

## Praat를 이용한 숫자음의 음향적 분석법\*

An acoustical analysis method of numeric sounds by Praat

양 병 곤\*\*

Byunggon Yang

### ABSTRACT

This paper presents a macro script to analyze numeric sounds by a speech analysis shareware, Praat, and analyzes those sounds produced by three students who were born and raised in Pusan. Recording was done in a quiet office. To make a meaningful comparison, dynamic time points in relation to the total duration of voicing segments were determined to measure acoustical values. Results showed that a strong correlation coefficient was found between the repetitive production of numeric sounds within and across the speakers. Very high coefficients among diphthongal numbers (0 and 6) which usually show wide formant variation were noticed. This supports that each speaker produced numbers quite coherently. Also, the frequency differences between the three subjects were within a perceptually similar range. To identify a speaker among others may require to find subtle individual differences within this range. Perceptual experiments by synthesized numeric sounds may lead to resolve the issue.

**Keywords : Praat, numbers, acoustical comparison, speaker identification**

### 1. 머리말

오늘날 개인용 컴퓨터에서 이용할 수 있는 다양한 음성 분석소프트웨어의 개발로 지난 수 년 동안 불가능했던 여러 가지 실험들이 가능하게 되었다. 하지만 대다수 프로그램들은 고정된 메뉴로 사용자의 불편을 조장하고, 음성 분석구간에서 일관된 분석위치를 지정하여 원하는 음향적 분석결과를 얻기가 매우 어려웠다. 따라서, 음성 분석을 실시할 때, 측정자의 오류가 반영되거나, 동일한 분석 작업을 힘들게 되풀이하는 경향이 많다. Praat는 Paul Boersma가 개발한 음성분석프로그램으로 서로 다른 종류의 컴퓨터 운영시스템에서도 작동할 뿐만 아니라, 사용자가 프로그램의 스크립트를 직접 작성함으로써 측정자의

\* 본 연구는 한국과학재단 특정기초연구(1999-2-302-016-5) 지원으로 수행되었음.

\*\* 동의대학교 영어영문학과

오류를 줄일 수 있고 되풀이되는 작업을 자동화시켜 보다 많은 음성자료를 신속하게 처리할 수 있다.

이 논문에서는 피치값이 나오는 유성음 구간을 일정한 비율로 나누어 음향적 파라미터를 구하는 스크립트를 직접 작성하고, 상대적으로 비교할 수 있는 값을 구하는 방법을 제시하고, 이어서 부산에서 태어나 계속 살아온 건강한 남학생 세 명이 세 번 발성한 숫자음 음성자료를 이 스크립트로 실제 분석하여 보기로 한다. 이 과정을 통해 어떤 음향적 특징이 화자마다 동일한지, 또 화자를 구별해주는 변수들은 어떤 것이 있는지 살펴봄으로써 앞으로 보다 많은 실험자료를 분석하여, 화자 확인에 필요한 기초자료를 찾는 데 도움을 주고자 한다.

## 2. 연구 방법

### 2.1 Praat의 구성

Praat를 실행하면 오브젝트 박스와 그림 창으로 나뉜다. 오브젝트에 분석하고자 하는 음성자료를 불러와서 이것을 분석기능을 선택해서 처리하게 한 뒤 그 결과를 그림 창에 나타내면 된다. 음성을 분석하는 메뉴는 여러 가지가 있지만 음원과 여과기 이론(Fant, 1960)에 따라 분류하여 보면 음원을 나타내는 피치분석과 여과기인 성도의 특징을 보이는 스펙트로그램 분석이 중요하다. 피치분석의 파라미터 지정 값에 분석창의 길이가 있는데 피치 값이 높은 여성의 목소리는 분석창을 5 ms정도로 좁혀 주는 것이 필요하고 남성의 목소리는 7 ms정도로 부여하는 것이 좋다. 정확히 분석창 구간보다 넓은 간격으로 되풀이될 때 측정값에 에러가 발생하겠지만, 일반적으로 전후의 값을 토대로 3개 이상의 후보 값을 구한 뒤에 처리하기 때문에 큰 문제는 없을 것이다. Praat와 다른 소프트웨어를 이용한 피치분석 방식에 대한 문제점과 실례는 양병곤(1998, 1999, 2000)을 참고하기 바란다.

스펙트로그램과 포먼트 분석은 성도의 여과 특성을 분명히 나타내준다. 스펙트로그램 분석 방식에서도 분석 구간을 어떻게 지정하는가에 따라 좁은 대역과 넓은 대역의 다양한 결과를 보여준다. 특히, 좁은 대역의 스펙트로그램을 택하면 주파수축의 해상도가 높아져서 배음구조를 쉽게 볼 수 있다. 이 배음 구조는 바로 피치의 변화를 나타내준다. 피치분석 알고리즘의 문제점을 벗어나려면 이런 좁은 대역을 사용하여 인쇄하는 것이 도움이 된다(양병곤, 1998). 넓은 대역의 스펙트로그램은 포먼트 구조를 쉽게 볼 수 있는데, Praat의 홈페이지에는 보다 정밀한 포먼트 값을 구하는 분석방식에 대한 각각의 문헌이 나타나 있기 때문에 독자적인 프로그래밍을 통해 구현할 분은 참고하길 바란다. 음성분석 과정에서 스펙트로그램의 윤곽이 매우 중요한 정보를 보여주기 때문에 항상 포먼트 분석 결과를 스펙트로그램 위에 나타내어 정확도를 확인할 필요가 있다.

### 2.2 분석 스크립트 구성

Praat의 스크립트 기능은 반복적인 작업을 자동화시키는 방법으로 다량의 자료를 처리할 때 매우 편리하다. 먼저 해당 숫자음의 피치 분석 스크립트를 다음과 같이 작성해보았다.

```

select Sound 'SoundName$'
length=Get duration
  To Pitch (ac)... 0.005 75 3 yes 0.03 0.45 0.01 0.35 0.14 500
  Draw... 0 0 0 500 no

```

이 스크립트는 해당 음성을 선택한 뒤 전체 지속시간을 구해 length 변수에 넣고, 0.005초의 간격으로 75 Hz 이상 500 Hz 이하의 피치 값이 될만한 데이터를 3개 뽑아 yes 다음에 나타나는 다섯 가지의 기준에 따라 걸러 정확히 측정한다. 이어서 분석된 결과를 그림 창에 나타내준다. 일반적으로 피치 값이 구해지는 구간은 성대의 떨림을 보이는 유성음 구간으로 판단해도 좋다. 따라서, 유성음 구간에서 포먼트와 진폭값 등 필요한 음향적 특성을 살펴보기 위해서는 피치 값의 분석 결과가 매우 중요하다. 앞으로 다룰 숫자음에 대한 음향적 분석을 자동화하기 위한 단계로 피치구간을 이용해보았다. 다음은 피치 값이 0 이상의 값을 가지는 지점을 찾아내는 스크립트이다.

```

procedure f0on
  select Pitch 'SoundName$'
  while 'f0'=0 and 'i'<'length'
    f0=Get value at time... 'i' Hertz Linear
    if 'f0'=undefined
      f0=0
    endif
    i='i'+0.005
  endwhile
  onsettime='i'
endproc

```

먼저, 피치값을 구한 파일을 선택한 뒤 전체 지속시간보다 작고 f0값이 0일 때는 0.005 초마다 이동하면서 f0값이 0인지 아닌지를 판단해 나가다가 0값보다 큰 지점의 시간 값을 시작점(onsettime)으로 지정한다. 피치값 가운데 무한 수나 정의되지 않는(undefined) 값은 0을 대입하여 처리한다. 이어서, 피치값이 다시 0으로 내려가는 지점을 유성음 구간의 끝으로 지정해야 하는데, 다음 스크립트는 이 과정을 나타낸다. 위의 f0 시작점을 찾는 스크립트에서 while의 조건을 'f0' > 0 and 'i' < 'length'로 지정하여 이 조건을 빠져 나온 지점이 유성음 구간의 끝점(offsettime)이 된다. 여기서 시작점과 끝점 사이의 값을 빼면 유성음 구간의 지속시간이 되고, 이를 4로 나누어 시작점과 끝점과 이들 사이에 세 개의 측정 지점을 구하면 비록 지속시간은 다르더라도 서로 비교할 수 있는 상대적인 위치 값이 된다. 보통 시작점에서 피치 값은 포착되더라도 스펙트럼이나 포먼트 분석 구간이 길 경우에는 정확한 포먼트값이 잡히지 않기 때문에, 다음과 같이 이 지점보다 0.01초 뒤의 시간 위치에서 전후 10 ms 구간의 평균값과 표준편차를 구하여 이들 측정값의 유효성을 판단할 필

요가 있다.

```

call avsdfo onsettime
      aaf0=Get value at time... aa Hertz Linear
      if 'aaf0'=undefined
        aaf0=0
      endif
      if 'aaf0' > 'f0high' or 'aaf0' < 'f0low'
        aaf0='avf0'
      endif
procedure avsdfo f0point
  f0on='f0point'-0.01
  f0off='f0point'+0.01
  avf0=Get mean... f0on f0off Hertz
  sdf0=Get standard deviation... f0on f0off Hertz
  f0high='avf0'+3*'sdf0'
  f0low='avf0'-3*'sdf0'
endproc

```

위의 스크립트에서 call avsdfo onsettime으로 이 점 앞뒤의 평균과 표준편차를 구한 뒤 평균값에 3표준편차 값을 더하거나 빼어서 구한 피치범위(f0low~f0high)를 찾는다. 실제 구한 피치 값이 이 범위 안에 들어있을 때는 이 값을 피치 값으로 지정하고 그렇지 않으면 이 지점 전후 20 ms 구간의 평균값을 대입한다. 일반적으로 피치 값의 변화는 청각적으로 매우 미세한 변화를 보이고 편차가 적기 때문에 이 범위 밖으로 나가는 경우에는 성대의 떨림이 이렇게 급작스럽게 변하지는 않을 것이라는 가정 하에 해당 20 ms 구간의 평균값을 취하게 한다. 이렇게 함으로써 에러를 어느 정도 제거하게 된다. To Intensity... 100 0.005는 음성의 크기를 나타내는 진폭을 구하는 방식으로 0.005초마다 100 Hz 이상의 음성의 특징을 포착한다. 주파수 값을 높이면 보다 더 세부적인 음성의 진폭변화를 볼 수 있다. 진폭 값도 피치 값과 마찬가지로 20 ms 구간내의 평균과 표준편차를 구하여 걸러주도록 한다.

이어서 성도의 공명 특성인 포먼트값을 구해야 하는데 분석 구간과 방식에 따라 포먼트값은 음절 초와 음절 말에서 변화가 심하다. 따라서, 이것을 다음과 같이 평균값과 표준편차를 이용해서 걸러주는 작업을 하는 것이 필요하다. 포먼트값의 변화는 앞서 다루었던 피치와 진폭의 변화보다는 약간 큰 편차를 보이기 때문에 분석할 시간점 앞뒤의 평균값과 2 표준편차를 기준으로 이보다 높거나 낮은 경우에는 음성이 갑자기 변하지는 않는다는 가정 하에 에러값으로 버리고 이 시간영역의 평균값을 취한다. 지면의 길이를 줄이기 위해 시작점(aa)의 F1 처리에 관한 스크립트만 제시한다.

```

select Formant 'SoundName$'
  call avsd_fm onsettime
    aafm1=Get value at time... 1 aa Hertz Linear
      if 'aafm1' > 'fmlhigh' or 'aafm1' < 'fmllow'
        aafm1='avfm1'
      endif
    fl=Get value at time... 1 onsettime Hertz Linear
      if 'fl' > 'flhigh' or 'fl' < 'fllow'
        fl='flave'
      endif
  procedure avsd_fm fmpoint
    fmon='fmpoint'-0.01
    fmoft='fmpoint'+0.01
    avfm1=Get mean... 1 fmon fmoft Hertz
    sdfm1=Get standard deviation... 1 fmon fmoft Hertz
    fmlhigh='avfm1'+2*'sdfm1'
    fmllow='avfm1'-2*'sdfm1'
  endproc
endproc

```

이렇게 각 지점의 음향적 분석값을 정보 창에 나타내고 동시에 스펙트로그램과 함께 나타낸 그림과 비교하여 해당 값의 유효성을 점검한 뒤 분석결과를 저장한다. 대체로 주위 잡음이 섞인 음성 자료에 대한 포먼트 값의 추적은 어려움이 많이 나오기 때문에 항상 스펙트로그램의 윤곽을 기준으로 분석 결과를 검증하는 것이 필요하다. 지금까지 작성한 숫자음 분석 스크립트는 [Http://www.dongueui.ac.kr/~bgyang/praat.script](http://www.dongueui.ac.kr/~bgyang/praat.script)에 게시되어 있다.

### 3. 숫자음 분석

숫자음은 부산 지역에서 태어나 초등학교 과정을 마치고 부산에 계속 거주한 대학생 세 명의 발음을 녹음했다. 첫 번째 화자 (s1)의 키는 170 cm, 나이는 26세이고, 두 번째 화자 (s2)는 168 cm, 24세이며, 세 번째 화자 (s3)는 180 cm, 25세이다. 이들 화자는 조용한 사무실에서 Intel Celeron 466 MHz 프로세서를 가진 컴퓨터에 11,025 Hz로 직접 입력했다. 마이크는 AKG를 사용했고 Shure Prologue 오디오 믹서를 통과하여 좌측 채널로 음성을 입력했다. 음성입력프로그램은 Csd1.5c를 사용했으며, 이 프로그램은 녹음하기 직전 무음 구간의 진폭 값 정보와 일반적으로 나타나는 음성구간의 진폭 값을 이용하여 자동으로 음성구간을 선택해서 파일로 저장해준다. 4연속 숫자음 데이터베이스로 수집된 음성자료 가운데 6970, 8352, 1425를 세 번씩 분명하게 발음한 자료를 분석했다.

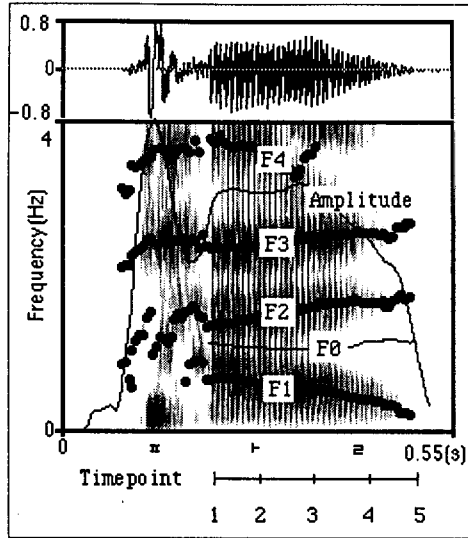


그림 1. s2가 발음한 숫자음 8의 파형과 스펙트로그램 : 스펙트로그램 위에는 여섯 가지 음향 분석결과를 보여준다.

이 세 가지 숫자음을 택한 이유는 0에서 9까지 모든 숫자가 들어 있기 때문이다. 먼저 각 숫자음을 열어서 각각의 숫자음에 해당하는 부분만 선택하여 저장했다. 분석은 앞 절에서 제시한 스크립트를 실행하여 스펙트로그램을 먼저 그리고 피치 곡선, 진폭 곡선, 포먼트 트랙을 더하여 그림 1과 같이 나타내었다. 정보 창에는 피치 값이 구해진 지점의 시작점과 끝점, 그리고 이 양 지점 사이에 일정한 간격으로 세 개의 분석지점을 선택하여 모두 다섯 개 지점의 음향적인 파라미터의 변화를 숫자로 포착해보았다. 이 분석구간은 각 숫자음의 음향 특성을 요약하는데 도움이 된다.

#### 4. 분석 결과 및 토론

##### 4.1 화자 내 변화 특성

먼저 s2가 안정되게 발성한 숫자음의 음향적 특징을 세부적으로 살펴보자. 그림 2는 숫자음 0의 s2가 세 차례 발성한 음성을 분석 스크립트로 처리한 결과를 보여준다.

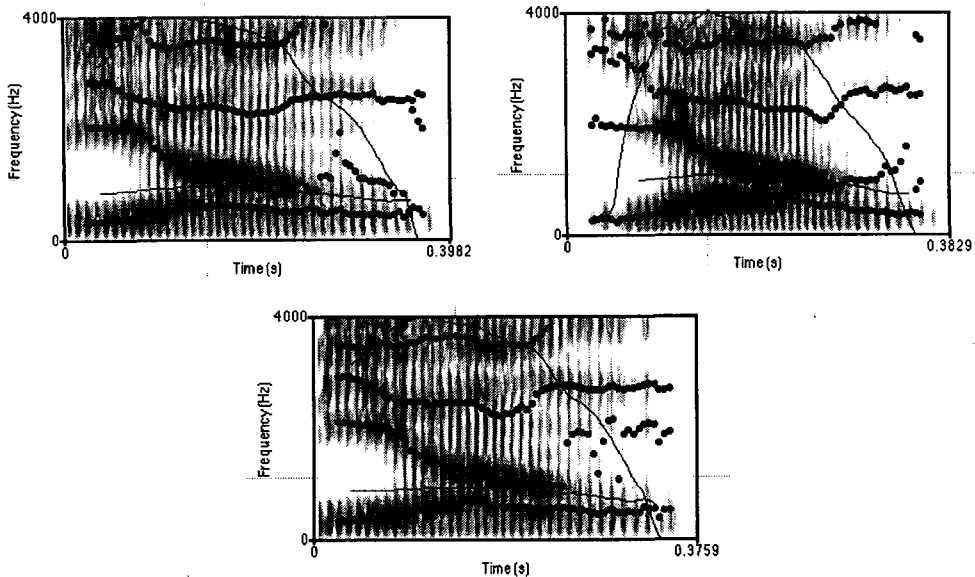


그림 2. 숫자음 0의 음향적 특징 분석: 동일한 화자가 세 번 발성한 것을 차례로 나타냈음.

위의 세 개의 스펙트로그램을 살펴보면 피치의 변화가 매우 비슷한 모양을 하고 있고, 진폭 변화도 중간에서 시작되어 1/3지점에서 최대 값을 가진 뒤 서서히 내려가는 모양을 하고 있다. 특히, 성도의 변화를 보이는 포먼트는 거의 비슷하다. 구체적으로, 턱의 변화를 보이는 F1은 동일한 구조를 보이고 있고 혀의 변화를 보이는 F2도 처음 1/3지점에서 동일한 곳에서 출발하여 서서히 중설모음 위치로 내려가다가 마지막 1/3지점에서 비음의 영향으로 불안정한 값들이 나타났다. 비음은 허파에서 생성된 기류가 연구개를 울리지 않는 상태에서 코로 새어나가도록 하는 모양을 하고 있기 때문에 일반적으로 F2 지점에 포먼트의 대역 값이 매우 높아지게 되고 F3에서 보다 좁은 대역 값을 갖게 된다(Pickett, 1987:77; Stevens, 1999:193). 따라서, 보다 넓은 대역 값을 가지는 F2 지점에서 포먼트의 진폭 값이 낮기 때문에 현재의 프로그램에서 포먼트 값을 구하는데 안정되지 않고 움직이는 에러가 많이 발생했다고 할 수 있다. 이렇게 F2의 편차가 많은 점을 이용하여 비음으로 분류할 수도 있겠다. 다음 표 1은 0이라고 발음했을 때 각 지점에서의 음향적 파라미터를 보여주고 있다.

표 1. 숫자음 0에 대한 다섯 지점의 음향적 파라미터값(세 번 발음을 각각 나타냄)

1회 270 ms				2회 320 ms				3회 300 ms			
F0	F1	F2	F3	F0	F1	F2	F3	F0	F1	F2	F3
124	350	1905	2533	106	335	1990	2714	108	349	2041	2765
134	577	1305	2392	116	575	1251	2368	114	589	1289	2429
130	597	1050	2239	112	543	1007	2274	110	553	980	2362
106	407	948	2535	94	439	1325	2562	93	439	1202	2669
88	353	1472	2616	85	429	587	2502	77	533	2034	2762

\* 횃수 다음의 값은 지속시간을 나타내며, 첫 번째 값은 F0, 두 번째부터 차례로 F1, F2, F3 값을 보여준다.

횃수 다음에 제시된 지속시간도 거의 비슷하게 나타났고 각 시간 점에 해당하는 음향 측정값들의 상관계수를 StatView로 구해 표 2에 나타내보았다.

표 2. s2 화자가 세 번 발성한 숫자음 0의 음향 측정값 사이의 상관계수

	1회	2회	3회
1회	1		
2회	0.977	1	
3회	0.993	0.958	1

각 횃수별로 매우 높은 상관계수를 보이고 있음을 알 수 있다. 이러한 계수는 화자가 발성할 때 매우 일정하게 발음했음을 나타낸다. 숫자음 0은 이중모음이 있기 때문에 화자 내에서도 포먼트 변화율이 많고, 사람마다 다르게 나타날 것으로 예상할 수 있는데 현재로서는 매우 일정한 패턴으로 발음했다. 이러한 경향은 유성음 지속시간을 기준으로 동적인 측정값을 구했기 때문이기도 하다. 이중모음으로 된 숫자음 6에서도 매우 비슷한 스펙트로그램과 음향적 파라미터의 변화를 보여주었다. 첫 번째 발음에서 피치값이 198 Hz로 나타나는 에러가 나타났음에도 불구하고, 전체적으로는 매우 높은 상관도( $r=0.968$  이상)를 보여서 스펙트로그램의 일치성을 뒷받침하고 있다. 다시 말해서, 화자는 아주 동일한 조음 동작으로 발음했음을 알 수 있다. F1, F3의 모양은 거의 완전한 일치를 보이고 있고 F2에서도 중간지점을 지나서 약간 위로 튀어나온 독특한 모양을 하고 있었다. 이러한 개인적인 특성을 파악하기 위해서는 이중모음이 포함된 숫자음의 분석은 다섯 개의 음향적 분석값으로는 부족할 것으로 예상된다. 하지만 보다 많은 지점의 분석값을 구하는 것은 그만큼 시간이 걸리게 되고, 이것은 오히려 합성 실험을 통해 몇 개의 분석값이 개인성을 나타내는데 충분한지 연구하여 해결해야 할 것이다.

단모음으로만 구성된 2와 5는 상관계수가 모두 0.996 이상이 되어 이중모음보다도 더 일치된 결과를 보여 주었다. 지속시간도 대체로 일치하고 있으며, 음향적 분석값도 거의 동일하다. 따라서, 이중모음이 들어간 0과 6과는 달리 이러한 변화가 적은 숫자음을 분석할 때는 다섯 개 지점의 음향적 파라미터를 일일이 분석할 필요가 없이 세 개 정도의 지



점만으로도 쉽게 음성의 특징을 포착할 수 있을 것으로 여겨진다. s1과 s3의 세 번의 발성자료간의 상관계수의 분포를 보면 최저 값이 0.927이고 두 사람의 모든 상관계수 값을 평균했을 때 0.988로 매우 높다. 한가지 지적할 것은 s3의 숫자음 5에서 최저 값이 나타났는데 이는 모음 [오]의 F1과 F2가 서로 접근하여 중간지점의 일부 값이 한 포먼트로 처리되었기 때문으로 여겨진다. 이러한 프로그램의 문제점은 해당 숫자음을 인식하여 예상되는 F1과 F2 포먼트 값의 위치 안에 두 개의 포먼트를 강제로 추출하도록 하여 보완할 수 있을 것이다.

#### 4.2 화자간 변화특성

그러면 이렇게 동일한 발음을 하는 세 사람 사이의 개인적인 특성 차이는 어디에서 오는 것일까? 먼저, 세 화자의 음성자료 가운데 변화가 많은 0을 비교해보자. 서로 비교할 수 있는 분석지점의 값들을 한꺼번에 비교하기보다는 주된 경향을 파악하기 위해 세 번의 발음의 평균을 낸 뒤 화자끼리 서로 빼어보았다. 그림 3은 숫자음 0을 발음한 세 화자(s1, s2, s3)의 F1, F2, F3 값을 보여준다. 이 그림에서 세 화자의 각 포먼트 값을 다섯 개 지점에서 나타내 보았다. 진폭에서는 절대값의 차이가 2 dB이었고, 피치 값에서는 8 Hz, F1에서 약 84 Hz, F2에서 186 Hz, F3에서 121 Hz의 차이를 보였다. 진폭 값의 차이는 매우 적었고, 피치 값의 차이도 최소 1 Hz에서 최대 20 Hz의 범위에 있었다. F1의 최소 차이는 4 Hz이고 최대차이는 202 Hz로 나타났으며, F2는 최소 25 Hz에서 최대 464 Hz가 나타났는데 이 큰 차이는 비음구간인 4 지점에서 나타났다. F3의 차이의 최소값은 4 Hz에서 최대 325 Hz를 보였다. 이러한 값의 변화는 양병곤(1995, 1996)의 단모음과 이중모음의 연구 결과에서 나타났던 지각 범위 안에 들어간다. 따라서, 청자에게는 동일한 발음으로 들리게 된다. 화자 간의 차이를 조사하기 위해서는 이 주파수 범위 안에서도 더 세분하여 관찰해야할 것이다. F1의 시작 부에서는 화자의 차이가 별로 없으나 중간지점에서 약간의 차이를 보이고 F2의 네 번째 지점에서 비음의 특징 때문에 발생한 포먼트 값의 차이를 보였다. F3의 두 번째 지점과 네 번째 지점에서 약간의 차이를 보였다. 따라서, 화자 간의 차이를 구하기 위해서는 각 포먼트에서의 미세한 차이를 확대하여 보는 방법이 필요할 것으로 여겨진다.

안정된 단모음으로 구성된 숫자음 2에서는 세 화자 사이의 음향적 차이를 구해본 결과, s1과 s3의 진폭 값의 차이는 1, 2지점에서 동일하고, 다른 화자와는 1에서 8 dB 차이를 보여 모두 비슷한 크기로 발성함을 알 수 있다. 피치 값의 차이는 절대값으로는 12 Hz로 매우 적고 최소 1 Hz에서 최대 24 Hz까지 차이가 있었다. 포먼트 값에서는 절대값의 평균이 F1에서 25 Hz이고, F2는 약 189 Hz, F3는 156 Hz였다. 최소값은 1 Hz의 차이까지도 보이며 최대값은 510 Hz까지 나타났다.

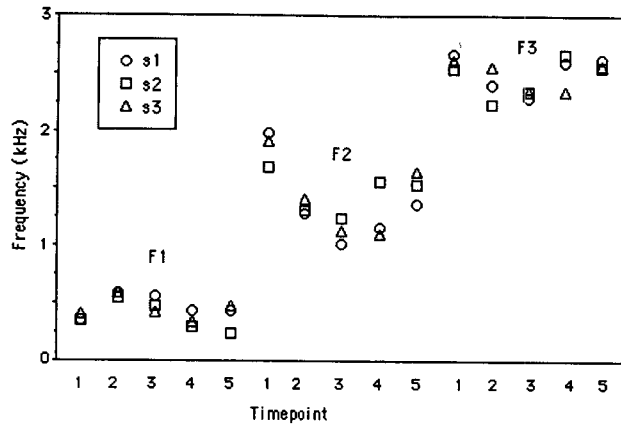


그림 3. 세 화자가 발성한 숫자음 0의 포먼트 변이그림  
: 가로축의 숫자는 측정지점을 나타낸다.

이러한 주파수 범위는 청각적으로 동일한 음성 범위에 들어가며, 동시에 개별적인 차이를 나타낸다. 따라서, 앞으로 화자 구별을 목표로 하는 장치를 개발하기 위해서는 이들 각각의 포먼트값의 차이를 극대화하고, 또한 컴퓨터 프로그램의 오류에서 생기는 포먼트값의 차이를 피할 수 있는 안정된 구간의 값, 특히, 유성음 구간의 값들을 비교하는 것이 필요할 것으로 여겨진다. 예를 들어, 숫자음 0, 3과 같은 경우에는 비음성에 의한 변화가 나타나는 4, 5의 시간점에서 F2값을 이용하여 컴퓨터로 학습시키게 되면, 동일한 화자라도 여러 번 발성할 때마다 값이 달라지게 되어 매우 변화가 심한 포먼트값이 모델로 설정이 될 것이다. 이 모델을 이용하여 동일 화자임을 확인시키는 것은 어려울 것으로 여겨진다. 만약 숫자음을 인식하는 모듈을 넣게 된다면, 숫자음마다 다른 주파수 범위의 포먼트값을 구하도록 사전 처리를 하게 할 수 있다. 예를 들어, 비음이 포함된 경우에는 F1과 F3만을 구하여 모델을 설정하는 방법도 가능할 것이다. 덧붙여, jitter나, 진폭값의 변화율인 shimmer와 같은 성대의 측정방식을 응용하여, 이중모음에서 각 포먼트값과 피치값의 단위시간 내의 변화율을 구해 화자들의 특색을 찾아가는 것도 한 가지 방법이 될 것이다.

## 5. 맺음말

지금까지 Praat를 이용하여 숫자음을 분석하기 위한 스크립트를 작성하였고, 이것을 이용하여 부산출신의 대학생 세 명이 각 숫자음을 3회 발음한 숫자음 발음을 분석하여 보았다. 자동화 스크립트를 사용함으로써 체계적으로 음성을 분석할 수 있음을 보였고, 실제 적용하는 과정에서 보다 더 정밀한 간격으로 분석해야 할 음성신호와 더 이상 세분할 필요가 없는 음성신호 등으로 나누어 볼 수 있었다. 적은 자료이지만 숫자음을 분석한 결과로는, 세 번의 발성에서 숫자음 0과 6의 경우에는 이중모음이라는 변화가 많은 특징에도 불구하고 전체지속시간으로 보아서 대응하는 위치에 해당하는 값들을 비교한 결과 상관계수

가 매우 높게 나타났다. 특히, 비음이 포함되어 있는 0에서는 제 2포먼트 근처에서 안정되지 않는 값들이 관찰되었는데, 대역 값이 넓기 때문에 생겨난 현상으로 추정된다. 단모음으로 구성된 2와 5에서도 이중모음보다 더 안정된 값들을 보이고 있음을 확인할 수 있었다. 이러한 결과는 화자가 매우 일관성 있게 발음을 한다는 점을 입증해준다. 화자간 변이의 폭은 청각적으로 동일하게 지각하는 범위 안에서 발음되고 있으며, 이 범위 안에서 개인간의 차이가 나타남을 보았다. 특히, 유성음 구간을 이용하여 상대적인 분석지점을 사용함으로써 상호 비교가 가능함을 보았다. 앞으로 이 방법을 이용하여 보다 많은 화자들의 발성을 분석하여보고, 아울러 서로 다른 화자가 발성한 숫자음의 음향적 특징을 분석하여 재합성한 음성파일을 청각실험을 통해 어느 정도의 변화가 화자의 특성을 구별해 주는지 연구해볼 계획이다.

## 참 고 문 헌

- [1] 구희산, 고도홍, 양 병곤, 김기호, 안상철. 1998. 음성학과 음운론. 서울: 한신문화사.
- [2] 양병곤. 1995. "합성한 한국어 단모음의 지각실험 연구." *언어* 20(3), pp. 127-146.
- [3] 양병곤. 1995. "합성한 한국어 이중모음의 지각실험 연구." *언어* 21(3), pp. 829-843.
- [4] 양병곤. 1998. "Signalize에 의한 피치 분석방법 고찰." *동의논집* 28, pp. 69-80.
- [5] 양병곤. 1999. "WinCECIL에 의한 억양 분석." *동의논집* 30, pp. 51-62.
- [6] 양병곤. 2000. "Praat에 의한 억양 분석 방법." *동의논집* 32, pp. 29-39.
- [7] Fant, G. 1960. *Acoustic Theory of Speech Production*. The Hague, Netherlands: Mouton.
- [8] Pickett, J.M. 1987. *The Sounds of Speech Communication: A Primer of Acoustic Phonetics and Speech Perception*. Austin, TX: Pro-Ed.
- [9] Stevens, K.N. 1998. *Acoustic Phonetics*. Cambridge, MA: MIT Press.

접수일자: 2000. 5. 05.

게재결정: 2000. 6. 04.

### ▲ 양 병 곤

부산시 진구 가야동 산 24

동의대학교 인문대학 영어영문학과(우: 614-714)

Tel: (051) 890-1227

e-mail: bgyang@hyomin.dongueui.ac.kr