미만의 적은 데이터 베이스로 무제한 음성합성기를 작성할 수 있다[9]. 그러나 음소단위의 연결합성의 경우 충분한 품질의 합성음을 얻기가 어려운데 이유는 조음효과(Coarticulatory Effect)에 기인한다. 조음의 효과는 음소파형 중간부분에서 가장 최소가 되는데 이 원리를 이용하여 음성을 합성하는 Diphone방식이 제안되었다[2]. Diphone는 음운 천이부를 포함하므로 비교적 적은 데이터로 합성의 명료성을 확보할 수 있으나 조음결합에 의한 음운변화를 완전히 포함하지는 못한다.반음절로 데이터 베이스를 구축하는 경우, 초성+중성 조합이 399개 중성+종성 조합이 147개로 기본적으로 546개의 음편이 필요하며 기타 변이음 및 천이구간 음편을 고려하면 약 700여개의 음편으로 비교적 고품질의 음성합성기를 작성할 수 있다. 반음절은 모음의 안정구간을 기준으로 전후로 양분된 데이터를 합성하는 방식으로 Diphone의 연결시 불연속 문제점이 없어 음질이 양호하지만 자음과 자음에 대한 데이터가 없어 음절과 음절사이의 조음결합에 잘 대처하지 못하므로 반음절과 Diphone을 병행하는 방법은 좋은 결과를 얻을 수 있다. 음절단위로 합성시스템을 제작할 경우, 2650자의 기본 음편과 변이음처리를 고려한다면 3600여자 정도의 음편을 녹음하여야 한다.

본 연구에서는 반음절 단위의 합성용 음성데이터 베이스를 사용하였다. 음성자료를 따로따로 녹음하여 데이터 베이스를 구축하는 경우 F0피치변화 패턴은 초반부에서 중반부까지 피치가 서서히 상승하고 중반부에서 후반부까지는 급격히 하강한다. 이를 연결 합성하면 합성음은 연속한 음이 아닌 한음씩 따로따로 들리게 된다. 이를 해결하기 위해서는 파형의 피치를 변경하여야 한다. 그러나 가공처리 단계를 추가할 때마다 음질이 저하되는 문제점이 있다. 본 연구에서 개발한 음성합성엔진은 반음절 단위로 구축된 데이터 베이스를 사용하고 있으나 음성합성엔진에서 합성단위마다 내부코드를 부여하므로 그 어떤 음성단위라도 쉽게 적용하여 사용할 수 있다.

IV. 음성합성엔진

음성합성엔진은 개체(Objects)로 구성된 소프트웨어 컴포넌트(Component)로 재사용과 확장이 가능하다. 음성합성엔진은 순수 소프트웨어만으로 구성하였으나 처리속도 향상을 위해 전용의 하드웨어로 제작할 수도 있다. 음성합성엔진은 연결합성을 위한 명령어 기반의 프로그램이 가능한 엔진(Programmable Engine)으로 현재 20여 가지의 명령어가 있으나 추가 및 삭제가 용이하다. 엔진은 컴맨드 스트림을 입력으로 하며 복수개의 음성 데이터뱅크를 사용할 수 있다. 즉 회화체 문장인 경우 화자의 목소리를 프로그램 수행 중에 동적으로 변경 가능하여 기존 한사람의 목소리로만 출력되는 TTS를 개선한다. 엔진내부에는 컴맨드 인터프리터가 있어 해당 명령어를 해독처리하며 5개의 독립된 임시메모리(Buffer)는 음성파형을 임시 보관한다. 그림 5는 음성합성엔진의 블록다이어그램이다.

4.1 음성합성엔진 명령어

명령어는 가변길이의 크기를 가지며 대표적인 명령어를 표 1에 요약하였다. 표 1에서 알 수 있는 바와 같이 음성합성엔진은 데이터뱅크에서 특정 코드에 해당하는 음성 데이터를 엔진 내부에 있는 버퍼에 적재한 후에 피치, 길이 및 에너지를 변경하며, 이미 처리되어 있는 다른 음성 데이터와 여러 가지 옵션에 의한 연결을 하고, 최종적으로 연주(출력)하는 기능을 한다. 음성합성엔진은 전자악기 MIDI의 시퀀서개념을 도입하였으며 명령어 16을 사용하여 특별히 데이터 뱅크를 변경하지 않으면 기본 데이터뱅크(Bank #0)에 있는 음성파형을 사용하지만 언제든지 가변가능하며 다른 뱅크를 선택하기 전까지는 현재 선택된 뱅크가 계속 유효하다.

4.2 피치조절

피치조절방법은 TD-PSOLA방식을 사용하였다[2]. 이 합성방식은 다음과 같이 크게 3단계로 구성된다. 제1단계, 피치단위의 분석(Pitch-synchronous Analysis) 단계이며, 제2단계, 제1단계에서 분석된 피치단위의 변경단계, 제3단계는 변경된 피치들의 재합성단계이다. 제1단계에서는 피치단위별로 윈도우함수를 곱하여 ST-Signal(Short Term Signal)를 만든다. 윈도우함수로는 일반적으로 해닝윈도우(Hanning Window)를 사용하며, 해닝 윈도우를 이용하는 이유는 깁스현상을 줄이기 위함이나 본연구에서는 삼각윈도우를 사용하되 제로크로싱 부분을 윈도우의 변두리(Boundary)로 하여 깁스현상(Gibbs Phenomenon)을 줄이면서 처리시간을 높여 실시간 사용이 가능하도록 하였다. 이로 인해 윈도우의 센터는 일반적으로 중심피치(피치마크) 부분을 선정하지만 본 연구에서는 중심피치 파형이 시작되는 제로 크로싱부분을 선정한다. 무성음의 경우는 일정시간을 윈도우로 선정한다. 피치마킹은 PSOLA방법의 중요한 요소로 잘못 마킹시 명료한 합성음이 나오지 않는다. 본 연구에서는 자동적으로 프로그램에 의해 피치마킹을 하는데 음성신호는 스테이셔너리(Stationary)시그널이 아니므로 피치마킹시에 윈도우의 센터가 원하는 위치에 존재하지 않으므로 위상불결합(Phase Mismatch)을 유발하는 문제점이 있다. 피치마킹을 음성데이터베이스 구축시 핸드마킹방법에 의해 실시하면 이 문제는 해결할 수 있다고 본다. 그러나 피치불결합(Pitch Mismatch)이나 주파수포락선 불결합(Spectral Envelope Mismatch)은 많은 양의 데이터 베이스를 구축함에 있어 피할 수 없는 PSOLA방법의 한계이다.

4.3 지속시간 조절

음성의 지속시간은 모음부분의 지속시간에 따라 결정되므로 모음부분의 한 구간 파형을 복사하여 삽입하거나 삭제함으로서 제어한다. 그러나 특정위치의 파형만을 삽입하여 지속시간을 길게 하는 경우 해당 피치 값만 일정기간 일정하게 유지되므로 매우 어색

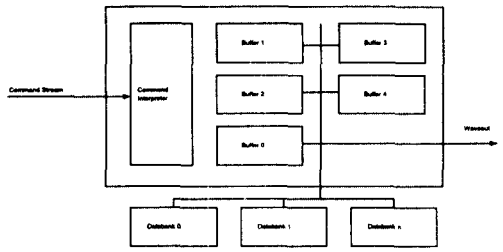

그림 5. 음성합성엔진
Fig. 5 Speech Synthesis Engine

표 1. 음성합성엔진의 주요 명령어
Table 1. Commands for Speech Synthesis Engine

Command	Descriptions
0	Stop engine(end of command stream) (ex) 0
1	Load wave data from the selected databank (ex) 1 <code#> <buff#>
2	Clear buffer (ex) 2 <buff#>
3	Adjust data to be the specified duration(mili second) (ex) 3 <S buff> <time> <D-buff>
4	Append data from one buffer to another buffer (ex) 4 <S buff> <D buff>
5	Stich data in two buffers (ex) 5 <S %> <D buff> <D%> <# of piches to be m,ixed>
7	Modify parts of pitches smoothly (ex) 7 <S buff> <s%> <e%> <s-pitch> <e-pitch>
8	Add silience (ex) 8 <S buff> <time>
10	Make the desired pich(Hz) (ex) 10 <S buff> <s%> <e%> <Hz><D buff>
12	Adjust pitch and stitch data (ex) 12 <S1 buff> <s%> <S2 buff> <e%>
13	Mix and append part of data in two buffers with reverve option (ex) 13 <S buff> <s%> <D buff>
14	Energy control (ex) 14 <S buff> <power%> <D buff>
16	Select databank(speech database) (ex) 16 <bank number>
97	Play wave data (ex) 97 <S buff>

하고 불멘소리가 들리게 된다. 지속시간의 변경시에는 모음의 안정기간 부분의 신호를 균등하게 복사 삽입하여야 자연스럽게 조절이 된다. 그림 6은 "가" 음의 원음과 균등하게 복사한 파형과 특정 파형구간만을 복사 삽입한 경우의 파형과 피치의 변화를 보인 것이다. 균등복사 삽입의 경우가 보다 원음과 유사한 F0 변화패턴을 갖음을 알 수 있다.

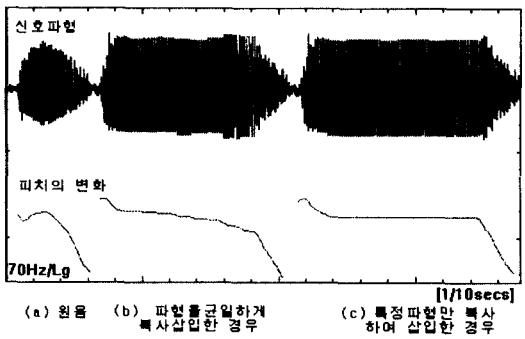

그림 6. 지속시간 변경방법 비교
Fig. 6 Comparisons of Two Duration Control Methods

4.4 에너지 조절

운율제어라 함은 음높이(Pitch)의 고저, 소리의 크기(Amplitude), 소리의 장단, 리듬 등을 제어하는 것을 말한다. 이 중 가장 인간의 귀에 민감한 것이 피치의 변화, 즉 주파수의 변화이며 그 다음은 소리의 장단 및 리듬이고 가장 덜 민감한 것이 소리의 크기이다. 하지만 에너지도 운율제어요소에 있어 중요한 요소이다. 에너지의 조절은 비교적 쉽게 할 수 있다. 즉 특정 피치 구간의 최대값을 산출한 후 원하는 에너지가 되도록 비율을 구하고 같은 구간에 있는 데이터 값에 방금 산출한 비율값으로 곱한다. TD-PSOLA 방식의 피치변경은 피치간격의 변화에 따라 주파수를 조절하는데 이때 에너지의 불균형이 생긴다. 그러나 에너지는 자연성과 명료성에 크게 영향이 없으므로 본 연구에서는 에너지 정규화(Energy Normalization) 과정을 수행하지 않는다.

4.5 피치조절 연결합성

음성합성엔진은 음성단위에 무관하게 연결합성할 수 있지만 반음절 단위의 음성데이터 베이스를 사용하고 있다. 두 개의 반음절을 연결 합성할 때 연결부위에

서 피치의 불연속이 생길 수 있다. 이는 음성데이터 베이스 구축시 피치를 조절하여 녹음하면 되지만 방대한 음성데이터 베이스구축시 피할 수 없는 문제점이다. 피치가 서로 다른 모음이 연결되면 두 개의 신호는 주파수 공간상에서 공존하므로 귀에서는 각각의 음성을 구별하게 된다. 이로 인해 단모음임에도 불구하고 2개의 모음소리 즉 복모음처럼 들리게 되어 부자연스런 합성음이 된다. 그림 7의 (a)는 피치가 다른 2개의 모음부분을 연결 합성한 경우이며 (b)는 연결한쪽 부분의 피치를 연결될 부분의 피치 값으로 조절한 후 연결한 경우로 자연스런 합성음을 얻을 수 있다.

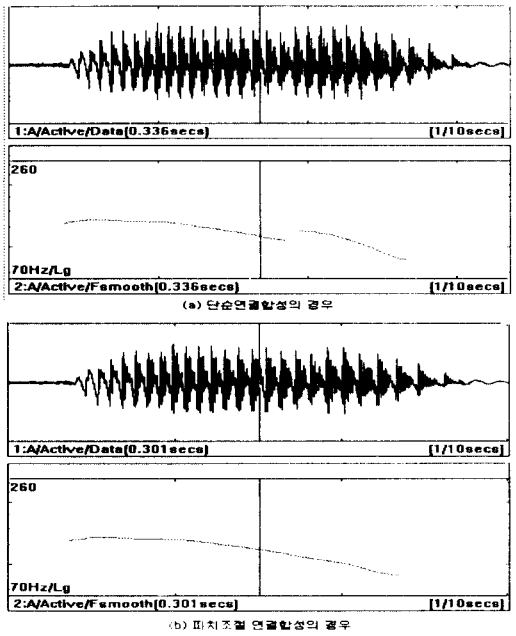

그림 7. 피치조절 연결합성
Fig. 7 Concatenation with Pich Adjustment

V. 음성합성실험

음성합성은 사운드카드가 장착된 퍼스널 컴퓨터, 윈도우95운영체제에서 소프트웨어만으로 실시간 동작하는 음성합성 엔진을 구동하여 실험하였다. 음성합성 데이터는 전자통신연구원에서 제공하는 반음절 단위의 합성용 남성화자 음성데이터로 데이터 크기 16비트, 샘플링 주파수 16KHz를 사용하였다. 음성 데이터는 엔진에서 사용하기 위해 에너지의 정규화 등 부분적 처리과정을 거쳐 특정 구조를 갖는 파일에 저장된다. 반음절 단위로 합성단위를 선정하였으므로 초성+중성의 조합 399개, 중성+종성의 조합 147개 종합 546개만을 사용하였으며 각각에 고유 내부코드를 부여하였다. 변이음 및 기타 추가되는 음은 547이후의 코드 부여로 쉽게 추가되므로 확장성이 좋다. 엔진은 코드번호를 이용하여 음성파형 데이터를 엔진내부에 적재한 후 가공 합성하므로 어떤 단위 즉 음소/음절 단위의 음성연결합성에도 쉽게 적용될 수 있다. 그러나 무제한 어휘의 음성합성시 중요한 요소는 자연성과 명료도의 향상이다. 음성파형 연결합성의 경우 합성음의 명료도는 매우 좋으나 자연성이 나쁘므로 이를 향상시키기 위해서는 음성파형의 단순연결보다는 자연적 운율이 되도록 가공과정이 필수적 인데, 가공 공정이 하나씩 추가될 때마다 일반적으로 음질은 계속 저하되는 문제점이 있다. 자연성을 향상시키기 위해 운율을 제어하기 위해서는 피치의 조절, 지속시간의 조절 및 에너지의 조절을 하여야 하는데 각각 7, 3, 14 등의 엔진명령어를 이용한다(표 1. 참조). 의문문이나 감탄문 및 평서문의 경우 어절 끝의 피치를 올리거나 내리거나 또는 일정하게 유지하도록 제어한다. 엑센트의 경우 피치를 올려주거나 지속시간을 길게 하거나 또는 에너지를 크기를 조절하며 리듬의 경우 이와 같은 패턴을 일정구간마다 반복함으로서 실현할 수 있다. 문장전체의 F0패턴은 낭독체와 대화체가 매우 다르게 나타나는데 현재 전자통신연구원을 비롯 연구소 및 학계에서 대화체에 대한 운율연구가 활발히 진행되고 있다. F0패턴이 점차 하강하는 패턴은 거의 모든 언어에서 공통으로 나타나는 현상인데 이는 허파에서 나오는 압력이 발음과정 중 점차 떨어지게 되어 성대의 진동주파수가 내려가는 물리적 현상에 기인한다. 그러므로 휴지 기간 후 처음 발음되는 음절의 피치는 높이고 그 이후는 점차 피치를 내리도록 제어하면 낭독체 문장의 운율을 비교적 쉽게 근사적으로 실현할 수 있다. F0피치 패턴은 전체적으로 불연속이 없도록 제어하는 것이 매우 중요하지만 일반적으로 음성데이터베이스 구축시 한자씩 녹음하면 한음절내에서 시작부분에서 피치가 증가되다가 모음의 정점에서 최대가 된 후 점차 감소하여 음절이 끝나는 부분에는 피치가 매우 낮아지게 된다(그

림 8(a) 참조). 그리하여 2개의 음절을 연결하면 피치가 연속되지 못하므로 외국인이 발음하는 듯이 한자 한자 끊어서 읽는 느낌을 주게 된다. 그림 8(b)는 음절지속시간을 길게 만들어 피치가 서서히 변하게 한 다음 파형의 후반 부분을 자르고, 다음 음절의 전체 피치를 낮춘 후 연결합성한 경우의 피치패턴의 변화를 보인 것이다. 이를 통해 자연성은 향상되었으나 신호처리단계를 여러번 거쳤으므로 명료성이 조금 떨어지며 종성이 있는 경우 상기 방법을 사용할 수 없으므로 이 방법은 부분적으로만 적용될 뿐 근본적 해결책은 아니며 데이터 베이스를 잘 구축하는 것이 가장 바람직한 해결책이라 생각한다.

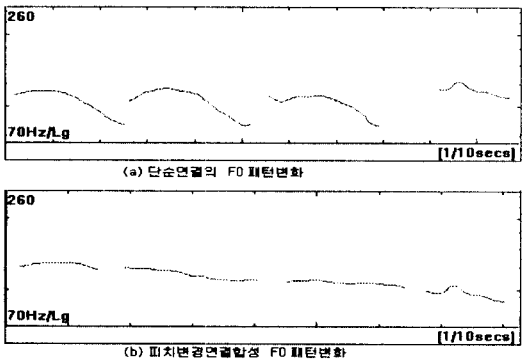

그림 8. 피치조절 단어합성
Fig. 8 Word Synthesis with Pitch Adjustment

본 연구에서는 비교적 적은 수의 음운변화 규칙과 운율제어 규칙을 사용하였다. 본 엔진의 TTS 적용과정은 다음과 같다. 텍스트파일을 메모리내로 읽어 들인 후 간단한 텍스트 정규화 과정을 한다. 예를 들면 숫자나 영어 스펠링 등 텍스트문장내의 글자를 소리나는 대로 한글로 표기한다. 다음은 완성형코드를 조합형으로 임시 변환한 후 내부처리 코드로 변환한다. 다음 단계는 국어의 경음화나 구개음화등 국어의 문법규칙에 따라 또는 연음처리 등의 문자소리처리(Letter-to-Sound)를 하고 최종적으로 종성대표음처리를 한다. 한국어의 종성의 종류는 많지만 실제로 소리나는 것은 7개밖에 없기 때문에 데이터베이스에는 종성 7개 종류만 있다. 다음단계는 음성합성엔진을 위한 명령어(Command)를 작성한다. 한 글자에 대한 명령어 작성시에는 낭독체 운율효과를 얻기 위해 최소 3-4개의 명령어가 사용된다. 첫 번째 명령어는 글자(내부코드)에 대한 해당 음성데이터를 메모리에 적제(Loading)하는 명령어(Command 1)이며 본 연구에서는 반음절 데이터베이스를 사용하므로 텍스트에 있는 문자가 종성이 있는 문자인 경우 이 코드에 해당하는 또 하나의 음성데이터를 메모리에 적재하여야 한다. 예를 들면 '학' 자의 경우 '하'에 대한 음성데이터와 '악'에 해당하는 음성데이터를 엔진내부에 있는 버퍼에 각각 적재한다. 다음은 음절합성을 위한 연결과정이며 연결방법에는 명령어가 여러 개(4, 5, 12, 13) 있으나 명령어 13를 주로 사용한다. 명령어 13은 2개 음절 모음정점부근에서 절단하여 연결하되 제로 크로싱 부분에서 결합하는데 그 이유는 피치정점부분에서 연결하는 경우 2개 파형의 에너지가 각각 다르므로 연결부분에서 고주파 성분이 생성되어 금속성 음이 들리게 된다. 이를 제거하기 위해서 저역 통과필터(LPF)를 사용할 수 있으나 필요한 고주파 신호까지 소멸되어 둔탁한 소리로 음질이 저하되며 또한 컴퓨터 처리 비용을 상승시키는 요인이 된다. 제로 크로싱 부분에서 연결하되 전단신호의 일부를 점차 감쇠시키며 후단의 연결되는 신호와 혼합(Mixing)한다. 다음은 길이의 조절(Command 3) 명령어이다. 데이터베이스에 저장된 데이터는 300msec 전후로 가급적 표준화를 하였지만 음절마다 고유지속시간이 다르고 또한 녹음시 이미 고정된 길이로 녹음되었으므로 그 길이가 최소 180msec에서 400msec일정하지 않다. 그러므로 음성합성시에는 데이터베이스에 저장된 데이터를 그대로 사용할 수 없으며, 문장에서의 역할 및 억양에 따라 그 길이가 달라 져야한다. 길이조종 후에는 피치조절 명령어를 사용하였다. 음절의 피치는 문장의 위치에 따라 달라지지만 낭독체의 경우 한숨구간(One Breathing Period)에서 피치가 서서히 떨어지는 일반규칙만을 적용하였다. 즉 피치는 공백이 없는 문자의 수에 따라 조절하되 두 번째 음절의 경우 첫 번째 글자보다 원래 녹음된 피치에서 2%씩 낮게 조절한다. 전술한 바와 같이 사람의 귀는 피치에 가장 민감한 반응을 보이므로 피치의 조절은 자연성여부의 가장 중요한 요소이다. 연결된 음성데이터의 피치가 불연속이 되지 않도록 연결하되 피치에 변화를 주는 기술이 음성합성기의 품질을 좌우한다. 본 연구에서는 변이음 데이터베이스가 없어 연결 합성실험을 하지 못하였지만 별도의 변이음에 대한 코드를 부여하고 음성시료를 확보한다면 자연성을 더욱 향상시킬 수 있을 것이

다. 그러나 피치의 조절만으로도 어느 정도 자연스런 변이음의 효과를 얻을 수 있었다. 예를 들면 '부부'의 소리에서 첫음절의 'ㅂ'과 두 번째 음절의 'ㅂ'음소의 소리가 각각 다른데 이를 피치가 점차 감소하는 '부' 음을 연결 합성하면 비교적 자연스런 합성음을 만들 수 있다. 표 2는 "바람과 햇님"이라는 문장을 음성으로 합성할 때에 엔진명령어의 일례를 보인 것이며, 그림 9는 표 2의 명령어에 의해 합성된 음성의 파형과 피치(F0)의 변화를 보인 것이다. 데이터뱅크에 같은 발성자의 피치가 각각 다른 같은 음소(음절)에 대한 음의 데이터를 복수 개를 제작한다면 좀더 정밀한 음성합성 제어를 할 수 있으며 아울러 남성 및 여성 또는 코러스도 쉽게 합성할 수 있다.

표 2. 음성합성을 위한 엔진명령어 일례
Table 2. Engine Commands for Speech Synthesis

Text	Engine Command						
초기화 명령어	2	0					
	16	0					
바	1	105	1				
	3	1	390	3			
	7	3	0	100	90	2	125
	13	0	10	2	10	5	
람	1	63	1				
	1	403	2				
	12	1	50	2	30	0	
	3	1	234	3			
	7	3	3	100	88	2	123
	13	0	10	2	5	5	
과	1	9	1				
	3	1	416	3			
	7	3	3	100	86	2	86
	13	0	10	2	10	5	
	8	0	120				
햇「핸」	1	274	1				
	1	407	2				
	12	1	50	2	30	0	
	3	1	312	3			
	7	3	0	100	84	2	119
	13	0	10	2	10	10	
님	1	41	1				
	3	1	234	3			
	7	3	0	100	82	2	117
	13	0	10	2	5	5	
종료 명령어	97	0					
	0						

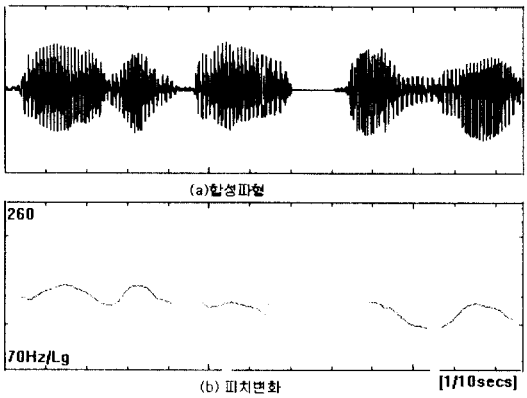

그림 9. 엔진에 의한 음성합성파형
Fig. 9 Synthesized Speech Waveforms

VI. 결 론

본 연구에서는 한국어 음성합성을 위한 프로그램어블 음성합성엔진을 개발하였으며 반음절의 음성합성 단위를 사용한다. 음성합성엔진는 TTS시스템을 구축하여 실험하였다. TTS적용에 있어 각각의 전처리 모듈을 객체(Object)로 설계하여 각 모듈이 독립적으로 쉽게 대체 가능하도록 하였다. 엔진에서 사용 가능한 명령어들로 구성된 컴맨드 스트림을 음성합성엔진에 입력함으로서 음성을 합성하는 방식은 구문분석, 어휘분석 등의 하이레벨과 파형의 편집 가공 등의 로우레벨을 완전히 분리 가능하므로 시스템의 융통성과 확장성을 높인다. 즉 하이레벨과 로우레벨을 분리하는 아키텍처는 팀워크 단위 및 공동연구에 있어 효율성을 높이며 또한 공개식 구조(Open Architecture)는 여러 소프트웨어의 조합적 사용(Mix-and-Match)이 가능하도록 하여 확장성과 이식성을 향상시킨다. 음성의 연결합성에 있어 본 구조를 사용하여 합성모듈을 용이하게 구성 및 이용할 수 있으므로 음성합성의 연구가 활성화되고, 또한 멀티미디어 시대에 음성을 이용한 응용 프로그램의 수요가 급증하고 있는 바, 본 연구가 이에 기여되길 희망한다.

참 고 문 헌

1. Gordon E. Pelton, Voice Processing, McGRAW-HILL, pp.13-32, 1993.

2. E. Moulines, F.J. Charpentier, "Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones", Speech Communication, vol. 9, no. 5-6, pp.453-467, 1990.
3. F.J. Carpentier, M.G. Stella, "Diphone Synthesis Using An Overlap-Add Technique for Speech Wave-Forms Concattenation", Proc. ICASSP, pp.2015-2018, 1986.
4. F.J. Carpentier, E. Mouliens, "TTS Algorithms Based on FFT Synthesis", ICASSP, pp.667-670, 1988.
5. Thierry Dutoit, Henri Leich, "MBR-PSOLA:Text-to-Synthesis Based On FFT An MBE Re-Synthesis of the Segments Data- base", Speech Communication, vol.12, 1993
6. 양진석, 김재범, 이정현, "운율 및 길이 정보를 이용한 무제한 음성합성기의 설계 및 구현", 한국정보처리학회 논문지, vol.3, no.5, pp.1121-1129, 1996.
7. 정국, 구희산, 이찬도, 김종미, "음성인식/합성을 위한 국어의 음성-음운론적 특성연구", 한국음향학회지, vol.13, no.6, pp,31-43, 1994.
8. 조철우, 김경태, 이용주, "합성음성평가를 위한 다음절 무의미 단어 생성과 이용에 관한 연구", 한국음향학회지, vol.13, no. 5, pp.51-58, 1994.
9. 박애희, 양진우, 김순협, "음소단위를 이용한 소규모 문자음성변환 시스템의 설계 및 구현", 한국음향학회지, vol. 14, no.3, pp.49-60, 1995.
10. A. Rosenberg, "Effects of Glottal pulse Shape on the Quality of Natual Vowels", J. Acost. Soc. Am, no.49, pp.583-590, 1971.
11. I. Titze, D. Talkin, "A Theoretical Study of the effects of the various Laryngeal Configurations on the Acoustics of Phon- ation", J. Acoust. Soc. Am. no.66, pp.60- 74, 1979.

이 희 만(Heeman Lee) 정회원
1984년 2월:고려대학교 전자공학과(공학사)
1986년 2월:한국과학기술원 전기 및 전자공학과(공학석사)
1994년 6월:Texas A & M Electrical Eng.(Ph.D)
1986년 1월~1990년 7월:산업연구원(KIET) 연구원
1994년 4월~1996년 1월:삼성중공업 중앙연구소 선임연구원
1996년 3월~현재:서원대학교 전자계산학과 조교수

김 지 영(Ji-yeong Kim) 정회원
1977년:뉴욕 주립대학교(SUNY-Binghamton)경영학 석사(MBA)
1979년:뉴욕 주립대학교(SUNY-Binghamton)전산학 석사
1984년:뉴욕 주립대학교(SUNY-Binghamton)전산학 박사(Ph.D)
1985년:오번대학교(Auburn Univ, Auburn AL) 전산과 조교수
1988년~현재:서원대학교 전자계산학과 교수