平成 13 年度

卒業研究論文

ロボットの聴覚機能実現の研究

98ｋｃ090　中島 一晃

98ｋｃ149　吉田 友輔

東京電機大学情報通信工学科

音響信号処理研究室

指導教授：金田 豊

目次

１．はじめに

われわれの周辺には音が充満し、音の存在は生活の背景として至極当然のこととなっているのである。そして音の中で生き、耳を持つことの利便は、全くはかり知れないものがある。特に、言葉という音は人と人とを結ぶ架け橋であり、日常生活の中で、音は人とその環境との間の最も重要な橋渡し役を担っているのである。情報伝達に、危険防止に、方向探知に、リクリエーション等などに欠くことのできない存在となっている。

人間はもちろん、あらゆる動物はまずその生命を維持するために環境に適応した行動をとらねばならない。生物の行動は変化に富んでおり、一見したところでは、いったいどのくらい行動の種類があるのか見当もつかない。しかし十分長い期間にわたって観察すると生物は周期的な生活をしており、ある行動のパターンはたびたび繰り返されている事が分かる。生物は幾種類かの型にはまった行動―威嚇、攻撃、逃避、食餌、求愛等など―を“持ち駒”として常時準備しており、環境からくる刺激の中にある特定の合図の現われを認めると、持ち駒の中の最も適したものを選んだものを選びその場面に適応しようとする。すなわちその場合の刺激（音声など）はある行動を起こす信号の役目となるのである。

「2000 年からの 10 年間は間違いなくロボットの時代」と言われるように、近い将来に人間の生活空間で活動する人間のパートナー型ロボットが多く誕生するであろう。

そこでロボットにも聴覚機能を搭載し、刺激である音声を発すると、音源方向や音声をロボット自身で認識し、その音声に適応した行動をとれることができるようにすることを目指す。

２．研究目的

ロボットの聴覚機能の実現において音声を取り込み、ロボットの動作制御までの一連の流れの中には多くの課題がある。まず色々な妨害音の中からの欲しい音声だけを取り込むこと。またその音声がどの方向から聞こえているかを判断する音源方向の検出。何を言われたかを判断する音声認識判断。その次にやっとその音声による命令に従うロボットの動作制御となる。今回のロボットの聴覚機能実現の研究においては方向検出と音声認識判断に重点をおいて研究を進めた。

２つのマイクロホンにより音を取り込みそれぞれに入ってくる音の時間の差をＭＡＴＬＡＢにおいて作成したプログラムによって計算し音源の方向を角度によって割り出す。

最終的にそのプログラムを市販のロボットに移行し、そのロボットに何らかのアクションを起こさせる。またそのプログラムにおいて音声認識判断を行い、それらの違いでロボットのアクションも異なるようにする。

３．音源方向の検出

３－１．音源方向の検出原理

距離がｄだけ離れた２つのマイクロホンＭ１、Ｍ２に到達する音波の距離差ξ、即ち時間差τｓによって角度θｓを割り出す。音波の録音状況を図１に示す。

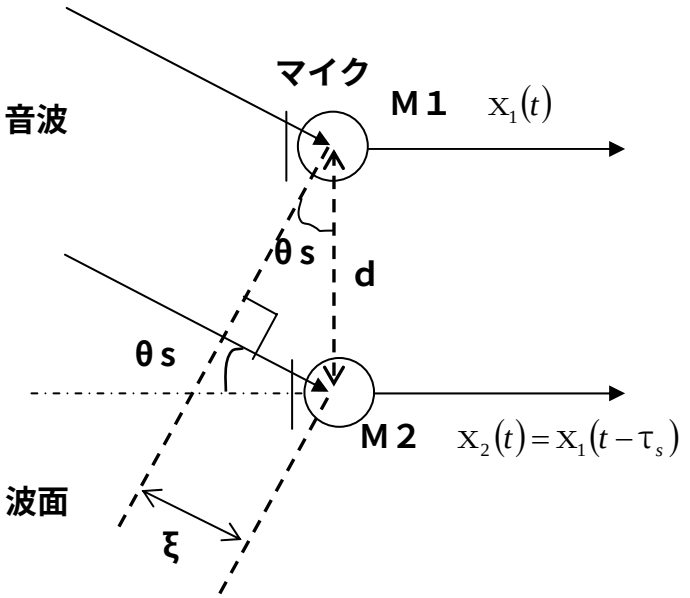

図１：音波とマイクロホンの位置関係

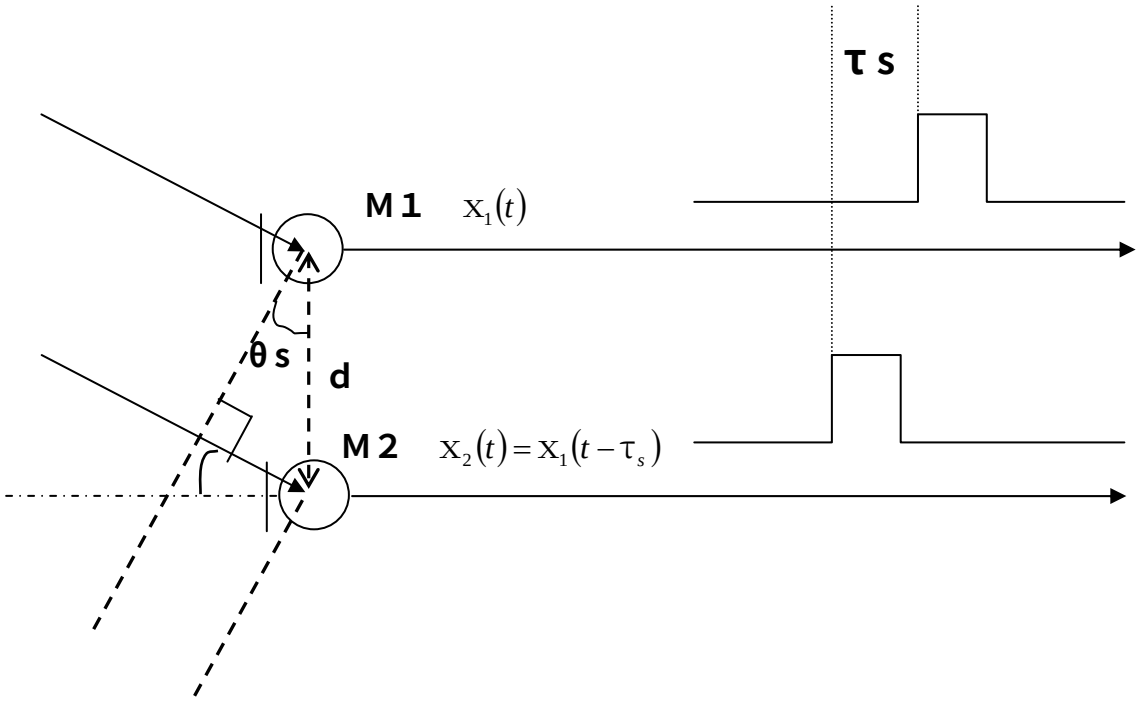

図２：受音信号と遅延時間の関係

まずθｓから到来した音波はマイクロホンＭ１において録音される。次に距離ξを進んだぶん時間τｓ遅れてマイクロホンＭ２に到達し録音される。この距離差ξは

$$\xi = d\sin\theta_s \qquad \cdots①$$

と表すことができる。図２からも分かるようにＭ２での受音信号ｘ$_2$（ｔ）はＭ１での受音信号ｘ$_1$（ｔ）と比べて音波が距離ξだけ進むのにかかった時間τｓだけ遅れた信号になっている。この時間差τｓは音速をｃとすると

$$\tau_s = \xi / c = (d\sin\theta_s)/c \qquad \cdots②$$

となる。①式と②式より

$$\theta_s = \sin^{-1}(c\cdot\ \tau_s / d) \qquad \cdots③$$

の関係式が成り立つ。したがって録音される時間差τｓが分かれば音の到来方向θｓつまり音源方向を割り出すことができる。

時間差τｓは受音信号ｘ$_1$（ｔ）、ｘ$_2$（ｔ）との相互相関関数φ$_{12}$（τ）から求めることができる。

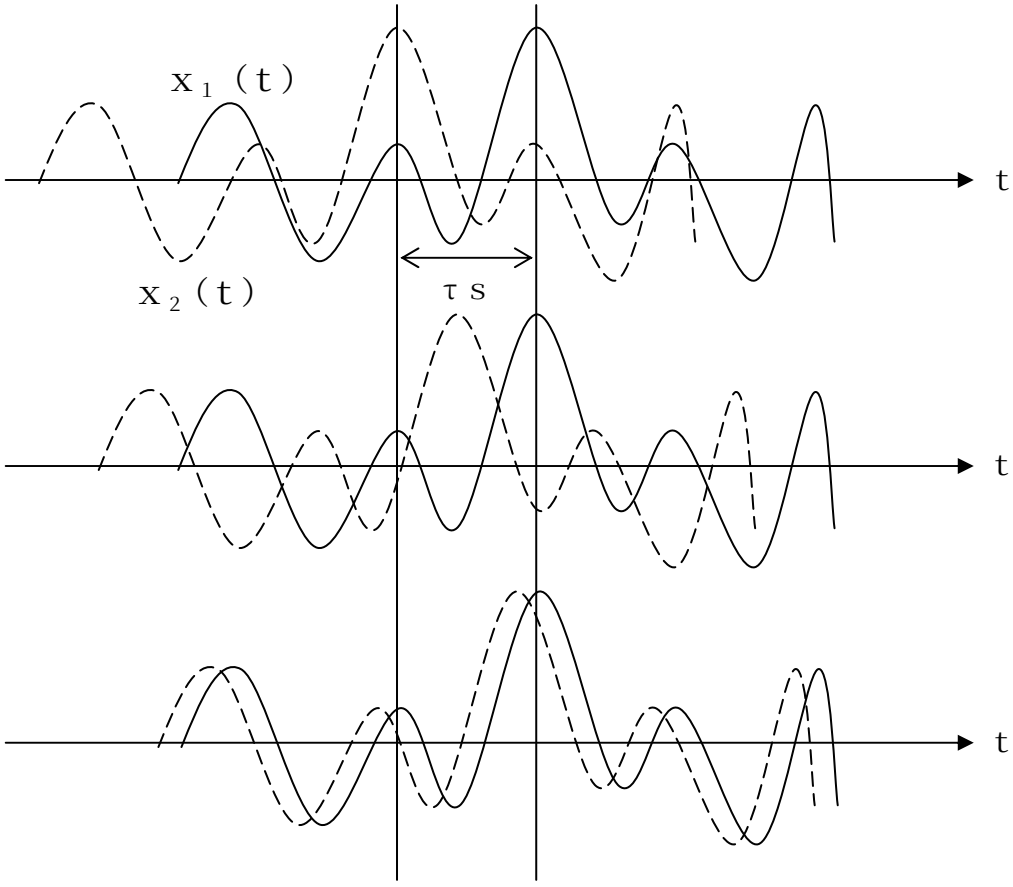

図３：相互相関関数の考えた

$\Phi_{12}(\tau)$の定義式

$$\phi_{12}(\tau) = (1/N)\sum\{x_1(t)\cdot x_2(t+\tau)\}$$

$$= (1/N)\sum\{x_1(t)\cdot x_1(t+\tau-\tau_s)\} \qquad \cdots④$$

において$\tau = \tau s$となる時に２つの受音信号$x_1(t)$、$x_2(t)$は同一波形となり、最大となるので$\Phi_{12}(\tau)$の最大値を与えるτを求めれば時間差τsが得られる。

３－２．音源方向検出のプログラム

実験条件としてマイクロホンＭ１、Ｍ２間の距離を 17[cm]とし、角度θはマイクロホンより人に向かい右側が－、左側を＋としマイクロホンＭ１、Ｍ２を回転させることによってθを±90[°]の範囲で変化させ測定を行う。

音源方向検出のプログラムを以下に示す。

```
%パラメータの設定
fs=48000                    %サンプリング周波数
c=340                       %音速
d=0.17                      %マイクロホン M1,M2 間の距離


[ｙ,fs]=wavread('d:\検出\record1\r45-2.wav')；
                            %音声を読み込んでくる

Ctsum=zeros(100,1)；        %100×1 の行列の Ctsum を作る

ysou=ｙ(：,1)；
xt=ｙ(：,2)；

taumax=30                   % taumax を 30 とする(1)

%相互相関関数の計算(2)

  for   tau=－taumax：taumax

        Ctsum(tau+31)=xt(taumax+1+tau      :      taumax+1+10000+tau)      ,
        *ysou(taumax+1：taumax+1+10000)/10000；

  end

figure(1)
plot(－taumax：taumax, Ctsum(1：taumax*2+1));
[yjiku xjiku]=max(Ctsum) %表示されたグラフの最大値を示す
                           ｘ軸の値を xjiku,
                           ｙ軸の値を yjiku とする

xjiku=xjiku－(taumax+1) %ｘ軸の値 0 が tmax+1 番目であるため
```

```
xlabel('時間 [サンプル]')  %ｘ軸のラベル表示
ylabel('相互相関関数')     %ｙ軸のラベル表示

whos

hoko=asin(c*xjiku/Fs/d); %時間差から角度[ラジアン]を求める

'kaku=',hoko/pi*180        %単位を[ラジアン]から[°]に変換
```

(1)taumax を 30 とした理由

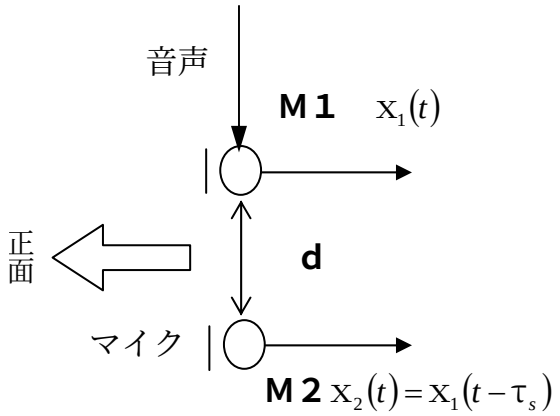

図４：τが最大になる時の音源とマイクロホンの位置関係

マイクロホン M1 での受音信号$x_1(t)$とマイクロホン M2 での受音信号$x_2(t)$の時間差τs が最大値(τ_{max})をとるときは、図４にあるように音波が正面方向に対してθs＝90[°]つまり直角に到来してきた場合である。

$$\tau_{max} = \xi / c$$

$$= d / c \quad [秒] \qquad \cdots⑤$$

上式の単位は[秒]となるので、[サンプル]にするにはサンプリング周波数 fs をかければいいので

$$\tau max = \tau_{max} \times fs$$

$$= d / c \times fs \quad [\text{サンプル}] \qquad \cdots⑥$$

実験条件と⑥式より

$$\tau\max = (0.17 / 340) \times 48000 = 24 \quad [\text{サンプル}]$$

となる。④式の相互相関関数$\phi_{12}(\tau)$は、τsで最大値をとることが分かっているので、$\phi_{12}(\tau)$が最大になる値は、τmaxより大きくなることはない。したがってプログラム中の変数 taumax は 24 を少し上まわる値である 30 に設定した。

(2)相互相関関数とは

相互相関関数とは異なる２つの信号波形に対して相関処理を行い、２つの信号波形の類似度を調べることである。N個のサンプリング値からなる信号波形$x_1(t)$と遅れ時間τs だけずらした信号波形$x_1(t+\tau)$においてそれぞれ対応する部分をかけ合わせて累積し、平均化する処理を行う。

３－３．音源方向検出の例

先に示したプログラムを実行すると次に示すグラフが得られる。

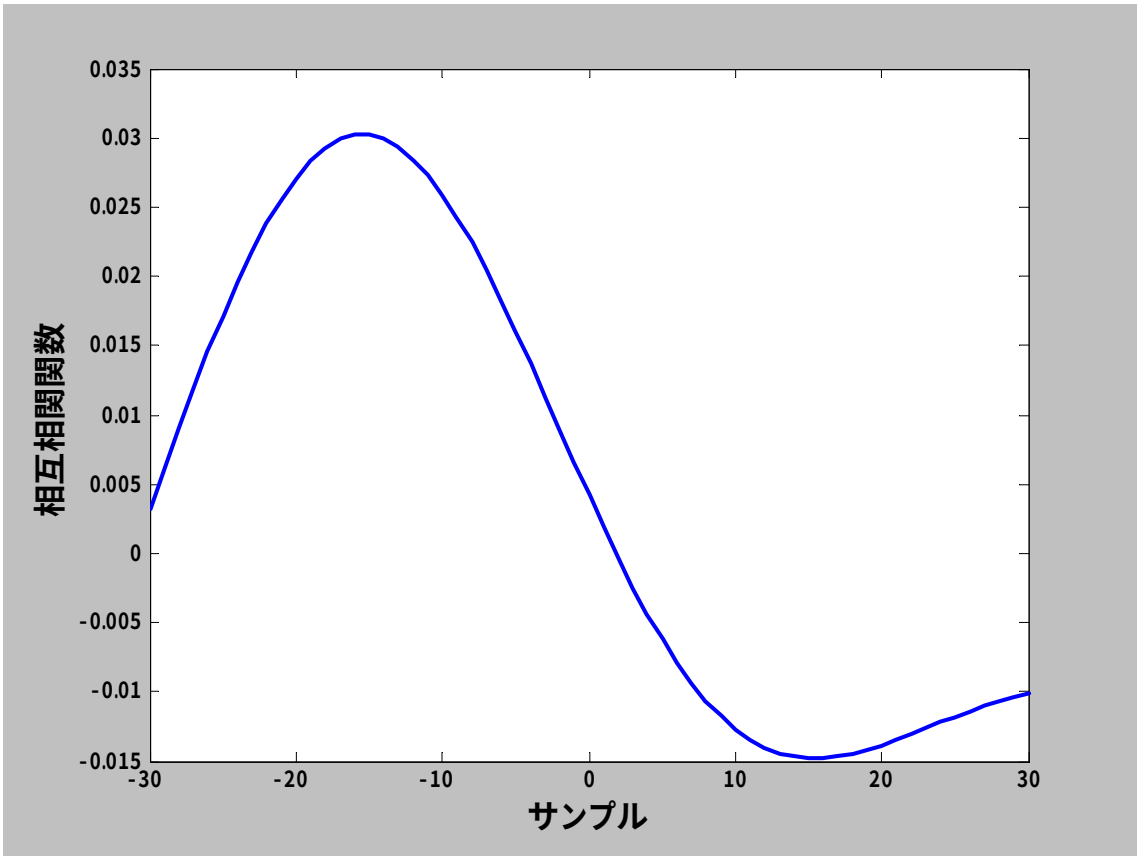

図５：相互相関関数のグラフ

相互相関関数が最大となる時のｘ軸の値が－16 となっているので音源方向はマイクロホンより人に向かい正面を０[°]として、左側に 41.81[°]となった。

この実験において使用した音声は左側 45[°]より録音したものを使用した。表１より分かるように 45[°]はサンプルが整数で表すことができない。そこで 45[°]に一番近い値は 17[サンプル]となるが次に近い値が 16[サンプル]の 41.81[°]となるので、評価としてはまずまずの結果と言えるだろう。

３－４．サンプルと検出角度の関係

サンプルと検出角度の関係を計算式により求める。

$$\theta_s = \sin^{-1}(c \cdot \tau_s / d)$$

τs はアナログ表示[秒]なので、ディジタル表示である(taus)[サンプル]にするにはサンプリング周波数 fs をかければいいので

$$taus = \tau_s \cdot fs$$
$$\tau_s = taus / fs$$

したがって

$$\theta_s = \sin^{-1}\left(c \cdot taus / fs / d\right) \qquad \cdots⑦$$

ここでθ s はラジアン計算している。よって求めたいθは

$$\theta = \theta_s / \pi \cdot 180$$

となるので

$$\theta = \sin^{-1}\left(c \cdot taus / fs / d\right) / \pi \cdot 180 \qquad \cdots⑧$$

で求めることができる。

実験条件であるパラメータを

fs=48000	(サンプリング周波数)
c=340	(音速)
d=0.17	(マイクロホン M1,M2 間の距離)

とし、求めた値をグラフにし次ページに示す。

表１：サンプルと検出角度の関係

サンプル	角度[°]
0.00	0.00
1.00	2.39
2.00	4.78
3.00	7018
4.00	9.59
5.00	12.02
6.00	14.48
7.00	16.96
8.00	19.47
9.00	22.02
10.00	24.62
11.00	27.28
12.00	30.00
13.00	32.80
14.00	35.69
15.00	38.68
16.00	41.81
16.96	45.00
17.00	45.10
18.00	48.60
19.00	52.34
20.00	56.44
20.78	60.00
21.00	61.05
22.00	66.44
23.00	73.40
24.00	90.00

サンプルと角度の関係は表１に示したようになったが、ここでは 90[°]を 24 等分(180[°]を 47 等分)にしたため、細かい角度まで求めることができなかった。また、角度が±90[°]に近くなると 1[サンプル]あがるごとに角度の開きが大きくなるので、人間の感覚と同じように正面付近はほぼ正確に方向検出をできだが±90[°]付近は誤差が大きく出ることがおおかった。

４．実音場における音源方向検出実験

４－１．近距離における方向検出

録音した音声の方向を検索する上でプログラムの精度を検証する必要がある。そこでマイクロホンから距離をおかずに近距離で録音し部屋の反射波やノイズ等の影響があまり出ないような状況で方向検出をした。

実験条件：マイクロホンと話者の位置関係を図６に示す。２つのマイクロホンＭ１、Ｍ２の距離ｄはそれぞれ 17[㎝]とし、角度θはマイクロホンより人に向かい右側が－、左を＋としマイクロホンＭ１、Ｍ２を回転させることでθを±90[°]の範囲で変化させて測定した。録音したデータは“あ”という母音のみとし、データの長さは１[秒]、１回目と２回目、３回目の録音は同一人物が行う事とする。

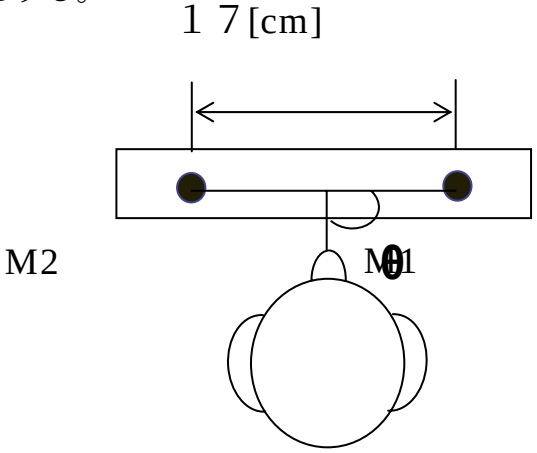

図６：音源とマイクロホンの関係

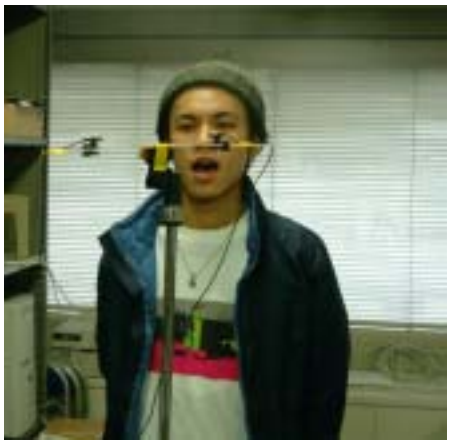

図７：録音風景

表２：音源方向と検出方向の関係

音源方向　θ[°]	検出方向θ[°]
0　１回目	4.7802
右　30　１回目	-24.6243
右　30　２回目	-27.2796
左　30　１回目	30
左　30　２回目	32.7972
右　45　１回目	-35.6853
右　45　２回目	-45.0995
左　45　１回目	45.0995
左　45　２回目	41.8103
右　60　１回目	-52.3415
右　60　２回目	-90
左　60　１回目	56.4427
左　60　２回目	56.4427
右　90　１回目	-90
右　90　２回目	-73.4022
右　90　３回目	-56.4427
左　90　１回目	90
左　90　２回目	90
左　90　３回目	90+39.7144i

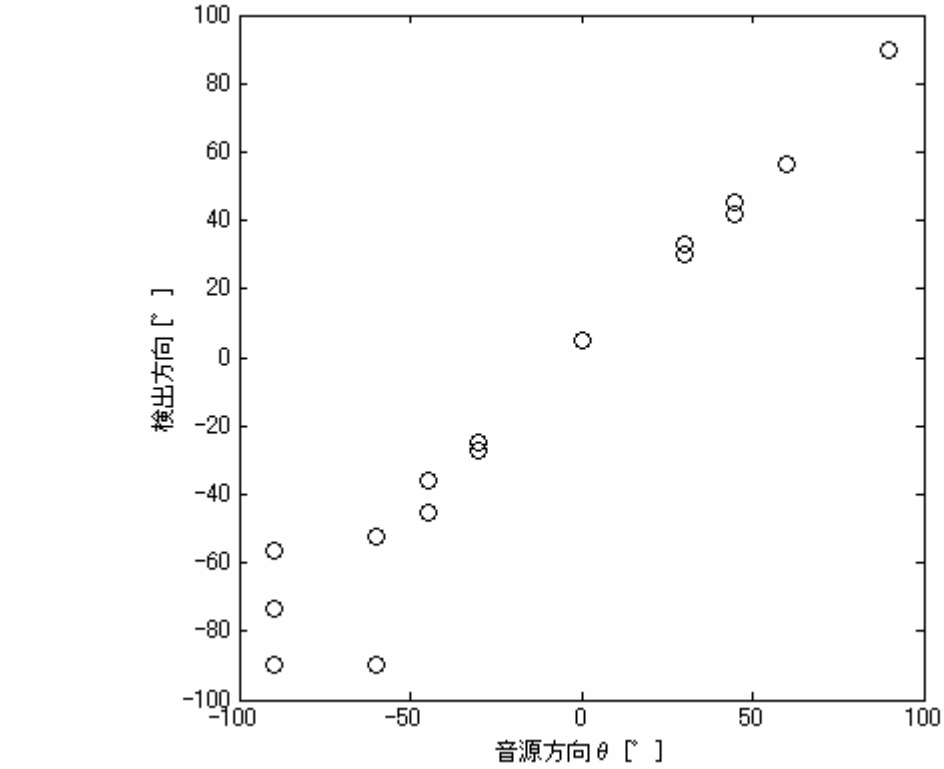

図８：音源方向と検出方向の対照図

図６に示すように近距離において録音した時の結果を表２と図８に示す。評価としては表２からも見て取れるようにさすがに近距離からの録音では、音源方向とプログラムで検出された方向が近い値をとっている。図８においてもプロットされた場所がほぼ直線に近い形になった。

４－２．周期音（母音）による距離をつけた実験

① マイクロホンに対していくつかの音源方向θをつけそれに距離をつけて録音する。

実験条件：マイクロホンと話者の位置関係を図９に示す。２つのマイクロホンＭ１、Ｍ２の距離ｄは 17[㎝]とし、角度θはマイクロホンより人に向かい右側が－、左を＋としマイクロホンＭ１、Ｍ２を回転させることでθを±90[°]の範囲で変化させて測定した。マイクロホンから人までの距離ｒをそれぞれ１[m]２[m]３[m]として録音を行った。録音したデータは“あ”という母音のみとし、データの長さは１[秒]、１回目と２回目の録音は同一人物が行う事とする。

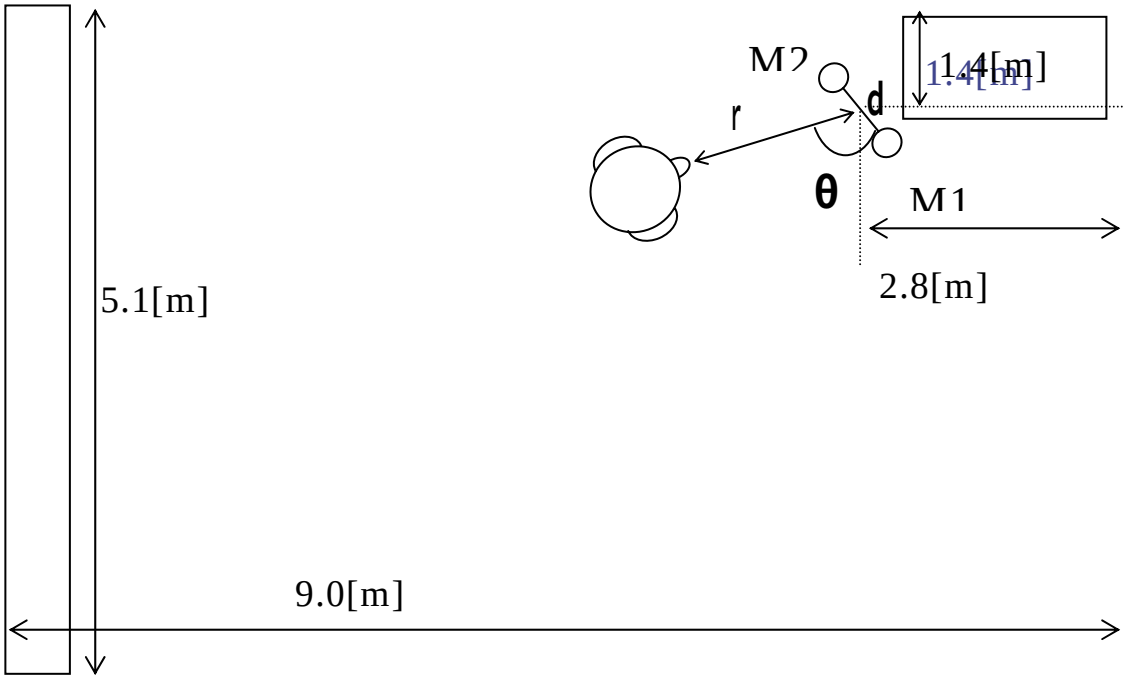

図９：部屋の端における録音状況

表３：音源方向θと検出方向の関係（部屋の端）

音源方向　θ[°]	マイクロホンからの距離		
	1[m]	2[m]	3[m]
	検出角度　θ[°]		
0　１回目	2.388	4.7802	14.4775
0　２回目	4.7802	4.7802	-12.0247
右　30　１回目	-4.7802	-12.0247	-4.7802
右　30　２回目	-22.0243	-16.9578	-7.1808
左　30　１回目	16.9578	22.0243	22.0243
左　30　２回目	19.4712	14.4775	16.9578
右　45　１回目	-48.5904	-45.0995	-14.4775
右　45　２回目	-45.0995	-56.4427	-56.4427
左　45　１回目	24.6243	30	90
左　45　２回目	45.0995	41.8103	90+16.48i
右　60　１回目	-56.4427	-56.4427	-12.0247
右　60　２回目	-52.3415	-45.0995	-66.4435
左　60　１回目	35.6853	48.5904	30
左　60　２回目	35.6853	45.0995	48.5904
右　90　１回目	-56.4427	-48.5904	-32.7972
右　90　２回目	-66.4435	-66.4435	-9.5941
左　90　１回目	73.4022	56.4427	24.6243
左　90　２回目	90+28.35i	90+23.23i	14.4775

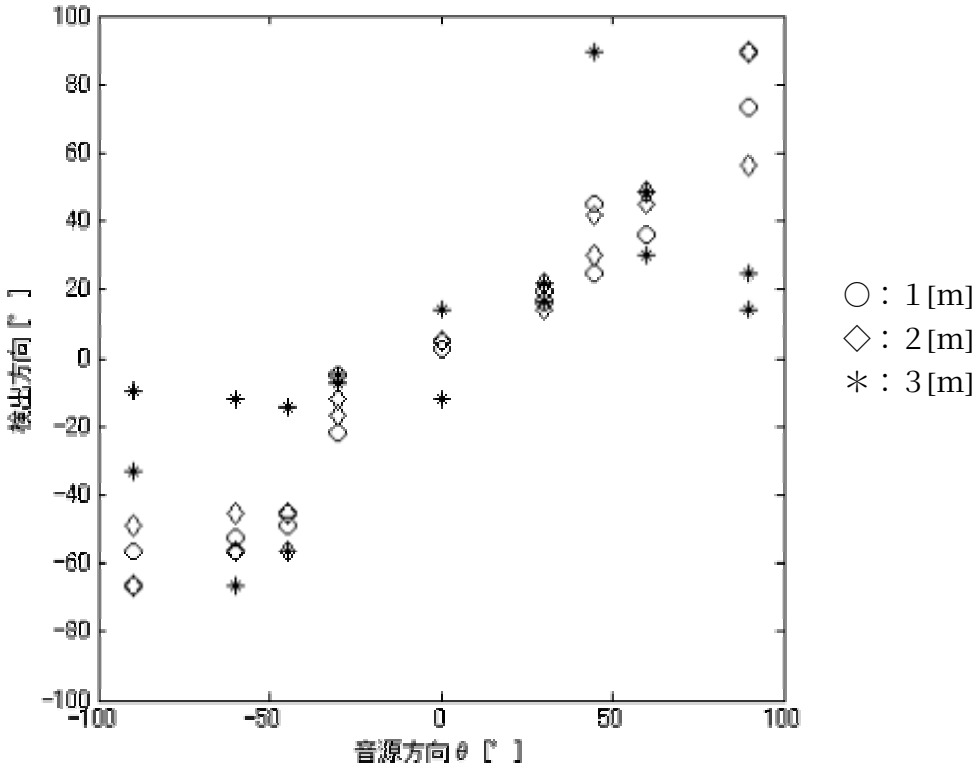

図 10：音源方向θと検出方向の対照図（部屋の端）

図９に示すようにマイクロホンを部屋の端に置いて録音した時の結果を表３と図 10 に示す。図 10 からも分かるように距離ｒが離れるにしたがって検出方向がずれてくる。ここで分かるのは３[m]の場合が一番分かりやすいが、±90[°]に近づくにつれ検出方向が０[°]に近づいている。

② マイクロホンの場所を変えて同様に録音する。

実験条件：マイクロホンと話者の位置関係は図 11 に示す通りである。このときは 1 回目と 2 回目の録音は別の人物が録音した。その他の条件は実験①と同様。

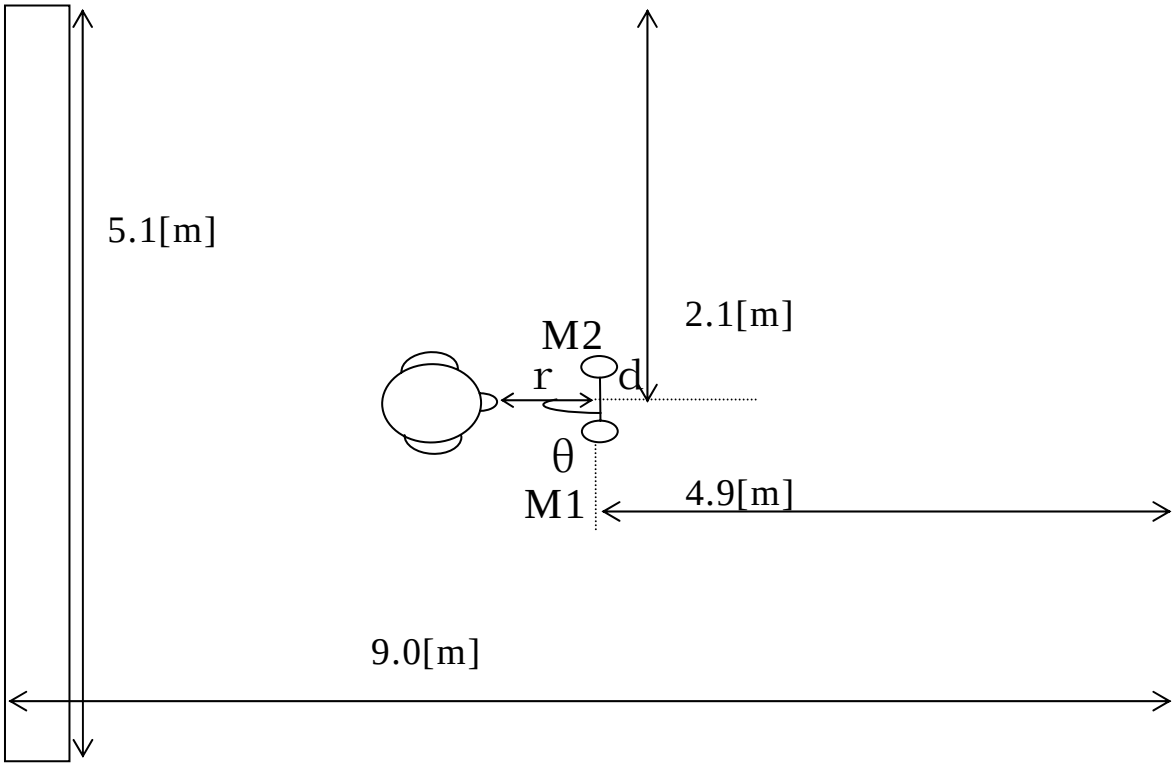

図 11：部屋の中心における録音状況

表４：音源方向θと検出方向の関係（部屋の中心）

音源方向　θ[°]	マイクロホンからの距離		
	1[m]	2[m]	3[m]
	検出角度　θ[°]		
0　１回目	4.7802	2.388	14.4775
0　２回目	2.388	2.388	0
左　30　１回目	22.0243	19.4712	90+42.76i
左　30　２回目	14.4775	16.9578	12.0247
右　30　１回目	-19.4712	-14.4775	-7.1803
右　30　２回目	-4.7802	-2.388	90+39.71i
左　45　１回目	27.2796	27.2796	73.4022
左　45　２回目	32.7972	24.6243	14.4775
右　45　１回目	-19.4712	-30	-30
右　45　２回目	-14.4775	-9.5941	-30
左　60　１回目	38.6822	32.7972	14.4775
左　60　２回目	24.6243	35.6853	4.7802
右　60　１回目	-35.6653	-27.2796	-19.4712
右　60　２回目	-38.6822	-24.6243	-12.0247
左　90　１回目	48.5904	16.9578	27.2796
左　90　２回目	41.8103	14.4775	7.1808
右　90　１回目	-22.0243	-24.6243	-9.5941
右　90　２回目	-48.5904	-45.0995	90+42.76i

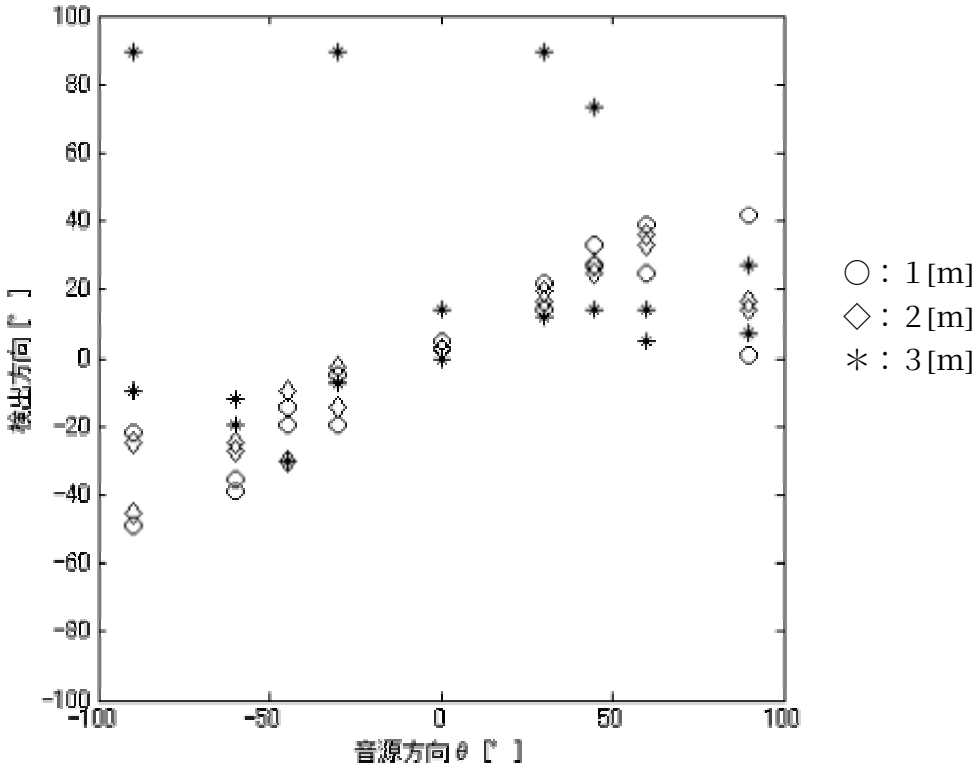

図 12：音源方向θと検出方向の対照図（部屋の中心）

図11に示すように部屋の中心にマイクロホンを置いた時の結果を表４と図12に示す。部屋の端に置いた時（図 10）と比べると全体的に０[°]に近い検出結果となっている。

４－３．不規則な波形の音を使用し距離をつけた実験

③　今までの録音では発声する音を「あ」等の母音（周期音）にしていた。しかし周期音で録音すると１周期ずれていてもそのずれを認識することができないという曖昧さがある。そこで濁音等の波形が不規則な音を使用し周期音の時と同様に部屋の端と中心の２通りの実験結果を図 13、14 に示す。

実験条件：マイクロホンと話者の位置関係は図９，12 に示す通りである。
その他の条件は実験①と同様。

表５：濁音時の音源方向θと検出方向の関係（部屋の端）

音源方向　θ[°]	マイクロホンからの距離		
	1[m]	2[m]	3[m]
	検出角度　θ[°]		
0　１回目	-4.7802	-4.7802	4.7802
0　２回目	-7.1808	-7.1808	-4.7802
右　30　１回目	-45.0995	-19.4712	24.6243
右　30　２回目	-41.8103	-32.7972	14.4775
左　30　１回目	61.045	41.8103	45.0995
左　30　２回目	56.4427	32.7972	-35.6853
右　45　１回目	-14.4775	-27.2796	-30
右　45　２回目	-24.6243	-27.2796	-35.6853
左　45　１回目	27.2796	24.6243	24.6243
左　45　２回目	32.7972	22.0243	35.6853
右　60　１回目	-22.0243	-16.9578	-30
右　60　２回目	-19.4712	-16.9578	-27.2796
左　60　１回目	16.9578	22.0243	16.9578
左　60　２回目	19.4712	22.0243	16.9578
右　90　１回目	-52.3415	-16.9578	-48.5904
右　90　２回目	-52.3415	-27.2796	-16.9578
左　90　１回目	56.4427	56.4427	19.4712
左　90　２回目	32.7972	41.8103	32.7972

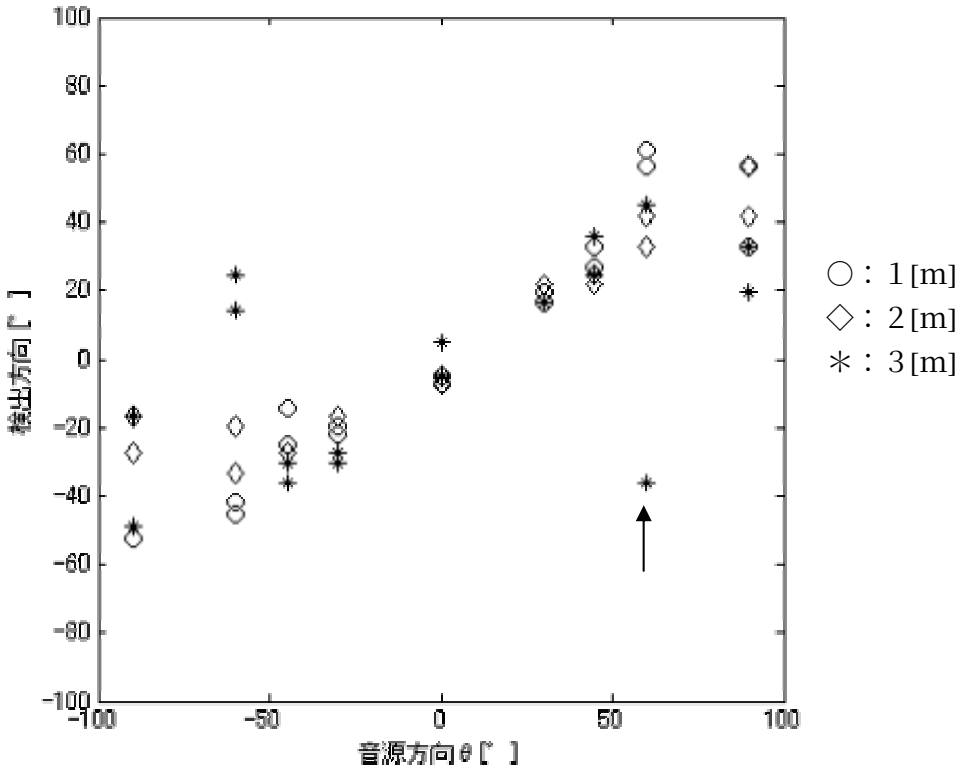

図 13：濁音時の音源方向θと検出方向の関係（部屋の端）

表６：濁音時の音源方向θと検出方向の関係（部屋の中心）

音源方向　θ[°]	マイクロホンからの距離		
	1[m]	2[m]	3[m]
	検出角度　θ[°]		
0　１回目	4.7802	4.7802	4.7802
0　２回目	4.7802	2.388	7.1808
右　30　１回目	-16.9578	-12.0247	-14.4775
右　30　２回目	-14.4775	-9.5941	-9.5941
左　30　１回目	35.6853	32.7972	27.2796
左　30　２回目	32.7972	19.4712	24.6243
右　45　１回目	-24.6243	-27.2796	-19.4712
右　45　２回目	-19.4712	-14.4775	-14.4775
左　45　１回目	41.8103	38.6822	32.7972
左　45　２回目	48.5904	35.6853	30
右　60　１回目	-38.6822	-35.6853	-30
右　60　２回目	-38.6822	-30	-24.6243
左　60　１回目	56.4427	45.0995	32.7972
左　60　２回目	45.0995	48.5904	45.0995
右　90　１回目	-52.3415	-38.6822	-27.2796
右　90　２回目	-61.045	-38.6822	-32.7972
左　90　１回目	61.045	52.3415	35.6853
左　90　２回目	66.4435	48.5904	45.0995

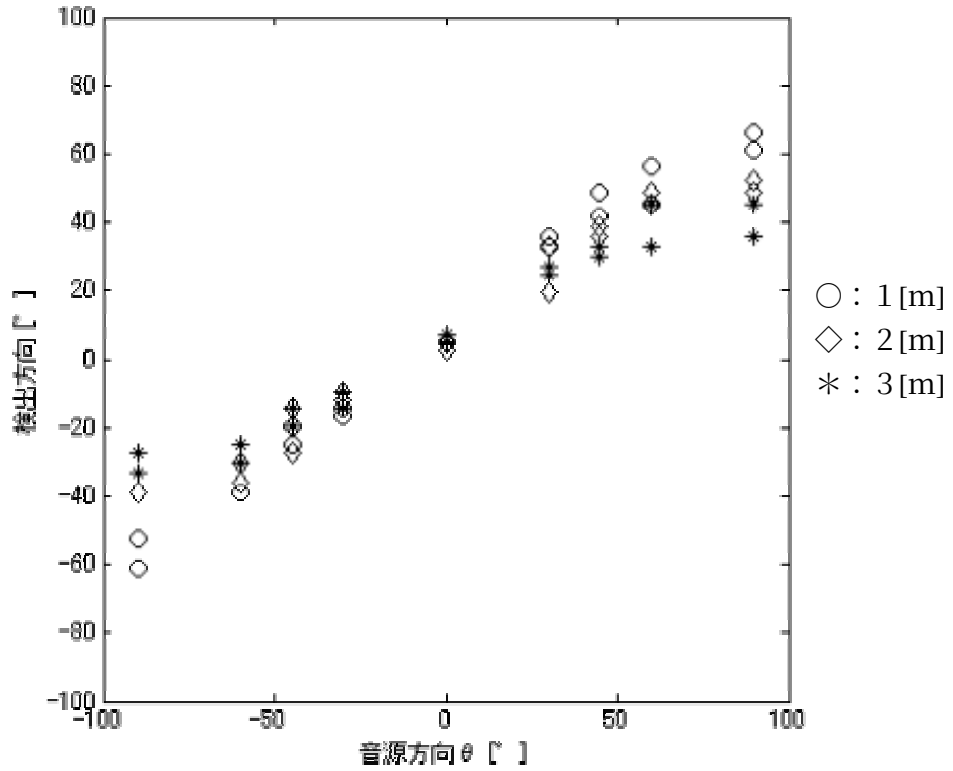

図 14：濁音時の音源方向θと検出方向の関係（部屋の中心）

図 10 と図 13、図 12 と図 14 を比較すると「あ」等の周期音と濁音等の不特定な波形ではやはり周期音の方が検出方向に散らばりがみられる。やはりこれは相互相関関数において計算される際、周期音のような信号波形より不特定な信号波形のほうが波形のずれを正確に捉える事ができるからではないかと考えられる。さらに図 13 と 14 を比較すると部屋の反射の影響が少ない部屋の中心で録音をおこなった図 14 の方がよりきれいなグラフとなった。

ここで図 13 において期待する値と違う点が数点ある。一例として図においての矢印で示した点の相関関数とサンプル数の関係を表した図が図 15 である。図 15 をみると本来右の山が高くなるはずだが左の山のほうに反射波が加わりエネルギー的に大きくなってしまいそのために正確な方向検出ができなかったと考えられる。

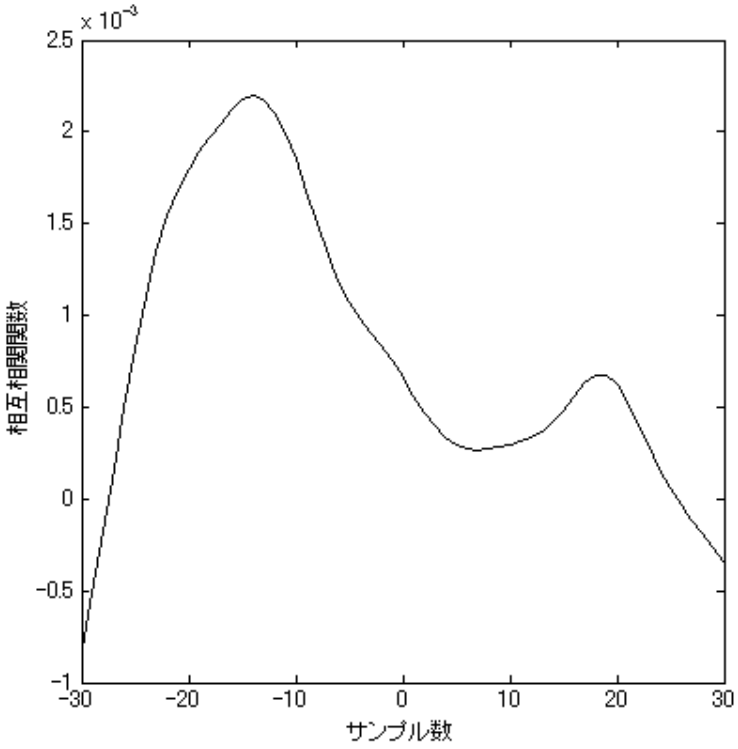

図 15：相関関数とサンプル数の関係

４－４．高域を強調する

今まで検出してきた周期音または濁音等の不規則な音波では、図 16 のように低域であるエネルギーの強い部分だけで検出してきた可能性がある。そこで図 17 のように高域のエネルギーの弱い部分を強調し、録音した音波のすべての部分において方向検出をするようにする白色化を試みた。

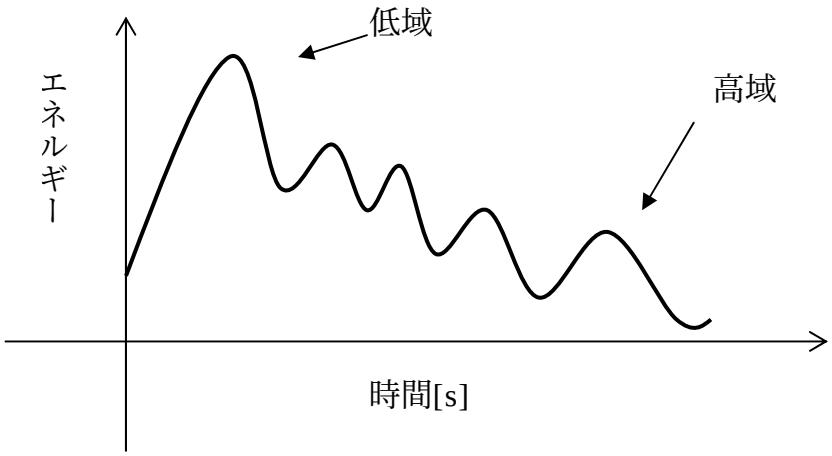

図 16：一般的な波形

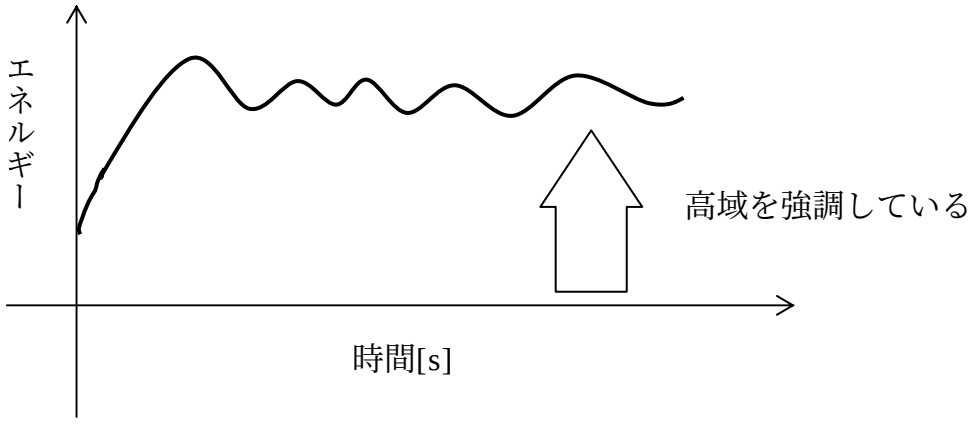

図 17：高域を強調した時の波形

以上のことをプログラムで表すと以下のようになる。

```
%パラメータの設定
fs=48000                    %サンプリング周波数
c=340                       %音速
d=0.17                      %マイクロホン M1,M2 間の距離

 [ｙ,ｆs]=wavread('d:\検出\record2\032.wav')；
                            %音声を読み込んでくる

%白色化

a=0.7
NN=length(y)
y0=y(2:NN,:)-a*y(1:NN-1,:);


Ctsum=zeros(100,1)；      %100×1 の行列の Ctsum を作る

ysou=ｙ0(：,1)；          %ysou に y0 の１列目を代入
xt=ｙ0(：,2)；            %xt に y0 の２列目を代入

taumax=30                 % taumax を 30 とする


%相互相関関数の計算

  for   tau=－taumax：taumax

        Ctsum(tau+31)=xt(taumax+1+tau      :      taumax+1+10000+tau)      ,
        *ysou(taumax+1：taumax+1+10000)/10000；

  end

figure(1)
plot(－taumax：taumax, Ctsum(1：taumax*2+1));
[yjiku xjiku]=max(Ctsum) %表示されたグラフの最大値を示す
                           ｘ軸の値を xjiku,
                           ｙ軸の値を yjiku とする
```

```
xjiku=xjiku－(taumax+1) ％ｘ軸の値０が tmax+1 番目であるため

xlabel('時間 [サンプル]')  ％ｘ軸のラベル表示
ylabel('相互相関関数')      ％ｙ軸のラベル表示

whos

hoko=asin(c*xjiku/Fs/d); %時間差から角度[ラジアン]を求める

'kaku=',hoko/pi*180        %単位を[ラジアン]から[°]に変換
```

結果としては検出方向には直接変化は現れなかった。ただ相互相関関数とサンプル数のグラフにおいて図 18 と図 19 を比べた時図に示した矢印の部分に微妙な差が出た。（以下の図は表 3 における０[°]、3[m]、２回目のものである。）

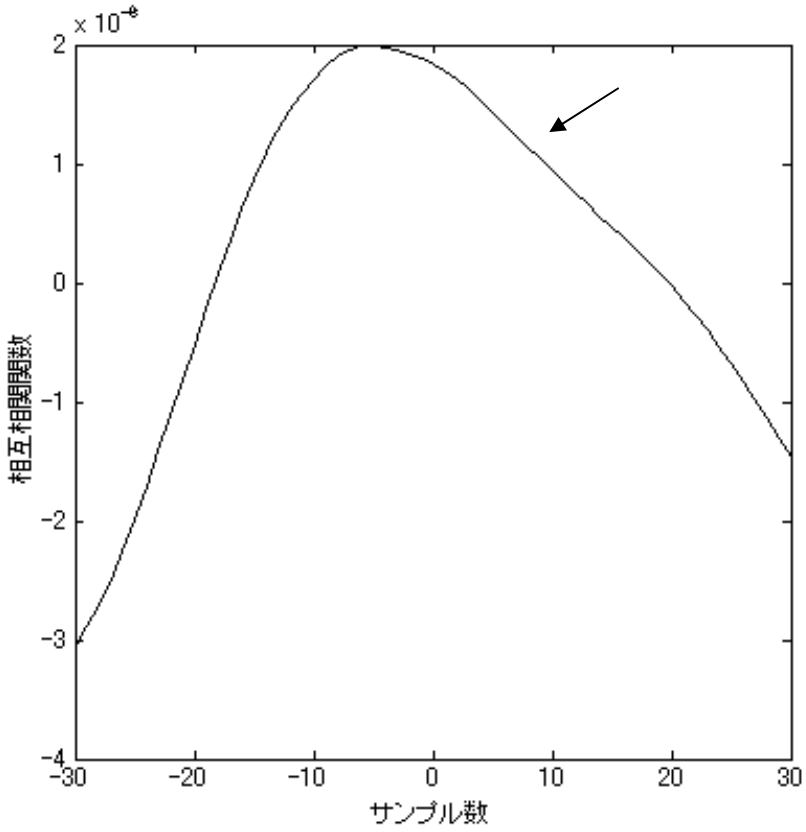

図 18：白色化していない相関関数とサンプル数の関係

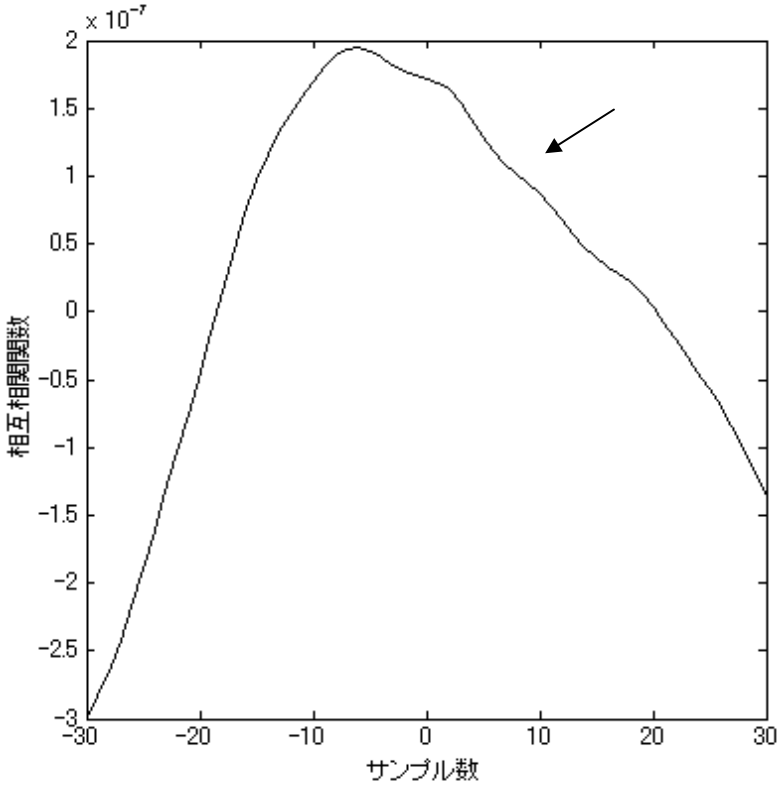

図 19：白色化した相関関数とサンプル数の関係

４－５．立体的に考える

今までの研究では、録音を平面的に行っていたが、90[°]付近では平面的に数度ずれていても検出方向に大きく影響を与えていたが、それは立体的（3 次元空間的）に考えても同様のことがいえる。

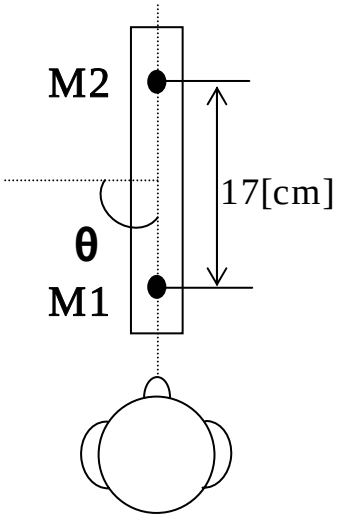

図 20：平面的に考えた場合

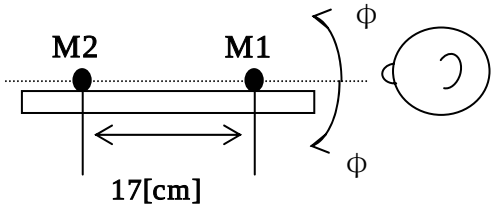

図 21：立体的に考えた場合

図 20,21 から分かるように平面的に見たθ=90[°]において（図 20）、音源位置が立体的に（上下に）数度ずれた場合（図 21）今回の判定方法では、平面方向のずれと判断してしまう。

そこで立体的に見た音源方向と検出できる信号の時間差について考察してみた。具体的には図 22 のように平面上での音源から 2 つのマイクロホンへの距離の差を立体的に考察してみた。

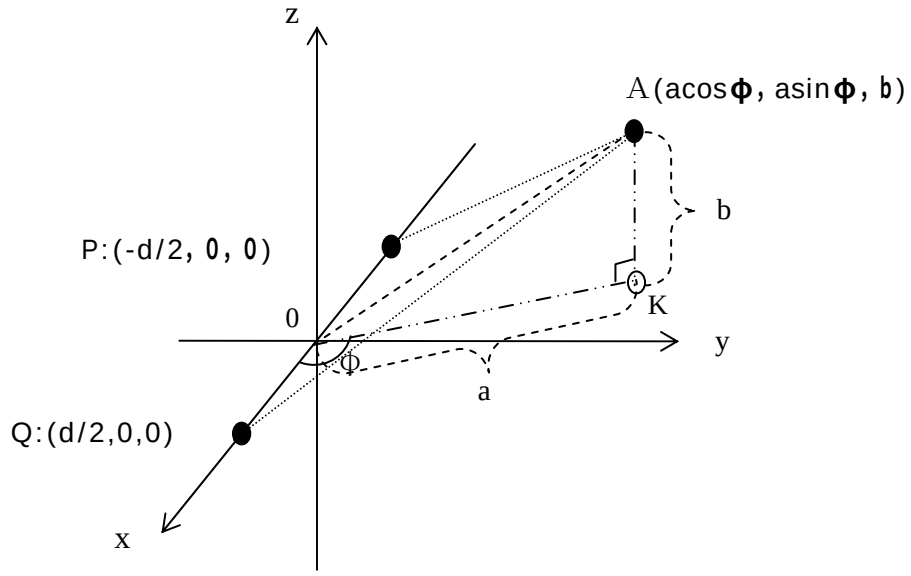

図 22：音源とマイクロホンの位置関係

Ａ：音源
Ｐ：マイクロホン M1
Ｑ：マイクロホン M2
φ：x-y 平面においての原点から音源への角度
ｄ：2 つのマイクロホン間の距離
ａ：Ａ－Ｋ間の距離
ｂ：Ａの高さ

音源 A から２つのマイクロホン Ｍ１、Ｍ２の距離の差をαとすると求める距離差αは以下の式で表せる。

$$\alpha = |A-P| - |A-Q|$$
$$= \sqrt{(a\cos\phi - d/2)^2 + (a\sin\phi)^2 + b^2} - \sqrt{(a\cos\phi + d/2)^2 + (a\sin\phi)^2 + b^2}$$

この式をプログラムの中に使用するのだが、本研究では実際にはそこまで到達しなかった。

５．音声認識判断

「音声に対してロボットを反応させる時にただ音に反応するだけではなく、音声をきちんと認識することができ、それに対応させて音声別に行動をとれるようなプログラムを組んでいきたい」という目標で研究を進めた。

本研究では“あ・い・う・え・お”といった母音のみの音声認識について重点をおいて検討をした。

５－１．音声認識判断の研究方法

まずはじめに“あ・い・う・え・お”といった母音のみをそれぞれ数回録音し、ＭＡＴＬＡＢに取り込み周波数スペクトルで表す。次にその周波数スペクトルを逆ＦＦＴ変換してケプストラムを求める。“あ”の音声において求めたケプストラムを図に示す。

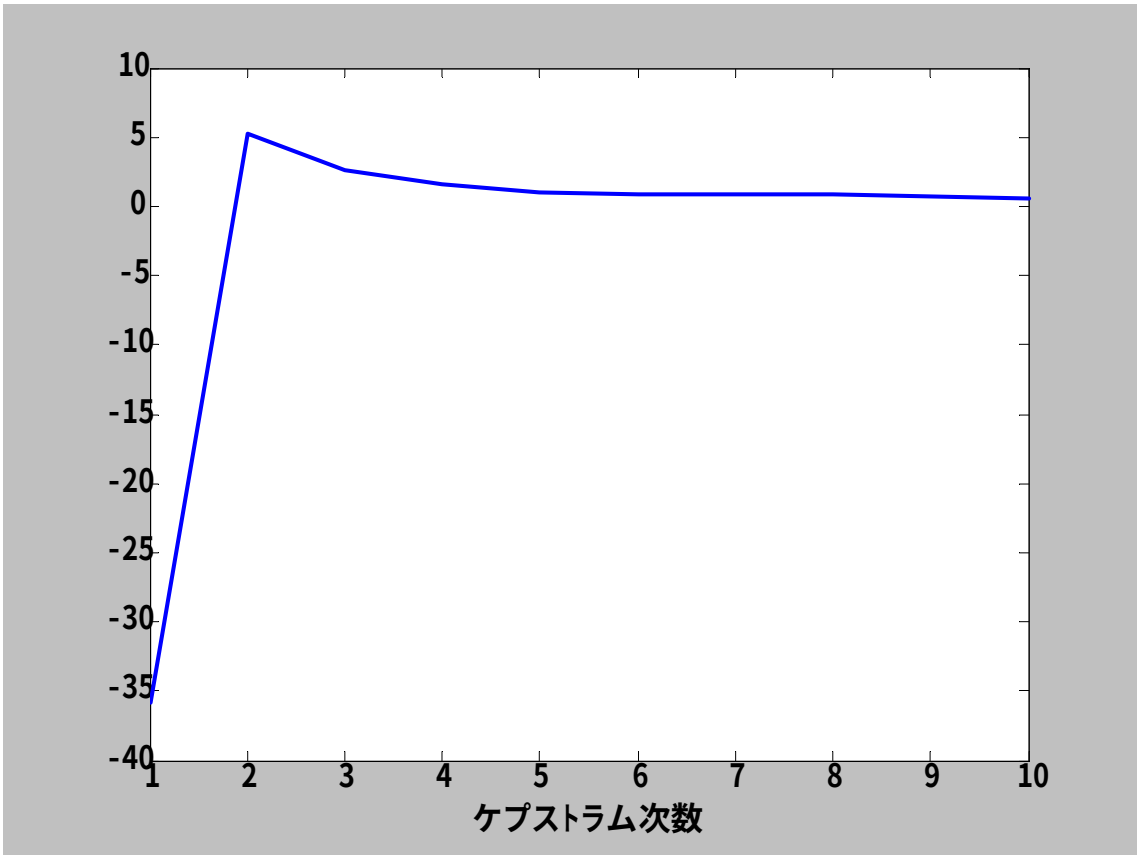

図 23：“あ”の音声ケプストラム

ここで２次ケプストラムは傾斜なので無視をして、３次ケプストラムをｘ軸、４次ケプストラムを ｙ 軸としてグラフを作成し、それぞれの特徴を捉え比較する。

５－２．音声特徴量の分析

音声特徴量の分析のプログラミングを以下に示す。

```
%パラメータの設定
fs=48000;                    %サンプリング周波数

[y,fs]=wavread('d:\検出\record6\a5.wav');
                             %音声を読み込んでくる

Ctsum=zeros(100,1);          %100×1 の行列の Ctsum を作る

ysou=y(:,1);                 % ysou に y の１列目を代入

%ＦＦＴ変換する
ip=1;
nfft0=2048;


for iii=1:20

yfsum=zeros(nfft0,1);
Nrep=1

 for i=1:Nrep
     y1=y(ip:ip+nfft0-1);
     yf=abs(fft(y1)).*abs(fft(y1));
     yfsum=yfsum+yfsum;
     ip=ip+nfft0;
 end

yfsum=yfsum/Nrep;

%逆ＦＦＴ変換する
NFFT=length(yfsum);
x=1:nfft0;
fx=(x-1)*(fs/2)/(NFFT/2);
```

```
ylog=10*log10(yf);
cep1=ifft(ylog);


figure(1);
plot(cep1(3),cep1(4),'o');      %3 次ケプストラムの値をｘ軸,
                                 4 次ケプストラムの値をｙ軸としてプロット

hold on
xlabel('3 次ケプストラム')       %ｘ軸のラベル表示
ylabel('4 次ケプストラム')       %ｙ軸のラベル表示
axis([0 7 0 3]);                %グラフの範囲指定

end
```

５－３．分析結果

図 24、25、26、27、28 は“ あ・い・う・え・お ”といった母音のみ１秒間録音したものをプログラムを実行させることにより音声を 20 個に区切りそれぞれの音声特徴を分析し、3 次ケプストラムの値と 4 次ケプストラムの値をプロットしたものである。

図 29 と図 30 は“ あ・い・う・え・お ”それぞれの母音を繰り返し重ねてプロットしたものである。見て分かるように 2 次ケプストラムと 3 次ケプストラムを使用して判別するより、３次ケプストラムと４次ケプストラムを使用した方が分解度はあがった。

図 31 と図 32 はそれぞれ“ あ・お ”の音声特徴と“ あ・い ”の音声特徴を表したものである。２つの図より、比較してみると“ あ・い ”の音声特徴の差ははっきりと見て取れるが、“ あ・お ”の音声特徴においてはほとんど似た音声特徴を持っており判別することは難しかった。

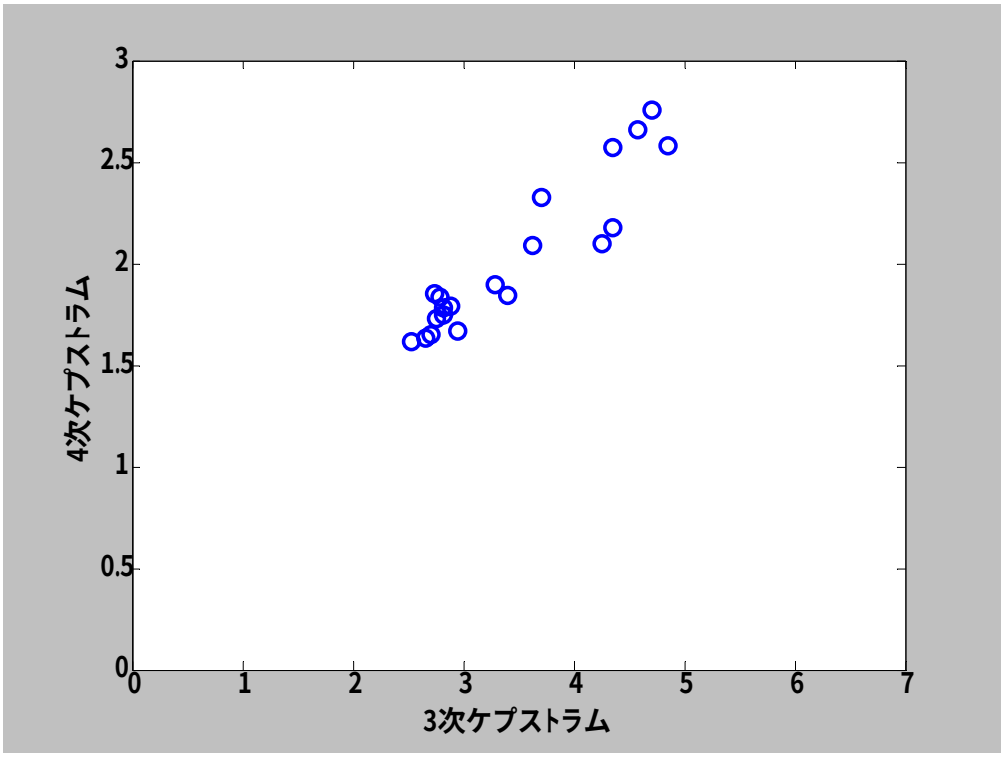

図 24 :“ あ ” の音声特徴

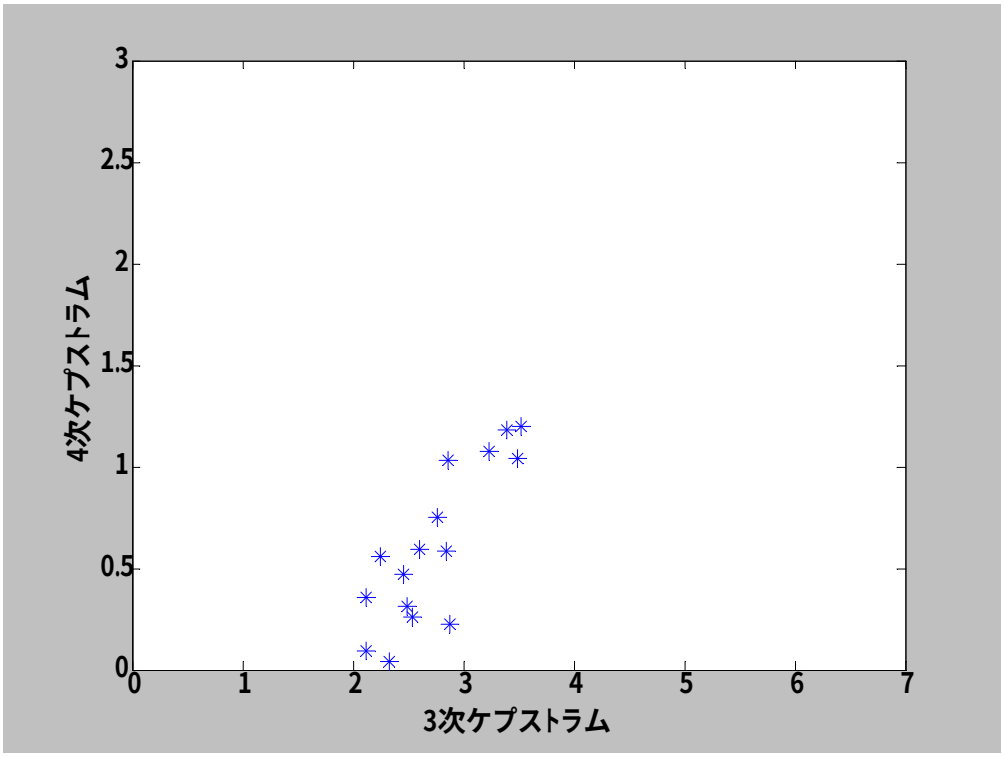

図 25 :“ い ” の音声特徴

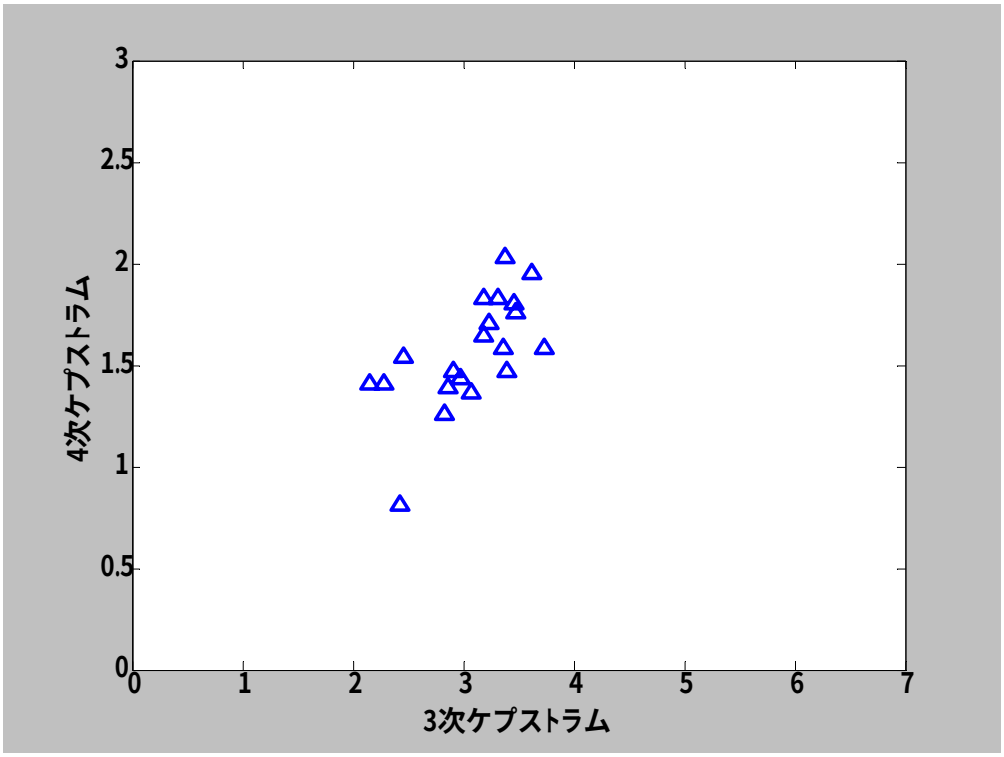

図 26：“ う ” の音声特徴

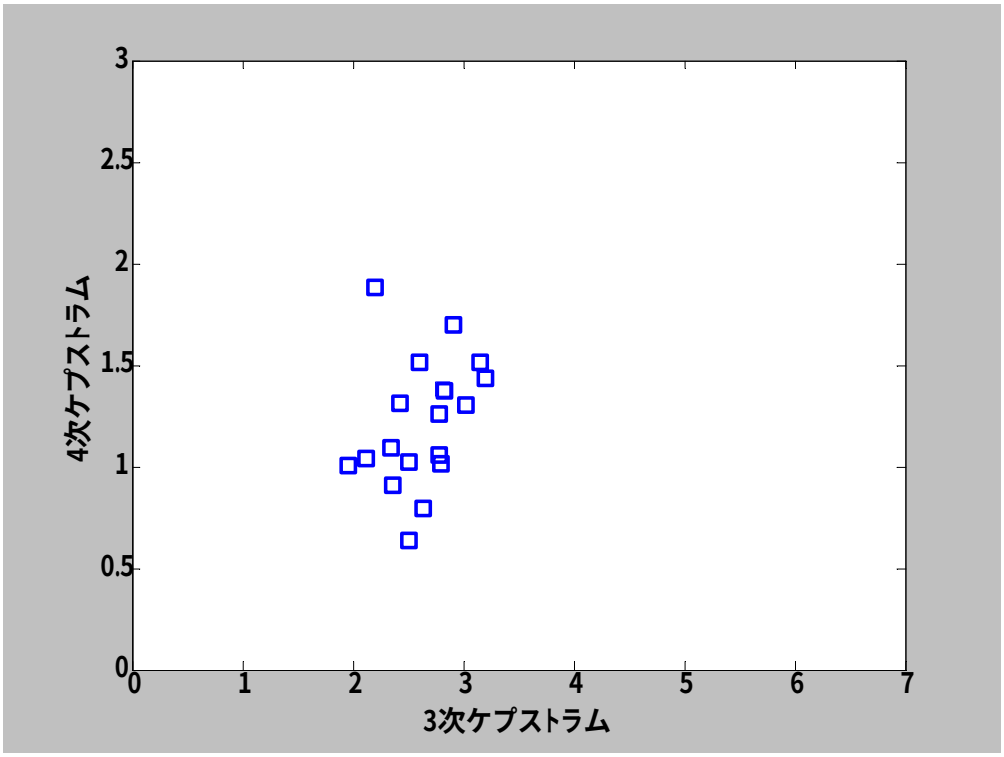

図 27：“ え ” の音声特徴

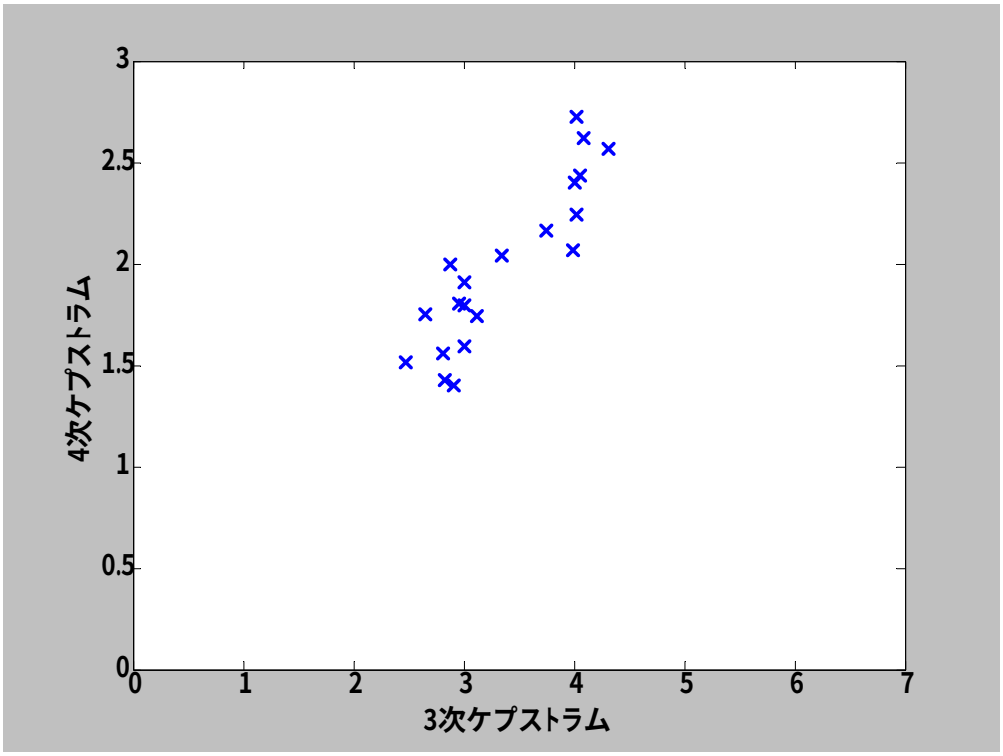

図 28：“ お ” の音声特徴

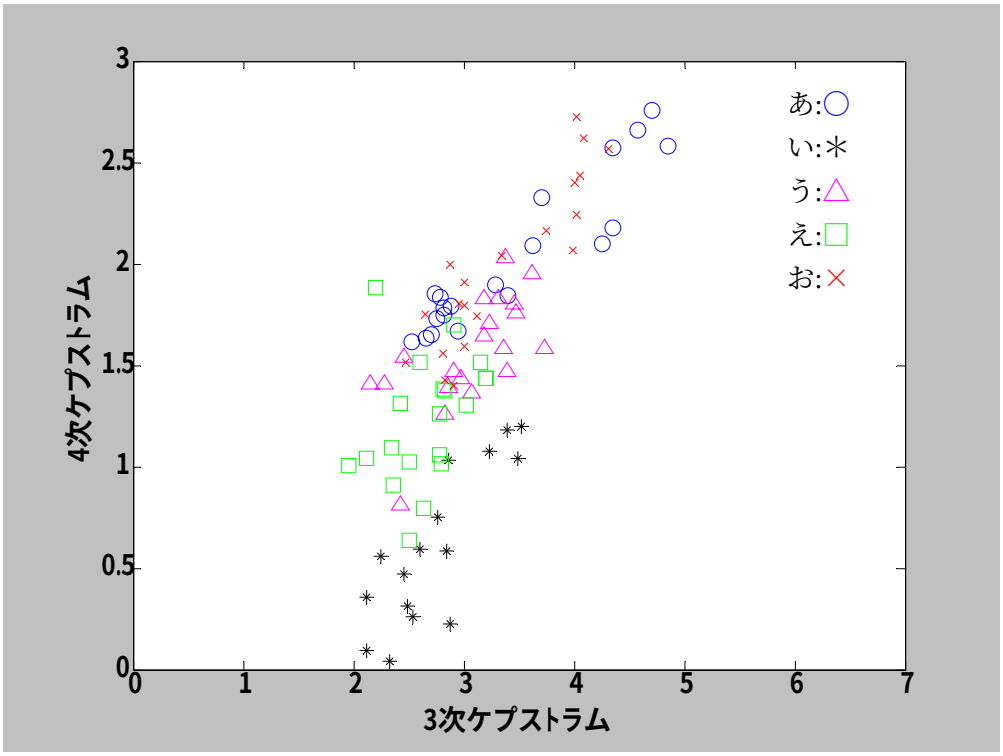

図 29：それぞれの音声特徴(3 次,4 次ケプストラムを使用)

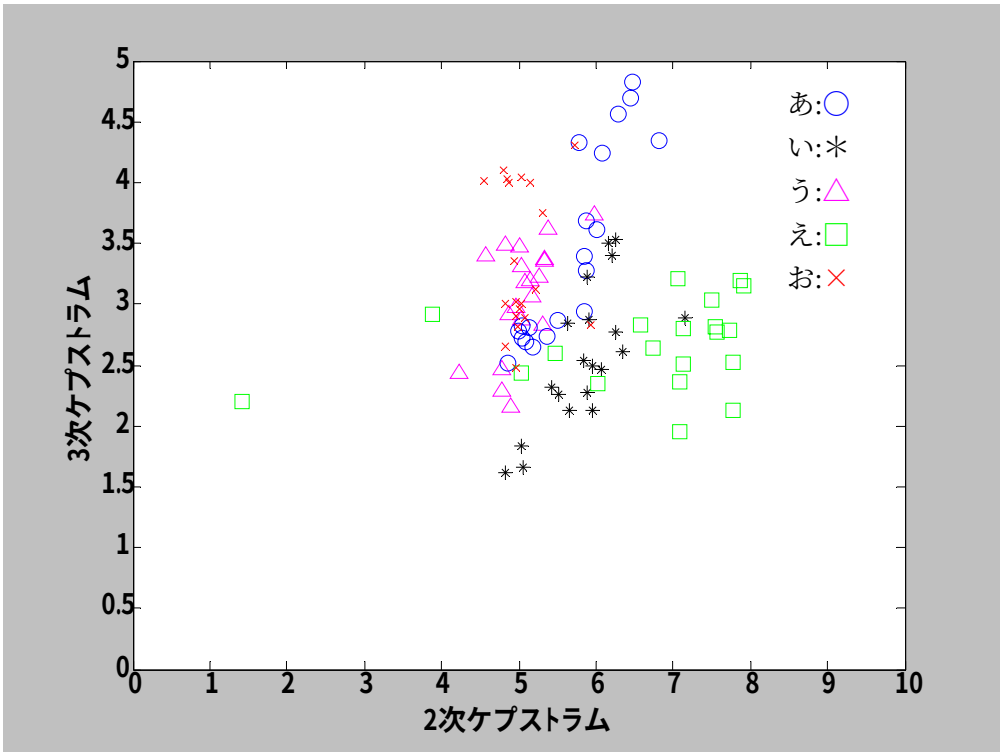

図 30：それぞれの音声特徴(2 次, 3 次ケプストラムを使用)

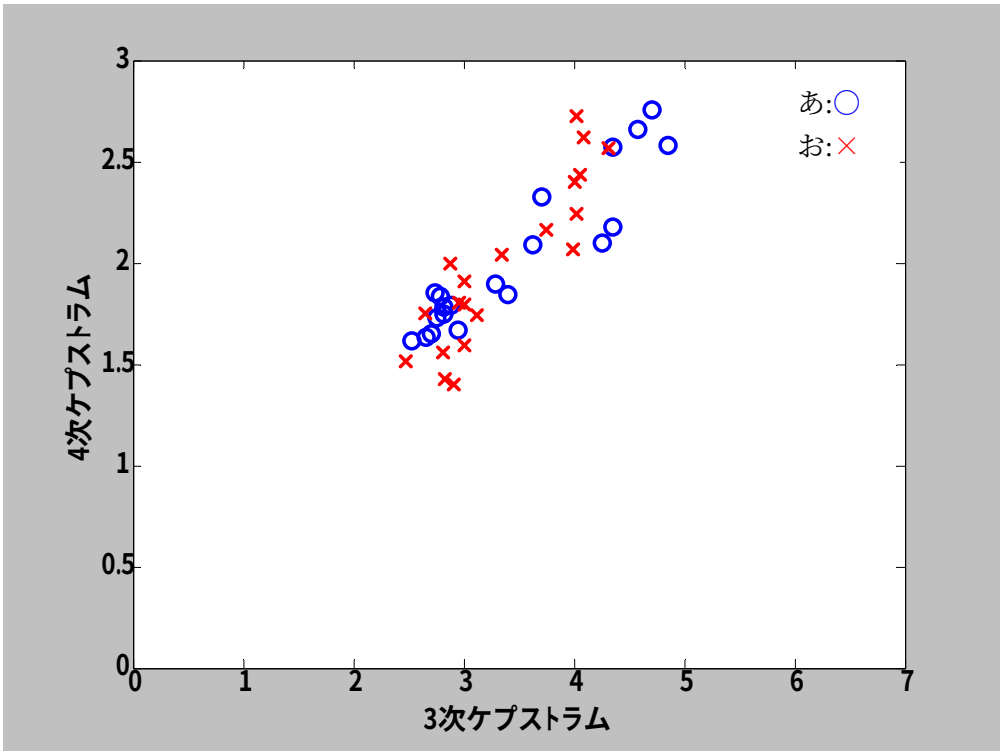

図 31 :“ あ・お ” の音声特徴

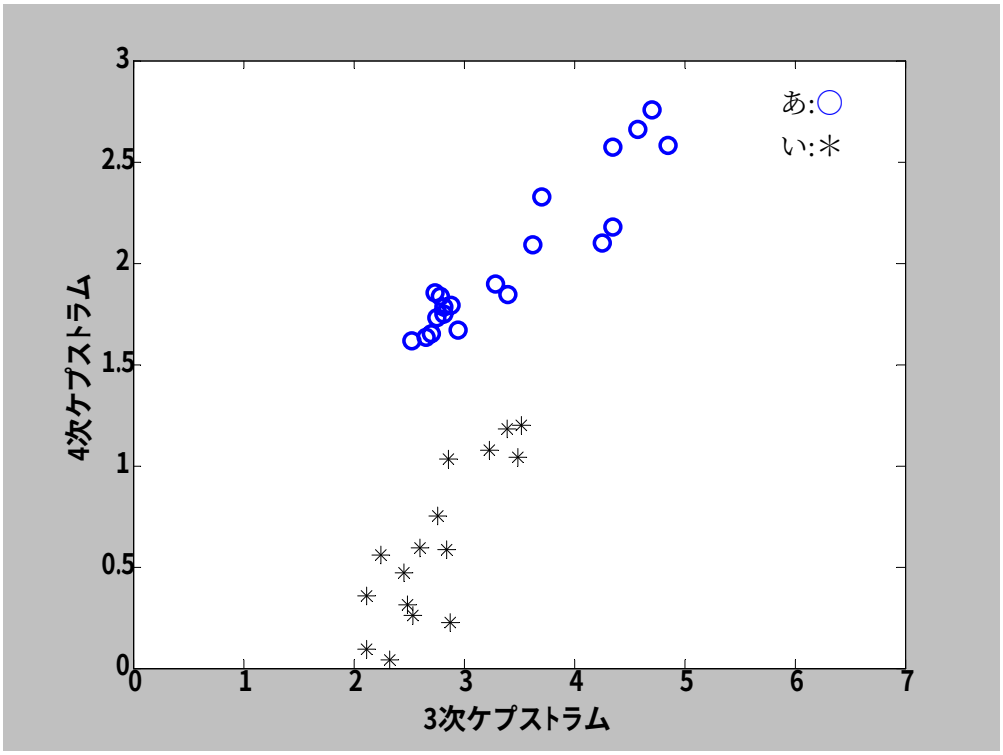

図 32 :“ あ・い ” の音声特徴

６．ロボットの動作制御の方法*1

音声認識判断を行い命令に従うロボットはパソコンで制御可能なＬＥＧＯロボットを使用した。使用したＬＥＧＯロボットの標準ファームウェア(3)を図に示す。

図 33：ＬＥＧＯロボットのプログラミング画面

このプログラミング画面は左上のビッグブロックでロボットの動作を指定することができる。その下にあるスモールブロックではモーターのパワー調節、光やタッチセンサー機能、Ｌｏｏｐやｉｆ文といったプログラムの流れを指定することができる。しかし肝心な音声センサーがなかった。この標準ファームウェアでは、32 個の変数と 10 個のタスク、そして８個のサブルーチンを利用できるようになっている。また、ＲＣＸ同士で１[Ｂｙｔｅ]のメッセージでやり取りすることも可能。

そこで標準ファームウェアを使用すると音声センサーがないので、一度音声をＰＣに取り込んでＭＡＴＬＡＢにおいて音源方向の角度を計算する。次に音声認識判断を行いそれからその角度や音声に対応した動きを考え、標準のファームウェアに手動で打ち込みロボットを動作させるという手間のかかる事になってしまった。

そこでロボットを動かすファームウェアも標準のものではなく、ＶｉｓｕａｌＢａｓｉｃ[4]やＬＥＧＯロボットのためのプログラミングソフトのＮＱＣ/ＲｃｘＣＣ[5]などを使用して、自分たちでファームウェアを作成していきたいと考えたがＶｉｓｕａｌＢａｓｉｃではＬＥＧＯがバージョンアップされていたので動作させることができなかった。ＮＱＣ/ＲｃｘＣＣにおいてもＳｐｉｒｉｔ．ｏｃｘ[6]というＲＣＸのコントロールやプログラムのダウンロードをするソフトウェアがなく動作させることができなかった。動作制御においては課題が残ってしまった。

(3)ファームウェア

ＬＥＧＯ ＭｉｎｄＳｔｏｒｍｓ「Ｒｏｂｏｔｉｃｓ Ｉｎｖｅｎｔｉｏｎ Ｓｙｓｔｅｍ（以下ＲＩＳ）」の標準的なプログラミング環境であるＲＣＸコードではパソコンで作ったプログラムをＲＣＸ（ロボットの頭脳）にダウンロードした後、私たちが[Ｒｕｎ]ボタンを押すと実行が開始される。このようなダウンロードされたプログラムを［Ｐｒｇｎ］ボタンが押されたら切り替えたり、[Ｒｕｎ]ボタンを押されたら実行したりというような処理を行うソフトウェアをＲＩＳでは「ファームウェア」と呼ぶ。

ファームウェアとは、パソコンのＯＳに似た基本的な機能を提供する為のソフトウェアのこと。ファームウェアは一般的にハードウェアとソフトウェアの中間的な位置に存在するソフトウェアで、マイコンを搭載している家電などで利用されている。例えば、エアコンでは温度か20[℃]になるように冷気を調節して部屋の中に送るといったような処理を行っている。

ただ家電などではほとんどの場合ファームウェアはＲＯＭに焼かれていて、私たちが簡単に交換できるようにはなってはいない。それにたいしてＲＣＸのファームウェアは私たちがダウンロードできるようになっているという大きな特徴がある。また、私たちが作成したＲＣＸコードを実行することができるという特徴もある。すなわち、ＲＣＸのファームウェアは、ＲＣＸコードを解釈しながら実行するインプリンタ（命令文を１つずつ解釈しながらプログラムを実行する方法）を搭載している。

(4) ＶｉｓｕａｌＢａｓｉｃでのプログラミング

ＲＩＳの標準のプログラミング環境（ＲＣＸコード）は、Ｓｐｉｒｉｔ．ｏｃｘというＡｃｔｉｖｅＸコントロールをしている。ＡｃｔｉｖｅＸコントロールとはいくつかの機能をまとめたプログラミング部品で、Ｓｐｉｒｉｔ．ｏｃｘにはＲＣＸをコントロールする機能がまとめられている。そして、ＶｉｓｕａｌＢａｓｉｃなどのＡｃｔｉｖｅＸコントロールに対応したプログラミング言語からこれを利用すれば、ハードウェアに関する知識がほとんどなくてもＲＣＸをコントロールプログラムが作れるようになっている。つまりＳｐｉｒｉｔ．ｏｃｘはＲＣＸを操作するためのもっとも重要な部分だといえる。

ＶｉｓｕａｌＢａｓｉｃを使ってＲＣＸをプログラムすることのメリットは、標準ファームウエアのもっている能力を使いきれるというメリットもあるが、パソコン周辺機器のようにＲＣＸをコントロールできる点に１番の魅力を感じる。例えば、パソコンの画面からマウスを使ってモーターを on/off にしたり、ＲＣＸの接続されたセンサーの値を読んでパソコンでいろいろな処理をしながらＲＣＸに命令を送るような場合、ＶｉｓｕａｌＢａｓｉｃを使ったプログラミングは非常に有効だといえるだろう。

(5) ＮＱＣ/ＲｃｘＣＣ

ＮＱＣとは「Ｎｏｔ Ｑｕｉｔｅ Ｃ：完全なＣではない」ということで、プログラミング言語Ｃに類似した構文をもつ簡単な言語である。ＮＱＣはＬＥＧＯ社とはまったく無関係に開発されていて、どんな形でもＬＥＧＯ社には属さない。伝統的なプログラミング方法、つまりテキストファイルとして記述し、コンパイルしＲＣＸにダウンロードして使う。グラフィックツールのように簡単には始められないが、プログラミングに親しんでいる人にとっては手ごろであり文字を使った言語はとても効率のよい開発ツールと言える。変数は 32 個、サブルーチンは 10 個、タスクは８個定義することができ標準ファームウェアの性能をさいだいげんに引き出すことができる。ＮＱＣの利点の１つは速度である。いくつかある他のツールは最速のコンピュータハードウェアでも遅くて不便である。ＮＱＣは旧式のハードウェアでもコンパイルできるうえ、高速にプログラムのダウンロードができる。Ｗｉｎｄｏｗｓ、ＭａｃＯＳ、Ｌｉｎｕｘの各環境用にプログラムが用意されている。短所としては学習が少し難しいところだろう。

ＲｃｘＣＣとは「Ｒｃｘ Ｃｏｍｍａｎｄ Ｃｅｎｔｅｒ」の略。これはＮＱＣ向けの総合開発環境で、リアルタイムにＲＣＸをコントロールすることもできる。こちらはＷｉｎｄｏｗｓ用のみとなる。

(6) Ｓｐｉｒｉｔ．ｏｃｘ

Ｓｐｉｒｉｔ．ｏｃｘは開発ツールではない。ＶｉｓｕａｌＢａｓｉｃなどの他のプログラミング言語で使用するＡｃｔｉｖｅＸコントロールで、ＲＣＸのコントロールやプログラムのダウンロードをするものである。Ｓｐｉｒｉｔ．ｏｃｘはＲＩＳのソフトウェアで Ｗｉｎｄｏｗｓのみで動作する。Ｓｐｉｒｉｔ．ｏｃｘのドキュメントはＬＥＧＯ社からＭｉｎｄＳｔｏｒｍｓ用のソフトウェア開発キッド（ＳＤＫ）として入手ができる。ＶｉｓｕａｌＢａｓｉｃでＲＣＸのプログラミングができるのは素晴らしいが、Ｓｐｉｒｉｔ．ｏｃｘは完全ではない。Ｓｐｉｒｉｔ．ｏｃｘはＶｉｓｕａｌＢａｓｉｃのプログラミングが働いているホストコンピュータからＲＣＸへの問い合わせと制御を行う。また、ＶｉｓｕａｌＢａｓｉｃプログラムでＲＣＸのプログラムをダウンロードすることができる。しかし、作成されたＲＣＸプログラムＳｐｉｒｉｔ．ｏｃｘの提供する分だけしか使えないという制約がある。ＶｉｓｕａｌＢａｓｉｃで使い慣れた関数や機能は実際にＲＣＸプログラムには使用できない。

まとめ

本研究ではロボットの聴覚機能実現を目的とした研究の第１歩として音源方法の検出、音声認識判断、ロボットの動作制御の検討をおこなった。

音源の方向検出においては人間の感覚と同じように正面付近はほぼ正確に方向を検出する事ができたが、角度が開いていくにつれ検出の結果に誤差が大きく出ることが多かった。つまり、音源が真横に近づくにつれて方向検出が不正確になった。これは人間にも同様のことが言える。また、発声する音の高低やその声の質、そしてマイクロホンを置く場所によってその部屋のノイズや反射波等による影響のために検出する音源方向に大きな変化があった。

音声認識判断においては、はっきりと判別可能な音声もあったが“ あ ”や“ お ”といった発声するときの口の開き具合や、口の中の形状が似ている音声では多くのプロットが重なり、音声特徴が似たものを持っていたため判別する事が難しかった。今回の研究では単音（母音）のみの音声認識判断の検討だけだったが、単語からちょっとした文章を認識し、判断していくことができるようになればロボットの聴覚機能実現に向けて大きく前進することだろう。

ロボットの動作制御においてはパソコンで制御可能な市販のＬＥＧＯ ＭｉｎｄＳｔｏｒｍｓを購入し作動させることを目的とした。ＬＥＧＯ ＭｉｎｄＳｔｏｒｍｓには音声センサーが搭載されていなかったので標準ファームウェアを使用せずにＶｉｓｕａｌＢａｓｉｃやNQC/ＲｃｘＣＣなどを使用して、自分たちでファームウェアを作成していきたいと考えた。しかし、Ｖｉｓｕａｌ ＢａｓｉｃではＬＥＧＯがバージョンアップされていたので動作させることができずNQC/ＲｃｘＣＣにおいてもＳｐｉｒｉｔ．ｏｃｘというＲＣＸのコントロールやプログラムのダウンロードをするソフトウェアがなく動作させることができなかった。残念ながら動作制御においては多くの課題が残ってしまった。

本研究においては結果を出せなかった研究もいくつかあったが現実的にロボットの聴覚機能実現のためにはこのほかにまだ数多くの研究が必要になる。それらに対応できるようにしていくことがロボットの聴覚機能実現につながっていくことだろう。

謝辞

本研究を進めるにあたりご指導下さいました金田豊教授ならびに、同研究室の研究生の方々に深く感謝申し上げます。

参考文献

＊1

（1）Jin Sato・白川裕記・牧瀬哲郎・倉林大輔・衛藤仁郎,
“LEGO MINDSTORMS ﾊﾟｰﾌｪｸﾄｶﾞｲﾄﾞ”
pp.79～91, 翔泳社, 1999.
http://www.legomindstorms.com/sdk/

（2）Davit Baum
“LEGO MINDSTORMS ﾗｰﾆﾝｸﾞ&ﾌﾟﾛｸﾞﾗﾐﾝｸﾞｶﾞｲﾄﾞ”
pp.38～, ｼｭﾌﾟﾘﾝｶﾞｰ・ﾌｪｱﾗｰｸ, 2001.