

한국어 정보원의 구조분석과 Code의 개선 (Some Structural Analysis of HAN GEUL Information Source and its Application to the Improved Coding Method)

李柱根*, 朴鍾旭*, 金昌善*

(Lee, Joo K., Park, Jong W., and Kim, Chang S.)

要 約

이 논문은 한국어 정보원의 구조를 분석하고 기본문자의 출현확율에 따른 KS몰스코오드를 재구성하여 전송속도 및 효율을 14% 이상 향상시키고 또 앞서 일부 제안된 바 있는 Δ parameter의 효과와 그것에 의한 한글의 단음절의 자동 식별방법 및 코오드의 개선 및 data entry의 합리화에 미치는 영향에 대해서 기술하였다. 아울러 단음절의 자동식별은 13~16%의 효율을 향상시킨다는 것도 명백히 하였다.

Abstract

In this paper, a preliminary structural analysis of the information source of HANGEUL is performed statistically.

And the results of the analysis are utilized to a KS HANGEUL code, improving its transmission rate of 14%, on the basis of probability rankings of the fundamental HANGEUL elements.

Furthermore Some problems about KS code, a coding method for the HANGEUL information processing, as well as data entry are realized.

1. 서 론

정보의 매체로서 code를 도입한 것은 오래 전 부터이며, 현재의 정보이론 이전의 몰스코오드에 까지 거슬러 올라간다.

컴퓨터의 발달과 더불어 code는 새로운 관점에서 검토되었고 각종 이론을 배경으로 한 많은 논문이 발표되었으며, 국제표준의 ISOR646-1967를 제정되어 각국은 이것을 준수하고 있다.

그러나 이것은 서구어를 기준으로 한 것이고, 7bit code는 꼭 차 있다. 따라서 한글의 code 화에는 여러 가지의 문제가 제기된다. 단말의 hardware의 구성을 고려해야 하고, 국제표준 code와 data entry 및 모든 주변장치에 이르기 까지 연동될 수 있는 간결하고, 최선의 합리화가 이루어질 수 있는 것이어야 한다.

특히 한글의 구조적인 특징과 keyboard의 통일문제

를 고려에 넣는 것이 대단히 중요하다. 이상의 제문제들은 상호 연관되므로 어느 하나가 결여되어도 장차 관련된 system의 개발에 고질적인 저해요소가 된다. 그러한 의미에서 현행 10여종의 한글 code와 data entry를 분석한 결과 많은 문제를 내포하고 있다.

code에 있어서 표준으로된 Morse code와 TTY code는 기본문자만 규정해 있고, 정보처리용 표준 7bit, 8bit code는 기본문자를 51자로 규정하고 있다. 기본문자를 두가지로 규정하고 있기 때문에 code가 서로 다르고 data entry가 통일이 되지 않는다. 정보처리용으로 불필요하게 51자를 규정했기 때문에 국제표준 data entry에 병용할때 지극 복잡해 진다.

최근에 한글의 처리에 관련된 연구가 각 방면에서 비상한 관심을 모으고 있는데 하지만 가장 중요하고도 기본적인 문제에 대해서는 스쳐가는 경향이 없지 않다. 따라서, 본 논문에서는 한글의 구조적인 성질을 규명하고, 한글의 입력방식과 단음절의 자동식별 code 및 data entry의 개선으로서 효율을 14% 이상 향상시키고 또 종래에는 명확히 제시해 있지 않은 단음절의 자동식별로 인한 효과는 최소 13%가 된다는 것을 명백히 하였다.

* 正會員, 仁荷大學校 電子科

**準會員, 崇田大學校 電子科

(Dept. of Electronics Eng.,
Inha University and
Sungjun University)

接受日字: 1978年 1月 25日

2. 한국어 정보원의 구조와 제문제

일반적으로 정보원은 언어의 구조에 따라 그의 성질이 달라진다. 그러나 한국어에서는 언어의 구조 뿐만 아니라 한글의 특이한 구조에 대한 영향이 강하게 작용한다. 따라서 code는 정보원의 기본요소의 규정에 직접 관련되고 또 기본요소는 data entry와 system설계에 관련된다. 한글의 특이한 구조는 기본요소의 규정에 따라 여러가지의 형태로 나타난다. 때문에 최량의 code의 합리화를 위해서는 우선 한국어 정보원의 구조를 명확히 파악하는 것이 중요하다. 이에 관련된 연구는 일부 발표한바 있다⁽⁴⁾. 여기서는 그것을 확인하고 code개선의 평가를 위한 추가적인 자료의 분석을 첨가한다.

언어의 정보원은 확율과정으로 규명되고 있으므로 충분한 통계자료와 자료가 편중되지 말아야 한다. 따라서 35종의 서적을 선택하고, 현대어 중에서 152726를 대상으로 하여 단음절 space, 단어 space를 고려에 넣은 자료를 computer로서 분석하고 입력 방식의 평가를 위한 자료로 삼았다.

분석방법은 한글의 구조적인 특징을 감안하고 모든 가능한 입력방식에 대한 평가자료를 얻기 위해서 6종류로 분류하고 그들 각각의 경우를 선택할 때의 합리성 여부를 평가한다. 이들 6종의 분류는 한글에서 가능한 입력방법의 하나에 귀착된다. 즉,

- (1) 기본문자 24 + $W_s = 25$ 자의 기준형
- (2) 기본문자 24 + $m_s + W_s$ 의 기준형
- (3) 기본문자 24 + $l_s + m_s + W_s$ 의 기준형
- (4) 기본문자 24 + $CC + VV + W_s = 34$ 자의 기준형
- (5) 기본문자 24 + $CC + VV + VVV + W_s = 52$ 자의 기준형
- (6) 받침등을 별도로 규정한 68자의 기준형

단 m_s : 단음절 space, W_s : word space, l_s : element space, CC : 쌍자음, VV : 중모음, VVV : 복합모음, 어느 입력방식이 정보처리의 단말로서 가장 효과적인가는 이들의 각각에 대한 구조분석이 필요하다.

(1)항은 국제표준 code 및 data entry에 한글을 병용할 때 가장 합리적인 형태가 되며, 또 한글의 본질적인 문제에도 관련된다. 그것은 Alphabet 26자 이내에 들어갈수 있기 때문에 여타방식에서의 모든 문제점을 해소할 수 있다. 그러나 이경우 단음절의 자동식별의 문제가 대단히 중요하다.

(2)항은 현용 TTY등에서 정보와는 관계없는 단음절 space를 문자마다 삽입하는 입력방식에 해당되는 것으로서 속도의 저하를 가져온다. 이것은 단음절식별

의 난점에 부딪혀 있기 때문이다.

(3)항은 Morse code의 경우로서 단음절 space의 element space가 상당히 차지한다. (4)항은 쌍자음과 중자음을 따로 규정한 33문자의 경우이다. 이때 문자의 합성은 용이하여 어떤 의미에서는 유효한 면도 있으나, 정보처리용 국제표준 code 및 data entry에 한글을 배당할 때 Alphabet 26자를 벗어나서 5개쌍자음 배당이 만족스럽지 못하다. 문자배당이 3단이상으로 되어 shift조작이 빈번해 지는 경향이 있다.

(5)항은 정보처리용 한국표준(KSC)으로 제정된 7bit, 8bit code에 해당되는 것으로서, 이것은 재래식 Typewriter의 구조에 가깝고, 52자를 기준으로 했기 때문에 code의 불요부분이 50%를 차지하고, data entry가 복잡해지며, shift조작이 빈번해지며 능률이 전하된다.

표 1. Frequency & Probability of Hangeul Elements.

Element	Frequency	Probability (P_i)	Rank	
1	ㄱ	29764	0.0650	6
2	ㄴ	32670	0.0714	5
3	ㄷ	15505	0.0338	12
4	ㄹ	24202	0.0529	8
5	ㅁ	11166	0.0244	16
6	ㅂ	9724	0.0212	17
7	ㅅ	23247	0.0508	10
8	ㅇ	48839	0.1067	2
9	ㅈ	12412	0.0271	14
10	ㅊ	2881	0.0062	19
11	ㅋ	769	0.0016	25
12	ㅌ	2144	0.0046	20
13	ㅍ	1888	0.0041	21
14	ㅎ	12929	0.0282	13
15	ㅊ	228140	0.4986	
15	ㅏ	42522	0.0929	4
16	ㅑ	1035	0.0022	24
17	ㅓ	23483	0.0513	9
18	ㅕ	7531	0.0164	18
19	ㅗ	19711	0.0430	11
20	ㅛ	1363	0.0029	22
21	ㅜ	11225	0.0245	15
22	ㅠ	1095	0.0023	23
23	ㅡ	25385	0.0554	7
24	ㅣ	44800	0.0979	3
		178150	0.3894	
25	S	51182	0.1118	1
TT		457472	0.9999	

최하의 Keyboard배열이 된다. (6)항은 재래식 Typewriter를 기준으로 한 것이며, Typewriter에서 문자 타이핑은 용이해지는 점도 있지만 data entry로서는 부적당하다.

국제표준 data entry를 감안할때 가장 바람직한 것으로 생각되는 것은 영문자 26자 이내에 들고 또 자음이 15개 이내의 것이 합리적 entry가 된다.

이점에서 (1)항의 입력방식이 유효하다는 지시적인 판단이 내려진다.

또 한글 keyboard는 자음과 모음을 파우의 어느쪽에 배당하는가 하는것도 중요한 문제의 하나이다. 이점 명확한 data없이 많은 논란이 있으나, 표 1의 분석 결과로부터 (1)식에 의하여 명백해졌다. 이것은 기본적으로 기본문자의 입력방식의 경우에 해당된다.

지금 자음C, 모음을 V, word space S의 확률빈도를 각각 $P(C)$, $P(V)$, $P(S)$ 로 할때

$$\begin{aligned}
 P(C) &= \sum_{i=1}^{14} P_i(C_i) = 0.4986 \\
 P(V) &= \sum_{i=1}^{10} P_i(V_i) = 0.3894 \\
 P(S) &= \sum_i P(S_i) = 0.1118
 \end{aligned}
 \tag{1}$$

이 된다. 따라서 keyboard의 배열을

$$P(C) < P(V) + P(S) \tag{2}$$

space조작을 (2)식에 기준을 두면 파우가 평형을 이룬다. 이것은 입력의 속도에도 관련되며, 왼쪽에 자음을 배당하는 것이 인간공학적으로나, 효율상으로 합리적이라 보겠다. 또 (.) 등 기호도 ISO에 따라 오른쪽에 배당되므로 오른쪽 부담이 약간 증가되어 인간공학적으로 합리적이다.

다음 3성음의 분류에 대한 확율은 그림 1과 같은 tree-gram로 표시되며, 입력형식 (6)항에 대한 자료가 된다. 그러나 이것은 code 및 data entry에서는 별로

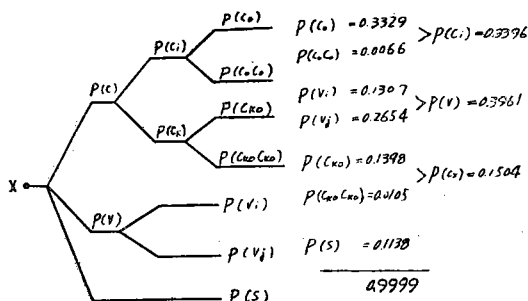


그림 1. 3성음의 확율
Fig.1. Tree-gram of monosyllable elements.

흥미없지만 일반 문장에서 자료가 될 수 있다.

또 표 1에서 word space의 frequency 51182, 단음절 space Z의 frequency는 101544이므로 표 1의 확률 분포는 다음식과 같이 계산된다.

$$\begin{aligned}
 P(C) &= 0.4081 \\
 P(V) &= 0.3186 \\
 P(Z) &= 0.1816 \\
 P(S) &= 0.0915
 \end{aligned}
 \tag{3}$$

이것은 단음절 space를 삽입하는 형용 SP 방식의 TTY에서 정보와는 무관한 단음절 space의 출현확율이 18.2%를 차지한다는 것을 명시한다. 따라서 단음절 자동식별의 NSP방식은 frequency가 최소 18.2% 개선된다는 것을 의미한다.

다음 받침이 있는 문자와 없는 문자를 각각 $P(C_k)$, $P(\phi)$ 라던 계산결과 다음과 같다.

$$\begin{aligned}
 P(C_k) &= 0.3317 \\
 P(\phi) &= 0.4986 \\
 P(S) &= 0.2510
 \end{aligned}
 \tag{4}$$

또 word space W를 제외한 기본문자의 총 frequency는 표 1에서 406290이고 통계자료의 문자수는 152726자로 부터 한글의 한 문자당 평균 기본문자의 수 β 는

$$\langle \beta \rangle = \frac{406290}{152726} = 2.66 \text{ [Letters]} \tag{5}$$

이 되어 이것은 논문(4)에서와 일치된다.

이상의 분석자료로부터 code평가의 척도가 되는 transmission rate 및 효율을 구할 수 있다.

code의 평균길이를 τ 라 할때

$$\begin{aligned}
 \tau &= \sum_{i=1}^q P_i(x_i) \tau_i \\
 \sum_i P_i(x) &= 1
 \end{aligned}
 \tag{6}$$

정보원의 평균 정보량은

$$H = - \sum_{i=1}^q P_i(x) \log P_i(x) \tag{7}$$

이고, transmission rate 혹은 효율 R은

$$R = \frac{H}{\tau} \tag{8}$$

$$\text{Redundancy } \Gamma = 1 - \frac{H}{\tau}$$

로서 구할 수 있다.

3. 한글의 몰스 code에 대한 평가

한글의 몰스 code는 체신부에서 제정되어 KS표준으로 되었고 체신, 내무, 외무, 국방 등 각 기관에서 해방직 후부터 사용해 왔으며, 60년대 초부터는 TTY

가 도입됨에 따라 점차 그 자리가 옮겨져 가고 있지만 아직 해양통신등 많이 사용되고 있다. 그러나 이 code의 제정후 평가자료가 제시해 있지 않고 또 그것은 국가의 전 신경에 영향을 준다는 점과 막대한 경제적 부담을 고려할 때, 그의 평가에 대한 고찰은 중요한 의미가 있다고 보겠다.

현대의 정보이론에서 언어의 생성은 확률과정으로 설명되고 생성확률 $P_i(x)$ 가 가장 큰 순위를

$$P_1(x) > P_2(x) > \dots > P_q(x) \quad (10)$$

로 할때 $P_i(x)$ 의 가장 큰 순위로부터 가장 짧은 code를 배당하는 것이 효과적이라 인정되고 있다⁽¹⁾. 또 가장 변장 code는 Huffman의 방법에 의해서도 구성된다.

Morse code는 Dot, Dash, space의 조합으로서 구성되며, 단위길이는 다음 도해표시로서 계산할 수 있다.

표 2의 도식표시와 표 1의 확율을 기준으로 code를 구성하면 표 3과 같다.

표 3에 기본요소의 확률분포 $P(x)$ 와 code 길이 τ_i 의 계산치를 보인것이며, 이들 관계를 그림 3에 보이었다 그림 2를 관측해 보면 현용 morse code K_i 는 확률분포 $P(x)$ 와 아무런 관련성이 없고, 합리적인 code구성이 이루어져 있지 않기 때문에 code의 평균 길이가 길

표 2. 몰스 code의 도식 표시
Table 2. The length of Morse code bits.

Dot	+-	2	code의 길이 τ_i
Dash	+++-	4	
Letter space	---	3	
Word space	-----	5	

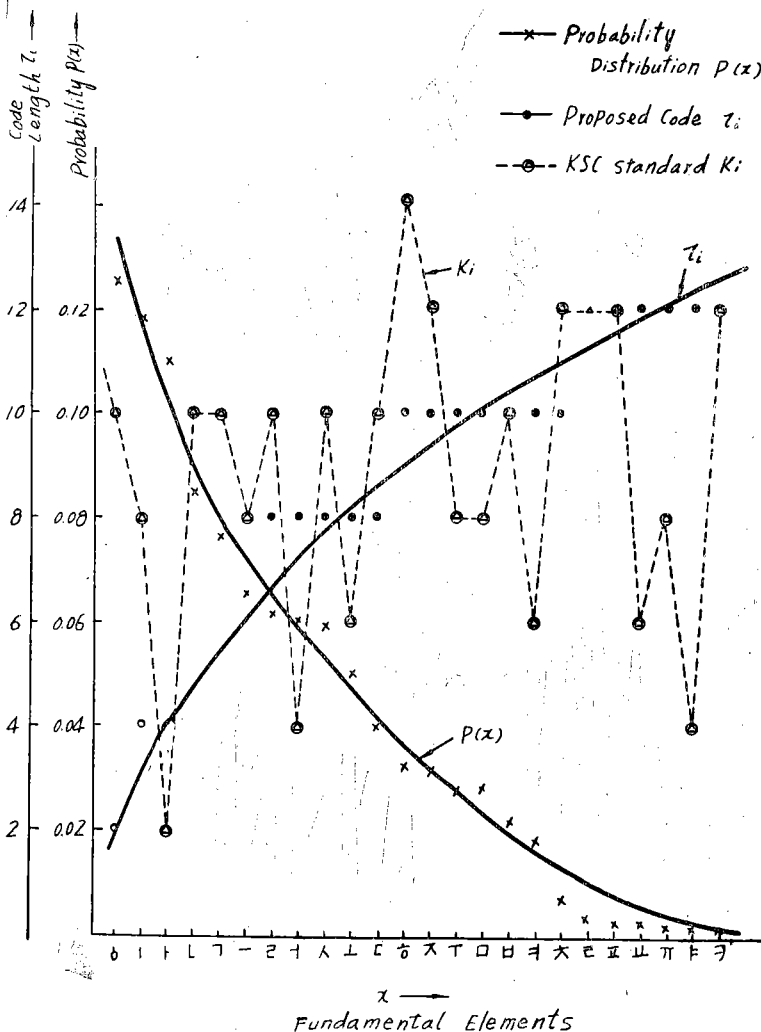


그림 2. 확률함수 $P(x)$ 와 code의 길이 τ_i 와의 관계
Fig.2. Relation ships between probability function $P(x)$ and code length τ_i ;

표 3. Morse code의 구성

Elements	Morse Code	τ_i	Frequency	Probability $P_i(x)$
ㅇ	-	2	48839	0.12020
ㅣ	---	4	44800	0.11026
ㅏ	---	4	42522	0.10465
ㅑ	---	6	32670	0.08041
ㅓ	---	6	29764	0.07325
ㅕ	---	6	25385	0.06248
ㅗ	---	8	24202	0.05956
ㅛ	---	8	23483	0.05779
ㅜ	---	8	23247	0.05721
ㅠ	---	8	19711	0.04851
ㅡ	---	8	15505	0.03816
ㅎ	---	10	12929	0.03182
ㅈ	---	10	12412	0.03054
ㅊ	---	10	11225	0.02762
ㅋ	---	10	11166	0.02748
ㆁ	---	10	9724	0.02393
ㄷ	---	10	7531	0.01853
ㄱ	---	10	2881	0.00709
ㄴ	---	12	2144	0.00527
ㅇ	---	12	1888	0.00464
ㅈ	---	12	1363	0.00335
ㅊ	---	12	1095	0.00269
ㅋ	---	12	1035	0.00254
ㆁ	---	12	769	0.00189
계			406290	$\tau = \sum_{i=1}^{14} P_i(x)$
				$\tau_i = 6.40090$

어지며, 동시에 통신속도가 떨어진다. 이에 반해서 제안된 code는 확률분포 함수 $P(x)$ 와 대칭으로 이루어지며 그 교점이 평균길이 τ 가 된다. 이것은 가장 사용 frequency가 높은 문자가 가장 짧은 code가 배당되어 통신속도가 개선된다. 즉 기존 code는 사용 frequency가 가장 높은 "ㅇ"를 code 길이 10을 배당한 데 비해서 제안된 code는 2인 것이 그 한 예이다. 다음 표 3으로부터 (5)~(7)식에 의하여 개선된 효율을 계산하면

표 4. 종래의 방법과의 비교

Table 4. Comparison with conventional method.

종래의 한글 Morse Code		제안된 한글 Morse Code
H	3.976	3.976
τ	8.281	6.401
R	0.480	0.621
Γ	0.520	0.379
개선된 효율		14.1%

표 4와 같다. 기존 code의 평균 길이 8.28인데 비해서 제안된 code의 평균 길이는 6.40이므로 개선된 효율은 14% 이상이 되고, redundancy Γ 는 52%에서 37.9%로 개선된다.

또 논문 (4)에서의 data로 부터 계산하면 효율은 16%가 개선된다. 따라서 효율의 개선도는 평균적으로 14~16%의 범위에 이른다고 보겠다.

4. Δ parameter의 설정과 효과

Morse code는 가변장 code인데 비해서 컴퓨터용 code는 Hard ware에 중점을 두었기 때문에 일반적으로 균일장 code를 사용한다. 정보처리용 한글의 code화에는 2절에서 언급한 바와 같이 여섯가지의 종류가 가능하다. 그러나 한글의 code화에는 이미 표준화된 ISO code를 염두에 두어야 하고, 또 관련된 data entry에서 합리적으로 이루어져야 하며, 국제표준의 각종 주변장치를 고려에 넣어야 한다. 나아가서는 한글의 특이한 구조에 대한 system의 설계를 고려에 넣고, 또 그것이 TTY에 이르기 까지 각종 keyboard에서 통일된 입력방식을 이루는 것이 중요하다.

그러나 한글에서는 이들 문제가 상반된 성격을 가지고 있어서 간단치 않다. 간단히 문자의 품질을 향상시키기 위해서는 기본입력수를 증가시키는 것이 합성이 용이 해지고, 입력수를 증가시키면 code 및 국제표준 data entry가 복잡해지며 불합리해진다. 24개 기본문자의 입력방식이 가장 바람직한 것이지만 단음절의 자동식별에 대한 난점이 있다.

현재 KS표준으로된 7bit, 8bit code는 기본문자를 51자로 규정하고 있다. 이 code의 문제점은 불필요하게 기본문자를 51자로 규정한 점이다. 때문에 국제표준 data entry에서 복잡해지며, 빈번한 shift조작의 고통을 자초하고, code의 불요부분이 50%를 차지하며, 한글의 Keyboard의 통일에 대한 저해요소로 작용한다.

또 TTY code 및 자판은 5bit code의 배당상 기본 24자와(ㄱ, ㅋ)만 규정하고 쌍자음은 규정해 있지 않다. 때문에 단음절 분리가 되지 않기 때문에 현용 TTY에서는 문자마다 space를 넣는다. 따라서 필연적으로 속도 저하를 가져온다.

그러므로 최근의 각종 data entry에서는 5개 쌍자음을 임의로 배당하고 있어 code 및 keyboard가 통일되지 않고, 또 서로 연동이 되지 않는다. 이들 제문제를 일거에 해결해 주는 방법으로서 저자의 일인은 5개 쌍자음의 후자음을 공동으로 사용하는 parameter를 정의한 바 있다.^(5,6,7,8) D부호는 Double, Dummy

의 두 머릿자를 따서서 D에 횡선을 그은 것이다. 이 D는 pattern 인식에서는 alphabet D와 동시판정이 내려질 염려가 있기 때문에 그 후 "△" parameter로 고쳤다. 이것은 한글 옛자를 선택한 것으로서 역사적 의미를 가지고 또 ASC II에 포함해 있지 않으므로 적당하다고 생각된다. 입력방식은

정의 1: △parameter는 ㄱ△, ㄷ△, ㄴ△, ㄱ△, ㅅ△(혹은 △ㄱ, △ㄷ, △ㄴ, △ㅅ, △ㅈ: 직접입력에 유효)의 입력으로부터 (ㄱ, ㄷ, ㄴ, ㅅ, ㅈ)의 5개 쌍자음을 표현 또는 생성된다.

정의 2: 자음 C, 모음 V의 연속입력으로 부터 최초의 모음 V의 앞 C앞이 한음절이고 또 최초의 V앞 △앞C앞을 한 음절로 처리한다.

↑CVC↑CVV↑C△VC↑CVC△↑CVCC

따라서 연속입력으로부터 단음절이 자동 식별된다. 이것은 종래의 24개의 기본문자의 입력으로서 분리되지 않은 것이다. 이 △parameter는 하나의 기호에 지나지 않은 극히 간단한 개념이지만 한글의 합성, 처리 및 data entry 등 system 설계의 전반에 걸쳐 비상한 효과를 발휘하는 중요한 작용을 한다.

△PARAMETER의 효과

(1) 5개의 쌍자음을 따로 두지 않고, 정의에 따라 △parameter의 작용에 의하여 5개의 쌍자음을 표현, 생성된다.

(2) 30종의 문자 form pattern이 9종류로 되어 문자합성이 간단해 진다. (합성 program이 $\frac{1}{3}$ 로 축소된다.)

(3) 종래의 단음절 space의 빈도 18.1%가 제거되며 단음절이 자동식별되고, 전송속도 및 효율이 13%로 향상되며, 또 shift조작이 없어진다.

(4) 24+1의 기본문자의 입력만으로서 음절 space없이 자동합성 되고 문자의 품질이 향상된다.

(5) 현용 각종 Keyboard에서 쌍자음을 제거하면 대부분이 표준 TTY Keyboard와 같게 되며 자동적으로 통일된다. (몇개의 기호만 조절)

(6) code가 자동적으로 합리화되고, 자음과 모음이 분리되고 또 모음은 종모음과 횡모음으로 분리되어 system의 구성에 따라 효과적이 된다.

(7) 종래의 논란이 되어온 keyboard의 자음과 모음의 좌우 배치가 balance 되어 인간공학적으로 합리화된다. $P(C) < P(V) + P(S)$

단 단어 space를 오른손으로 조작

(8) key의 연타에 대한 결점이 제거된다.

(9) △parameter로서 쌍자음을 직접 생성할때는 더욱 효과가 있다.

(10) typist의 특별훈련의 필요성이 없다.

(11) 현재 51자 표준 code의 불요부분이 제거된다.

ASC II와 똑같이 보이지만 ASC II가 갖지 않은 특별한 기능을 가진다. 즉 한글의 자음과 모음이 두 군으로 분리되고, 또 모음은 다시 종모음과 횡모음 군으로 분리하였기 때문에 이들 부분 code는 한글의 구조적인 의미를 가지게 되어 한글에 관한한 system의 구성에 따라서는 기능면에서 효과적인 작용을 하게 된다.

부록에서 (ㄱ, ㄷ, ㄴ)의 세계 종모음은 컴퓨터 단말에서는 반드시 필요한 것은 아니고, 본질적인 것을 아니지만, 이들 출현 확률은 3.3%으로 되어, 한글만 사용되는 TTY에서는 속도가 약간 향상되고, code 및 data entry에서 공간의 여유가 있기 때문에 배당하였으므로 설계자의 자의로 선택여부를 결정해도 무방하겠으나, 능률 향상으로 두는 편이 유리하다.

또 b_7, b_6 bit를 차례로 삭제하면 그대로 6bit, 5bit TTY용으로 사용될 수 있다. 또 자동동음의 분류에서 한국어에서는 주로 자음만 사용되고 모음은 종속적으로 이용되므로 모음의 구조적인 분리를 위하여 모음 "]"의 위치를 옮겨 놓았다. 그러므로써 모음의 위치별 code분리가 자동화되고 또 두 group의 구조적 특징점을 검출할 수 있다. 부록의 한 한 page별 한글 code화는 ISOR-646의 규정에 따른것이고, shift SI, SO로서 ASC II와 한글 code를 분리한다. 또 ISOR-646규정에서는 자국의 사정을 감안해서 national use를 규정하고, 5/11, 12, 13 및 7/11, 12, 13의 간을 개방하고 있어 원화기호 ₩는 한글란에 넣을 수도 있겠지만, 수자 mode를 고려에 넣고 ASC II 5/12위치에 배당하였고, 한글 page의 기능부호 등은 ASC II을 그대로 사용한다.

5. 결 과

이 연구의 성과를 종합하면 다음과 같이 요약된다.

(1) 정보처리과정에서 문자로 작용하는 단음절 space의 미치는 영향을 분석하고, 지금까지 명확히 제시되지 않은 단음절 자동식별로 효율이 13% 이상 향상된다는 것을 명시하여 system 설계의 평가자료를 제공하였다.

(2) KS Morse code를 분석하고 합리적인 code의 구성으로서 전송속도를 14~16% 개선하였다.

(3) 빈도 18.1%의 단음절을 자동식별하므로 속도 및 효율이 13% 향상된다.

(4) △parameter의 도입으로 문제점을 일지에 제거

하고 쌍자음 생성, 단음절의 자동식별, code 및 data entry의 합리화 모든 문제가 해결된다.

(5) (모음 $P(V)$ + (단어 space $P(S)$)로 할때 자음 $P(C)$ 와 바란스 된다는 것을 계산하므로써 자판의 자음과 모음의 위치 배정에 중요한 자료가 되며 종래의 논쟁을 구명하였다.

이상은 한글 단말장치의 설계에 있어서 중요한 자료가 될 것으로 기대된다.

參 考 文 獻

1. C.E. Shannon: Prediction and Entropy of printed English. Bell system J. vol. 29. 1951.
2. G.A. Barnard III: statistical calculation of word Entropies for Four western languages. IEEE.

Trans. vol T-1 1955.

3. Joo K. Lee: Korean, Character Display by variable combination method KEIO Engineering Reports vol.26. No.10 1973.
4. 李柱根, 崔興文: 韓國語單音節의 Entropy에 관한 研究, 電子工學會誌 vol.11 No.3 1974.
5. 李柱根, 南宮在贊: 한글의 단음절의 자동식별 전자공학회지 vol.13. No.5 1976.
6. 南宮在贊: 한글의 단음절의 자동식별 및 전자 system 인화대학원 석사논문 1976-9.
7. 한글 Display 및 인식문제 재미 과학자 학술대회 1977. 4.
8. 한글의 code 개선, 전자공학회 하기 세미나

부록 ASC II와 한글 code
ASC II and HANGEUL code

parity bit		S I							SO															
b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	N	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1
0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	NUL (TG) DLE SP	0	@	P	⁽²⁾ ⁽³⁾ P											
0	0	0	1	1	(TC ₁) SOH	DC ₁	!	1	A	Q	a	q									△	ㅏ	ㅑ	ㅓ
0	0	1	0	2	(PC ₂) STX	DC ₂	"	2	B	R	b	r									△	ㅕ	ㅗ	ㅛ
0	0	1	1	3	(TC ₃) ETX	DC ₃	#	3	C	S	c	s									△	ㅓ	ㅕ	ㅗ
0	1	0	0	4	(TC ₄) EOT	DC ₄	\$	4	D	T	d	t									△	ㅓ	ㅕ	ㅗ
0	1	0	1	5	(TC ₅) ENQ (TC ₈) NAK	%	5	5	E	U	e	u									△	ㅓ	ㅕ	ㅗ
0	1	1	0	6	(TC ₆) ACK (TC ₉) SYN	&	6	6	F	V	f	N									△	ㅓ	ㅕ	ㅗ
0	1	1	1	7	BEL (TC ₁₀) ETB ⁽²⁾	'	7	7	G	W	g	v									△	ㅓ	ㅕ	ㅗ
1	0	0	0	8	FE ₀ (BS)	CAN)	8	H	X	h	x									△	ㅓ	ㅕ	ㅗ
1	0	0	1	9	FE ₁ (HT)	EM	(9	I	Y	i	y									△	ㅓ	ㅕ	ㅗ
1	0	1	0	10	FE ₂ (LF)	SUB	*	:	J	Z	j	z									△	ㅓ	ㅕ	ㅗ
1	0	1	1	11	FE ₃ (VT)	ESC	+	;	K	[k	{									△	ㅓ	ㅕ	ㅗ
1	1	0	0	12	FE ₄ (FF)	IS ₄ (FS)	,	<	L	₩	l										△	ㅓ	ㅕ	ㅗ
1	1	0	1	13	FE ₅ (CR)	IS ₃ (GS)	-	=	M]	m	}									△	ㅓ	ㅕ	ㅗ
1	1	1	0	14	SO	IS ₂ (RS)	.	>	N	^	n	-									△	ㅓ	ㅕ	ㅗ
1	1	1	1	15	SI	IS ₁ (US)	/	?	O	-	o	DEL									△	ㅓ	ㅕ	ㅗ

*. 왼쪽 ASC II 기능, 수자 부호 등은 오른쪽의 한글판에 공동으로 사용한다.