

包括的な支援環境における義肢装具の研究開発支援ツール開発 Developing prosthetic and orthotic research tools in comprehensive assistive environment

梶谷 勇[†]
Isamu Kajitani

1. はじめに

昨今の情報通信技術やロボット技術の進歩により、様々なセンシングデバイスや駆動デバイスなどが容易に入手できるだけにとどまらず、それらを組み合わせた統合システムの構築を支援するソフトウェア技術も利用できるようになってきた。今後、我々が生活する環境の中に多くのセンサやデバイスが組み込まれ、意識せずにさまざまな支援を受けることのできる世の中になっていくであろう。

このような技術革新の恩恵を障害者や高齢者の生活支援に活用すべく、機器の連携による包括的な支援環境の構築を目指した研究開発が行われている。例えば厚生労働省の平成21年度障害者保健福祉推進事業(障害者自立支援機器等研究開発プロジェクト)において、著者らは「障害者が自立して住みやすい住環境モデルの構築」というテーマの研究開発を行い、複数の異なる機能モジュール群を連携して支援を行う3つの包括支援環境(①歩行困難な肢体不自由者を支援する住環境モデル、②脳卒中後遺症による脳機能障害者の調理訓練を支援する住環境モデル、③視覚障害者を支援する住環境モデル)を構築した。[1]

一方、事故や病気などによる手足の切断、あるいは先天的な障害に対して機能の一部を再建、あるいは補完するために用いる義肢[2]や、手足の麻痺などにおいて活用される装具[2]においても、近年、電動化・電子制御化の流れが徐々に進みつつあり、既に製品として入手可能なものも存在する。

電動あるいは電子制御された義肢装具として、かつては電動義手[3]が代表的なものであったが、近年では、義足や装具においても電動・電子制御の製品が開発されている。電動義手としては、Otto Bock社のMYOBOCK®システムが代表的で、3本指のハンドによる把持動作が基本となっている。他Touch Bionics社のi-Limb™ Handや、Otto Bock社のMichelangelo®など5本指を持つハンドや、肘や肩関節に相当する部品も多くの種類[2]が販売されている。

義足に関する技術は著しく進歩しているが、その先駆けとなったのはナブテスコ社のインテリジェント義足[4]で、膝関節のシリンダを電子制御することで、膝下の動きを歩行スピードにあわせて調整可能である。同様に、Otto Bock社のC-Leg®やOssur社のRHEO KNEE®でも膝関節の動きを、Ossur社のPROPRIO FOOTでは足関節を電子制御する。また、Ossur社のSymbionic Leg®は前述のRHEO KNEE®とPROPRIO FOOTとを組み合わせた製品で、これらを連携した制御により快適な歩行を実現する。さらにOssur社のPOWER KNEE™では膝関節を電動モータで駆動することにより、容易に椅子から立ち上がったたり、階段をのぼったりできる。電子制御された装具としては、MR流体ブレー

[†] 国立研究開発法人産業技術総合研究所 National Institute of Advanced Industrial Science and Technology

キを用いた下肢装具[5]などが開発されているほか、OTTOBOCK社ではC-Leg®の技術を用いた膝関節装具(C-Brace®)[6]が開発されている。またロボット技術を用いたパワーアシスト装具[7]の開発も活発である。

このように多機能化する義肢装具の現状がある一方、その結果として、どのように複数の機能を人間の意志で操作するかが依然として課題として残っている。例えば電動義手においては、その解決方法として、筋電のパターン認識による方式が長く研究され、研究室内では良好な結果が得られてきたものの、実際の生活の中で活用できるレベルの物はいまだに存在していない。このような工学的なアプローチに対し、医学的な解決方法としてTMR(Targeted Muscle Reinnervation)[8]と呼ばれる手術を用いた方式が開発された。そこでは、切断前に腕に伸びていた神経の末端を胸の筋肉などに再接合する手術を施すことで、手を動かそうとすると神経を再接合した胸筋等が収縮し、その時の筋電を用いることで、多機能な電動義手を操作できる。

しかしながら、この方式では高度な技術を要する手術を必要とするために誰もが気軽に試せるわけではない。並行して、工学的な解決手段の研究開発を進めておくことで、将来的な相乗効果を期待できる。そこで我々は、これまでの包括的な支援環境に関する研究開発[1]の知見を活かし、住環境との相互作用を活用することによって、電動義肢装具の操作を容易にする方式の構築を目指している。例えば、把持対象の情報(傾きや柔らかさなど)を環境側から入手し、義手の操作を部分的に自動化(傾きや把持力を自動調整)することで、操作者の負担の軽減することを目指している。しかしながら、このような包括支援環境は発展途中にあり、どのような支援の形態が有効であるのか、今後、多くの試みを行っていく必要がある。本論文では、このような包括支援環境における電動義肢装具の研究開発を活性化するため、そこで必要となる2つの研究開発支援ツールの開発と評価を報告する。

以下2章では、本研究の背景として包括支援環境の基盤技術として用いたロボット用のソフトウェア技術であるRTミドルウェア[9]と、電動義肢用の通信技術として開発が進められているPDCP(Prosthetic Device Communication Protocol)[10]について説明する。続く3章、4章で要素技術の開発と評価について述べた後、5章で本論文をまとめる。

2. 背景

2.1 RTミドルウェア(RTM)

産業用ロボットのような同一仕様のロボットをたくさん生産してコスト競争するのではなく、今後、我々の生活空間内で人間と共存しながら活動する生活支援ロボットでは、多品種少量生産による付加価値の競争になると考えられる。このため、これまでとは異なるロボット生産方式として、機能モジュールごとに個別に開発された部品を必要に応じ

て組み合わせてロボットを構築するような方式が必要である。

このような生産方式を支える基盤ソフトウェア技術として、ロボット用のミドルウェア (RT ミドルウェア: 以下では RTM と略記) は、NEDO (新エネルギー・産業技術総合開発機構) の 21 世紀ロボットチャレンジプログラム「ロボット機能発現のための要素技術開発」において研究開発がスタートした。その後、国際標準化団体 OMG (Object Management Group) でインタフェース仕様の標準化が進められ、2008 年 4 月に OMG 公式標準仕様となった。

RTM は、RT コンポーネント (以下では RTC と略記) と呼ばれるソフトウェアモジュール群と、それらを相互に接続したロボットシステムを構築するソフトウェアのフレームワークである。Microsoft Windows や Linux など複数の環境において動作し、また、C++ や Python など複数の開発言語に対応するため、異なる環境で開発した RTC を相互に接続して一つのロボットシステムを構築することが可能である。

RTC としては、画像処理用のソフトウェアモジュール等がサンプルとして用意されているだけでなく、インターネット上のポータルサイトにおいて様々なモジュールが公開されている。さらに RT ミドルウェアコンテストが毎年開催され、新しいソフトウェアモジュールが活発に開発されている。

前述の「障害者が自立して住みやすい住環境モデルの構築」プロジェクト[1]では、模擬住宅環境内の家電、窓、照明、センサ類に加え、家具や什器を電動化するモータモジュール等が RTM を用いてネットワーク化された。そこでは、肢体不自由者がビジョンや音声認識を用いて操作可能な住環境システムや、住環境内のセンサ群と連携して調理訓練を支援する作業支援ツール、AR マーカーや無線タグを用いて視覚障害者の生活を支援する住環境モデルが構築された。

2.2 PDCP (Prosthetic Device Communication Protocol)

多機能な電動義肢装具を目指し、複数の機能モジュールを組み合わせる方式を採用したものとしては ToMPAW (Totally Modular Prosthetic Arm with High Workability) プロジェクトがある。[11] ここでは、ビルやホームオートメーションで用いられる LonWorks ベースのコミュニケーションプロトコルが用いられ、複数モジュールを連携させて高機能な電動義手を実現した。福祉機器全般では、EC (現 EU) の TIDE-Project で開発された CAN (Controller Area Network) ベースの M3S (Multiple Master Multiple Slave intelligent interface for the rehabilitation environment) がある。

これらの流れを受け、電動の義肢装具におけるオープンなコミュニケーションプロトコル作成が開始され、現在、University of New Brunswick を中心に PDCP (Prosthetic Device Communication Protocol) として標準化がすすめられている。[10]

PDCP は M3S と同様に CAN ベースのコミュニケーションプロトコルである。PDCP では 1 つのホスト (アービトラータと呼ぶ) と複数の機能モジュール (ノードと呼ぶ) からなるシステムを構成する。アービトラータは各ノードのバインド、ノードのパラメータ設定などを行うことで、

ノード間のデータのやり取りを制御する。例えばノードの入力ポートに対してデータソースのノード ID を設定することで、そのノードはデータソースとして設定された ID のノードから送られたデータのみを取得する。また、各 PDCP ノードはデバイスビーコンと呼ばれるデータを定期的にアービトラータに対して送信する。

送受信するデータフォーマットは、先頭の 0.5 バイトで優先度とデータの方向を決定し、例えば最も優先度の高いバインドリクエストは 6、アービトラータからノードへのコマンドなどは 3、ノードから出力されるデータなどは 2 である。続く 1 バイトがノード ID、次の 0.5 バイトがデータ長で、その先にデータ長で指定した長さのデータが続く。例えば (6) (02) (7) (01) (0100) (0A00) (0200) は、先頭 (6) は前述のように優先度とデータの方向で、これは最も優先度が高く、ノードから送信されるデータを表している。続く (02) はこのノードのデフォルトのノード ID で、バインドリクエストした際にこの ID が未使用であれば、そのまま使用される。続く (7) がデータ長で、次の (01) がバインドリクエストコマンドである。その後の 6 バイトのデータは、2 バイトごとにベンダー ID、製品 ID、シリアル番号を設定することになっている。また、(3) (02) (4) (04) (04) (01) (01) は、先頭 (3) (02) が優先度とデータの方向とノード ID で、次の (4) がデータ長、(04) がコマンド (set device parameter)、(04) がパラメータ ID、(01) がチャンネル番号、(01) がパラメータ値である。

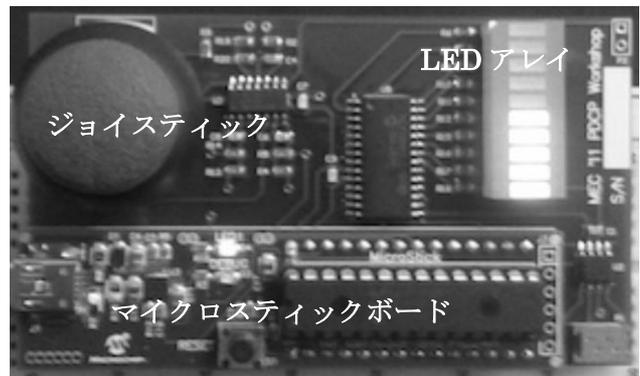


図 1 PDCP Workshop 用評価ボード

PDCP はまだ開発途中であるが、University of New Brunswick の開発チームが定期的にワークショップを開催しており、図 1 に示すワークショップ用の評価ボードを用いて、その機能を確認することが可能である。この評価ボードは MICROCHIP 社のマイクロスティックボード (図 1 下) と、入力デバイスとしてのジョイスティック (図 1 左上)、出力デバイスとしての LED アレイ (図 1 右上) に加え、通信用の CAN インタフェースを持つ。前述の (3) (02) (4) (04) (04) (01) (01) は、ノード 02h のチャンネル 01h の出力をイネーブルする命令で、例えばワークショップ用評価ボードの場合、ジョイスティックの X 軸側の AD 変換値を出力する。LED アレイを点灯する場合は、例えば (3) (05) (4) (04) (07) (01) (02) を送ると、ノード 05h のチャンネル 01h に対し、データソースをノード 02h と設定する。その後、ノード 02h の出力をイネーブルすると、ノード 02h のジョイスティック値に応じてノード 05h の LED アレイが点灯す

る仕組みである。なお、ワークショップ用評価ボードでは、データ送信用と受信用でファームウェアを入れ替えるようになっており、ここで説明した例では、ノード 02h は送信用、ノード 05h は受信用のファームウェアである。

3. 開発した研究開発支援ツール

ここでは、前章で説明した RTM と PDCP を連携し、包括支援環境における電動義肢装具の研究開発を実現するために開発した要素技術である RTM-PDCP 連携技術と、PDCP 対応 IO ボードについて述べる。

3.1 RTM-PDCP 連携技術

異なるネットワークを構成する RTM と PDCP を相互に接続するためには、図 2 に示す模式図のような両者をブリッジする機能が必要である。このため、RTC を仮想的なノードとして PDCP ネットワークに参加させる機能モジュール (PDCP_Wapper-RTC) を作成した。PDCP_Wapper-RTC は、自身が RTM 上の RTC であると同時に、他の RTC を仮想的な PDCP ノードとしてバインドし、PDCP ノードとの相互通信を実現する。(図 3)

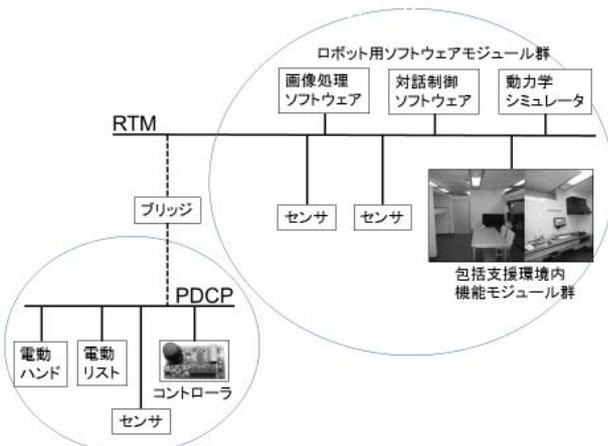


図 2 RTM-PDCP 連携の模式図

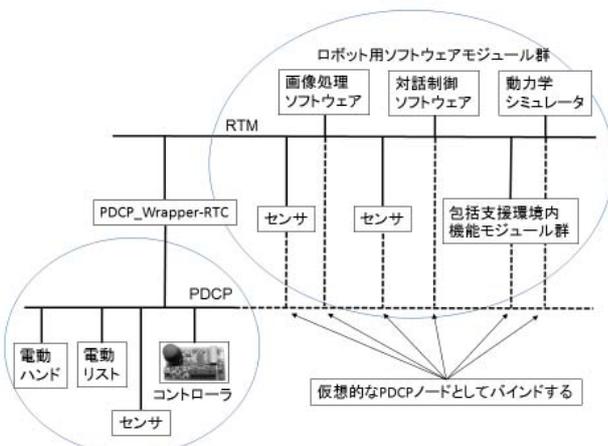


図 3 PDCP-Wrapper-RTC 模式図

図 4 に PDCP-Wrapper-RTC 実行時のスクリーンキャプチャ図を示す。PDCP-Wrapper-RTC は Python で書かれたソフトウェアモジュールである。図 4 の後ろ側のウィンドウが RTM のグラフィカル開発環境であるが、PDCP-Wrapper-RTC 起動時には、この図にあるように 1 つの入力ポート (ARB2NODE) と 2 つの出力ポート (NODE2ARB, BEACON) を持つ。ARB2NODE ポートはアービトラータからノードに送信する制御コマンドを入力するポートであり、このポートを用いることで、RTC から PDCP ノードを制御可能である。出力ポートは PDCP 側の通信データを観測するためにあり、NODE2ARB ポートは各ノードからアービトラータに届くメッセージを出力し、BEACON ポートはノードから届くデバイスビーコンを出力する。



図 4 PDCP-Wrapper-RTC 動作画面 (手前が PDCP-Wrapper-RTC のコマンドライン。後ろが RTM グラフィカル開発環境上で見た PDCP-Wrapper-RTC)

通常 RTC の入出力ポートは固定であることが多いが、PDCP-Wrapper-RTC の場合は接続する RTC の数が事前にはわからないため、必要に応じてポート数を自在に変えることができる仕様とした。図 4 手前のウィンドウが PDCP-Wrapper-RTC のコンソールウィンドウで、表 1 に示すコマンドで操作する。例えば RTC を PDCP ネットワークに参加させる場合、まず RTC を接続するポートを作成する必要があるため、`addinport` や `addoutport` コマンドを用いて入出力ポートを追加する。この場合、ノード番号は 80h 以降が割り振られる仕様となっている。ここで作成されたポートに対して RTC を接続すると、仮想的な PDCP ノードとして RTC が PDCP ネットワークに接続したことになる。なお、PDCP-Wrapper-RTC は PDCP のアービトラータとしても機能し、通常の PDCP ノードは PDCP-Wrapper-RTC に対してバインドリクエストを出すことでノード番号がバインドされる。この場合、各ノードは 7Fh 以下のノード番号が割り当てられる。また、`delport` コマンドで削除することができる。

表 1 PDCP-Wrapper-RTC のコマンド表

コマンド	機能
?	コマンド一覧表示
nodels	バインドされているノードリスト表示
nodecat	ノードの情報を表示
senddata	ノードに対してメッセージ送信
addinport	RTC入力ポートの追加
addoutport	RTC出力ポートの追加
delport	RTCポートの削除
getparam	ノードのパラメータを取得
setparam	ノードのパラメータを設定
reset	ノードをリセット
beacontest	デバイススピーコン確認

nodels コマンドはバインドされているノードのリストを表示し、nodecat コマンドは引数としてノード番号を与えると、そのノードの情報を表示する。以下、実行例を示す。

```

>nodels
02:PDCP NODE
80:RTC NODE
81:RTC NODE
>nodecat 02
- node_id: 02
- node: PDCP node
- device_vender_id: 02
- device_product_id: 0100
- device_serialt_number: 0A00
    
```

なお、80h 以降のノードに対して nodecat コマンドを用いた場合、RTC 未接続の場合は

```

>nodecat 80
- node_id: 80
- node: RTC node(N.C.)
- connector_profile:
  - name: N.C.
    
```

と表示されるが、接続済みのポートの場合は以下の例のように****の部分で接続済みのコンポーネントの情報が確認できる。

```

>nodecat 81
- node_id: 81
- node: RTC node(outport)
- connector_profile:
  - name: PDCP_Wrapper0.dynamicPort81_****
    
```

3.2 PDCP 対応 IO ボード

図 1 に示すワークショップ用の評価ボードは外部入出力を持たないため研究開発に用いることはできない。そこで、

AD変換入力 (3ch) とモータ制御用のPWM出力 (4ch) を持つ研究開発用 IO ボードを製作した。(図 5)

前述のジョイスティック値の取得やLEDアレイの点灯と同様に、各ノードの入出力チャンネルのパラメータを設定することで、AD 変換値を取得したり、モータの回転を制御したりできる。本稿では、通信機能に加え、PWM 出力によるモータ制御機能と AD 変換機能の動作検証を報告する。

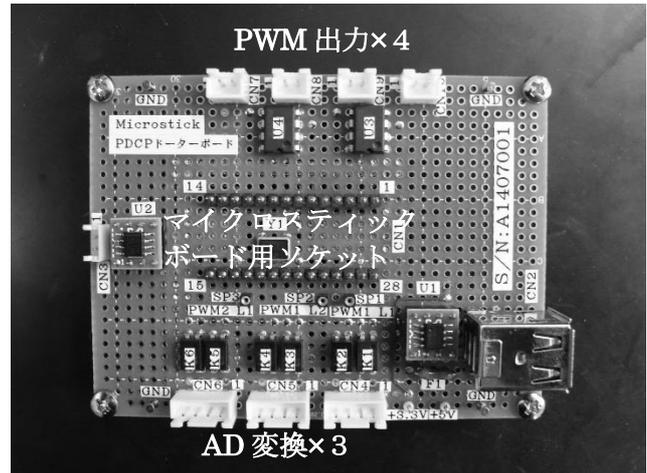


図 5 PDCP 対応 IO ボード

4. 開発ツールの評価

4.1 RTM-PDCP 連携実験

ここでは PDCP_Wrapper-RTC の機能を検証するため、入出力ポートの追加と、RTM と PDCP ノード間の通信を確認する実験について述べる。4.2 で詳説する実験と同じくノート PC に RTM の実行環境を構築して PDCP_Wrapper-RTC を実行し、USB-CAN インタフェースで PDCP ネットワークと接続して実験を行った。

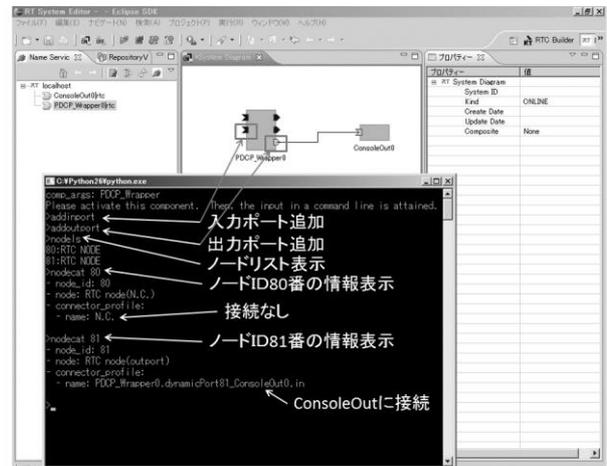


図 6 入出力ポートの追加実験のスクリーンキャチャ

4.1.1 RTC 用入出力ポート追加実験

PDCP_Wrapper-RTC の基本機能である RTC 接続ポートの追加機能を検証する。図 6 にスクリーンキャプチャを示すが、`addinport` で入力ポート、`addoutport` で出力ポートを追加した。図 6 後ろ側のグラフィカル開発環境上で、入力ポート、出力ポートが増えているのが確認できる。この図では、作成した出力ポートに対して `ConsoleOut` というテキスト出力用の RTC を接続してある。また、`nodels` コマンドでノード ID リストを、`nodecat` コマンドで、80h 番には接続がないが、81h 番には `ConsoleOut` コンポーネントが接続されているのを確認できる。

4.1.2 RTM-PDCP 通信実験

RTC と PDCP ノード間の通信を検証する。まず RTC から PDCP への通信を確認するため、PDCP_Wrapper-RTC に追加した入力ポートにテキスト入力用 RTC (`ConsoleIn`) を接続し、そこで入力した値に応じて評価ボードの LED アレイの点灯を確認する。次に逆方向の通信の確認のため、PDCP_Wrapper-RTC に追加した出力ポートにテキスト出力用 RTC (`ConsoleOut`) を接続し、評価ボードのジョイスティックの出力値を表示する。

まず、RTC から PDCP への通信の確認方法を示す。

1. PDCP_Wrapper-RTC と `ConsoleIn`-RTC を起動する。
2. 評価ボードの電源を入れてバインドする。(以下ではノード ID を 02h 番として説明する。)
3. `addinport` で入力ポートを追加し `ConsoleIn` と接続する。(ノード ID 80h 番が割り当てられたとする)
4. `senddata 302404070180` を実行する。(※ノード ID 02h 番のチャンネル 1 番に対し、データソースのノード ID を 80h 番と設定する。)
5. `ConsoleIn` で数値を入力すると、値に応じて LED アレイが点灯するのを確認する。
6. `ConsoleIn` で 0000 と入力すると LED アレイが消灯するのを確認する。

次に、PDCP ノードから RTC への通信の確認方法を示す。

1. PDCP_Wrapper-RTC と `ConsoleOut`-RTC を起動する。
2. 評価ボードの電源を入れてバインドする。(以下ではノード ID を 02h 番として説明する。)
3. `addoutport` で出力ポートを追加し `ConsoleOut` と接続する。(ノード ID 80h 番が割り当てられたとする)
4. `senddata 302404040101` を実行する。(※ノード ID 02h 番のチャンネル 1 番に対し、出力イネーブルを設定する。)
5. ジョイスティックを X 軸方向に動かすと `ConsoleOut` に表示される値が変化するのを確認する。

(注: `addoutport` で追加したノードは、ソースノード ID を設定できないので、接続した RTC 側で受信データのソース ID を確認する必要がある。)

4.2 IO ボード動作検証

ここでは、開発した IO ボードの動作を検証するため、図 7、8 に示す実験システムを構築し、PDCP ノード間の通信によるモータの制御と、RTC と PDCP ノード間の通信によるモータの制御の動作検証について述べる。図 7、8 で、ノート PC に RTM の実行環境を構築し、前述の PDCP_Wrapper-RTC を実行し、USB-CAN インタフェースで PDCP ネットワークと接続する。

4.2.1 PDCP ノード間通信実験

試作した PDCP 対応 IO ボードとワークショップ用評価ボード間の通信により、IO ボードの動作を検証する。IO ボードの PWM 出力には著者が試作した電動義手[12]を接続し、ワークショップ用評価ボードのジョイスティックで、義手の指先の開閉を操作する。

以下の手順で各ボードを設定し、ノード間の通信を確認した。

1. PDCP_Wrapper-RTC を起動する。
2. 評価ボードをバインドする。(以下ではノード ID を 02h 番として説明する。)
3. IO ボードをバインドする。(以下ではノード ID を 05h 番として説明する。)
4. `senddata 305404070102` を実行する。(※ノード 05h のチャンネル 01h に対し、データソースのノード ID を 02h と設定する。)
5. `senddata 302404040101` を実行する。(※ノード 02h のチャンネル 01h に対し、出力イネーブルを設定する。)
6. 評価ボードのジョイスティックを動かす、電動義手の指先が開閉するのを確認する。

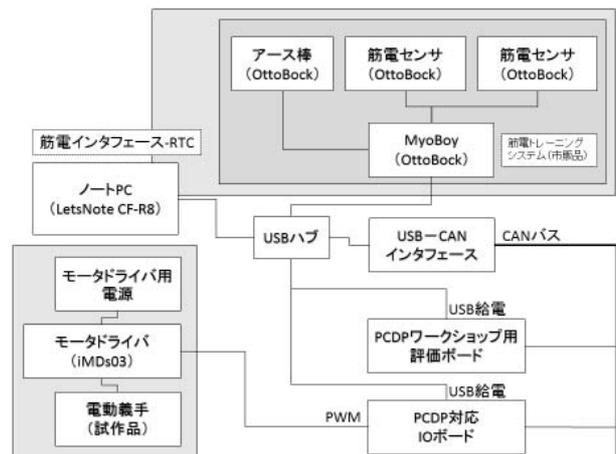


図 7 IO ボード動作検証実験システムのブロックダイアグラム

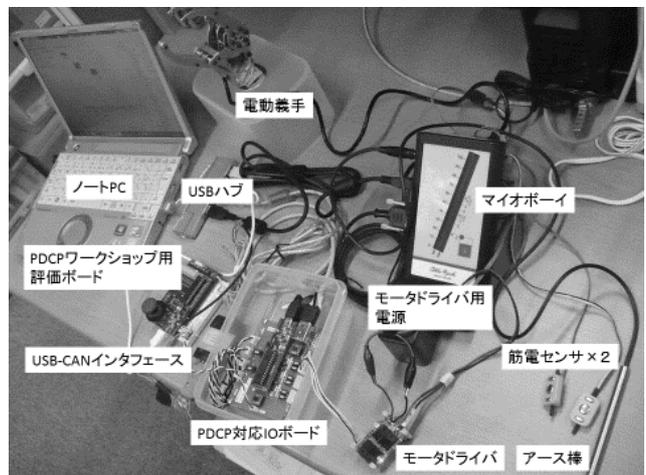


図 8 IO ボード動作検証実験システム

4.2.2 RTC-IO ボード間通信実験

試作した PDCP 対応 IO ボードと RTC の通信による動作検証を行った。まず、RTM 側から IO ボードの PWM 出力の制御を確認するため、RTM 上で筋電インタフェース-RTC と呼ぶ著者が試作したソフトウェアコンポーネントを実行する。[13] 筋電インタフェース-RTC は Otto Bock 社の筋電義手 MYOBOCK®システム用のトレーニングシステム MyoBoy® 757M11 を用いた入力インタフェースである。MyoBoy®は最大2個の筋電センサ (Otto Bock 社製) を接続し、各センサの出力電圧レベルをデバイス上の LED 表示で確認しながら訓練を実施するものである。センサの出力電圧は AD 変換されて USB で PC に入力し、PC 上の専用トレーニングソフトウェアで訓練を行うことも可能である。この RTC を用いることで、測定した筋電センサの出力値を RTM 上で活用できる。ここでは、筋電インタフェース-RTC を PDCP_Wrapper-RTC を用いて仮想的に PDCP ネットワークに参加させ、筋電センサの出力値を用いて、IO ボードに接続する電動義手を操作する。以下の手順にて、この機能の動作を確認した。

1. 筋電センサを前腕の屈筋群と伸筋群付近の皮膚表面に固定し、アース棒をこれらの筋から遠い部位の皮膚表面に接触させる。
2. PDCP_Wrapper-RTC と筋電インタフェース-RTC を起動する。
3. IO ボードをバインドする。(以下ではノード ID を 05h 番として説明する。)
4. addinport で入力ポートを追加し筋電インタフェース-RTC と接続する。(ノード ID 80h 番が割り当てられたとする)
5. senddata 305404070180 を実行する。(※ノード 05h のチャンネル 01h に対し、データソースのノード ID を 80h と設定する。)
6. 筋電センサを着けた手の手首を掌背屈して屈筋群と伸筋群を交互に伸展することで、電動義手の手先が開閉することを確認する。

次に、IO ボードの AD 変換入力を RTC に送信し、その動作を以下の方法で検証した。ここでは、IO ボードの AD 変換入りに、前述の筋電センサの出力を MyoBoy®を介さずに直接接続し、センサ出力を RTC 側で確認する方式を用いた。

1. PDCP_Wrapper-RTC と ConsoleOut-RTC を起動する。
2. IO ボードの電源を入れてバインドする。(以下ではノード ID を 02h 番として説明する。)
3. addoutport で出力ポートを追加し ConsoleOut と接続する。(ノード ID 80h 番が割り当てられたとする)
4. senddata 302404040101 を実行する。(※ノード ID 02h 番のチャンネル 1 番に対し、出力イネーブルを設定する。)
5. 筋電センサを前腕屈筋群付近の皮膚表面に接触させ、筋収縮に伴い、ConsoleOut に表示される値が変化することを確認する。

5. おわりに

包括的な支援環境において電動義肢装具の操作負担の軽減をめざした研究開発を行うため、本稿では、そこで必要となる 2 つの研究開発支援ツールの開発について述べた。

本稿では、想定する義肢装具開発において最も基本的なシステム構成である筋電センサと電動義手だけを用い、開発したツールの動作を検証した。

今後、PDCP を採用した商品の開発が University of New Brunswick などを中心に進められると考えられるため、その動きと連携し、具体的な操作負担軽減に関する研究開発を進めていきたい。

今後の課題として、現状では有線のシステム構成となっており、活動範囲が限定されるため、無線モジュールの採用が必要である。

謝辞

本研究は、厚生労働省平成 21 年度障害者保健福祉推進事業 (障害者自立支援機器等研究開発プロジェクト) 「障害者が自立して住みやすい住環境モデルの構築」の終了後に、著者が所属する研究機関の内部予算で継続的に実施してきたものである。PDCP に関しては University of New Brunswick の Yves Losier 先生と Adam Wilson 先生に指導いただいた。

参考文献

- [1] 谷川民生, "障害者が自立して住みやすい住環境モデルの構築", 障害者自立支援機器等研究開発プロジェクト平成 21 年度総括研究報告書, (2010).
- [2] 澤村誠志, "切断と義肢 第4版", 医師薬出版, (1999).
- [3] 陳隆明, "筋電義手訓練マニュアル", 全日本病院出版会, (2006).
- [4] 中川昭夫, 小嶋功, 長倉裕二, 大藪弘子, 高瀬泉, 町田勝広, 陳隆明, "インテリジェント大腿(IP)義足に適した切断者の訓練と IP 義足の原理の応用(3 部 代替・評価)", バイオメカニズム, No.14, pp.217-226, (1998).
- [5] 谷田惣亮, 菊池武士, 大月喜久子, 安田孝志, 古荘純次, 小澤拓也, 藤川孝満, 森本正治, "コンパクト MR 流体ブレーキを用いたインテリジェント短下肢装具 3 次試作機の開発と足関節弛緩性麻痺患者への適用", 生体医工学, Vol.48, No.1, pp.50-58, (2010).
- [6] A. Kannenberg, B. Zacharias, D. Fish, D. Lindsey, "A Microprocessor Stance And Swing Control Orthosis Improves The Walking Ability Of Traditional Kafo Users", ISPO World Congress Abstracts Book, (2013).
- [7] Nilsson, Anneli et al. "Gait Training Early after Stroke with a New Exoskeleton - the Hybrid Assistive Limb: A Study of Safety and Feasibility." Journal of NeuroEngineering and Rehabilitation 11 (2014).
- [8] TA Kuiken, GA Dumanian, RD Lipschutz, LA Miller, KA Stubblefield, "The use of targeted muscle reinnervation for improved myoelectric prosthesis control in a bilateral shoulder disarticulation amputee", Prosthetics and Orthotics International, Vol.28, No.3, pp. 245-253, (2004)
- [9] 安藤慶昭, "初心者のための R T ミドルウェア入門-OpenRTM-aist-1.0.0 とその使い方-", 日本ロボット学会論文誌, Vol.28, No.5, pp. 550-555, 2010.
- [10] Losier, A. and Wilson, A. Moving Towards an Open Standard: The UNB Prosthetic Device Communication Protocol. The 13th World Congress of the International Society for Prosthetics and Orthotics, Leipzig, Germany, May 2010.
- [11] Kyberd PJ, Poulton AS, Sandsjö L, Jönsson S, Jones B, Gow D. The ToMPAW modular prosthesis: A platform for research in upper-limb prosthetics. J Prosthet Orthot. 2007; 19(1):12-21.
- [12] 梶谷 勇, 樋口 哲也, "筋電義手研究開発プラットフォーム", 第 27 回バイオメカニズム学術講演会(SOBIM2006)予稿集, pp.161-162, (2006).
- [13] 梶谷勇, 三輪昭生, 神徳徹雄, "義肢装具研究開発用 RT コンポーネント", ロボティクス・メカトロニクス講演会講演概要集, (2011).