

平成 22 年度卒業論文

自動演奏ギターのブリッジミュート装置開発

(ミュート音の解析及びミュート奏法を用いた演奏)

福岡工業大学工学部

知能機械工学科

07E2020 金子 大祐

07E2061 松本 耕治

指導教員 河村良行

目次

第1章 緒言	1
1.1 研究背景	
1.2 研究目的	
第2章 装置概要	3
第3章 装置全体の点検	7
3.1 ソレノイドの検査	
3.2 PCI ボードの出力チェック	
3.3 PCI ボード、プログラムの点検	
第4章 ブリッジミュート装置の製作	10
4.1 設計	
4.2 回路	
4.3 組立	
第5章 ブリッジミュート音の測定	13
5.1 実験内容	
5.2 実験方法	
5.3 測定結果と比較	
第6章 周波数解析	16
6.1 解析方法	
6.2 結果	
6.3 人間の演奏との比較	
第7章 制御プログラム	20
第8章 演奏プログラム	26
第9章 結言	40
9.1 結言	
9.2 今後の課題	
9.3 謝辞	
参考文献	42
付録	44

第 1 章

緒言

1.1 研究背景

近年ロボット産業は目を見張るほどの成長を遂げており、なかでも人々への娯楽を目的としたアミューズメントロボットが注目されてきている。本研究ではジョージア工科大学の CrazyJ のような楽器演奏ロボットに着目し、ギターを用いた自動演奏ロボットの開発、高性能化を目的とする。研究の最終目標として単調、機械的な演奏を超えて人間が演奏するような強弱やリズムの個性を表現でき、かつ人間では到底できないような速度や奏法を演奏することである。

1.2 研究目的

今年度は、前年度よりも簡単かつ応用が利くブリッジ部分で行うミュート奏法を使用する。今年度の研究目標は、ミュート装置の開発とその装置を作動させるプログラムの作成、ミュート音の解析及びブリッジミュート奏法を用いた演奏プログラムの開発である。左手部分で弦に軽く触れる事でミュートを行っていた前年度よりも、ミュートが行いやすく制御しやすいブリッジミュートを採用する事で、演奏の幅を広げる事ができる。

第 2 章 装置概要

装置の概要を図 2.1 に示す。装置のシステムは図 2.2 に示す。弦を押さえる押弦用ソレノイド群には図 2.3 に示すソレノイド(CA-154804)を用いて1~4フレットを押さえる。人間において弦を弾くピック制御にはステッピングモータを用いてモータの回転動作で制御を行う。水平方向制御を図 2.4 に示すモータ(kh56jm2-951)で行い、垂直方向制御を図 2.5 に示すモータ(KT35FM1-552)で行い弦を弾く動作を分割する。

ソレノイド制御においては、32ch デジタル入出力ボード(PCI-2703A)をプログラムにより出力させ、トランジスタのダーリントン接続回路を通しスイッチングを行う。ミュート装置はミュート制御用のプログラムで制御を行う。水平方向及び垂直方向制御はステッピングモータパルスボード(LPC-742020)からパルスを出力し制御を行う。今年度はブリッジミュート奏法を行うためにブリッジミュート装置を製作した。ブリッジミュート装置の概要を図 2.5 に示す。ブリッジミュート奏法とは、弦の端に右手の側面を軽く触れる事によって、弦の音を短く切る奏法である。

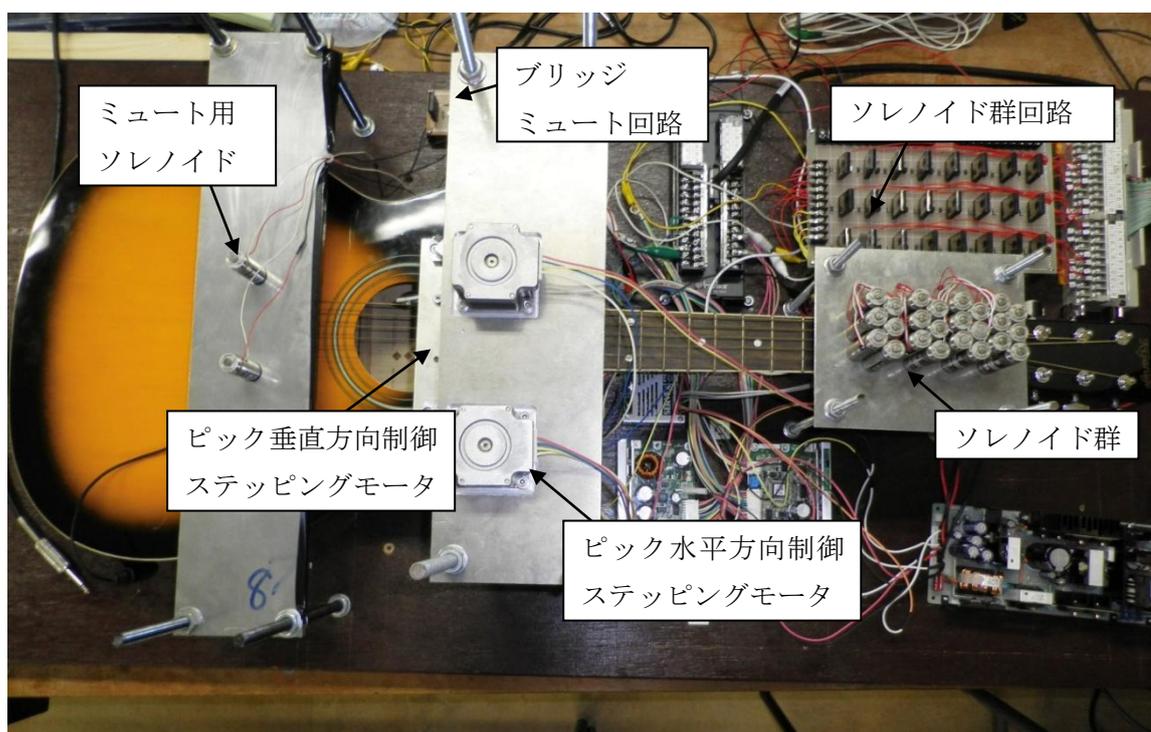


図 2.1 自動演奏ギター（装置全体図）

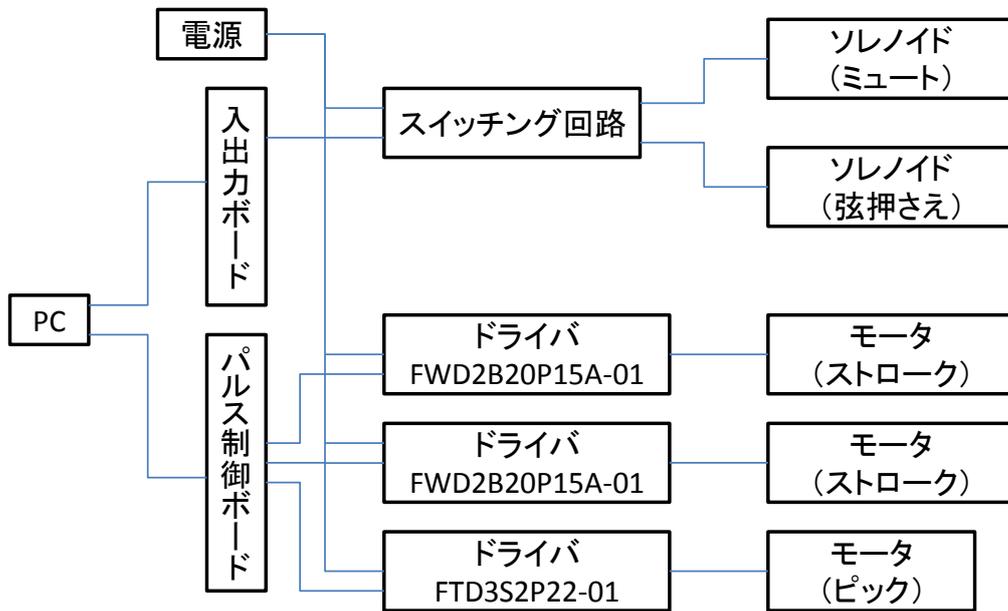


図. 2. 2 自動演奏ギター (システム図)

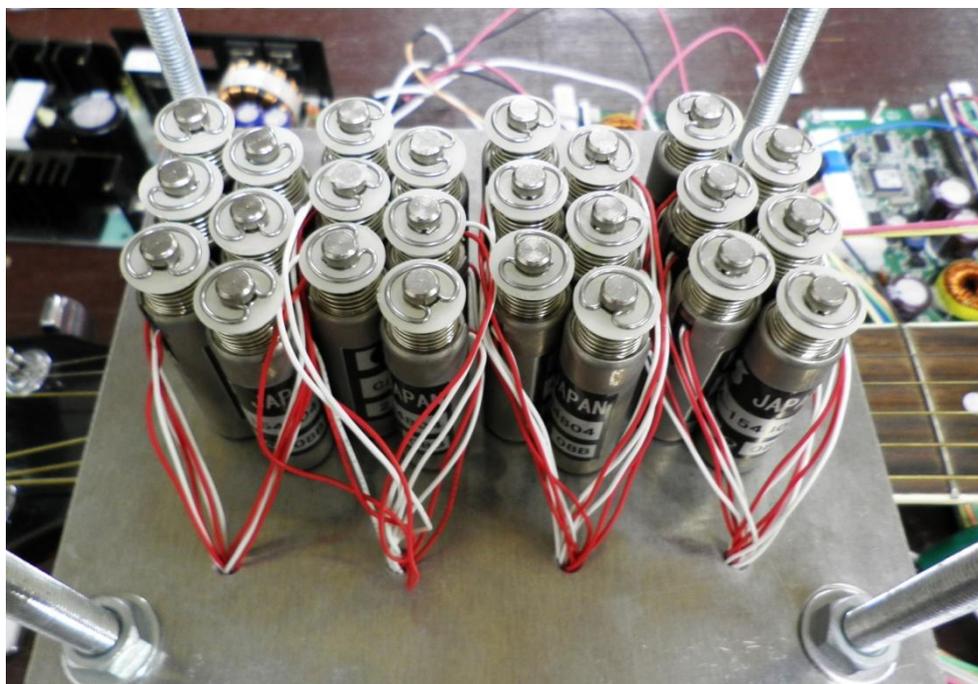


図 2. 3 押弦用ソレノイド群

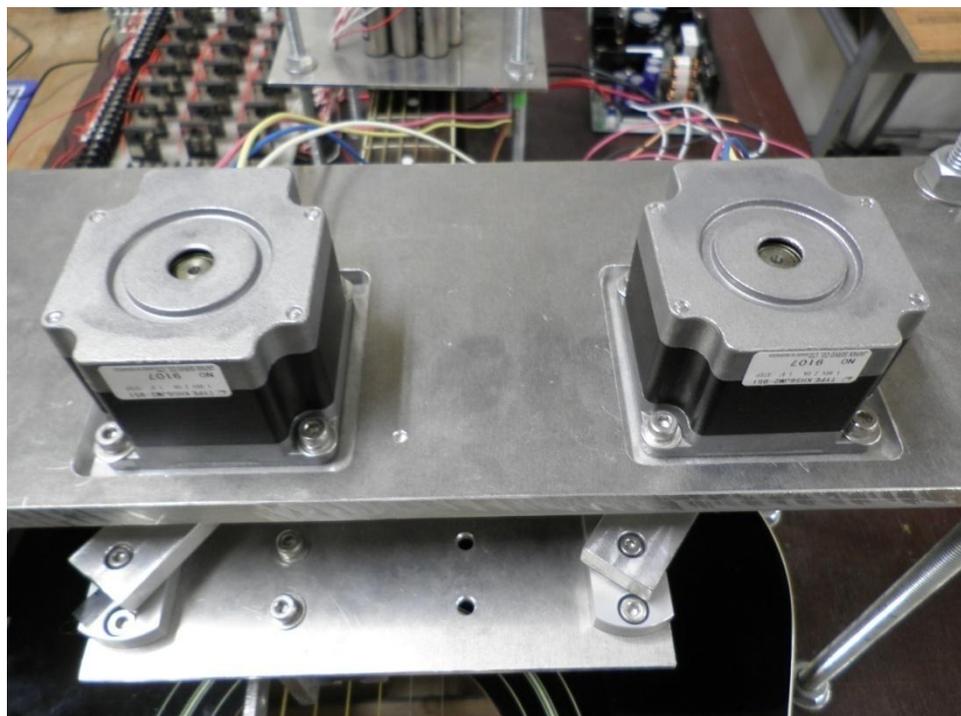


図 2.4 ステッピングモータ（水平方向制御）

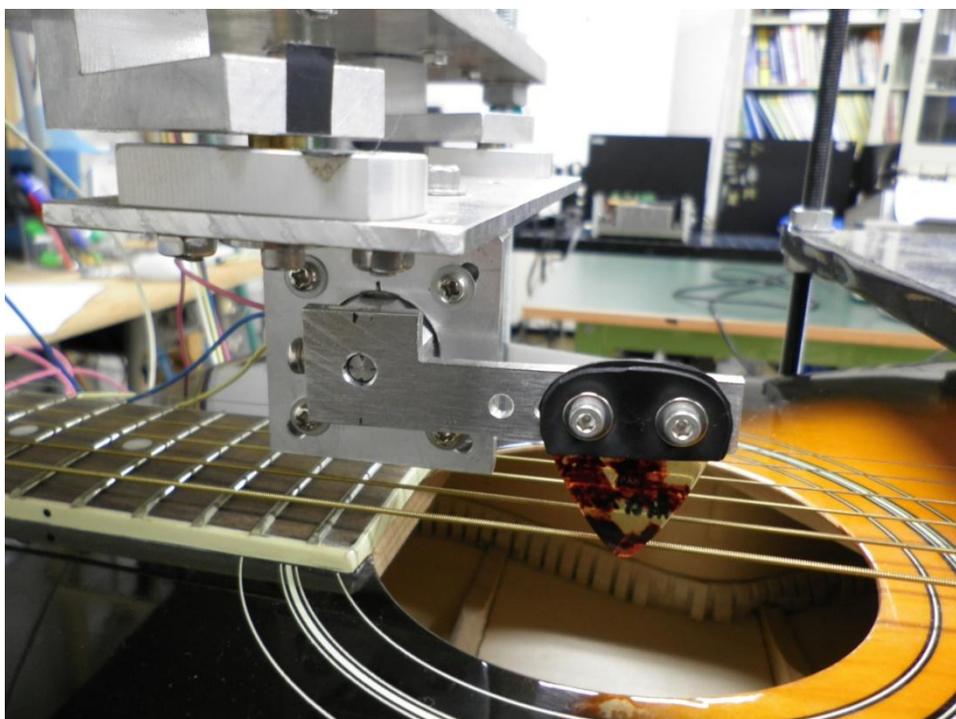


図 2.5 ステッピングモータ（垂直方向制御）

第 3 章

装置全体の点検

3.1 ソレノイドの検査

作動しないソレノイドが多かったため、ソレノイドや回路のはんだ付けを行った。しかし、それでも作動しないソレノイドが多かったため別の原因があると判断し、実験を行った。ソレノイド自体に原因があると考え、ソレノイドにどれだけ電流が流れているかを測定した。測定方法は、プログラム作動時に回路にオシロスコープを接続し電流を測定する。結果として、ソレノイドには約 1.0A の電流が流れ正常に作動していることがわかった。

また、ソレノイド作動時にノイズが発生し始めた、しかしソレノイドには正常に電流が流れていたため原因とは考えられない。回路側に問題があると推測し調査したところ、ステッピングモータ側の制御ボードが接触不良を起こしていることが判明した。配線を結線し直すとノイズは納まった。

3.2 PCI ボードの出力チェック

ソレノイドの配線を修理しても作動しない箇所が多いため、ソレノイドの制御ボードに問題があると考え、PCI ボードのマニュアルを参考に LED 点灯回路による実験を行った。図 3.1 に示す点検を行うために実験回路を作り、ソレノイドを動かすプログラムの命令を発信する。LED が点灯することで、正しく電流が流れているかを調べた。結果として、表 3.1 に示すチェックの部分が点灯しなかった。このためソレノイドが動かない原因のほとんどは PC から端子台までにあるとわかった。

3.3 PCI ボード、プログラムの点検

PCI ボード、接続コネクタに問題があると考え同じ型式のものと交換し 3-2 で行った実験を再度行った。しかし結果は変わらず、二つに問題は見られなかった。次に、ソレノイドの制御プログラムに着目し問題点がないか確認を行った。

結果として、ソレノイド制御プログラム内の端子台を指定するプログラム番号が対応していない数字や重複した数字を使用していた。このためプログラムが正しく作動せず LED が点灯しなかった。

間違ったプログラム番号を対応させる事で、点灯しなかった箇所は全て点灯した。

第 4 章

ブリッジミュート装置の製作

4.1 設計

今年度の研究目標であるブリッジミュート奏法を行うために、ブリッジミュートを行う装置を製作した。まず装置の大まかな構造を考え、ブリッジミュート装置の設計を手書きで行い、設計図を付録のミュート装置設計図に示す。

4.2 回路

ミュート装置にはソレノイドを使用し、左手側同様にトランジスタを用いたダーリントン回路でソレノイドを動かす。及び装置回路図を図 4.1 に示す。回路に使用した部品は以下のようになっている。

- ・トランジスタ 2SC1815GR(Tr1)
- ・パワートランジスタ 2SC5200(Tr2)
- ・抵抗 105[K Ω](R1)
- ・抵抗 500[Ω](R2)
- ・ダイオード 1N4007

4.3 組立

加工した材料を組み立て、回路とソレノイドを接続し固定した。完成した装置を図 4.2 に示す。二つのソレノイドを動かすことによってギターのリッジ付近にシリコンゴムを張ったアルミ板が接触し、弾いた弦の音を短く切るミュート奏法を行う。シリコンゴムは人間の手の弾力に近づけるため、硬度 5° のシリコンゴムシートを使用した。

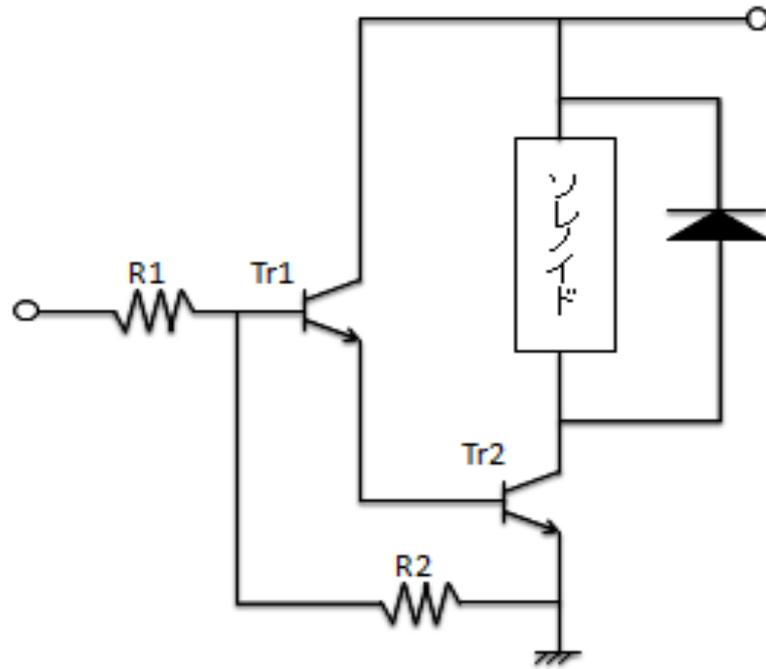


図 4.1 ブリッジミュート装置回路

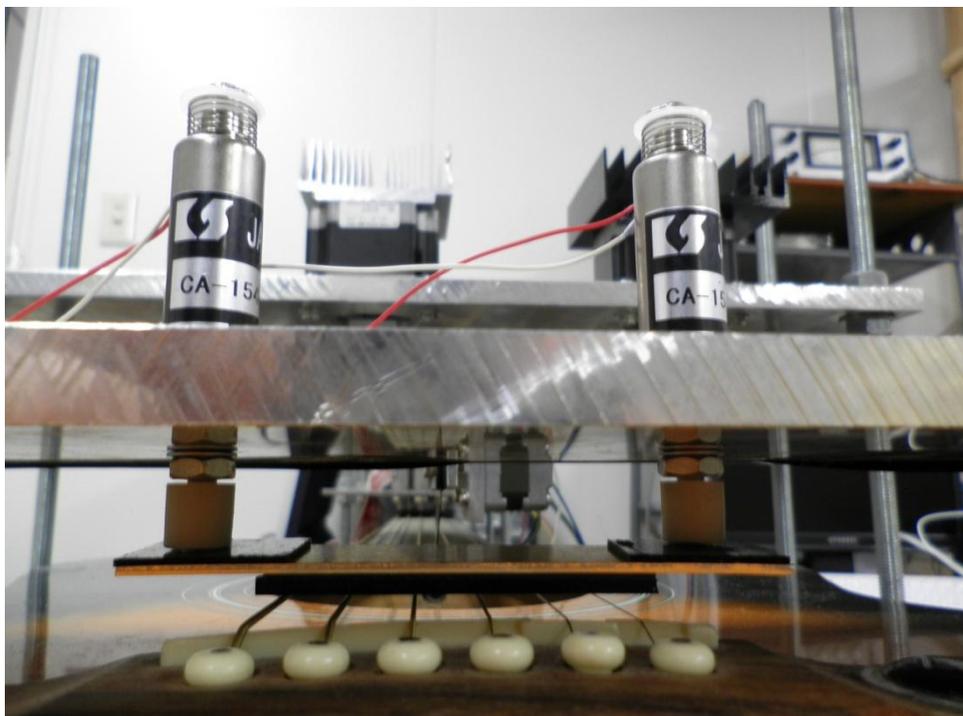


図 4.2 ブリッジミュート装置駆動部分

第5章

ブリッジミュート音の測定

5.1 実験内容

製作したブリッジミュート装置がブリッジミュートを行えているか、またブリッジミュート奏法にどのような特徴があるのかを調べるため、ミュート装置の使用による影響を測定し、解析を行った。

実験内容はミュート装置を ON、OFF それぞれの状態で行った場合の音の比較である。今回は6弦単音を実験対象とし、ダウンピッキング1回での弦の音量と振動し続ける時間を測定した。

5.2 実験装置

測定方法として、解析ソフト Sound Engine Free を使用した。図 5.1 に使用画面を示す。実験の流れを図 5.2 に示し、内容はパソコンから自動演奏ギター装置へ命令を送り、次に命令を受けた装置が動作し弦を弾く。弦の振動音をギターに取り付けたピックアップが電気信号に変換し、信号をギターアンプへ送る。最後にギターアンプで増幅された音がパソコンのマイク端子へ送られ、解析ソフトにより音が測定される。

5.3 結果及び考察

実験結果は図 5.3、図 5.4 に示す。図 5.3 はブリッジミュート装置を使用しない場合の測定結果で、これを基本状態とする。図 5.4 はブリッジミュートを使用した状態である。図は縦軸を音量(dB)、横軸を時間(秒)で時間は点線一つを1秒としている。

二つを比較してみると、弦の音量は弾き始めにおいて基本状態のほうが少し大きくなっている。弦振動の減衰は基本状態は緩やかで、ミュート状態は急激に振動が減衰している。基本状態で弦を弾いた時よりもミュート装置を使用した状態で弦を弾いたときの方が弦の振動時間は3分の1程度まで短くなっていることがわかった。

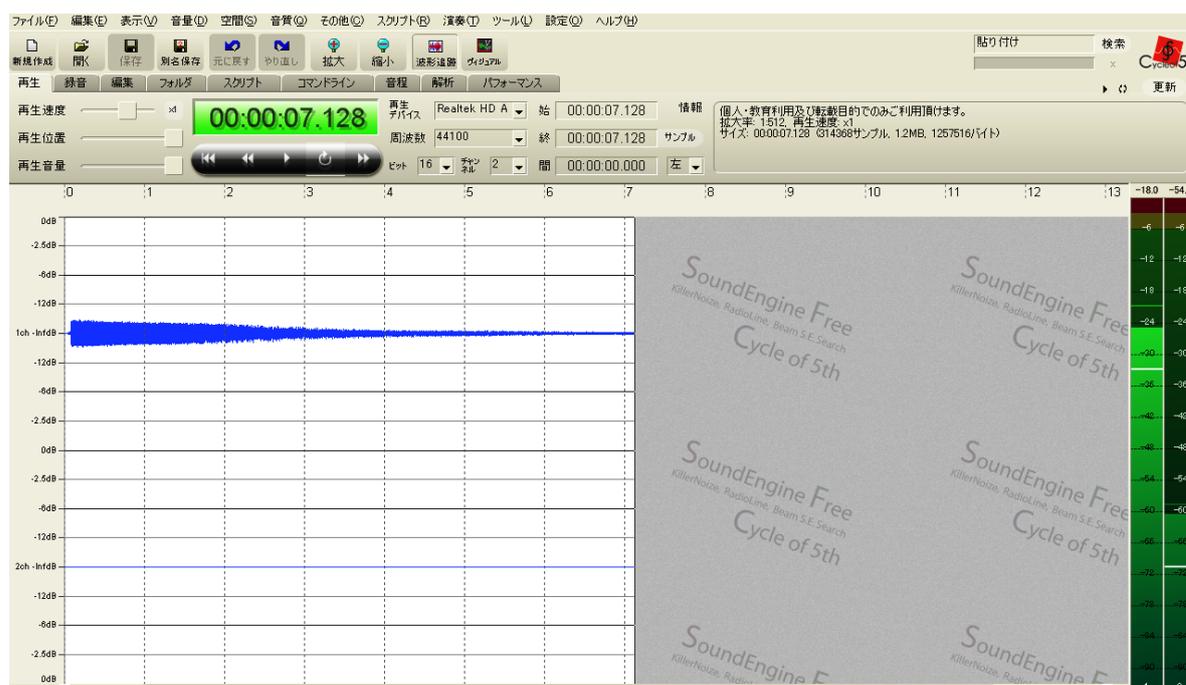


図 5.1 Sound Engine Free 使用画面

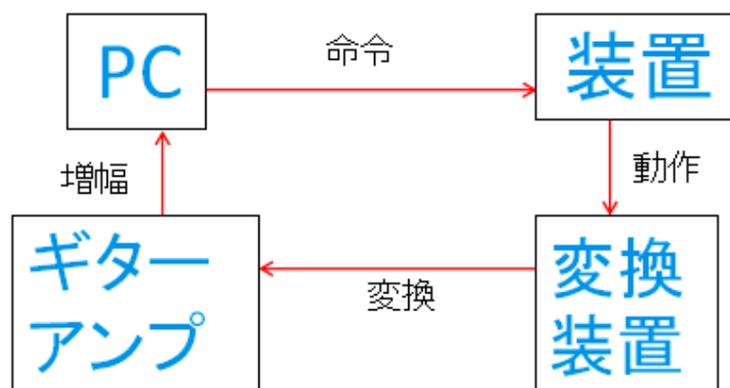


図 5.2 測定実験簡略図

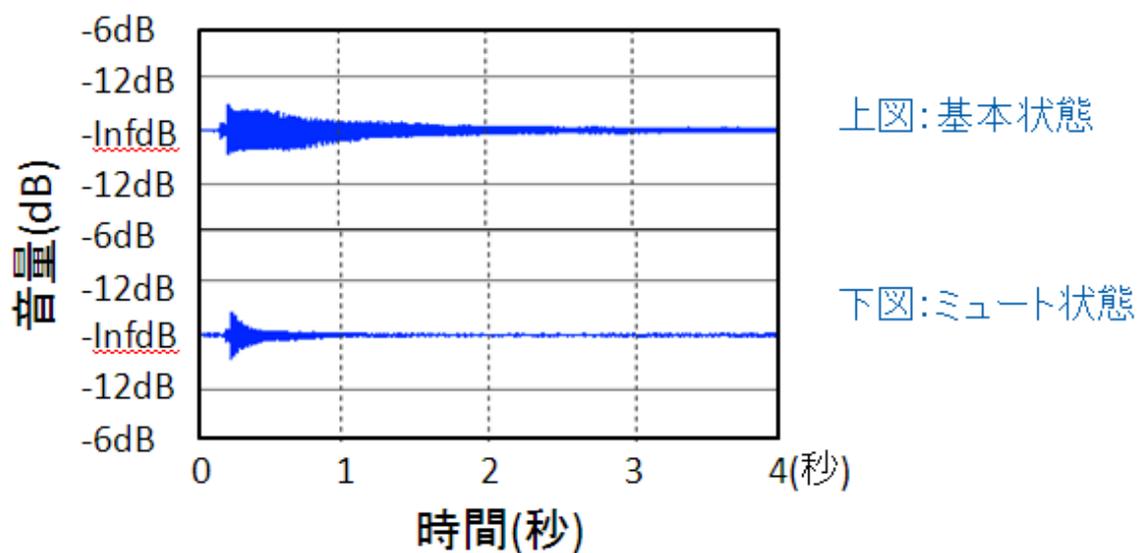


図 5.3 音量、時間測定

第 6 章

周波数解析

6.1 解析方法

Sound Engine Free で測定した 6 弦単音弾きのデータを、MATLAB2010 を用いてフーリエ変換による周波数解析を行い、ブリッジミュート装置の使用が周波数にどのように変化を与えるか比較した。MATLAB 使用画面を図 6.1 に示す。

6.2 結果及び考察

解析した結果は図 6.2、図 6.3 のような結果になった。図は縦軸を音圧、横軸を周波数とし、この図において音圧とは周波数成分を見やすくするために dB の任意スケールである。

二つを比較すると基本となる波はどちらも同じ周波数上にあることが確認できた。しかし基本状態の方がミュート時に比べて音圧は強く高調波成分も豊かであった。このことからブリッジミュート奏法の特徴はただ単に音を短く切るだけでなく、弦の振動を抑える事で周波数に対する音圧、高調波を減衰させる働きを持つことがわかった。

6.3 人間との演奏の比較

また、ブリッジミュート装置は周波数成分の観点において、人間の演奏に近い波形を持つかを調べるため、人間が 6 弦単音をブリッジミュート奏法を行った時の音量、時間を同じように測定し周波数解析を行った。図 6.4、図 6.5 に示す。比較すると、少し音圧は変わるが周波数成分は同じ位置にあり、全体の波形も似ている事が確認できる。このことから、ブリッジミュート装置はブリッジミュート奏法を正しく行えていると言える。

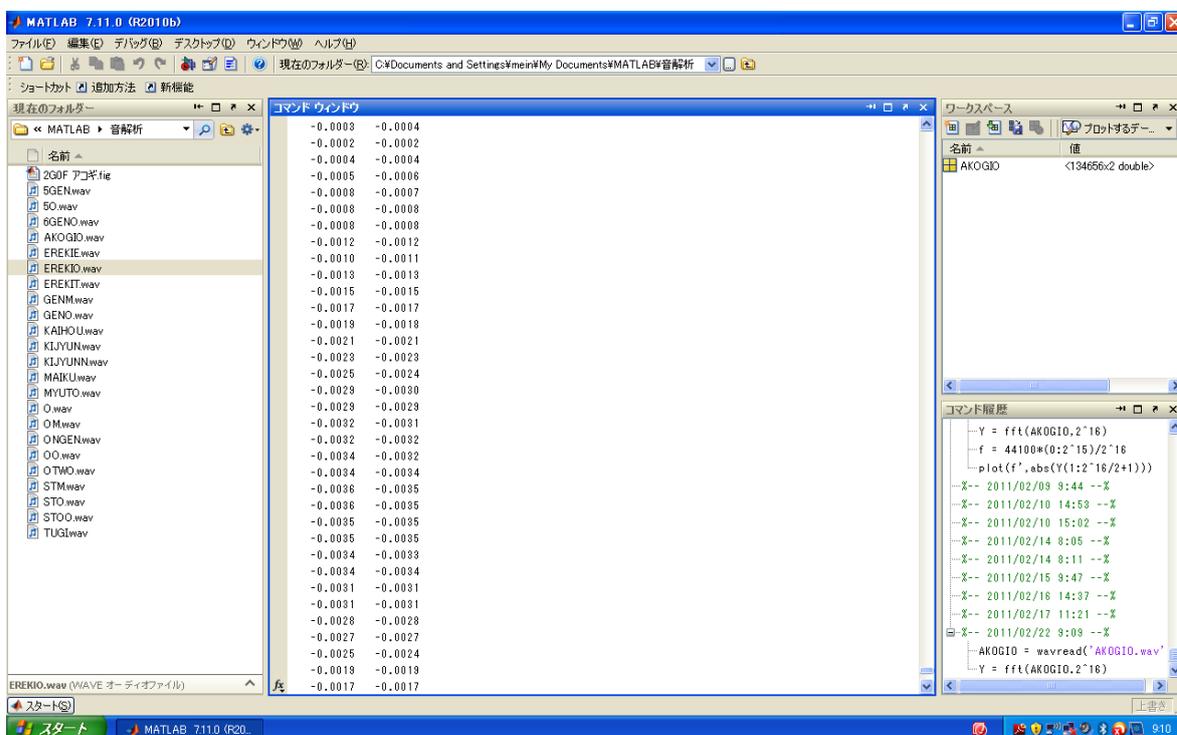


図 6.1 MATLAB 使用画面

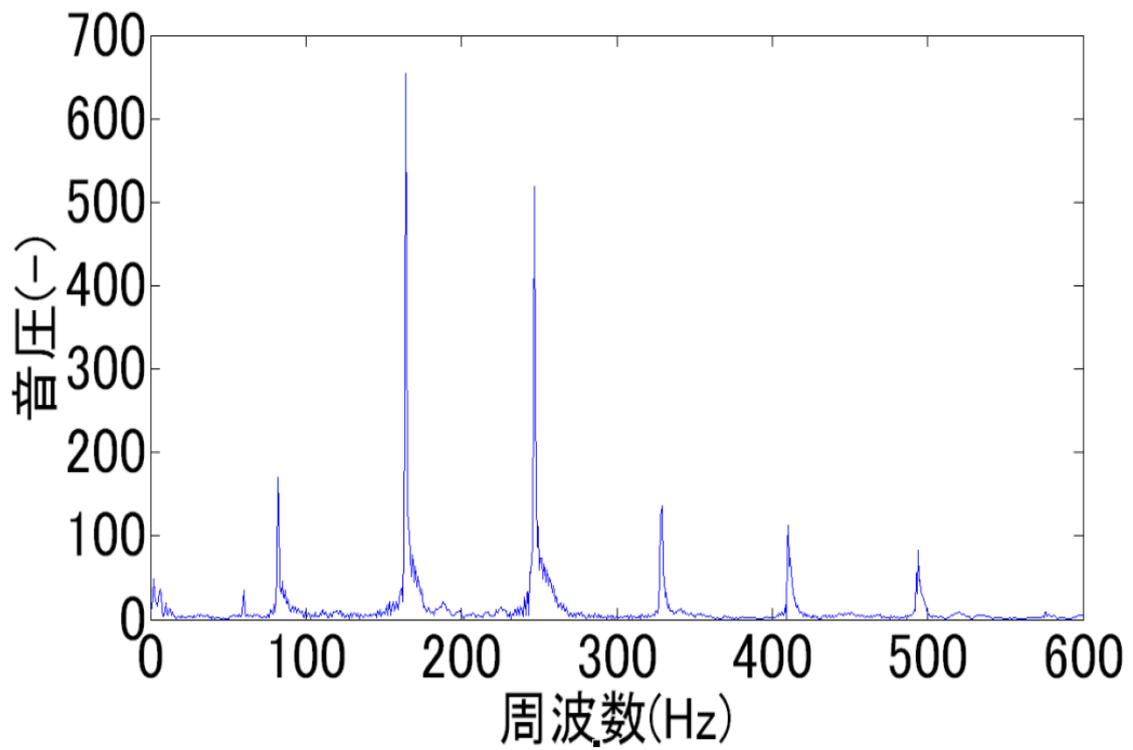


図 6.2 基本状態の周波数解析結果

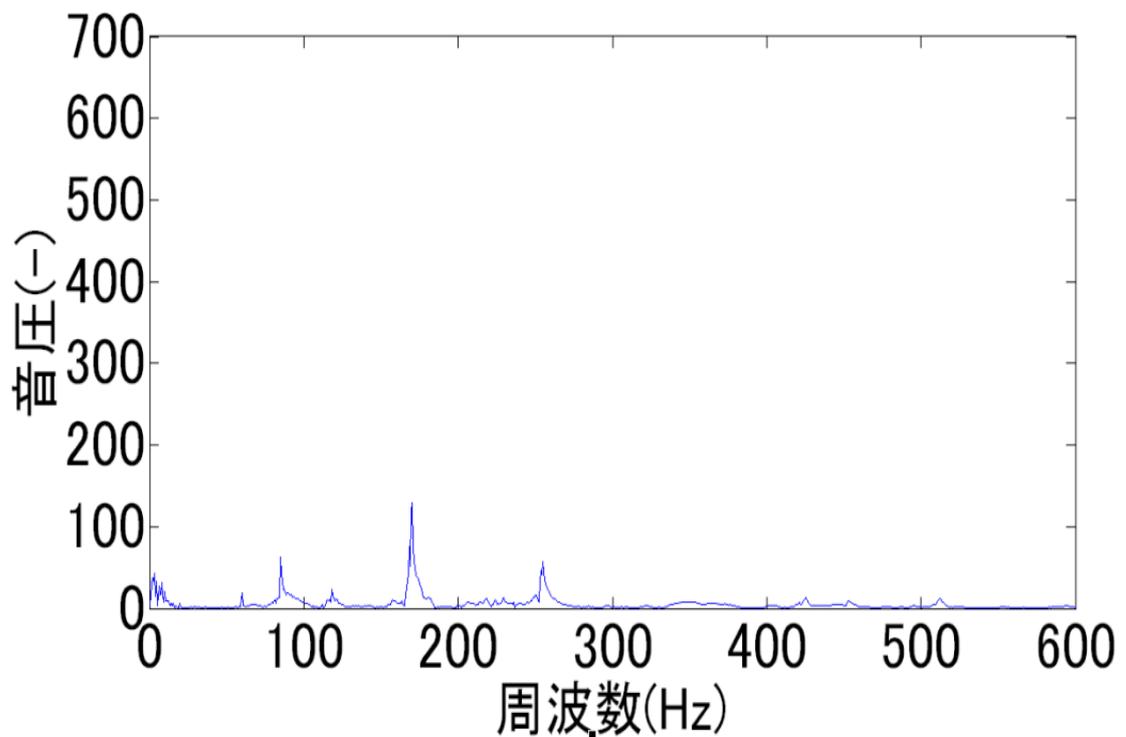


図 6.3 ブリッジミュート状態の周波数解析結果

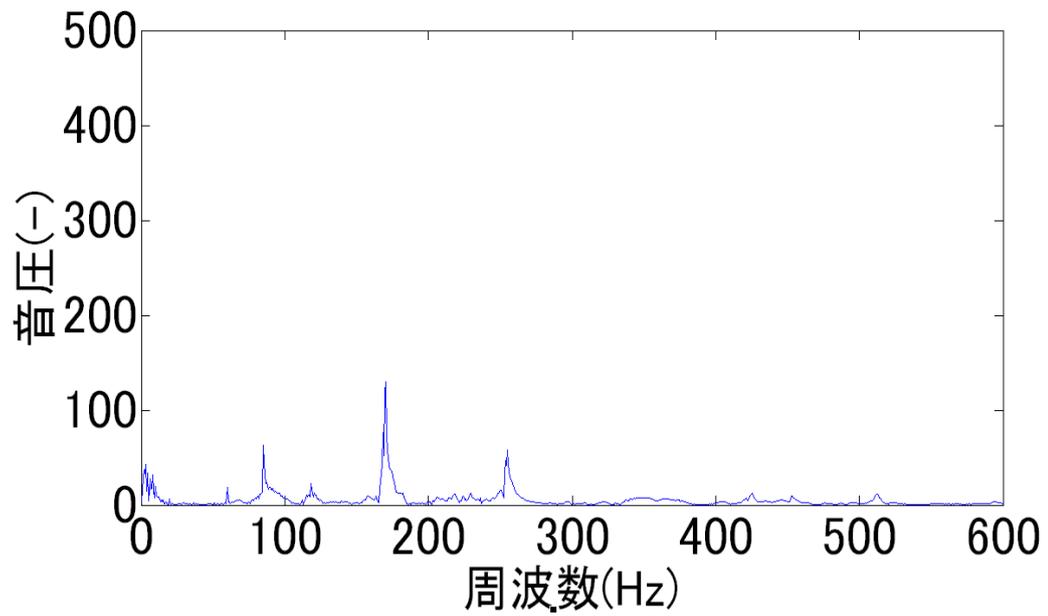


図 6.4 ブリッジミュート装置の演奏

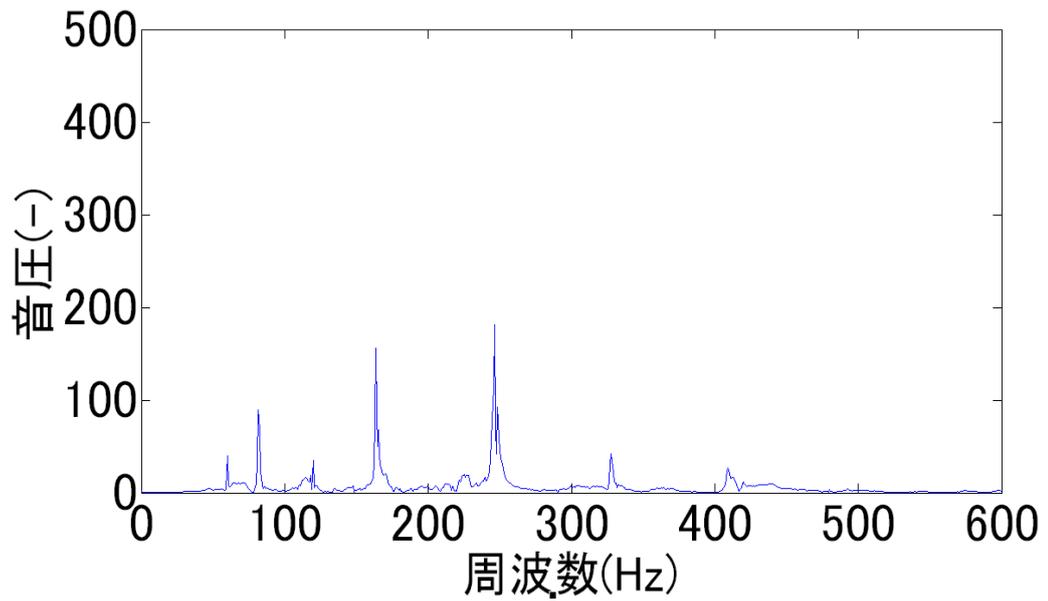


図 6.5 人間の演奏

第7章

制御プログラム

本研究では、OS:WindowsXP、制御言語:Visual Basic6.0 を使用してプログラムを作成し、ソレノイドとモータの制御を行った。

今年度はミュート装置を制御するためのプログラムを作成した。前年度は左手側のソレノイドを用いてミュートを行っていたので、演奏中に行うのが難しく開放弦の音しか出せなかった。しかし、ブリッジミュート奏法を採用したアルミ板のついたソレノイド2個を同時に制御することで、開放とミュートをすばやく使い分ける事ができ、左手側のソレノイドを使用しないので演奏の幅を格段に広げることができた。

製作したプログラムは前年度のプログラムに組み込み図 7.1 に示す。また、プログラムの制御コードを以下に記載する。

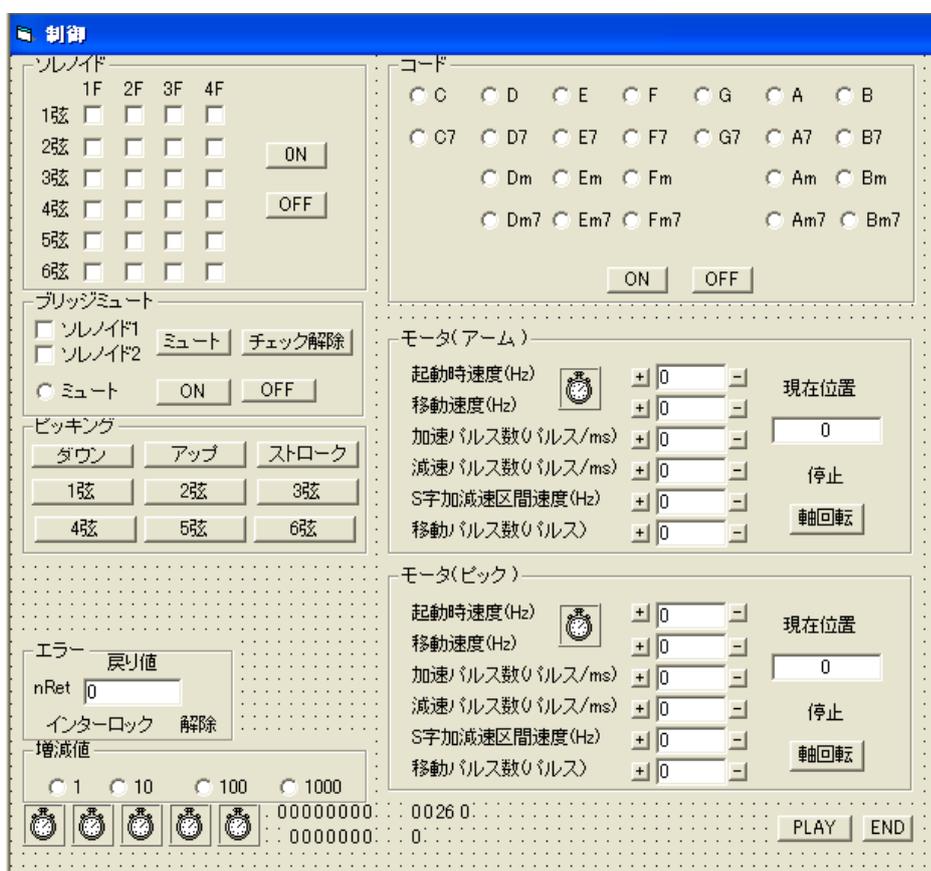


図 7.1 制御プログラム

ブリッジミュート制御プログラムのプログラムコードを示す。

'変数宣言

'軸回転の変数

Dim DevInfo As MTRDEVICE

Dim Motion As MTRMOTION

Dim lpszName As String

Dim lpszName2 As String

Dim nRet As Long '関数の実行結果

Dim DH As String 'デバイスハンドル選択

Dim a As Long

Dim b As Long

Dim c As Long

Dim d As Long

Dim e As Long

Dim f As Long

Dim i As Integer

'位置カウンタの変数

Dim nPos As Long

Dim nPos1 As Long

'コールバックの変数

Dim Mask As MTREVENTTABLE

Dim pEvent As MTREVENTREQ

Dim pEvent2 As MTREVENTREQ

'エラー検出

Option Explicit

'ソレノイド(PCI-2703A)

Dim hDevicehandle As Long

Dim ipszName As String 'デバイス名

```
Dim pnBuffer(32) As Long      '状態を設定するバッファ
```

```
Private Sub Command1_Click()
```

```
'ボードの初期化処理-----
```

```
ipszName = "FBIDIO1" & Chr(0)
```

```
hDevicehandle = DioOpen(ipszName, FBIDIO_FLAG_SHARE)
```

```
If hDevicehandle = -1 Then
```

```
    MsgBox ("デバイスのオープンに失敗しました")
```

```
    Exit Sub
```

```
End If
```

```
'ツレノイドの選択-----
```

```
    If Check1.Value = 1 Then
```

```
        pnBuffer(0) = 0
```

```
        nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 13, 1)
```

```
    Else
```

```
        pnBuffer(0) = 1
```

```
        nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 13, 1)
```

```
    End If
```

```
    If Check2.Value = 1 Then
```

```
        pnBuffer(0) = 0
```

```
        nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 15, 1)
```

```
    Else
```

```
        pnBuffer(0) = 1
```

```
        nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 15, 1)
```

```
    End If
```

```

'終了処理-----
If nRet <> 0 Then
    MsgBox ("データの入力に失敗しました")
    nRet = DioClose(hDevicehandle)
    Exit Sub
End If

'終了処理
nRet = DioClose(hDevicehandle)
If nRet <> 0 Then
    MsgBox ("デバイスのクローズに失敗しました")
End If
End Sub

Private Sub Command2_Click()
'ボードの初期化処理-----
ipszName = "FBIDIO1" & Chr(0)
hDevicehandle = DioOpen(ipszName, FBIDIO_FLAG_SHARE)

If hDevicehandle = -1 Then
    MsgBox ("デバイスのオープンに失敗しました")
    Exit Sub
End If

'ソレノイドの電流を切る
pnBuffer(0) = 1
nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 13, 1)

pnBuffer(0) = 1
nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 15, 1)
'終了処理-----
If nRet <> 0 Then
    MsgBox ("データの入力に失敗しました")
    nRet = DioClose(hDevicehandle)

```

```
Exit Sub
End If
```

```
'終了処理
```

```
nRet = DioClose(hDevicehandle)
If nRet <> 0 Then
    MsgBox ("デバイスのクローズに失敗しました")
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
'ミュートの選択
```

```
If Option1.Value = True Then
```

```
    Check1.Value = 1
```

```
    Check2.Value = 1
```

```
End If
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
'ミュートを解除する
```

```
    Check1.Value = 0
```

```
    Check2.Value = 0
```

```
'ミュートの解除を解除する
```

```
Option1.Value = False
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub
```

第 8 章

演奏プログラム

自動演奏ギターの動作確認を行うために、ステッピングモータの動作確認プログラムを作成した。プログラムは参考資料に記載する。

ミュートは簡単に操作できるように、作成したプログラムを前年度の操作画面に追加した。ミュートの使用時と未使用時において、聴覚的に違いがでるか調べるため、まずミュートを自由に ON、OFF 出来るようなプログラムを作成し、曲を演奏することでミュート装置を使用したとき変化が出るか調べた。演奏プログラムを図 8.1 に示す。

結果として弦を開放した状態とミュート装置を作動させた状態を演奏した所音の違いがはっきりと聞き取れ、ブリッジミュート奏法を行えていることが確認できた。また、新たに作成しブリッジミュートを使用した曲、前年度までの演奏をブリッジミュートを混ぜて演奏した曲のプログラムコードを以下に記載する。

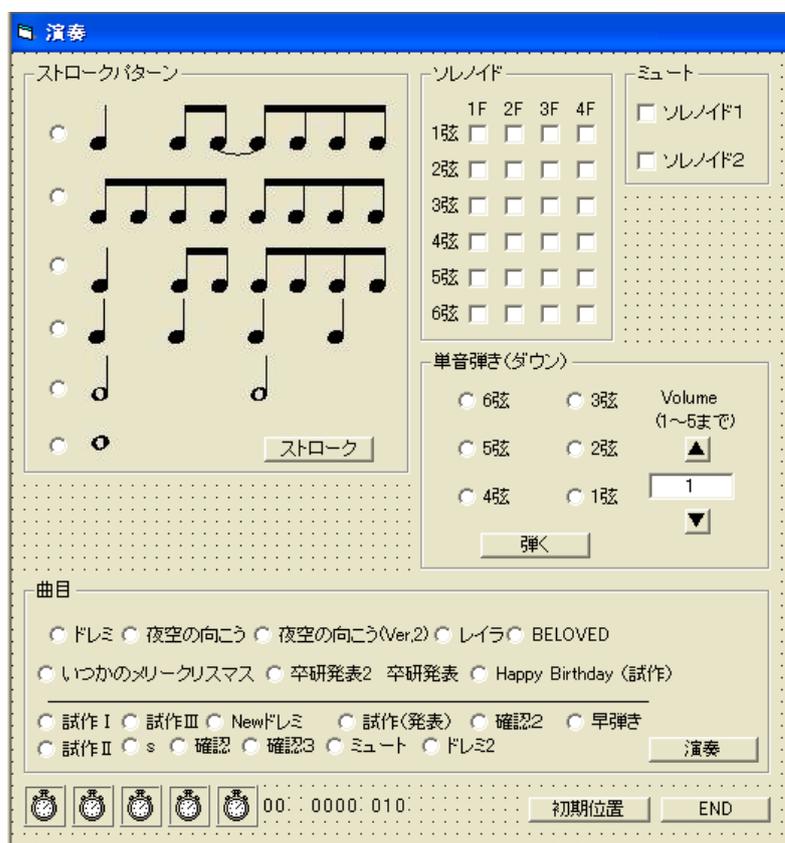


図 8.1 演奏プログラム

演奏プログラムのプログラムコードを示す。

'ドレミ演奏

```
ElseIf Option25.Value = True Then
  Select Case Val(lblPlay.Caption)
    Case 1
      Call Solo1To2
      lblTiming.Caption = 0
    Case 3
      Call PickDown1
      Call Solo2To3
      Check17.Value = 1
      lblTiming.Caption = 0
    Case 4
      Call PickUp1
      lblTiming.Caption = 0
    Case 7
      Call Solo3To4
      Call PickDown01
      lblTiming.Caption = 0
    Case 8
      Check17.Value = 0
      Call PickUp01
      lblTiming.Caption = 0
    Case 11
      Call Solo4To9
      Check10.Value = 1
      Call PickDown4
      lblTiming.Caption = 0
    Case 12
      Call PickUp4
      lblTiming.Caption = 0
    Case 15
      Check16.Value = 1
```

Call Solo9To4
Call PickDown3
Check10.Value = 0
lblTiming.Caption = 0

Case 16
Call PickUp3
lblTiming.Caption = 0

Case 19
Call Solo4To5
Call PickDown1
Check16.Value = 0
lblTiming.Caption = 0

Case 20
Call PickUp1
lblTiming.Caption = 0

Case 23
Call Solo5To10x
Check9.Value = 1
Call PickDown1
Check16.Value = 0
lblTiming.Caption = 0

Case 24
Call PickUp1
lblTiming.Caption = 0

Case 27
Gen2
Check9.Value = 0
lblTiming.Caption = 0

Case 31
Call Solo6To11
Check2.Value = 1
Call PickDown001
lblTiming.Caption = 0

Case 32
Call PickUp001
lblTiming.Caption = 0

```
Case 50
    Check2.Value = 0
    lblTiming.Caption = 0
End Select
```

```
-----
'キラキラ星
-----
```

```
ElseIf Option33.Value = True Then
    Select Case Val(lblPlay.Caption)
```

```
Case 1
    Call Gen5
    Check14.Value = 1
    lblTiming.Caption = 0
```

```
Case 6
    Call Gen5
    lblTiming.Caption = 0
```

```
Case 11
    Call Gen6
    Check14.Value = 0
    Check13.Value = 1
    lblTiming.Caption = 0
```

```
Case 16
    Call Gen6
    lblTiming.Caption = 0
```

```
Case 21
    Call Gen5
    Check13.Value = 0
    lblTiming.Caption = 0
```

Case 26
Call Gen5
lblTiming.Caption = 0

Case 31
Call Gen6
Check13.Value = 1
lblTiming.Caption = 0

Case 41
Call Gen6
Check13.Value = 0
Check11.Value = 1
lblTiming.Caption = 0

Case 46
Call Gen6
lblTiming.Caption = 0

Case 51
Call Gen6
Check11.Value = 0
lblTiming.Caption = 0

Case 56
Call Gen6
lblTiming.Caption = 0

Case 61
Call Gen4
lblTiming.Caption = 0

Case 66
Call Gen4
lblTiming.Caption = 0

Case 71

Call Gen5

Check14.Value = 1

lblTiming.Caption = 0

Case 81

Call Gen6

Check14.Value = 0

Check13.Value = 1

lblTiming.Caption = 0

Case 86

Call Gen6

lblTiming.Caption = 0

Case 91

Call Gen6

Check13.Value = 0

Check11.Value = 1

lblTiming.Caption = 0

Case 96

Call Gen6

lblTiming.Caption = 0

Case 101

Call Gen6

Check11.Value = 0

lblTiming.Caption = 0

Case 106

Call Gen6

lblTiming.Caption = 0

Case 111

Call Gen4
lblTiming.Caption = 0

Case 121
Call Gen6
Check13.Value = 1
lblTiming.Caption = 0

Case 126
Call Gen6
lblTiming.Caption = 0

Case 131
Call Gen6
Check13.Value = 0
Check11.Value = 1
lblTiming.Caption = 0

Case 136
Call Gen6
lblTiming.Caption = 0

Case 141
Call Gen6
Check11.Value = 0
lblTiming.Caption = 0

Case 146
Call Gen6
lblTiming.Caption = 0

Case 151
Call Gen4
lblTiming.Caption = 0

Case 161

Call Gen2
Check25.Value = 1
lblTiming.Caption = 0

Case 166
Call Gen2
lblTiming.Caption = 0

Case 171
Call Gen3
Check25.Value = 0
lblTiming.Caption = 0

Case 176
Call Gen3
lblTiming.Caption = 0

Case 181
Call Gen3
Check20.Value = 0
lblTiming.Caption = 0

Case 186
Call Gen3
lblTiming.Caption = 0

Case 191
Call Gen3
lblTiming.Caption = 0

Case 201
Call Gen1
Check26.Value = 1
lblTiming.Caption = 0

Case 206

```
Call Gen1
lblTiming.Caption = 0
```

```
Case 211
Call Gen1
Check26.Value = 0
lblTiming.Caption = 0
```

```
Case 216
Call Gen1
lblTiming.Caption = 0
```

```
Case 221
Call Gen2
Check29.Value = 1
lblTiming.Caption = 0
```

```
Case 226
Call Gen2
lblTiming.Caption = 0
```

```
Case 231
Call Gen2
Check29.Value = 0
Check25.Value = 1
lblTiming.Caption = 0
```

```
End Select
```

```
-----  
'カエルの歌  
-----
```

```
If Option19.Value = True Then  
    Select Case Val(lblPlay.Caption)
```

Case 1

Call Gen5

Check17.Value = 1

lblTiming.Caption = 0

Case 4

Call Gen4

Check17.Value = 0

lblTiming.Caption = 0

Case 7

Call Gen4

Check10.Value = 1

lblTiming.Caption = 0

Case 10

Call Gen4

Check10.Value = 0

Check16.Value = 1

lblTiming.Caption = 0

Case 13

Call Gen4

Check16.Value = 0

Check10.Value = 1

lblTiming.Caption = 0

Case 16

Call Gen4

Check10.Value = 0

lblTiming.Caption = 0

Case 19

Call Gen5

Check17.Value = 1

lblTiming.Caption = 0

Case 24

Call Gen5

Check17.Value = 1

Check25.Value = 1

Check26.Value = 1

lblTiming.Caption = 0

Case 27
 Call Gen4
 Check17.Value = 0
 lblTiming.Caption = 0

Case 30
 Call Gen4
 Check10.Value = 1
 lblTiming.Caption = 0

Case 33
 Call Gen4
 Check10.Value = 0
 Check16.Value = 1
 lblTiming.Caption = 0

Case 36
 Call Gen4
 Check16.Value = 0
 Check10.Value = 1
 lblTiming.Caption = 0

Case 39
 Call Gen4
 Check10.Value = 0
 lblTiming.Caption = 0

Case 42
 Call Gen5
 Check17.Value = 1
 lblTiming.Caption = 0

Case 47
 Call Gen5
 Check17.Value = 1
 lblTiming.Caption = 0

Case 52
 Call Gen5
 Check17.Value = 1
 lblTiming.Caption = 0

Case 57
 Call Gen5

Check17.Value = 1
lblTiming.Caption = 0

Case 62
Call Gen5
Check17.Value = 1
lblTiming.Caption = 0

Case 67
Call Gen5
Check17.Value = 1
Check25.Value = 0
Check26.Value = 0
lblTiming.Caption = 0

Case 69
Call Gen5
Check17.Value = 1
lblTiming.Caption = 0

Case 71
Call Gen4
Check17.Value = 0
lblTiming.Caption = 0

Case 73
Call Gen4
lblTiming.Caption = 0

Case 75
Call Gen4
Check10.Value = 1
lblTiming.Caption = 0

Case 77
Call Gen4
Check10.Value = 1
lblTiming.Caption = 0

Case 79
Call Gen4
Check10.Value = 0
Check16.Value = 1
lblTiming.Caption = 0

Case 81
 Call Gen4
 Check16.Value = 1
 lblTiming.Caption = 0

Case 83
 Call Gen4
 Check16.Value = 0
 Check10.Value = 1
 lblTiming.Caption = 0

Case 86
 Call Gen4
 Check10.Value = 0
 lblTiming.Caption = 0

Case 89
 Call Gen5
 Check17.Value = 1
 lblTiming.Caption = 0

End Select

第 9 章

結言

結言

- (1) 幅広いミュート奏法を行うことができるようになった。
- (2) 弦振動時の周波数成分を可視化できるようになった。
- (3) 曲の演奏にもブリッジミュートが組み込めるようになった。

今後の課題

- (1) ブリッジミュートを用いたストロークによるコードの演奏。
- (2) 各弦を弾いたときの音量、時間の測定とその場合の周波数解析を行う。
- (3) 演奏が単調なため、人間と同じような斜めからのストローク動作を行えるようにピックを回転させるなどの改良を行う必要がある。

謝辞

本研究に当たり数々のご指導、ご助言をして頂いた河村良行教授、ブリッジミュート装置を加工するにあたりご指導、ご助言を頂いた工作センターの先生方、本当にありがとうございました。

参考文献

(1) http://www.me.gatech.edu/mechatronics_lab/Projects/Fall00/group3/photo.htm

(2) 白土義男, “トランジスタ回路活用のポイント” 日本放送出版協会

(3) <http://www.cycleof5th.com/products/soundengine/>

(4) 三谷政昭, “信号解析のための数学” P,128~141[フーリエ変換] 森北出版

付録

付録 2. 動作確認プログラム

①ピック 2 ストローク：ストロークの速度計測に使用
(Form1)

```
If lblSt.Caption = 1 Then
    Call PickDown
    Call Arm5GDown
    lblSt.Caption = 2
ElseIf lblSt.Caption = 3 Then
    Call PickUp
    Call Arm6GDown
    lblSt.Caption = 4
ElseIf lblSt.Caption = 5 Then
    Call PickDown
    Call Arm6GDown
    lblSt.Caption = 6
ElseIf lblSt.Caption = 7 Then
    Call PickUp
    Call Arm6GDown
    lblSt.Caption = 8
ElseIf lblSt.Caption = 9 Then
    Call PickDown
    Call Arm6GDown
    lblSt.Caption = 10
ElseIf lblSt.Caption = 11 Then
    Call PickUp
    lblSt.Caption = 12
ElseIf lblSt.Caption = 13 Then
    Call PickDown
    Call Arm6GUp
    lblSt.Caption = 14
ElseIf lblSt.Caption = 15 Then
    Call PickUp
    Call Arm6GUp
    lblSt.Caption = 16
ElseIf lblSt.Caption = 17 Then
    Call PickDown
```

```
    Call Arm6GUp
    lblSt.Caption = 18
ElseIf lblSt.Caption = 19 Then
    Call PickUp
    Call Arm6GUp
    lblSt.Caption = 20
ElseIf lblSt.Caption = 21 Then
    Call PickDown
    Call Arm5GUp
    lblSt.Caption = 22
ElseIf lblSt.Caption = 23 Then
    Call PickUp
    lblSt.Caption = 24
ElseIf lblSt.Caption = 25 Then
    Call PickDown
    Call Arm5GDown
    lblSt.Caption = 26
ElseIf lblSt.Caption = 27 Then
    Call PickUp
    Call Arm6GDown
    lblSt.Caption = 28
ElseIf lblSt.Caption = 29 Then
    Call PickDown
    Call Arm6GDown
    lblSt.Caption = 30
ElseIf lblSt.Caption = 31 Then
    Call PickUp
    Call Arm6GDown
    lblSt.Caption = 32
ElseIf lblSt.Caption = 33 Then
    Call PickDown
    Call Arm6GDown
    lblSt.Caption = 34
ElseIf lblSt.Caption = 35 Then
    Call PickUp
    lblSt.Caption = 36
```

```

ElseIf lblSt.Caption = 37 Then
    Call PickDown
    Call Arm6GUp
    lblSt.Caption = 38
ElseIf lblSt.Caption = 39 Then
    Call PickUp
    Call Arm6GUp
    lblSt.Caption = 40
ElseIf lblSt.Caption = 41 Then
    Call PickDown
    Call Arm6GUp
    lblSt.Caption = 42
ElseIf lblSt.Caption = 43 Then
    Call PickUp
    Call Arm6GUp
    lblSt.Caption = 44
ElseIf lblSt.Caption = 45 Then
    Call PickDown
    Call Arm5GUp
    lblSt.Caption = 46
ElseIf lblSt.Caption = 47 Then
    Call PickUp
    lblSt.Caption = 0

```

②6,5,3,1 弦のストロークプログラム : ストロークプログラムの理解に使用
(Form1)

```

If lblSt.Caption = 1 Then
    Call PickDown
    Call Arm5GDown
    lblSt.Caption = 2
ElseIf lblSt.Caption = 3 Then
    Call PickUp
    Call Arm6GDown

```

```
    lblSt.Caption = 4
ElseIf lblSt.Caption = 5 Then
    Call PickDown
    Call Arm6GDown
    lblSt.Caption = 6
ElseIf lblSt.Caption = 7 Then
    Call PickUp
    Call Arm6GDown
    lblSt.Caption = 8
ElseIf lblSt.Caption = 9 Then
    Call PickDown
    Call Arm6GDown
    lblSt.Caption = 10
ElseIf lblSt.Caption = 11 Then
    Call PickUp
    lblSt.Caption = 12
ElseIf lblSt.Caption = 13 Then
    Call PickDown
    Call Arm6GUp
    lblSt.Caption = 14
ElseIf lblSt.Caption = 15 Then
    Call PickUp
    Call Arm6GUp
    lblSt.Caption = 16
ElseIf lblSt.Caption = 17 Then
    Call PickDown
    Call Arm6GUp
    lblSt.Caption = 18
ElseIf lblSt.Caption = 19 Then
    Call PickUp
    Call Arm6GUp
    lblSt.Caption = 20
ElseIf lblSt.Caption = 21 Then
    Call PickDown
    Call Arm5GUp
    lblSt.Caption = 22
```

```
ElseIf lblSt.Caption = 23 Then
    Call PickUp
    lblSt.Caption = 24
```

③2 ストローク（強 20 弱 13）：ピック入れ込み深さによる音量の変化に使用

```
If lblSt.Caption = 1 Then
    Call PickDown2
    Call Arm5GDown
    lblSt.Caption = 2
ElseIf lblSt.Caption = 3 Then
    Call PickUp2
    Call Arm6GDown
    lblSt.Caption = 4
ElseIf lblSt.Caption = 5 Then
    Call PickDown01
    Call Arm6GDown
    lblSt.Caption = 6
ElseIf lblSt.Caption = 7 Then
    Call PickUp01
    Call Arm6GDown
    lblSt.Caption = 8
ElseIf lblSt.Caption = 9 Then
    Call PickDown2
    Call Arm6GDown
    lblSt.Caption = 10
ElseIf lblSt.Caption = 11 Then
    Call PickUp2
    lblSt.Caption = 12
ElseIf lblSt.Caption = 13 Then
    Call PickDown2
    Call Arm6GUp
    lblSt.Caption = 14
ElseIf lblSt.Caption = 15 Then
```

```

    Call PickUp2
    Call Arm6GUp
    lblSt.Caption = 16
ElseIf lblSt.Caption = 17 Then
    Call PickDown01
    Call Arm6GUp
    lblSt.Caption = 18
ElseIf lblSt.Caption = 19 Then
    Call PickUp01
    Call Arm6GUp
    lblSt.Caption = 20
ElseIf lblSt.Caption = 21 Then
    Call PickDown2
    Call Arm5GUp
    lblSt.Caption = 22
ElseIf lblSt.Caption = 23 Then
    Call PickUp2
    lblSt.Caption = 24

```

④点灯プログラム：LED点灯実験回路に使用

```

'-----
'変数宣言
'-----

```

```

'コールバックの変数
Dim Mask As MTREVENTTABLE
Dim pEvent As MTREVENTREQ
Dim pEvent2 As MTREVENTREQ

```

```

'エラー検出
Option Explicit

```

```

'ソレノイド(PCI-2703A)
Dim hDevicehandle As Long

```

```
Dim ipzName As String      'デバイス名
Dim pnBuffer(32) As Long  '状態を設定するバッファ
```

```
Private Sub Check1_Click()
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
'ボードの初期化処理-----
```

```
ipzName = "FBIDIO1" & Chr(0)
```

```
hDevicehandle = DioOpen(ipzName, FBIDIO_FLAG_SHARE)
```

```
If hDevicehandle = -1 Then
```

```
    MsgBox ("デバイスのオープンに失敗しました")
```

```
    Exit Sub
```

```
End If
```

```
'ツレノイドの選択-----
```

```
    If Check1.Value = 1 Then
```

```
        pnBuffer(0) = 0
```

```
        nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 9, 1)
```

```
    Else
```

```
        pnBuffer(0) = 1
```

```
        nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 9, 1)
```

```
    End If
```

```
'終了処理-----
```

```
If nRet <> 0 Then
```

```
    MsgBox ("データの入力に失敗しました")
```

```
    nRet = DioClose(hDevicehandle)
```

```
    Exit Sub
```

End If

'終了処理

nRet = DioClose(hDevicehandle)

If nRet <> 0 Then

MsgBox ("デバイスのクローズに失敗しました")

End If

End Sub

Private Sub Command2_Click()

'ボードの初期化処理-----

ipszName = "FBIDIO1" & Chr(0)

hDevicehandle = DioOpen(ipszName, FBIDIO_FLAG_SHARE)

If hDevicehandle = -1 Then

MsgBox ("デバイスのオープンに失敗しました")

Exit Sub

End If

'ツレノイドの電流を切る

pnBuffer(0) = 1

nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 9, 1)

'終了処理-----

If nRet <> 0 Then

MsgBox ("データの入力に失敗しました")

nRet = DioClose(hDevicehandle)

Exit Sub

End If

'終了処理

nRet = DioClose(hDevicehandle)

If nRet <> 0 Then

MsgBox ("デバイスのクローズに失敗しました")

End If

End Sub