# お片付けロボット

大西有佳里　酒井彩美　西田茉佑子　疋田理沙

## １．緒言

近年、私たちの生活の中でロボットというものが身近になりつつある。そこで、私たちは自分たちの手でロボットの製作、プログラミングを試みようと考えた。一概にロボットといえども様々なものがあるが、その中でも私たちは生活の補助となるようなものを製作しようと考えた。そこで、現在販売されているお掃除ロボット「ルンバ」のように、床の掃除をするロボットを製作することを目標とし、研究を開始した。

私たちはロボットを製作するにあたって、レゴ社のプログラミングロボット教材であるレゴマインドストームを用いて製作を進めた。ロボット本体は組み立てブロック玩具であるレゴブロックを用いてロボットをつくり、モーターを乗せた教材付属のブロックを繋ぐことでロボットを動かせることができる。研究では、マインドストームの第一世代であるレゴマインドストーム education RCX と次世代の NXT を用いた。

## ２．方法

2－1. レゴマインドストーム education RCX による製作

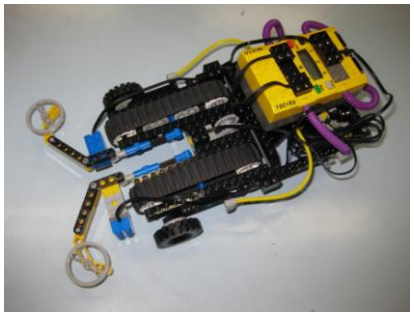


図1.収集ロボット

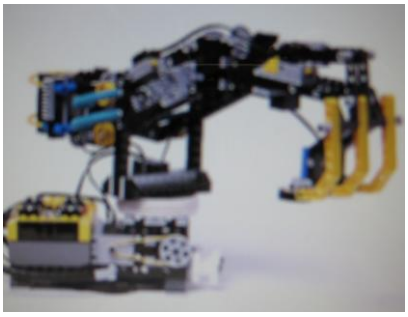


図2.回収ロボット

マインドストームの第一世代である RCX は外部通信ができる IR ポートを持つ。専用の赤外線インタフェースを通して、ロボットと外部パソコンを赤外線通信で繋ぐことができる。ロボットを動かすために作成したプログラムをパソコンから送り、ロボットに内蔵された RAM に登録することができる。

今回の実験では、物体の接触を感知できるタッチセンサー(図3)と明るさを識別できるライトセンサー(図4)との2つを用いた。

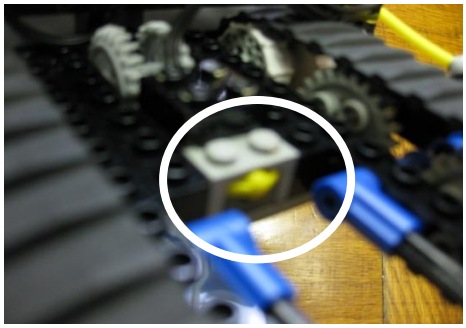


図3.タッチセンサー

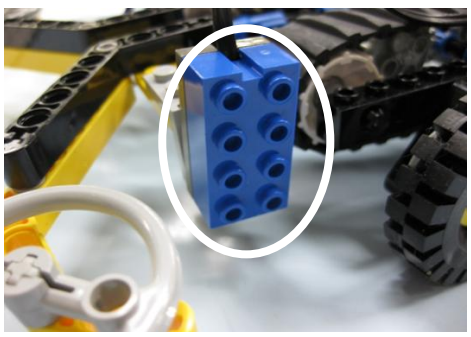


図4.ライトセンサー

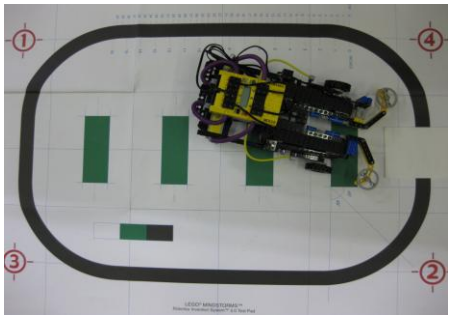



図5.黒い枠線

白い紙に黒い線を引いて枠(図5)を作った。その枠線に囲まれた範囲内で「お片付け」する対象となる物体を認識、収集し、回収箱へ片づけるという一連の動作をするロボットの製作を試みた。しかし、1台のロボットではこの作業をすることが困難であったため、2台のロボットに分担して行わせた。

・収集ロボット

範囲内を動き、対象物を認識し運ぶ役割を担うロボットである。

ロボットが範囲内から出ないようにするために、ライトセンサーⅠを(図4)のように下向きに設置し地面の明度を認知させ、黒い線を認識したときに斜め後ろに後退するようにプログラミングした。対象物が前面のバーの部分に入れば、バーの奥に取り付けたライトセンサーⅡが対象物を感知し、ベルトが回転して持ち上げる。その後、ライトセンサーⅠにより範囲の枠の線をたどって前進し、黒い線の切れ目に到達すると地面の色が白くなったことを認識して停止する。ベルトが逆向きに回転し、対象物を降ろす。

・回収ロボット

収集ロボットによって運ばれた対象物を認識し、回収箱へ入れる役割を担うロボットである。

アームの先にライトセンサーを取り付け、対象物が置かれていることを感知できるようにした。下に対象物が置かれるとライトセンサーが反応し、アームが物体を持ち上げ180度回転して、後方に置かれた回収箱に入れる。(図6)

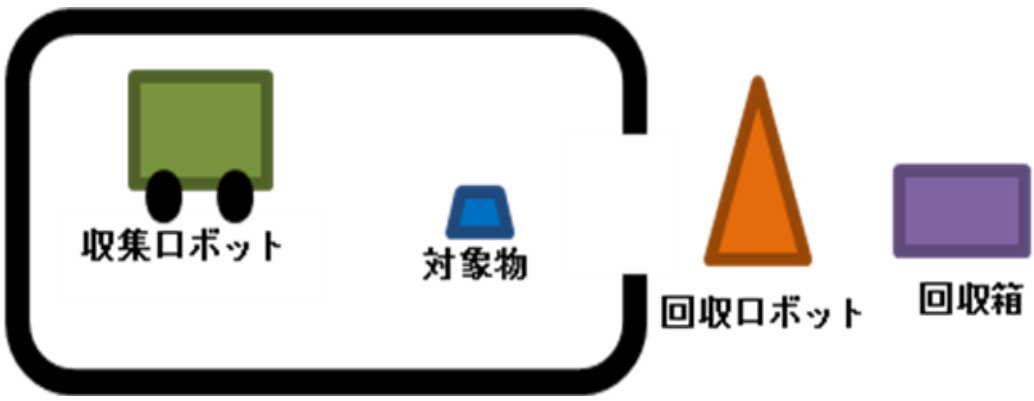



図6.実験手順

レゴマインドストーム education RCX では2台に片づけを分業していたが、これらを1台で行うために代わりにレゴマインドストーム education NXT を用いた。

２－２．レゴマインドストーム education NXT による実験

図7.NXT で作成したロボット

マインドストームの第二世代であるeducation　NXTは、USBケーブルを用いてロボットとパソコンを接続し、ロボット内部のメモリーにプログラミングデータを保存することができる。NXT　ではライトセンサーとタッチセンサーに加えて、新たに超音波センサーが導入された。超音波センサーは対象物との距離を測ることができる。

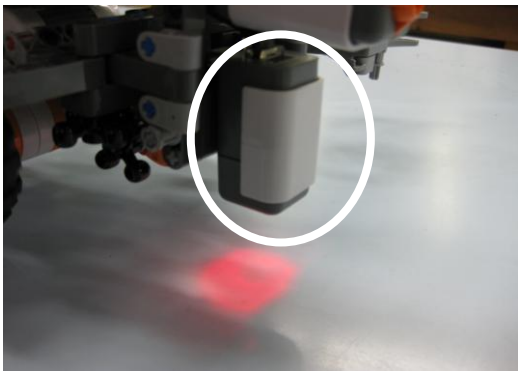


図8.ライトセンサー

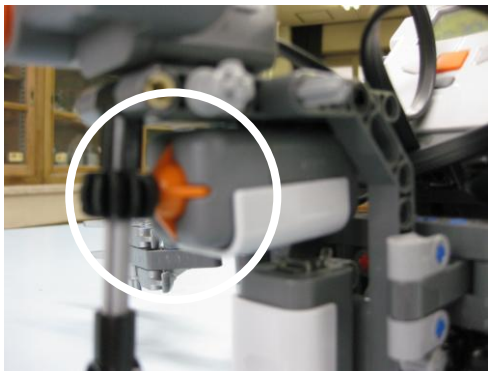


図9.タッチセンサー

図10.超音波センサー

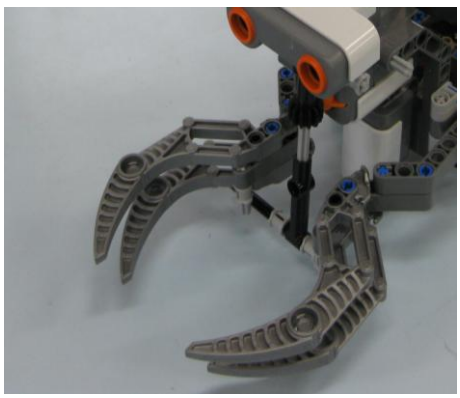


図11.アーム

これらのセンサーを用いて以下の手順で製作を進めた。

今回は段ボール紙で囲いを作った。ロボットと囲いとの距離が 15 ㎝以下になると超音波センサーが反応し、左斜め後ろに後退する。タッチセンサーが対象物を認識すると、アーム(図11)を閉じてつかみ、そのまま前進する。このとき、対象物をつかんだ後も壁に衝突しそうになれば斜め後ろに後退することができる。ライトセンサーで黒い線を認識した後、対象物を図12の ☆ に落とす。その後まっすぐ後退し、始めの操作に戻る。また、この動作を繰り返すことで、いくつでも対象物を拾うことができる。

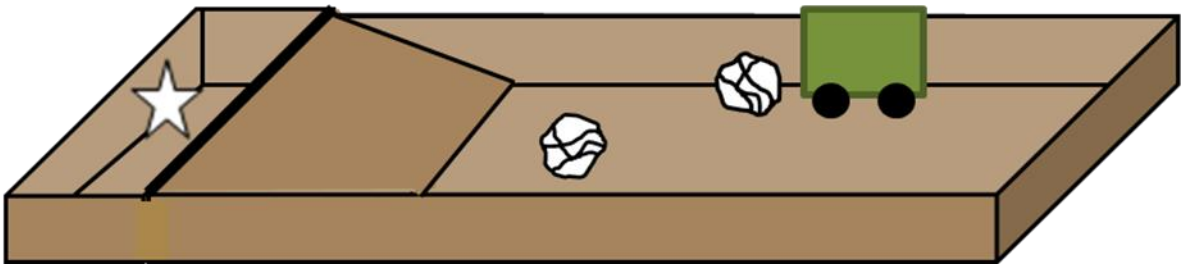


図12.実験手順

## ３．考察

３－1.レゴマインドストーム education RCX

収集ロボットはライトセンサーⅠで地面の明るさを感知し、決められた範囲内を移動した。続いて、ライトセンサーⅡで対象物の存在を確認・識別し、持ち上げた。その後タッチセンサーが対象物を持ち上げたことを確認し、そのまま運ぶことができた。しかし曲線周辺では予想したように進まず、回収ロボットの所まで行くことができなかった。 その理由として、次の事柄が挙げられる。

・ロボットが特定の線の上をたどる方法として、2つのライトセンサーで線を挟むようにすることが理想的であるが、今回は地面の色を認識するためのライトセンサーを1つしか用いなかったこと

・プログラミングにおいて変数をうまく利用できなかったこと

３－2.レゴマインドストーム education NXT

超音波センサーが壁に近づいたことを認識した後、左斜め後ろに後退するようにプログラミングをした。そのため一度斜面(図13)に入り損ねると、ロボットは範囲内を遊走し、再び斜面に入るまで時間がかかってしまった。したがって、ターンバックする方向をランダムにする必要があったと考えられる。

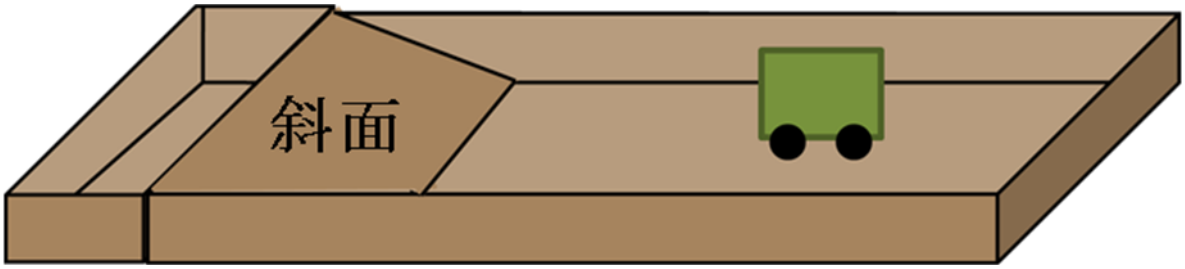



図13. 斜面

## ４．結論

目標にしていた市販のようなお掃除ロボットではないが、対象物を片付けるという面において正しく動くロボットの作成に成功した。

# ゲームソフトプログラミング

藤村優志　上松悠人　江澤友基　南上和也　小島滉介　中鉢大介

## １．緒言

私たちは、ゲームソフトはどのような仕組みで動いているのかということに興味をもった。そしてそれを知るために、C 言語（プログラミングに使用する言語）を使って、シューティングゲーム（STG）のプログラミングをすることにした。作成するにあたって使用したコンパイラ（プログラムを実行するためのソフト）は、”Microsoft Visual C++ ２００８”で、コンピュータゲーム作成用の DX ライブラリを利用した。今回作成する STG は、自機を操作して敵機を撃ち落とすことができるものを目標とした。

## ２．方法

１．はじめに、STG を組成する自機と敵機を作成する。

①プレイヤーが操作可能な座標を設定し、その座標に当たり判定をつけて自機とする。

②決められたパターンにしたがって自動的に動く座標を設定し、その座標を敵機とする。

２．　次に、STG には必要不可欠であるビーム能力をそれぞれの機体につける。

①自機､敵機の座標に自動的にビームの初期位置の座標を設定し､そこからビームを動かす。

②ビームを出す際にビームが重ならないようにビームを発射する間隔を設定する。

３．　次に、ゲームの流れをつくるために、体力やスコアを設定する。

①自機の座標が敵のビームか敵機の座標と重なった時、体力を減らし、０になるとゲームオーバーとする。

②敵機の座標が自機からのビームの座標と重なった時、スコアを増加させる。

４．　最後に、ゲームをよりおもしろいものとするために、ボスステージを作成する。

①スコアが一定に達した時に、ボス戦へと進ませるためにボス戦を開始させる関数を呼び出す。

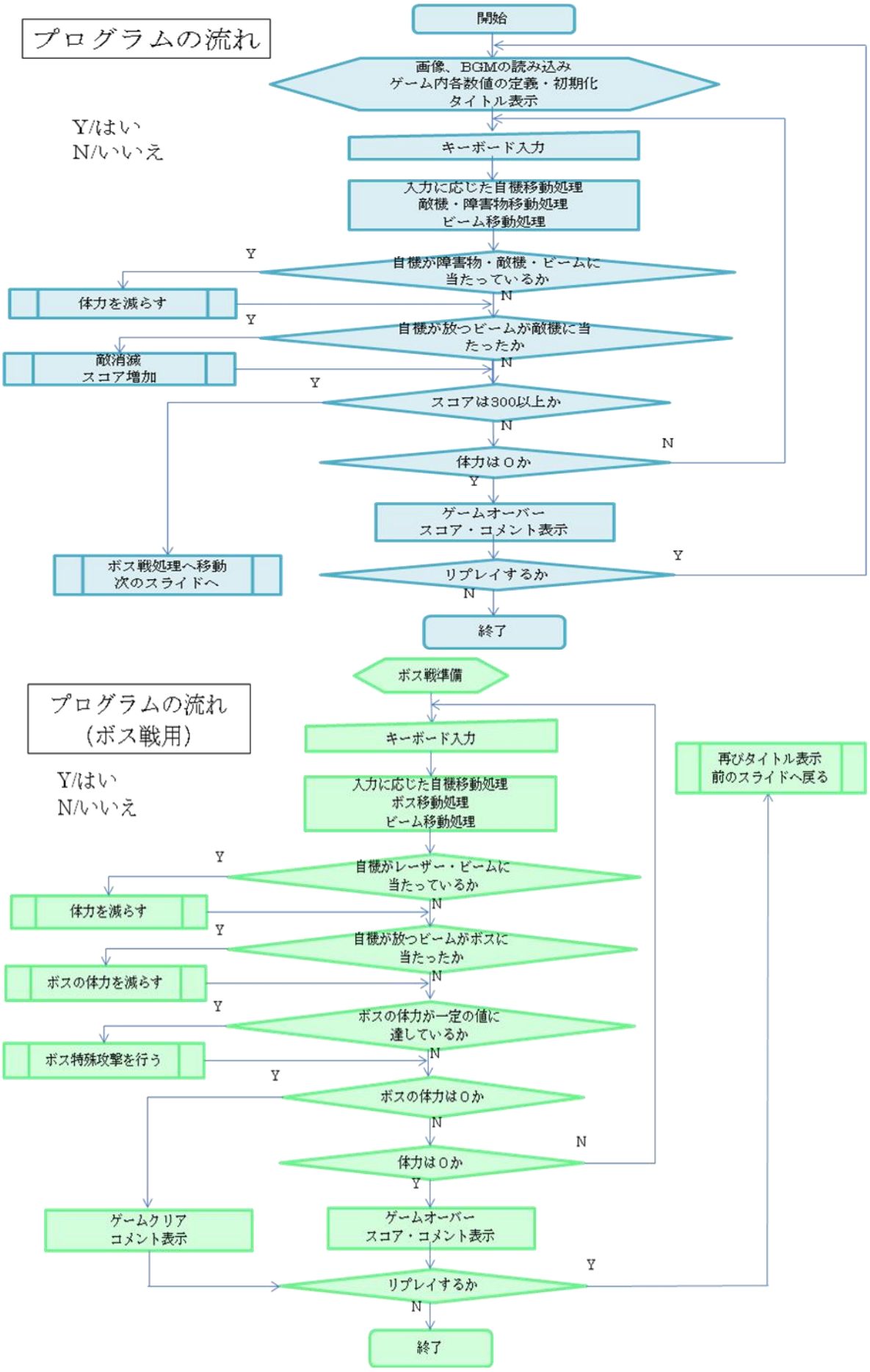
②ボスがそれまでの敵と異なる攻撃方法を行うようにするために、ビームの量や動き方を変える。

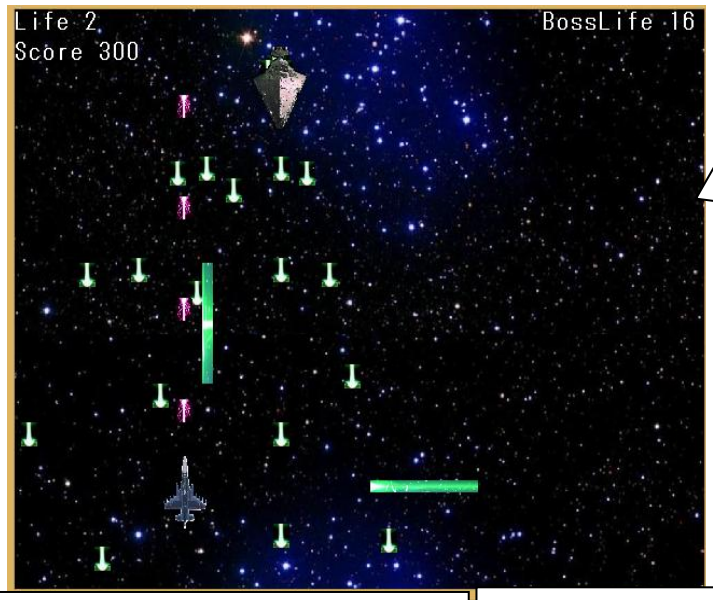
③一定の体力になると、攻撃パターン(ビームの量、動き方)を少し変えたり、特殊攻撃を行うようにする。

④ボスの体力が０になるとゲームクリアとする。

## ３．結果

ゲームを完成させることに成功した。


プログラムの流れ
Y/はい
N/いいえ
開始
画像、BGMの読み込み
ゲーム内各数値の定義・初期化
タイトル表示
キーボード入力
入力に応じた自機移動処理
敵機・障害物移動処理
ビーム移動処理
自機が障害物・敵機・ビームに当たっているか
体力を減らす
自機が放つビームが敵機に当たったか
敵消滅
スコア増加
スコアは300以上か
体力は０か
ゲームオーバー
スコア・コメント表示
ボス戦処理へ移動
次のスライドへ
リプレイするか
終了
プログラムの流れ
（ボス戦用）
Y/はい
N/いいえ
ボス戦準備
キーボード入力
入力に応じた自機移動処理
ボス移動処理
ビーム移動処理
自機がレーザー・ビームに当たっているか
体力を減らす
自機が放つビームがボスに当たったか
ボスの体力を減らす
ボスの体力が一定の値に達しているか
ボス特殊攻撃を行う
ボスの体力は０か
体力は０か
ゲームクリア
コメント表示
ゲームオーバー
スコア・コメント表示
リプレイするか
終了
再びタイトル表示
前のスライドへ戻る




実際のゲーム画面（ボス戦）の様子。自機を操作して相手にビームを当てて倒すことを目標としてゲームを進めていく。画面上には体力等を表示している。

図１：実際のゲーム画面

ゲーム画面を座標で表したもの
自機の領域とビームの領域の一部が一致したときに体力が減る。敵機についても同様の処理を行う。移動についても座標のｘ。ｙの数値を利用して処理を行う。

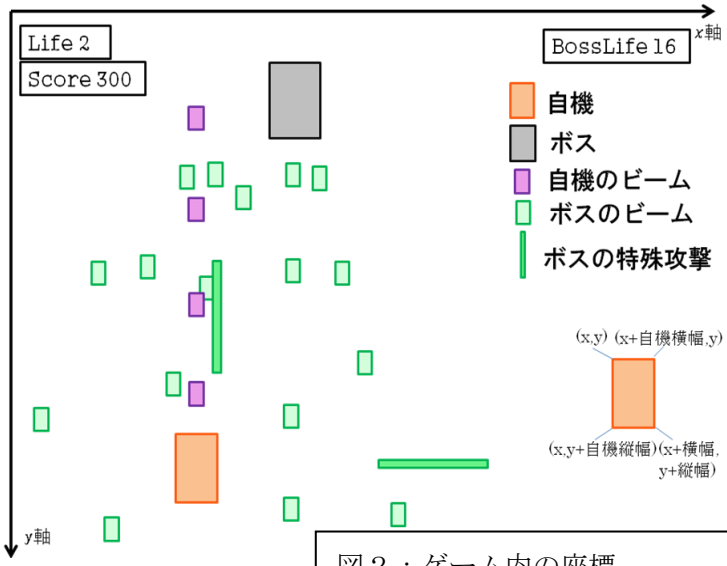



図２：ゲーム内の座標

## ４．考察

ボス戦を追加することで、最初はただ敵を撃ち落とすことしかできなかったため目標がはっきりせず、全然ゲームらしさのないものだったが、「ボス戦までたどり着いてボスを倒す」という目標を作ったことで、よりゲームらしさの増した面白いゲームにすることができた。課題としては、ボスの動き方やビームが単調なものとなってしまったことがあげられる。動き方に関しては、複雑化するとゲームをクリアすることがほぼ不可能、あるいは非常に難しくなってしまうので、複雑化することができなかった。ビームに関しても似たようなことが言えるが、自分たちがプログラミングできる範囲内で、かつバランスのとれた難易度になるビームのうたせ方が見つからなかったため、単調なビームを打たせることしかできなかった。

## ５．結論

多数の障害物や敵機、自機の同時移動、ボス専用ステージの作成をすることができた。また、攻撃方法を追加することによって、ゲームをより面白くすることができた。しかし、敵の動きが単調であったり、難易度がうまく調節できないなどの課題も残った。

# 渋滞シミュレーション

岡本直也、松井壮太、山本憲弘

## １．緒言

現在、交通渋滞は非常に深刻な社会問題のひとつとなっている。そこで私たちは渋滞現象を数理モデル化しシミュレーションすることで、交通渋滞を緩和させる方法を探ろうと試みた。

まず一本道における車の基本的な振舞いを、「セルオートマトン」という情報数学的理論を用いて表現できるようにした。そしてそれを用いてより発展的な試みとして、信号待ちによる渋滞が発生している道路に迂回路を付け加え、一定の割合で迂回させることで全体の交通流をスムーズにさせるような道路網のモデル化に取り組んだ。そしてそのモデルをエクセル VBA を用いてシミュレーションし、迂回させる割合と交通流のスムーズさとの相関を調べた。

## ２．方法

1. セルオートマトンを用いて車の動きを数理モデル化

図①はセルオートマトンという情報数学的理論を用いてモデル化した車の流れの様子である。横軸が道路、縦軸が時間である。1と表記された部分に車が存在するとする。

時刻tの時、車のすぐ前に他の車が無いと、次の時刻 t＋1では車は一つ前に進むとした。

図②は進行上のルールである。例えば110⇒？0？という結果は、前が0であれば一つ前に進み、1であれば進まないという条件を定めることによって決定されたルールであり、どのセルもその前の時刻の前後を含む三つのセルから決定されるということを示している。同様のルールを数え上げると、上の三つのセルは0か1かを満たすことより、2の3乗＝8通りある。

次にセルオートマトンの応用の第一歩として、単純な一本道に信号を一台設置した道路のモデル化に取り組んだ。図③がそのモデルである。

図③において、14セル目と15セル目の間の部分に信号を置き、青信号が5ステップ、赤信号が4ステップ続くとする。時刻tで青信号に変わったとすると時刻t＋5まで時間経過とともに渋滞が緩和されていき、t＋6において信号が赤に変わることにより、それ以降渋滞が再度発生していく様子が見て取れる。

| 車の動き→ | (時間) |
|---|---|
| 1001101110101010 00 | t |
| 0101011101010101 00 | t+1 |
| 0010111010101010 10 | t+2 |

図①　セルオートマトン

| | |
|---|---|
| 100⇒？1？ | 110⇒？0？ |
| 010⇒？0？ | 101⇒？1？ |
| 001⇒？0？ | 011⇒？1？ |
| 000⇒？0？ | 111⇒？1？ |

図②　８通りのルール

| 車の動き→ | （時間） |
|---|---|
| 00000111111111000000000 0 | t |
| 00000111111110100000000 0 | t＋1 |
| 00000111111101010000000 0 | t＋2 |
| 00000111111010101000000 0 | t＋3 |
| 00000111110101010100000 0 | t＋4 |
| 00000111101010101010000 0 | t＋5 |
| 00000111010101010101000 0 | t＋6 |
| 00000110101011001010100 0 | t＋7 |
| 00000101010111000101010 0 | t＋8 |

図③　信号待ちの渋滞

さらに先ほどモデル化した信号待ちによる渋滞が発生するような道路に迂回路を付け加え、一定の割合で迂回させることで全体的な車の流れをスムーズにさせるような道路網のモデル化に取り組んだ。

以下はモデルの概要である。

1、分岐点までのセル数を52個とおく。直進する場合、ゴールまでのセル数は42個で25番目のセルに信号が存在するとする。分岐する場合、ゴールまでのセル数は52個で直進するより10セル分遠回りになるとする。迂回ルートの長さが直進ルートの約 5/4 倍になるよう留意した。

2、スタート地点において車を流入させる割合を確率1/2とし、乱数を用いてランダムなタイミングで車が流入するものとする。

3、分岐点における迂回ルートに進む分岐率を任意数Pとおき、直進する確率を1－Pとおく。

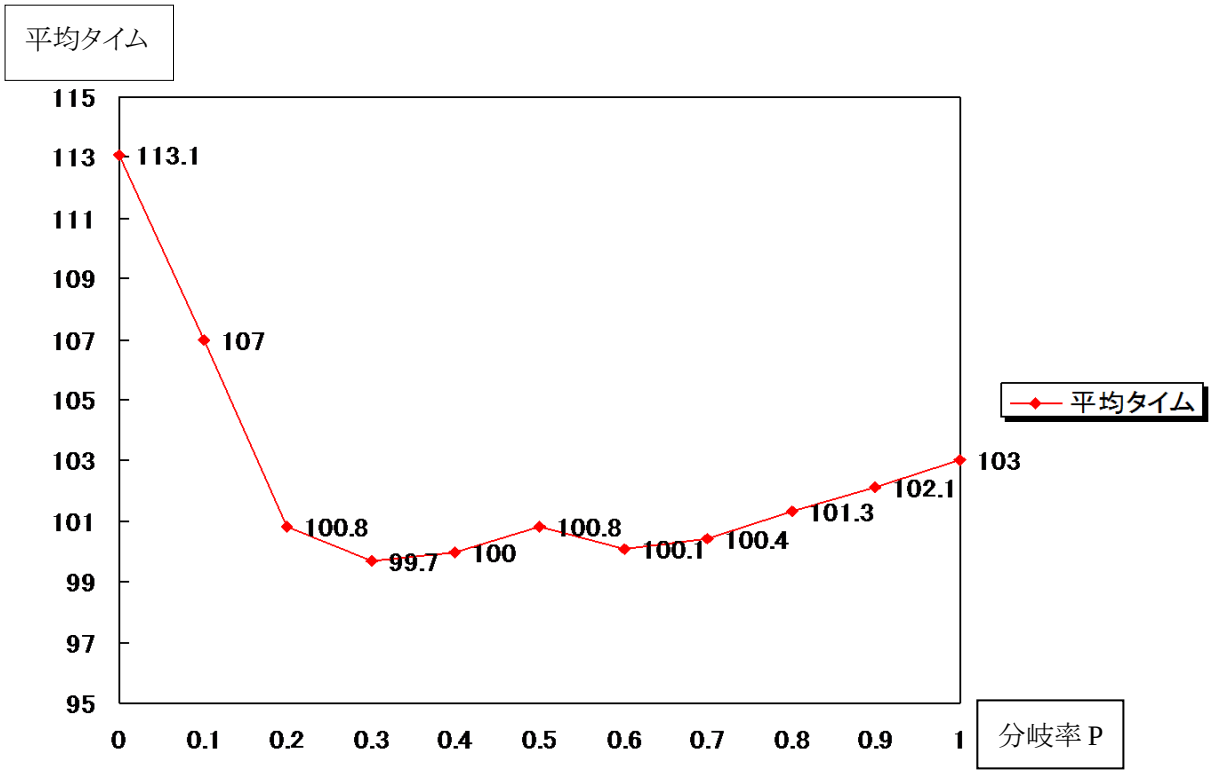
4、青信号を5ステップ、赤信号を4ステップとする。また、黄色信号はドライバーの意思が介入してしまうため考えないものとする。

以上のモデルをエクセルVBAシミュレーションできるようプログラミングする。なお、そのプログラムは末尾に添付する。

シミュレーション

分岐率Pに0から1まで0．1ずつ動かした値を与え、最初にスタートした車がゴールした時、モデル内に十分車がたまって渋滞現象が発生しているとして、そのときスタートした車に注目しゴールまでのタイムを調べる。この操作を各Pの値ごとに30回繰り返し、平均タイムをとる。
そして分岐率Pと平均タイムとの関係を調べていくことで最も渋滞の少ない分岐率を探る。

## ３．結果




図④　分岐率と平均タイムの相関図

## ４．考察

・分岐率0のとき車はすべて直進し迂回路をつけ加えない場合と同様にすべて信号に引っ掛かってしまい大きな渋滞が起こってしまうので、値が0に近づくほど平均タイムが飛びぬけて高くなっていると考えられる。

・分岐率1のとき車はすべて迂回し遠回りするので、値が1に近づくほど平均タイムは高くなっていると考えられる。

・分岐率0．5のとき私たちの予想に反して平均タイムが大きくなるのは、車の半数を直進させることにより信号待ちによる渋滞がある程度起こることに加えて、残る半数の迂回させた車が平均タイムを引き上げていることが理由として考えられる。

## ５．結論

本研究の主な成果は、今回扱ったルールのセルオートマトンのような「車の基本的な振舞いしか表現できない道具」を用いて信号と迂回路を加えたような発展的な道路網をモデル化し、さらに分岐率というパラメータと平均タイムの相関を調べることができたことである。

今後の拡張としては、流入条件、道のりの長さ、信号のステップ数のそれぞれをパラメータとし、平均タイムとの相関を調べていくことが挙げられる。

以下プログラムを添付

```
Sub 渋滞シミュレーション()

Do While Not ActiveCell.Value = 11
   ActiveCell.Offset(-1, 0).Activate
   一列
   Loop
ActiveCell.Offset(-300, 0).Activate
End Sub


Sub 一列()'

If ActiveCell.Value = 0 Then
   ActiveCell.Offset(1, 0).Activate
   流入
   Else: ActiveCell.Offset(0, 1).Activate
    If ActiveCell.Value = 0 Then
    ActiveCell.Offset(1, -1).Activate
     流入
     Else: ActiveCell.Offset(1, -1).Activate
       ActiveCell.Value = 1
        ActiveCell.Offset(-1, 0).Activate
        End If
End If
分岐前
分岐
ルートA 信号前
信号
ルートA 信号後
ルートB
End Sub


Sub 流入()

ActiveCell.Value = "=RAND()"
If ActiveCell.Value < 0.5 Then
```

```
    ActiveCell.Value = 0
Else: ActiveCell.Value = 1
End If
ActiveCell.Offset(-1, 0).Activate
End Sub
```

```
Sub 分岐前()

Do While Not ActiveCell.Value = 5
    移動
    ActiveCell.Offset(-1, 0).Activate
Loop
ActiveCell.Offset(0, -1).Activate
End Sub
```

```
Sub 分岐()

If ActiveCell.Value = 0 Then
    ActiveCell.Offset(1, 79).Activate
    ActiveCell.Value = 0
    ActiveCell.Offset(0, -77).Activate
    ActiveCell.Value = 0
    ActiveCell.Offset(-1, 0).Activate
Else: ActiveCell.Offset(0, 150).Activate
    ActiveCell.Value = "=RAND()"
    If ActiveCell.Value < 0.7 Then
      ActiveCell.Offset(0, -71).Activate
       If ActiveCell.Value = 0 Then
         ActiveCell.Offset(1, 0).Activate
         ActiveCell.Value = 1
         ActiveCell.Offset(0, -77).Activate
         ActiveCell.Value = 0
         ActiveCell.Offset(-1, 0).Activate
      Else: ActiveCell.Offset(1, 0).Activate
        ActiveCell.Value = 0
         ActiveCell.Offset(0, -77).Activate
         ActiveCell.Value = 0
         ActiveCell.Offset(0, -2).Activate
```

```
          ActiveCell.Value = 1
          ActiveCell.Offset(-1, 2).Activate
        End If
    Else: ActiveCell.Offset(1, -71).Activate
      ActiveCell.Value = 0
       ActiveCell.Offset(-1, -77).Activate
       If ActiveCell.Value = 0 Then
         ActiveCell.Offset(1, 0).Activate
         ActiveCell.Value = 1
         ActiveCell.Offset(-1, 0).Activate
       Else: ActiveCell.Offset(1, 0).Activate
         ActiveCell.Value = 0
          ActiveCell.Offset(0, -2).Activate
          ActiveCell.Value = 1
          ActiveCell.Offset(-1, 2).Activate
       End If
    End If
End If
End Sub


Sub ルートA信号前()

Do While Not ActiveCell.Value = 6
   移動
   ActiveCell.Offset(-1, 0).Activate
Loop
ActiveCell.Offset(0, -1).Activate
End Sub


Sub 信号()
'
If ActiveCell.Value = 0 Then
   ActiveCell.Offset(1, 3).Activate
   ActiveCell.Value = 0
   ActiveCell.Offset(-1, 0).Activate
Else: ActiveCell.Offset(0, 2).Activate
   If ActiveCell.Value = 8 Then
     ActiveCell.Offset(1, 1).Activate
```

```
        ActiveCell.Value = 0
        ActiveCell.Offset(0, -3).Activate
        ActiveCell.Value = 1
        ActiveCell.Offset(-1, 3).Activate
    Else: ActiveCell.Offset(0, 1).Activate
      If ActiveCell.Value = 0 Then
         ActiveCell.Offset(1, 0).Activate
          ActiveCell.Value = 1
          ActiveCell.Offset(-1, 0).Activate
       Else: ActiveCell.Offset(1, 0).Activate
         ActiveCell.Value = 0
          ActiveCell.Offset(0, -3).Activate
          ActiveCell.Value = 1
          ActiveCell.Offset(-1, 3).Activate
       End If
    End If
End If
End Sub


Sub ルートA信号後()

Do While Not ActiveCell.Value = 9
    移動
    ActiveCell.Offset(-1, 0).Activate
Loop
ActiveCell.Offset(0, 1).Activate
End Sub


Sub ルートB()

Do While Not ActiveCell.Value = 10
    移動
    ActiveCell.Offset(-1, 0).Activate
Loop
ActiveCell.Offset(2, -182).Activate
End Sub
```

```
Sub 移動()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(0, 1).Activate
    いち
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(0, 1).Activate
    ぜろ
Else
    ActiveCell.Offset(1, -51).Activate
End If
End Sub


Sub いち()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(0, 1).Activate
    いちいち
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(0, 1).Activate
    いちぜろ
Else
    ActiveCell.Offset(1, 0).Activate
End If
End Sub


Sub ぜろ()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(0, 1).Activate
    ぜろいち
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(0, 1).Activate
    ぜろぜろ
Else
    ActiveCell.Offset(1, 0).Activate
End If
End Sub
```

```
Sub いちいち()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 1
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 0
Else
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 0
End If
End Sub
```

```
Sub いちぜろ()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 1
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 1
Else
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 1
End If
End Sub
```

```
Sub ぜろいち()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 1
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 0
Else
    ActiveCell.Offset(1, -1).Activate
```

```
    ActiveCell.Value = 0
End If
End Sub


Sub ぜろぜろ()

If ActiveCell.Value = "1" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 0
ElseIf ActiveCell.Value = "0" Then
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 0
Else
    ActiveCell.Offset(1, -1).Activate
    ActiveCell.Value = 0
End If
End Sub
```

付録①　プログラム

## ６．参考文献

『図解雑学よくわかる渋滞学』
西成活裕著　ナツメ

## ７．謝辞

最終発表会にてご助言頂いた東京工業大学、赤池敏宏先生、大阪府教育センター、広瀬祐司先生、中川明子先生ありがとうございました。

# Dyson-羽根がないのに風が吹く！？

由良真悟　田中信悟　平土井悟　古橋直也　梅森靖史

## １．はじめに

ダイソンエアマルチプライヤーという扇風機がある。扇風機には羽根があるというのが普通であるが、ダイソン扇風機には羽根がないのが大きな特徴である。それによってむらの無い風を起こすことができ、子供が指を入れても危険がない、インテリアとして近未来的デザインである、掃除がしやすいなど様々なメリットがある。

中でも私達は「フレームの溝から出てくる気流が周囲の空気を巻き込むことで１５倍の風量となる」という部分に着目した。今回私達はダイソン扇風機の送風の仕組みを解明するために流体についての研究を行った。まず実験の際必須であるとされている風洞実験機を調べ、自作し、気流を可視化することに取り組んだ。そしてとくに興味をひかれた「１５倍の風量」という部分について仕組みに解明に取り組んだ。

ダイソン扇風機では普通の扇風機と違い羽根がないが、どこから風を送っているのかに注目すると土台上の（図１のように）丸いフレーム部内側のスリット（裂け目）から送風が始まる。図２よりフレームの溝から流れ出た気流は速い流れとあるので、私達はその速い気流が周りの空気を引き込むのではないかと考えた。

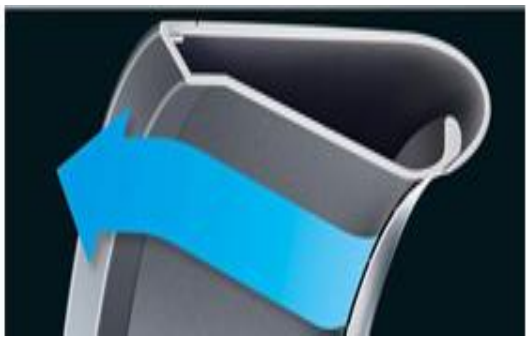


図 １　ダイソン扇風機のフレームの形状

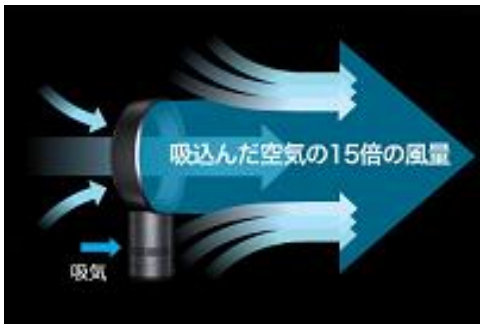
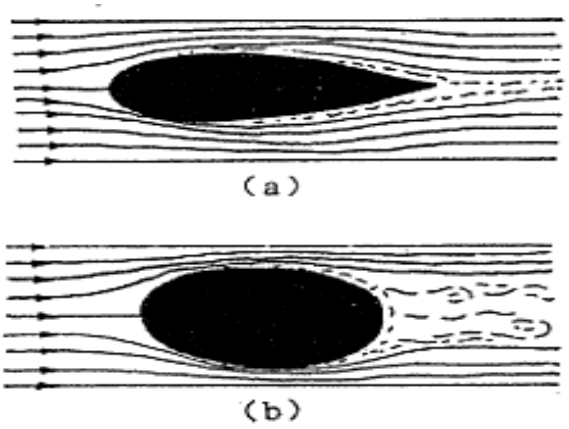



図 ２　ダイソン扇風機の風量のイメージ

次に、私達はフレームのスリットから吹き出した空気がどのように流れるかに着目し、スリットから吹き出す空気の流れ方に一番大きく影響を与えるのはダイソン扇風機内で流れている空気と最も接触しているフレームの形状であると考えた。そこで、異なる形状を持った数種類の物体を用意し、それらに気流を当てることで、フレームから吹き出した空気とフレームの形状との関係性を調べた。
気流は図３のように乱流と層流に大別することができる。




(a)層流（整った流れ）

(b)乱流（乱れた流れ）

**図　３　層流と乱流の違い**

層流の方は気流制御上効率がよく、ダイソン扇風機が送り出す風は層流であると考えられる。私達はどのようなフレームの形状なら層流を送り出すことができるかを研究することにした。

## ２．実験と実験結果

実験①　もとの送風された風が周りの空気を引き込む現象

フレームのすき間から出た風がまわりの空気を引き込むことを調べるため、図３のように次のような実験を行った。図４のように、まず気柱管に煙を溜め、地面に鉛直に立て、気柱管の上側に横から気流を当てた。ここで横から流した風をフレームのすき間から出た風とし、気柱の空気をフレームのまわりの空気とした。

※この実験では、煙を遅い気流、横からあてた気流を速い流れとして考える。

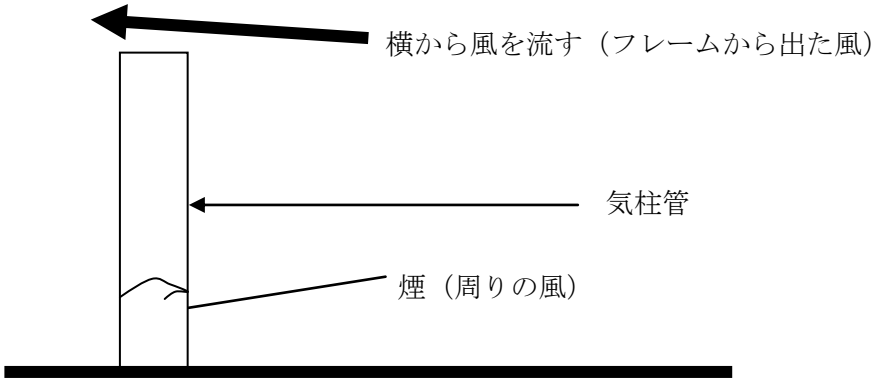



**図　４　実験①のイメージ**

実験①の結果

　図５の四角部分のように、気柱管下側から上側へ煙が移動し、また上から煙が流れ出している。このことから気柱管内部の煙が横から流した風に引っ張られるように流れたのが分かる。

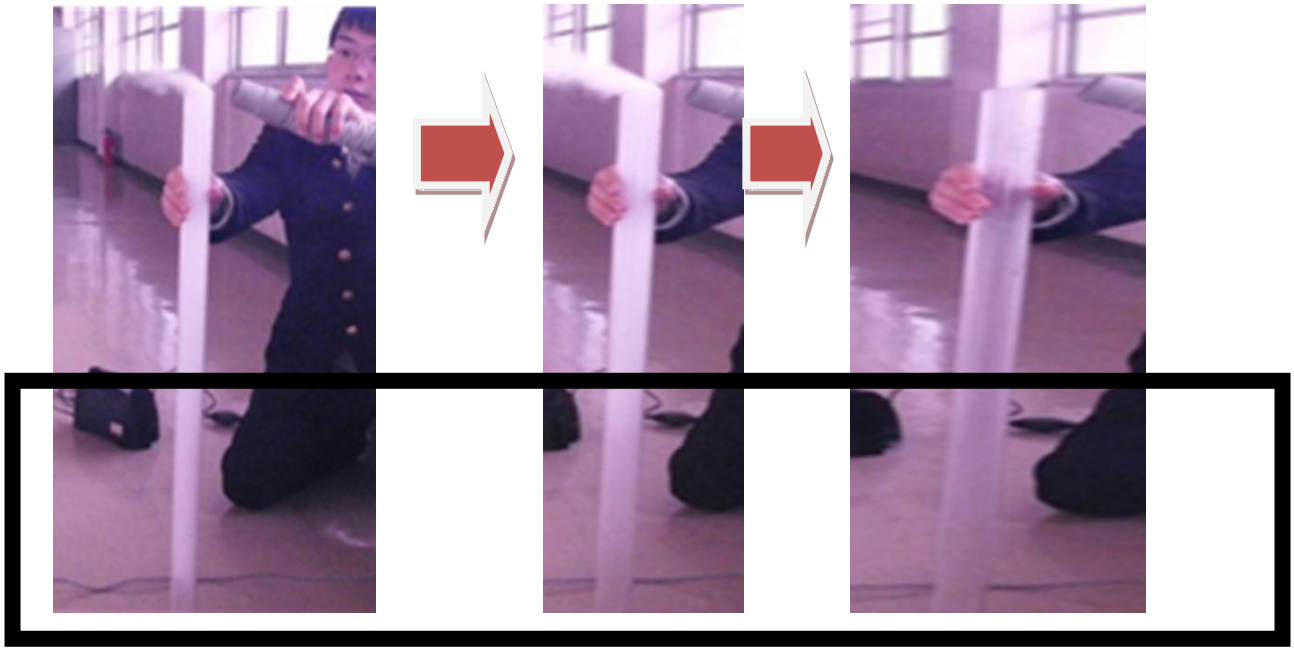


**図５　実験①の結果**

実験②　フレームの形状を考える実験

物体にぶつかった気流の様子を観察することで、ダイソン扇風機のフレームから出た空気がどのような気流を作るか研究した。

（Ⅰ）風洞の作製

気流の様子を観察するため風洞実験機を作製した。風洞実験機とは外部影響の少ない環境で気流の実験を行うための洞である。扇風機により送り込まれた乱れた気流が風洞の吹き込み口で整えられた状態にし、それ以降風洞内部では整った風が流れるようにする風洞を作製した。

扇風機からの風を整えるために吹き込み口にメッシュを取り付け、気流がまんべんなく吹き込むようにした。その後に「ハニカム」と呼ばれる六角形の筒を組み合わせた蜂の巣状のものを吹き込み口に設置した。これによって乱れ吹き込んだ風が細かく分けられ、ハニカム通過後は風が整った状態にすることができる。さらに吹き込み口から観測胴にかけ滑らかに絞りを作ることによって気流を細く整えられるようにした。風が整った後、風洞の「観察洞」と呼ばれる部分に入る。観察洞には、外から中の様子を知ることができるように片面にアクリル板を取り付けてある。

**図６　風洞実験機**

（Ⅱ）気流の様子の観察

ダイソン扇風機の気流制御はフレームに形状によってなされていると予想し、どのような形状が送風に適しているか調べるために、球、円柱、流線形の３種類にした。また、速さによる違いがないか確かめるのに扇風機の強弱で風速を変えた。

A、球の場合

**図７　球体を　気流中に置いたとき**

図７のように球にぶつかった空気は球にしばらく沿って、剥離した後乱れた。
また、風速を変えると、速いほうがより乱れる様子を観察できた。

B、円柱の場合

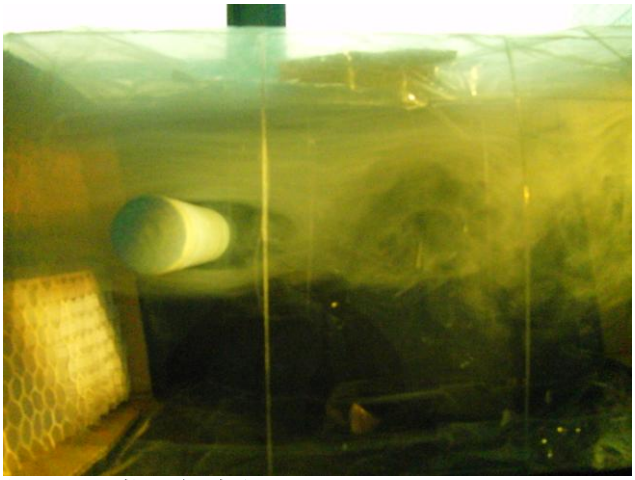


**図８　円柱を気流中に置いたとき**

円柱にぶつかった気流は、円柱から離れた後球と同じように乱れたが、この実験では円柱後方で列をなすように渦が出来、「渦列」が観察できた。

C、流線形の場合

流線形の物体にぶつかった気流は物体を離れることなく沿って流れて乱れが生じることはなかった。

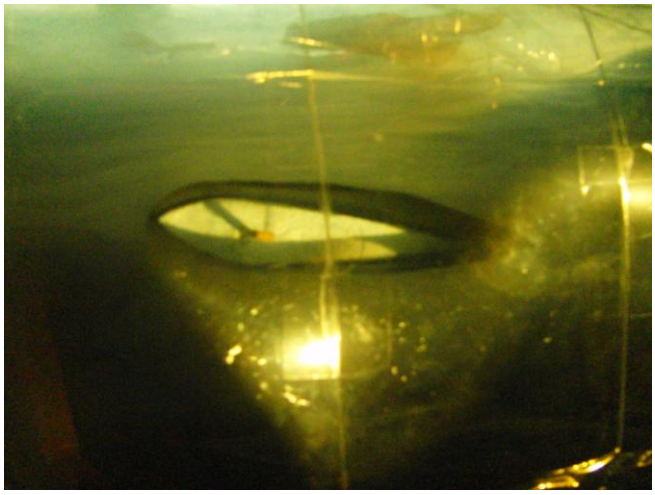


**図９　流線形の物体を気流中に置いたとき**

なお風速が遅いほど気流の乱れが少なくなり層流に近づいた。

## ３．考察

遅い煙は速い気流の方へ移動することが実験によって確かめられた。このことから、ダイソン扇風機は、フレームの周りから速い気流を流すことで後方の遅い気流を巻き込み、全体としてフレームから最初にから出た気流よりも大きな気流を発生させていると考えられる。

煙を流して気流を可視化することによる気流中に物体に置いた時の気流の変化を調べた。球体を気流中に置くと、煙は球に沿い、しばらくすると球から剥離して気流が後方で乱れた。円柱を気流中に置くと、球と同じように円柱に沿ってから剥離し、円柱後方で渦列が出来た。球の場合に渦列が見られなかったのは、球の周りに３次元的に渦列が発生し、それらが混じり合ってきれいな渦列に見えなかったからだと考えられる。流線型の物体を気流中に置くと、煙は最後まで物体に沿い、その後も乱れることがなかった。球の結果と合わせて、最後まで物体に沿ったかどうかが乱流発生にかかわるのではないかと考えられる。このことよりダイソン扇風機のフレームの形は断面が流線形になるように作られていると考えられる。

## ４．結論

以上の実験結果から、ダイソン扇風機は以下の２つの性質を利用しているといえる。

①フレームの隙間から流れのはやい風を送り出すことにより周りの空気を巻き込み、大き

な風量を得ている
②フレームの形状を流線型にすることにより、まっすぐに風を送り出している。
　これらによりダイソン扇風機はムラのない大きな風量をつくりだすことを可能にしたと考えられる。

## ５．参考文献

ダイソン公式ホームページ　http://www.dyson.co.jp/

## ６．謝辞

大阪市立大学の小原顕先生に多くの助言とコメントをいただきました。心より感謝申し上げます。

# 音と記憶

瓦井太雄　砂辺泰山

## １．緒言

私たちは以前より知っていた「忘却曲線」をヒントに、音と記憶の関係について調べることにした。忘却曲線は、ドイツの心理学者であるヘルマン・エビングハウスが無意味な文字列を目(視覚)と手(触覚)で記憶し、節約率という独自の指標を用いてグラフ化したものである。それに対して私たちは無意味なフレーズを耳(聴覚)のみで記憶し、その正確さをまた違った独自の方法で調べることで、忘却曲線に類似したグラフを作った。

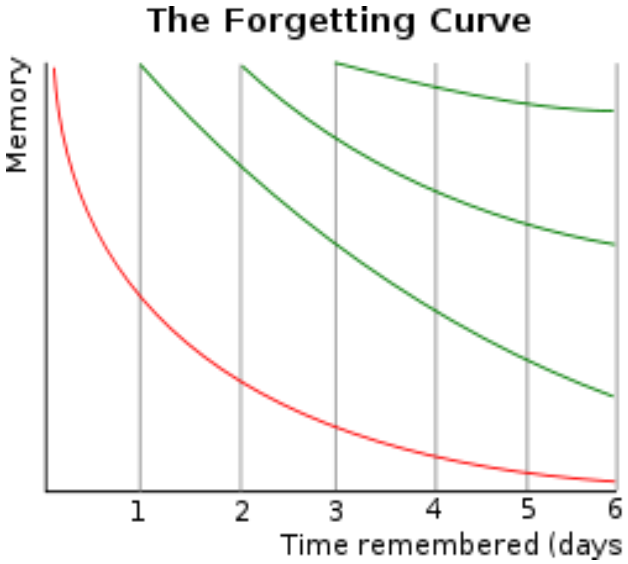
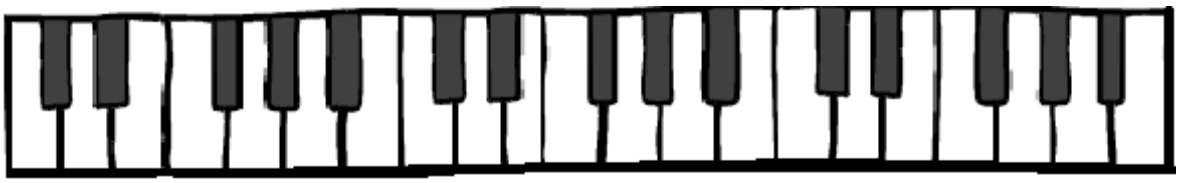



図１　忘却曲線

## ２.方法

(1) フレーズの作成

ランダムに選んだ音階の異なる５つの音を並べたものを１フレーズとした。



②　　⑤　　③　④　　①

図２　ピアノ鍵盤(１)

―　　― ―

図２の番号の順で鍵盤を弾いた場合　ソ レ ド ミ ソ　　となる。

使用した音は１３０Ｈｚのドの音から９９１Ｈｚのシの音までの３オクターブで、音色はピアノとした。

ただし、フレーズの作成時に以下のようなものは除外した。

・単調に上がる、もしくは下がるもの　　例　ドミファラシ　ドシソファレ

→特徴的な音の並びがあると記憶に残りやすくなり、記憶の正確さの測定に影響が出ると考えたため

・同じ音が連続して入っているもの　　例　ドドレシソ　　レソソシミ

→同上

・ピアノ鍵盤の黒鍵に当たる音(図３の楕円の部分)

→先行して行った予備の実験で正解率が非常に低かったため

図３　ピアノ鍵盤(２)

(2)記憶の正確さの実験

・(1)で作成したフレーズを楽器アプリケーション「PianoV」を用いて被験者に２回続けて聞かせる。その後一定時間(６０、１００、１４０、１８０[S])が経過してから被験者に「PianoV」を弾くことでフレーズを再現してもらう。再現した音の正解率を、グラフを書く際の指標とした。

・実験回数は８４回、被験者は実験者２人で行った。

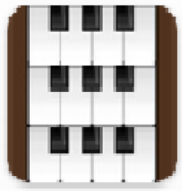


図４「PianoV」

## ３．結果

実験結果より、作成したグラフは次のようになった。

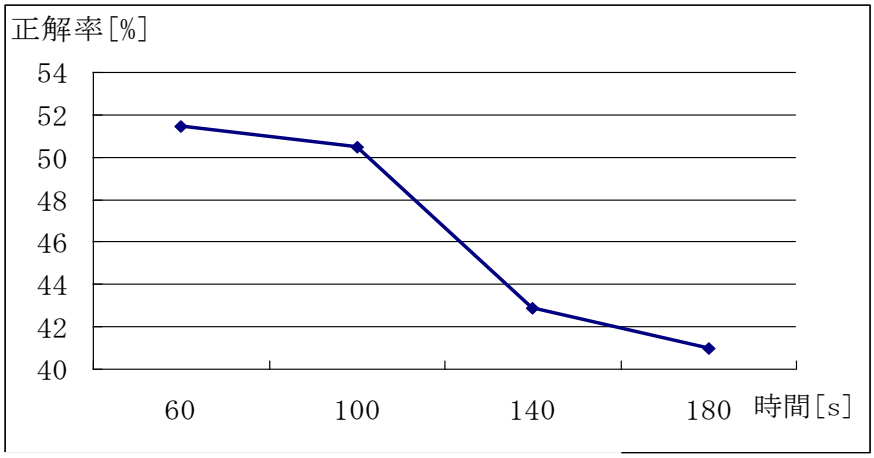



図５　経過時間と正解率の関係

図５より、初めにフレーズを聴いてから時間が経てば経つほど忘れていくことが分かる。

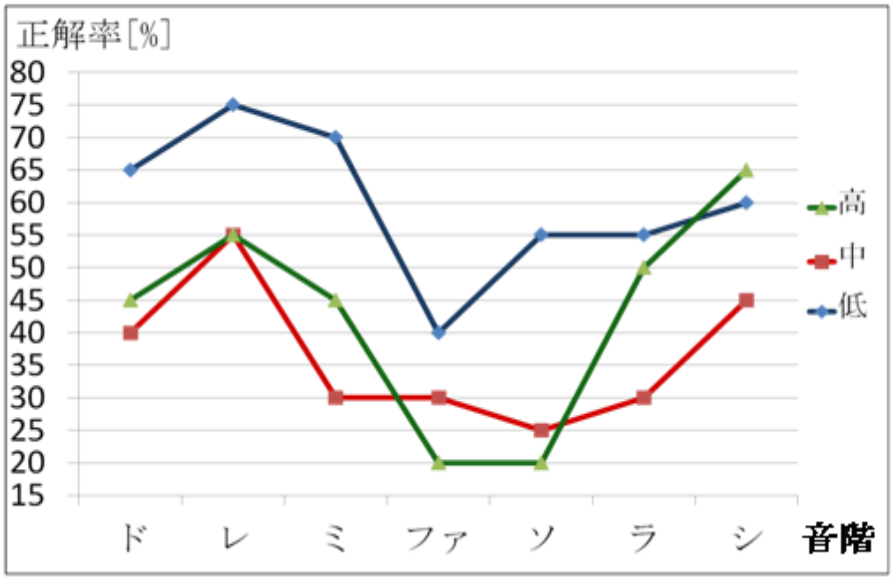



図６　音階と正解率の関係(１)

図６は実験に使用した音をオクターブ毎に低・中・高の３つのパートに分け、音階と正解率の関係をグラフにしたものである。これを見ると、パート毎に正解率に差はあるものの、概形には相似点が見られる。実験を進めるうちに音階の中に覚えやすいものとそうでないものがあるように感じられたので、図６の結果を音階ごとにまとめた。

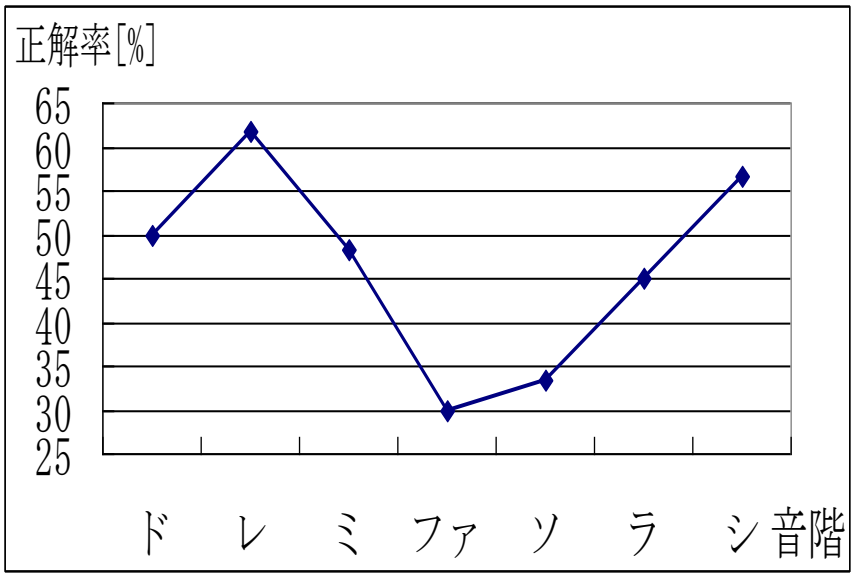



図７　音階と正解率の関係(２)

図７より、ドからシの中で、シ・ド・レの正解率が特に高く、ファ・ソは特に低いことが分かる。

## ４．考察

・エビングハウスの実験から人間の視覚、触覚から得られた情報は時間とともに忘却していくことがわかるが、聴覚の場合も同様であったため、図５のような結果が得られたのだと考えられる。

・私たちが実験でフレーズを暗記している際に、ドの音が印象に残りやすく、ドを基準として暗記しがちであると感じたのだが、図６・図７を見ると、ドとその周辺の音の正解率が高いことが数値でも明らかであり、人は音を聞く際ドを基準にしているのではないかと考えられる。

## ５．結論

・実験から時間の経過とともに、聞いた音の再現における正解率は低下していくという結果が得られた。音のフレーズは一度聴いてから１分後には５０％ほどしか覚えておらず、その後も時間の経過とともに正解率は低下していく。

・実験による音階ごとの正解率を比較すると、シ・ド・レの音は高く、これに対してファやソの音は低くなっていた。

# 音と材質

在田愛菜　坂東里緒奈　藤田華澄　宮野里菜子

## １．緒言

私たちは、“弦楽器の音色はボディの材質によって異なる”ということに興味を持った。
一般的にギターによく使われるウォルナットとマホガニーという２つの木材を用いた。
ここで音色について説明しておく。おんさの音のような純音では波形は正弦曲線になるが、
ピアノのような楽器の音では、波形はたくさんの振動数の波形が混ざることにより正弦曲線にはならない。つまり、音に含まれているいろいろな振動数によって音色の違いができる。
この原理を使っているのがイコライザーで、各振動数を調節することができる。イコライザーで音色を作るときに、低い振動数をあげるとあたたかみのある音、高い振動数をあげるとシャキシャキした音になると言われる。

また、一般的にギターのボディに用いる木材(ウォルナットとマホガニー)について、「ウォルナットは硬くシャキシャキした音」「マホガニーはあたたかく深みのある音」になると言われる。
イコライザーのことをふまえ、ウォルナットでは高い振動数のスペクトルが大きく、マホガニーでは低い振動数のスペクトルが大きく現れるのではないかと予想した。

そこで２つの木材の相違点である「かたさ」つまり「弾性」に着目すると、弾性と相関する物理量として物体に固有の値「固有振動数」があるので、これについて研究した。

まず、私たちは、２種類の木材(ウォルナットとマホガニー)を用意し、それぞれの固有振動数を測定した。この２種類の木材に別途作成した弦を固定する台(弦固定台)を付け、次にその固有振動数の１／４の振動数で弦を弾き、響いた音のスペクトルを測定した。
このようにして、私たちは、木材の固有振動数と、その木材を使った２種類の簡易ギターから出る音のスペクトルとの関係を研究した。

## ２．方法

実験①　固有振動数の測定
木材の固有振動数を打撃音法によって求めた。

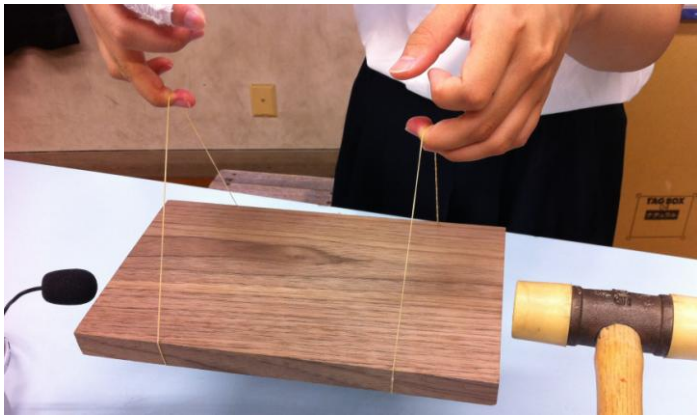


図１（固有振動数測定の様子）

図１のように、木材を輪ゴムで吊るし、木材の短い方の辺を木槌で叩いた。
長い方の辺でも同様に行い、出た音を audacity を用いて解析した。この audacity は振動数ごとの音の大きさがわかる。

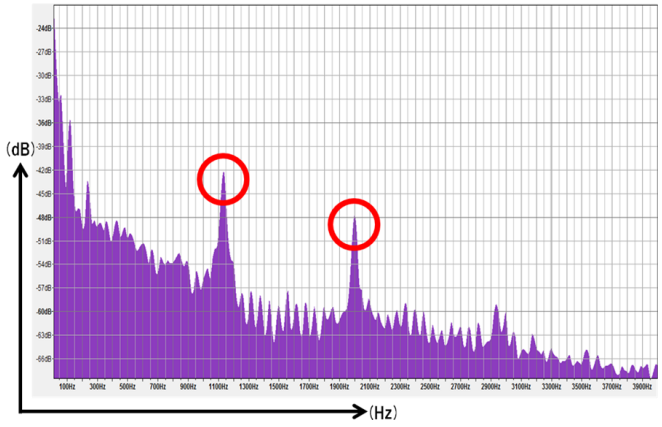



図２（スペクトルの例）

図２のように他の振動数よりも大きく出ている振動数に注目し、２種類の木材の短い辺と長い辺において、それぞれ１０回ずつ計測した。

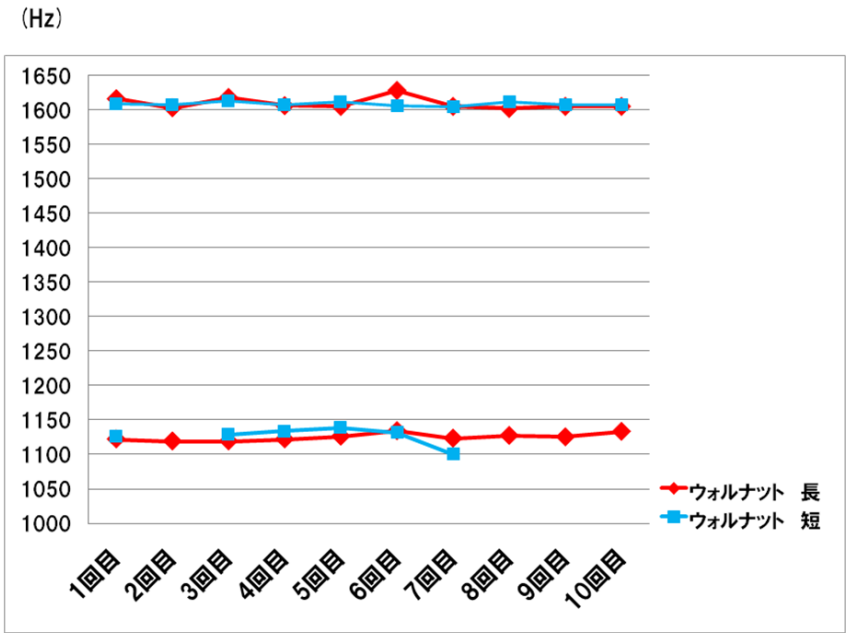



図３（ウォルナットの固有振動数測定結果）

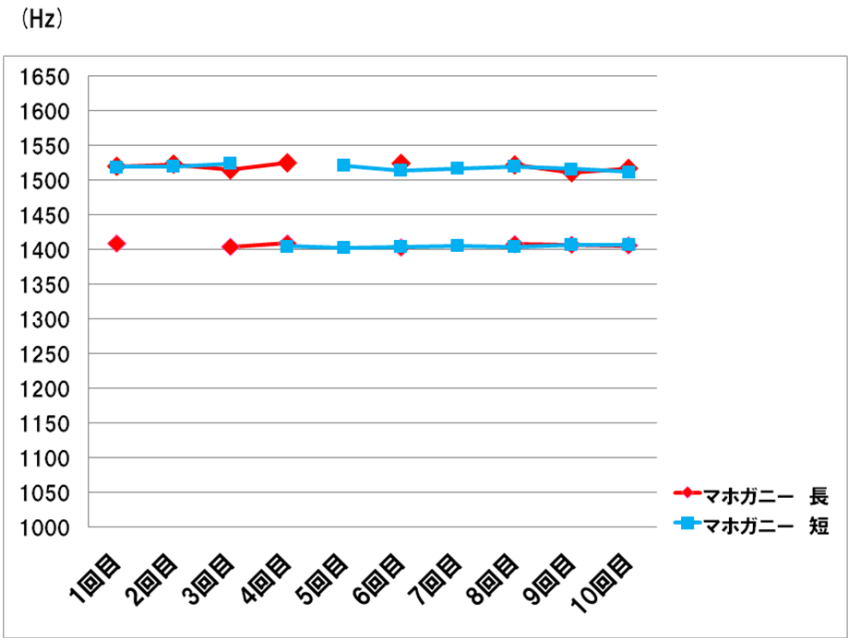



図４（マホガニーの固有振動数測定結果）

図２のように他より大きく出ていた振動数それぞれで１０回分の平均をとると、図３、図４のようなグラフになる。図３については、大きく出ている振動数は長い辺で１１２５、１６０９Hz、短い辺では１１２７、１６１０Hz、となり、これらよりウォルナットの固有振動数を１１２６Hz、１６１０Hz と定めた。
図４より、同様にマホガニーの固有振動数を１４０６Hz、１５１９Hz と定めた。

実験②
実験①で用いた木材に弦固定台を付けて簡易版弦楽器を作成した。その簡易版弦楽器を指で弾き、ボディとして使う木材に共鳴させて出た音を解析した。

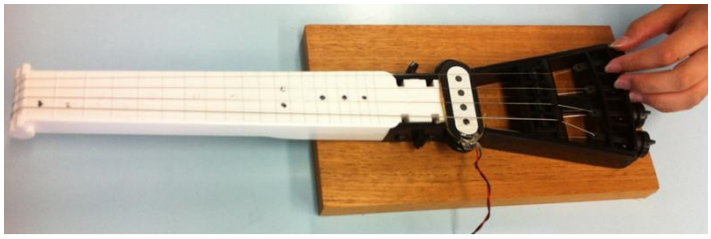


図５（弦楽器と木材）

このとき弦楽器で弾いた音は、最初の実験で定めた固有振動数の１／４の振動数とその前後の値の振動数である。１／４の振動数で弾いたのは、今回使った簡易版弦楽器では、出せる振動数が約３００Hz～約１０００Hz と限られているからである。楽器は弾いた音とその倍音も出す、つまり、固有振動数の１／４の振動数で弾いても弦からは木材の固有振動数も出ている。より正確な固有振動数の値を得るために１／４の振動数とその前後の値の振動数についても調べた。

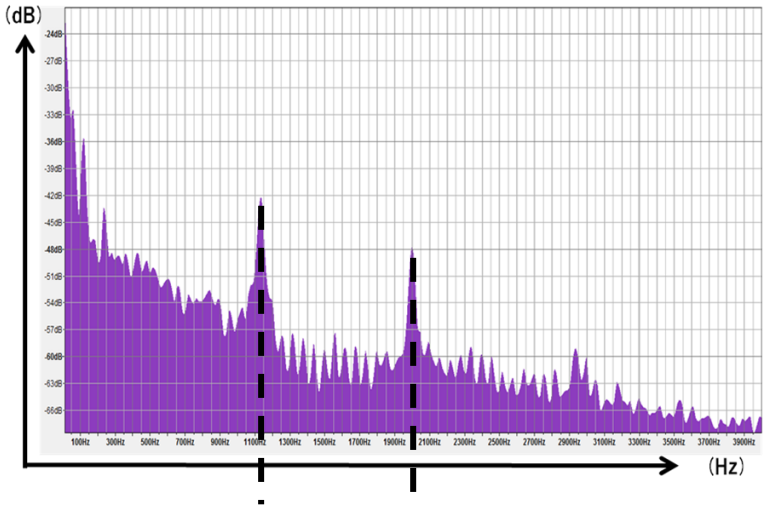



図６（スペクトルの例）

dB の定義「基準量 A に対する B の比が X(dB)のとき、$B/A=10^{0.1x}$ 」を用いて、
A を基準としたとき B-A=X(dB)(X＜0)
A(Hz)の強さと B(Hz)の強さの比は $A:B=1:10^{0.1x}$ となる。

例えば、A に弾いた音１００Hz、B に固有振動数４００Hz を代入すると、
振動数 A を１．０とした振動数 B との比が１：０．０１と求められる。

以上の計算式を用いて、弾いた音を１．０とした固有振動数の比を次のグラフのようにまとめた。

**３．結果**

★印が付いているところは、実験①によって定めた固有振動数のちょうど１／４の振動数であり、この比が最も良く出ると予想した。

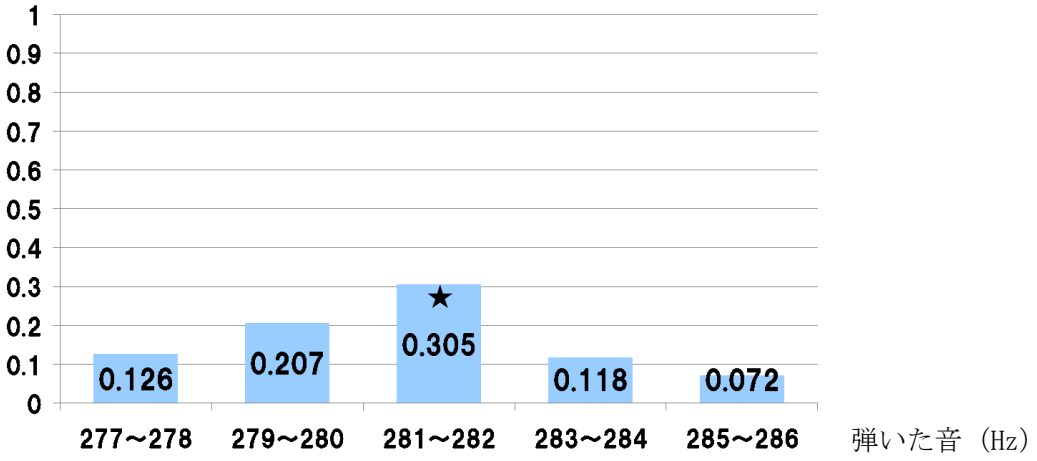



図７（ウォルナット１１２６Hz の１／４周辺を弾いたときの割合）

実験①で定めた固有振動数のちょうど１／４である２８１～２８２Hz の比が
よく出ていてグラフは山型になっている。

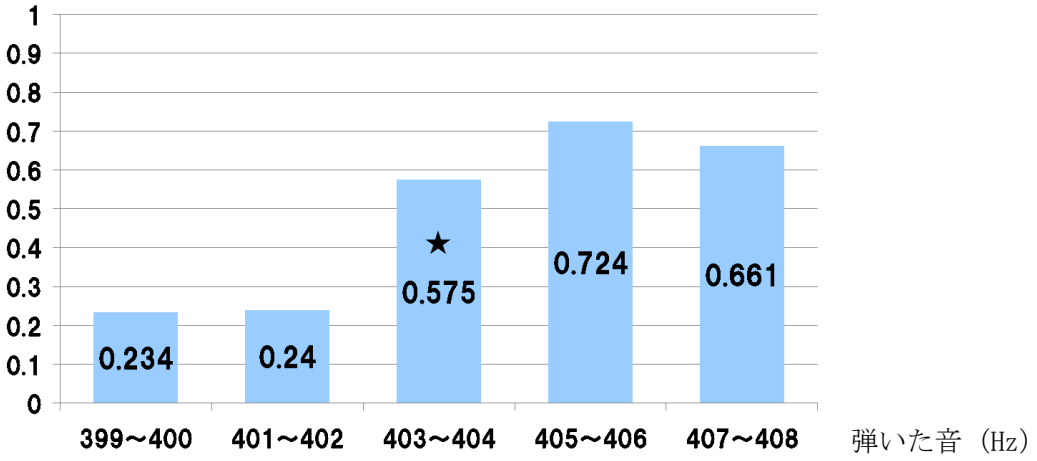



図８（ウォルナット１６１０Hz の１／４周辺を弾いたときの割合）

定めた固有振動数の１／４の振動数よりも少し高い振動数での比が最も高くなっていることから
本来の固有振動数は最初の実験で定めた値よりも少し高いと考えられる。
図８のグラフは４０５～４０６Hz を中心に山型になっている。
また、１１２６Hz と１６１０Hz のどちらの時も弾いた音を１とした固有振動数の割合は
０．１桁以下となっている。

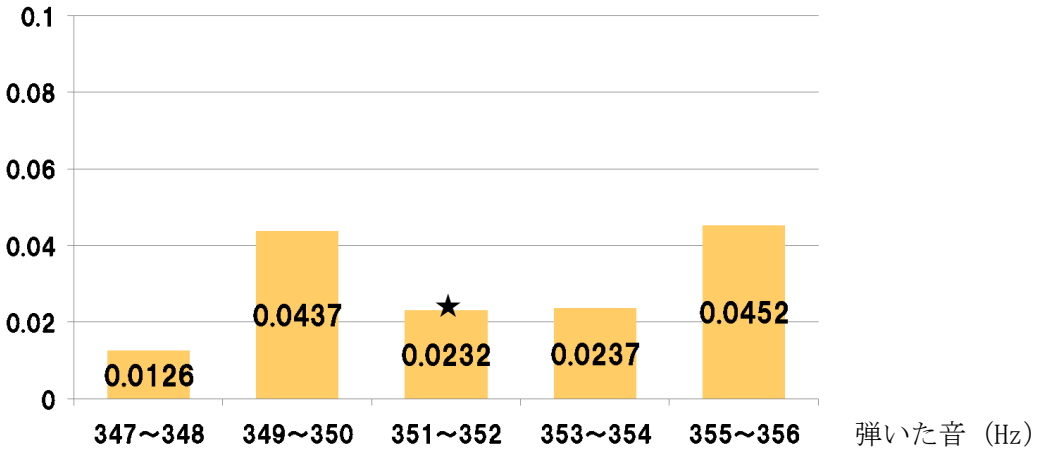



図９（マホガニー１４０６Hz の１／４周辺を弾いたときの割合）

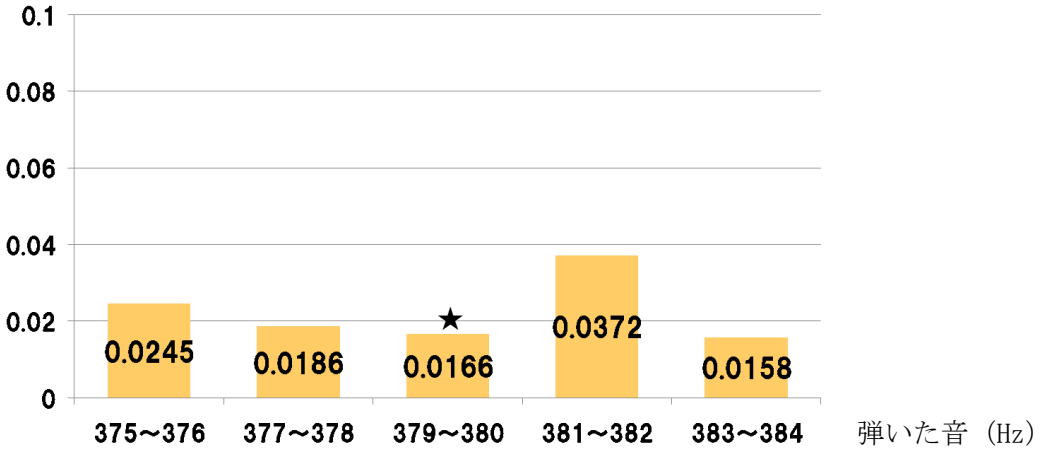



図１０（マホガニー１５１９Hz の１／４周辺を弾いたときの割合）

どちらの振動数でも、実験①で定めた固有振動数の１／４の振動数の比はあまり出ていない。
グラフの形にも単調な増減等は見られず、凹凸がある。
また、弾いた音を１とした固有振動数の割合は０．０１桁以下が多くウォルナットに比べると比が小さくなっている。

**４．考察**

| | ウォルナット | マホガニー |
|---|---|---|
| グラフの形 | なだらかな山型 | 凹凸がある |
| 弾いた音を1．0としたときの比 | 0．072～0．724 | 0．0126～0．0452 |

表1（実験結果のまとめ）

ウォルナットはグラフが山型になり、マホガニーは凹凸があった。
弾いた音を１．０としたときの固有振動数の割合はウォルナットでは０．０７２～０．７２４と比較的大きく、マホガニーでは０．０１２６～０．０４５２と比較的小さくなった。

## ５．結論

実験②から得られたグラフの形から、ウォルナットは固有振動数が出やすいことから共鳴の範囲がせまく、いろんな振動数が混ざりにくいことがわかる。このことがウォルナットのシャキシャキした音に起因していると考えられる。

また、マホガニーは固有振動数が出にくいことから共鳴の範囲が広く、いろんな振動数が混ざりやすいことがわかる。このことがマホガニーのあたたかみのある音に起因していると考えられる。

## ６．参考文献・使用ソフト

学研「大人の科学」

FFTana・audacity