触覚センサによる筋肉の硬さ変化測定の試み

知能機械システム工学科

トライボロジー研究室

指導教官：竹内

1020118

高村哲農

目次

１．研究背景

本研究の背景として、現在、我国の医療現場では少子高齢化社会の影響を受けて、慢性的な介護者不足が問題となっている。また、介護現場における介護者の高齢化も大きな問題である。介護者が高齢化することにより、肉体労働が大半を占める介護現場はあまりに過酷で高齢者には困難である。また介護を受ける側の介護要求は時間を問わず、介護者に肉体的、精神的にも多大な負担を強いるのである。そこで、高齢者がある程度の用件を介護者を呼ぶ事なく自分で解決できるならば、介護者にかかる労働負担をいくらか軽減できる可能性が考えられる。そのためには、歩行支援機、起立支援機など介護支援機の安全性の向上が必要不可欠といえる。そこで、立ち上がりや歩行の際の下半身における筋肉の作動状態を把握できれば、リハビリの事前に筋肉疲労や、ケイレンといった危険な状態を察知できる可能性があり、患者のリハビリ中の怪我などを未然に防げると考えられる。また、患者にとっても個人のプライバシーが守られることにより、自信にもつながり、行動力も向上することで健康面も向上し、更なる長寿命化も期待できる。

近年、筋肉の作動状態を把握する目的で開発されたセンサもいくつかあるが、代表的なものとして、筋肉の活動電位を測定する筋電計があげられる（図 1)。しかし、筋電計は筋肉が瞬発的に作動する時は感度よく筋肉の活動電位を捉えられるが、筋肉が持続的に緊張しているかどうかを測定することは難しいとされている。また、最近では密封されたエアの圧力変化により対象物の硬さ変化を捉えられるとされるエアパック式センサ（図 2）も開発されたが、エアパック式センサは粘着テープのようなものでセンサ部を押し付ける構造のため、表面変位の影響が大きく、筋肉の硬さ変化との対応がつきにくいという一面もある。また両システムは大変高価であるという難点があり、立ち上がり支援機などの介護用機器といった一つのシステムに取り付けるには高すぎるという問題がある。

より安価で手軽に筋肉の硬さ変化が測定できるセンサが開発できれば、歩行支援機など介護用機器の制御系に組み込むことが可能で、リハビリの効率化、安全性の向上に寄与できると考えられる。

そのために本研究では触覚センサによる筋肉の硬さ変化測定の可能性を検討した。

図 1.筋電計

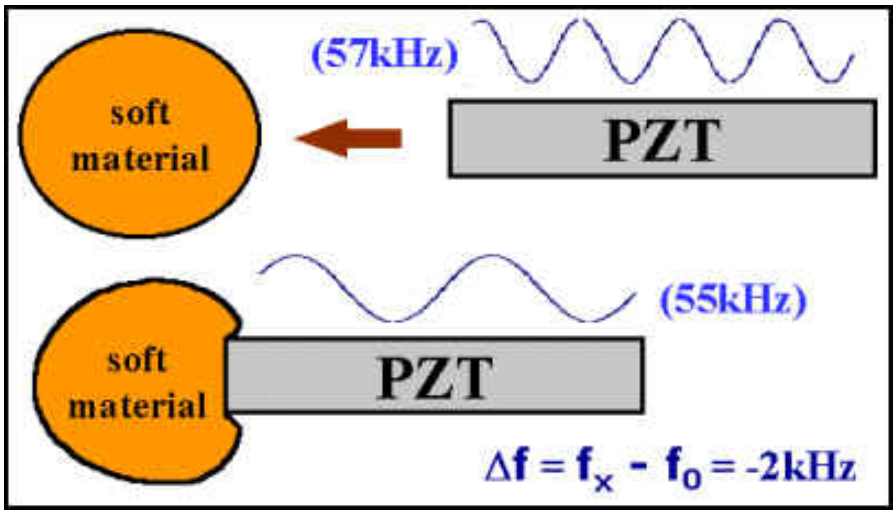
図 2.エアパック式センサ

２．触覚センサの基本原理

物にはそれぞれ固有の振動数があり、硬いものは高い固有振動数をもち、軟らかいものは低い固有振動数をもっている。コップ単体をたたくと高い音がする。今度はコップにゴムを貼り付けてたたくと、そのとき音は低くなる。

この音が低くなる理由はコップ＋ゴムの固有振動数が下がる為である。そこでコップ単体の際の振動数を f0 とし、ゴムを貼り付けた際の振動数を fx とした場合、その差 Df はゴムの振動特性を表している。

実際の触覚センサはコップではなく圧電素子をもちいる。



圧電素子は交流電圧を負荷すると振動する。逆に振動が加えられると交流電圧が発生するという特性がある。そこで、駆動用の圧電素子と検出用圧電素子を組み合わせ、検出用素子から出力された振動の変化をとらえ、帰還回路を通じて駆動素子に帰還させると同時にその変化を出力する。

そして、センサ先端部を一定の接触圧で物質に押し当てたとき、この物質が十分に硬ければ接触面積の増大が少なく、周波数変化量はスティフネス効果となり、また軟らかい場合は接触面積が増大して質量効果となるので、このときの周波数変化を求めることで物質の硬さ、軟らかさをリアルタイムで検出することが出来るのである。

３．実験装置

　システムの基本構成を図３に示す。
実験装置は、触覚センサ、アンプ、周波数カウンタ、パーソナルコンピュータにより構成されており、センサ部には駆動用と検出用の２つの圧電素子が用いられている。センサ先端のナイロン球が相手物質に触れることにより発振回路の周波数変化が生じ、その変化量から対象物の硬さが相対的に判別できるという仕組みである。

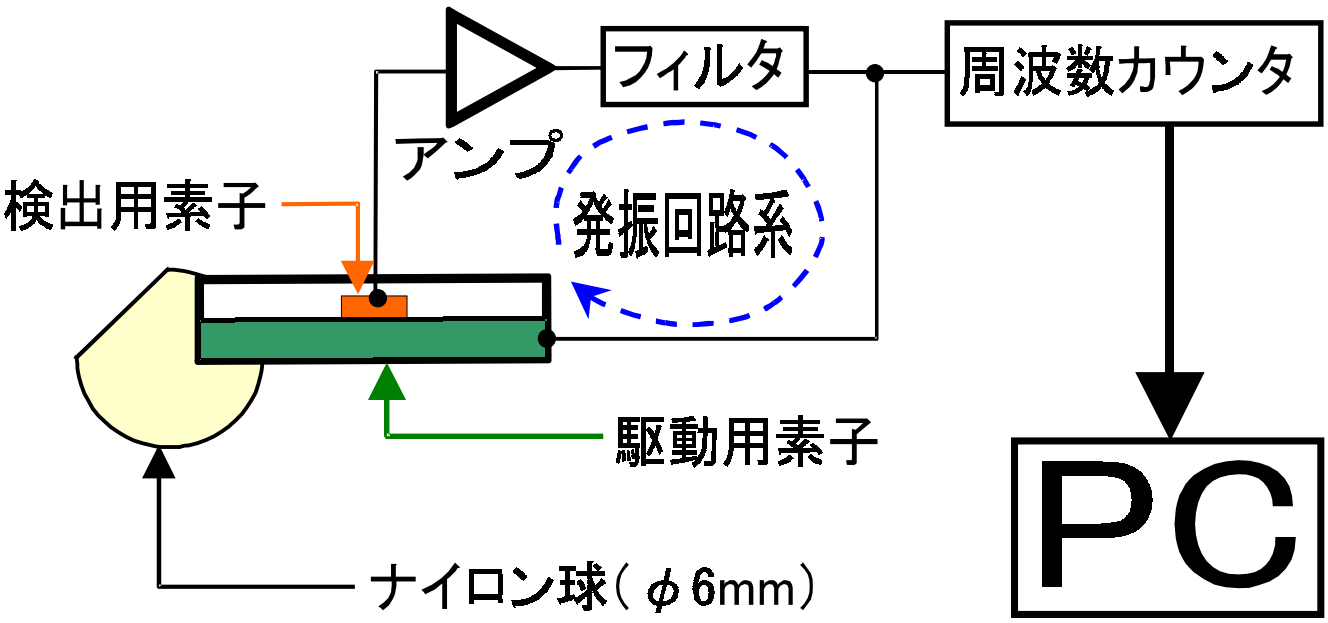


図３.触覚センサ系の基本構成図

４．実験方法

本実験では、触覚センサによる筋肉の硬さ測定を前腕部前面、後下腿部について測定した。前腕部前面の筋肉の硬さ測定における測定点を図４、また測定条件を一定にするためにセンサ固定用治具を作成したので図５に示す。

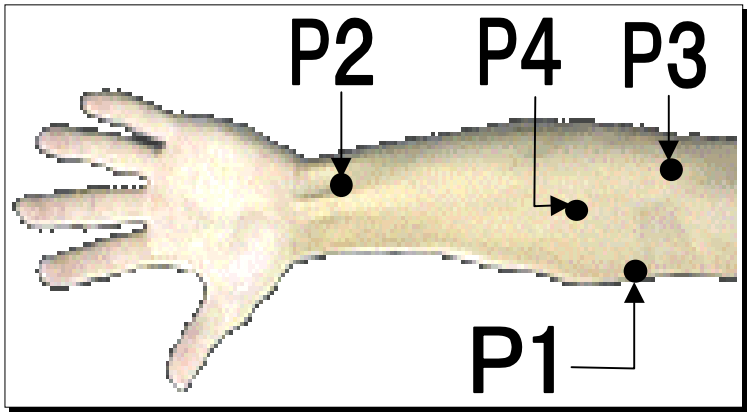


図４.前腕部前面の筋肉の硬さ測定点

P1 は長掌筋の硬さ変化に対応しており、握ることで硬さが著しく変化するとされる部位である。P2 は腕橈骨筋と橈側手根屈筋の間に位置し、硬さ変化は無いとされる点。P3 は腕橈骨筋、P4 は橈側手根屈筋の筋肉の硬さ変化に対応した測定点であり、長掌筋と比べると握り動作においては比較的硬くなりにくいとされる点である。

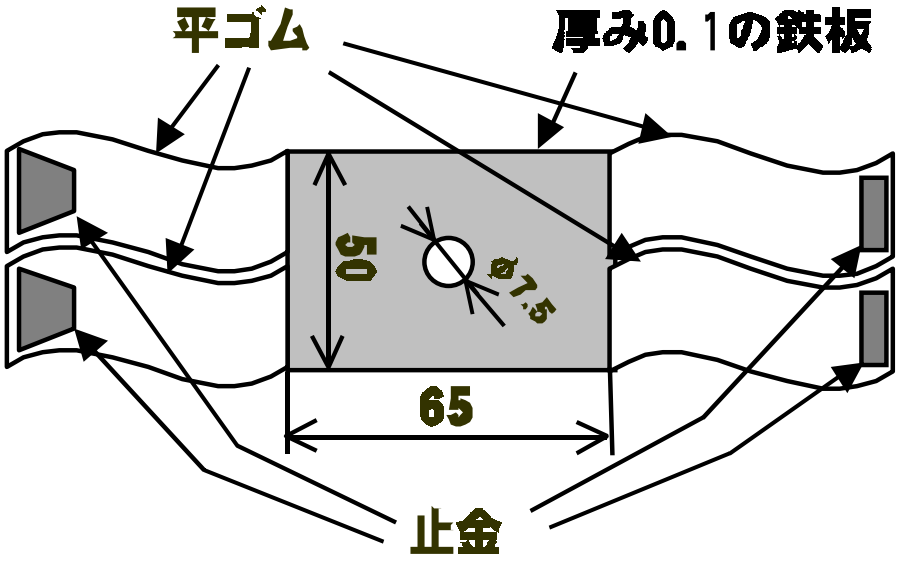


図５.センサ固定用治具

前腕部前面の筋肉の硬さ測定は、腕をテーブルに楽に置き、握力以外の力がかかり難い状態で握力計を握る。触覚センサ固定用治具により P1～P4 のいずれかに取り付ける（図 6）。

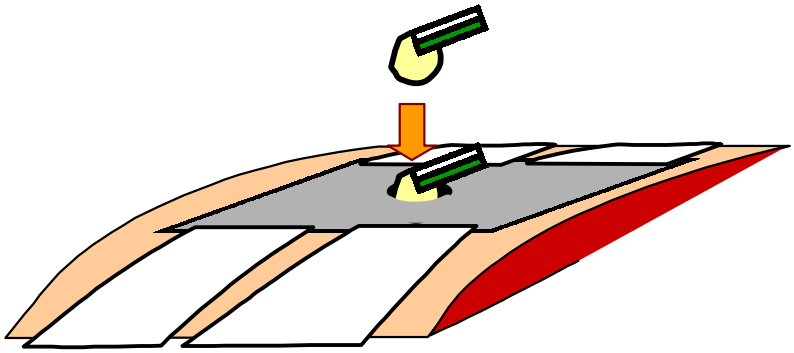
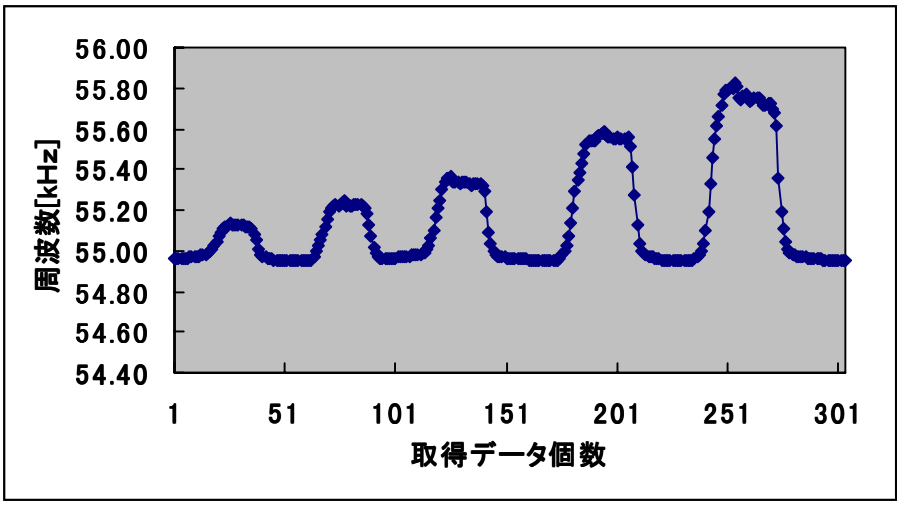


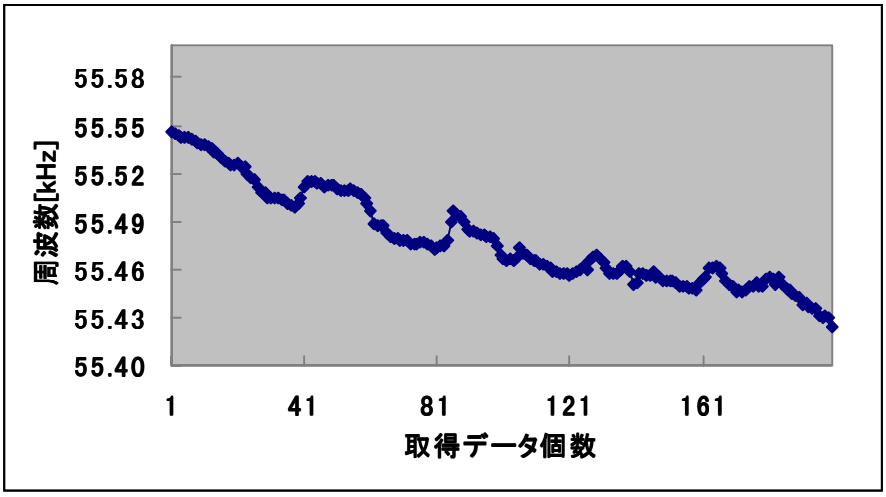
図 6.センサ固定用治具の取り付け状態

固定用治具は対象となる腕にゴムで巻きつけるように固定する。そして、治具中央部の穴から触覚センサの探触子が測定対象の皮膚に乗せられるような状態であり、センサ部を押しつける力を極力一定に保つように工夫されている。

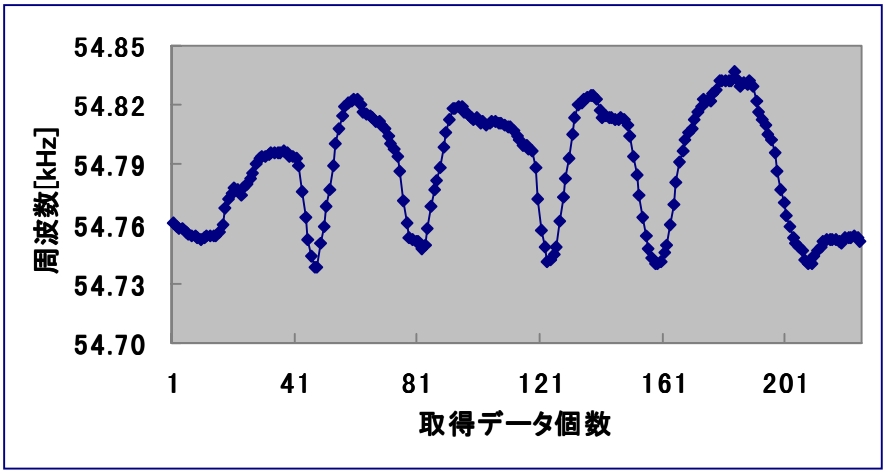
握力計により、50N まで握り、1 秒後に緩める。次に、前回よりも 50N 多い 100N まで握り、1 秒後に緩める。これを 50N，100N，150N，200N，250N と 50N ずつ増やし測定した。各測定点における実測値を以下に示す。



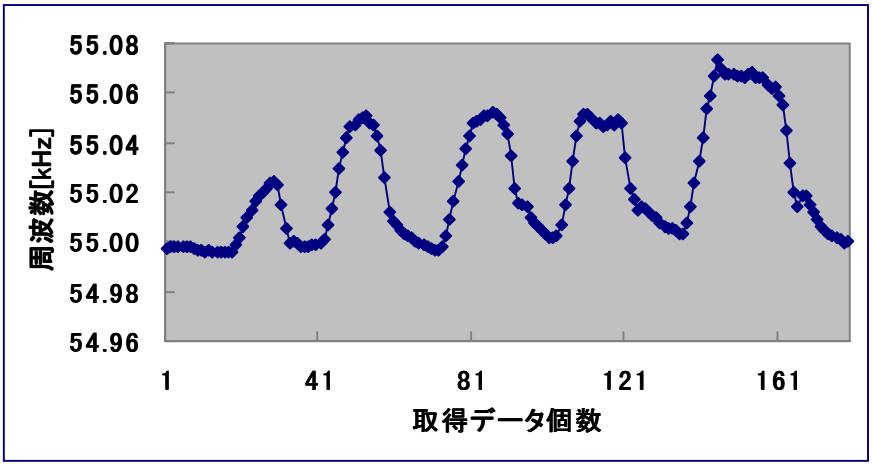
Ｐ１における実測値



Ｐ２における実測値

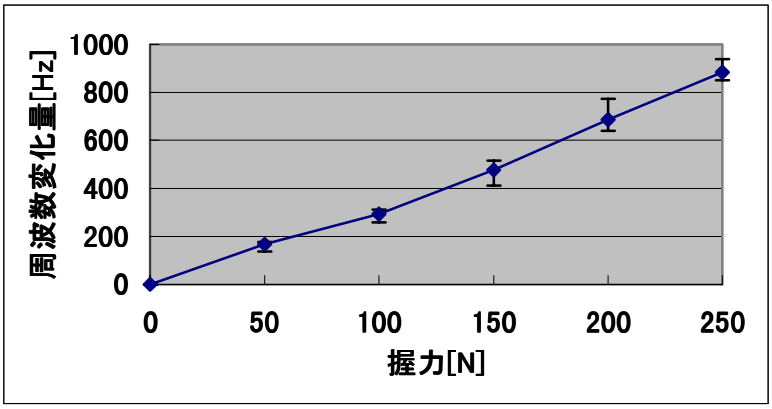


Ｐ３における実測値

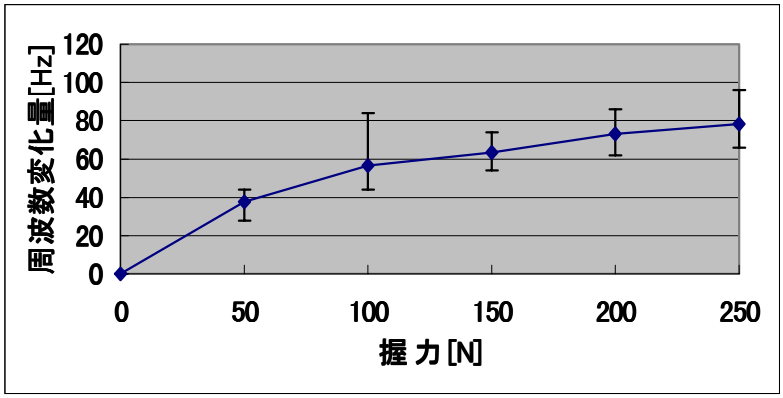


Ｐ４における実測値

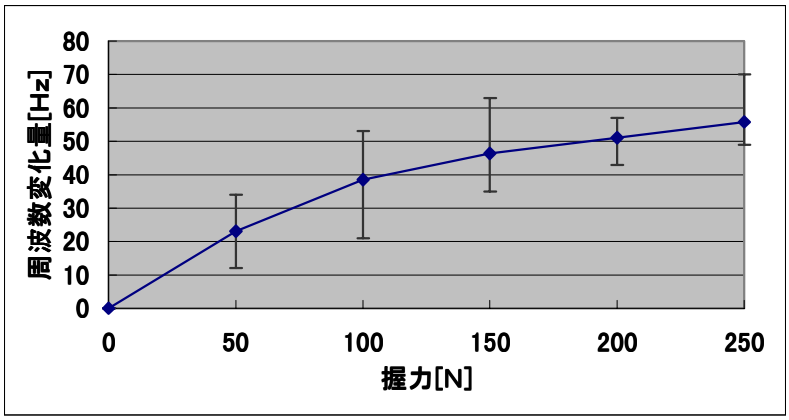
この測定を各測定点ごとに 10 回行い、それらの周波数変化の各負荷に対応したピーク値の平均値を求めたものを以下に示す。なお P2 においては測定結果を見ても分かる通り周波数変化らしきものが得られず硬さ変化が無いと判断した。



Ｐ１における負荷別平均値



Ｐ３における負荷別平均値



Ｐ４における負荷別平均値

このＰ１，Ｐ２，Ｐ３，Ｐ４の各測定点における負荷別の周波数変化量の平均値をまとめたものを図７に示す。なおＰ２における周波数変化量はないものとしている。

５．実験結果

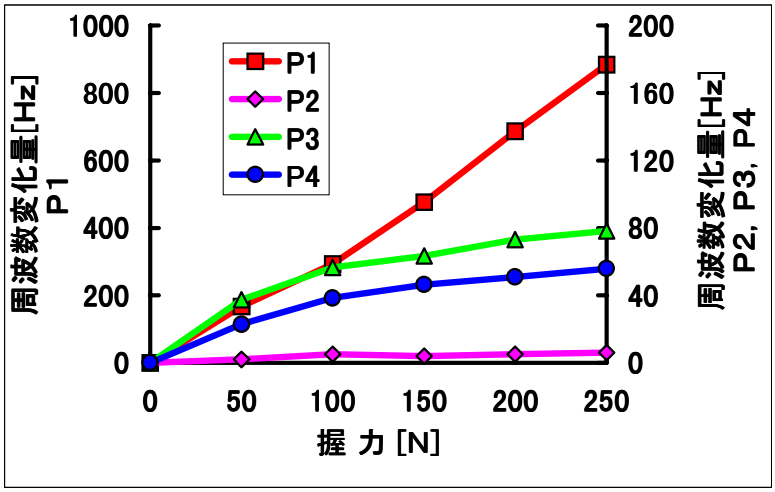


図７．各測定点の握力と周波数変化量の関係

この結果より、P1 における測定結果が、P3、P4 に比べると、周波数変化量が大きく、リニアな傾向を示していることが分かる。

また，触覚センサが筋肉の硬さ変化を捉えている事を確かめるために、Hardness Tester（C 型硬度計）による測定も同様に行った。

Hardness Tester の概略を図８に示す。

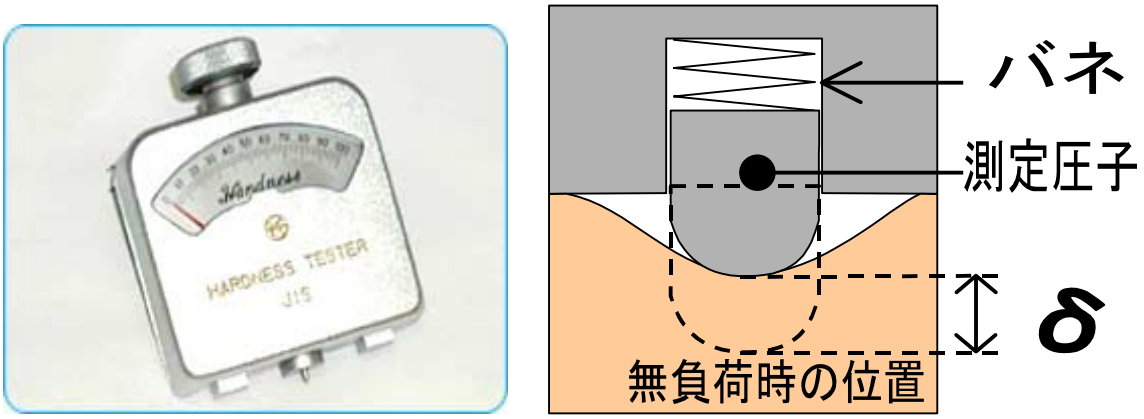
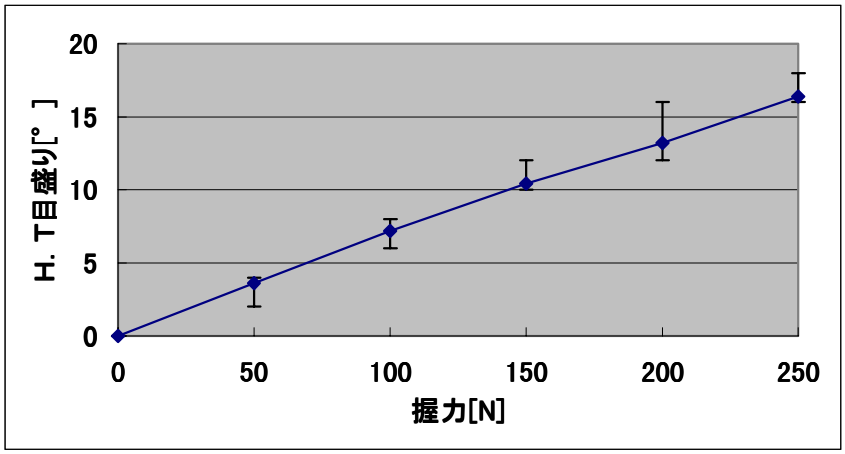


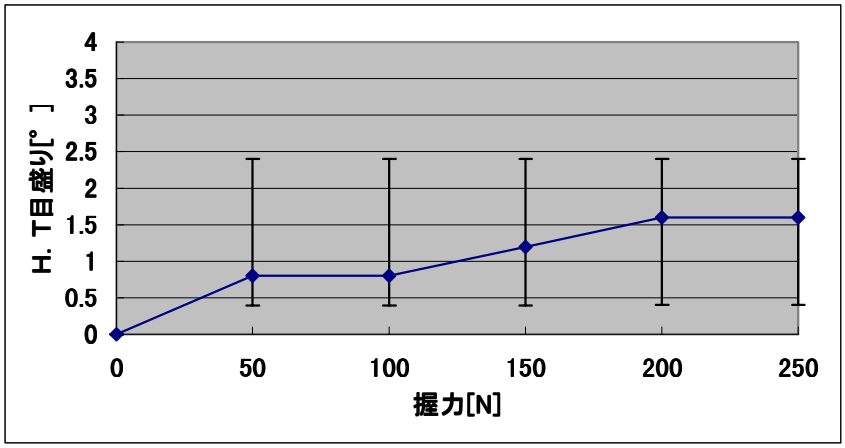
図８.Hardness tester 概略図

Hardness Tester は、バネで押されている測定圧子を対象物に押し当てたときの測定圧子の埋没深さδからその物質の硬さを測定する機器であり、測定に影響する深さは、前述の触覚センサより深いとされている。

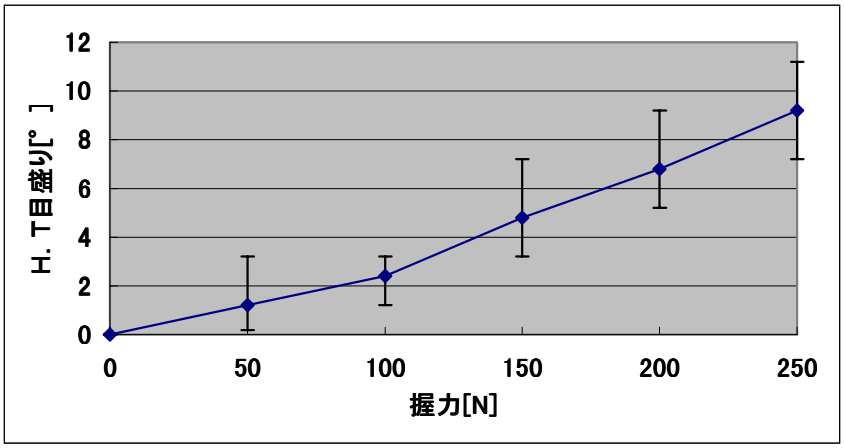
各測定点ごとに 10 回測定した値の平均値を以下に示す



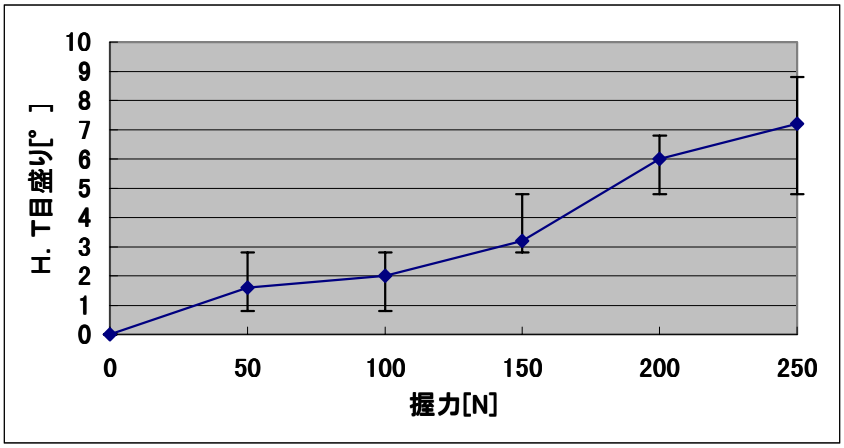
Ｐ１における硬さ変化量の平均値



Ｐ２における硬さ変化量の平均値



Ｐ３における硬さ変化量の平均値



Ｐ４における硬さ変化量の平均値

測定結果をまとめたものを図９に示す。

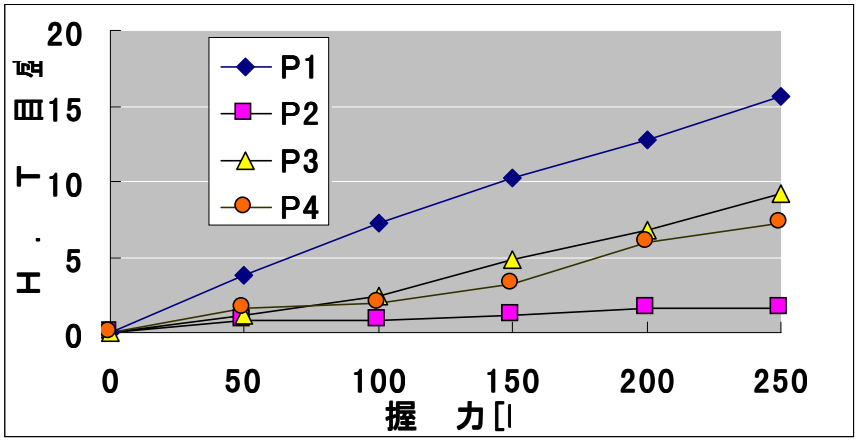


図９．各測定点の握力と硬さ変化量の関係

図７と図９を比較すると、触覚センサの結果ではＰ１における周波数変化量が、もっとも大きくまた、リニアな傾向を示しているのに対してＰ３，Ｐ４は変化量が小さいという傾向が示されている。また、Hardness　Tester による測定結果においてもP1 の硬さ変化がもっとも大きく、リニアな傾向を示すのに対してＰ３，Ｐ４の硬さ変化は小さいというほぼ同様の傾向を示していることが解かる。この結果から、本センサにより測定された周波数変化は、筋肉の硬さ変化に対応するものと判断できる。

次に、応用例として後下腿部の筋肉の硬さ変化を測定した。
後下腿部の測定点を図 10 に、測定方法を図 11 に示す。

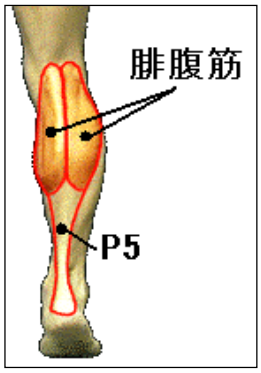


図 10.後下腿部の測定点

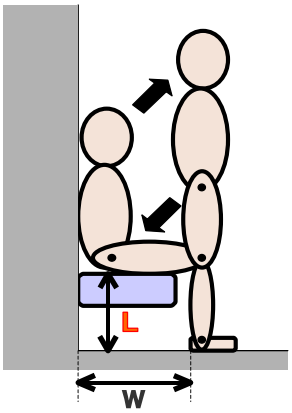
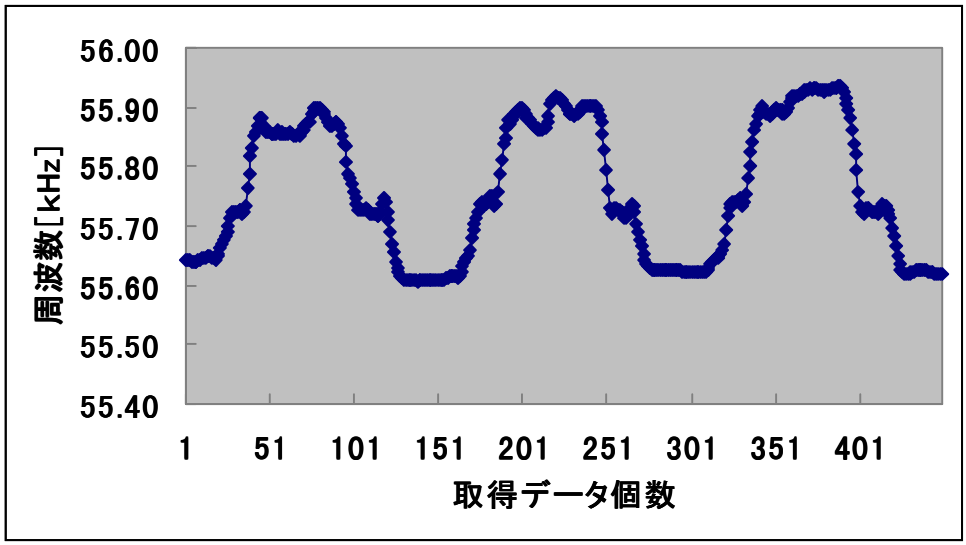
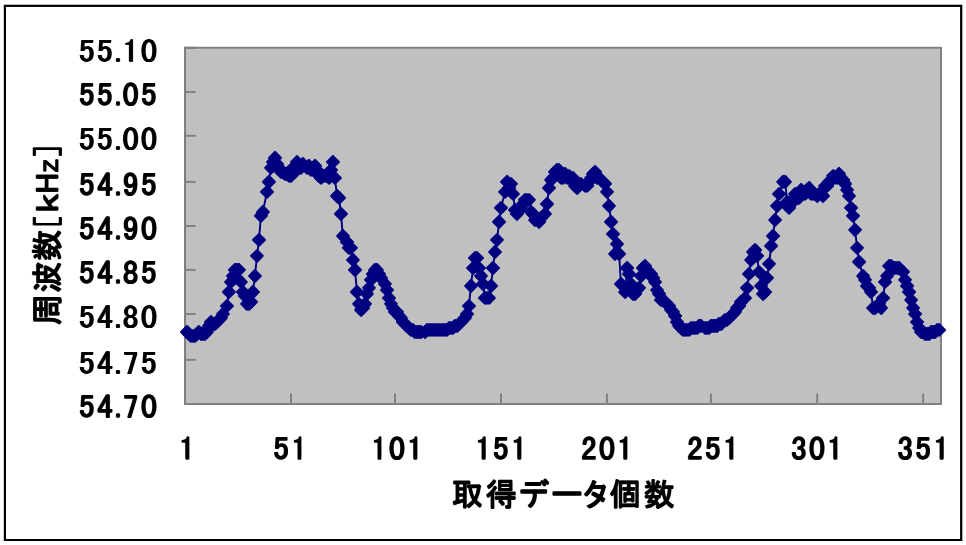


図 11.後下腿部の筋肉の硬さ測定方法

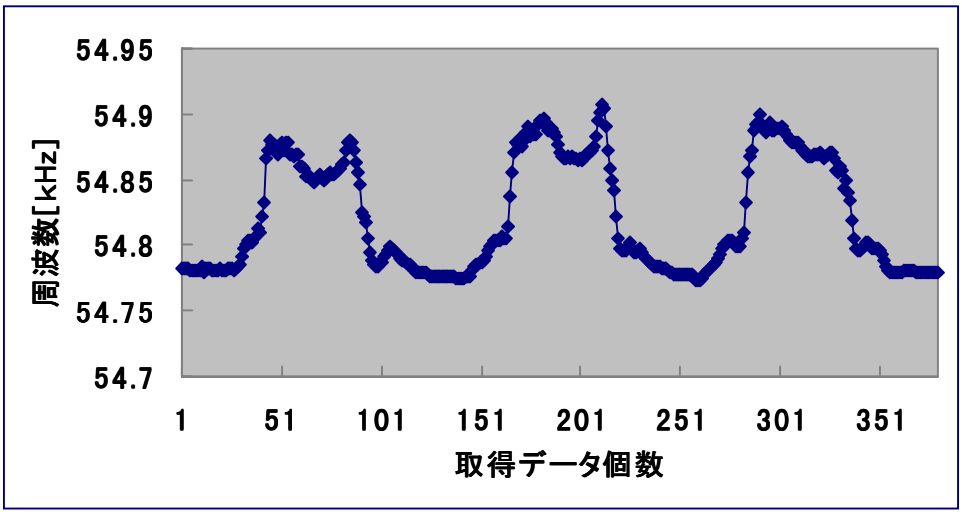
後下腿部の筋肉の硬さ測定は、測定点に、立ち上がり、歩行、持ち上げなど殆どの測定に応用できる腓腹筋の内側頭と外側等がまとまる点に P5 を設ける。椅子に腰掛け、壁から踵までの距離 W を一定とし、椅子の高さ L を 350［mm］、400［mm］、500［mm］の三段階に変化させて何もつかまる事無く立ち上がり、2 秒後に座る。といった一連の動作における筋肉の硬さ変化を測定した。
各高さごとの測定における実測値を以下に示す。



L = 350mm における立ち座り測定



L = 400㎜ における立ち座り測定



L = 500mm における立ち座り測定

以上の立ち座りにおける周波数変化の実測値から、立ち上がり時に一度ピークがあり、また座り時にも一度ピークがあるという特徴的な周波数変化が生じることが分かる。

図 12 に立ち上がり、座る際の代表的な周波数変化を示す。

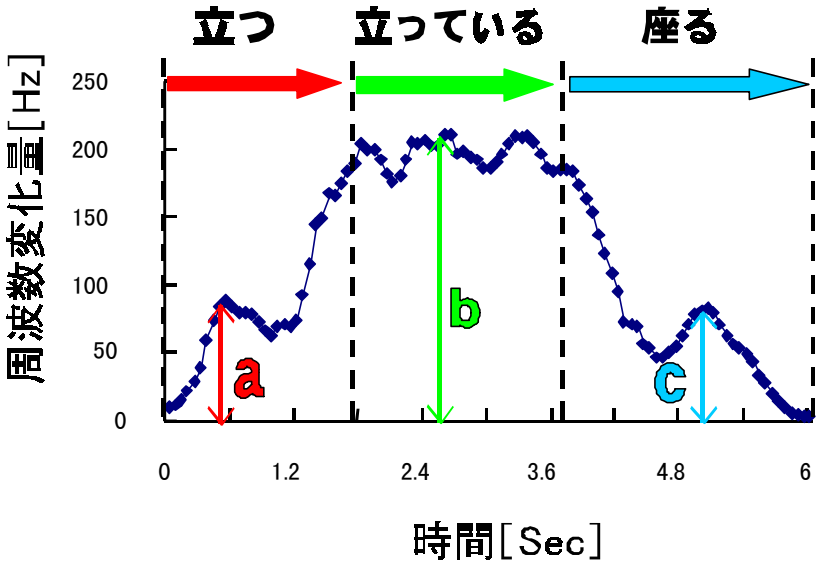


図 12.代表的な周波数変化と評価点

本実験で、立ちあがり、座る際の周波数変化において、特徴的なポイントをａ，ｂ，ｃの評価点と設定し、その点における周波数変化量を各椅子の高さごとに平均値を取り、比較したものを図 13 に示す。

通常立ちあがる際に椅子が低すぎると「辛い」と感じる。逆に椅子がある程度高くなると「楽だ」と感じる事から、椅子が低すぎる状態から、徐々に高くなるにしたがって足にかかる負荷が徐々に軽減されていくと考えられる。

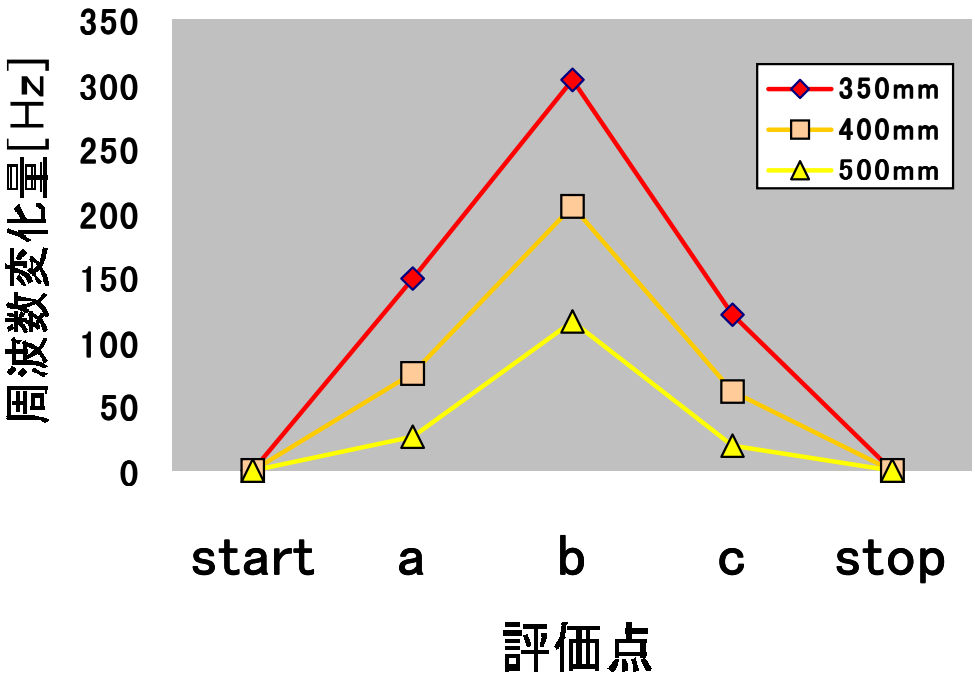


図 13.評価点ａ，ｂ，ｃにおける平均周波数変化量

また、同様の実験を Hardness Tester による実験も行った。Hardness Tester は立ち上がりや、座る際の微妙な変化であるａ，ｃといった値を捉えることは大変難しいことから、立ち上がったときの測定点における筋肉の硬さの値から、座った状態の筋肉の硬さを差分することで表した。測定結果を図 14 に示す。

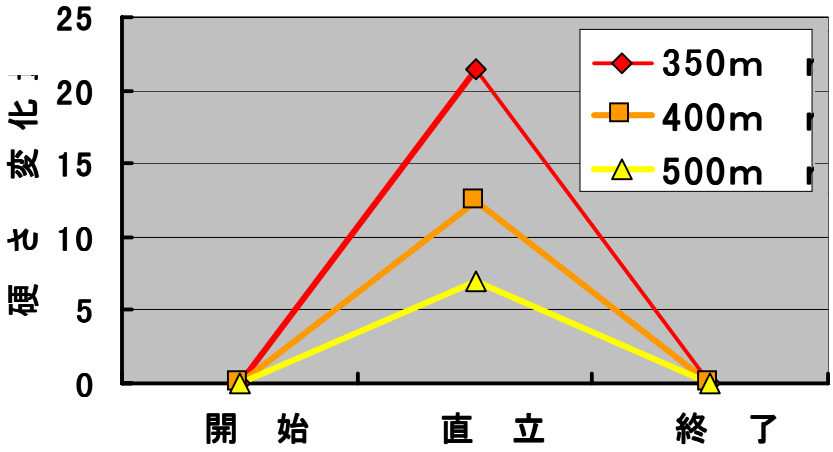


図 14. Hardness Tester による測定結果

図 13 と図 14 を比較すると、触覚センサの結果では椅子の高さＬが 350mmにおける周波数変化量が最も大きく、高さＬが 400mm、500mmと高くなるにしたがって周波数変化量が減少していることが分かる。また、Hardness Tester による測定結果においても、高さＬが 350mmにおける硬さ変化量が最も大きく、高さＬが 400mm、500mmと高くなるにしたがって硬さ変化量が減少していることが分かる。

これにより、椅子が徐々に高くなるにしたがって足への負担が徐々に軽減されていることが本センサによる周波数変化によって捉えられていることが分かる。

６．まとめ

本実験により、一般的な筋肉の硬さ変化と運動との関係が触覚センサにより測定できる可能性が明らかになった。

７．参考文献

・日本機械学会 編　『バイオメカニクス概説』

・ｲﾝﾀｰﾈｯﾄﾎｰﾑﾍﾟｰｼﾞ　『電脳体表解剖学図譜』
http://biking.taiiku.tsukuba.ac.jp/thesis/doguu/GR/GR.html

・丸山工作 著　　『筋肉のなぞを追って』

８．付録

使用機器

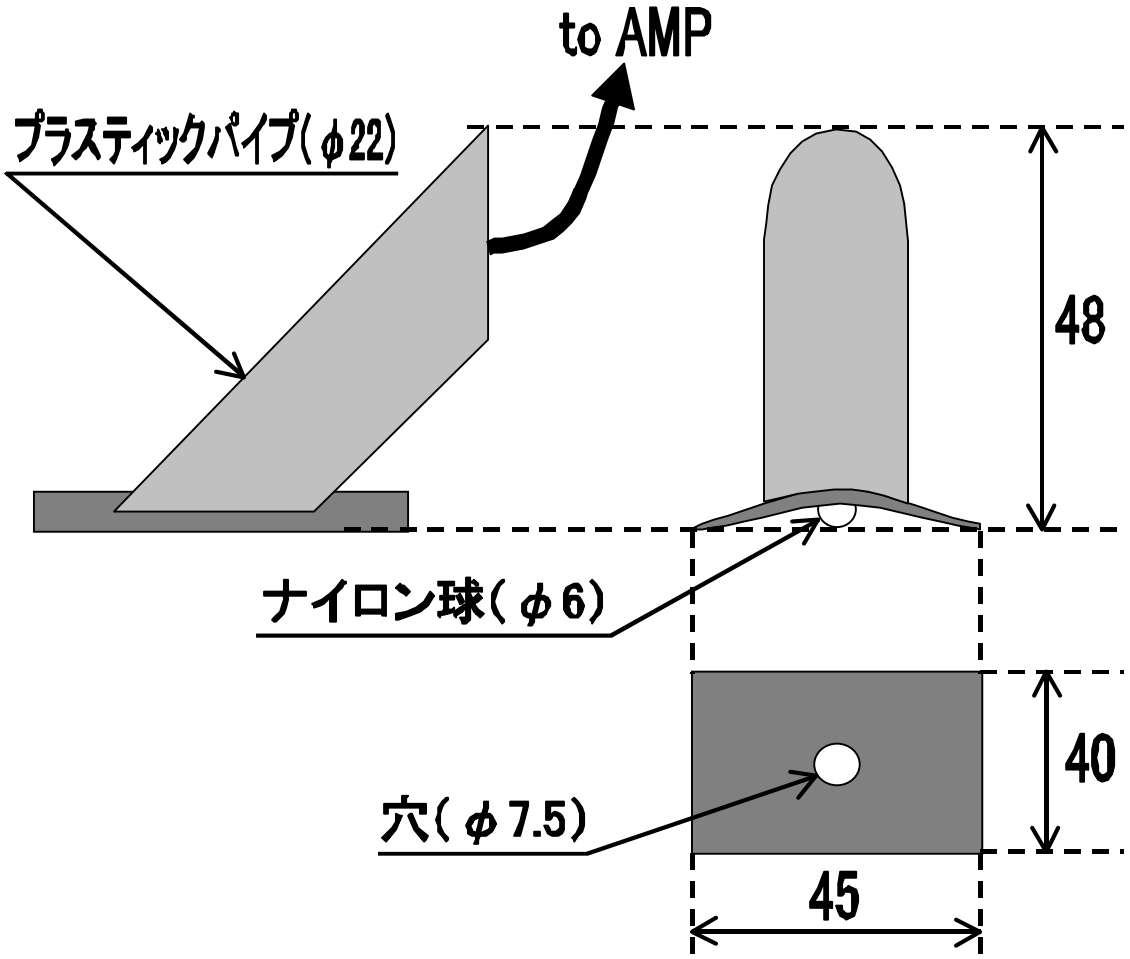
周波数カウンタ

メーカー：　ADVANTEST
機器名：　UNIVERSAL COUNTA
型番　：　TR5822

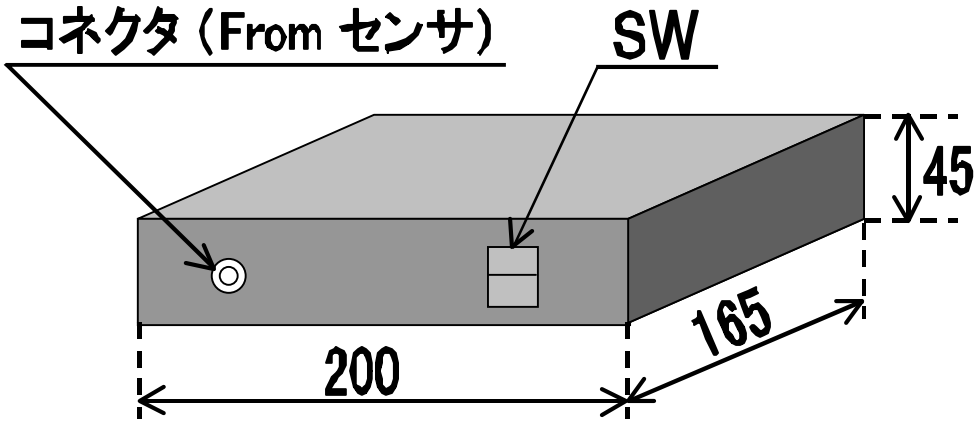
握力計

メーカー：　OSAKA HATA
機器名：　“ TARZAN ”
型番　：　DYNAMO METER 50kg

センサ外観及び寸法

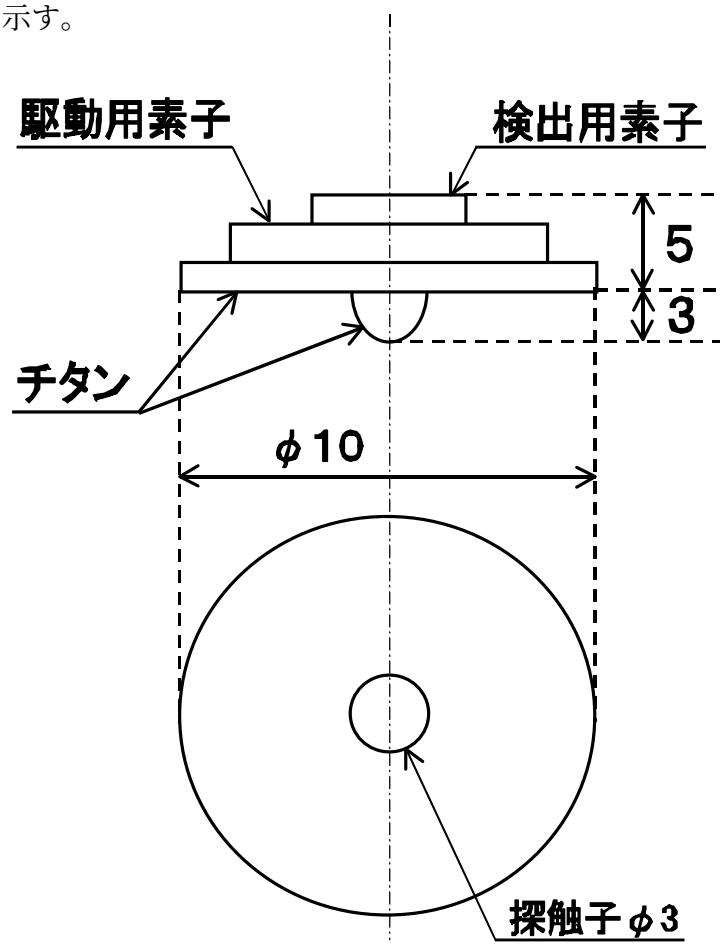


アンプ外観及び寸法

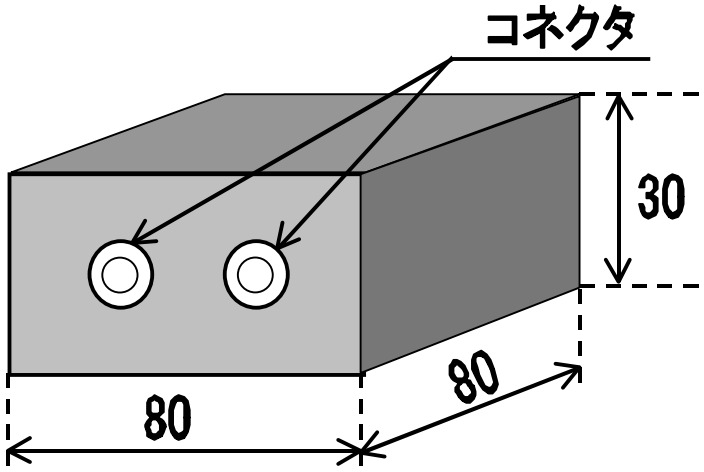


小型センサの開発について

今回実験に用いたセンサの外観を見ると分かるとうり、人体に取り付けるには多少大きすぎることから、小型センサ及びアンプを開発した。小型センサ及びアンプの概略を以下に示す。

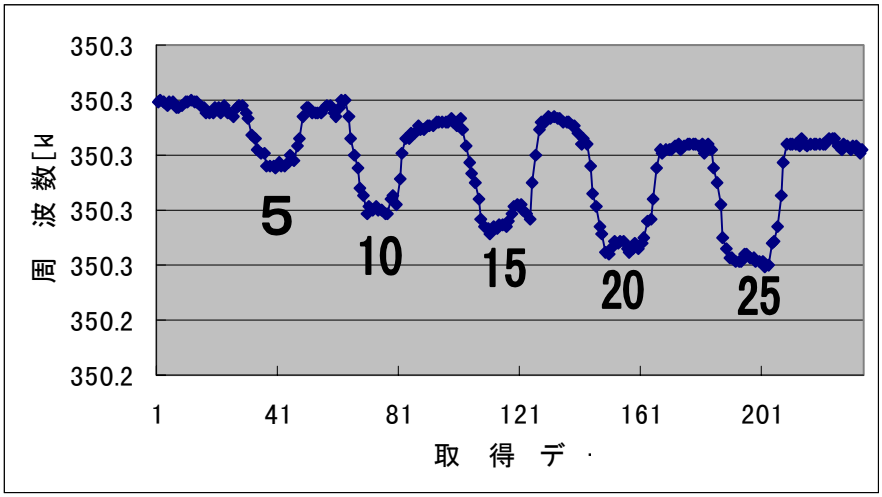


小型センサ概略図

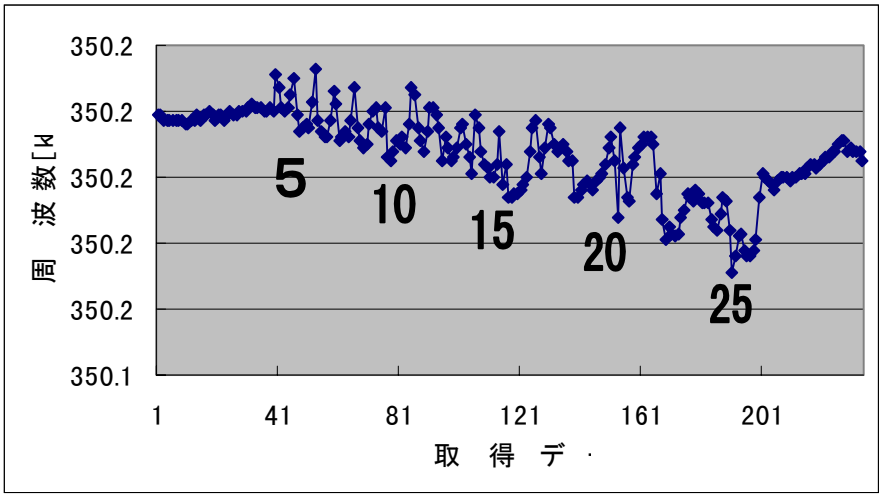


アンプ外観図

小型センサによるP1における測定結果　握力計表示：5ｋｇ～25ｋｇ



＜センサ装着直後から測定＞



＜センサ装着後約１分経過から測定＞

測定結果から、従来センサに比べると、握力が大きくなるにつれて周波数変化は減少する傾向が見られる。このセンサは対象物の音響インピーダンスを利用したセンサ(センサの基本原理参照)であるため、通常探触子に触れる物体の密度が高くなると周波数は上昇しなくてはならない。逆に対象物に探触子が埋没する量にしたがって周波数は減少する特性を持っている。そして、装着後約１分経過した時点で同様の実験を行なったが、装着直後のような周波数変化は見られなかった。

これらのことから、この小型センサは筋肉の硬さ変化を捉えているというよりもむしろ皮膚のごく表面における変位（埋没量）を検出していると考えられる。従来のセンサと小型センサがこのように異なる特性を示した原因として、探触子がナイロン球からチタンに変更されている点や、従来のセンサが57kHz程度で振動するのに対して、小型センサは350kHzとかなり高周波になっているのも一つの原因と考えられる。

小型センサのまとめ

　今回は筋肉の硬さ変化測定が目的であったため、今回の実験には適さなかったが、センサの探触子であるチタンの部分を多少大きくすることでセンサ自体の質量が増し、固有振動数を低周波領域にシフトすることが出来れば、従来センサと同様の結果が得られることも考えられる。また、皮膚の硬さ、質感などを測定対象とするならば、現在の状態で皮膚の潤い具合などを測定できるかもしれない。したがって、触覚センサを小型化できる可能性と、小型化する意義は十分にあると考えられる。

測定に使用したプログラム
GPIB 駆動用

```
Option Explicit

'Long 変数
Public Ret As Long                          'リターンコード
Public Srlen As Long                        '文字列の長さ
Public Cnt As Long                          'カウンタ
Public Cmd(32) As Long                      'コマンド配列
Public Pstb(32) As Long                      'シリアルポールした結果の配
列
'Boolean 変数
Public ExitFlag As Boolean
'String 変数
Public Srbuf As String                      '送信文字列
Public DevName As String                    '相手機器名
Public DevNum As String                     '相手機器のリスト位置
Public RetStr As String                      '戻り値(コード)からのエラー
を特定した文字列
Public TempStr As String                    '汎用文字列

'ｷｰﾜｰﾄﾞの定義
Public Const GP_GTL As Long = &H1           'MltLine サンプルで使用
Public Const GP_SDC As Long = &H4
Public Const GP_PPC As Long = &H5
Public Const GP_GET As Long = &H8
Public Const GP_TCT As Long = &H9
Public Const GP_LLO As Long = &H11
Public Const GP_DCL As Long = &H14
Public Const GP_PPU As Long = &H15
Public Const GP_SPE As Long = &H18
Public Const GP_SPD As Long = &H19
Public Const GP_MLA As Long = &H20
Public Const GP_UNL As Long = &H3F
Public Const GP_MTA As Long = &H40
Public Const GP_UNT As Long = &H5F
'Windows で規定されている値
Public Const HELP_CONTEXT = &H1             'ヘルプで使用
Public Const HELP_QUIT = &H2
Public Const HELP_CONTENTS = &H3

'ACX-GPIB(W32) Help コンテキスト
Public Const HLP_SAMPLES = 274
```

```
Public Const HLP_SAMPLES_BASIC = 275
Public Const HLP_SAMPLES_EVENT = 276
Public Const HLP_SAMPLES_MULTILINE = 277
Public Const HLP_SAMPLES_MULTIMETER = 278
Public Const HLP_SAMPLES_POLLING = 279
Public Const HLP_SAMPLES_PARARELL = 280
Public Const HLP_SAMPLES_VOLT = 281

'LoadProperty, SaveProperty で使用
Public Const ERR_FILE_NOT_FOUND = 190
Public Const ERR_FILE_COULD_NOT_OPEN = 191
Public Const ERR_FILE_WRITE = 192
Public Const ERR_FILE_READ = 193
Public Const ERR_FILE_UNKNOWN = 194
Public Const ERR_FILE_INVALID_FORMAT = 195

'Win32 API のコールのための宣言<ヘルプ表示で使用>
Declare Function WinHelp Lib "user32" Alias "WinHelpA" (ByVal hWnd As Long, ByVal
lpHelpFile As String, ByVal wCommand As Long, ByVal dwData As Long) As Long

'/// [エラーチェック(表示するメソッド名と戻り値を引数として受け取ります。)]
//////////////
Sub ErrCheck(Cmd As String, RetCode As Long, RetStr As String)
Dim ErrText As String                                   '返す文字列のバッファ
Dim UpperCode As Long               '戻り値を&HFF00 でマスクした数値

    ErrText = Cmd + " : 正常"
    If (RetCode = 0) Then RetStr = ErrText: Exit Sub    '正常終了

    Select Case (RetCode)
        Case 3:    ErrText = Cmd & " : FIFO 内にデータが残っています。"
        Case 80:   ErrText = Cmd & " : I/O アドレスエラーです。"
        Case 82:   ErrText = Cmd & " : レジストリ設定エラーです。"
        Case 128:  ErrText = Cmd & " : 受信予定数オーバー<受信>｜SRQ
                                  未発見<POLLING>"
        Case 200:  ErrText = Cmd & " : スレッドの作成に失敗しました。"
        Case 240:  ErrText = Cmd & " : 強制終了([ESC] 確認)しました。"
        Case 241:  ErrText = Cmd & " : ファイル入出力エラーです。"
        Case 242:  ErrText = Cmd & " : アドレス指定エラーです。"
        Case 243:  ErrText = Cmd & " : バッファ指定エラーです。"
        Case 244:  ErrText = Cmd & " : 配列サイズエラーです。"
        Case 245:  ErrText = Cmd & " : バッファが足りません。"
        Case 246:  ErrText = Cmd & " : 不正なオブジェクト名です。"
        Case 247:  ErrText = Cmd & " : デバイス名横のチェックボックスが
```

```
                                                              無効です。"
        Case 248:  ErrText = Cmd & " : データ型が不正です。"
        Case 249:  ErrText = Cmd & " : これ以上デバイスを追加不可能です。"
        Case 250:  ErrText = Cmd & " : デバイス名が見つかりません。"
        Case 251:  ErrText = Cmd & " : デリミタがデバイス間で違います。"
        Case 252:  ErrText = Cmd & " : GPIB エラーです。"
        Case 253:  ErrText = Cmd & " : デリミタのみ受信<受信文字数=0>"
        Case 254:  ErrText = Cmd & " : タイムアウトエラーです。"
        Case 255:  ErrText = Cmd & " : パラメータエラーです。"
    End Select

    '////////// IFC & SRQ 受信ステータスメッセージ //////////    :CheckStatus:
    'UpperReturnCodeEnable で上位バイトを有効にした時に関係します。
    UpperCode = RetCode And &HFF00          '上位ビットをマスクします。
    Select Case (UpperCode)
        Case &H100: ErrText = ErrText & " + [SRQ]受信<STS>"
        Case &H200: ErrText = ErrText & " + [IFC]受信<STS>"
        Case &H300: ErrText = ErrText & " + [SRQ][IFC]受信<STS>"
    End Select

    RetStr = ErrText

End Sub

'///     [    初    期    化    サ    ブ    ル    ー    チ    ン    ]
////////////////////////////////////////////////////////////
Function GpibInit(Acx As Object, RetSts As String) As Long

    Acx.exit                                            '再初期化防止のため
    Ret = Acx.Ini                                       'ボードの初期化
    If (Ret <> 0) Then
        Call ErrCheck("Ini", Ret, RetSts)
        GpibInit = Ret
        Exit Function
    End If
    'マスタの時のみ以下の IFC、REN を実行します。
    If (Acx.MasterSlave = 0) Then
        Ret = Acx.Ifc                           'IFC(Interface Clear)の送出
        If (Ret <> 0) Then
            Call ErrCheck("Ifc", Ret, RetSts)
            GpibInit = Ret
            Exit Function
        End If
        Ret = Acx.Ren                           'リモートラインを有効にする
```

```
        If (Ret <> 0) Then
            Call ErrCheck("Ren", Ret, RetSts)
            GpibInit = Ret
            Exit Function
        End If
    End If

    RetSts = "初期化を完了しました。"                    '正常終了
    GpibInit = Ret

End Function

'///      [GPIB      終      了      サ      ブ      リ      ー      チ      ン      ]
////////////////////////////////////////////////////////
Function GpibEnd(Acx As Object)

    If (Acx.MasterSlave = 0) Then Ret = Acx.Resetren 'リモートラインのリセッ
ト(MasterOnly)
    Ret = Acx.exit                                    '終了処理の実行

End Function

Function DevidedString(Base_Str As String, Str_Cnt As Long) As String

'Base_Str の中から","で区切られた Str_Cnt 番目の文字列を返します。
'Str_Cnt=1 の時、先頭の文字列を返します。また、Str_Cnt 番目の文字列が
'なかった場合,また Str_Cnt=0,Str_Cnt>100 だった場合には ""を返します。
Dim StrLenPre(100) As Long
Dim StrLenAft As Long
Dim BaseLen As Long
Dim Count As Long

    If (Str_Cnt = 0) Or (Str_Cnt > 100) Or (Base_Str = "") Then
        DevidedString = ""
        Exit Function
    End If

    '渡された文字列の長さを取得します
    BaseLen = Len(Trim(Base_Str))
    StrLenPre(1) = 0

    For Count = 1 To Str_Cnt
        '","のある位置を取得します
        StrLenAft = InStr(StrLenPre(Count) + 1, Base_Str, ",")
```

```
        If StrLenAft = 0 Then
            '指定された位置より後ろに","が見つからなかった場合
            If Count = Str_Cnt Then
                '区切られた最後の位置の場合
                StrLenAft = BaseLen + 1
            Else
                '区切られた数より、指定された位置(Str_Cnt)が
                '大きかった場合(結果として""を返します)
                StrLenAft = 1
            End If
            Exit For
        End If
        '次の検索開始位置の指定
        StrLenPre(Count + 1) = StrLenAft
    Next
    DevidedString = Trim(Mid(Base_Str, StrLenPre(Str_Cnt) + 1, (StrLenAft) -
(StrLenPre(Str_Cnt) + 1)))

End Function

'///      [UNT,UNL      コ      マ      ン      ド      を      送      出      ]
////////////////////////////////////////////////////////
Public Sub SendComand(Acx As Object)

    Cmd(0) = 2                          'Command Count
    Cmd(1) = GP_UNL                     'UNL = &H3F
    Cmd(2) = GP_UNT                     'UNT = &H5F
    Ret = Acx.Comand(Cmd(0))
End Sub
```

ディスプレイ表示及び、データ保存用

```
Option Explicit


Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

Private Sub form_load()
TextRet.Text = ""
DevName = AcxGpib1.DeviceName1      '１番目のデバイス名を指定
'GpibTest.Width = 6480
```

```
End Sub

Private Sub initialize_click()
AcxGpib1.TimeOut = 5000
Ret = GpibInit(AcxGpib1, RetStr)                'GP-IB の初期化
Call DetectErr("Initialize", Ret, RetStr)          'エラーチェック
If (Ret <> 0) Then Exit Sub                      'エラーの場合
'    Extend.Enabled = True                      '履歴を Enabled にします
End Sub
Private Sub sokutei_click()
Dim Date3 As String
Dim Data1, Data2 As Long
Dim DPdata As Long
Dim f2g0 As String
Dim i As Single
Srlen = Len(f2g0)
If Right(Dir1.Path, 1) = "\" Then
    Date3 = Dir1.Path & Text1.Text & ".csv"
Else
    Date3 = Dir1.Path & "\" & Text1.Text & ".csv"
End If
Open Date3 For Output As #1
'ユニバーサルカウンタにコマンド（f2g0）を送信
'Ret = AcxGpib1.Talk(DevName, 4, f2g0) '送信文字列：Srlen=4
For i = 0 To 50
'データの受信
Srlen = 4096
Srbuf = String(Srlen, " ")
Ret = AcxGpib1.Listen(DevName, Srlen, Srbuf)
Data1 = Mid(Srbuf, 1, Srlen)
If i = 0 Then Data2 = Data1

DPdata = (Data1 - Data2) / 10
Write #1, Val(Data1), Val(DPdata)
AcxTrend1.DisplayData (DPdata)
DoEvents
Next i
Close #1
End Sub

Private Sub exit_click()

Ret = GpibEnd(AcxGpib1)
```

```
Unload Me

End Sub
' [画面更新]
Private Sub DetectErr(KeyWord As String, Ret As Long, RetStr As String)
    Call ErrCheck(KeyWord, Ret, RetStr)   'エラーコードからのエラー特定
    TextRet.Text = RetStr
'    RetCode.Text = Ret
    Refresh                                          '画面の更新
End Sub
```