

RC サーボコントローラ 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

木下 敏治* 西山 功一** 久本 将貴*** 高橋 慎弥****

Development of Coordinated Motion Control System for a SDP(shoulder disarticulation prosthesis) with 6 degrees of freedom using RC servo controller AGB65-RSC

Toshiharu KINOSHITA , Kouiti NISHIYAMA , Masaki HISAMOTO and Shinya TAKAHASHI

Synopsis

The purpose of this research is to develop a SDP that is lighter and has more torque than the one developed 2006 year at our college. 2007 year's SDP has a simple structure with 6 degrees of freedom. We have used RC servo motor that has more torque for the actuator, RC servo controller AGB65-RSC for the control board, and Visual C# 2005 for a control software. We have checked whether the motors operate correctly according to input of values into the control software, and verified that, by using two motors simultaneously, the Θ_2 (shoulder disarticulation prosthesis) angle falls within an error of plus or minus 3 degrees.

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 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・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が少ない。

*電子工学科

**四国計測

***専攻科電子通信システム工学専攻

****川崎テクノサービス

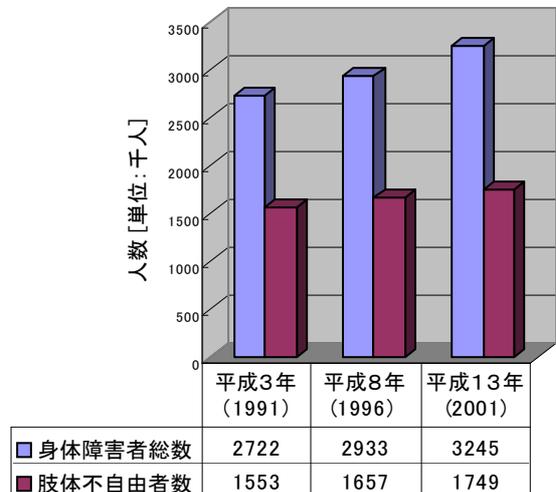


図 1. 身体障害者(18 歳以上)の全国推計数

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研究・開発を行った。

2. 研究の概要

軽量で安価な義手の作成を目標に、2006年度専攻科2期生新原君が製作した6自由度電動義手を元にして、より軽量で高出力な電動義手の製作を行った。

アクチュエータには軽量で高出力なRCサーボモータ KRS-4014HV Red Version (近藤科学製)等を使用し、新たに義手を製作した。制御基板にはRCサーボコントローラ AGB65-RSC (浅草ギ研製)を用いた。制御ソフトについては Visual C# 2005を用いて作成した。そのソフトでRCサーボモータの教示機能を利用して目標値通りに義手が動作するかを実験した。図2は製作した義手の外観である。

3. 製作した義手の構造

6自由度 (θ_1 上腕水平回転部, θ_2 上腕挙上部, θ_3 上腕回旋部, θ_4 肘屈曲部, θ_5 前腕回旋部, θ_6 手の開閉部)のシンプルな構造で義手の材料には軽量なアルミ板とアルミチャンネルを使用した。アクチュエータは大きなトルクが必要な上腕部には近藤科学の KRS-4014HV (電源電圧 10.8V, トルク 40.8 kg \cdot cm, 重量 65.0g), それほどトルクの必要のない前腕部には KRS-4024HV (電源電圧 10.8V, トルク 10.5kg \cdot cm, 重量 48.5g)を使用した。上腕挙上部にはモータを2個使用してトルクの増大をはかった。手についてはものをつかみやすくするため大きめに製作し、さらに滑り止めをとりつけた。その結果、義手の全長は 1660[mm], 重量は 1124[g]となり、236[g]軽量化することができた。表1が製作した義手の詳細なデータである。



図2. 2007年度製作した6自由度肩義手

表1. 肩義手の骨組みの重量とモータの重量

全長			
縦	830mm	横	190mm
上腕部			
縦	400mm	横	90mm
前腕部			
縦	430mm	横	170mm
重量			
上腕部	603.5g	前腕部	520.5g
モータ重量	260g	モータ重量	162g
骨組み	343.5g	骨組み	358.5g
全体重量		1124g	

4. 制御方法

4.1 RCサーボモータ

研究で使用したRCサーボモータは入力信号と電源のみで動作する。駆動電圧はHVタイプなので9[V]~12[V]である。入力信号として約700[μ sec]~2300[μ sec]のパルス数を数[msec]~20[msec]の周期で入力することでモータの角度を1度単位で決められる。またPICでも制御が可能である。通常に動作させる場合は入力したパルス幅に相当する角度に制御され、260度に及ぶサーボ動作範囲の位置制御が可能である。

この他に教示機能 (ポジションキャプチャー) が利用できる。100[μ sec]~200[μ sec]のパルスが入力されると、サーボは 100[μ sec]以内に信号線を入力から出力に切り換えてサーボの位置 (回転角度) に対応するパルス幅を出力する。このパルス幅を外部で測定することにより、サーボの現在位置情報を取得することができる。この教示機能を使って、肩義手の動作角度を読み取り、測定する。

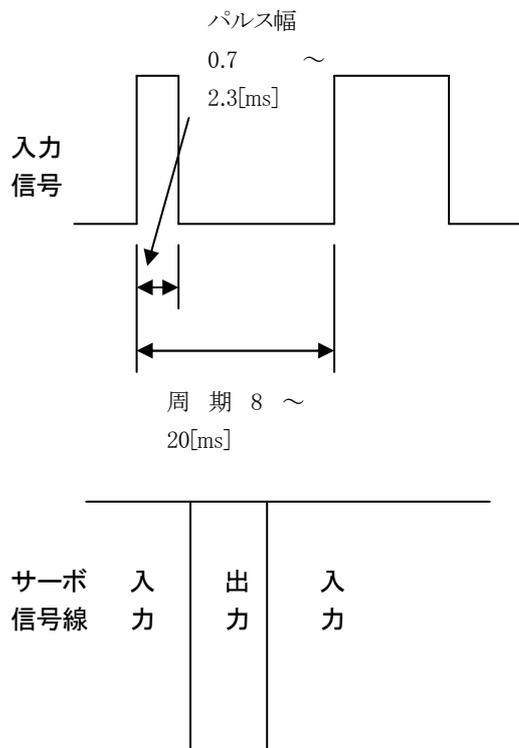


図 3. 入力信号とサーボ信号との関係

4.2 制御基板

制御基板には RC サーボコントローラ A5-RSC を使用した。これをパソコンと接続して義手の制御を行う。基板用電源は 7.5[V] から 25[V] で、瞬間耐圧は 35[V]。寸法は 34×34[mm] で重量は 9[g] と小型で軽量である。また、このコントローラは

- 12 個の RC サーボの回転位置と速度を 1 つのシリアルラインで制御できる
- ID を変更し 4 つの AGB65-RSC を接続することで最大 48 個の RC サーボモータを同じシリアルラインで駆動できる
- 他の AGB シリーズ (4 点感圧センサ) と接続でき、同じシリアルラインで使用できる

- RC サーボへのスピード指定が可能
- 通信速度は 115[Kbps], 9600[bps] のどちらかを選択
- 教示機能に対応しているため、脱力時にその角度を読み取ることができる
- 8 チャンネルのアナログ入力に対応している以上のような特徴がある。今回、通信速度は標準設定の 9600[bps] で実験を行った。

この制御基板を使用した理由については、

- 制御できる最大モータの数が 48 個と多いこと
- もともと PIC 等のマイコンでの制御を主としてつくられたものなので PIC への移行が容易にできること
- 他の AGB シリーズと接続でき、8 チャンネルアナログ入力に対応しているため容易にセンサ入力を取り入れることができる

これらの理由から AGB65-RSC を使用した。図 4 は実際の基板の写真である。

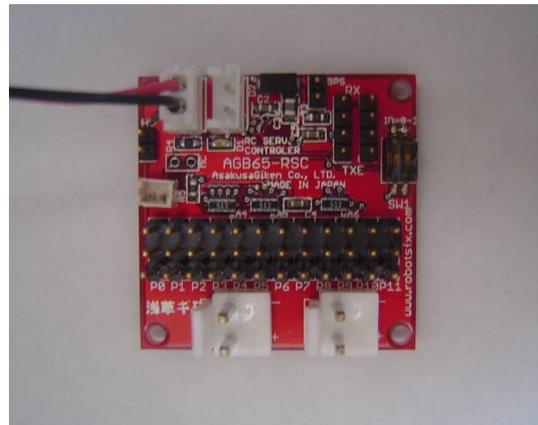


図 4. AGB65-RSC

AGB65-RSC は TTL 電圧レベル (論理 0 が 0[V], 論理 1 が 5[V]) で、パソコンは RS232C 電圧レベル (論理 1 が +3[V] ~ +12[V], 論理 0 が -3[V] ~ -12[V]) と異なるので AGB65-RSC は直接パソコンに接続できない。そこで、AGB65 シリーズ PC 接続ボード AGB65-232C (浅草ギ研製) を使用する。AGB65-232C は AGB65 シリーズとパソコンを接続するためのモジュールで、電圧レベルの変換を行い、通信を可能とするものである。基板用電源は 5[V] (AGB65-RSC の基板より電源を取得) で寸法は 19×10[mm] で重量は 1[g] となっている。図 5 はこれらを接続した時のブロック図である。

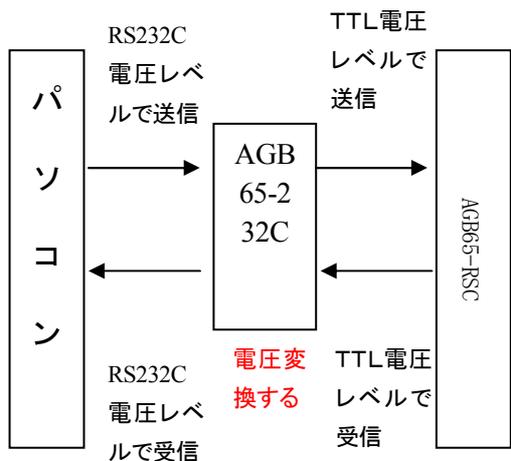


図 5. AGB65-RSC と AGB65-232C とパソコンとの接続図

4.3 制御ソフト

制御ソフトの作成には「Visual C# 2005」を使用した。「Visual C# 2005」を選んだ理由は、ボタンやテキストボックスを配置したりして、好きなように画面をデザインすることができ、シリアルポートを使った通信をするためのプログラム記述が容易にできるからである。今回は義手の製作から行ったため義手が入力した目標値通りに動作するか確認する必要があった。そのため動作実験を主な目的とするソフトを作成した。内容は目標値を入力し、目標値通りにモータの角度を動かす、RC サーボモータの教示機能を使って、モータの角度を読み取りそれを画面上に表示させるというものである。これは浅草ギ研のホームページに掲載されているプログラムを参考にして作成した。図 6 は作成したソフトの操作画面である。上腕水平回転部から手の開閉部までの制御を個別または全てできるようになっており、上腕挙上部についてはモータ 1 個で駆動させる場合と 2 個で駆動させる場合の 2 通りできるようにした。

操作方法について上腕水平回転部を例にして説明を行う。始めに ID, 番号, 速度, (初期値), 目標値, を入力する。「ID」とは基板の ID のことで基板どうしを接続した時にそれぞれを識別するためのもので、基板上で 0~3 を選べるようになっている。初期設定は 3 となっているので図 6 には 3 が入力されている。「番号」は個別の RC サーボモータを指定するもので図 6 のように 0 が入

力されている場合は基板の P0 の位置に接続されたモータを駆動させることになる。「速度」は回転速度を指定するのではなく、指定した位置に止まるまでの時間で、入力値×15[ms]となる。図 6 では 50 が入力されているので目標値まで回転するのに 750[ms]かかることになる。「目標値」の入力は直接入力するかボタンを押して 5 度ずつ増減させ調節する。次に「駆動」ボタンをクリックし義手を駆動させる。RC サーボモータを駆動させるには「255」、「ID」、「送信命令数」、「2」、「番号」、「目標値」、「速度」をモータに送信することが必要で、このボタンをクリックした時にこれらのデータを送信するようにプログラムされている。「255」はシンクロバイトと呼ばれる基板にデータの通信開始を知らせるためのデータで、「2」はモータ 1 個を動かす時の決められた命令の番号である。最後に「測定」ボタンをクリックし測定値を表示させる。このボタンをクリックすると教示機能を利用するために必要な命令（「255」、「ID」、「送信命令数」、「2」、「番号」）を送信し、モータ側から送信されたデータを受信すると画面に表示されるようにプログラムされている。しかしこの部分にバグがあるので改良する必要がある。以上が上腕水平回転部の操作法であり、他の部分もこれと同じようになっている。また、上腕挙上部はモータの個数「1 個」を選択すると 2 個のモータで駆動させるためのボタンを無効にしてクリックできないようにした。逆にモータの個数「2 個」を選択するとモータ 1 個で駆動させるボタンが無効になる。モータ 2 個で駆動させる場合は「駆動 A, B」ボタンをクリックした時にモータを駆動させる命令を二つのモータ同時に送ることで実現している。「全て駆動」ボタンをクリックした時は上腕水平から手の開閉まで、それぞれの部分に入力されている速度と目標値に従って全てを駆動させるようになっている。「全て初期値へ」ボタンをクリックした時は上腕水平から手の開閉まで、それぞれの部分に入力されている速度と初期値に従って全てを駆動させるようになっている。

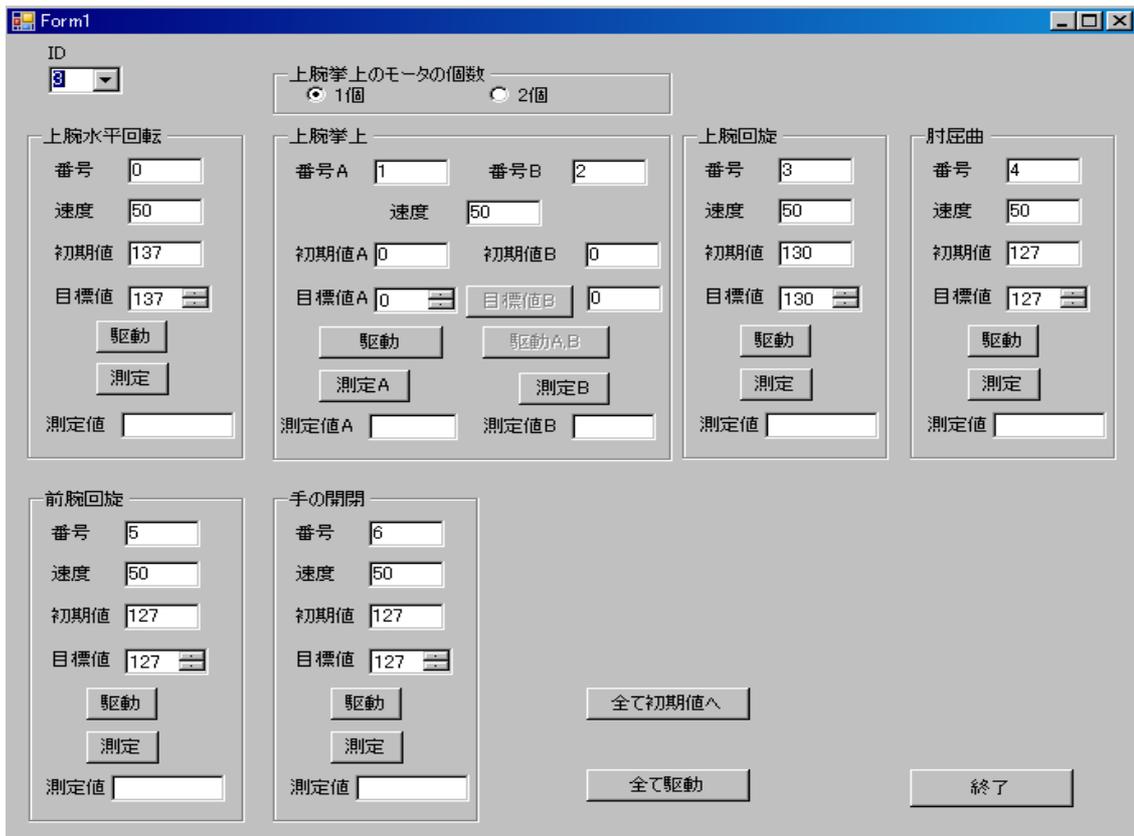


図 6. 操作画面のデザイン

5. 測定方法

義手の可動範囲を調べるため次のようにして測定を行った。

RC サーボモータの駆動電圧は安全のため 9[V]で測定

- 1)目標値を入力して義手を駆動させる
- 2)分度器により義手の上昇角度を目測する
- 3)教示機能を用いて RC サーボモータの角度を測定する。この時モータが脱力状態になるので手で支えて義手が出らないようにする。(教示機能を使用したときは量子化誤差により 2, 3 度の誤差が生じる。)
- 4)義手が上昇しなくなるまで(2)~(4)の操作を繰り返す(上腕挙上の測定時は 2 個のモータの角度のずれを少なくするため、目標値まで駆動させた後、一度初期値に戻してから次の目標値まで駆動させる)

6. 測定結果

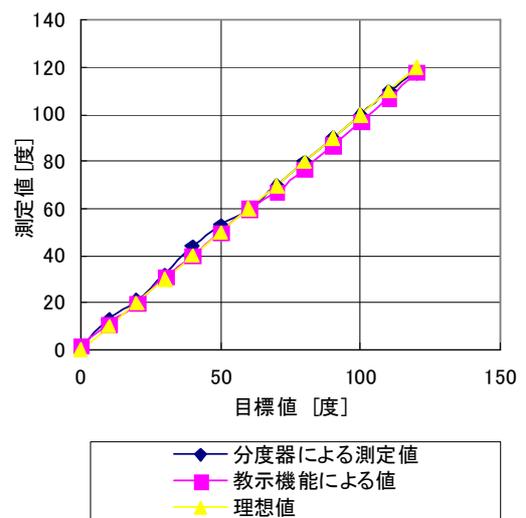


図 7. 肘屈曲の目標値と測定値の関係

表2. 肘屈曲の場合

電源電圧 9[V]			
目標値[度]	測定値[度]	教示値[度]	理想値[度]
0	2	2	0
10	13	11	10
20	21	20	20
30	32	31	30
40	44	40	40
50	53	50	50
60	60	60	60
70	70	67	70
80	80	77	80
90	90	87	90
100	100	97	100
110	110	107	110
120	118	118	120

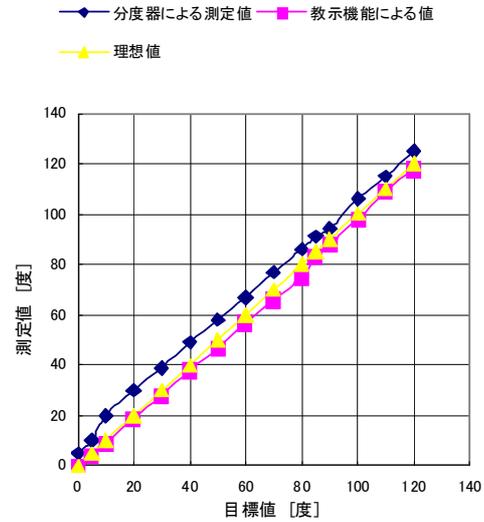


図9. 上腕挙上の目標値と測定値の関係 (肘0度でモータ1個、手の開閉部外した場合)

表3. 肘と上腕挙上を曲げた時の
上腕回旋の可動範囲 ○：0~255[度]

	上腕挙上 0 [度]	上腕挙上 30 [度]	上腕挙上 60 [度]	上腕挙上 90 [度]
肘 0 [度]	○	○	○	○
肘 30 [度]	○	○	○	○
肘 60 [度]	○	○	○	○
肘 90 [度]	○	○	○	○

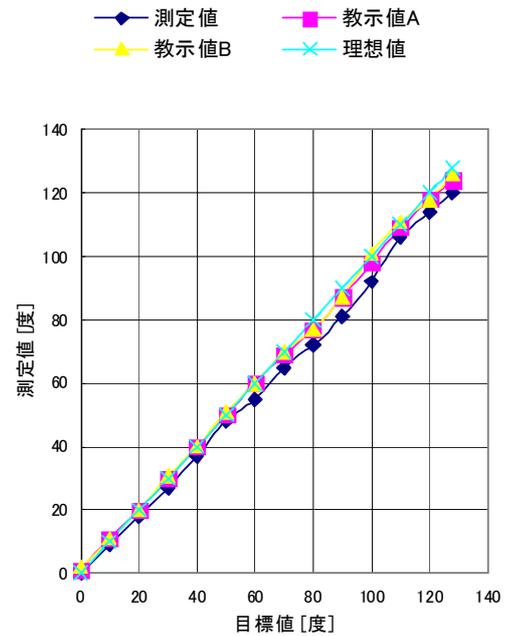


図10. 上腕挙上の目標値と測定値の関係 (肘0度でモータ2個の場合)

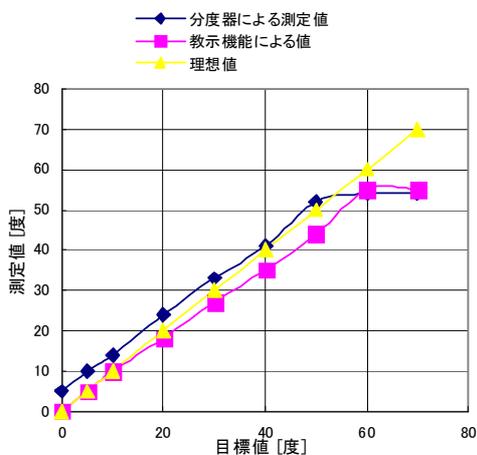


図8. 上腕挙上の目標値と測定値の関係 (肘0度でモータ1個の場合)

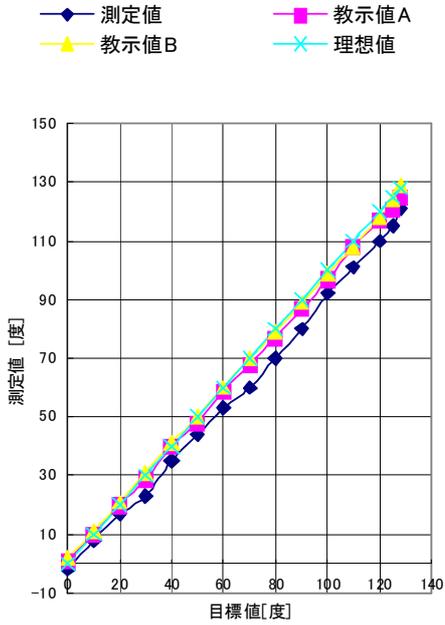


図11. 上腕挙上の目標値と測定値の関係 (肘30度でモータ2個の場合)

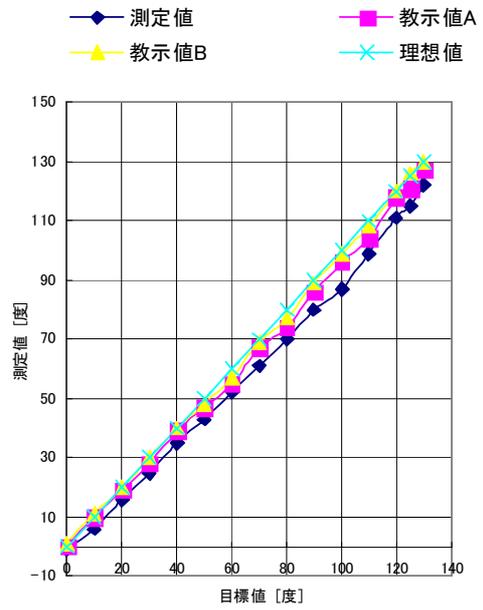


図13. 上腕挙上の目標値と測定値の関係 (肘90度でモータ2個の場合)

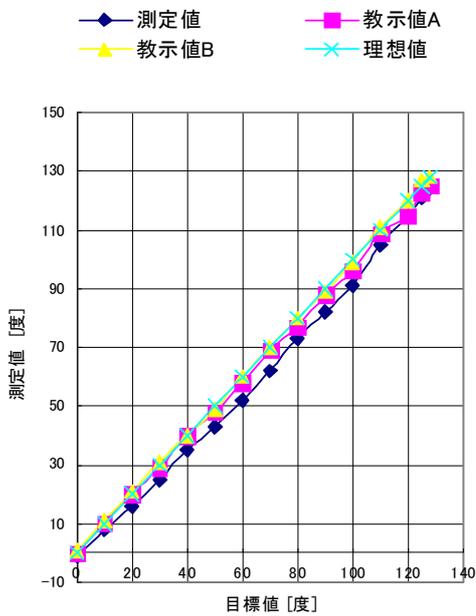


図12. 上腕挙上の目標値と測定値の関係 (肘60度でモータ2個の場合)

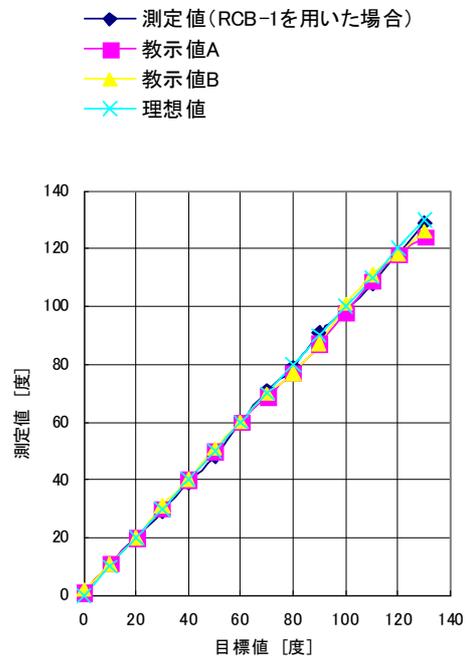


図14. 上腕挙上のRCB-1での測定値と教示値の比較(肘0度でモータ2個)

7. 考察

(1) 義手について

・肘屈曲部 (θ₄)

図7のように分度器の値, 教示値共に目標値通りの値になった。

・上腕回旋部 (θ₃)

表3のように肘の角度と上腕挙上の角度がどんな値でもモータの駆動範囲の限界まで駆動した。

・上腕挙上部 (θ₂)

モータ1個で上腕挙上を行った場合は図8のように目標値通りに義手は動作しなかった。目標値が60度を超えると義手は上がらなくなった。肘を曲げた状態で駆動させた場合も同様の結果となった。今回製作した義手は手の開閉部の重量が重いので手の開閉部を外した状態での動作実験も行ってみた。その結果が図9であり, 目標値通りに動作した。

モータ2個で上腕挙上を行った場合は図10のように目標値通りに義手を動作させることができ, 130度近くまで上げることができた。肘を30度, 60度, 90度曲げた状態でも目標値通りに動作した(図11~図13)。以上の結果より上腕挙上部はモータを2個使用の方が良いと思われる。手の開閉部を軽量化する事でモータ1個でも上腕挙上部を動作させることは可能だが, ものを持った場合にトルク不足となるのでモータが2個必要である。しかしモータを2個使用したときはそれぞれのモータの角度が少しくずれてしまった場合, 義手を動作させるにつれてそのずれが大きくなり最終的にモータが壊れてしまう原因となる。そのため, ずれが生じないように正確に製作を行う工夫や, ずれた場合にそれを補正するようなプログラムが必要である。また連続で動作させる場合モータの温度が高くなりすぎて故障する可能性があるので, ファンを取り付けるなどの放熱させるための工夫が必要となる。

(2) 制御基板について

今回の研究では2005年度から使用している近藤科学の制御基板 RCB-1 とは異なる浅草ギ研の制御基板 AGB65-RSC を使用して測定を行った。その結果図14のように同じような測定結果を得ることができた。このことから AGB65-RSC でも RCB-1 と同じように制御可能であることが確認できた。今回の測定は通信速度 9600[bps]で行ったので基板の設定を変更して通信速度 115.2[kbp

s]でも同様に動作するか確認することが今後の課題である。

(3) 制御ソフトについて

制御ソフトの作成には Visual C# 2005 を使用したが, シリアル通信を利用するための記述が容易にできる, プログラムのコードの入力ミスが入力時に画面上に表示されるのでミスが少なくすむ等, とても使いやすかった。

作成したソフトにはシリアルポートに受信された教示値のデータを画面上に表示させる箇所のプログラムにバグがある。これを直すことが一番の課題である。このバグは「シリアル通信受信イベント」を使用した際に起きるバグなのでデータを受信した時に表示させるのではなく, ボタンをクリックした時にデータの値を表示させるように変更することでバグが無くなることは確認できた。しかしこの方法ではボタンが多くなり操作性が悪くなるので「シリアル通信受信イベント」を利用した方が良い。また, 「測定」ボタンをクリックしたときに目標値と全然違う「255」という値が表示される誤動作が起こる場合があったので, これをなくすことが課題である。今回は測定時に, この誤動作が起こった場合もう一度「測定」ボタンをクリックすることで正確な値を表示させた。モータが目標値に到達して止まる前に「測定」ボタンをクリックした場合も同じ症状が現れるので, プログラムのバグが原因でデータの通信がうまくいかないことが誤動作の原因だと思う。

今回自作した制御ソフトは近藤科学の「Heart To Heart」と比べると, 操作性が悪く機能も不足していた。しかし, 「Heart To Heart」よりも速度の指定が細かくできる, 操作画面を自分の好きなように作成できる等の利点もあった。また, センサ入力により, いろんな制御方式を利用することも可能で, C#に対する十分な知識があれば「Heart To Heart」以上に使いやすく多機能なソフトを作成できると思われる。

8. 謝辞

最後になりましたが, 本研究にあたり, 御指導していただいた電子工学の先生方ならびにさまざまな面でお世話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)赤澤堅造：動力義手・装具の研究開発の現状と将来 BME Vol.13, No.2, 34-41(1999)
- 2)新原洋人：RC サーボモータを用いた 6 自由度肩義手の協調動作制御システムの開発, 平成 18 年度専攻科特別研究論文集(2007)
- 3)川口輝久：ゼロからスタート Visual C#.NET 超入門, 毎日コミュニケーションズ, (2003)
- 4)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制御システム, 本誌, 第 31 号, 119-127(2003)
- 5)木下 他:RC サーボモータを用いた 6 自由度肩義手の協調動作制御システムの開発, 本誌, 第 35 号, 33-44(2007)
- 6)木下 他:高トルク・軽量の RC サーボモータを用いた 6 自由度肩義手の開発, 本誌, 第 35 号, 45-52 (2007)