

目次

1	はじめに	1
1.1	背景	1
1.2	入力装置とロボットの自由度の関係	2
1.3	従来研究	2
1.4	本研究の目的	3
1.5	次章以降について	3
2	アプローチ	4
2.1	概要	4
2.2	ロボットの操作	4
2.3	ロボットへの教示	6
2.4	ロボットへの追加教示	6
2.5	ロボットへの修正教示	7
2.6	まとめ	7
3	実験環境と周辺システム	9
3.1	実験環境	9
3.2	周辺システムの作成	9
3.2.1	ジョイスティック-PC間プログラム	10
3.2.2	クライアントプログラム	11
4	実験プログラム	15
4.1	概要	15
4.2	システム概要	15
4.3	システムの設定	16
4.4	初回教示について	17
4.5	ロボットへの追加教示	18
4.6	ロボットへの修正教示	19
4.7	まとめ	19
5	実験	21
5.1	概要	21
5.2	追加教示実験	21
5.2.1	概要	21
5.2.2	教示1回目	21
5.2.3	教示2回目	23
5.2.4	教示3回目	24
5.2.5	まとめ	26
5.3	修正教示実験	26
5.3.1	概要	26
5.3.2	教示1回目	26
5.3.3	教示2回目,3回目	27
5.3.4	まとめ	28

5.4	まとめ	29
6	おわりに	32
6.1	本研究のまとめ	32
6.2	今後の課題	32
	謝辞	34
	参考文献	35

1 はじめに

1.1 背景

近年，ロボットは目覚ましい進化をとげ，工場内で同じ作業を繰り返すような産業用ロボットなどだけではなく，複雑な環境で，さまざまなタスクを行うロボットが現れ始めている．このようなロボットは今後，例えば家庭内や一般社会などの，より複雑な環境にも浸透していくことが予想される．

しかし，一方で複雑な環境で複雑なタスクを行う場合，それを行うロボットも要求を満たすために多くの自由度やセンサなどを持つ必要がある．これらを制御し，タスクを実行するためには，複雑なプログラムの作成など，多くの労力が必要になる．このような負担を軽減するための試みとして，ロボットに動きの教示を行う方法がある．

これまで，ロボットへの教示方法として，さまざまな研究が行われている[1][2]．

例えば，ロボットの自由度が現在の位置（角度）を取得出来るセンサを持っている場合，これを用いてロボットを直接動かすことで教示を行うという方法がある．この方法は，ロボットを直接見て，ロボットに触れながら教示を行うことが出来る．教示を行う際，多くの情報を得ることが出来るため，教示は容易であると考えられる．しかし，ロボットを遠隔操作する場合や，ロボットが直接教示出来ないほど大きい場合などには用いることが出来ない．また，教示している間は，ロボットに動きを行なわせることが出来ないといった問題がある．

次に，マスター・スレーブシステムによる遠隔操作による教示システムを例にする．マスター・スレーブシステムとは，オペレータの操作するマスターロボットと，マスターロボットからの入力により操作されるスレーブロボットが存在するシステムのことである．このようなシステムを用いてスレーブロボットへの教示を行うことを考える．マスターロボットとスレーブロボットが同じような構造をしている場合，オペレータは遠隔操作でありながら，多くの情報を得ることが出来る．また，マスター側のロボットのみを人の扱いやすい大きさにすることで，直接教示は出来ないような大きなロボットに対しても教示が可能となる．また，スレーブ側の位置（角度）センサを用いなくても，マスター側の位置（角度）センサを用いることでスレーブロボットへの教示が可能であるため，スレーブロボットは教示されている間も動き続けることが出来る．しかし，マスターロボットとスレーブロボットは基本的に1対1であるため，別な種類のスレーブロボットを操作するためには，マスターロボットも変える必要がある．

そこで本研究では，簡易入力装置，あるいは汎用入力装置を用い，ロボットへの教示を行うことにする．簡易入力装置を用いる利点として，上で述べたマスター・スレーブシステムのように入力装置がスレーブロボットにハードウェア的に依存しないことが挙げられる．スレーブロボットが変更になったとしても元々汎用目的である簡易入力装置は，ソフトウェアの変更のみで対応することが出来る．

1.2 入力装置とロボットの自由度の関係

ロボットに対して入力装置を用いて動きを教示することを考える場合、どのような環境でどのようなタスクを実行するかによりロボットと入力装置は変わる。本研究では、入力装置が自由度を持っていることを前提条件として、ロボットの自由度と入力装置の自由度を1対1対応させることを考える。このような方法を取ることで、さまざまなロボットと入力装置に適用することを考えることができる。また、ロボットの全自由度に対して、自由に教示出来ることが確実なシステムとなる。そこで、ロボットと入力装置の自由度の関数に注目し、ロボットの自由度を N 、入力装置の自由度を M とすると、 N と M の関係は、 $N = M$ 、 $N > M$ 、 $N < M$ の3つに分けることができる。

$N = M$ であるシステムを考えると、ロボットの自由度と入力装置の自由度を1対1対応させることで、オペレータはロボットのすべての自由度に対して同時に教示することが出来る。しかし、ロボットの自由度が20を超えるような多自由度ロボットでは、入力装置の自由度も20を超えることになり、操作が複雑になることで、オペレータの負担は増大する。

$N < M$ であるシステムを考えると、 N と M を1対1対応すること考えると入力装置の自由度のうち $N - M$ 分の自由度は不必要である。そこで、これを無効とみなすことでロボットへの教示を実現する。しかし、 $M = N$ の場合と同じように、ロボットの自由度の増大にともなって操作は複雑になってしまう。

$M < N$ であるシステムを考えると、 M と N を1対1対応させることは出来ないことが分かる。本研究ではこの $M < N$ の場合について述べる。

1.3 従来研究

従来研究として、「研磨ロボットのためのジョイスティック支援による力制御教示システム」[3]について述べる。これは、研磨ロボットに対して目標軌道を簡易操作で安全に作成できるジョイスティック支援による力制御教示システムについて研究を行ったものである。6自由度を持つ研磨ロボットに対して3自由度のジョイスティックを用いてロボット操作を実現している。また、力制御教示システムを実現し、ロボットの姿勢制御を行っている。このシステムで操作出来るのはロボットの姿勢のみであり、各自由度に対して自由な教示を行うことは出来ない。

このように、入力装置よりより大きい自由度を持つロボットに対して教示を行うための研究は過去に行われているが、それらのほとんどは一部の自由度のみ操作可能であったり、姿勢や空間に対する入力であり、操作出来ない自由度が存在する。このような自由度はたいていの場合、入力装置の入力により、操作出来るパラメータに追従する用に作成されている。

しかし、工場内で決まった動きを繰り返すことを要求されるロボットの場合は、このようなロボットへの教示方法が有効であるが、複雑な環境でさまざまなタスクを行うロボットの場合、従来の方法のように、単に他のパラメータに追従する自由度や姿勢制御のような自由度を直接操作出来ない教示システムでは不可能な動きの教示が必要になる可能性がある。そのため、全自由度に対して自由に教示を行うことが出来るシステムが必要となる。

1.4 本研究の目的

本研究では，多自由度を有するロボットに対して，入力装置を用いた動きの教示方法を提案する．また，入力装置は自由度を持ち，かつロボットの自由度が入力装置の自由度より大きい場合に限定して，入力の切り替えによって，ロボットの全自由度に対して自由に教示出来るシステムを提案する．

このようなシステムを実現することが出来れば，多くの自由度を持つ複雑なロボットを簡易な入力装置を用いて，全自由度に教示を行えることを示すことが出来る．

1.5 次章以降について

2章では，本研究のアプローチについて述べる．3章では，実験環境と作成した周辺システムについて述べる．それらを用いて実際に作成したシステムを4章で述べる．5章で作成したシステムを用いて実験を行い，6章で全体の考察を行う．

2 アプローチ

2.1 概要

本章ではロボットへの教示システムに関するアプローチについて述べる。

まず、入力装置を用いてロボットを操作することを考える。ここでのロボットの操作とは、入力装置の自由度がロボットの自由度より小さい場合において、入力装置からの入力データを用いて動きのデータを作成し、動作することが出来ることである。次に、ロボットへ教示することを考える。作成した動きのデータを保存し、後で同じ動きを行わせることが出来ることをロボットのへの教示とする。このようなシステムを考えることで、ロボットへの教示システムを実現する。また、このように作成され、保存された入力データに対して自由に追加、修正を加えることが出来る必要があると考えられる。全体の概要は図1のようになる。

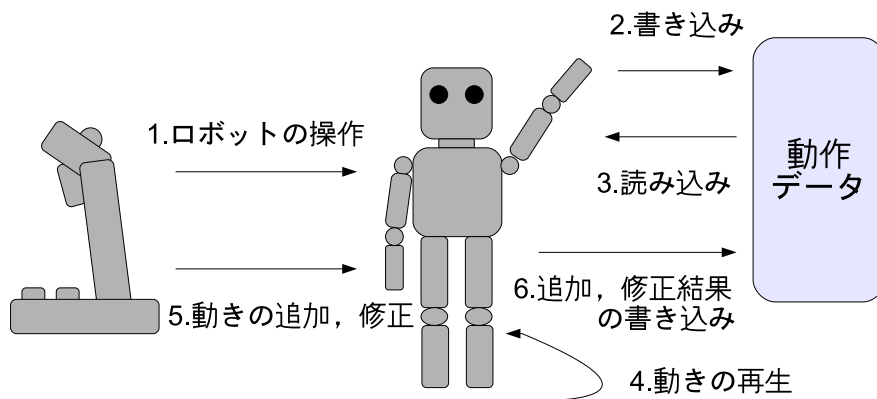


図 1: アプローチ概要

入力装置からの入力によりロボットを操作する部分が「1. ロボットの操作」にあたり、ロボットへの教示部分が「2. 書き込み」、「3. 読み込み」、「4. 動きの再生」である。このように教示したデータに対して追加、修正を行う部分が「5. 動きの追加、修正」と「6. 追加、修正結果の書き込みである」さらに動きの追加、修正を繰り返す場合、「3. 読み込み」から「6. 追加、修正結果の書き込み」を繰り返すことで、要求される動きを作成していく。

そこで、本研究のアプローチを「ロボットの操作」「動きの教示」「動きの追加」「動きの修正」の4つに分けて説明する。2.2節で「ロボットの操作」、2.3節で「動きの教示」、2.4節で「動きの追加」、2.5節で「動きの修正」について述べ、最後にまとめとして、2.6節で全体のアプローチについて述べる。

2.2 ロボットの操作

多自由度ロボットを、それよりも少ない自由度を持つ入力装置を用いて自由に操作することを考える。「自由に操作」を実現するために、本研究ではロボットの1自由度に対して、入力装置の1自由度を対応させる。これにより、ロボットの持つ各自由度を直接操作することが出来る。しかし今回は、入力装置の自

自由度はロボットの自由度よりも少ないような環境を想定しているため、ロボットの自由度すべてにジョイスティックの自由度1つ1つを割り当てることは出来ない。そこで、本研究ではロボットの自由度を仮想的に分割し、これらを切り替えて操作することを提案する。

ロボットの自由度を N 、入力装置の自由度を M とすると、ロボットの自由度が入力装置の自由度より大きい場合、関係式は $N > M$ となる。このままではロボットの自由度に対して入力装置の自由度を1対1で割り当てることが出来ないため、 N を仮想的に分割し、 N_1, N_2, N_3 というものを想定する。この時、分割した自由度に対して入力装置を1対1で割り当てることを考えると、 N_1, N_2, N_3 はそれぞれ M 以下である必要がある。このように分割した自由度のかたまりごとに切り替えて操作する方法を用いることにより、分割した自由度のかたまりごとに入力装置の各自由度を割り当てることが出来る。ただし、この方法では同時に操作出来る自由度は M 以下に制限される。概要図を2に示す。

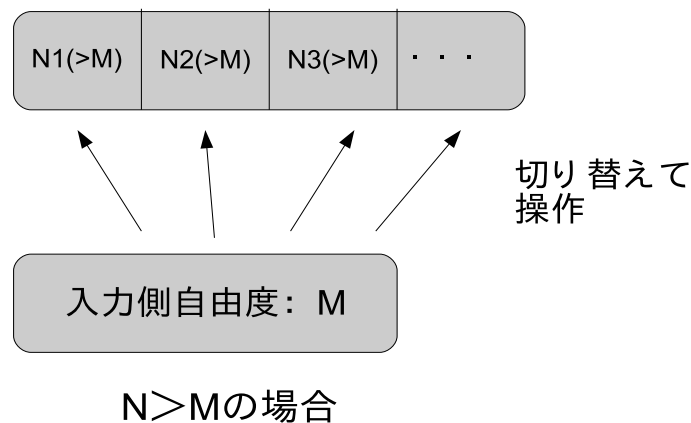


図 2: ロボットの操作手法

例として、ロボットの自由度が4であるアームロボットと、入力装置の自由度が2であるジョイスティックを用いたシステムに対して、今回のロボット操作のアプローチを適用することを考える。ロボットの自由度にはそれぞれ A, B, C, D 、入力装置の自由度には X, Y という名前がついているものとする。ロボットの自由度が N 、入力装置の自由度が M であるので、 $N = 4, M = 2$ となり、関係式は $N > M$ である。そこで、 N を N_1, N_2 に分割する。この時 N_1, N_2 はそれぞれ $N_1 = 2, N_2 = 2$ となる。例えば、 A と B, C と D のかたまりにそれぞれ分割したとすると、この自由度の範囲ごとに入力装置の自由度を割り当てることが出来る。たとえば、 $A-X, B-Y, C-X, D-Y$ のようにそれぞれを割り当てる。この時 A と B のグループと C と D のグループを切り替えることで、同じ入力装置の同じ自由度を用いたとしても、別な自由度を操作することが可能である。しかし、 A と C を同時に操作することは出来ないため、順次操作するグループを切り替える必要がある。あるいは $A-X, B-X, C-Y, D-Y$ のように対応させ、 A, C グループと B, D グループに分けるといった工夫が必要となる。

2.3 ロボットへの教示

入力装置を用いてロボットに動きの教示を行うことを考える．入力装置からの操作データから実際のロボットの動きを作成し，これを動作データとして記録しておく．これを必要な時に，自由に呼び出ることが出来るようなシステムとする．また，入力装置によりロボットへの教示が行われている間，ロボットはその教示された動きにしたがって動作している．これにより，ロボットに教示している動きを視覚的にオペレータへ伝えることが出来る．概要図を図3に示す．

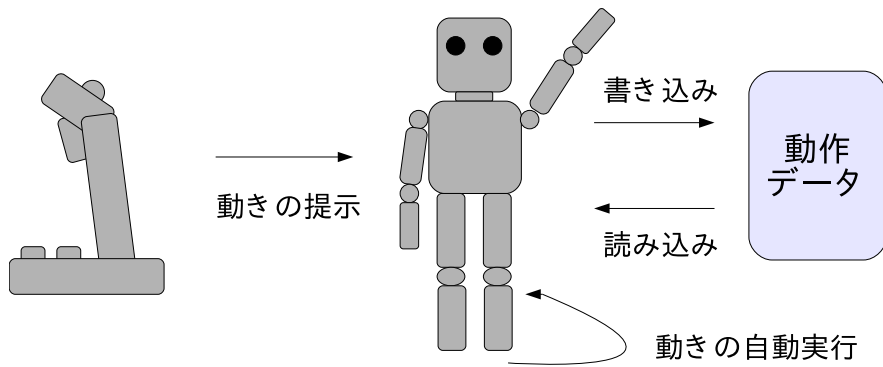


図 3: 動きの教示概要

2.4 ロボットへの追加教示

ロボットが教示した動きを実行している間，教示していない自由度に対して，追加で教示を行う．ロボットは前回の教示データに，入力装置による操作を加えた動きを動作データとして記録する．これを必要な時に呼び出し，実行することが出来る．また，このように追加教示された動作データに対して，さらに追加教示を行うことが出来る．これを繰り返すことで，すべての自由度に教示を行うことも可能である．追加教示されている間，ロボットは過去の動作データにリアルタイムで追加教示されている入力データを加えた動きを実行する．これにより，ロボットに教示している動きを視覚的にオペレータへ伝えることが出来る．概要図を図4に示す．

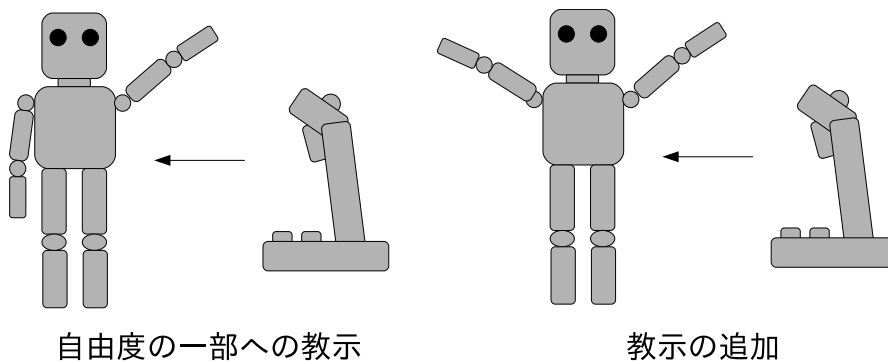


図 4: 動きの追加教示概要

例として、ロボットの自由度が6であるアームロボットと、入力装置の自由度が2であるジョイスティックを用いたシステムに対して、今回の追加教示のアプローチを適用することを考える。ロボットの自由度にはそれぞれA, B, C, D, E, F, 入力装置の自由度にはX, Yという名前がついているものとする。まず、ロボットの自由度をAとB, CとD, EとFグループに分割し、AとBに対して教示を行う。この時に保存された動作データを実行中に、CとDに教示を行う。さらにCとDへの教示後の動作データを実行中にEとFに対して教示を行うことが出来る。

2.5 ロボットへの修正教示

ロボットが教示した動きを実行している間、すでに教示され、動いている自由度に対して、入力装置を用いてその動きに修正を加える。ロボットは前回の教示データに、入力装置による修正教示データを合わせた値を新たな教示データとして記録する。これを必要な時に呼び出し、実行することが出来る。また、このように修正教示された動作データに対して、さらに修正教示を行うことが出来る。これを繰り返すことで、よりよい動き、あるいは状況にあった動きを生成していく。概要図を図5に示す。

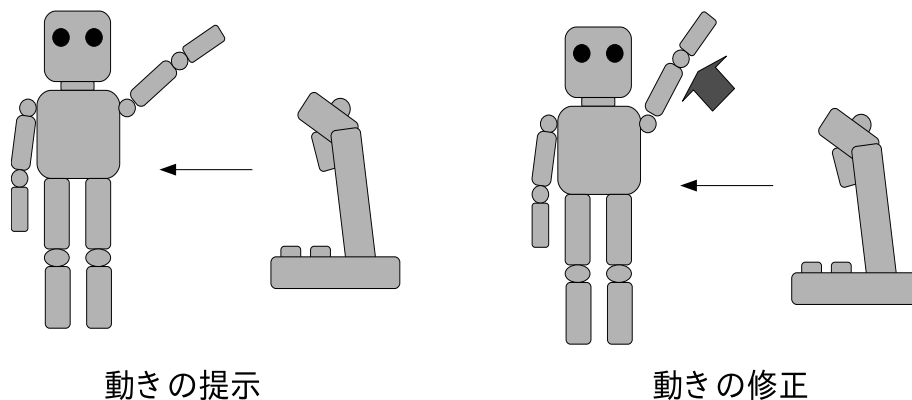


図 5: 動きの修正教示概要

例として、ロボットの自由度が2であるアームロボットと、入力装置の自由度が2であるジョイスティックを用いたシステムに対して、今回の追加教示のアプローチを適用することを考える。ロボットの自由度にはそれぞれA, B 入力装置の自由度にはX, Yという名前がついているものとする。まず、ロボットの自由度A, Bに対してX, Yを用いて教示を行い、動作データを記録する。この動作データを実行中にさらにA, Bに対して修正教示を行う。またこの時、A, Bへの教示は何度でも行うことが出来る。

2.6 まとめ

2.1節では、どのようなアプローチによりロボットに対して教示を行うか概要を述べた。その詳細なアプローチとなるのが2.2節から2.5節となる。今

回のアプローチでは，ロボット操作による結果が教示データとして記録される形になるので，本研究ではまず，2.2節でロボットに対して教示を行うために必要なロボット操作について述べた．次にロボット操作のアプローチを用いて，どのようなアプローチにより教示を行うかを2.3節で述べ，さらに応用として，教示結果に対して追加，修正教示を行う方法を2.4節,2.5節で述べた．実際のシステムでは，2.3節のロボットへの教示により1度目の教示を行い，その後2.4節,2.5節の追加，修正教示を用いてロボットの全体の動きを作成していく．

3 実験環境と周辺システム

3.1 実験環境

本研究に用いた実験環境を説明する．ロボットには，SPC-001 を用いた．これは，Speecys 社により販売されている人型ロボットの 1 つであり，多数のサーボやセンサーを備える．主な仕様を表 1 に，外観を図 6 に示す．

また OS として，NetBSD をベースに開発された SpeecysOS を用いており，TCP/IP，UDP/IP 通信プロトコルをサポートし，このため，ネットワークを利用したシステムが作成可能である．SpeecysOS の元となる OS である NetBSD について述べる．NetBSD は，4.4BSD-Lite をもとに開発された UNIX 互換の OS であり，サポートしているハードウェアの数が非常に多いのが特徴である．また，優れたセキュリティーや高い安定性を備えているため，学術機関などを中心に導入が進んでいる．また，フリーウェアとしてソースコードと共に無償で公開されている．

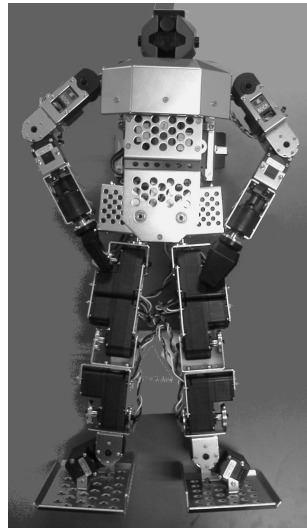


図 6: ロボットの外観

本研究ではサーボをおもに用いた実験を行うため，サーボについてはより詳しい仕様を表 2 にまとめ，外観を図 7 に示す．また，プロセッサユニットの詳細な仕様を表 3 にまとめ，外観を図 8 に示す．

入力装置として EXTREME 3D PRO を用いた．これは Logitech 社から販売されているジョイスティックである．主な仕様を表 4 に，外観を図 9 に示す．

その他，Windows がインストールされた PC，およびデータ送信のための無線ルータを用いる．

3.2 周辺システムの作成

ジョイスティックによるロボットの遠隔操作を実現するために，まずジョイスティックからの入力を PC で受け取り，その後ロボットへデータを送信するクライアントプログラムとそのデータを受信するサーバプログラムを作成した．

表 1: ロボットの主な仕様

名前	SPC-001
関節可動部自由度	23 自由度
LED (眼部分)	3色*2 セット
35 万画素 CMOS カラーカメラ	1 個
マイク入力	2 個
音声出力用スピーカー	1 個
サーボ	21 個
ジャイロセンサー	3 軸
3 軸 G センサー	1 セット
寸法 / 重量	約 50cm/3.7kg
無線 LAN	IEEE802.11b 準拠



図 7: サーボの外観

システムの概要を図 10 に示す。ジョイスティック-PC 間は USB 接続による有線接続とし、PC-ロボット間は TCP/IP プロトコルを用いた無線接続とする。

周辺システムとして、ロボット操作、教示以外のシステムであるジョイスティック-ロボット間の通信プログラムと、クライアントプログラムについて述べる。

3.2.1 ジョイスティック-PC 間プログラム

まず、ジョイスティック-PC間のプログラムについて述べる。今回用いた PC は Windows OS であるため、ジョイスティックからの入力値の収集には DirectInput を用いた。DirectInput とは、Microsoft 社が提供している DirectX ライブラリの一部であり、マウス、キーボード、ジョイスティック等のユーザインタフェースからの入力値を収集することが出来る API である。これを用い、ジョイスティックからの入力データを収集する。

次に PC-ロボット間の通信プログラムについて述べる。PC についてはジョイスティック-PC 間の通信に用いたものであるため、OS は Windows である。ロボットについては NetBSD に改良を加えたものであり、クライアントプログラ

表 2: サーボの主な仕様

重量	93g
動作速度	0.17sec/60度 (9.6V)
可動範囲	240度
出力トルク	21kg・cm(9.6V)
電源電圧	9.6V
制御方式	RS485
通信速度	最大 1.3Mbps
通信方式	双方向, コマンド式, 半 2 重
センス情報	角度, トルク, 温度など
キャリブレーション	25 点補正
耐久性	ラジコンサーボの 10 倍程度



図 8: プロセッサユニットの外観

ラム, サーバプログラムともに TCP/IP 通信のための関数を用いたプログラミングが可能である。今回は, ロボットをサーバと見なして PC 側からの接続を待ち受けるようにする。PC 側からの接続があった場合, これを許可し, ジョイスティックの入力データを受け取る。

3.2.2 クライアントプログラム

クライアントプログラムは, 先に述べた DirectInput によりジョイスティックからの入力データの受け取るプログラムと, そのデータをロボットに送信するための TCP/IP プログラムを組み合わせた GUI システムである。クライアントプログラムの外観を図 11 に示す。

上から 1 段目の「X=-18880」「Y=5782」「Z=-7424」はそれぞれ, ジョイスティックからの X 軸方向の直接的なアナログ入力データ, Y 軸方向の直接的なアナログ入力データ, Z 軸方向の直接的なアナログ入力データを表す。この入力データは, ニュートラルの位置を 0 とした時, 下限-32767 から上限 32768 程度の値をとるが, 後述する「ジョイスティックオフセット修正」により実行するたびに値が多少増減する。上から 2 段目の「2」という数字は, ジョイスティッ

表 3: プロセッサユニットの主な仕様

CPU	MPC5200(264 ~ 400MHz)
ROM	FLASH MEMORY 16MB
RAM	DDR SDRAM 128MB
OS	NetBSD
FPGA	各種タイミング生成その他
CF Slot	メモリーカード CF × 1
CF Slot	無線 LAN(802.11b) 用 CF スロット × 1
Serial I/F	RS232C × 1ch
Audio I/F	AC97 CODEC 対応
Audio コネクタ	スピーカー × 2 マイク × 2
Camera	NTSC
表示用 LED	単色
RS485 Serial I/F	モータ・センサ類の I/F 用 × 7ch
外部用 5V 電源	5V × 2ch
WM 出力	PWM × 2ch(頭部用)
サーバ機能	WWW・FTP・TELNET サーバ
外部寸法	W91mm × D55mm
電源仕様	DC9.6V ニッケル水素バッテリー

表 4: 入力装置の主な仕様

名前	EXTREME 3D PRO
アナログ入力	3 自由度
ボタン入力	12 個
8 方向ハットスイッチ	1 個
スロットル	1 個
接続形式	USB

ク付属の「ボタン 2」が押されている状態を示す。ボタンは 0 番から 11 番まであり、複数個同時に押された場合は押された番号すべてが画面に表示され、ロボットに送信される。上から 3 段目の「X=-73」「Y=22」「Z=-29」については、1 段目の「直接的なアナログ入力データ」に補正を加えて、分かりやすく、扱いやすくした値である。値の下限は-100で上限は 100 とした。補正については以下の式を用いた。

$$X = X_0/256, Y = Y_0/256, Z = Z_0/256$$

X,Y,Zはそれぞれ補正後の X 軸, Y 軸, Z 軸の値であり, X_0, Y_0, Z_0 はそれぞれ補正前の X 軸, Y 軸, Z 軸の値である。補正後の各軸の値 X,Y,Z が下限 (-100) を下回る場合, その軸の値は-100 とする。逆に X,Y,Z が上限 (100) を上回る場合, その軸の値は 100 とする。ロボットに送信されるデータは 2 段目の「ボタン」データと三段目の「補正後のアナログ入力データ」である。



図 9: 入力装置の外観

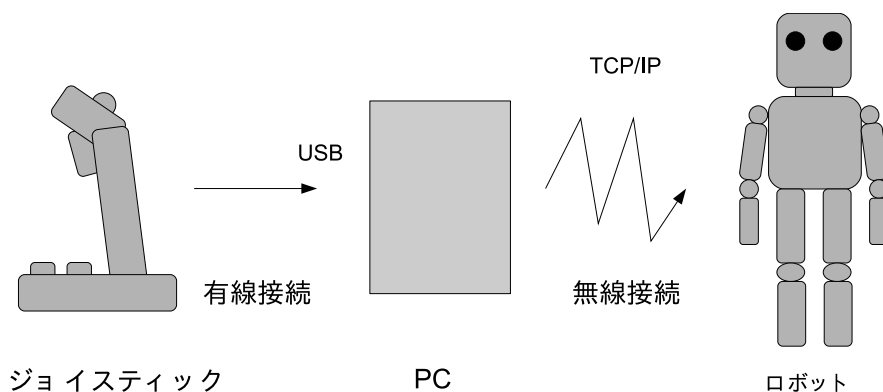


図 10: 周辺システム概要

次にプログラムのメニューについて説明する。「ファイル」をクリックすると「終了」がポップアップする。これをクリックすることでプログラムは終了する。「接続」をクリックすると、「開始」と「切断」がポップアップする。「開始」はロボットとの接続を確立し、通信を開始する。「切断」はロボットとの通信を切断するために用いる。ロボットとの接続を開始するためには、ロボット側でサーバプログラムを実行し、待ち受け状態にしておく必要がある。「モード」をクリックすると、「教示」と「修正」がポップアップする。初期状態（何も教示していない状態）から新しく教示を行う場合、「教示」をクリックする。この場合、それまでに教示した動きはリセットされる。教示後の動きに対して追加で教示を行う場合や、修正を行う場合は「修正」をクリックする。モードは接続を開始する前に決定しておく必要がある。「その他」をクリックすると、「ジョイスティックオフセット修正」がポップアップする。これはジョイスティックのニュートラル状態の微妙なずれを修正するものであり、ロボットへ教示する前に行うことで、ニュートラルの位置を設定することが出来る。

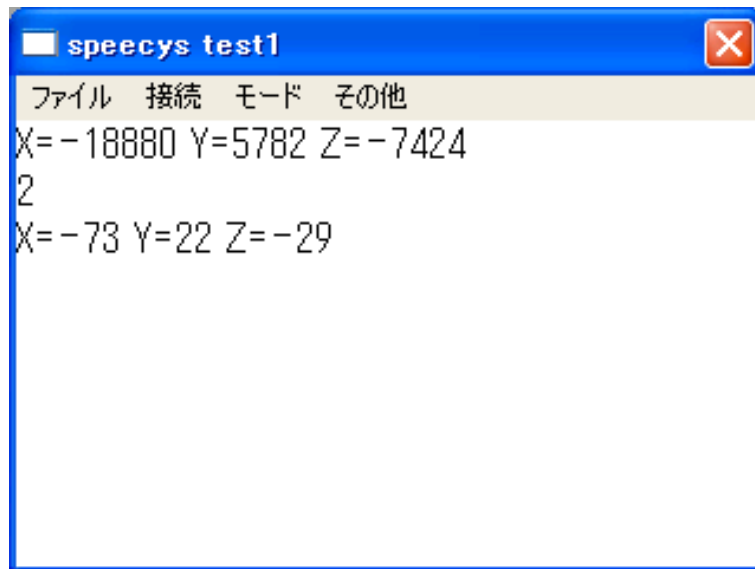


図 11: クライアントシステムの外観

クライアントプログラム使用の流れは、プログラム実行 「ジョイスティックオフセット修正」 「モードの設定」 「ロボットとの接続と教示の開始」 「教示の終了と切断」 「終了」 プログラム終了となる。

4 実験プログラム

4.1 概要

本章では、実際に作成したシステムとその詳細な内容について述べる。入力装置を用いてロボットに教示するにあたって、入力装置には3自由度を持つジョイスティックを、ロボットには23自由度を持つ多自由度ロボット SPC-001 を用いた。

ロボットへの教示の手順としては、まず1回目の教示として、ロボットの動作データを作成し、保存するための初回教示を行い、その後、この動作データに対して追加教示、修正教示を行う流れとなる。また、教示中ロボットは、常に教示された結果の動きを行っている。これにより、オペレータに視覚的にロボットの動きを理解出来るようにした。

まず、4.2節で、システムの概要について述べ、4.3節でシステム全体に共通する設定について述べる。次に4.4節でロボットへの初回教示に関するシステムについて述べ、4.5節ではロボットへの追加教示について、4.6ではロボットへの修正教示についてそれぞれ述べる。最後に4.7節で全体のまとめを行う。

4.2 システム概要

入力装置を用いてロボットへ動きを教示する方法について述べる。入力装置からの入力データを用いてロボットの動きを作成、または追加、修正し、これを動作データとして保存することが出来るシステムである。ロボットの1自由度に対して、入力装置の1自由度を対応させる。今回はロボットに SPC-001 を想定した。また、入力装置として、3自由度を持つジョイスティックを想定した。

ロボットの自由度の数を N 、入力装置の自由度の数を M とすると、 $N > M$ の関係であるため、 N を分割し N_1, N_2, N_3 とする。この時、 $N_1, N_2, N_3 \dots$ はそれぞれ M 以下の値である必要がある。このように分割したあとの各自由度のグループを切り替える方法として、ジョイスティックに付属するボタンを用いることでこれを実現する。これについて説明するためにまず、分割後の各自由度のグループに対して、グループ1、グループ2、…グループ K と名前をつける。グループ1に対してボタン1を割り当て、グループ2に対してボタン2を割り当てる…グループ K に対してボタン K を割り当てる、ということを行い、すべてのグループに対してボタンを1つずつ割り当てるものとする。何も押していない状態ではジョイスティックのアナログ入力(自由度)からの入力を受け付けず、各自由度への教示は行われず、またロボットは動かない。ボタン1を押した状態でアナログ入力を行うと、グループ1に属する自由度にジョイスティックのアナログ入力(自由度)が割り当てられ、教示可能となる。同じようにボタン2を押した状態でアナログ入力を行うと、グループ2に属する自由度にジョイスティックのアナログ入力(自由度)が割り当てられ、ボタン K を押した状態でアナログ入力を行うと、グループ K に属する自由度にジョイスティックのアナログ入力(自由度)が割り当てられる。

今回は動作データの保存として角度を用いた。動作データの保存に角度を用いた理由として、ロボットのサーボは角度によって制御されていることが挙げ

られる．動作データを実行する際，直接このデータを用いることができるため，データの管理がしやすく，変更が容易である．

4.3 システムの設定

本研究では，ロボットについて，SPC-001の自由度のうち，6自由度を用いてシステムを作成した．また，ジョイスティックについては3自由度を用いた．これにより，ロボットの自由度 $N = 6$ ，入力装置の自由度 $M = 3$ となる．ロボットについては，SPC-001の持つすべての自由度を用いたわけではないが， $N > M$ の関係がなりたっているため，提案したアプローチの実装としては，ロボットの自由度を限定しても問題ないを考える．この時，ロボットの自由度を3自由度ずつに分け， $N_1 = 3, N_2 = 3$ とし，それぞれのグループにグループ1，グループ2と名前をつけると，グループを切り替える際に必要なボタンの数は2個となる．そこでジョイスティック付属のボタンを2つ用意し，これをボタン1，ボタン2とする．SPC-001は23の自由度を持つが，今回は6自由のみを扱うため，残りの17自由度については無効とする．想定したシステムの設定の概要を図12に示す．

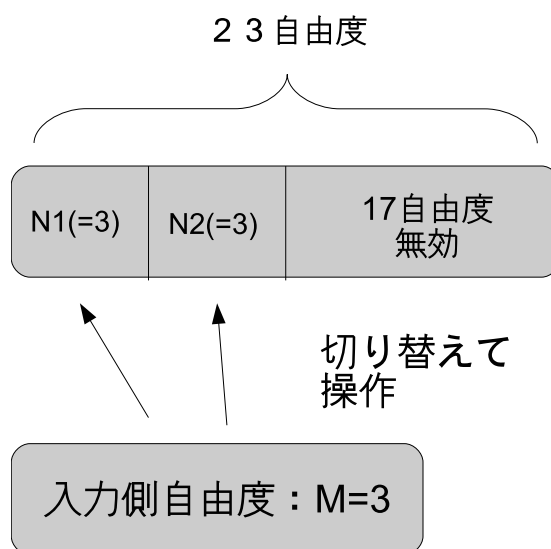


図 12: ロボット教示のシステム設定

また，データの保存形式は表5のようにテキスト形式で保存される．

表 5: 動作データの保存方法

servo1 0ms	servo2 0ms	...	servo21 0ms
servo1 500ms	servo2 500ms	...	servo21 500ms
servo1 1000ms	servo2 1000ms	...	servo21 1000ms
...

今回は，500msごとにジョイスティックのデータを取得しているため，動作データの保存も500msおきのデータとなる．また，動作データ終了の地点は，ど

のくらいの時間ロボットに対して教示を行ったかにより変わる．例えば，1500msの間教示されると，動作データも1500msまで保存される．同じように5200msの間教示されると，動作データは5000msまで保存されることになる．

4.4 初回教示について

教示データの無い状態において，新たに教示データを作成し，教示結果を残すことが出来る．また，この教示データはあとで実行することが出来る．

今回，ロボットの各自由度を入力装置を用いて教示する方法として，速度制御を用いた．ただし，速度制御は初回教示のみである．

速度制御の場合，ジョイスティックからのアナログ入力の数値は速度として扱われる．しかし，ロボットは角度により制御されているため，以下の式により，変換を行った．

$$\theta_{i+1} = \theta_i + (v * t)$$

θ_i は i 回めのサーボへの出力であり，角度出力となる． v は i 回めのジョイスティックからの入力である． t は v を角度に変換するための係数であり定数となる．今回の場合，ロボットへの動作命令を500msごとに行っているため， $t = 0.5$ とする．つまり，入力変数であるジョイスティックからの入力 v に定数 t をかけ，現在のサーボの角度を足し合わせたものが新たなサーボ角度となり各サーボへの出力となる．この時， θ_i が保存される動作データである．初回教示の例を図13に示す．

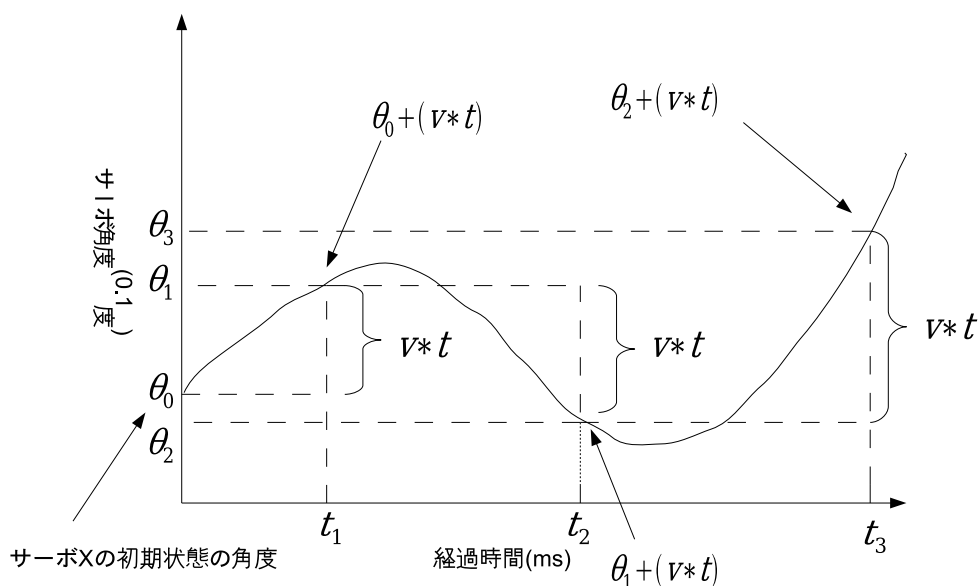


図 13: 初回教示の例

4.5 ロボットへの追加教示

すでにロボットの一部自由度へ教示を行った動作データを実行している間に、まだ教示されていない自由度に対して、入力装置を用いて動きを追加することが出来る。実行されている動作データに対して、値を修正することで、動きの追加を行う。動き追加を行うための式は以下で表される。

$$\theta_i = F_i + \theta'_i$$

この式を現在教示しているサーボに対して適用する。ここで、 θ_i は i 回目の出力（ロボットへの出力と追加教示の結果の保存）である。 F_i は再生中の動作データの i 個目である。例えば、サーボ 2 に対して教示することを考えると、元の動作データのサーボ 2 の i 番目である F_i に θ'_i を加えたものを新たな教示データとする形となる。ロボットへの追加教示においては、新たに教示を行うサーボについては、まだ動きを教示していないということより、 F_i はほぼ一定であると考えられる。そのため、新たに追加教示を行った部分のロボットの動きは、ほとんど θ'_i によって決まることになる。 θ'_i は以下の式で表される。

$$\theta'_i = \sum_{k=0}^{i-1} \theta''_k + (v_i * t)$$

$\sum_{k=0}^{i-1} \theta''_k$ は前回までのジョイスティックからの入力データを角度に変換したものである。 v_i は今回の入力であり、ジョイスティックを用いて、速度として入力を行う。 t は v を角度に変換するための係数であり、定数である。今回の場合、ロボットへの動作命令を 500ms ごとに行っているため、 $t=0.5$ とする。このような θ'_i を「累積入力データ」とする。追加教示の例を図 14 に示す。

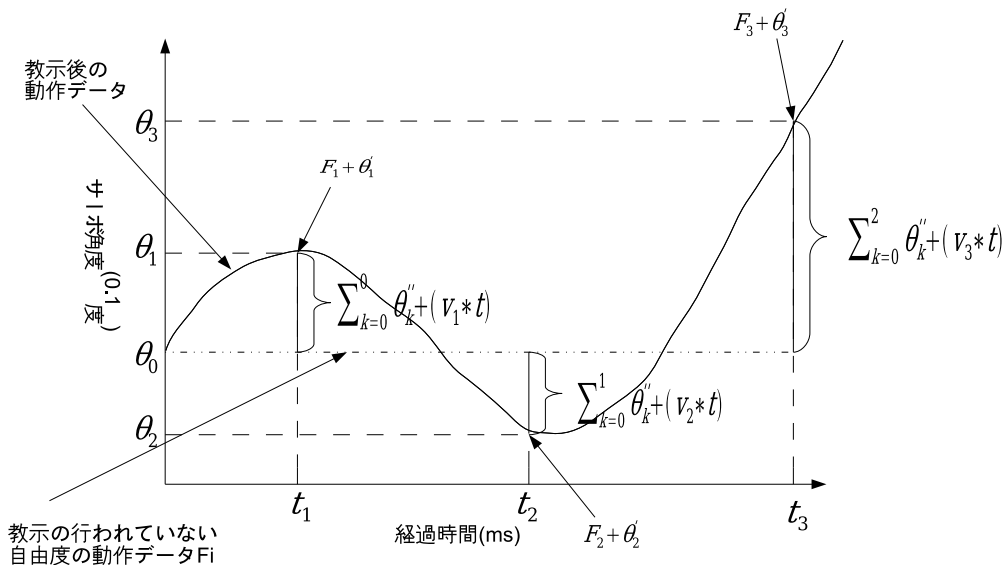


図 14: 追加教示の例

以上の式を用いて表 5 のように保存されている動作データに対して修正を行い、動きを追加する。この際、もし過去の動作データよりも早く、教示を終了した場合、それ以降の時間の教示データは削除される。また逆に過去の動作データよりも長く教示を行った場合、過去の動作データには F_i が存在しないため、存在する一番最後の F_i を用いて動きの追加を続行することが出来る。

4.6 ロボットへの修正教示

すでにロボットの自由度に対して教示を行った動作データを実行している間に、過去に教示されている自由度に対して入力装置を用いて動きを修正することが出来る。実行されている動作データに対して、値を修正することで、動きの修正を行う。動き修正を行うための式は以下で表される。

$$\theta_i = F_i + \theta'_i$$

この式を現在教示しているサーボに対して適用する。ここで、 θ_i は i 回目の出力（ロボットへの出力と追加教示の結果の保存）である。 F_i は再生中の動作データの i 個目である。例えば、サーボ 2 に対して教示することを考えると、元の動作データのサーボ 2 の i 番目である F_i に θ'_i を加えたものを新たな教示データとする形となる。 θ'_i は以下の式表される。

$$\theta'_i = \sum_{k=0}^{i-1} \theta''_k + (v_i * t)$$

$\sum_{k=0}^{i-1} \theta''_k$ は前回までのジョイスティックからの入力データを角度に変換したものである。 v_i は今回の入力であり、ジョイスティックを用いて、速度として入力を行う。 t は v を角度に変換するための係数であり、定数である。今回の場合、ロボットへの動作命令を 500ms ごとに行っているため $t=0.5$ とする。このような θ'_i を「累積入力データ」とする。修正教示の例を図 15 に示す。

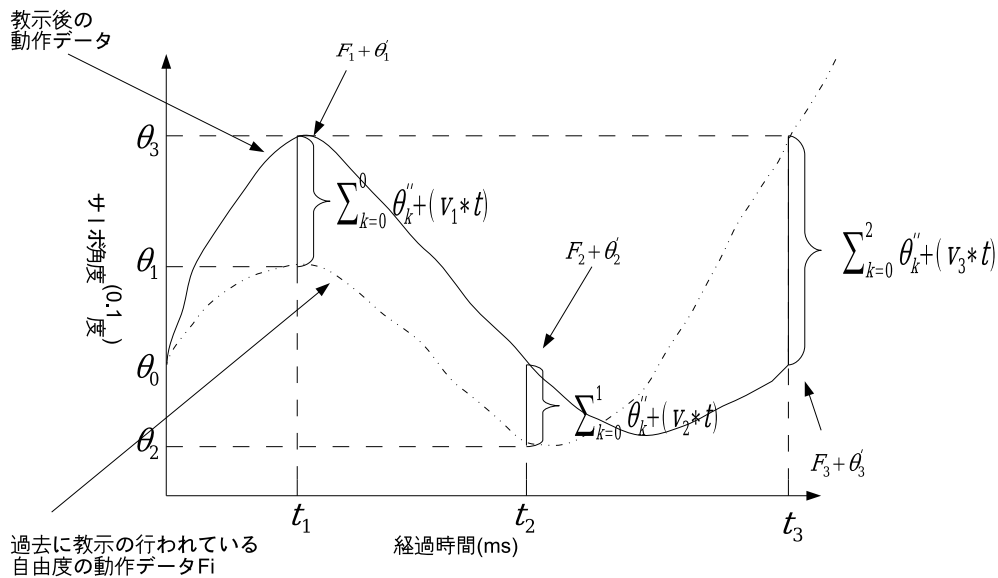


図 15: 修正教示の例

以上の式を用いて表 5 のように保存されている動作データに対して修正を行い、動きを修正する。

4.7 まとめ

初回教示の場合、

$$\theta_{i+1} = \theta_i + (v * t)$$

であり、 θ_{i+i} が出力となるので、 v の入力を行った時点から次の時間に記録されることが分かる。初回の教示ならば、これは問題にならないが、追加教示時、修正教示時の入力では、過去の動作データとの連動が求められるので、速度制御のように時間差を考えなくてもよい累積入力データ制御を用いた。

本研究では追加教示と修正教示では同じ関数を用いて教示を行っているが、静止している自由度に対する制御と、動いている自由度に対する制御では、操作感覚が大きく異なる。静止している自由度に対する制御では動作データ F_i を定数と見ることが出来、唯一の入力変数が入力 v のみと考えることができる。しかし、動いている自由度に対する制御では、動作データ F_i も入力変数となる。このため、修正教示の場合は F_i を考慮しながら教示を行う必要があり、追加教示と比べて難解となる。このため、本章では追加教示と修正教示を明確に分けて述べた。

5 実験

5.1 概要

本章では，作成したシステムを用いて「追加教示」「修正教示」2つの実験を行い，考察する．5.2節で追加教示実験について，5.3節で修正教示実験について述べ，5.4節で本章全体の考察を行う．また「追加教示実験」「修正教示実験」とも，最初の教示については，速度制御による初回教示となる．

5.2 追加教示実験

5.2.1 概要

本実験では，両腕の肩サーボ4個と肘サーボ2個を用いる．肩サーボはロボット正面に対して，前後方向に動く1サーボと左右方向に動く1サーボについているため，片腕に2サーボとなる．これらの6つのサーボにはそれぞれ順番があり，右肩の前後方向に動くサーボが14番，右肩の左右方向に動くサーボを15番，右肘のサーボが16番，左肩の前後方向に動くサーボが18番，左肩の左右方向に動くサーボが19番，左肘のサーボが20番となる．これらの概要図を図16に示す．

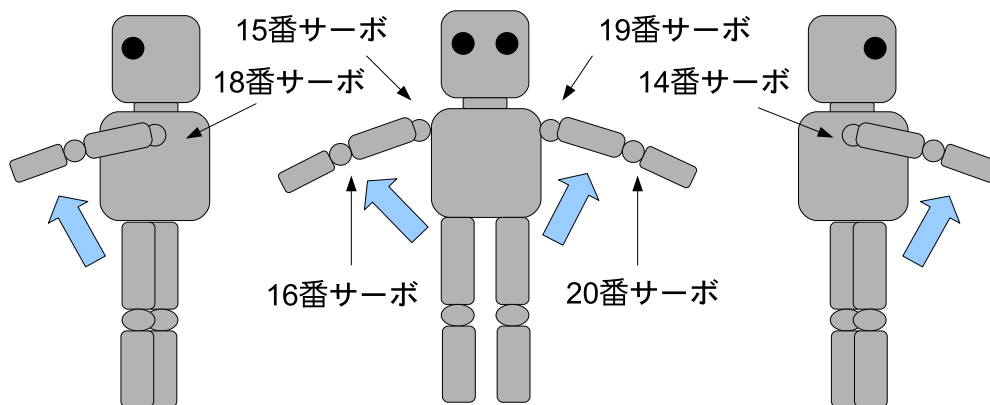


図 16: 各サーボの位置と番号

ロボットに対して追加教示実験を行う．まず，1度めの試行で右肩の14番サーボと15番サーボに教示を行い，2度めの試行で左肩の18番サーボと19番サーボに教示を行う．さらに3度めの試行で16番サーボと20番サーボに教示を行う．初期位置は両肩サーボとも体に沿うように下に下げた状態であり地面と垂直である．両肘サーボとも曲がっていない状態であり，体に沿うように地面と垂直になっている．

5.2.2 教示1回目

1度目の教示について述べる．まず，14番サーボを用いて腕を前方向に地面と水平になるまで上げる．その後初期位置まで下げ，今度は15番サーボを用

いて右方向に腕を地面と水平になるまで上げ、初期位置まで下げる。1度目の教示の概要図を図17に示す。

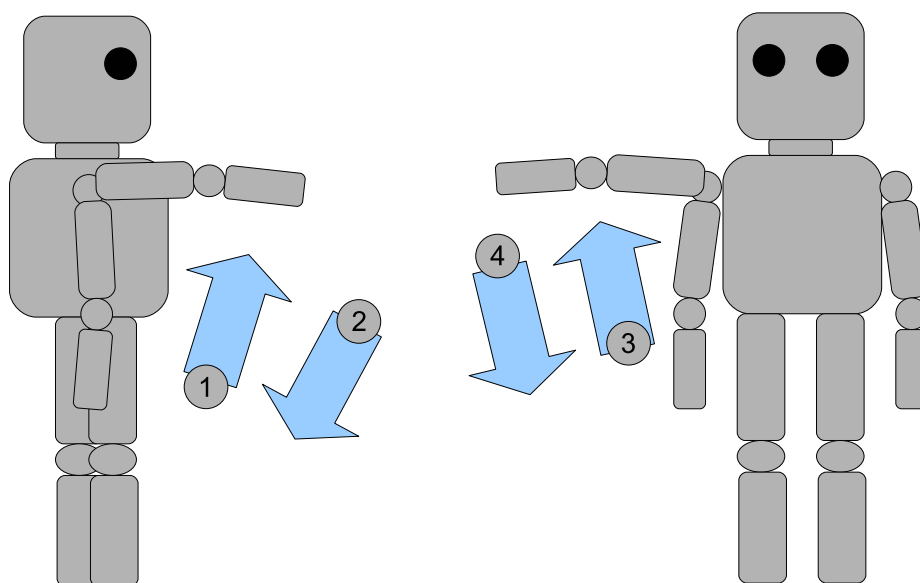


図 17: 追加教示実験 1 回目概要

この時 14 番サーボについては、初期位置の角度は 900(0.1 度) 程度であり、前方向に腕を上げると、角度は減少し、水平付近で 0(0.1 度) 程度となる。15 番サーボについては、初期位置の角度は 900(0.1 度) 程度であり、右方向に腕を上げると、角度は減少し、水平付近で 0(0.1 度) 程度となる。教示された結果を図 18 に示す。

1000(ms) 付近から 14 番サーボが上がり始め、4000(ms) 付近で上がりきり、8000(ms) 付近で初期位置まで戻っているのが分かる。また 8000(ms) 付近から 15 番サーボが上がり始め、11000(ms) 付近で上がりきり、15000(ms) 付近で初期位置まで戻っているのが分かる。

14 番サーボについて、初期位置が 1000(0.1 度) を超えてしまっているのは、14 番サーボが若干、地面と垂直方向より、ロボットの背中方向に傾いていたためと思われる。腕を水平位置まで上げた際、0(0.1 度) を下回ってしまっているのは、腕が水平方向まで上がっていることを目測で確認したため、水平よりも高い位置まで上げてしまったと考えられる。また、前述 2 つの問題に関して、ロボットを組み立てる際、組み立て方によって若干傾いてしまい、それにより位置のずれが生じてしまうことがある。経過時間 8000(ms) 付近で 0(0.1 度) 付近まで値が大きく下がっているが、これはサーボの角度の値を取得する際、値の取得ミスによる誤りデータであり、実際にはこの値の位置まで腕が上がることはなく、初期位置まで下がり続けている。

15 番サーボについて、腕を水平位置まで上げた際、0(0.1 度) を下回ってしまっているのは、14 番サーボと同じように、腕が水平方向まで上がっていることを目測で確認したため、水平よりも高い位置まで上げてしまったと考えられる。

16 番サーボについて、若干値に変動があるのは 14 番、15 番サーボを操作している際は、ボタン 1 を押すことによる操作となるので、ジョイスティックの

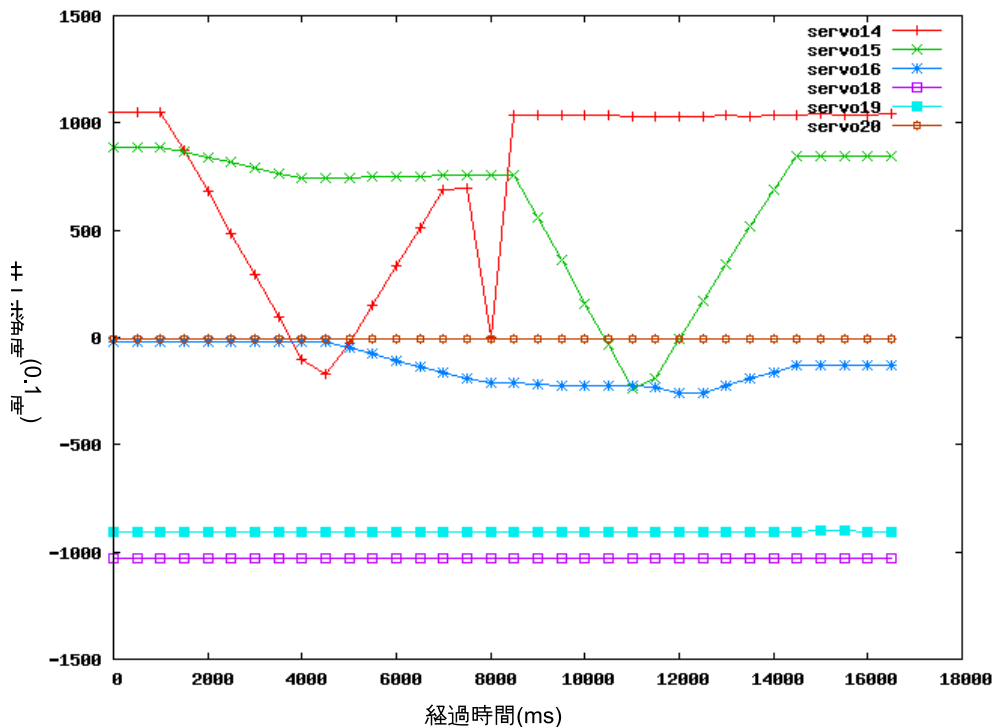


図 18: 追加教示実験 1 回目結果

各軸に 14 番, 15 番, 16 番サーボが割り当てられている．このことから, 操作していない自由度も多少ぶれてしまうためである．

5.2.3 教示 2 回目

2 度目の教示について述べる．1 度目に教示された動作データを実行中に 18 番, 19 番サーボに対して教示を行う．まず, 1 度目の教示により 14 番サーボが地面と水平方向になるまで上がっていく動作に合わせて, 左肩サーボである 18 番サーボを用いて, 腕を前方向に地面と水平になるまで上げる．その後 14 番サーボに合わせて初期位置まで下げ, 今度は 19 番サーボを用いて左方向に腕を地面と水平になるまで上げ, 初期位置まで下げる．この時も 15 番サーボの動きに合わせて水平方向まで上げ, 初期位置まで下げる．2 度目の教示の概要図を図 19 に示す．

この時 18 番サーボについては, 初期位置の角度は $-900(0.1 \text{ 度})$ 程度であり, 前方向に腕を上げると, 角度は増加し, 水平付近で $0(0.1 \text{ 度})$ 程度となる．19 番サーボについては, 初期位置の角度は $-900(0.1 \text{ 度})$ 程度であり, 右方向に腕を上げると, 角度は増加し, 水平付近で $0(0.1 \text{ 度})$ 程度となる．

教示された結果を図 20 に示す．

14 番, 15 番サーボについては, 1 度目の教示のままなのでここでは述べない．

1000(ms) 付近から 14 番サーボに合わせて 18 番サーボが上がり始め, 4000(ms) 付近で上がりきり, 8000(ms) 付近で初期位置まで戻っているのが分かる．また 8000(ms) 付近から 15 番サーボに合わせて 19 番サーボが上がり始め, 11000(ms) 付近で上がりきり, 15000(ms) 付近で初期位置まで戻っているのが分かる．

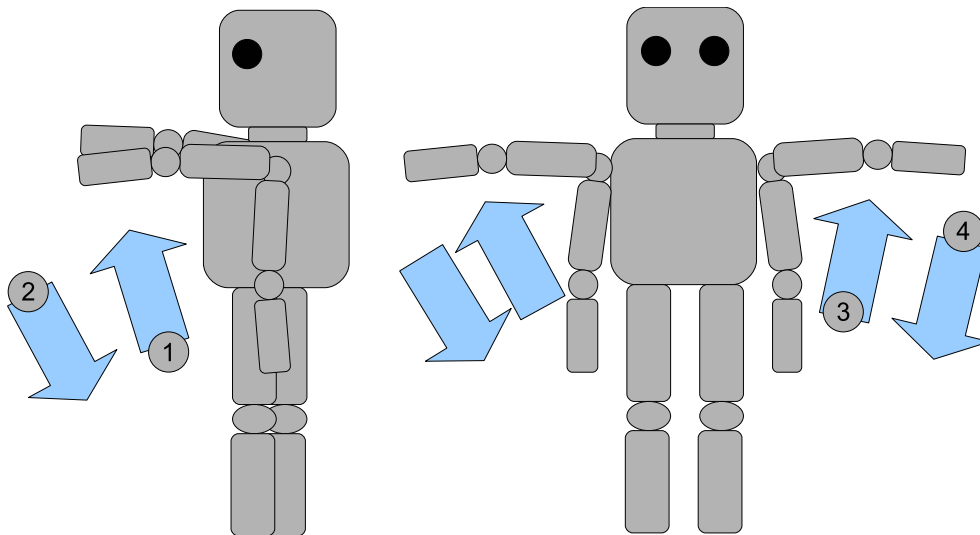


図 19: 追加教示実験 2 回目概要

18 番サーボについて、初期位置が $-1000(0.1 \text{ 度})$ を下回ってしまっているのは、18 番サーボが若干、地面と垂直方向より、ロボットの背中方向に傾いていたためと思われる。腕を水平位置まで上げた際、 $0(0.1 \text{ 度})$ を超えてしまっているのは、腕が水平方向まで上がっていることを目測で確認したため、水平よりも高い位置まで上げてしまったと考えられる。また、前述 2 つの問題に関して、ロボットを組み立てる際、組み立て方によって若干傾いてしまい、それにより位置のずれが生じてしまうことがある。8000(ms) から 14000(ms) 付近で若干値が増えているのは、19 番サーボを操作する際、18 番サーボに対応するジョイスティックの軸方向にぶれてしまったためと考えられる。

19 番サーボについて、腕を水平位置まで上げた際、 $0(0.1 \text{ 度})$ を超えてしまっているのは、18 番サーボと同じように、腕が水平方向まで上がっていることを目測で確認したため、水平よりも高い位置まで上げてしまったと考えられる。

20 番サーボについて、若干値に変動があるのは 18 番、19 番サーボを操作している際は、ボタン 2 を押すことによる操作となるので、ジョイスティックの各軸に 18 番、19 番、20 番サーボが割り当てられている。このことから、操作していない自由度も多少ぶれてしまうためである。また、経過時間 7000(ms) 付近で値が上がっているが、これはサーボの角度の値を取得する際、値の取得ミスによる誤りデータであり、実際にはサーボの動きにはほとんど変化はない。

5.2.4 教示 3 回目

3 度目の教示について述べる。2 度目に教示された動作データを実行中に 16 番、20 番サーボに対して教示を行う。まず、両肩サーボ (14 番、18 番) が前方向に地面に垂直に上がり、初期位置まで戻るまでの間に両肘サーボを体の外側に 90 度程度曲げる。その後、両肩サーボ (15 番、19 番) が横方向に地面に垂直まで上がり、初期位置まで戻るまでの間に両肘サーボを初期位置まで戻す。3 度目の教示の概要を図 21 に示す。

ジョイスティックのボタン 1 に 14 番、15 番、16 番サーボを割り当て、ボタン

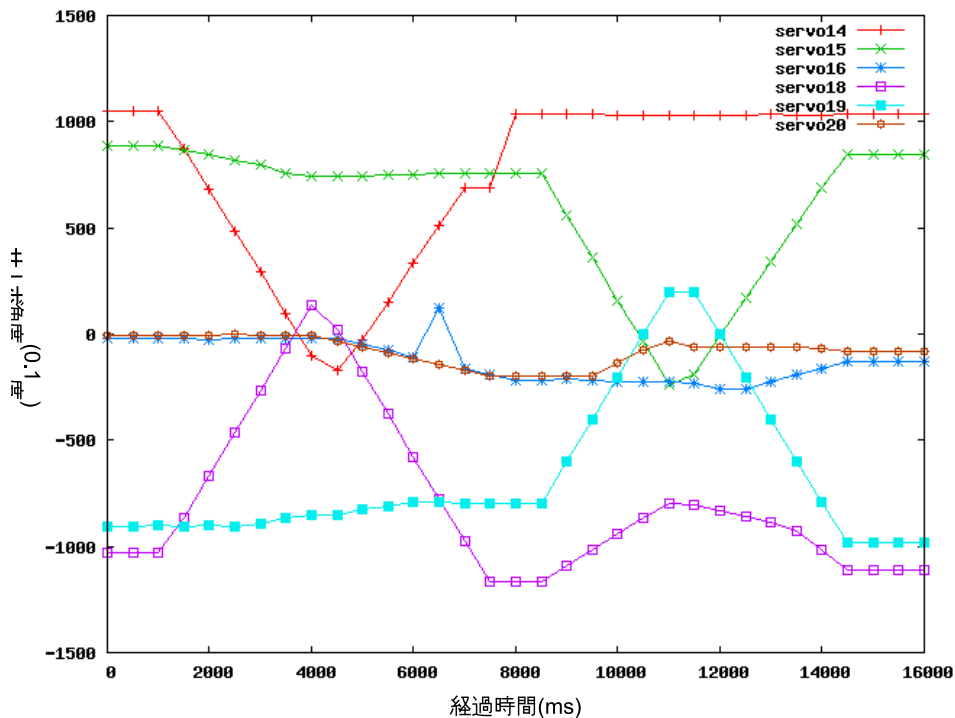


図 20: 追加教示実験 2 回目結果

2 に 18 番, 19 番, 20 番サーボを割り当てて, 分割して操作しているため, 16 番サーボと, 20 番サーボを同時に動かすことは出来ない. そこで, 両肘サーボを体の外に曲げる時は, まず, 16 番サーボについて教示を行い, 90 度程度曲がったのを確認してから, 20 番サーボについて教示を行う. 初期位置まで戻す時も同じように 16 番サーボに教示を行ったあと, 20 番サーボについて教示を行う.

この時, この時 16 番サーボについては, 初期位置の角度は 0(0.1 度) 程度であり, 外側に肘を曲げると, 角度は増加する. 20 番サーボについては, 初期位置の角度は 0(0.1 度) 程度であり, 外側に肘を曲げると, 角度は減少する.

教示された結果を図 22 に示す.

14 番サーボについて, 1000(ms) 付近, 8000(ms) 付近, 13000ms(付近) で角度の値が急激に下がっているが, これはサーボの角度の値を取得する際, 値の取得ミスによる誤りデータであり, 実際には 14 番サーボがこのような動きを行うことはない.

15 番, 18 番, 19 番サーボについては, 2 度目の教示のままなのでここでは述べない.

経過時間 2000(ms) 付近で 16 番サーボの教示が始まり, 4000(ms) 付近で約 90 度曲げ終わり, 続いて 4000(ms) 付近から 20 番サーボの教示が始まり, 6000(ms) 付近で約 90 度曲げ終わっている. 同じように 10000(ms) 付近から 12000(ms) 付近の間に 16 番サーボは初期位置まで角度を戻し, 12000(ms) から 14000(ms) の間で 20 サーボは初期位置まで角度を戻しているのが分かる.

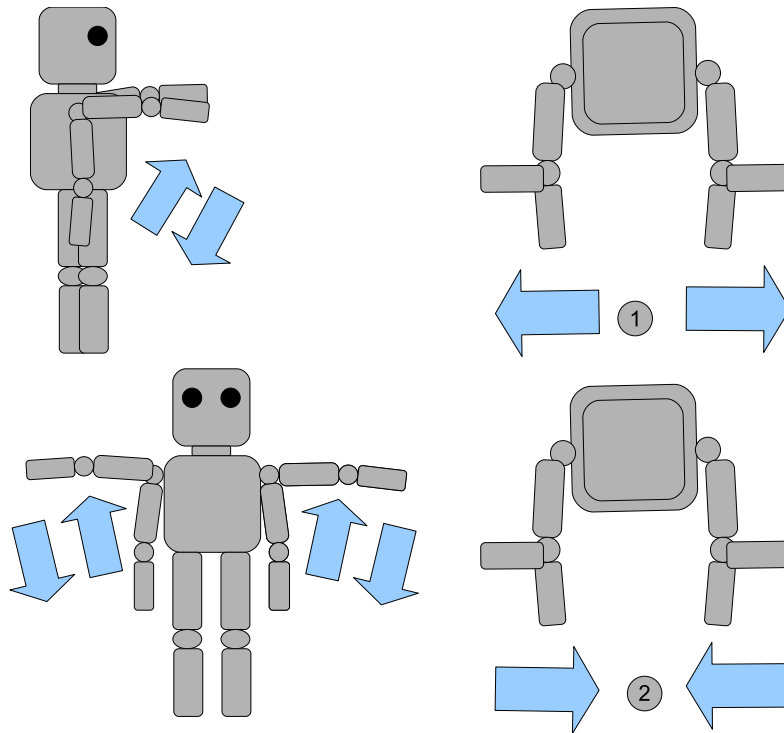


図 21: 追加教示実験 3 回目概要

5.2.5 まとめ

5.2.2 から 5.2.4 までの結果より，ロボットに対して正しく追加教示されていることが分かる．3 回目の教示後の動きは，1.「両腕を前方向に上げながら，両肘を外側に曲げる」，2.「肘はそのまま，両腕を下げる」，3.「両腕を左右に広げながら両肘を元に戻す」，4.「両腕を元に戻す」という動きとなる．

5.3 修正教示実験

5.3.1 概要

本実験では，右肩サーボ 2 個を用いる．肩サーボはロボット正面に対して，前後方向に動く 1 サーボと左右方向に動く 1 サーボついているため，2 サーボとなる．サーボには順番があり，前後方向に動くサーボが 14 番，左右方向に動くサーボが 15 番，である．

ロボットに対して修正教示実験を行う．まず，1 度目の試行で右肩の 14 番サーボと 15 番サーボに教示を行い，2 度目の試行でこれらのサーボに修正を加え．3 度目の試行でさらに修正を加える．腕の初期位置は体に沿うように下に下げた状態であり地面と垂直である．

5.3.2 教示 1 回目

1 度目の教示について述べる．まず，14 番サーボを用いて腕を前方向に地面と水平になるまで上げる．今度は 15 番サーボを用いて右方向に腕を 90 度曲げ，その後初期位置まで下げる．1 度目の教示の概要図を図 23 に示す．

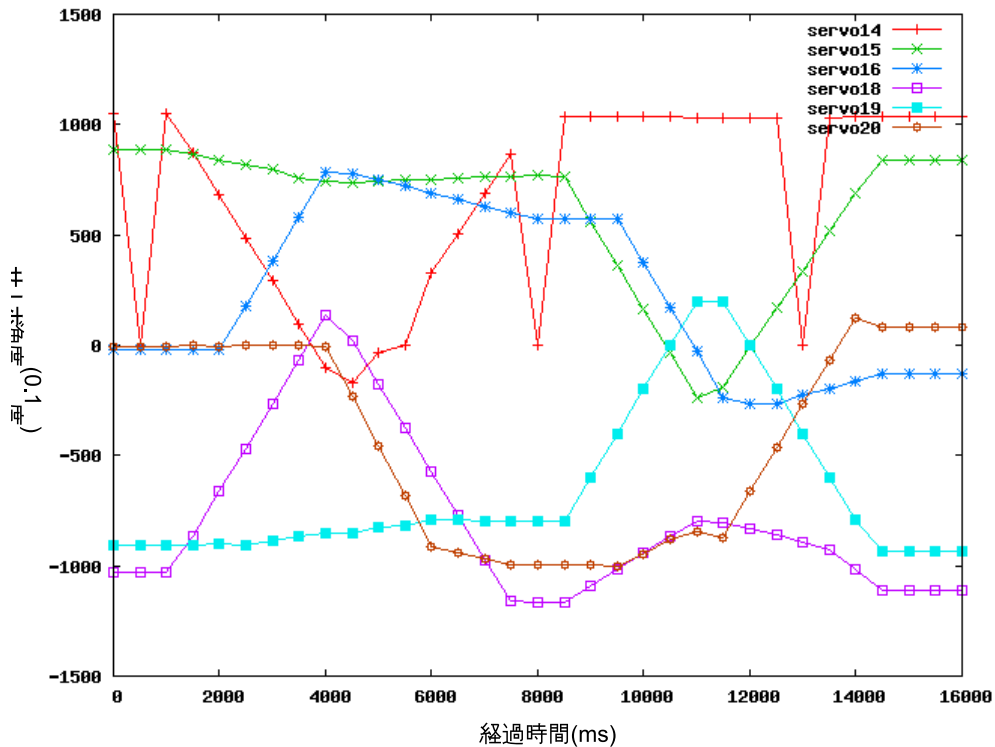


図 22: 追加教示実験 3 回目結果

この時 14 番サーボについては、初期位置の角度は 900(0.1 度) 程度であり、前方向に腕を上げると、角度は減少し、水平付近で 0(0.1 度) 程度となる。15 番サーボについては、初期位置の角度は 900(0.1 度) 程度であり、14 番サーボにより腕が上がった状態から右方向に腕を広げると、角度は減少する。腕を初期位置に戻す際は 14 番、15 番サーボともに初期位置に戻す必要があるため、まず 14 番サーボの角度を先に戻し、その後 15 番サーボを戻すことにする。

教示結果を図 24 に示す。

14 番サーボが地面に水平方向まで上がりきるのが 4000(ms) 付近である。その後 15 番サーボにより、90 程度腕が広げられる。6000(ms) 付近から 14 番サーボが初期位置に向かって動きだし、8000(ms) 付近でほぼ初期位置まで戻る。その後 15 番サーボが初期位置まで戻るよう動き始め、12000(ms) 付近で戻りきっているのが分かる。

14 番サーボについて、6000(ms) 付近、8000(ms) 付近、12000(ms) 付近で急激に角度の値が下がっているが、これはサーボの角度の値を取得する際、値の取得ミスによる誤りデータであり、実際にはこのような急激な動きは行われていない。また、8000(ms) 付近から 12000(ms) 付近まで緩やかに値が変化しているが、これはより初期位置に近づけるために微調整を加えているためである。

5.3.3 教示 2 回目, 3 回目

2 度目, 3 度目の教示について述べる。

2 度目は、1 度目に教示された動作データが実行されている間に動いているサーボに対して修正教示を行う。まず、14 番サーボが腕を前方向に地面と水平

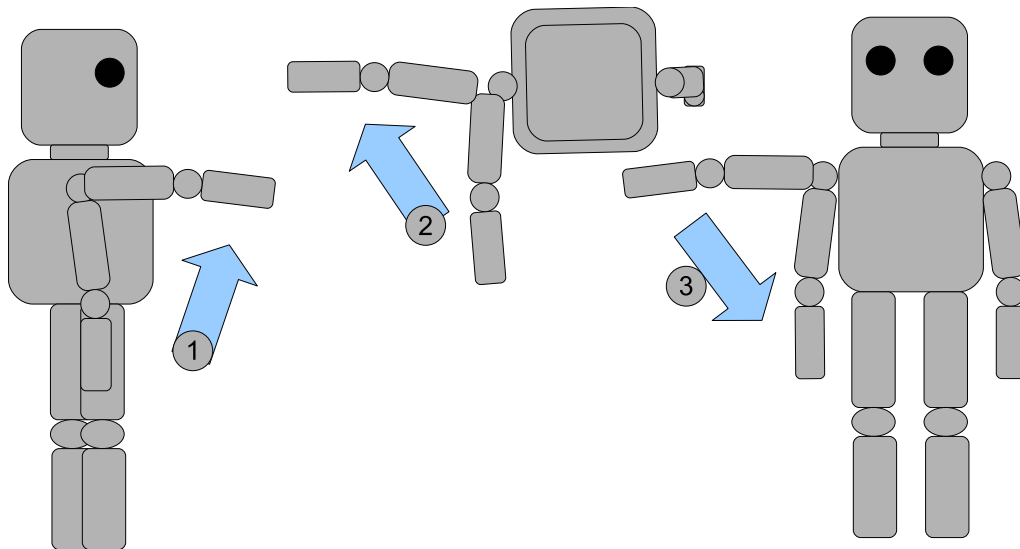


図 23: 修正教示実験 1 回目概要

になるまで上げている間，14 番サーボの同じ方向にさらに入力を加えて水平以上の位置まで腕を上げる．今度は 15 番サーボが右方向に腕を 90 度曲げている間，15 番サーボの同じ方向にさらに入力を加えて 90 度以上腕を曲げる．その後初期位置まで下がるように 14 番，15 番サーボに入力を加える．

また 3 度目の教示実験も同じように，2 度目に教示された動作データに対して行う．

2 度目，3 度目の教示の概要図を図 25 に示す．また，2 度目の教示の結果を図 26 に．3 度目の教示の結果を図??に示す．

まず，2 度目の修正教示の結果について，1 度目の結果と比べると，14 番，15 番サーボともに値の変化が大きくなっていることが分かる．14 番サーボの場合，腕がもっとも上がっている時の値が，1 度目と比べて約 40 度多く上がっていることが分かる．また，15 番サーボについても，1 度目と比べて約 20 度多く曲がっていることが分かる．1000(ms) 付近および 14000(ms) 付近で急激に角度の値が下がっているが，これはサーボの角度の値を取得する際，値の取得ミスによる誤りデータであり，実際にはこのような急激な動きは行われていない．

3 度目の修正教示の結果では，2 度目の結果と比べてサーボの値の変化が大きくなっている異が分かる．14 番サーボの場合，腕がもっとも上がっている時の値が，2 度目と比べて約 30 度多く上がっていることが分かる．これは，1 度目の結果と比べると，70 度程度水平方向より上に動いていることが分かる．また，15 番サーボについても，2 度目と比べて約 40 度多く曲がっていることが分かる．これは，1 度目の結果と比べると，90 度曲がる動きに，さらに 60 度程度加えられていることが分かる．

5.3.4 まとめ

5.3.2 と 5.3.3 の結果より，ロボットに対して正しく修正教示されていることが分かる．1 回目の教示後の動きが，地面と水平になるまで前方向に腕を上げ，そのまま腕を右に広げて，その後下げるという動きであったのに対して，3 回

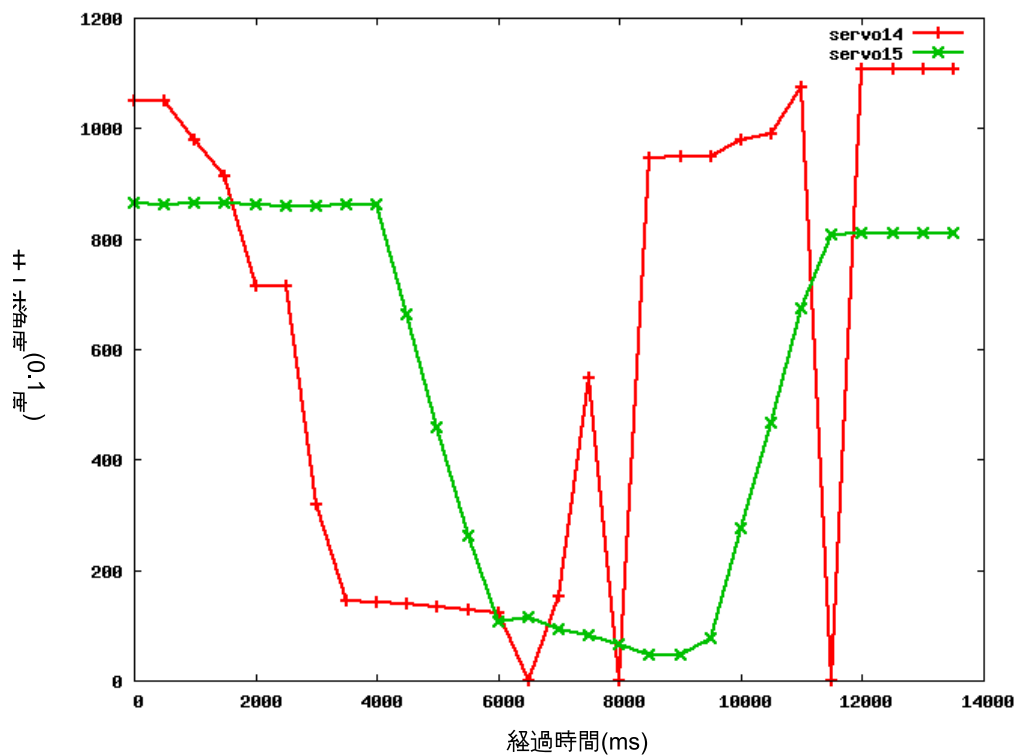


図 24: 修正教示実験 1 回目結果

目の教示後の動きは、腕を前方向に大きくあげ、そのまま腕を回して1周するような動きに変わっている。

5.4 まとめ

2つの実験より、今回作成したシステムについて、「追加教示」「修正教示」が正しく行われていることを確認した。また、「追加教示」「修正教示」ともに1回めの教示は速度入力による初回教示であり、これも実験により、正しく教示されていることを確認した。

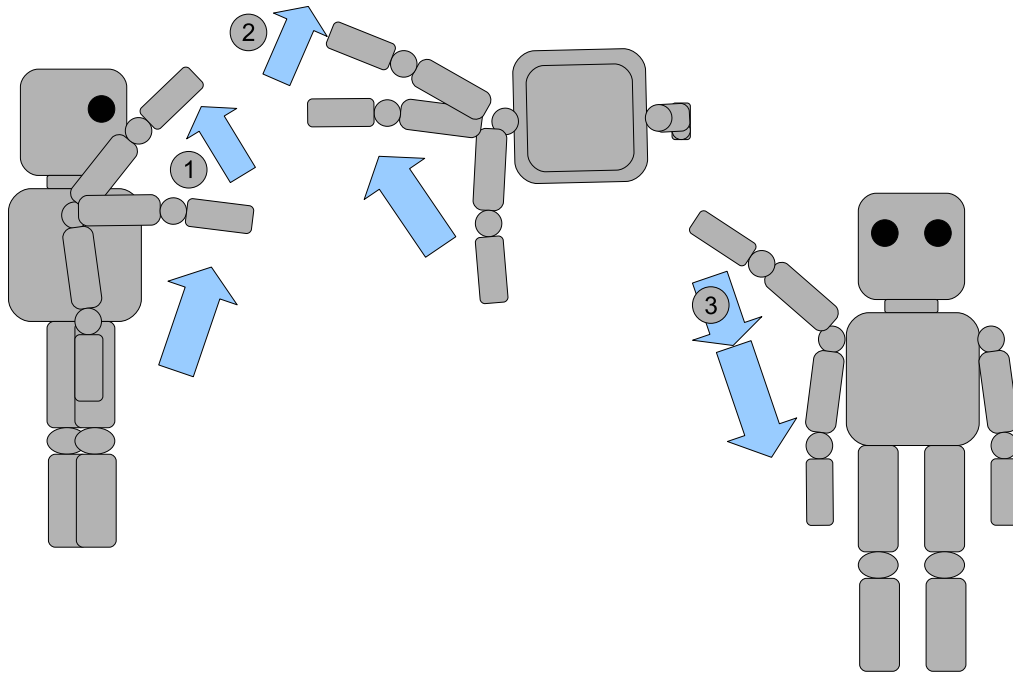


図 25: 修正教示実験 2 回目, 3 回目概要

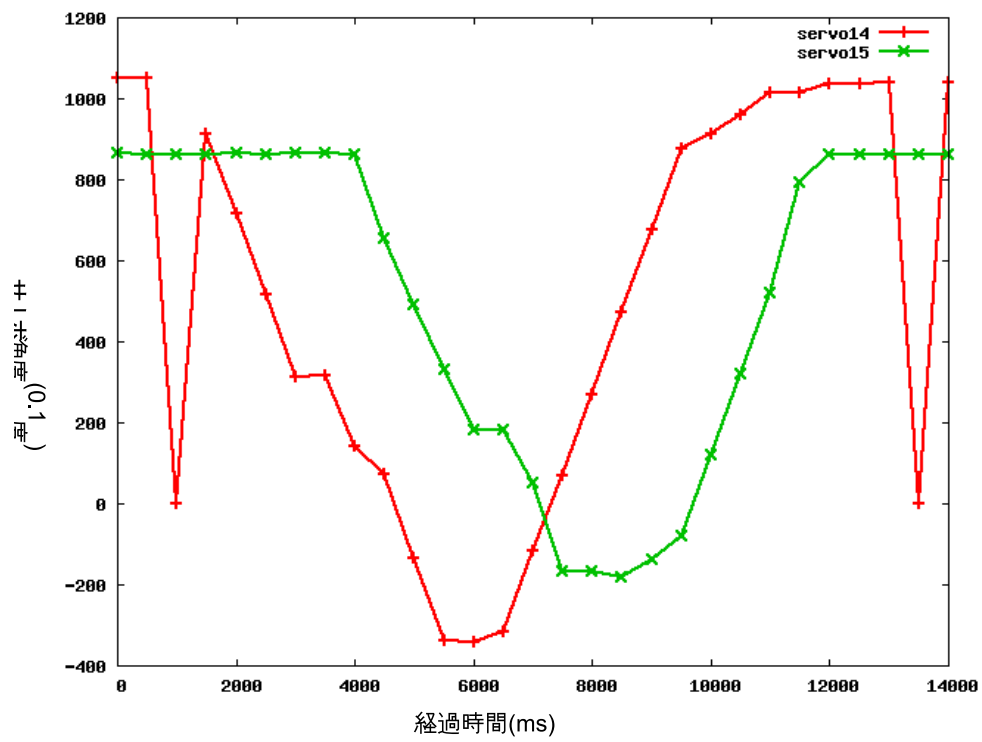


図 26: 修正教示実験 2 回目結果

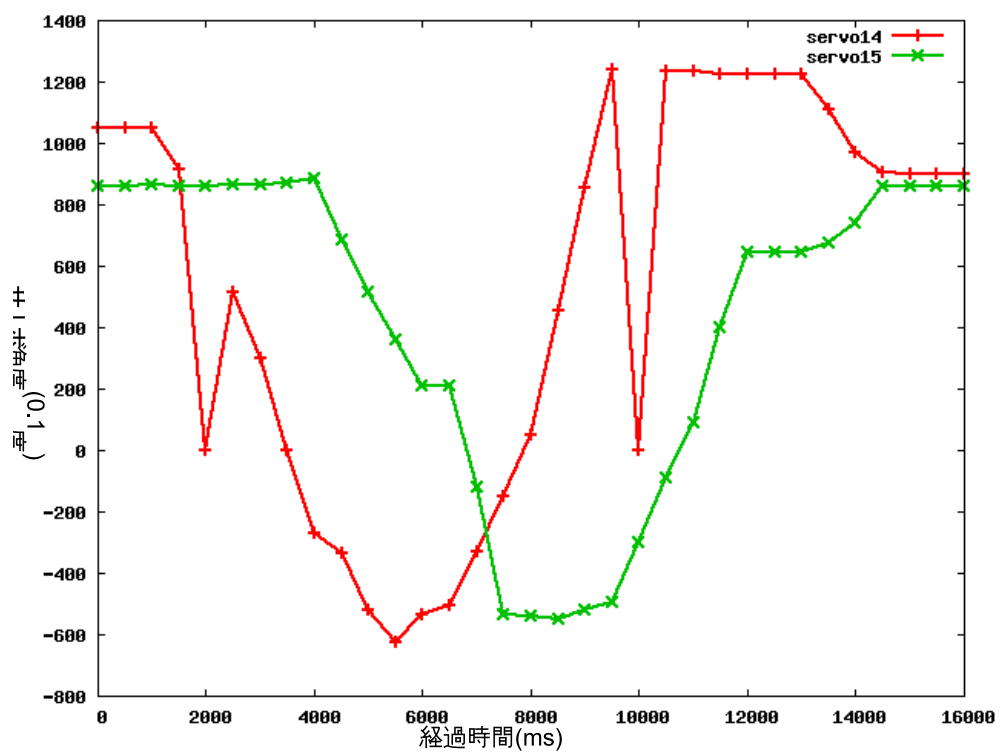


図 27: 修正教示実験 3 回目結果

6 おわりに

6.1 本研究のまとめ

本研究では，ロボットよりも少ない自由度を持つ入力装置を用いて，ロボットへ自由な教示を行う方法の提示という目的のもと，アプローチとしてロボットの自由度を分割し，各々に対して教示を行うという方法を提案した．

自由度の分割とは，ロボットが持っている自由度を，仮想的に複数にグループ分けし，このグループごとに操作を切り替えることであり，この方法を用いることで，ロボットよりも少ない自由度の入力装置ですべてのロボットの自由度に教示することを可能にした．

しかし，1度にすべての自由度に対して教示を行うことは出来ないため，すべての自由度に対して教示を行う方法として，ロボットが過去に教示された動きを実行している間に，別な自由度に対して追加で教示が行うことが出来るシステムを提案した．これにより，過去の動作データでは教示されていなかった自由度に対して新たに追加教示することが可能であり，これを繰り返すことで，すべての自由度に対して同時に教示している場合と同じような動きを作成可能となる．

また，教示したすべての自由度に対して，自由な修正を加えられるシステムを提案した．これは，過去に教示された動きを実行している間に，そのロボットの動きに対して修正を加えることが出来るシステムである．これにより，過去に作成した動作データをよりよい動きに修正することが出来る．修正教示は何度も繰り返すことが出来，またグループ切り替えにより，すべての自由度に修正教示が可能である．そのため，すべての自由度に対して同時に修正を加える場合と同じような修正教示が可能となる．

これらの提案したシステムを実際のロボットと入力装置を用いて実装した．入力装置からの教示データとして，1度目の教示では速度制御を用い，2度目以降の追加，修正教示では，累積入力データによる制御を用いた．累積入力データとは，追加，または修正教示が始まった時点から，現在までの入力データを用いて加算したものである．これを過去の教示データと合わせることで，新しい教示データとすることが出来る．

このシステムを用いて実験を行った結果を示し，今回の実験環境において，ロボットに対して自由な動きを教示出来ることを示した．

本研究により，本論文のアプローチを用いることで，多くの自由度を持つ複雑なロボットに対してジョイスティック等の簡易入力装置により自由な教示を行うことが出来ることを示した．

6.2 今後の課題

まず，本研究のアプローチの問題点として，ロボットの自由度と入力装置の自由度の差が多くなるほど切り替え操作が難しくなるという点を上げる．今回の実験では，ジョイスティックの自由度3に対して操作する自由度は最大で6であったため，2つのボタンを用いて教示することが出来た．しかし，今回使用したロボットは全部で23の自由度を持っているため，これらを自由度3の

ジョイスティックで操作しようとする、8つのボタンを使い分ける必要がある。本研究で用いたロボットよりも多くの自由度を持つロボットも多く存在するため、切り替え操作はさらに複雑になる。これを解決するために、ボタン切り替え以外のロボットの自由度の切り替え方法も模索する必要がある。

次に、今回作成したシステムの問題点として、では教示データを1つしか保存出来ないという点を上げる。単純に動作データごとにファイルを分けて保存する方法も考えられるが、ジョイスティックのような簡易入力装置を用いてこれらの動作データの中から最適なものを選ぶための方法を考える必要があり、また、最適なファイルを選ぶことが出来たとしても、そのファイルに記録されている動作データを忠実に実行することしか出来ず、今回用いた多自由度ロボットが活躍するような複雑な環境で複雑なタスクを行うといった場合にはそれぞれの環境、それぞれのタスクに合わせた教示が必要となり、運用が難しくなる。そのため、動作データの保存方法を改良する必要がある。

また、本研究ではロボットへの教示のための入力方法として、1度目の教示のみ速度入力とし、2度目以降の追加、修正教示の入力方法とはことなる方法を採用した。しかし、これらは同じ入力方法に統一したほうが分かりやすいと思われる。また、今回用いた入力方法とは別な方法もためし、よりよい方法を模索する必要があると考える。

その他問題点として、ロボットからセンサーデータを取得する際、取得したデータが異常値であるかどうかを判断し、修正する方法について、最良の方法が見つからなかった点を上げる。今後、さまざまな方法により、判断、修正する方法を模索する必要がある。

謝辞

本論文を結ぶにあたり，日頃から様々な面で有益な御指導・御助言をいただきました倉重健太郎先生，畑中雅彦先生，本田泰先生，須藤秀紹先生，渡部修先生に深く感謝の意を表します．また多忙の中，査読を行なってくださった修士の尾上由紀子さん，学部4年の木島康隆さん，幾世橋将文さんに感謝の意を表します．

参考文献

- [1] 原督 泉敏之 飛田正俊, "溶接ロボットオフライン教示システム K-OTS ", 神戸製鋼技報, 54, 2, pp.96~99, 2004 .
- [2] 小倉崇 羽根田淳 岡田慧 稲葉雅幸, "自律行動の誘導に基づく等身大ヒューマノイドの全身動作のオンライン教示システム "第 23 回日本ロボット学会 学術講演会, 1F25, 2005 .
- [3] 永田寅臣 渡辺桂吾, "研磨ロボットのためのジョイスティック支援による力制御教示システム ", 日本機械学会論文集 C 編, 67, 655, pp. 177-184