

肩義手のための RC サーボコントローラー AGB65-RSC を 用いた感圧センサーの開発

木下 敏治* 三好 敦士** 西原 大樹***

Development of the Pressure-Sensitive Sensor using RC Servo Controller AGB65-RSC for SDP (shoulder disarticulation prosthesis)

Toshiharu KINOSHITA, Atsushi MIYOSHI and Daiki NISHIHARA

Synopsis

The purpose of this research is improvement of the hand for Shoulder Disarticulation Prosthesis (SDP) with 6 degrees of freedom developed 2007 year at our college. Concretely, it is making the movement of human hand and measuring pressure of the holding objects by the automatic operation. The control board is RC servo controller AGB65-RSC and sensor unit is AGB65-4FS. For making the control program used “Visual C #2005”. Operation of the hand by one RC servo motor and the measurement value of the pressure-sensitive sensor were able to be displayed. The artificial hand with one RC servo motor can move like a human hand.

1. まえがき

厚生労働省の「身体障害児・者等実態調査」によると、平成 13 年度の 18 歳以上の身体障害者数は 324 万 5 千人で、そのうち肢体不自由者数は 174 万 9 千人と全体の 53.9[%]となっている。

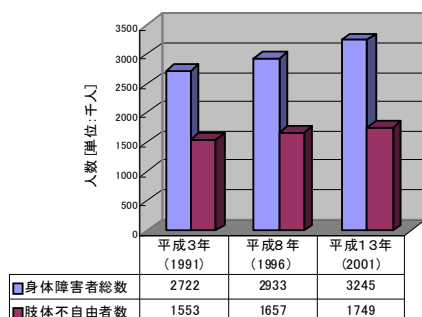


図 1. 身体障害者(18 歳以上)の全国推計数

* 電子工学科

** 和歌山大学工学部光メカトロニクス学科

*** フジケンエンジニアリング

図 1 がそのグラフである。この肢体不自由者数は平成 3 年度の調査では 155 万 3 千人、平成 8 年度の調査では 165 万 7 千人、平成 13 年度の調査では 174 万 9 千人と、毎年増加の傾向にある。それに対して義手や義足の開発は遅れている。特に義手は上腕¹⁾・前腕・手関節・手指など切断部位によって種類も変わり、義足に比べて使用目的が多く、高度な制御を必要とし研究開発が難しいとされている。このため日本では、普及している義手の 9 割が装飾義手と呼ばれる外観の復元を重視した、動かない義手であり、動力義手の普及率は低い。そして動力義手の中でも特に、肩から指までの駆動を行う肩義手については実用化の例は TDU 全腕義手(東京電機大学 斎藤教授²⁾)や、ユタアーム(米 ユタ大学)などしかなく、ほとんど実用化されていない³⁾。その理由として上腕から手先まで制御するのは自由度がとて多く難しい、実際の腕の重さに対して義手の重量が重すぎる、価格が高い、操作性が悪く装着してからの使用訓練が非常に困難である、機能が少なく不便である、などの問

題が考えられる。これらの点を解決することを最大の目標として義手の手先具の研究・開発を行った。前腕義手の研究の進捗現状は参考文献 4～20 に示す。

2. 研究の概要

軽量で安価な義手の作成を目標に、2006 年度の卒業研究生が製作した 6 自由度肩義手をもとにした顔面方位制御システム^{21) 22)}の開発のための高性能な手先具の開発を目標とした。2006 年度の手先具はサーボモーターによる簡単な開閉動作によって物体保持をさせるものであった。また、その形状も人間の手先とはかけ離れた外観であった。2007 年度の研究目標としては、

- ・ 人の手先に似た構造にする
- ・ 軽量化を試みる
- ・ センサーでモーターの動きを制御する

の 3 点である。

バイオメタルファイバーを用いた 5 自由度の義手の手先具の研究については参考文献 23 に詳しく説明してある。

アクチュエータには軽量で高出力な RC サーボモーター KRS-4014HV Red Version (近藤科学製) を使用し、新たに手先具を製作した。制御基板には RC サーボコントローラー AGB65-RSC²⁴⁾、センサーユニットには AGB65-4FS (浅草ギ研製) を用いた。通信速度は両方ともに、115kbps にも対応しているが、今回は通常設定値の 9600bps で使用した。制御ソフトについては Visual C# 2005²⁵⁾ を用いて作成した。これは、自由に機能や操作画面をつくることができ、シリアル通信の記述が容易にできるためである。構成部品には主に、カーボンファイバーという軽量で丈夫な素材を使用した。構造上、強度が求められる部分にはアルミプレートを用いた。製作した手先具に感圧センサーを取り付け、物体を持たせたときの圧力値を表示させ、物体個々のデータをとる。また、適正保持ができるようにプログラミングを行うことである。

3. センサー系の構成

2007 年度の研究では感圧センサー AGB65-4FS の他にサーボモーターを使用するので、AGB65-RSC を接続した。2007 年度新たに使用

した感圧センサー AGB65-4FS を図 2 に示す。これは、1 つのセンサーユニットにつき、4 個までの圧力測定部を接続することができる。AGB65-RSC については、参考文献 24 に詳しく記述してある。また、実際の構成は図 3 のような配線となっている。感圧センサー用電源、モーター制御基盤用電源、モーター動作電源、の計 3 つの電源を必要としている。



図 2. 感圧センサー AGB65-4FS

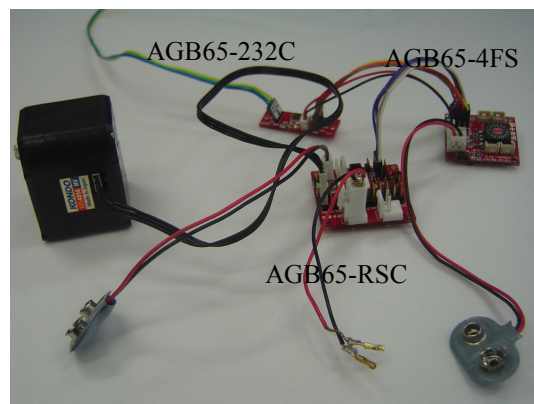


図 3. センサー系の構成図

4. プログラム

この手先具系の動作に当たってのプログラミングの方法を具体的に示す。

- ・ 感圧センサーの表示プログラムの作成
- ・ リアルタイムでセンサーを表示させるプログラムに書き換える
- ・ モーター動作のプログラムを追加

4.1 感圧センサーの表示

感圧センサーの値を表示するプログラムによって作成したフォームを図 4 に示す。

プログラムの動作方法 (Visual C# 2005 を用いた動作 プログラム名: 自動機構)

1. Visual C #2005 を起動し, 任意のプログラムを選択する。
2. [F6]でエラーチェックして[ctrl + F5]でビルドする。
3. 電源を AGB65-RSC×2 , AGB65-4FS に接続する。線が別々に別れているので A LINCO (電源) の装置を使う場合は後ろ側の端子にそれぞれ接続する。
4. モーター1 個の駆動しか記述していないので, AGB65-RSC の[p0]にモーターからの配線を接続する。
5. [モーターの位置], [モーターの速度]などの項目を任意の値に変更していく。
6. [駆動], [初期位置], などのボタンを押して動かせる。



図 4. センサー値表示フォーム

この作成したプログラムは, 図 4 の下に位置している“4 点読み取り” ボタンを押すことによって 4 つのセンサーの値を表示させることができる。

次に, ボタンを押すことなしにセンサーの値を表示させるプログラムの作成について記載する。

<センサー値表示プログラム>

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        private byte[] sendData;

        public Form1()
        {
            InitializeComponent();
            serialPort1.PortName = "COM1";
            serialPort1.BaudRate = 9600;
            serialPort1.Open();
        }

        private void readButton_Click(object sender, EventArgs e)
        {
            sendData = new byte[4];
            sendData[0] = 255;
            sendData[1] = Convert.ToByte(id
2.Text);

            sendData[2] = 1;
            sendData[3] = 30;
            serialPort1.Write(sendData, 0, 4);
        }
    }
}
```

4.2 感圧センサーのリアルタイム表示

先に記載した感圧センサーの表示プログラムに Visual C# の機能の 1 つである, “Timer” という機能を使うことによってボタンを押すことなしにセンサー値を表示させるプログラムを作成した。そのプログラムによって作成したフォームを図 5 に示す。

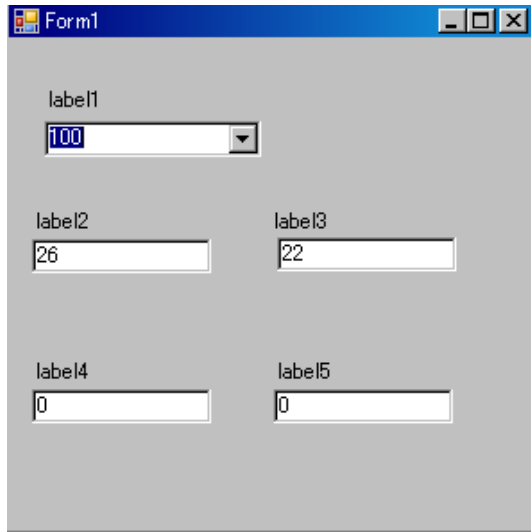


図 5. リアルタイム表示フォーム

<リアルタイム表示プログラム>

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        private byte[] sendData;
        private byte[] rcvData;
        private byte[] result;
        static private byte gx = 0;
        static private byte gy = 0;
        static private byte gz = 0;
        static private byte gp = 0;

        public Form1()
        {
            InitializeComponent();
            serialPort1.PortName = "com1";
            serialPort1.BaudRate = 9600;
            serialPort1.Open();
            timer1.Interval = 100;
            timer1.Enabled = true;
```

```
timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            sendData = new byte[4];
            sendData[0] = 255;
            sendData[1] = Convert.ToByte(I
d.Text);

            sendData[2] = 1;
            sendData[3] = 30;
            serialPort1.Write(sendData, 0, 4);
        }
```

```
private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
```

```
    rcvData = new byte[1];
    int l = 0;
    serialPort1.Read(rcvData, 0, 1);
    if (rcvData[0] == 255)
    {
        serialPort1.Read(rcvData, 0, 1);
        if (rcvData[0] == (Convert.ToByte(I
d.Text)))
        {
```

```
            serialPort1.Read(rcvData, 0, 1);
            l = rcvData[0];
            result = new byte[l];
            for (int i = 0; i < l; i++)
            {
                while (serialPort1.ReadBufferSize ==
0);

                serialPort1.Read(result, i, 1);
            }

            gx = result[1];
            gy = result[2];
            gz = result[3];
            gp = result[4];
            p1.Text = gx.ToString();
            p2.Text = gy.ToString();
        }
    }
}
```

```

    {
        {
            {
                {
                    p3.Text = gz.ToStr
                }
            }
        }
    }
    ing();
    p4.Text = gp.ToStr
    ing();
}

```

4.3 モーターの動作を追加

RC サーボモーターを駆動させるプログラム
を今まで作成したプログラムに加えて記述する。

＜サーボモーター駆動プログラムの追加＞

```
private void runButton_Click(object sender, EventArgs e)
{
    sendData = new byte[7];
    sendData[0] = 255;
    sendData[1] = Convert.ToByte(I
d.Text);

    sendData[2] = 4;
    sendData[3] = 2;
    sendData[4] = Convert.ToByte(s
N.Text);

    sendData[5] = Convert.ToByte(s
P.Text);

    sendData[6] = Convert.ToByte(s
S.Text);

    serialPort1.Write(sendData, 0, 7);
}
```

上記のプログラムを追加することによってモータの駆動を制御できるようになった。図6に現段階でのフォームを示す。

動作

ID

サーボ番号 P1 p2

サーボ位置 p3 p4

サーボ速度

図 6. モーター駆動を追加したフォーム

※モーターを初期位置(127)に戻すボタンを追加している。

5. 動作実験

本格的に手先具に取り付ける前に感圧センサーAGB65-4FSの印加荷重による表示値を測定した。これは、浅草ギ研から提供されている記述には、測定10N(約1kgf)、破壊荷重45N(約4.6kgf)、最低測定荷重が約200gと記載されているが、それ以上の荷重がかかったとき、どのような値を示すのかを知ることが必要であったので、図7のように重量によるセンサー値を測定した。実際の人間の手による保持時の荷重は、2Kg付近と仮定した。上記のように、測定開始荷重は約200gとされているが、今回の測定では約300g～380gの間に0→1へとセンサーが反応しだした。これは、圧力測定部分が小さい球形をしているため、力のかかり方に偏りが出てしまったためだと思われる。

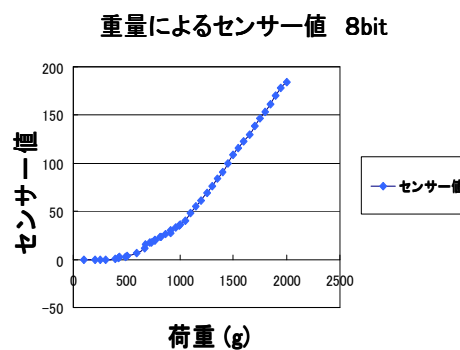


図 7. 重量によるセンサー値
(※1000g 以降は 50g ごと)

6. 手先具の作成

6.1 手先具 1 号機

上記のように感圧センサーを動作させることに成功したので、次に簡単な構造の手先具に取り付けて動作させた。作成した手先具を図 8 に示す。

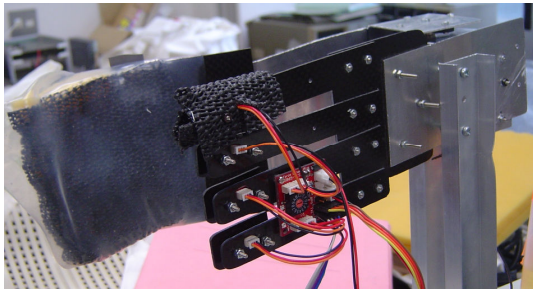


図 8. 手先具 1 号機

図 8 の手先具を使って、重さがそれぞれ 100g, 150g, 200g のペットボトルを保持したときのモーターの位置によるセンサー値の特性のグラフである。モーターの位置とは、初期位置を 127 として、1 度ずつ角度を減らせていったときの位置のことである。それぞれのセンサーを

P1 → 人差し指

P2 → 中指

P3 → 薬指

P4 → 小指

(それぞれ第 1 間接部)に取り付けた。

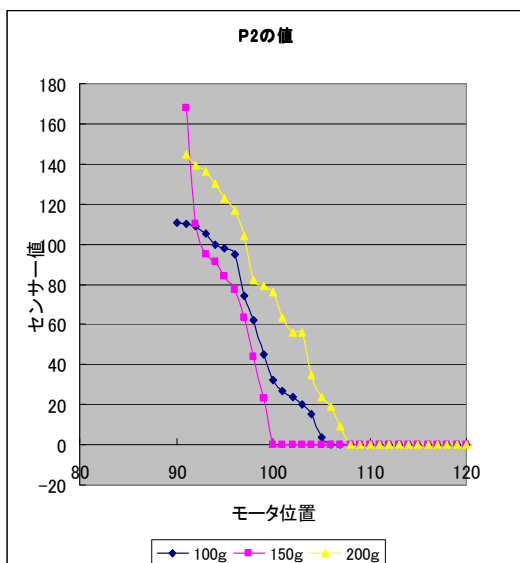


図 9. P2 (中指) の変化

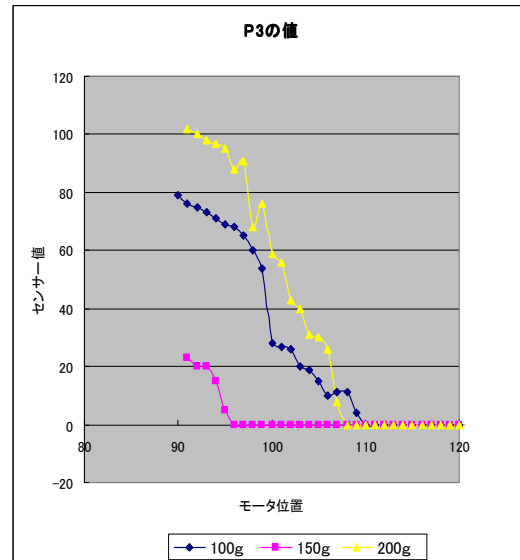


図 10. P3 (薬指) の変化

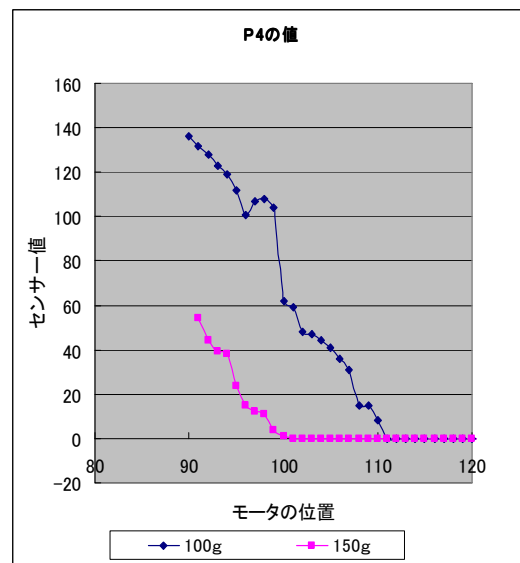


図 11. P4 (小指) の変化

測定終了のモーター位置は、センサー部やモーター部への負荷を考え、90とした。図 9 は人差し指に装着した P1 のみ反応しなかった。重量が重くなるほどセンサー測定開始位置が早くなる、または遅くなる、のどちらかを想定していた。しかし、センサーの感度の問題か、物体保持の方法が悪かったのかは不明だが、物体の重量とセンサー測定開始位置との関係性は見出せなかった。図 10 も同様に関係性は見出せなかった。図 11 については 200g のペットボトルを保持した際には P4 が反応しなかったので 100g, 150g のみのグラフとなっている。この測定

では、瞬間で表示値が変化するため、 ± 2 程度の誤差を含んでいる、また、12bit での表示モードでは、 $\pm 5 \sim 10$ 程度の瞬間的な表示値変化が起こることがわかった。この測定では、センサー値からかなり大きな圧力がセンサーにかかっていることがわかる。よって、その圧力を小さくするため、手先具の構造を考え直す必要がある。

6.2 手先具 2 号機

上記のような問題点が明らかになったが、解決法はセンサー部分に圧力を加える部分にあることがわかったので、つぎの手先具の作成に入る。手先具 2 号機は、3 本の指をもつ手先具とした。手先は、単純な開閉動作のみではなく、指の曲げ、そして伸ばしができるような構造にした。その実現のために、2005 年度卒業研究生である菅 直人君の研究で使用された指の機構²³⁾を参考にした。その構造については図 12 に示すとおりである。

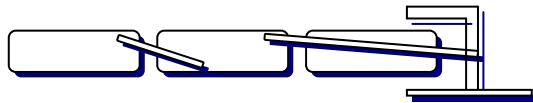


図 12. 構造（平常時）

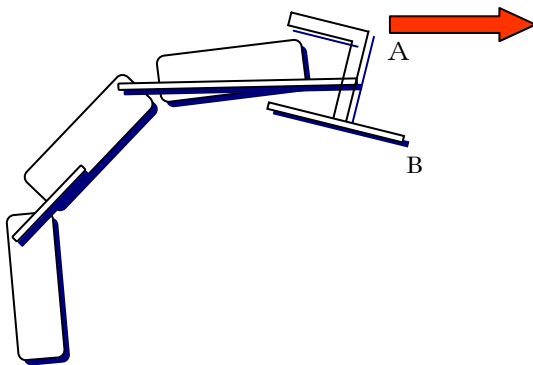


図 13. 構造（力を加え曲げるとき）

上記の構造より図 13 のように A 点に力を加えることによって指先を曲げることができる。また、指先を伸ばそうとするときは、B 点に矢印と同じような力を加えることによって伸ばすことができる。この構造は、単純、かつ軽量な指先を実現できると考えられる。図 14 に作成した指先を示す。



図 14. 作成した指先

また、今までに木下研究室で作成された手先具は 6 自由度、つまり手首の曲げの動作が実現されていなかったため、手首の曲げ動作ができる手先具を作成した。しかし、それに伴い前腕の回旋(05)が制御できない構造となってしまった。

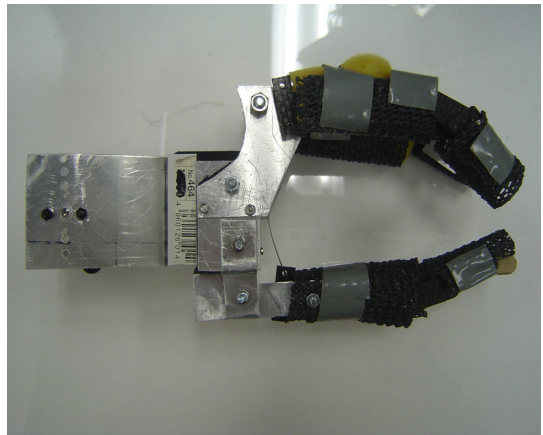


図 15. 手先部分（側面）



図 16. 手先部分（前面）

モーターを固定するために高い強度が求められる部分にはアルミプレートを使用した。図 13 のような力を加えるのに、サーボモーターを使用し、ワイヤーで巻き取ることによって指先を曲げる動きをさせた。また、指先を伸ばすときに

は、モーターを逆回転させもう一方のワイヤーを巻き取るようにしている。表面には滑り止めをつけている。次に、手首部分を加えた手先具 2 号機の構成図を図 17 に示す。

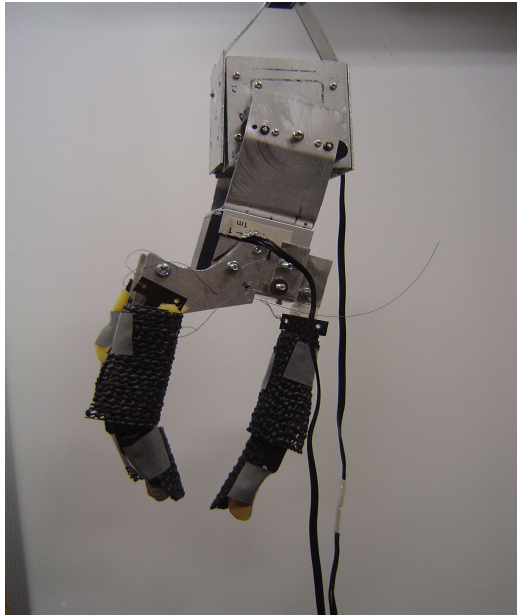


図 17. 手先具 2 号機

手先具 2 号機は、人間の指先の動きに似せようとしたが、指の曲げ、伸ばしの両方にワイヤーを用いているため、工作精度のばらつきもあって動きがスムーズでなく、満足のいく動きをしなかった。このため、感圧センサーと高性能な手先具をあわせる手先具 3 号機には、菅君の指先の構造を使用することを断念し、ワイヤーによる単純な開閉動作のみの手先具を作成することとした。

6.3 手先具 3 号機

手先具 3 号機では作成したプログラムと新たな手先具をあわせて、2006 年度の卒業研究生の作成した 6 自由度肩義手の手先に取り付けることを目標として作成した。そのため、構造の改良だけでなく、軽量化も重要となった。人の手先に似た構造にするため、指は 5 本あることが望ましいと考えた。しかし、パーツの増加に伴って手先具自体の重量も増加してしまうため、モーターを固定する部分以外は、極力重量の軽いカーボンファイバーを使用することを念頭において設計を始めた。また、手先具 2 号機では指先の曲げ、伸ばしともにワイヤーを使用していたが、今回は指先の曲げにはワイヤーを使い、

伸ばしにはバネを使用した。バネを使用することによって構造の簡単化と軽量化を実現できるはずである。図 18 に手先具 3 号機の寸法を示す。

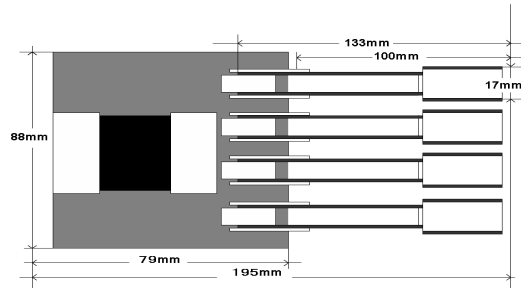


図 18. 手先具 3 号機の寸法

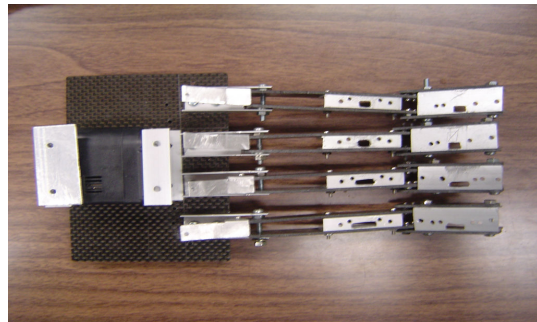


図 19. 手先具 3 号機 (手のひら部分)

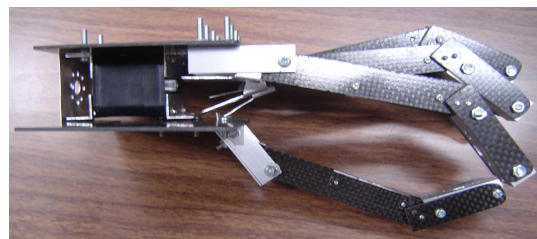
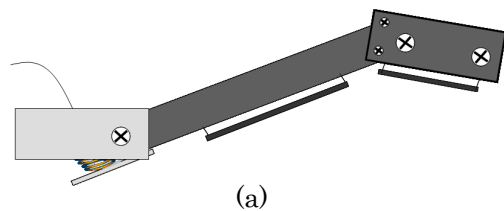


図 20. 手先具 3 号機 (側面)

図 19 のそれぞれの指に計 2 つの感圧センサー装着のためのアルミパーツをカーボンファイバーの間に固定している。次に、指の開閉の仕組みについて図 21 に示す。



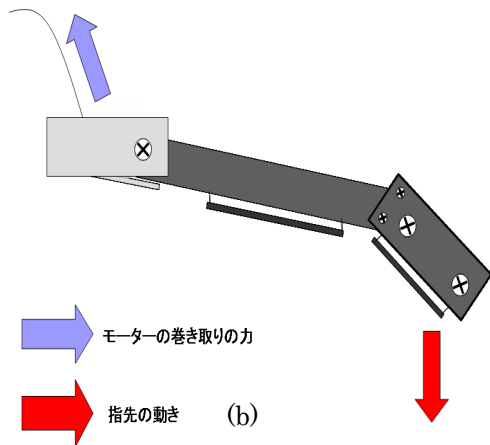


図 21. 指の開閉の仕組み 1

図 21(a)のときには、ワイヤーはゆるい状態になっているので、バネによって指先が開く仕組みになっている。また反対に、(b)のときには、モーターによってワイヤーを巻き取ることによって、バネを縮めさせ、指先を閉じる仕組みになっている。

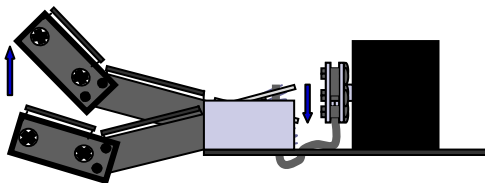


図 22. 指の開閉の仕組み 2

5本の指を同時に開閉するため、モーター自体に大きな負荷がかかってしまう可能性があるが、人間の構造に似たものにするため、あらかじめ考慮に入れている。

図 23～27 に感圧センサーを装着した手先具 3 号機を示す。

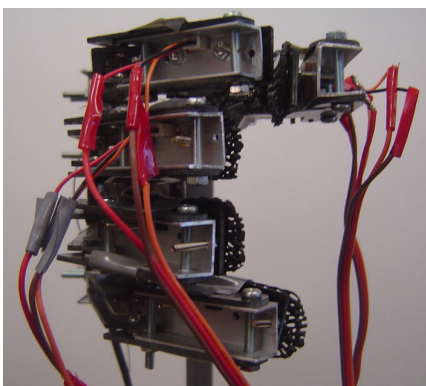


図 23. 感圧センサーを接続した手先具 3 号機

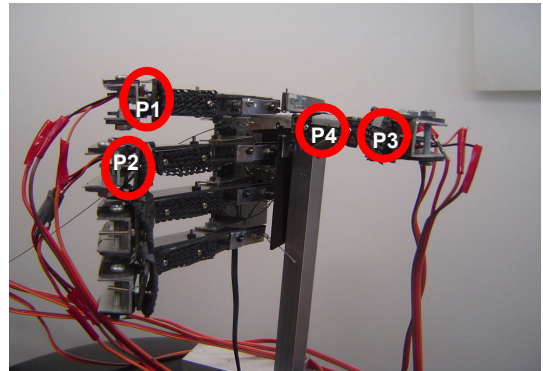


図 24. 指先を開いた様子



図 25. 指先の様子

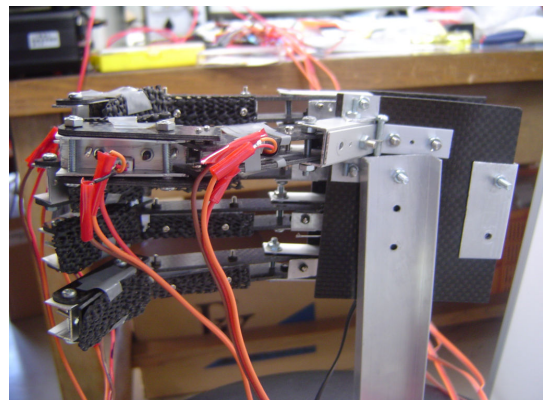


図 26. 親指側からの様子



図 27. 指の根元とモーター駆動部

センサーの装着位置

P1 → 人差し指

P2 → 中指

P3 → 親指 第1間接

P4 → 親指 第2間接

センサーの取り付け位置については、物体を保持するときに親指は重要な役割を果たしているため、親指に2つのセンサーを装着した。

7. 手先具3号機での動作

次に、この手先具3号機を用いた動作実験を行った。その方法は今までと同様である。保持させた物体は、ペットボトル（空容器,100g,200g,300g）野球ボール、ペンの3種類である。それぞれについてのデータをまとめたグラフを図28に記載する。なお、このプログラムは適正保持時にモーターの動きを停止させるプログラムと同じであるが、この実験では作動させないようにしている。

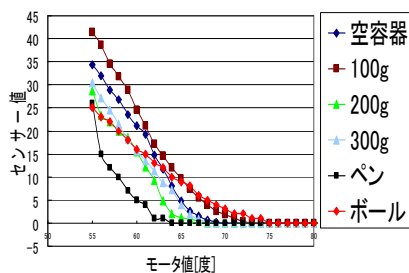


図 28. モーター角度によるセンサー値

図28より、ペットボトルはどのような重量であっても測定開始位置に違いはあっても似たような傾きとなっていることがわかる。また、野球のボール、ペンもそれぞれ特異な傾きとなっていることが容易に理解できる。本来なら感圧センサーを動作させ、新たな手先具を製作することが目的の研究であったが、このグラフの測定開始位置から保持した物体を表示することを新たな目標としてプログラムを作成した。まずは、当初から予定されていた追加プログラムについて記載する。

付け加えたプログラム

①モーター駆動のプログラミングに条件を加えて、動作を制御。

②条件を満たさなくなったとき（圧力がかかっ

た時）、その時のモーター値を表示。

①について述べると、ある一定以上の圧力がセンサーに感知されたとき、モーターに手先を閉じる命令がされても、プログラムで条件判定をし、それ以上モーターを駆動させないようにするものである。ある一定以上の圧力をセンサー値“10”と設定した。②については、①の動作制御が作動したときのモーター位置を表示するというプログラムである。

<追加したプログラム①>

```
private void sp_ValueChanged(object sender,
                                EventArgs e)
{
    if (result[1] < 10)
    {
        if (result[2] < 10)
        {
            if (result[3] < 10)
            {
                if (result[4] < 10)
                {
                    sendData = new byte[7];
                    sendData[0] = 255;
                    sendData[1] =
                        Convert.ToByte(id.Text);
                    sendData[2] = 4;
                    sendData[3] = 2;
                    sendData[4] =
                        Convert.ToByte(sn.Text);
                    sendData[5] =
                        Convert.ToByte(sp.Text);
                    sendData[6] =
                        Convert.ToByte(ss.Text);
                    serialPort1.
                        Write(sendData, 0, 7);
                }
            }
        }
    }
}
```

※ “sp_ValueChanged”はクリックするごとにモーターを駆動させる命令

<追加したプログラム②>

```

if(result[1] > 10)
{
    ts.Text = Convert.ToString(sp.Text);
}
if(result[2] > 10)
{
    ts.Text = Convert.ToString(sp.Text);
}
if(result[3] > 10)
{
    ts.Text = Convert.ToString(sp.Text);
}
if(result[4] > 10)
{
    ts.Text = Convert.ToString(sp.Text);
}
    
```

上記 2 つのプログラムを追加したことによって適正な物体保持ができるようになった。まだ弱くもろいものはセンサーが圧力を感知する前に破損してしまう恐れがあるが、ある程度の成功を得ることができた。図 29・30 にペンを保持したときの様子を示す。

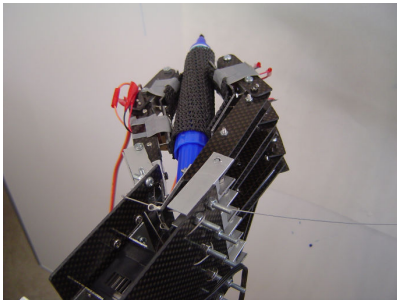


図 29. ペン保持の様子 1

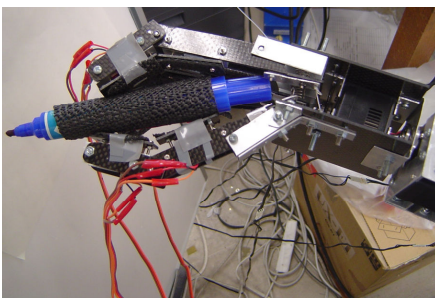


図 30. ペン保持の様子 2

では次に、先ほど述べた、保持した物体を表示

することについて記載する。これは図 28 から、ペットボトル（空容器）、野球ボール、ペンの 3 つを抜き出して考察していく。

また、手先具 3 号機を 2006 年度の 6 自由度肩義手にとりつけボールの投げ動作を行わせることに成功した。これについては 2009 年度研究紀要木下・西原他（6 自由度肩義手の協調動作制御システム用モーション作成の研究）に記載しているので、そちらを参照されたい。

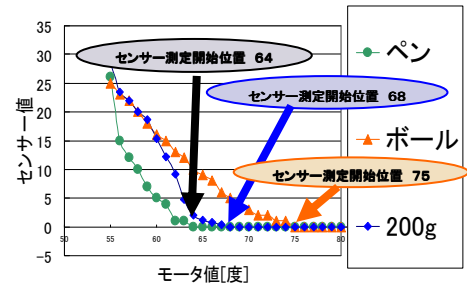


図 31. 図 28 を 3 つのデータのみにしたときの特性

モーターはホームポジションを 127 として、角度を 1 度ずつ減らせていきながら指先を閉じていく。このグラフより、ボールはその形状が他の 2 つよりも大きいため、センサー測定開始位置が 75 と、大きい位置になっている。また、ペンは細長いために、測定開始位置が 64 と一番小さくなっている。ペットボトルはその中ほどとなっている。このことから、

- ・ センサー測定開始のモーター位値が 75 以上ならば保持した物体は「ボール」
- ・ センサー測定開始のモーター位置が 64 以下ならば保持した物体は「ペン」
- ・ それ以外ならば保持した物体は「ペットボトル」

と仮定してプログラムの追加を行った。

<追加プログラム>

```

gl = Convert.ToByte(sp.Text);
if (result[1] > 0 && gl > 75)
{
    bo.Text = "ボール";
}
else if (result[3] > 0 && gl > 75)
{
    
```

```

        bo.Text = "ボール";
    }
    else if (result[1] == 0)
    {
        bo.Text = "ペットボトル";
    }
    else if (result[3] == 0)
    {
        bo.Text = "ペットボトル";
    }
    else if (result[1] > 0 && gl < 65)
    {
        bo.Text = "ペン";
    }
    else if (result[3] > 0 && gl < 65)
    {
        bo.Text = "ペン";
    }
}

```

※ result[1],[3] : 感圧センサーP1, P2 の値。
センサーが感知するのは主に親指と人差し指であるため。

今回完成したフォームを図 32 に示す。

図 32. 完成したフォーム

「物体表示」というボタンを押すことによって

図 33. 物体表示

図 33 のように保持した物体をわずか 3 種類だけが表示することができた。

8. 考察

8.1 プログラムについて

物体にある程度の圧力がかかったときにモーターの動きを制御し、それ以上加圧しないようにすることはできた。しかし、物体が滑り出したときに自動でモーターを動かすプログラムを作成することができなかった。物体が滑り出したときはセンサーで感知する圧力値に変化が生じるはずである。その変化に応じてモーターの角度を動かし、適正保持圧力に動かせるプログラムを作成することが、顔面方位制御方式の自動化を担うはずである。

8.2 手先具について

本研究では計 3 個の手先具を製作していった。手先具 3 号機は人間の手のように 5 本の指を持たせることができたが、指自体の自由度は根元の 1 自由度のみとなってしまった。菅君の指先の構造をうまく取り入れ、改良することによってより人間の手先に近いものができるはずである。また、軽量化のためには 1 個のモーター指先を動かすことが最善の方法であることは確かである。よって、ワイヤーを使って 5 本の指を動かせるという方法はかなり有用であるが、モーターに負荷がかかり過ぎないようにする工夫が必要である。つまり、より効率的にワイヤーにモーターの動きを伝えるように、ワイヤーチューブを使用すること、指先の開閉をうまく行わせることの 2 つを実現させる必要がある。

9. 謝辞

最後になりましたが、本研究にあたり、熱心にご指導していただいた電子工学科の全先生方ならびにさまざまな面でお世話、御教授していただいた実習係の方々に深く感謝いたします。

参考文献

- 1) 木下敏治：形状記憶合金アクチュエータを用いた上腕義手のマイクロコンピュータシ

- ステム, 託間電波高専研究紀要第 31 号, 119-127, (2003)
- 2) 斎藤之男他: 人間形ロボットハンドの高機能化に関する研究, バイオメカニズム, 155-165, (2002)
- 3) 赤澤堅造: 動力義手・装具の研究開発の現状と将来, BME(医用電子と生体工学), Vol.13, No.2, p.34-41(1999)
- 4) N. Dechev et al: Multiple finger, passive grasp prosthetic hand, Mechanism and Machine Theory 36, 1157-1173, (2001)
- 5) M. C. Carrozza et al: The Development of a Novel Prosthetic Hand-Ongoing Research and Preliminary Results, IEEE/ASME Transactions on Mechatronics Vol.7, 108-114, (2002)
- 6) Jimenez et al: First Approach of Pneumatic Anthropomorphic Hand, 2001 Proceedings of the 23rd Annual EMBS International Conference, October 25-28, Istanbul, Turkey, 1412-1414, (2001)
- 7) William Craelius: The Bionic Man: Restoring Mobility, SCIENC Vol 295, 1018-1021 (2002)
- 8) Peter J. Kyberd et al: An Intelligent Anthropomorphic Hand, with Automatic Grasp, Robotica volume 16, 531-536, (1998)
- 9) Peter Kyberd: The intelligent hand, IEEE REVIEW, 31-35, (2000)
- 10) C. Whiteley et al: Using practice led design research to develop an articulated mechanical analogy of the human hand, Journal of Medical Engineering & Technology, Volume 22, Number 5, 226-232, (1998)
- 11) B. Massa et al: Design and Development of an Underactuated Prosthetic Hand, 2002 IEEE International Conference on Robotics & Automation, 3374-3379, (2002)
- 12) Richard F. ff. Weir: Design Issues in the Development of an Externally-Powered Partial Hand Prosthesis, BED-Vol. 39, Advances in Bioengineering, 313-314, (1998)
- 13) Richard Colwell et al: The Ut Prosthetic Hand, ASME International Mechanical Engineering Congress & Exhibition, 1-3, (1996)
- 14) F. Sebastiani et al: Experimental analysis and performance comparison of three different prosthetic hands designed according to a bio mechatronic approach, 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 64-69, (2003)
- 15) A. Z. Escudero et al: Inverse Kinematics for a Parallel Myoelectric Elbow, 23rd Annual EMBS International Conference, 1427-1430, (2001)
- 16) Kevin Englehart: A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control, IEEE Transactions on Biomedical Engineering VOL50 NO.7, 848-854, (2003)
- 17) Patrizia Scherillo et al: Parallel Force/Position Control of a Novel Biomechatronic Hand Prosthesis, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 920-925, (2003)
- 18) Yiorgos A. Bertos et al: Microprocessor Based E.P.P Position Controller For Electric-Powered Upper-Limb Prostheses, 19th International Conference IEEE/EMBS Oct. 30 Nov.2, 2311-2314, (1997)
- 19) Chih-Lung Lin et al: A Speech Controlled Artificial Limb Based on DSP Chip, 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society Vol.20 No5, 2704-2705, (1998)
- 20) Han-Pang Huang et al: DSP-Based Controller for a Multi-Degree Prosthetic Hand, 2000 IEEE International Conference on Robotics and Automation, 1378-1383, (2000)
- 21) 木下敏治他: 形状記憶合金ロボット(肩義手自在 2 号)の顔面方位制御システムに関する基礎的研究, 日本産業用ロボット工業会, No. 85, 70-76, (1992)
- 22) 久本将貴: 6 自由度肩義手の顔面方位制御システムの開発, 平成 20 年度中国四国地区高等専門学校 専攻科生研究交流会要旨集, 95-96, (2008)
- 23) 菅 直人: バイオメタルを用いたロボットフィンガの開発, 2005 年度卒業論文, (2005)
- 24) 木下敏治他: RC サーボコントローラー AGB65-RSC を用いた 6 自由度肩義手の協調動作制御システムの開発, 託間電波高専研究紀要第 36 号, 47-55, (2008)
- 25) 有限会社ガリバー【著】: Visual C# 2005 基礎 300 の技, 技術評論社, (2007)