## 修士論文

## 健康増進を目的とする全方向自律移動ロボットの開発

An Omni-directional Autonomics Mobile Robot for Promotion of Health

高知工科大学大学院基盤工学専攻 種植 健二 <u>目次</u>

# 目次

第1章序:	章	
1.1	はじめに ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
1.2	ロボット産業の推移 ・・・・・・・・・・・	3
1.3	本研究の意義 ・・・・・・・・・・・・・・	3
第2章家	庭環境におけるロボット	
2.1	家庭用ロボットに必要な条件 ・・・・・・・・	5
2.2	全方向移動の必要性・・・・・・・・・・・・	5
2.3	動作予告の必要性・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
2.4	対象別認識の必要性・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
第3章全	方向移動メカニズム	
3.1	球駆動メカニズム ・・・・・・・・・・・	10
3.2	移動メカニズム ・・・・・・・・・・・・・	11
3.3	移動速度導出方法・・・・・・・・・・・・	17
3.4	制御システム構成 ・・・・・・・・・・・・・	19
第4章実	<b>験</b>	
4.1	各種センサテスト ・・・・・・・・・・・・	25
4.2	モータ速度回転制御 ・・・・・・・・・・・	35
4.3	全方向移動テスト ・・・・・・・・・・・・	60
4.4	移動方向予告実験・・・・・・・・・・・・・	75
4.5	人と物を区別し回避するための実験 ・・・・・・	77
第5章結	論	
5.1	まとめ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	90
5.2	今後の課題・・・・・・・・・・・・・・・・	91

<u>目次</u> <u>ii</u>

5.3 各部詳細とプログラム ・・・・・・・ 91

## 参考文献

謝辞

第1章 序章 1

## 第1章 序章

## 1.1 はじめに

少子高齢化社会に向け,高齢者とのコミュニケーション,健康管理などの福祉関係の仕事全でを人に任せるのは非常に困難である.そのため,近年高齢者・障害者の生活支援を目的とする福祉介護ロボットの研究開発が盛んに行われている. しかし,病気や障害に陥ってからの介護機器を開発することが必要であると同時に,健康増進や疾病予防に積極的に寄与することも大事である.本研究では,健康増進の一手法であるマッサージロボットを開発することを目的として,マッサージできるマニピュレータ本体を搭載する全方向自律移動ロボットのプラットホームを開発し,実際の環境を想定した基礎実験を行った.

本論文は5章からなり,具体的内容は次のとおりである.

第1章では,ロボットの歴史,産業用ロボットと家庭用ロボットの違いについて述べた上で,本研究の意義を説明する.

第2章では,家庭環境におけるロボットに必要な機能を考察する.家庭用ロボットは人間と同じ空間で活動するので,産業用ロボットと異なり,安全性を確保しなければならない.本論文では安全性を向上するための手法として,ロボットの動きを人間に提示する簡単且つ有効な方法を提案する.また,物のような障害物を回避するだけではなく,人間の動いている動的環境においてはスムーズな移動機能も必須であることを指摘して,人と物とを識別ためのセンサを用いることにより,異なる回避動作を実現する.

第3章では,本ロボットプラットホームの移動機構と制御系を紹介し,其々の特徴について述べる.使用環境として普通の家庭を想定しているので,従来の2輪駆動式の移動ロボットは回転半径が必要となり適用することができない.そのため,球型アクチュエータを用いた全方向移動機構を利用することを提案して,その全方向動作メカニズムについて解析する.続いて,制御システムの構成を紹介して,制御仕様を説明する.

第4章では,実際の家庭を想定した環境で,全方向自律移動ロボットのプラットホームの運動制御に関する基礎実験を行う.具体的にまず,自律性を実現するための各種センサについて確認実験を行った後に,実環境での全方向移動実験により球型アクチュエータの有効性を確認する.さらに安全性を向上するために,次の二つの実験を行う.一つ目としては,ロボットから人に対しての情報伝達を行うことにより安全性を向上する.もし人間が,ロボットが次にどのような行動を行うのかが予測できれば,適切な動作を取れるのでより安全性が確保できると考え,ロボットに LED を付けて,ロボットの移動方向を人間に提示する簡単な方法を採用する.ここで,提示のある無しの場合の実験結果により,異なる安全性が得られることを示す.二つ目は,人と物とを区別するためのセンサを用いることにより,「人を認識した場合」、「物を認識した場合」においてロボットの動作方法に区別をつけることにより,人に対する安全性を確保するためにどのような動作を行う必要がある

のか実験を通して検討する.第5章では,実験結果を検討し,本全方向自律移動ロボットプラットホームの実際システムへの適用可能性について議論する.最後に今後改善すべき問題点を明記しておく.

## 1.2 ロボット産業の推移

ロボット技術は様々な分野で活躍できる可能性を秘めたテクノロジーであり,現在にいたるまであらゆる種類のロボットが人間社会に利益をもたらしてきた.しかし,開発されてきたロボットの多くは工場などで使用される産業用ロボットといわれるものであり,正確に素早く,そしてより多くの製品をしあげる事を目的に,高精度,高出力,高速度といった技術開発が行われてきた.しかし,近年では人には出来ない危険な作業を行うための「レスキューロボット」や「原子炉における補修,修理ロボット」、そして,人との共存を目的とした人にやさしいロボットの研究開発が盛んに行われるようになった.例えば,人との触れ合いに重点を置いた「ペットロボット」、「コミュニケーションロボット」、「アミューズメントロボット」、人の介護支援を目的とした「医療,福祉支援ロボット」などがある.最近では,家庭用ロボットとしても多くのロボットが研究開発され,実際にすでに一般家庭で使用されているものもある.また,近年非常に注目された2足歩行ロボットについては様々な企業が研究開発を行い,すでに階段昇降や踊りなどの動作が行えるほどロボット産業の進歩は目覚しいものとなっている.本研究ではこのロボット産業推移の中,「医療,福祉支援ロボット」の開発に着目し,一般家庭において使用可能な健康増進用ロボットについて研究開発を行う.

## 1.3 本研究の意義

今日、出生率の低迷化に伴う少子化、医療技術の進歩による高齢者の増加という2点から生じる問題の一つとして少子高齢化が挙げられる.そして、この問題が進行するにつれて、看護、介護といった職種における人員不足という問題が重要となって来る.このような人員不足が発生すると、高齢者や障害者に対して「コミュニケーション」、「健康管理」といった生活支援を全て人に任せるのは非常に困難となる.そこで、高齢者や障害者の生活支援を目的とする福祉介護ロボットの研究開発を行う必要性が生じる.しかし、病気や障害に陥ってからの介護機器を開発することが必要であると同時に、健康増進や疾病予防に積極的に寄与することも重要である.そこで本研究では健康増進の一手法であるマッサージロボットを開発することを目的とする.

本論稿では,このマッサージロボットのマニピュレータを搭載する移動機構であるプラットホームの開発を行う.一般家庭内のような狭く複雑な環境において,人とロボットが共存するためにロボットは安全を重視し,また人の障害物にならないようにスムーズに回避移動しなければならない.そこで,ロボットには人と物とを見分け尚且つスムーズに移動するための機能が必要となる.そこで,この2点に着目し実際の一般家庭を想定した移動ロボットの基礎実験を行う.

具体的にどの程度少子高齢化が進行しているのかについて図 1-3-1 ,表 1-3-1 に日本の人

<u>第1章 序章 3</u>

#### 口推移データを示す.

図 1-3-1 は年齢構造を「年少人口」、「生産年齢人口」、「老年人口」の3つにまとめた人口推移データ [10] であるが、1997年に初めて老年人口が年少人口を上回り徐々にその差が開いているのが図 1-3-1 から分かる.

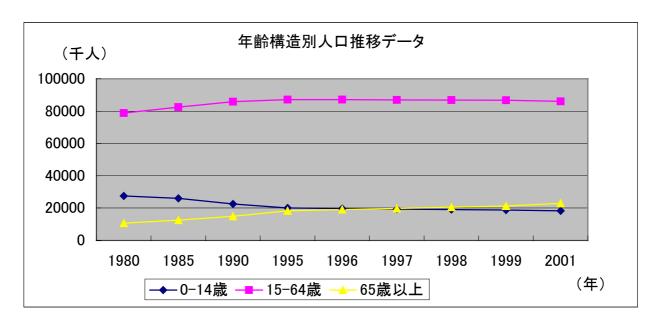


図 1-3-1. 人口推移グラフ

表 1-3-1. 人口推移データ

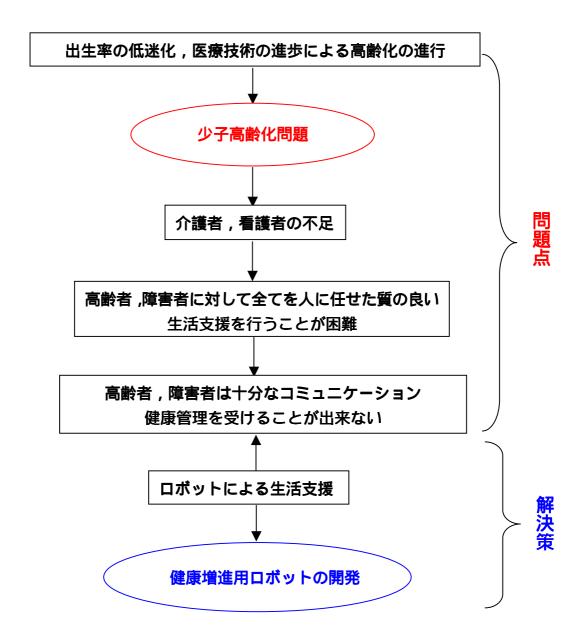
		1980年	1985年	1990年	1995年	1996年	1997年	1998年	1999年	2001年
1	0-14	27507	26033	22486	20014	19686	19366	19059	18742	18260
2	15-64	78835	82506	85904	87165	87161	87042	86920	86758	86080
3	65歳以上	10647	12468	14895	18261	19017	19758	20508	21186	22930

(単位:千人)

注:一般的に人口の年齢構造は1.年少人口(15歳未満),2.生産年齢人口(15歳以上 65歳未満),3.老年人口(65歳以上)に区分される.

次に,研究推移についての概略図を図1-3-2に示す.

第1章 序章 4



1-3-3. 研究推移概略図

## 第2章 家庭環境におけるロボット

## 2.1 家庭用ロボットに必要な条件

家庭用ロボットにおいて必要とされることは主に次のである.[7],[9]

- 1.人に危害を加えない
- 2.人に不安を与えない
- 3.人に恐怖を与えない

そこで,健康増進用ロボットを開発するにあたり,これら3つの条件を満たしたロボットを開発する必要がある.

一般的に工場などで用いられる産業用ロボットは、確立されたマニュアルに基づき、スキルのある人間が操作し、しっかりとした管理の元で安全性が保たれている。しかし、簡単なマニュアルに基づき誰でも取り扱うことの出来るロボットが一般家庭のような狭く、複雑な環境において人と共存するためには、人に「危害」、「恐怖感」、「不安感」を与えずに、人に対して常に安全を確保する必要がある。従来の産業用ロボットでは、「高精度」、「高出力」、「高速度」といった如何に効率よく仕事をこなす事ができるかという事に対して重要視されてきたが、家庭用ロボットには産業用ロボットのような条件をそのまま適応することは非常に危険である。例えば、ロボットが人をマッサージしていた場合、何らかのエラーが発生しロボットが緊急停止をしたとする。この場合ロボットは人間に対し何らかのマイナス要因を与えている状況にあり、その要因が継続して人間に与えられることにより人間は「危害」、「恐怖感」、「不安感」をロボットから受けることになる。そこで、人と共存し常に人と接する環境においては、先にあげた「人に危害を加えない」、「人に不安を与えない」、「人に恐怖を与えない」という3点を踏まえたロボット開発を行わなければならない。

そこで,本ロボットでは「全方向機能」、「移動方向予告機能」、「人認知機能」について基礎実験を行うことにより,先の3点を満たす一般家庭で使用可能な健康増進用ロボットの移動手段となるプラットホームの開発を行う.

## 2.2 全方向移動機能の必要性

ではまず初めになぜ全方向移動機構が必要となるのかについて全方向移動機能を有さないロボットを図 2-2-1,全方向移動機能を有するロボットを図 2-2-2 に示し述べる.

図 2-2-1 に示されたような全方向移動機能を有さないロボット(例えば自動車のような移動機構)であれば黒枠内に移動するために縦列駐車を行うように舵角操作を行うことにより1番から2番の位置に移動し、それから2番から3番へ移動するために何度も舵角操作を行いながら移動しなければならない.しかしながら図 2-2-2 に示されたような全方向移動機能を有すような機構であれば黒枠内に収まるために何度も最低一回のみの舵角操作

により移動することが可能となる.このような全方向移動機能は狭い環境において非常に 有効な機構であり,一般家庭での使用に適していると言える.

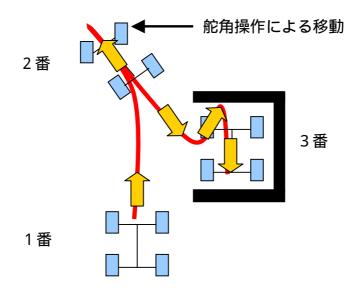


図2-2-1.全方向移動機能無し

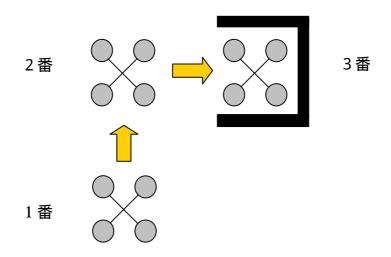


図2-2-2. 全方向移動機能有り

続いて,全方向移動機構について簡単に説明する.現在では数多くの全方向移動ロボットが研究開発されており,全方向移動機構の多くは主に次の3つに分類される. [1]-[6]



図 2-2-3. 球型機構

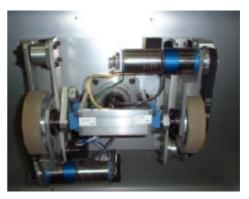


図 2-2-4. 車輪型機構

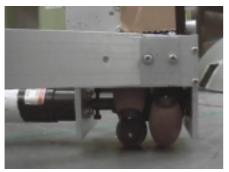


図 2-2-5. オムニホイル型機構

これらの機構の中で,一般的に全方向移動機構としては車輪機構が多く用いられている.しかし、本研究においてのロボットを使用する環境は、屋外のように砂利道があったり様々な段差があったりするような場所や工場のような広い場所ではなく,一般家庭のような狭く障害物が多く存在するような空間での使用を目的としている.そのため,家庭用ロボットは単純作業しか出来ない産業用ロボットと違って,人の動きに応じて,臨機応変に移動するための機能を持つことが必要となってくる.つまり,人との共存を目的とした家庭用ロボットは、人間にとって安全に作業をするロボットでなくてはならない.そのためには,狭い場所であっても無駄なくスムーズに動作をすることが必要となる.つまり,ロボットは自身の向きを変更することなく,ロボットが人にとって障害物とならないように円滑に作業を行いながら多様な動きをすることが重要である.また,一般家庭のような場で重要視されることの一つに「騒音問題」がある.ロボットに限らず,家電製品(洗濯機,掃除機等)は静かに動作する設計となっているものが多く,また実際にそのような物が好まれている.そこで,表 3-1-1 に示された特徴を踏まえ,「全方向移動可能」、「静かに移動」という2点を備えた機構として本ロボットには「球型アクチュエータ」を使用する.

	長所	短所
多少粗い路面でも地形に関係な <b>車輪</b> 移動することが可能		全方向へ移動するために、舵角操作を必要とする
舵角操作などを必要とせずに全 球 方向へ移動か可能		路面状態に作用されやすく, 使 用場所が限定される
オムニホイル	舵角操作などを必要とせずに 全方向へ移動か可能	移動時に細かい騒音が発生しやすい

表 3-1-1. 全方向移動に使用される機構の特徴

### 2.3 動作予告の必要性

日常生活において,人と人とのコミュニケーションにおいては何らかの予備動作(目の動きによる視線,筋肉の動き,言葉)があるため,円滑に様々な作業を行うことが可能である.しかし,人とロボットの間でそのようなコミュニケーションは確立されておらず,人にとってロボットが次にどのような動作を行うのか判断するのは非常に困難である.そこで,時々刻々と変化があり,狭く複雑な家庭環境であっても,人とロボットがともに円滑に作業するためには,人はロボットが次にどんな動作を行うのか知る必要がある.そこで,本研究では「移動方向予告機能」によってロボットから人への情報伝達を行い,その情報を人がロボットから得ることによりコミュニケーションを行う.そして,実験では実際にこのような予告機能を用いることにより,人が安心感を持ってロボットと共存し円滑に作業を行うことが可能となるか,実際にアンケートを行うことにより検証する.実験については第4章について述べる.

## 2.4 対象別認識の必要性

先の 2.2 で述べた「ロボットから人への情報伝達」と同様に重要となることが「人からロボットへの情報伝達」である .狭く煩雑な空間において ,人と物とを等価値に置いた様々な動作をロボットが行っていたのでは非常に危険であり ,人に危害を与える危険性を生じる . そこで , ロボットは人と物とを区別し ,人の行動をロボットが認識することにより , 余裕を持った動作を行い ,人に安心感を与えなければならない . そのためには人が「今何をしているのか」,「どこに向かっているのか」,「どこにいるのか」など ,人に関する詳細

なデータをロボットは常に把握し続けていなければならない.そこで,本ロボットには焦電型 MP モーションセンサを用いることにより,人と物とを区別するための簡単な基礎実験を行う.実験ではロボットが人を認識した動作をすることによって,実際にロボットの近くにいる人がどのように感じるのか検証する.実験については第4章で述べる.

## 第3章 全方向移動メカニズム

## 3.1 球駆動メカニズム

球型アクチュエータ機構動作について図 3-3-1,図 3-3-2に示し述べる.本機構では,モータを回転させることによりモータが赤色矢印で示されたように手前側へ回転することにより,ローラリングがモータの回転方向とは逆に奥側へ回転することになる.このような動作によりローラリングに接している球がローラリングと同時に回転する.そうすることによって移動するための動力を得ることが出来る構造となっている.そして,図 3-3-2に示されたようにローラリング下部周辺にフリーローラを八方向に配置することにより,全方向移動しても球に負荷を掛けることなくスムーズに全方向へ移動することが出来る機構となっている.

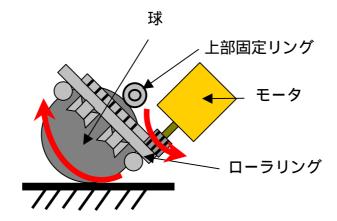


図3-1-1. 球型アクチュエータ概略図

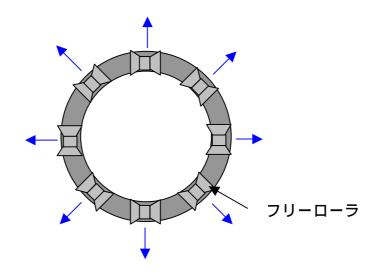


図3-1-2.下から見たローラリング概略図

## 3.2 移動メカニズム

続いて移動メカニズムについて図3-3-3,図3-3-4を用い説明する.

先の図 3-3-1 で示された球型アクチュエータは図 3-3-3 のように球をタイヤに置き換えることにより移動メカニズムが理解できる.路面と角度をもってタイヤが回転する場合,タイヤはまっすぐに進むことなく図 3-3-4 に示されたように円形を描くように移動する.このような原理を用いた移動メカニズムを採用している.

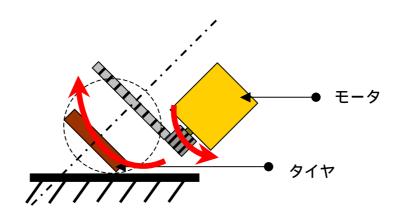


図3-2-1. 球駆動

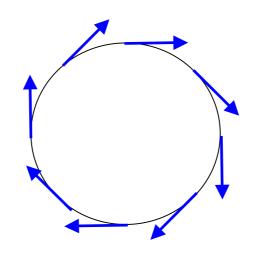


図3-2-2. タイヤの描く軌跡

ではこの駆動メカニズムを用い実際にどのように動作することにより,全方向へ移動が可能となるのか,概略図を図 3-4-1~3-4-11 に示し,表 3-4-1 に移動方向別の球の回転方向について示す.

(1) 本機ロボットが「前後左右方向」へ移動する場合について,図 3-2-3~3-2-6 に示す.

## パターン1:前進移動

図 3-2-3 に示されたように, 黄色で示された各モータが赤矢印で示されたように回転することにより各球はそれとは逆回転をする.図 3-2-2 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより直進移動が可能となる.

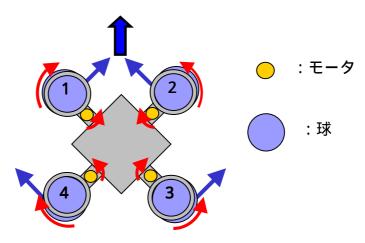


図3-2-3. 直進移動

## パターン2:後進移動

図 3-2-4 で示されたように, 黄色で示された各モータが赤矢印で示されたように回転することにより各球はそれとは逆回転をする.図 3-2-2 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより後進移動が可能となる.

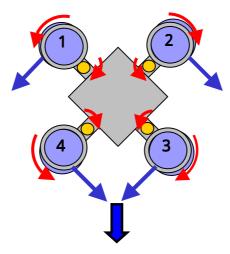


図3-2-4. 後進移動

#### ・パターン3:右移動

図 3-3-7 で示されたように, 黄色で示された各モータが赤矢印で示されたように回転することにより各球はそれとは逆回転をする.図 3-2-4 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより右移動が可能となる.

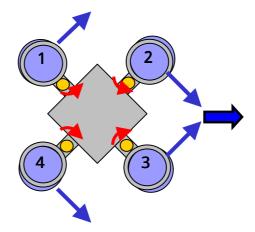


図3-2-5. 直進移動

#### ・パターン4:左移動

図 3-2-6 で示されたように, 黄色で示された各モータが赤矢印で示されたように回転することにより各球はそれとは逆回転をする.図 3-2-2 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより左移動が可能となる.

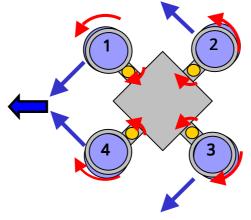


図3-2-6. 直進移動

(2)図3-2-7~3-2-10に示したように,斜め移動を行う場合,対角上に位置する一対のモータを駆動することにより表3-2-1に示すように球は回転する.その時,もう一対の対角上に位置するモ-タは駆動しておらず,球はローラリング周辺に固定されたフリーローラによって進行方向へ回転する.

#### ・パターン5:右上移動

図 3-2-7 で示されたように, 黄色で示された2つのモータが赤矢印で示されたように回転することにより2つの球はそれとは逆回転をする.図 3-2-2 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより左移動が可能となる.このとき駆動していないアクチュエータに関してはフリーローラにより移動方向へ球自体が進行方向へ回転する.

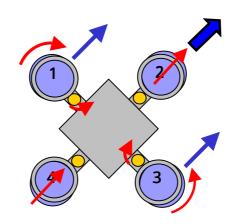


図3-2-7. 右上移動

## ・パターン6:右上移動

図 3-2-8 で示されたように, 黄色で示された2つのモータが赤矢印で示されたように回転することにより2つの球はそれとは逆回転をする.図 3-2-2 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより左移動が可能となる.このとき駆動していないアクチュエータに関してはフリーローラにより移動方向へ球自体が進行方向へ回転する.

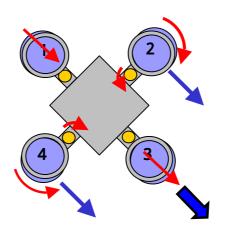


図3-2-8. 右下移動

#### ・パターン7:左上移動

図 3-に-9 で示されたように,黄色で示されたにつのモータが赤矢印で示されたように回

転することにより2つの球はそれとは逆回転をする.図3-2-2で示されたように各球には 青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより左移動が可能となる.このとき駆動していないアクチュエータに関してはフリーローラにより移動方向へ球自体が進行方向へ回転する.

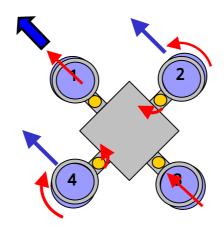


図3-2-9. 左上移動

#### ・パターン7:左上移動

図 3-2-10 で示されたように,黄色で示された2つのモータが赤矢印で示されたように回転することにより2つの球はそれとは逆回転をする.図 3-2-2 で示されたように各球には青矢印で示されたように力ベクトルが生じる.そしてこの力ベクトルを合成することにより左移動が可能となる.このとき駆動していないアクチュエータに関してはフリーローラにより移動方向へ球自体が進行方向へ回転する.

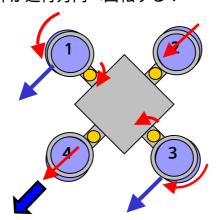


図3-2-10. 左下移動

斜め 45 度方向以外への任意方向への移動領域に対しては,対角上に位置する4つのモータの回転を微妙に変化させることにより生じる力ベクトルを制御することにより移動することが可能である.

:ロボット中心点

(3)回転移動について説明する.回転移動を行う場合,各球は全て同方向へ回転することとなる.

#### ・パターン8:右回転

図 3-2-11 に示されたように ,全てのモータを左回転することにより球が右回転するこのときに青矢印のような力ベクトルが生じるがこれらを合成しても力はゼロとなるために黒点を中心として回転するとなる .

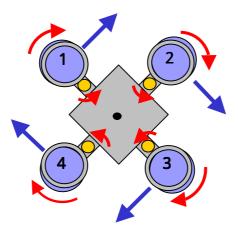


図3-2-11. 右回転

## ・パターン9:左回転

図 3-2-12 に示されたように ,全てのモータを右回転することにより球が左回転するこのときに青矢印のような力ベクトルが生じるがこれらを合成しても力はゼロとなるために黒点を中心として回転するとなる .

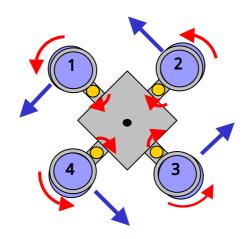


図3-2-12. 左回転

これら (1)~(3)の移動パターンを組み合わせることにより,全方向に対してスムーズに移動することが可能となる.

## 3.3 移動速度導出方法

移動速度導出について図 3-3-1,図 3-3-2 を用いて,式 3.3.1~3-3-4 に示し,具体的に各球が回転することによりどのようなベクトル力が生じ,前進,斜め,回転,任意方向斜め移動するかについて,図 3-4-9~3-4-11 に示す.[3]

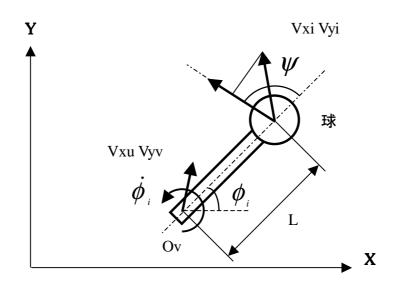


図 3-3-1. 球移動速度

Vxi Vyi : 球速度

Vxu Vyv : ロボット中心速度

Vai : active方向速度 Ov : ロボット中心速度

L: ロボット中心から球中心までの長さ

Ov: ロボット中心ψ: 梁の角速度

 $oldsymbol{\phi}_i$ : 球回転方向と梁のなす角度

$$\begin{bmatrix} V_{xi} \\ V_{yi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -L \sin \phi_i \\ 0 & 1 & L \cos \phi_i \end{bmatrix} \begin{bmatrix} V_{xv} \\ V_{yv} \\ \phi_i \end{bmatrix}$$
 (3.3.1)

$$v_{ai} = v_{xi} \cos(\phi_i + \psi) + v_{yi} \sin(\phi_i + \psi)$$

$$= v_{xu} \cos(\phi_i + \psi) + v_{yv} \sin(\phi_i + \psi) + \dot{\phi}_i L \sin \psi$$
(3.3.2)

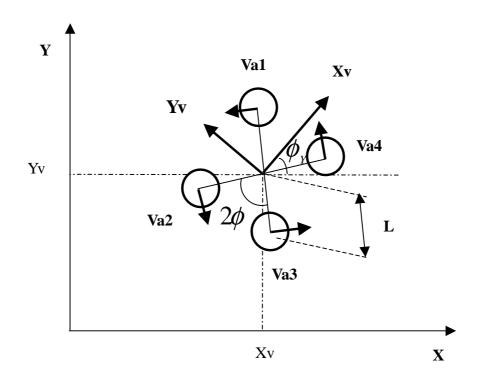


図 3-3-2. ロボット移動速度

Va1, Va2, Va3, Va4 : 球速度

L: ロボットの中心から球の中心までの長さ

 $2\phi$  : X軸とXvのなす角度

 $\phi_{_{_{\boldsymbol{y}}}}$  : 2梁間の角度

$$\phi_{1} = \phi_{v} + \phi_{Y} \qquad \phi_{2} = \phi_{v} - \phi_{Y} + \pi 
\phi_{3} = \phi_{v} + \phi_{Y} + \pi \qquad \phi_{4} = \phi_{v} - \phi$$
(3.3.3)

式 3.3.3 を式 3.3.2 に代入

$$\begin{bmatrix} V_{a1} \\ V_{a2} \\ V_{a3} \\ V_{a4} \end{bmatrix} = \begin{bmatrix} \cos(\phi + \psi) & \sin(\phi + \psi) & L\sin\psi & L\sin\psi \\ \cos(\phi + \psi) & -\sin(\phi + \psi) & L\sin\psi & -L\sin\psi \\ -\cos(\phi + \psi) & -\sin(\phi + \psi) & L\sin\psi & L\sin\psi \\ -\cos(\phi + \psi) & \sin(\phi + \psi) & L\sin\psi & -L\sin\psi \end{bmatrix} \begin{bmatrix} V_{xv} \\ V_{yv} \\ \phi_v \\ \phi \end{bmatrix}$$
(3.3.4)

## 3.4 制御システム構成

本ロボットは一般にパソコンに使用されているマザーボードを使用する事により,安価で拡張性高いシステムを構築している.そのため必要に応じてシステム仕様を変更することが非常に用意であり,ロボット開発の段階に応じたシステムを構築することが可能である.そのため,OS などに依存されることなくロボットを制御することが可能である.また,故障などの不具合が起こった場合には迅速な対応を行うことが出来る.

初めに,本システムの制御システムブロック線図を図3-4-1示す.

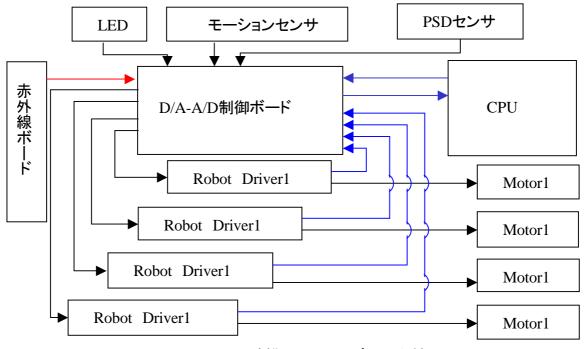


図 3-4-1.制御システムブロック線図

(1)各種センサの特徴と反応別の制御構成について次に説明する.

#### (a) タッチセンサについて

タッチセンサは障害物に物理的接触をしてしまった場合に動作する.

制御構成は,図3-4-1に示すように,障害物に接触した場合 D/A から電圧が出力されその電圧が A/D に入力される.その反応に対して CPU が D/A-A/D 制御ボードを介すことによりロボットドライバへ電圧を出力するように命令をし,ロボットドライバはその入力された電圧に応じてモータへ電圧を供給する.

#### (b)PSD センサについて

PSD とは Position Sensitive Detector といい.いわゆる距離測定素子で,一般に PSD と言うと受光素子単体のことをさす.PSD を用いて距離を測定するには,受光素子である PSD に,発行素子の LED,信号処理回路などが必要であるが,今回はこれらすべてが一体となった距離測定センサを使用した.距離測定の原理はいわゆる三角測量で,反射光の位置情報を元にして障害物の距離を測定する.この方法には,反射物の色や反射率,使用環境の明るさによる影響が少ないというメリットがり,実際の距離計測原理について図 3-4-2,図 3-4-3 を用いて示す.実際の PSD センサについては図 3-4-4 に示す.

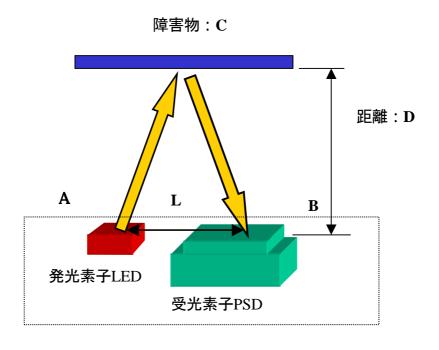


図 3-4-2. 距離計測原理

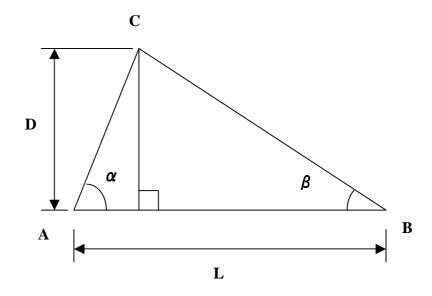


図 3-4-3. 三角測量法

$$D = L \frac{\sin\alpha + \cos\beta}{\sin(\alpha - \beta)}$$
 (3.4.1)

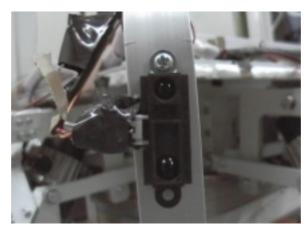


図 3-3-4. PSD センサ

障害物 C に対して二点 A,B から対象物への角度 ( 、 ) と 2 点間 AB の距離 ( L ) が分かれば対象 C への距離 ( D ) が計算することができる。これが三角測量法です。

このような PSD センサを使用する場合 , 図 3-5-1 に示されるように A D/D A ボードの A D に配線されており , センサからは常時電圧値が A/D に入力される . そして , 本ロボットで設定された距離の反応に対して CPU が D/A-A/D 制御ボードを介すことによりロボットドライバへ電圧を出力するように命令をし , ロボットドライバはその入力された電圧に

応じてモータへ電圧を供給する.

### (c) 焦電型 MP モーションセンサについて

今回使用する焦電型 MP モーションセンサは焦電素子を用いたアンプ内臓タイプの人体 検出赤外線センサである.焦電型 MP モーションセンサとは「周囲と温度差のある人(物) が動く際に起こる赤外線の変化を検出するセンサ」である.表 3.5.1 にセンサの検出性能 と検出対象の条件を示す.このセンサの検出可能な波長は約 5 μ m以上であり、人体の波 長は 10 μ mとなっているのでこのセンサを用いて人体検出を行うことが可能となる.実際 の焦電型 MP モーションセンサについては図 3-4-5 に示す.

注)焦電素子とは:素子表面をあらかじめ帯電させておき、検出対象物の出す赤外線によって変化する「素子表面の電荷の状態」を「電圧変化」としてとりだすようなセンサ素子のこと。

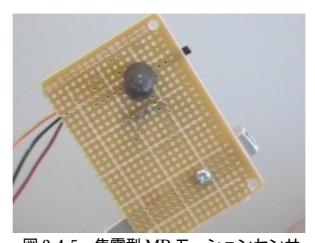


図 3-4-5. 焦電型 MP モーションセンサ

検出距離		最大 5m
検出範囲	左右角	100度
1/1040	上下角	82 度

表 3.4.1 センサ性能

1.背景との温度差: (3±1 度)以上 2.移動スピード: 0.3~2.0m/s

3.検出対象は人体: サイズは 700×250mm

焦電型 MP モーションセンサは図 3-5-1 に示されるように A D/D Aボードの A D に配線されており,センサからは常時一定電圧値が A/D に入力されているが,人体に反応したときのみ電圧値が変化する.そして,それに応じた動作を CPU が D/A-A/D 制御ボードを介すことによりロボットドライバへ電圧を出力するように命令をし,ロボットドライバはその入力された電圧に応じてモータへ電圧を供給する.

(2) ロボット用インタ - フェ - スボ - ド RIF-01 との Titech Robot Driver, 各種センサ, リモコンボード, LED のポ - ト使用詳細を以下にまとめる.(制御プログラム上では Joystick, Sonic Sensor, Sonic Sensor Board を使用しているためこれらを使用する場合のポートについても示す.)

RIF-01 使用ポ・ト ( CN1-D/A ) 表 3-5-2 . Robot Driver 使用ポ・ト

ROBOT DRIVER1 DA00
ROBOT DRIVER2 DA01
ROBOT DRIVER3 DA02
ROBOT DRIVER4 DA03

表 3-5-3 . Sonic Sensor 使用時ポ - ト

	CN1	DA04
SONIC SENSOR1	CN2	DA05
	CN3	DA06
	CN1	DA07
SONIC SENSOR2	CN2	DA08
	CN3	DA09

RIF-01 使用ポ・ト ( CN2-A/D ) 表 3-5-4 . Joystick 使用時ポ・ト

	AD00
Joystick	AD01
	AD02
	AD03

表 3-5-3 . LED 使用ポート

LED1	DA04
LED2	DA05
LED3	DA06
LED4	DA07

表 3-5-5 . Touch Sensor 使用ポート

	AD04
Touch Sensor	AD05
	AD14
	AD15

表 3-5-6 . Sonic Sensor Board 使用時ポ・ト

		AD05
SONIC SENSOR1	5V	PC 電源
	GND	PC 電源
		AD04
SONIC SENSOR2	5V	PC 電源
	GND	PC 電源

## RIF-01 使用ポ - ト (CN3-UPP1)

表 3-5-7. 赤外線リモコンボード使用ポート

	UPP00	UPP04
赤外線リモコンボード	UPP01	UPP05
	UPP02	UPP06
	UPP03	UPP07

## 表 3-5-8. 焦電型 MP モーションセンサ使用ポート

焦電型 MP モーションセンサ	UPP08
	UPP09

## 第4章 実験

## 4.1 各種センサテスト

( ) PSD センサの出力電圧値安定化テスト

ここでは, PSD センサを用いた 2 種類の実験を行い, その性能について述べる.

実験方法: ここでは,PSD の電圧値を安定化させるための実験を行う.方法は, (a)PSD センサのみを使用し,電圧値の値を測定する.(b)PSD センサのみを使用し,平均化された電圧値を測定する.(c)PSD センサにコンデンサを取り付け,それによる安定化を行い,電圧値を測定する.(d)PSD センサとコンデンサ,そしてその値の平均化を行い,その電圧値の測定を行う.この4パターンのテストグラフ化したものを下記に示し,その中から最適なものを用いる.(PSD のみを使用した場合以外のデータ値は,それぞれの実験で5回ずつテストを行う)

#### (a)PSD センサのみを使用した場合

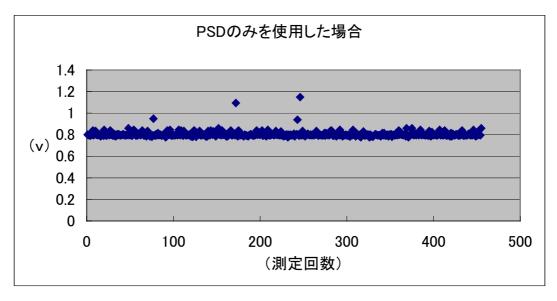


図 4-1-1

結果: この場合,測定最中に何度もおよそ 0.2 V 以上の電圧値の変化が見られる. (b)PSD の電圧値を 1 0 回平均化した場合

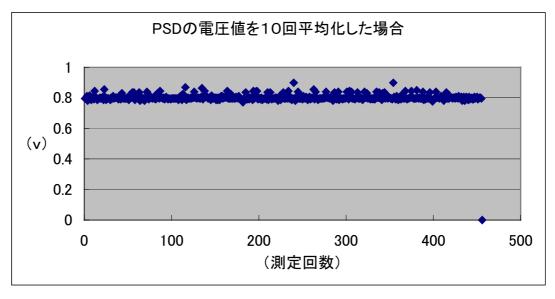


図 4-1-2

結果:10回平均をとった場合,電圧変化はおよそ0.1V程度に収まっている.

(c)PSD センサにコンデンサを取り付けた場合

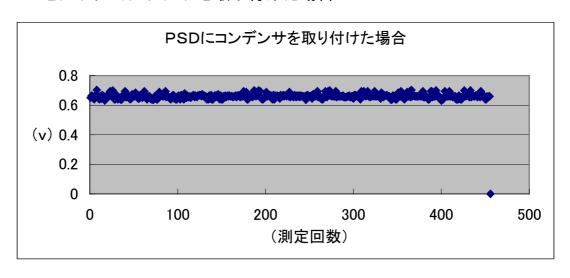


図 4-1-3

結果:この場合 PSD センサの出力電圧値変化はおよそ 0.1 V 以内に収まっている.

<u>第4章 実験 27</u>

### (d)PSD センサにコンデンサを取り付け,平均化を用いた場合

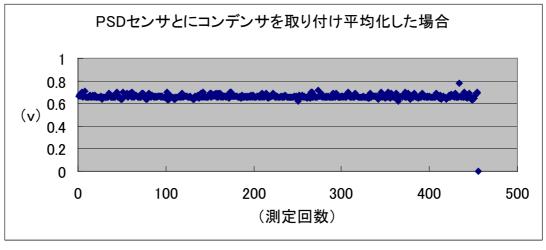


図 4-1-4

結果:この場合,およそ 0.1V 以内に収まってはいるが,たまに 0.1V 以上の電圧値が測定された.

考察: PSD センサのみを使用した場合には 0.1 V 以上の誤差が生じることがあるのでこのパターンは使用できない、続いて、平均を取ったものはソフトにより平均化演算を行うため、センサの増設などにより CPU に対する負荷が大きくなり、リアルタイムでの計測に支障をきたすのでこのような方法を継続的に使用することは不可能である、コンデンサのみ使用した場合とコンデンサと平均化を用いたものを比較すると余り差が見られない、また後者の場合は平均化を用いているので、先の理由から使用する事は不可である、これらのことを踏まえ、本ロボットには PSD センサにコンデンサを取り付けたものを使用する・

<u>第4章 実験</u> 28

## ( ) PSD センサの性能テスト

実験方法:実験は PSD センサの距離に対する電圧値の測定をする.またその際に使用する壁となる材質が変化した場合の電圧値の変化を確認する実験を行う.使用する材質は「木材,紙,布,金属,プラスチック,発泡スチロール」の6種類とする.(これらは一般家庭に良く見られる物として選ぶ)

#### (a)木材を使用した場合

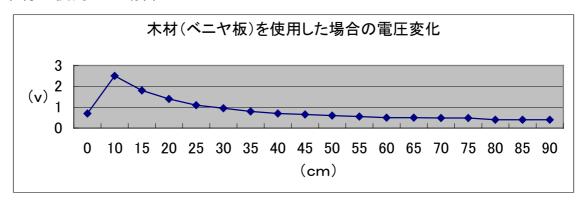


図 4-1-5

## (b)紙(光沢のないもの)を使用した場合

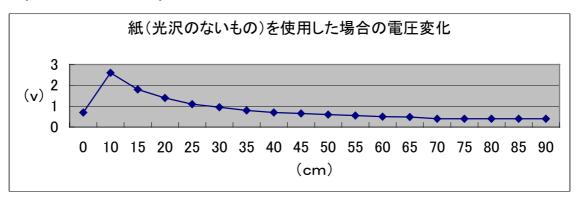


図 4-1-6

#### (c)紙(光沢紙)を使用した場合

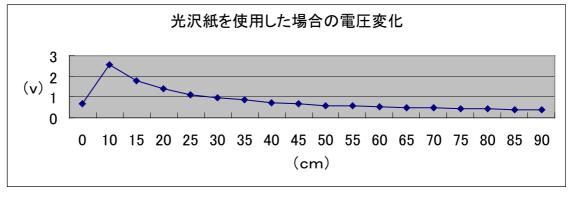


図 4-1-7

## (d)布(模様のあるもの)を使用した場合

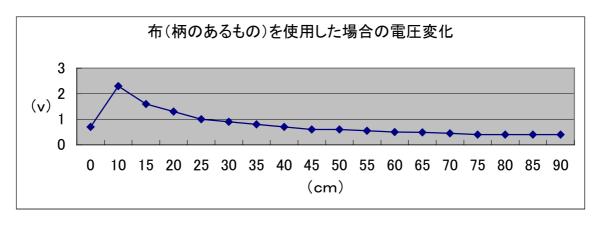


図 4-1-8

### (e)金属(塗料で塗られたもの)を使用した場合

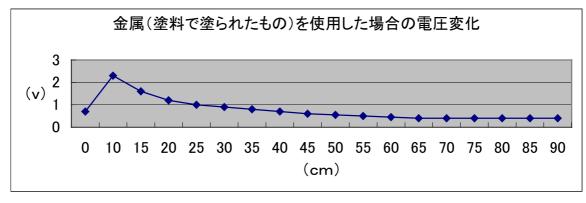


図 4-1-9

## (f)金属(アルミニウム板)を使用した場合

この場合は測定するたびに電圧値が変化してしまい測定するのは不可能であった.

#### (f)プラスチック(透明なもの)を使用した場合

この場合もアルミニウム板と同様に測定するたびに電圧値が変化してしまい 測定するのは不可能であった.

## (g)プラスチック(不透明なもの)を使用した場合

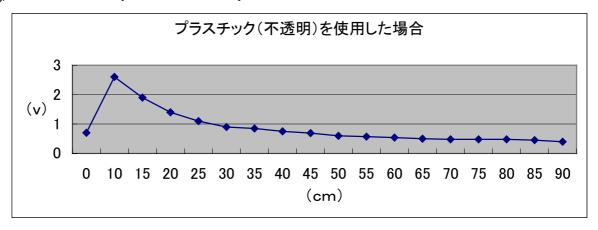


図 4-1-10

## (h)発砲スチロールを使用した場合

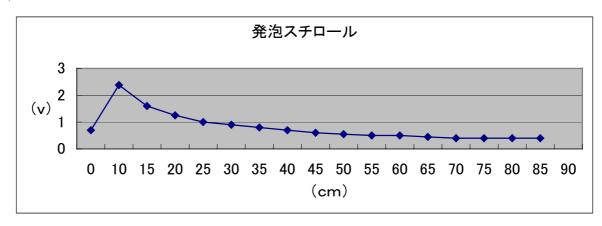


図 4-1-11

## (i)ゴムを使用した場合

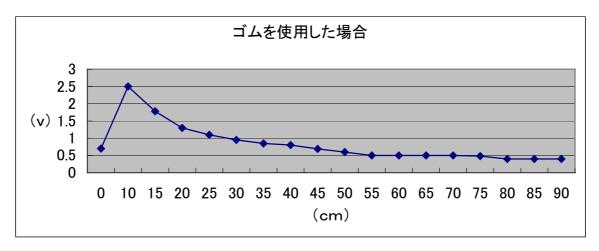
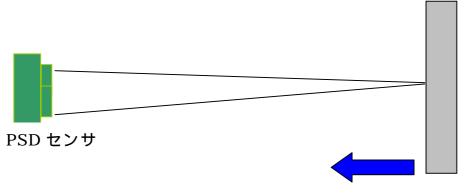


図 4-1-12



障害物を PSD センサに接近させる

図 4-1-13. 計測実験

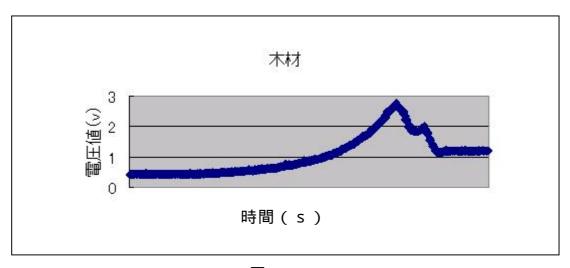


図 4-1-14

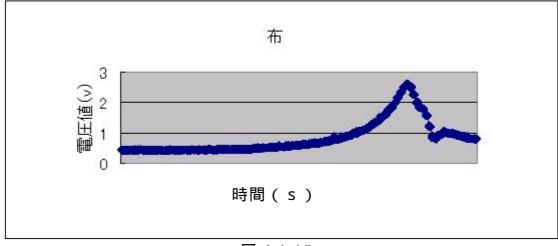


図 4-1-15

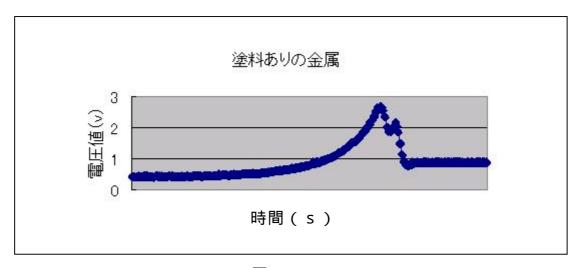


図 4-1-16

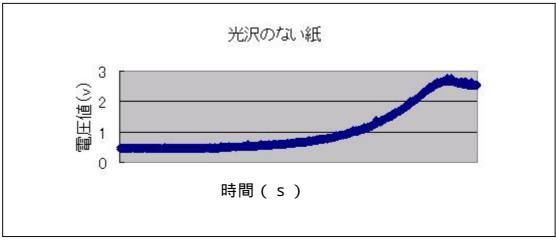


図 4-1-17

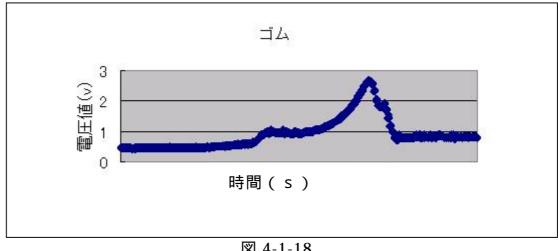


図 4-1-18

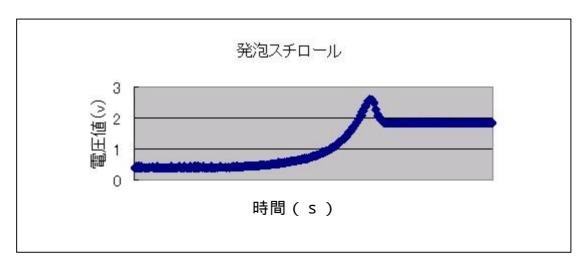


図 4-1-19

## 結果と考察

PSD センサは光沢の強いもの,つまり鏡のようなものや,透明なものに対し ては出力電圧値が不安定となり,正確に測定することが出来ない.しかし,不 透明なものや、光沢のないものであればかなりの精度を持って距離を測定する ことが可能であることが分かった.

( ) 焦電型 MP モーションセンサがどの程度有効であるかについて実験を行う.

実験方法:この実験では,実際に焦電型 MP モーションセンサがどの程度有効であるのかについて,実験を行う.実験方法は,まず初めに,「人」,「物」の区別が実際につくのかどうかについて実験を行い,続いてどの程度の距離と範囲に渡って計測が可能であるかについて実験を行う.実験に用いた物質としては「紙」,「金属」「発泡スチロール」,そして「人」に対する反応実験を行う.この 4 つの実験では,センサに対して下記の2種類のパターンについて検証する.

- 1. 水平方向の動作を行った場合
- 2. 垂直方向の動きを行った場合

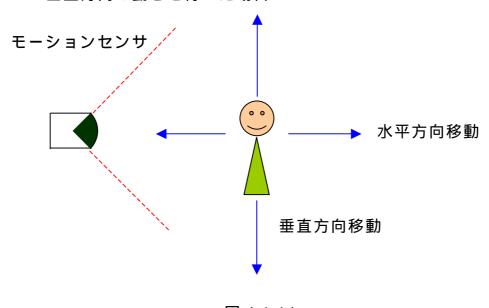


図 4-1-14

結果と考察:人に対するセンサの反応は水平,垂直移動ともに微妙な動作であっても非常に敏感に反応する結果となった.その他の物質の場合はセンサに対して垂直移動を行った場合にはよく反応することが多くあったが,センサに対して水平移動させた場合にはあまり反応することはなく,測定物(動く人,物)と周りの環境との間に高い温度差が生じた場合にセンサは反応する.そして,焦電型 MP モーションセンサは仕様上 5 μ m以上の波長を検知するため,この波長以上の赤外線の変化を検出,そして,温風などによって誤差反応を起すことが多いことがあり,このような環境条件のために今回の実験では不安定な測定結果であった可能性が大きい.しかし,測定範囲に関しては仕様と同様の結果が得られた.これらの結果から,実際に行う人認知実験においては限定された条件のもと実験を行う.

## 4-2 モータ速度回転制御

ここでは4つのモータの速度回転を出来る限り近似させるために,ロボットドライバの TG (擬似タコジェネレータ)機能を用いる.

通常ロボットドライバの調整は手動で行い出来る限り4つのドライバから出力される電圧値が等しくなるようにテスターで測定しながら調整し、各モータの速度回転を制御するが、このような方法には限界があり、10%以上誤差が生じることが多く、人の手によって限りなく4つのモータの速度回転を制御することは非常に困難である.そこで、TG機能を用いることによりソフトウェア上で制御電圧に対してフィードバックを行い、ロボットドライバからモータに対して実際に出力される電圧値の修正を行う.そして、実際にこの方法がどの程度ソフトウェアによるフィードバックが有効であるか検証する必要がある.そこで次のような実験を行う.

実験方法:まず初めに,ロボットが移動している間に,TG からどの程度電圧が出力されているか測定する.この時の制御電圧値は  $0.0V\sim0.5V$  までを使用する.

次に,実際に TG を用いることにより,制御電圧値にフィードバックを与えた場合についての TG からの出力電圧値を測定する.実際の配線については図 4-2-1 に示す.

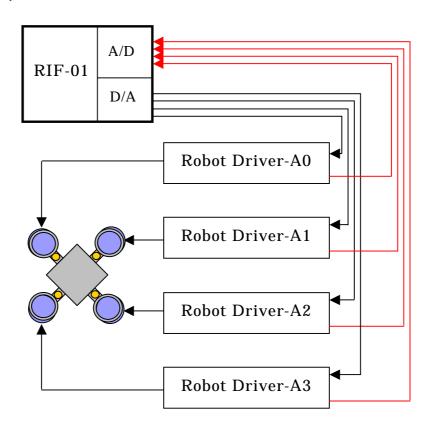


図 4-2-1. 配線概略図

### 実験1:TG からの出力電圧測定

### ( )制御電圧 0.0V の場合

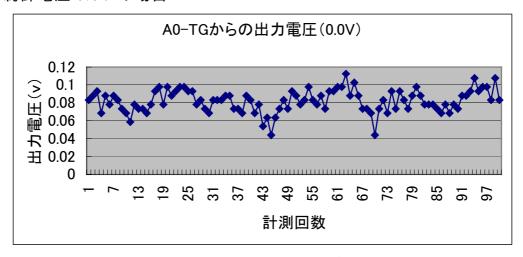


図 4-2-2. 制御電圧 0.0V の場合の A0-TG

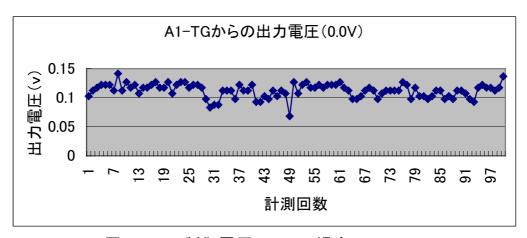


図 4-2-3.制御電圧 0.0V の場合の A1-TG

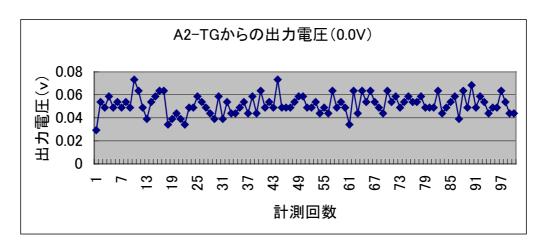


図 4-2-4.制御電圧 0.0V の場合の A2-TG

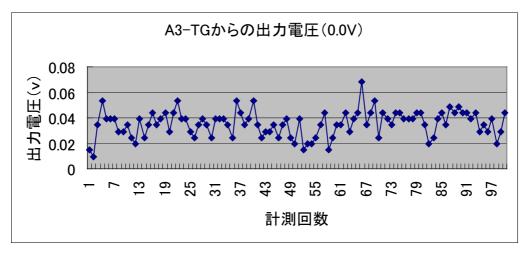


図 4-2-4.制御電圧 0.0V の場合の A3-TG

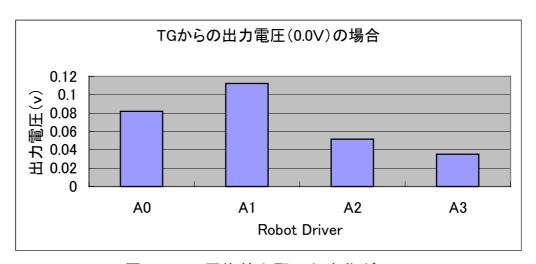


図 4-2-5. 平均値を取った変化グラフ

表 4-2-1.制御電圧が 0.0V の場合の各平均値

	Α0	A1	A2	A3
電圧値	0.081982	0.112207	0.051514	0.035352

#### ( )制御電圧 0.1V の場合

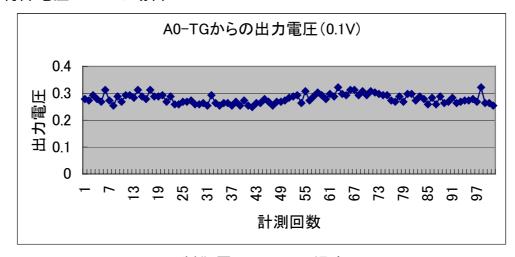


図 4-2-6.制御電圧 0.1V の場合の A0-TG

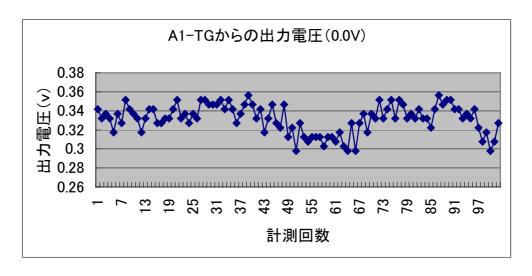


図 4-2-7. 制御電圧 0.1V の場合の A1-TG

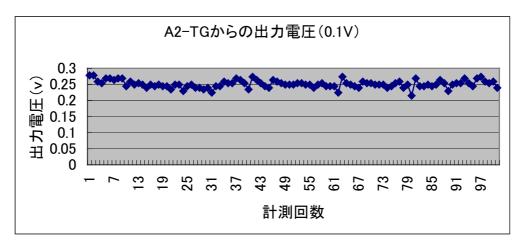


図 4-2-8.制御電圧 0.1V の場合の A2-TG

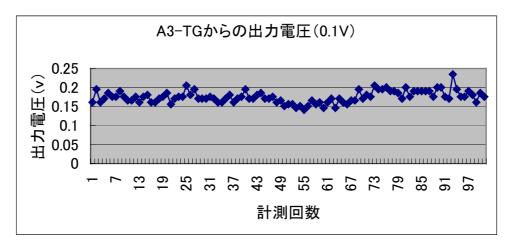


図 4-2-9. 制御電圧 0.1V の場合の A3-TG

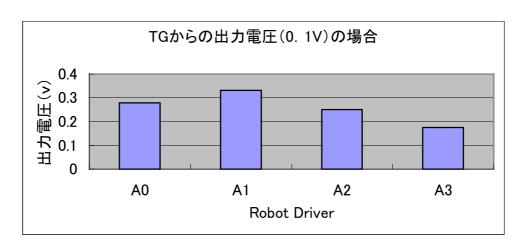


図 4-2-10. 平均値を取った変化グラフ

表 4-2-2.制御電圧が 0.1V の場合の各平均値

	Α0	<b>A</b> 1	A2	A3
出力電圧	2.565253	3.046875	2.303356	1.607777

### ( )制御電圧 0.2V の場合

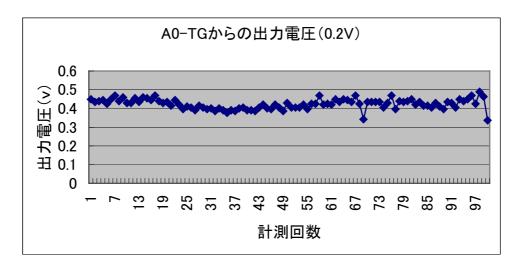


図 4-2-11. 制御電圧 0.2V の場合の A0-TG

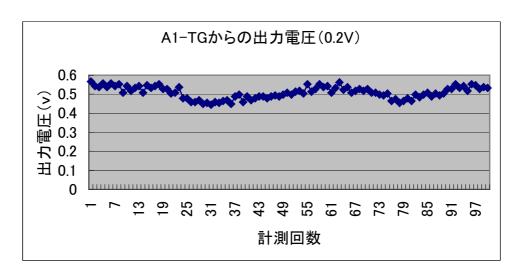


図 4-2-12.制御電圧 0.2V の場合の A1-TG

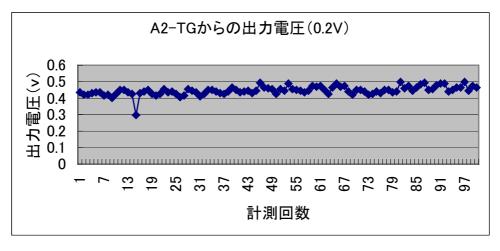


図 4-2-13.制御電圧 0.2V の場合の A2-TG

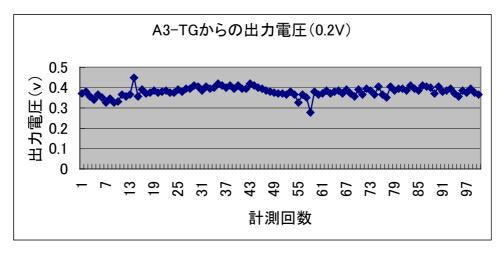


図 4-2-14. 制御電圧 0.2V の場合の A3-TG

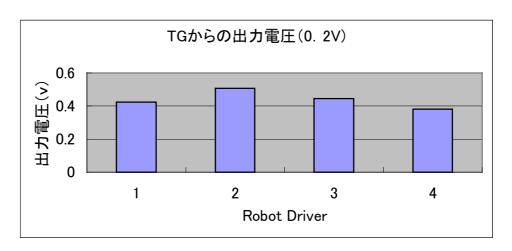


図 4-2-15. 平均値を取った変化グラフ

表 4-2-3.制御電圧が 0.2V の場合の各平均値

	Α0	<b>A</b> 1	A2	A3
出力電圧	0.423242	0.508008	0.444971	0.380615

# ( )制御電圧 0.3V の場合

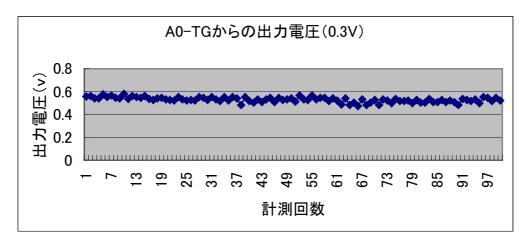


図 4-2-16. 制御電圧 0.3V の場合の A0-TG

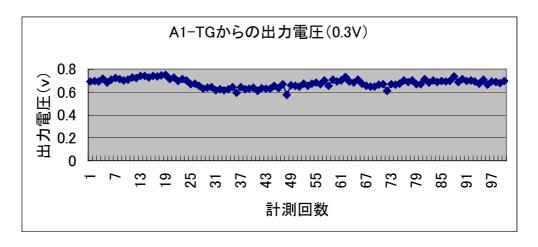


図 4-2-17. 制御電圧 0.3V の場合の A1-TG

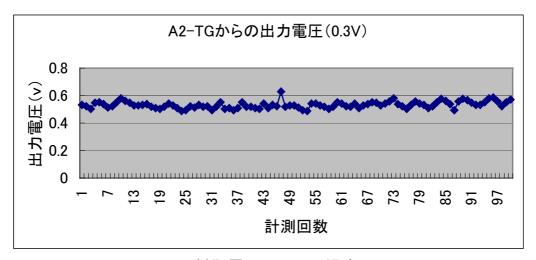


図 4-2-18. 制御電圧 0.3V の場合の A2-TG

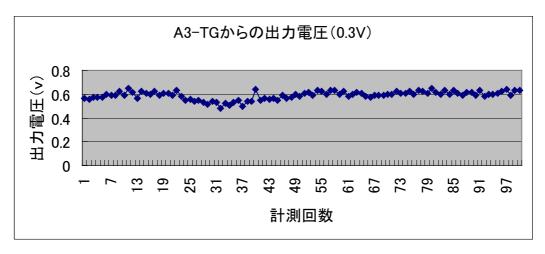


図 4-2-19. 制御電圧 0.3V の場合の A3-TG

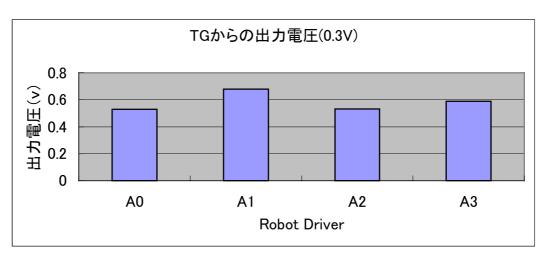


図 4-2-20. 平均値を取った変化グラフ

表 4-2-4.制御電圧が 0.3V の場合の各平均値

	Α0	<b>A</b> 1	A2	A3
出力電圧	0.529541	0.679688	0.531885	0.589258

#### ( )制御電圧 0.4V の場合

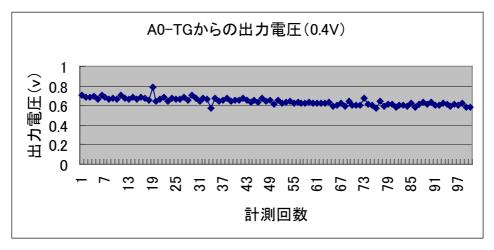


図 4-2-21. 制御電圧 0.4V の場合の A0-TG

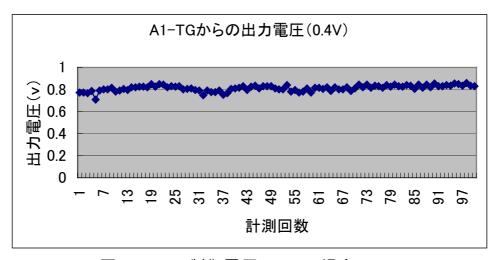


図 4-2-22. 制御電圧 0.4V の場合の A1

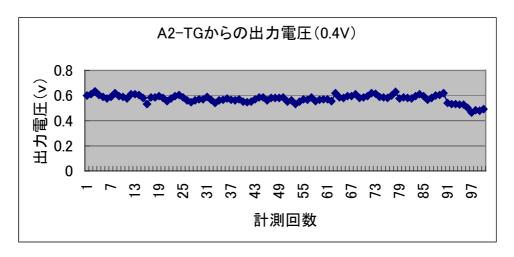


図 4-2-23. 制御電圧 0.4V の場合の A2-TG

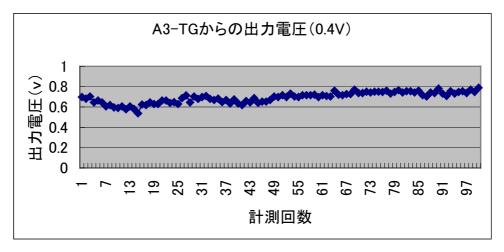


図 4-2-24.制御電圧 0.4V の場合の A3-TG

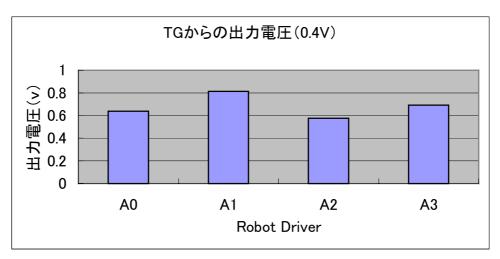


図 4-2-25. 平均値を取った変化グラフ

表 4-2-5.制御電圧が 0.4V の場合の各平均値

	Α0	<b>A</b> 1	A2	A3
出力電圧	0.639404	0.812695	0.574951	0.693408

#### ( )制御電圧 0.5V の場合

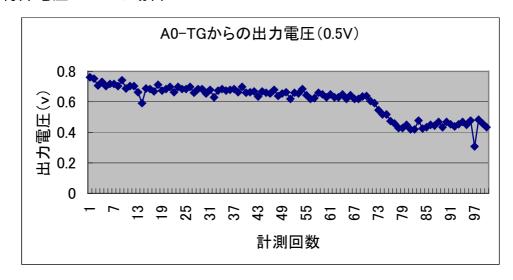


図 4-2-26. 制御電圧 0.5V の場合の A0-TG

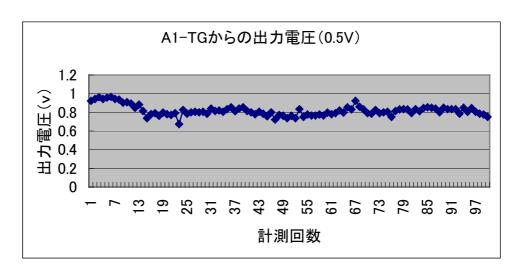


図 4-2-27.制御電圧 0.5V の場合の A1-TG

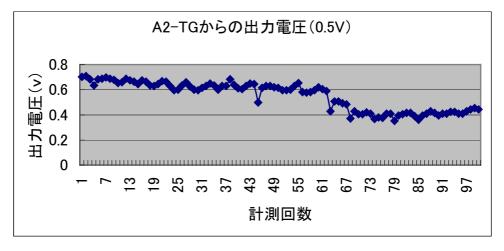


図 4-2-28.制御電圧 0.5V の場合の A2-TG

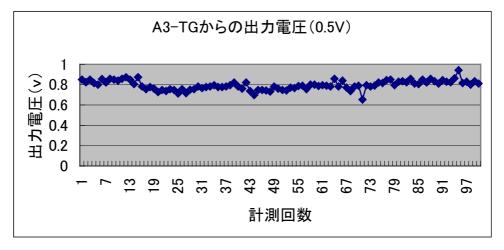


図 4-2-29.制御電圧 0.5V の場合の A3-TG

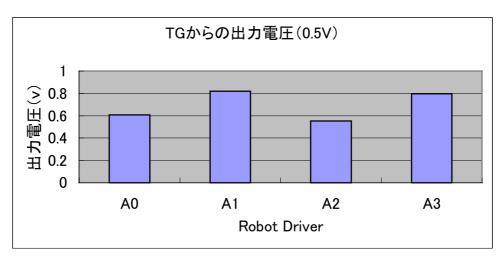


図 4-2-30. 平均値を取った変化グラフ

表 4-2-6.制御電圧が 0.5V の場合の各平均値

	Α0	<b>A</b> 1	A2	A3
出力電圧	0.60791	0.818066	0.5521	0.796436

結果:各平均値を取ったグラフを見ても分かるように制御電圧値とは異なり, 非常に不安定な TG からの出力電圧を計測した .このために先の実験では直線的 な移動を行うことが困難であったと思われる . そこで , TG から出力されてい る電圧値を修正するために制御電圧値に補正を与える .

実験 2: TG からの出力電圧値を測定することにより,それぞれのモータドライバに出力される制御電圧に対しての誤差修正を行うことにより,どの程度走行性能に影響を及ぼすか実験を行う.

### ( )修正後の制御電圧 0.0V の場合

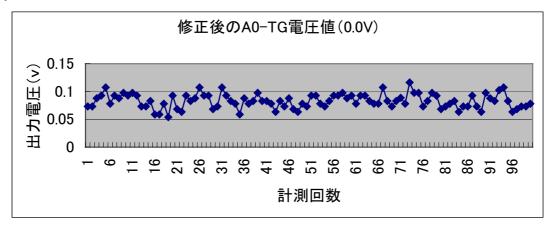


図 4-2-31. 修正後の制御電圧 0.0V の場合の A0-TG

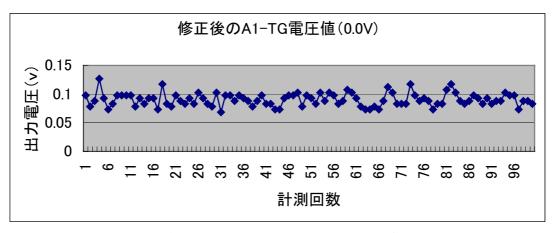


図 4-2-32. 修正後の制御電圧 0.0V の場合の A1-TG



図 4-2-33. 修正後の制御電圧 0.0V の場合の A2-TG

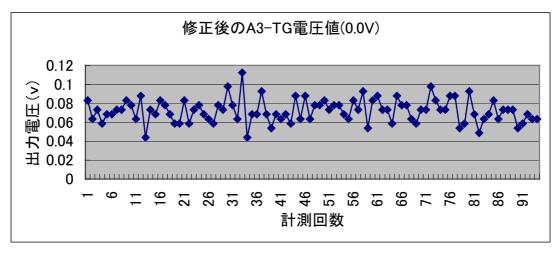


図 4-2-34. 修正後の制御電圧 0.0V の場合の A3-TG

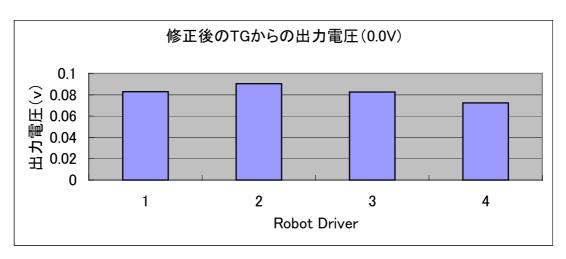


図 4-2-35. 修正後の平均値を取った変化グラフ

表 4-2-7. 修正後の制御電圧が 0.0V の場合の各平均値

	Α0	<b>A</b> 1	A2	А3
修正後の出力電圧	0.082954	0.090283	0.082715	0.072461

## ( )修正後の制御電圧 0.1V の場合

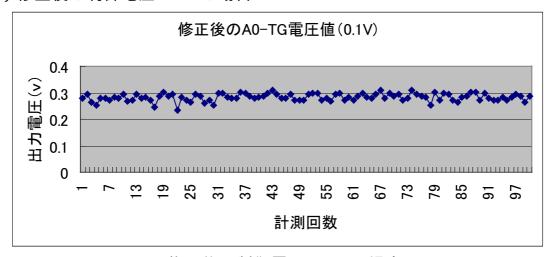


図 4-2-36. 修正後の制御電圧 0.1V の場合の A0-TG

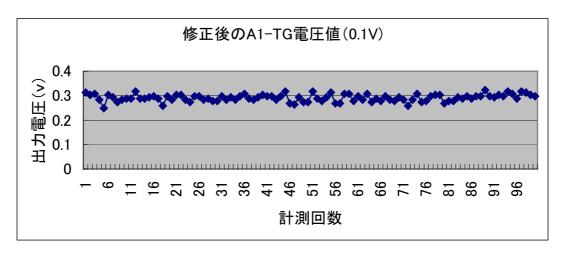


図 4-2-37. 修正後の制御電圧 0.1V の場合の A1-TG

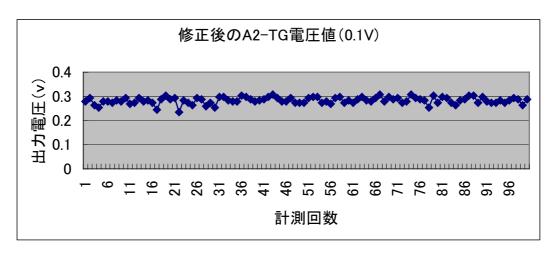


図 4-2-38. 修正後の制御電圧 0.1V の場合の A2-TG

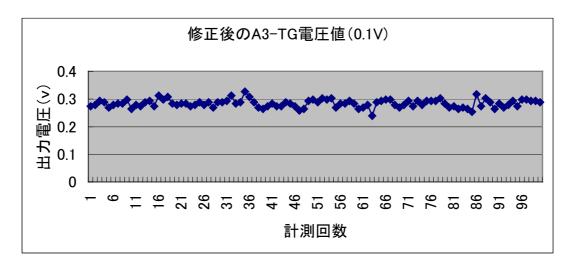


図 4-2-39. 修正後の制御電圧 0.1V の場合の A3-TG

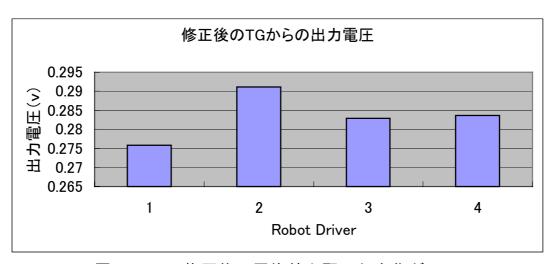


図 4-2-40. 修正後の平均値を取った変化グラフ

表 4-2-8. 修正後の制御電圧が 0.1V の場合の各平均値

	Α0	<b>A</b> 1	A2	A3
修正後の出力電圧	0.275856	0.291113	0.282861	0.283643

#### ( )修正後の制御電圧 0.2V の場合

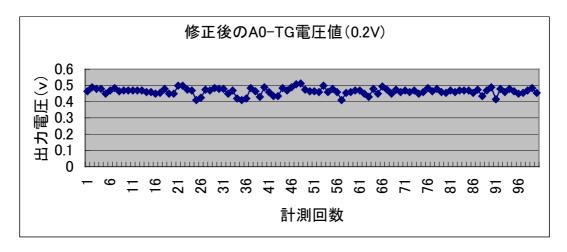


図 4-2-41. 修正後の制御電圧 0.2V の場合の A0-TG

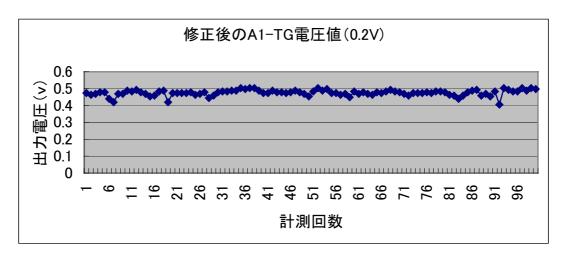


図 4-2-42. 修正後の制御電圧 0.2V の場合の A1-TG

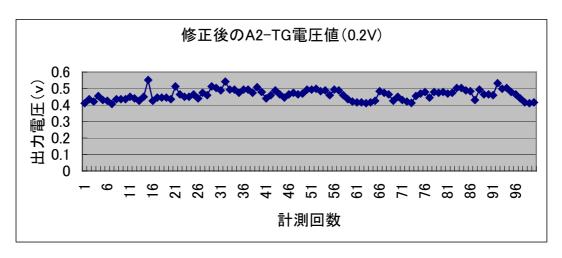


図 4-2-43. 修正後の制御電圧 0.2V の場合の A2-TG

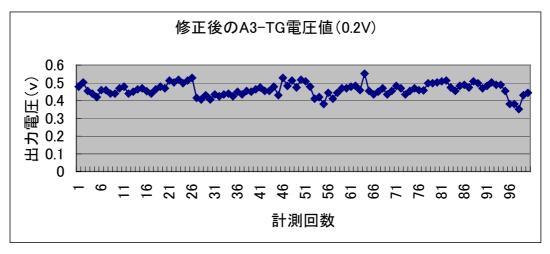


図 4-2-44. 修正後の制御電圧 0.2V の場合の A3-TG

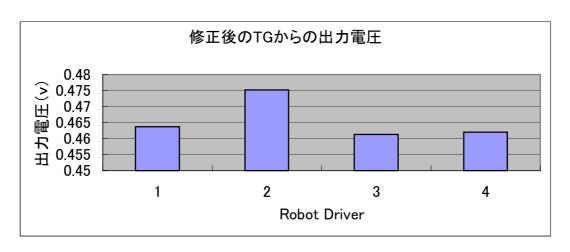


図 4-2-45. 修正後の平均値を取った変化グラフ

表 4-2-9. 修正後の制御電圧が 0.2V の場合の各平均値

	Α0	<b>A</b> 1	A2	А3
修正後の出力電圧	0.463672	0.475244	0.461254	0.462037

<u>第4章 実験 54</u>

#### ( )修正後の制御電圧 0.3V の場合

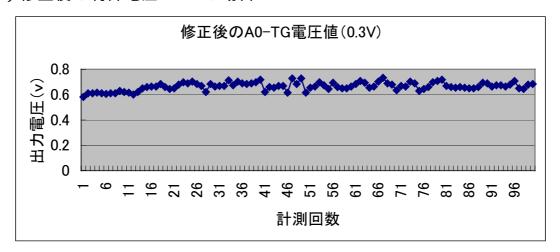


図 4-2-46. 修正後の制御電圧 0.3V の場合の A0-TG

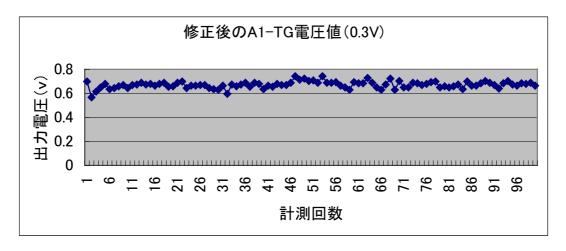


図 4-2-47. 修正後の制御電圧 0.3V の場合の A1-TG

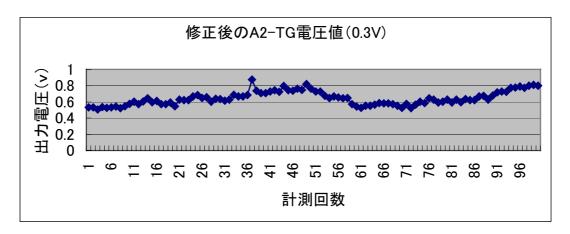


図 4-2-48. 修正後の制御電圧 0.3V の場合の A2-TG

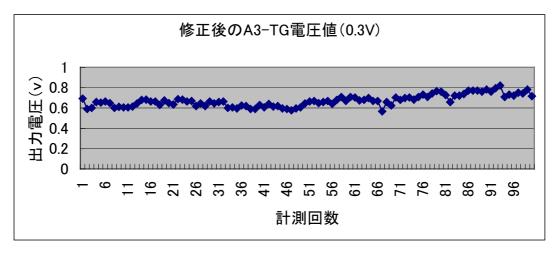


図 4-2-49. 修正後の制御電圧 0.3V の場合の A3-TG

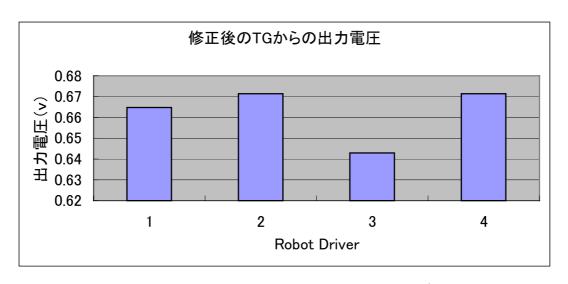


図 4-2-50. 修正後の平均値を取った変化グラフ

表 4-2-10. 修正後の制御電圧が 0.3V の場合の各平均値

	Α0	A1	A2	А3
修正後の出力電圧	0.664746	0.671338	0.642969	0.671436

#### ( )修正後の制御電圧 0.4V の場合

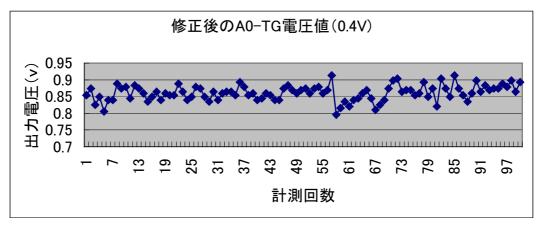


図 4-2-51. 修正後の制御電圧 0.4V の場合の A0-TG

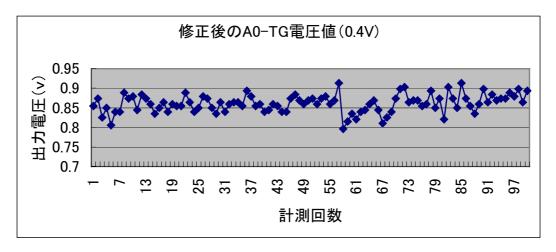


図 4-2-52. 修正後の制御電圧 0.4V の場合の A1-TG

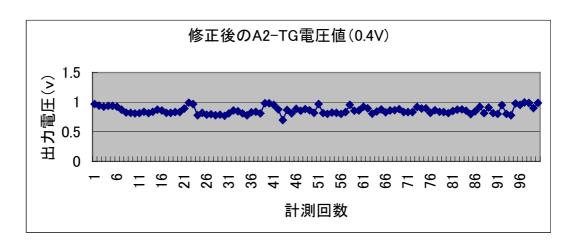


図 4-2-53. 修正後の制御電圧 0.4V の場合の A2-TG

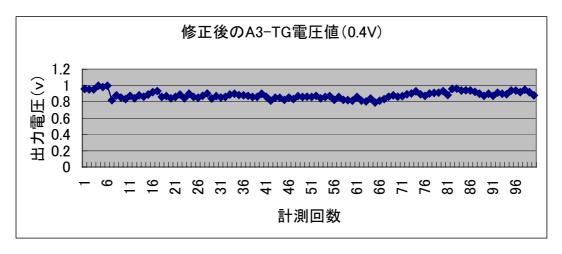


図 4-2-54. 修正後の制御電圧 0.4V の場合の A3-TG

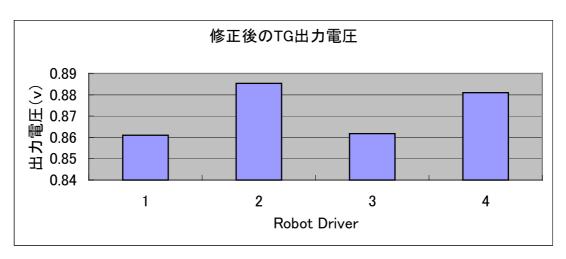


図 4-2-55. 修正後の平均値を取った変化グラフ

表 4-2-11. 修正後の制御電圧が 0.4V の場合の各平均値

	Α0	<b>A</b> 1	A2	А3
修正後の出力電圧	0.861035	0.885285	0.861758	0.880957

#### ( )修正後の制御電圧 0.5V の場合

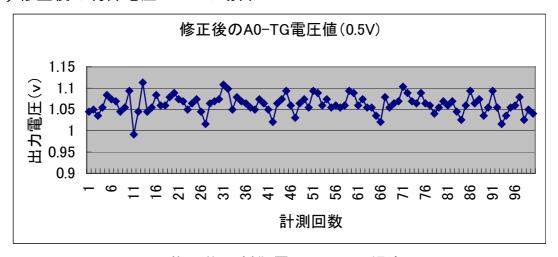


図 4-2-56. 修正後の制御電圧 0.5V の場合の A0-TG



図 4-2-57. 修正後の制御電圧 0.5V の場合の A1-TG

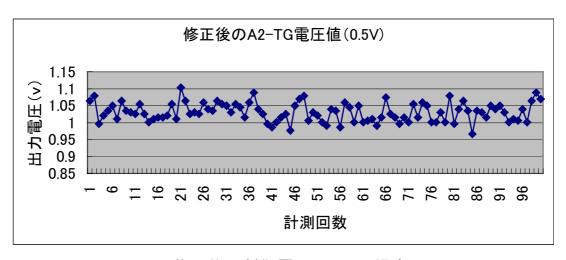


図 4-2-58. 修正後の制御電圧 0.5V の場合の A2-TG

第 4 章 実験 59

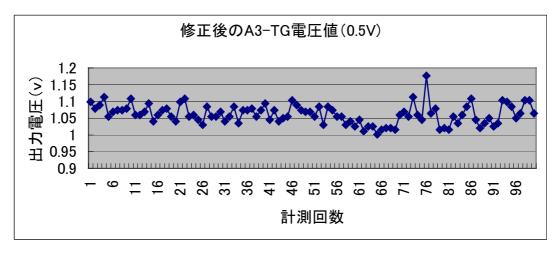


図 4-2-59. 修正後の制御電圧 0.5V の場合の A3-TG

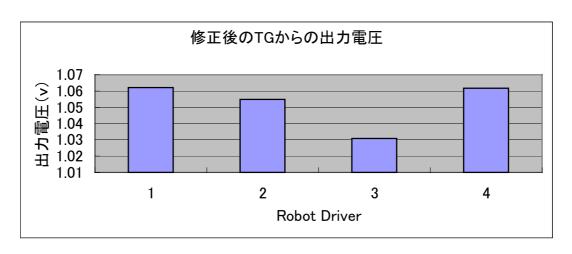


図 4-2-60. 修正後の平均値を取った変化グラフ

表 4-2-12. 修正後の制御電圧が 0.5V の場合の各平均値

	Α0	<b>A</b> 1	A2	А3
修正後の出力電圧	1.062109	1.054785	1.030762	1.061719

実験結果: TG からの修正を制御電圧へ与えなかった場合,各モータへの出力電圧は非常に不安定であり,TG 値測定では 0.1V 以上の誤差が生じる結果となった.しかし,TG からの電圧値を元にし制御電圧にフィードバックをかけた結果,それぞれの Robot Driver の TG 電圧値は比較的安定し,誤差は 0.1V 以内に収まる結果となった.この結果,モータへの出力電圧も安定は安定し,各モータの速度回転は近似されていることが分かる.

## 4.3 全方向移動テスト

実験 1:全方向移動性能についてのテストを行う前に,まずロボットドライバの特性について調べる.

実験方法:ロボットドライバの出力電圧が最大 12V 付近に達するまで制御電圧 を 0.5V ずつ増加させ,制御電圧とロボットドライバの出力電圧を測定 する.本ロボットではモータ供給電圧の最大値を 12V と設定している ためである.

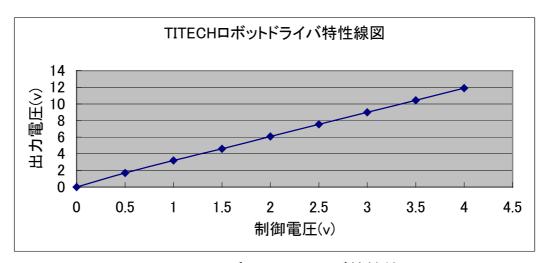


図 4-3-1. ロボットドライバ特性線図

制御電圧(v)	0	0.5	1	1.5	2	2.5	3	3.5	4
出力電圧(v)	0	1.7	3.2	4.6	6.09	7.54	9	10.44	11.9

表 4-3-1. 数值表

続いて図 4-3-1 で計測した出力電圧を 0.5V ずつ変化を与えた制御電圧前後に対する出力電圧の差分を図 4-3-2 に示す.

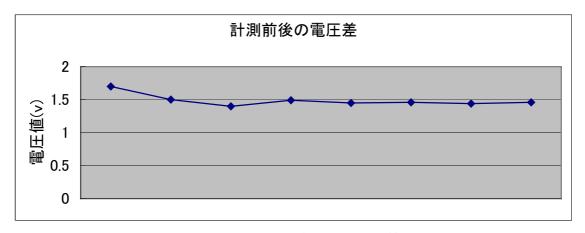


図 4-3-2. 計測前後の電圧差

丰	1 2 2	±_	L SHI	<del>-:</del>	炒	$\boldsymbol{\Phi}$	雷	$\overline{}$	¥
বহ	4-3-2	<u> </u>	測	問り	1攵	v	电	ഥ	左

計測前後差(v) 1.7	1.5	1.4	1.49	1.45	1.46	1.44	1.46
--------------	-----	-----	------	------	------	------	------

結果:図 4-3-1 を見て分かるように比較的に電圧値は比例しており,図 4-3-2 で示された計測前後の出力電圧を見てもある程度安定していることが分かる.この差分に関して平均をとると,約 1.48V となっており,制御電圧に対して約 3 倍程度の出力電圧となっていることが分かる.本ロボットではモータの定格電圧は 32V となっているが,今回モータを駆動させるための供給電源は最大で 12V としているので,制御電圧は最大 4V までとする.

実験 2:実際に与えた制御電圧値の変化に伴った移動速度計測を行う.

実験方法: 図 4-3-2 に示したように ,計測方法は電圧値を 0.5V~3V まで 0.5V 刻みで変化させたときに 1m 進むのにどの程度時間がかかるか ,それぞれ 5 回ずつ計測を行う .

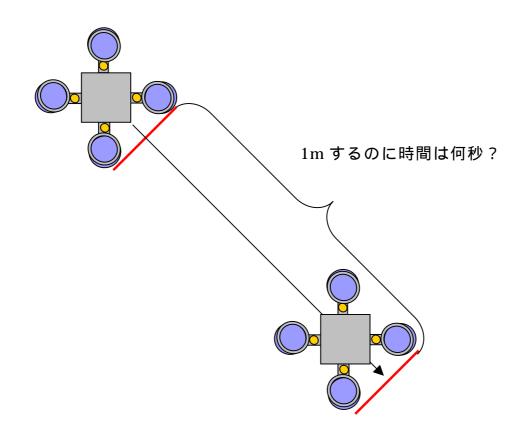


図 4-3-3. 速度計測方法

<u>第4章 実験</u> 62

## ( )制御電圧 0.5V 与えた場合の速度計測

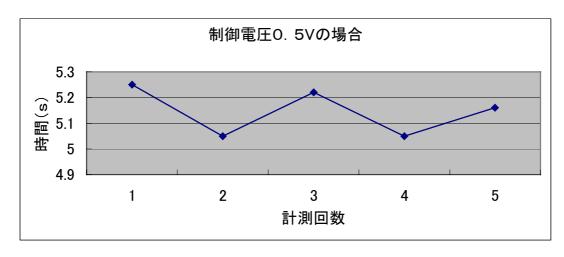


図 4-3-4. 制御電圧 0.5V で 1m 移動するのにかかった時間

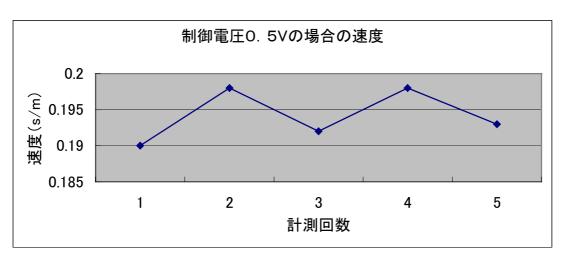


図 4-3-5.制御電圧 0.5V の場合の速度

表 4-3-3. 時間と速度

計測回数	1	2	3	4	5	平均
時間(s)	5.25	5.05	5.22	5.05	5.16	5.146
速度(m/s)	0.19	0.198	0.192	0.198	0.193	0.1942

## ( )制御電圧 1.0V の場合の速度計測

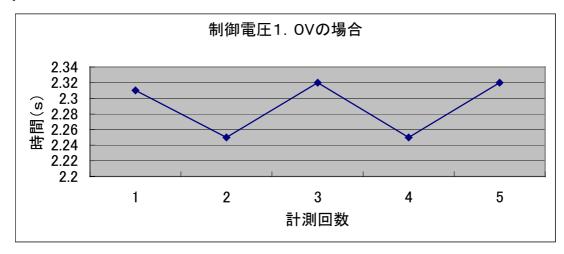


図 4-3-6. 制御電圧 1.0V で 1m 移動するのにかかった時間



図 4-3-7. 制御電圧 0.5V の場合の速度

表 4-3-4. 時間と速度

計測回数	1	2	3	4	5	平均
時間(s)	2.31	2.25	2.32	2.25	2.32	2.29
速度(m/s)	0.433	0.392	0.431	0.392	0.431	0.4158

<u>第4章 実験</u> 64

## ( )制御電圧 1.5V の場合の速度計測

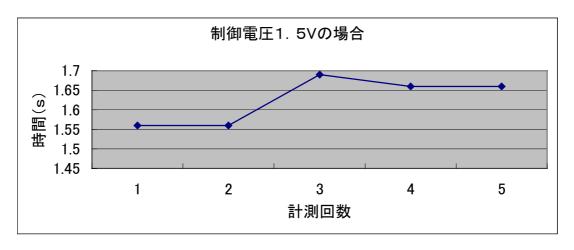


図 4-3-8. 制御電圧 1.5V で 1m 移動するのにかかった時間

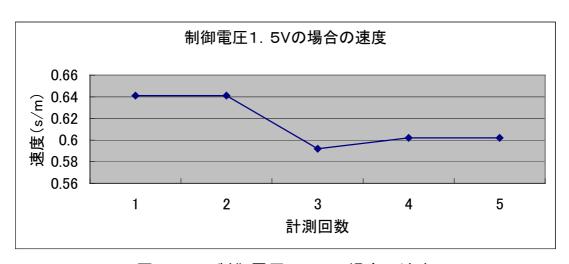


図 4-3-9. 制御電圧 1.5V の場合の速度

表 4-3-5. 時間と速度

計測回数	1	2	3	4	5	平均
時間(s)	1.56	1.56	1.69	1.66	1.66	1.626
速度(m/s)	0.641	0.641	0.592	0.602	0.602	0.6156

<u>第4章 実験</u> 65

## ( )制御電圧 2.0V の場合の速度計測

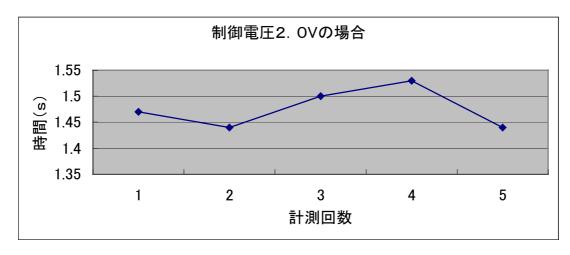


図 4-3-10. 制御電圧 2.0V で 1m 移動するのにかかった時間

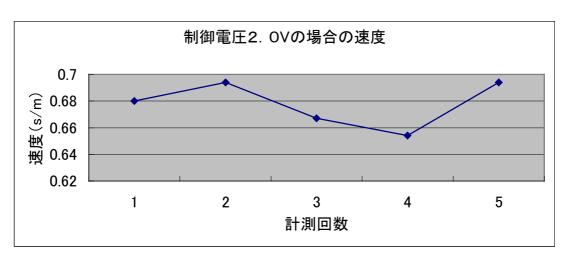


図 4-3-11.制御電圧 2.0V の場合の速度

表 4-3-6. 時間と速度

計測回数	1	2	3	4	5	平均
時間(s)	1.47	1.44	1.5	1.53	1.44	1.476
速度(m/s)	0.68	0.694	0.667	0.654	0.694	0.6778

## ( )制御電圧 2.5V の場合の速度計測

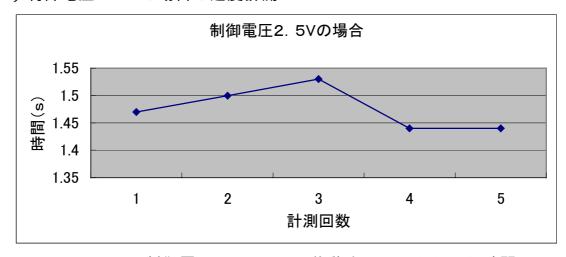


図 4-3-12. 制御電圧 2.5V で 1m 移動するのにかかった時間

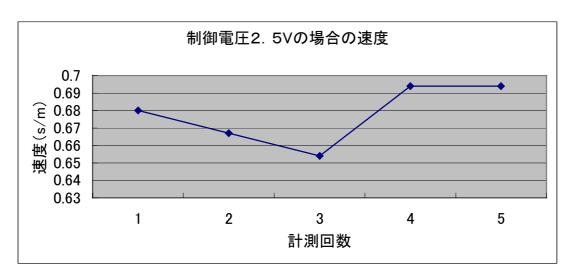


図 4-3-13.制御電圧 2.5V の場合の速度

表 4-3-7. 時間と速度

計測回数	1	2	3	4	5	平均
時間(s)	1.47	1.5	1.53	1.44	1.44	1.476
速度(m/s)	0.68	0.667	0.654	0.694	0.694	0.6778

### ( )制御電圧 3.0V の場合の速度計測

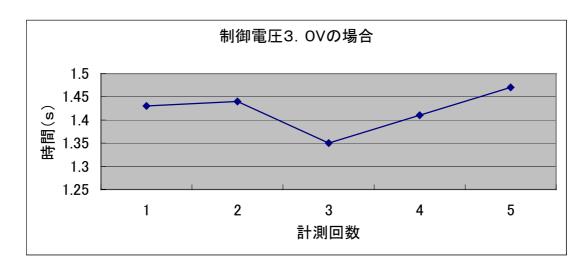


図 4-3-14. 制御電圧 3.0V で 1m 移動するのにかかった時間

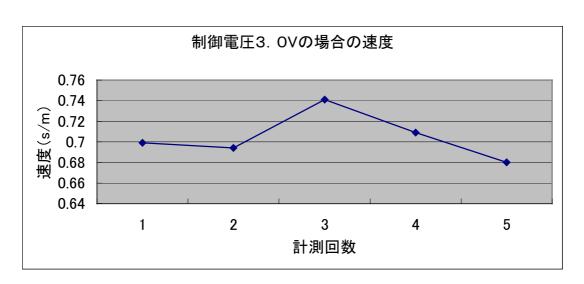


図 4-3-15. 制御電圧 3.0V の場合の速度

表 4-3-8. 時間と速度

計測回数	1	2	3	4	5	平均
時間(s)	1.43	1.44	1.35	1.41	1.47	1.42
速度(m/s)	0.699	0.694	0.741	0.709	0.68	0.7046

これらの測定結果をまとめたものを図 4-3-15 に示す.

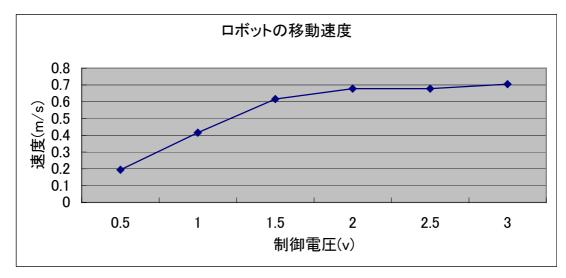


図 4-3-16.制御電圧変化に伴う速度変化

表 4-3-9. 時間と速度

制御電圧(v)	0.5	1	1.5	2	2.5	3
速度(m/s)	0.1942	0.4158	0.6156	0.6778	0.6778	0.7046

結果:制御電圧  $0.5V\sim1.5V$  おいては,速度は制御電圧に比例して増加するが, 2V 以上の制御電圧を与えた場合には,速度に変化が現れなかった。これは,路面と球型アクチュエータに装着されている球との間にすべりが生じたためである.そのため現状では使用する制御電圧を 1.5V までとする.

実験 4 : 全方向移動ロボットの有効性テストについて述べる.

実験方法:全方向移動の有効性を示すために,実際に全方向移動の特性を使用したスムーズな移動を行った場合と,その特性を用いずにスムーズな移動を行わない場合について,移動するためにかかる時間差を走行経路パターン別に測定し,実際にどの程度全方向移動が有効であるかについて述べる.実験で使用するモータへの制御電圧は 0.5Vとし実験を行う.

パターン 1: 簡単な曲がり角をロボットが移動する場合にどの程度時間がかかる か,5回タイムを計測する.

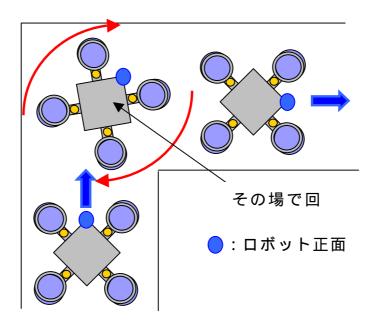


図 4-3-17. 経路パターン 1 (スムーズ移動なし)

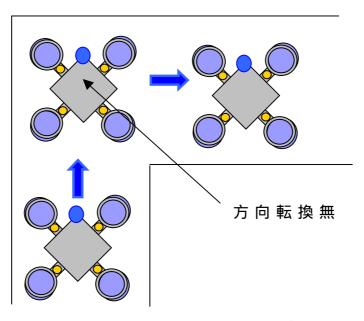


図 4-3-18. 経路パターン 1 (スムーズな移動)



写真 4-3-1. 実験風景

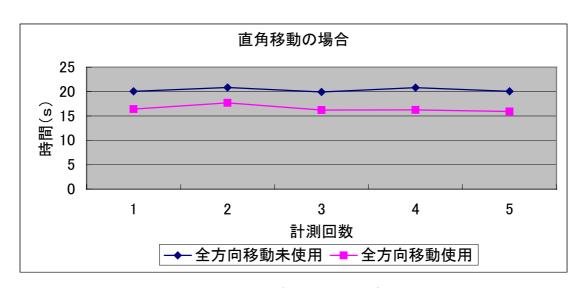


図 4-3-19. 直角移動比較グラフ

	スムーズ移動(s)	スム―ズ移動無し(s)
1	16.41	20.03
2	17.69	20.84
3	16.21	19.93
4	16.25	20.81
5	15.9	20.06

表 4-3-1. 直角移動かかる時間

結果:グラフを見ても分かるように直角移動を行った場合,全方向移動機能の 特性を生かしたスムーズな移動を行った場合と,スムーズな移動を行わな かった場合では,約4秒近くも差が生じる.

パターン 2: 斜め移動を取り入れた経路を移動した場合にどの程度時間差が生じるか,5回タイムを計測し検証する.

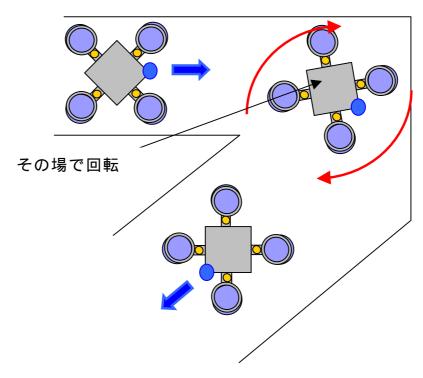


図 4-3-20. 経路パターン 2 (スムーズ移動無し)

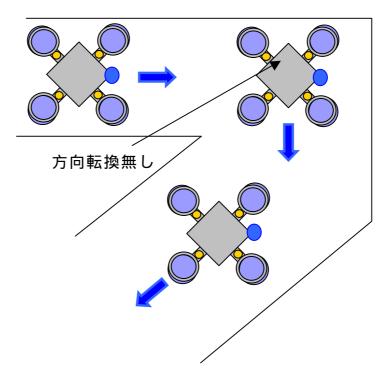


図 4-3-21. 経路パターン 2 (スムーズ移動)

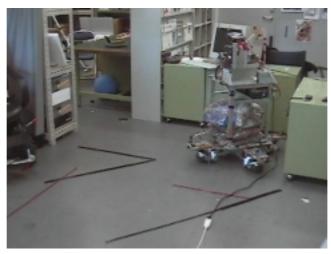


写真 4-3-2. 実験風景 2

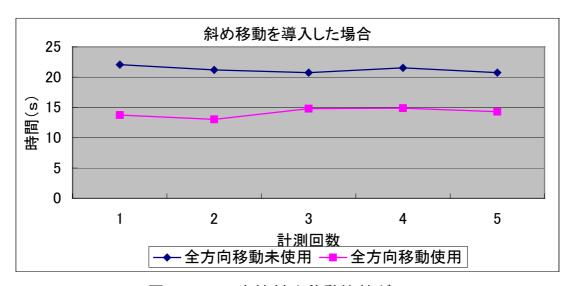


図 4-3-22. 直線斜め移動比較グラフ

	スムーズ移動(s)	スムーズ移動なし(s)
1	13.75	22.06
2	13.03	21.19
3	14.81	20.75
4	14.87	21.54
5	14.31	20.75

表 4-3-2. 直線斜め移動にかかる時間

結果:先の実験と比べ,斜め移動を入れた複雑な経路になると全方向移動を使用した場合と使用しなかった場合とを比べると,約7秒近く差が生じる. つまり方向転換を行う回数が増えると同時に全方向機能の特性を使用したスムーズ移動に比べ,スムーズ移動を使用しない場合は遅れが生じる. <u>第4章 実験</u> 73

パターン 3: 曲線移動を取り入れた経路を移動した場合にどの程度時間がかかるか,5回タイムを計測する.

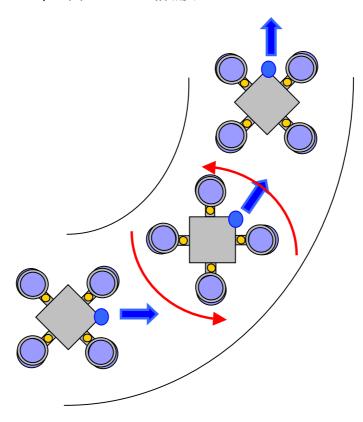


図 4-3-23. 経路パターン 3 (スムーズ移動なし)

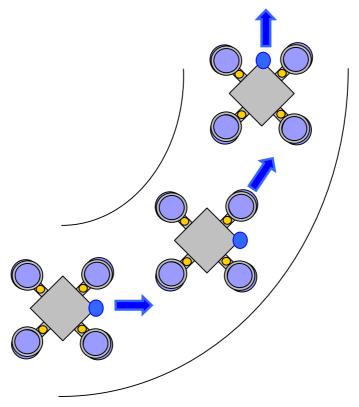


図 4-3-23. 経路パターン 3 (スムーズ移動)

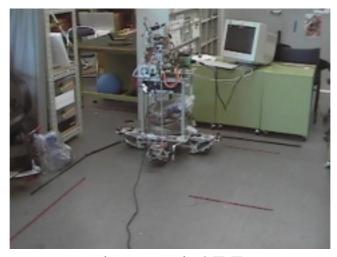


写真 4-3-3. 実験風景 3

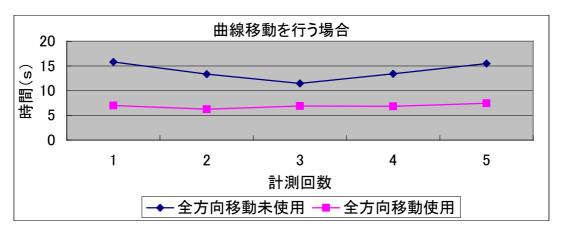


図 4-3-3. 曲線移動比較グラフ

	スムーズ移動(s)	スムーズ移動無し(s)
1	6.99	15.81
2	6.25	13.34
3	6.9	11.47
4	6.84	13.41
5	7.47	15.5

表 4-3-3. 曲線移動にかかる時間

結果と考察:曲線移動においても,パターン1,2と同様に全方向移動を使用した場合とそうでない場合を比較すると約7秒の差が生じる.つまり,ロボットが回転する分だけ余分な時間が生じるので,その分スムーズな移動に比べ遅れが生じる.そして,このような単純な経路のみではなく様々な複雑な経路組み合わさった環境においては移動性能

第 4 章 実験 75

においてかなりの差が生じるであろう.これらの実験から全方向移動機能の特性を生かしたスムーズ移動を行った場合と行わなかった場合とを比べた場合,全方向移動機能を有する移動ロボットは有効であると言え,全方向移動機能を有するロボットと全方向移動機能を有さないロボット間では,その移動性能において大きな差を生じることとなる.本ロボットは一般家庭での使用をもくてきとしているため,どのような狭い経路であってもスムーズに移動することが必要となるため,このような全方向移動機能は非常に重要である.

# 4.4 移動方向予告実験

ここでは,ロボットに移動方向予告機能を搭載することによって人間への安全性に対する情報伝達手段としてどの程度有効であるかについて実験を行う.

実験方法:図 4-4-1 のようにロボットが人に接近し,人に近接した状態で一定のパターンを用いて人の前方を移動する場合ついて,LED による移動方向予告機能を使用した場合と,未使用の場合とについて移動速度を変更した場合における実験に対してアンケートを行う.

#### 実験条件

- 1. 予告が行われてからロボットが移動するまでの間の時間を「予告無し,予告と同時に移動,1秒,2秒間の間を取ってから移動」の4パターンについて実験を行う.
- 2. 速度に関しては「遅い」、「普通」、「速い」の3パターンを用意する.遅い = 0.3V,普通 = 0.5V,速い = 1.5V の制御電圧を用いる.速度と制御電圧 値の関係は表 4-3-9 に示す.
- 3. 人とロボットは向かい合って,人は止まっている状態とする.
- 4 . アンケートの対象は 20 代の男女 5 名
- 5. 「ロボットからの人間に対する情報伝達が安全性にとってどの程度有効であるか?」という質問に対して5段階評定を行う.

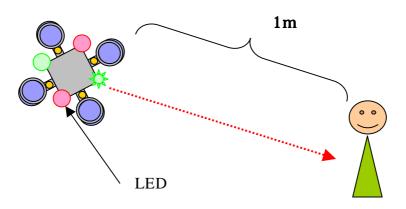


図 4-4-1. 予告動作実

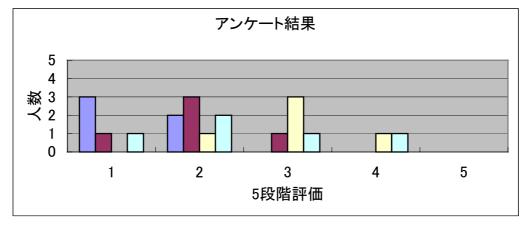


図 4-4-2. 遅い速度の場合

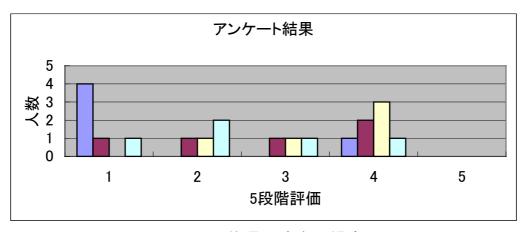


図 4-4-3. 普通の速度の場合

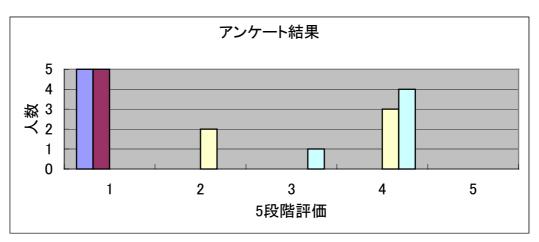


図 4-4-4. 速い速度の場合

: 予告無しで移動 : 1s後に移動

: 予告と同時に移動 : 2s後に移動

<u>第4章 実験 77</u>

結果と考察:アンケート結果から分かるように,遅い速度に関して予告1s程度間があれば,安心感が得られるであろうという結果が得られた.
一段階速度が上がった場合,その間の必要度がましていることが分かる.しかし,速い速度に関しては予告無し,同時では危険であり,最低でも1s程度の予告,あるいは,2s程度の間が必要であるという結果が得られた.全体を通して見ると,速度が上がるにつれ,予告されてからロボットが動くまでの間に時間が必要であるということが分かる.これらの結果から今回行った移動予告機能を行うことによって人に対して安心感を与えることが可能であると言える.

# 4-5 人と物を区別し回避するための実験

まず、実際に様々な経路パターンを実際にどの程度移動することが可能であるかについて実験を行う.

続いて,モーションセンサを使用し,人と物との区別を行いながら回避行動を行うために必要な安全性を得るために,人が安心に感じる移動速度,ロボットと人との距離について実験を行う.

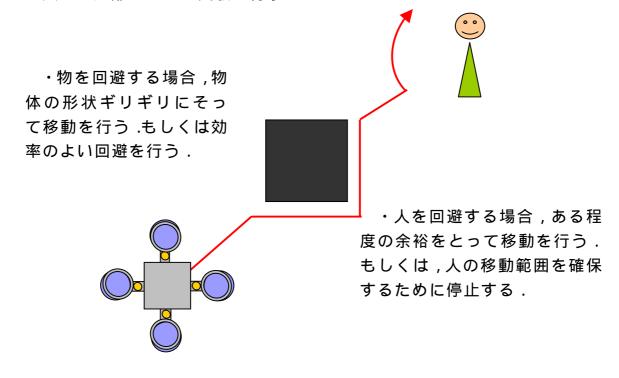


図 4-5-1. 人と物とを区別

#### 実験1: 障害物を回避するにあたってのロボット動作実験

一般家庭のような狭い環境において移動するためには精度よく障害物や経路を認識し、正確に移動することが必要となる、そこで、本ロボットに搭載した PSD センサが実際に経路移動を精度よく行うことが可能であるか検証する必要 <u>第4章</u> 実験 78

がある.そこで,この各 PSD センサがどのように反応することにより,様々な 経路を移動するのかについて提示し、この提示された方法により実際に全方向 機能の特性を生かし、経路移動が可能であるか実験を行う、PSD センサは図 4-5-2 に示されたようにロボット周り8方向に1つずつ配置されている.

PSD センサは図に示されたように円形に近い かたちで8方向に配置されている.

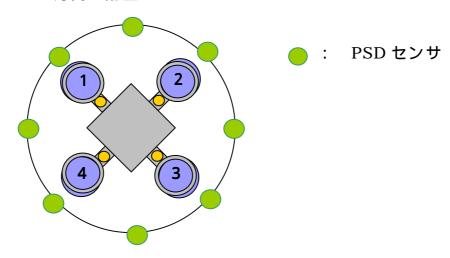
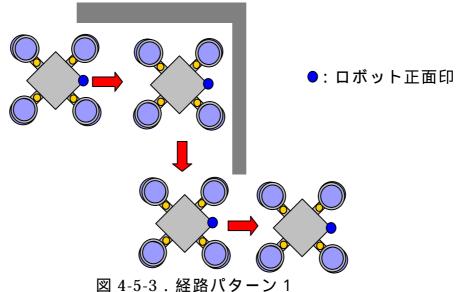


図 4-5-2. PSD 配置概略図

#### 経路パターン 1

経路パターン1においては、ロボットが前進している途中に、正面に図4-5-3 に示したような経路がある場合について実験を行う. PSD センサは図 4-5-4 に 示されたように 6,7,10,12,13番が反応し,その場合は右方向へ移動となり, 障害物の反応がなくなった場合にまた直進移動を行う、そのとき指定された PSD 反応電圧値を表 4-5-1 に示す.



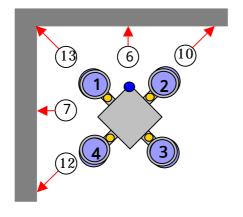


図 4-5-4. PSD 反応パターン 1

# 表 4-5-1. 反応電圧値

PSD6番: 1.2V PSD7番: 1.2V PSD10番: 1.1V PSD12番: 1.1V PSD13番: 1.1V



写真 4-5-1. 実験風景

#### 経路パターン 2

経路パターン 2 においては,ロボットが前進している途中に,正面に図 4-5-5 に示したような経路がある場合について実験を行う. PSD センサは図 4-5-6 に示されたように 6 ,8 ,10 ,11 ,13 番が反応し,その場合は左方向へ移動となり,障害物の反応がなくなった場合はまた直進移動を行う.そのとき指定された

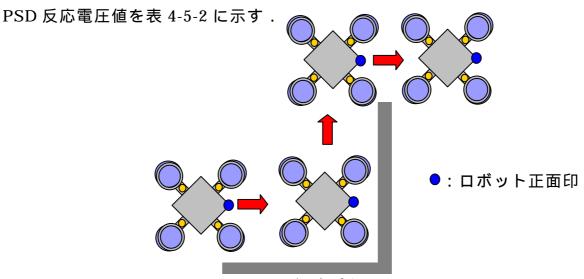


図 4-5-5. 経路パターン 2

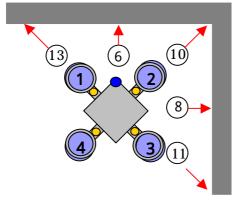


図 4-5-6. PSD 反応パターン

#### 表 4-5-2. 反応電圧値

PSD6 番: 1.2V PSD8 番: 1.2V

PSD10 番:1.1V

PSD11 番:1.1V

PSD13 番:1.1V

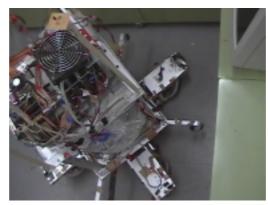
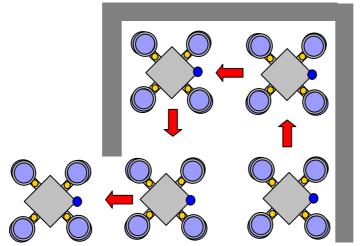


写真 4-5-2. 実験風景

#### 経路パターン3

経路パターン 3 においては ,図 4-5-7 に示されたようなコの字型経路がある場合について実験を行う . PSD センサは図 4-5-4 に示された反応と図 4-5-8 に示されたように 7 , 9 , 11 , 12 , 13 番が反応することにより , その場合は赤矢印で示されたような経路移動を行う . そのときに指定された PSD 反応電圧値を表 4-5-3 に示す .



:ロボット正面印

図 4-5-7. 経路パターン3

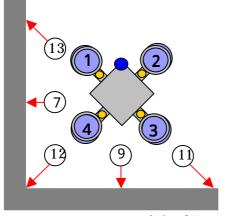


図 4-5-8. PSD 反応パターン 3

### 表 4-5-3. 反応電圧値

PSD7番: 1.2V PSD9番: 1.2V PSD11番: 1.1V PSD12番: 1.1V PSD13番: 1.1V

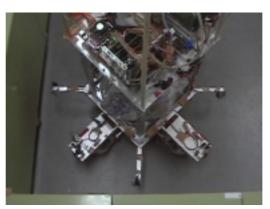


写真 4-5-3. 実験風景

#### 経路パターン 4

経路パターン 4 においては ,図 4-5-9 に示されたようなコの字型経路がある場合について実験を行う . PSD センサは図 4-5-6 に示された反応と図 4-5-10 に示されたように 8 , 9 , 10 , 11 , 12 番が反応することにより , 赤矢印で示されたような経路移動を行う .そのときに指定された PSD 反応電圧値を表 4-5-4 に示す .

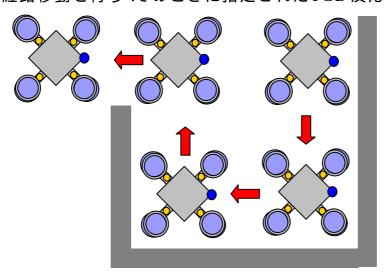


図 4-5-9. 経路パターン 4

:ロボット正面印

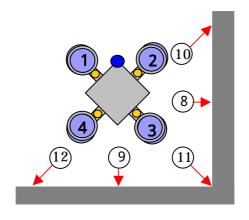


図 4-5-10. PSD 反応パターン 4

# 表 4-5-4. 反応電圧値

PSD8 番:1.2V

PSD9 番:1.2V

PSD10 番: 1.1V PSD11 番: 1.1V

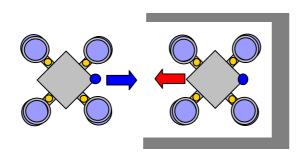
PSD12 番:1.1V



写真 4-5-4. 実験風景

#### 経路パターン5

経路パターン 5 においては、ロボットは青矢印で示されたように前進移動を行っている途中で、図 4-5-11 に示されたようにコの字型に囲まれた空間へ入り込んでしまった場合について実験を行う、その回避方法としては、180 度向きを変更することによってコの字型の空間を抜け出すのではなく、向きを変更することなく赤矢印で示されたように後進することによってこの空間を回避する、PSD センサは図 4-5-12 に示されたように 6,7,8,10,11,12,13番が反応する、そのとき指定された PSD 反応電圧値を表 4-5-5 に示す、



:ロボット正面印

図 4-5-11. 経路パターン 5

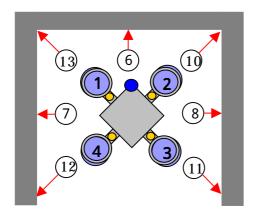


図 4-5-12. PSD 反応パターン 5

## 表 4-5-5. 反応電圧値

PSD6 番;1.2V
PSD7番:1.2V
PSD8 番:1.2V
PSD10 番:1.1V
PSD11 番:1.1V
PSD12 番:1.1V
PSD13 番:1.1V

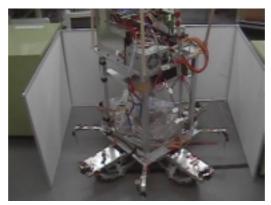


写真 4-5-5. 実験風景

#### 経路パターン6

経路パターン 6 においては、図 4-5-13 に示されたように前進移動途中に狭い経路を信仰しなければならない場合について実験を行う.PSD センサは図 4-5-14 に示されたように 7, 8, 10, 11, 12, 13 番が反応し,その場合は前進方向へ移動となる.そのとき指定された PSD 反応電圧値を表 4-5-6 に示す.

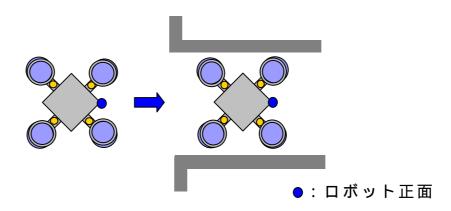


図 4-5-13. 経路パターン 6

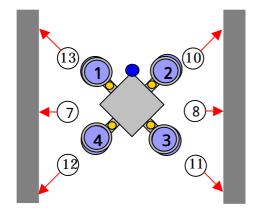


図 4-5-14. PSD 反応パターン 6

#### 表 4-5-6. 反応電圧値

PSD7番: 1.2V PSD8番: 1.2V PSD10番: 1.1V PSD11番: 1.1V PSD12番: 1.1V PSD13番: 1.1V

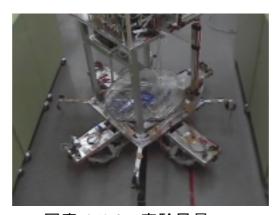


写真 4-5-6. 実験風景

#### 経路パターン7

経路パターン 7 においては、図 4-5-14 に示されたように狭い経路への横移動を行う場合について実験を行う.PSD センサは図 4-5-15 に示されたように 6 , 9 , 10 , 11 , 12 , 13 番が反応し,その場合は右または左方向へ移動となる.そのとき指定された PSD 反応電圧値を表 4-5-7 に示す.

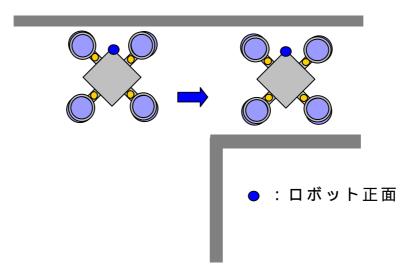


図 4-5-14. 経路パターン7

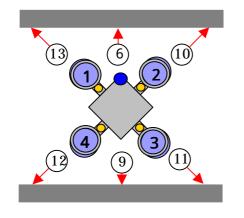


図 4-5-15. PSD 反応パターン 7

### 表 4-5-7. 反応電圧値

PSD6 番: 1.2V PSD9 番: 1.2V PSD10 番: 1.1V PSD11 番: 1.1V PSD12 番: 1.1V PSD13 番: 1.1V

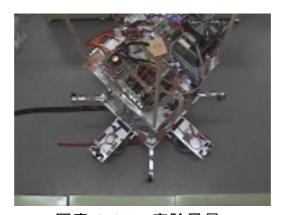


写真 4-5-7. 実験風景

#### 経路パターン8

経路パターン 8 においては,図 4-5-16 に示すように狭い経路移動中に斜め左上移動を行う場合について実験を行う.PSD センサは図 4-5-17 に示されたように 10 , 12 番が反応し,その場合は左上または左下へ移動となる.そのとき指定された PSD 反応電圧値を表 4-5-8 に示す.

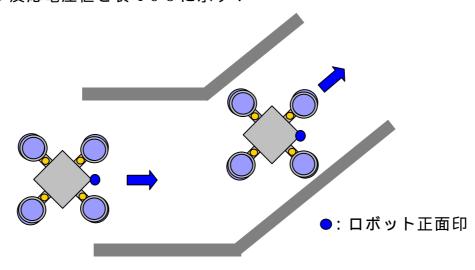


図 4-5-16 経路パターン 8

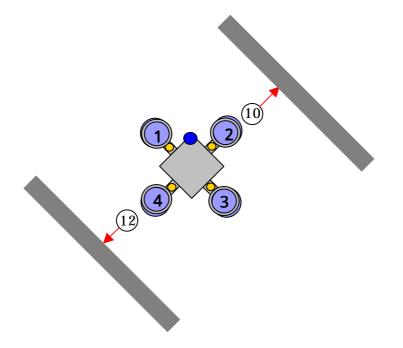


表 4-5-8. 反応電圧値

PSD10 番: 1.1V PSD12 番: 1.1V

図 4-5-17 経路パターン 8



写真 4-5-8. 実験風景

結果と考察:今回行った9つの経路回避パターンを行った結果比較的スムーズに回避移動を行うことが出来た.しかし,PSD センサの個数が8個のために、より複雑な障害物回避を行うには不足である.なぜなら,PSD センサは図 4-5-20 に示されたように広範囲にわたって反応するのではなく、図 4-5-21 に示されたように点のような小さな範囲でのみ障害物を判別するために少しでも障害物から外れた場合には障害物を認識することが出来なくなる.そこで、実際に障害物回避を行うためには複数のセンサを用いなければならない.

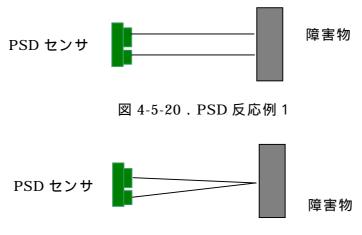


図 4-5-21. PSD 反応例 2

実験 2: 一般家庭のような狭い空間においてはロボットの移動速度やロボットの行動には細心の注意を払わなければならない. さもなければ,ロボットは人に「恐怖感」,「不安感」を与え,あるいは「けが」をさせるなど,人にとってマイナス要因を与えるだけの機械となってしまう. 初めに図 4-5-22 に示すように,どの程度の移動速度が人にとって安心できるかについて実験を行う.

続いて,図 4-5-24 に示すように,ロボットと人との間の距離を 50 cm間隔で  $50 \text{cm} \sim 2 \text{m}$  の距離をとり,ロボットが人を感知し動作を起すまでの間にどの程度距離をおけば,人にとって安心してロボットと空間を共有し生活を行うことが出来るか実験を行う.

( )図 4-5-22 に示されたようにロボットが人に向かって移動してくる場合のロボットの適切な移動速度を知るために,実際に人にアンケートを行う.アンケート結果は図 4-5-23 に示す.電圧値にする速度は表 4-3-9 に示す.

#### 実験条件

- 移動速度:0.3V,0.4V,0.5V,0.6V,0.7V について実験を行う。
- 2 . 移動空間は 4×4×4m空間を想定する .
- 3 . アンケートの対象は 20 代男女 5 名
- 4. 評価は5段階評価
- 5 . ロボットと向かい合ったまま止まっている人に対してロボットから移動を行う .



図 4-5-22

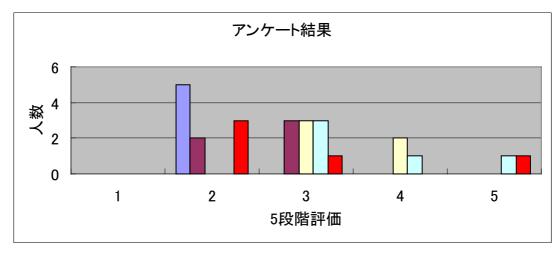


図 4-5-23. 安心出来る移動速度

🔲 : 制御電圧 🔙 : 制御電圧

: 制御電圧 : 制御電圧

結果と考察: 図 4-5-23 から  $0.5V\sim0.6V$  程度の電圧値に対する速度が適度であるという結果が得られた.しかし,今回用いた条件以外にも「ロボットの大きさに変化がある場合」、「 $4\times4\times4$  以上に大きな空間,あるいは狭い空間」、「ロボットが作業を行いながら移動する場合」、「高齢者や障害者に対してのアンケート」など様々な条件に関しても検証する必要がある.

今回はこの結果を元に「人とロボットの距離」に関する人の安心感に対する 実験を行う.

( )図 4-5-12 に示すように,人が止まっているロボットに接近した場合,どの程度の距離でロボットが人を認識し回避行動を行えば,人は安心してロボットと同じ空間を共有することが可能であるのかについてのアンケート結果を図4-5-13 に示す.

#### 実験条件

- 1. 距離:0.5m,1m,1.5m,2m
- 2. 制御電圧: 0.5V
- 3 . 移動空間は 4×4×4m空間を想定する .
- 4 . アンケートの対象は 20 代男女 5 名
- 5. 評価は5段階評価
- 6. 止まっているロボットに対して人から歩み寄る.

第 4 章 実験 89

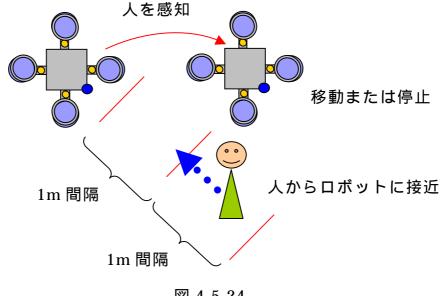


図 4-5-24

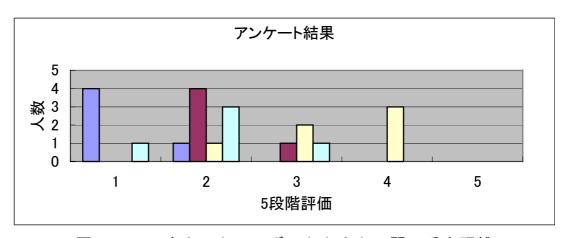


図 4-5-25. 安心できるロボットと人との間の反応距離

: 距離0.5m : 距離1.5m : 距離1m : 距離2m

結果と考察:実験の結果,ロボットの制御電圧を0.5Vとした速度を用いた場合,人が安心できる距離として,図4-5-25のアンケート結果から分かるようにおよそ1-1.5m程度の距離が必要という結果が得られた。今回は制御電圧を0.5Vとした速度を用いた実験を行ったが,安心できる距離と速度の関係については十分に考慮することが必要であり,ロボットの動作や環境に対する条件も様々な場合を考慮する必要がある。今回行った実験条件では,人が安心してロボットと共存するために必要な条件として,ロボットの最適移動速度は制御電圧0.5V,その時の移動方向予告に必要な時間としては1-2s程度,人とロボットとの間に必要な距離はおよそ1-1.5mが必要という結果となった。

# 第5章 結論

#### 5.1 まとめ

今回 ,健康増進を目的とした全方向自律移動ロボットのプラットホームを確立するための様々な検討に基づく基礎実験を行った .

第 1 章ではロボットの歴史,産業用ロボットと家庭用ロボットの違いについて述べた上で,本研究の意義を説明した.

第2章では、家庭環境におけるロボットに必要な機能を考察した.家庭用ロボットは人間と同じ空間で活動するので、産業用ロボットと異なり、安全性を確保しなければならない.安全性を向上するための手法として、ロボットの動きを人間に提示する簡単且つ有効な方法を提案した.また、物のような障害物を回避するだけではなく、人間の動いている動的環境においてはスムーズに移動機能も必須であることを指摘して、人と物とを識別することが必要であることについて述べた.

第3章では、本ロボットプラットホームの移動機構と制御系を紹介し、その特徴について述べた。使用環境として普通の家庭を想定しているので、従来の2輪駆動式の移動ロボットは回転半径が必要であるので適用できない。そのため、ボール型アクチュエータを用いた全方向移動機構を利用することを提案して、その全方向動作メカニズムについて解析した。つづいて、制御システムの構成を紹介して、制御仕様を示した。

第4章では,実際の家庭を想定した環境で,全方向自律移動ロボットのプラットホームの運動制御に関する基礎実験を行った.具体的にまず,自律性を実現するための各種センサについて確認実験を行った後に,実環境での全方向移動実験により球型アクチュエータの有効性を確認した.さらに安全性を向上するために,次の二つの実験を行った.一つ目は,ロボットから人に対しての情報伝達を行うことにより安全性を向上ことが可能であるか検討した.もし人間が,ロボットが次にどのような行動を行うのかが予測できれば,適切な動作を取れるのでより安全性が確保できると考え,ロボットに LED を付けて,ロボットの移動方向を人間に提示する簡単な方法を採用することにより,提示のある無しの場合の実験結果により,異なる安全性が得られることを示しめした.こつ目は,人と物とを区別するためのセンサを用いることにより,「人を認識した場合」、「物を認識した場合」においてロボットの動作方法に区別をつけることにより,人に対する安全性を確保するためにどのような動作を行う必要があるのか実験を通して検討した.

# 5.2 今後の課題

一つ目の課題としてロボットを自律させるためには,ロボット自信が自分の常に位置を正確に把握しロボットの行動性能を向上させる必要がある.そのためにジャイロセンサ,GPS などを用いた位置確認をロボット自信が行わなければならない.

二つ目の課題として、人と物とを区別するためには焦電型 MP モーションセンサ、PSD センサだけでは不十分であり、今回行った実験条件以上に厳しい条件下においてロボットが自律し、人に対する安全性を確保した上で作業を行うためには、人を区別するために画像処理を併用、そして、物を区別するために新たなるセンサを用いることにより、より正確に人や物を認識しなければならない、また、今回行ったアンケートは 20 代の若者を対象に行ったが、高齢者や障害者に対してもアンケートを行うことにより安全性について再検討しなければならない、そして、ロボットと人との距離だけでなく、ロボットの移動速度、人の移動速度、そしてロボットが作業を行っている場合、人とロボットが非常に接近している状態など様々な条件を考慮した上で安全性について慎重に考慮しなければならない、

三つ目の課題として,現在のシステム構成では非常に電力消費が激しいため, 実走行中において 20 分程度でシステム停止を起す危険性があり,この問題に 関しても見直す必要がある.

これらを踏まえた上で、今後はより一般家庭に近い環境下で自律移動可能な ロボットのプラットホームを作成し、健康増進を目的としたロボットを作成す る必要がある.

# 5.3 各部詳細と制御プログラム

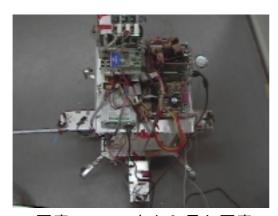


写真 5-3-1. 上から見た写真

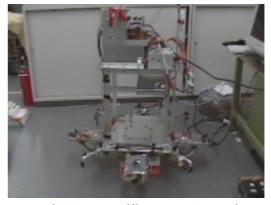
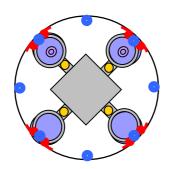


写真 5-3-2. 横から見た写真

第5章 結論 92

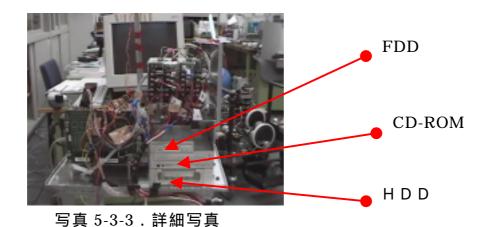


◎ : モーションセンサ配置 2 個を配置

PSD センサ 8 個を配置

◎ :タッチセンサ配置8個を配置

図 5-3-1. 各種センサ配置図



リモコンボード Robot Diver RIF01 ボード

写真 5-3-4. 詳細写真

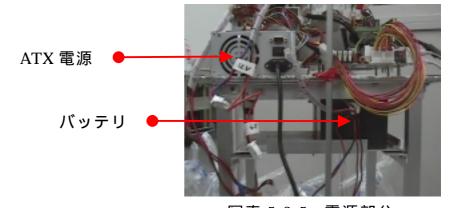


写真 5-3-5. 電源部分

第5章 結論 93

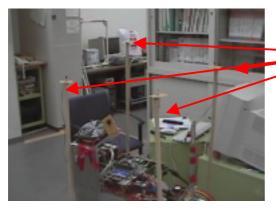
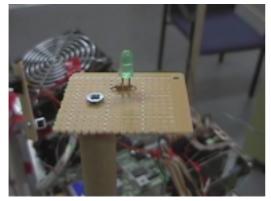


写真 5-3-6. 方向指示 LED

方向指示のための LED4 個



前進後進方向へは緑色の LEDを使用し,左右移動に は赤色の LED を使用する.

写真 5-3-7. 方向指示 LED 拡大写真

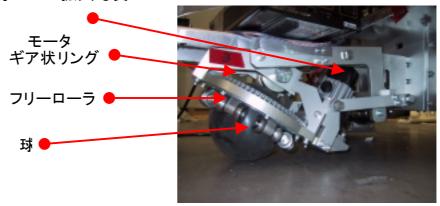


写真 5-3-8. 横から見たアクチュエータ



写真 5-3-9. フリーローラ拡大部分

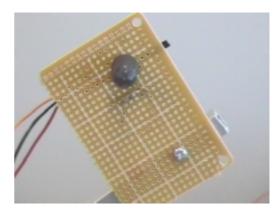


写真 5-3-10. 焦電型モーションセンサ



写真 5-4-11. PSD センサ

続いて今回ロボット制御を行うにあたって,使用したプログラムの一例を示す.

```
#include "conio.h"
#include <stdio.h>
#include <stdib.h>
#include <stdio.H>
#include <windows.h>
#include <process.h>
#include <math.h>
#include "mmsystem.h"
#include<time.h>
```

```
unsigned int data;
unsigned int base_adr;
#define VecToDaddrs (dh, dv) (((dv)+8 * 16 + ((dh) + 8))
#define
        ND 4
#define base_adr 0x100
#define
         cls()
printf("\frac{1}{2}x1b[2J")
void upp1_word_write(void);
int UPP_AD();
void UPP_DAD();
double sonic(double *,double *,double *,double *);
double tuch(double *,double *,double *,double *,double *,double *,double
*.double *):
double kannsuu(double M);
void Automactic(void);
void Move_Type0(void);
void Move_Type1(void);
void Move_Type2(void);
void Move_Type3(void);
void Move_Type4(void);
void Move_Type5(void);
void Move_Type6(void);
void Move_Type7(void);
void Move_Type8(void);
void Move_Type9(void);
void Move_Type10(void);
void Move_Type11(void);
FILE *fp;
int main(int argc, char* argv[])
      double N=0.7, n=0.0, X=0.0;
      unsigned int m_upp;
      double a0,a1,a2,a3,a6,a7,a8,a9,a10,a11,a12,a13;
      int R;
fp = fopen( "open.dat", "w");
              初期化
```

```
regst(base_adr);
_outp(base_adr+2,0x11);
_outp(base_adr,0xff);
printf("Start TypeR");
     do{
           _outpw(base_adr+2,0x03);
           m_upp= _inpw(base_adr);
     }while(m_upp!=1);
     for(R=0;R<=1000;R++)
           printf("TYPE R%d\u00ean",R);
     do{
           _outpw(base_adr+2,0x03);
           m_upp= _inpw(base_adr);
           a6=adinpw(base_adr,6);
           a7=adinpw(base_adr,7);
           a8=adinpw(base_adr,8);
           a9=adinpw(base_adr,9);
           a10=adinpw(base_adr,10);
           a11=adinpw(base_adr,11);
           a12=adinpw(base_adr,12);
           a13=adinpw(base_adr,13);
////// 修正制御
                      ///////TG からの出力電圧/////////
                      a0=adinpw(base_adr,0);
                      a1=320-adinpw(base_adr,1);
                      a2=320-adinpw(base_adr,2);
                      a3=adinpw(base_adr,3);
       リモコン操作
```

```
if(m_upp >= 1 \&\& m_upp <= 256){
                    daoutpw(base_adr,0,N);
                    daoutpw(base_adr,1,-N);
                    daoutpw(base_adr,2,-N);
                    daoutpw(base_adr,3,N);
             }
                                      //////
             else if(m_{upp} == 2){
                    daoutpw(base_adr,0,-N);
                    daoutpw(base_adr,1,N);
                    daoutpw(base_adr,2,N);
                    daoutpw(base_adr,3,-N);
}
             else if(m_{upp} == 4){
                                       //////
                    daoutpw(base_adr,0,-N);
                    daoutpw(base_adr,1,-N);
                    daoutpw(base_adr,2,N);
                    daoutpw(base_adr,3,N);
             }
                                        //////
              else if(m_{upp} == 8){
                    daoutpw(base_adr,0,N);
                    daoutpw(base_adr,1,N);
                    daoutpw(base_adr,2,-N);
                    daoutpw(base_adr,3,-N);
             }
                                        //////
                                                右回り
             else if(m_{upp} == 16){
                    daoutpw(base_adr,0,-N);
                    daoutpw(base_adr,1,-N);
                    daoutpw(base_adr,2,-N);
                    daoutpw(base_adr,3,-N);
             }
                                        //////
                                                 左回り
             else if(m_{upp} == 32){
                    daoutpw(base_adr,0,N);
                    daoutpw(base_adr,1,N);
```

```
daoutpw(base_adr,2,N);
                        daoutpw(base_adr,3,N);
                  }
                  else if(m_{upp} == 64){
                        daoutpw(base_adr,0,N);
                        daoutpw(base_adr,1,0);
                        daoutpw(base_adr,2,-N);
                        daoutpw(base_adr,3,0);
                  }
}
            else{
                                                 //出力 CH1
                  daoutpw(base_adr,0,n);
                                                 //出力 CH2
                  daoutpw(base_adr,1,n);
                  daoutpw(base_adr,2,n);
                                                 //出力 CH3
                  daoutpw(base_adr,3,n);
                                                 //出力 CH4
                  daoutpw(base_adr,4,n);
                                                 //出力 CH1
                                                 //出力 CH2
                  daoutpw(base_adr,5,n);
                                                 //出力 CH3
                  daoutpw(base_adr,6,n);
                  daoutpw(base_adr,7,n);
                                                 //出力 CH4
                  }
}while(m_upp!=128);
                  daoutpw(base_adr,0,0.0);
                                                 //出力 CH1
                                                 //出力 CH2
                  daoutpw(base_adr,1,0.0);
                  daoutpw(base_adr,2,0.0);
                                                 //出力 CH3
                  daoutpw(base_adr,3,0.0);
                                                 //出力 CH4
                  daoutpw(base_adr,4,0.0);
                                                 //出力 CH1
                  daoutpw(base_adr,5,0.0);
                                                 //出力 CH2
                  daoutpw(base_adr,6,0.0);
                                                 //出力 CH3
                                                 //出力 CH4
                  daoutpw(base_adr,7,0.0);
      cls();
      return (0);
}
       /////////
                  自律
                                  void Automactic(void){
      double N=0.4, n=0.3;
```

```
unsigned int m_upp;
    double a6,a7,a8,a9,a10,a11,a12,a13,S0,S1,S2,S3;
    _outpw(base_adr+2,0x03);
    m_upp= _inpw(base_adr);
do{
    _outpw(base_adr+2,0x03);
    m_upp= _inpw(base_adr);
sonic(&S0,&S1,&S2,&S3);
       tuch(&a6,&a7,&a8,&a9,&a10,&a11,&a12,&a13);
if(S0>4.0 \&\& S0<6.0){
             Move_Type3();
        }
        else if(S2>4.0 && S2<6.0){
             Move_Type3();
        }
        else if(S0>4.0 && S0<6.0 && S2>4.0 && S2<6.0){
             Move_Type3();
        }
        else if(a6>2.0){
             Move_Type3();
        }
        else if(a7>2.0){
             Move_Type3();
        }
```

```
else if(a6>2.0 \&\& a7>2.0){
      Move_Type7();
}
else if(a8>2.0){
      Move_Type2();
}
else if(a9>2.0){
      Move_Type4();
}
else if(a8>4.0 && a9>4.0){
       Move_Type8();
}
else if(a10>4.0){
      Move_Type4();
}
else if(a11>4.0){
      Move_Type1();
}
else if(a10>4.0 && a11>4.0){
       Move_Type6();
}
else if(a12>4.0){
      Move_Type1();
}
else if(a13>4.0){
       Move_Type3();
}
else if(a12>4.0 && a13>4.0){
      Move_Type5();
```

```
}
          else{
               Move_Type0();
          }
     }while(m_upp!=2);
     cls();
     printf("オート終了\n");
}
double kannsuu(double M)
     double a6;
     a6=adinpw(base_adr,6);
     M=-(2/11)*a6+(61/110);
}
    Move_Type_define
double N=0.4:
                    通常
                               //
          void Move_Type0(void){
               daoutpw(base_adr,0,N);
               daoutpw(base_adr,1,-N);
               daoutpw(base_adr,2,-N);
               daoutpw(base_adr,3,N);
          }
          //
                    上移動
                               //
          void Move_Type1(void){
               daoutpw(base_adr,0,N);
               daoutpw(base_adr,1,-N);
               daoutpw(base_adr,2,-N);
               daoutpw(base_adr,3,N);
          }
```

```
//
             下移動
                          //
void Move_Type2(void){
      daoutpw(base_adr,0,-N);
      daoutpw(base_adr,1,N);
      daoutpw(base_adr,2,N);
      daoutpw(base_adr,3,-N);
}
//
             右移動
                          //
void Move_Type3(void){
      daoutpw(base_adr,0,N);
      daoutpw(base_adr,1,N);
      daoutpw(base_adr,2,-N);
      daoutpw(base_adr,3,-N);
}
//
                          //
             左移動
void Move_Type4(void){
      daoutpw(base_adr,0,-N);
      daoutpw(base_adr,1,-N);
      daoutpw(base_adr,2,N);
      daoutpw(base_adr,3,N);
}
//
             斜め右上移動 //
void Move_Type5(void){
      daoutpw(base_adr,0,N);
      daoutpw(base_adr,1,0);
      daoutpw(base_adr,2,-N);
      daoutpw(base_adr,3,0);
}
//
             斜め左上移動 //
void Move_Type6(void){
      daoutpw(base_adr,0,0);
      daoutpw(base_adr,1,-N);
      daoutpw(base_adr,2,0);
```

```
daoutpw(base_adr,3,N);
}
      //
            斜め右下移動
                          //
void Move_Type7(void){
      daoutpw(base_adr,0,0);
      daoutpw(base_adr,1,N);
      daoutpw(base_adr,2,0);
      daoutpw(base_adr,3,-N);
}
      //
           斜め左下移動
                          //
void Move_Type8(void){
      daoutpw(base_adr,0,-N);
      daoutpw(base_adr,1,0);
      daoutpw(base_adr,2,N);
      daoutpw(base_adr,3,0);
}
//
             右回転移動
                                 //
void Move_Type9(void){
      daoutpw(base_adr,0,N);
      daoutpw(base_adr,1,N);
      daoutpw(base_adr,2,N);
      daoutpw(base_adr,3,N);
}
//
                                 //
             左回転移動
void Move_Type10(void){
      daoutpw(base_adr,0,-N);
      daoutpw(base_adr,1,-N);
      daoutpw(base_adr,2,-N);
      daoutpw(base_adr,3,-N);
}
//
                    停止
                                 //
void Move_Type11(void){
      daoutpw(base_adr,0,0.0);
```

```
daoutpw(base_adr,1,0.0);
                 daoutpw(base_adr,2,0.0);
                 daoutpw(base_adr,3,0.0);
           }
double sonic(double *S0,double *S1,double *S2,double *S3){
int z,Z=250;
     for(z=0;z<=Z;z++){}
                 daoutpw(base_adr,4,0.0);
                                              //DA4
                 daoutpw(base_adr,5,0.0);
                                              //DA5
                 daoutpw(base_adr,6,0.0);
                                              //DA6
                 *S0=adinpw(base_adr,5);
                                                //AD5
     }
     for(z=0;z<=Z;z++){
                 daoutpw(base_adr,4,5.0);
                                              //DA4
                 daoutpw(base_adr,5,5.0);
                                              //DA5
                 daoutpw(base_adr, 6, 5.0);
                                              //DA6
                 *S1=adinpw(base_adr,5);
                                              //AD5
     }
     for(z=0;z<=Z;z++){
                 daoutpw(base_adr,7,0.0);
                                              //DA7
                 daoutpw(base_adr,8,0.0);
                                              //DA8
                 daoutpw(base_adr,9,0.0);
                                              //DA9
                 *S2=adinpw(base_adr,4);
                                              //AD4
     }
     for(z=0;z<=Z;z++){
                 daoutpw(base_adr, 7, 5.0);
                                              //DA7
                 daoutpw(base_adr, 8, 5.0);
                                              //DA8
                 daoutpw(base_adr, 9, 5.0);
                                              //DA9
                 *S3=adinpw(base_adr,4);
                                              //AD4
     }
```

5 章 結論 105

```
return(0);
}
 *a9,double
double tuch(double *a6,double
                        *a7,double
                                 *a8,
                                      double
*a10,double *a11,double *a12, double *a13){
                                //DA4
         daoutpw(base_adr, 10, 5.0);
         *a6=adinpw(base_adr,6);
         *a7=adinpw(base_adr,7);
         *a8=adinpw(base_adr,8);
         *a9=adinpw(base_adr,9);
         *a10=adinpw(base_adr,10);
         *a11=adinpw(base_adr,11);
         *a12=adinpw(base_adr,12);
         *a13=adinpw(base_adr,13);
return(0);
}
  void regst(unsigned baseadr)
{
    reg_data=0xcf;
                     //マニュアル
                             12P 参照
_outpw(baseadr+4, reg_data); //ジャンパーとの認証使用機能の召還
    erroff=(baseadr+4);
    //return(erroff);
}
int regstoutpw(unsigned baseadr ,unsigned int regstno)
{
    reg_data=regstno;
    _outpw(baseadr, reg_data);
    return 0:
}
             D/A 出力
```

```
double daoutpw(unsigned baseadr ,int portno, double value )
      double da_data;
      if( (portno >= 0) & (portno <= 15) ){
        da_data = (value + 10.0) / 20.0 * 4096.0; // 2048 =
                                                         0v
        if (da_data <
                         0) da_data =
                                         0;
        if (da_data > 4095) da_data = 4095;
         _outpw(baseadr+2, 0xf0+portno);
        _outpw(baseadr, (unsigned)da_data);
      }
else{
             printf("チャンネル Era¥n");
      }
return(0);
}
A/D インプット
                                               double adinpw(unsigned int baseadr, unsigned int portno)
double
             volt;
int
             ad_read_data;
unsigned int ad_write_data;
      ad_write_data = portno*8+5;
      if( (portno >= 0) & (portno <= 15) ) {
           _outpw(baseadr+2, 0x70+portno);
           _outpw(baseadr , portno*8+5);
           for(i=0;i<10; i++) ad_write_data=i;</pre>
           ad_read_data = (signed)_inpw( baseadr );
      if (ad_read_data & 0x0800) ad_read_data |= 0xf000;
        volt = (double)ad_read_data / 2048 * 10;
             return(volt);
      }
      else{
        erron=(1);
        return(erron);
      }
}
```

# 参考文献

- [1] 軣,横井,小森谷,村田:屋内環境用移動プラットホームの開発と姿勢安定化のための移動制御法,ロボティクスメカトロニクス講演会'00 講演論文集,2P2-41-052,2000.
- [2] E. Nakano et al.: 1983 Int. Conf. On Advanced Robotics, pp.277-284, 1983
- [3] Wada Masayoshi , Haruhiko H Asada : Design and Control of a Variable Footprint Mechanism for Holonomic Omni directional Vehicles and its Application to Wheelchairs, IEEE Transaction On Robotics And Automation, vol15, No6
- [4] 淺間一,佐藤雅俊,嘉悦早人,尾崎功一,松元明弘,遠藤勲:3自由度独立駆動型 全方向移動ロボットの開発,日本ロボット学会誌 vol14, No2, pp95-100
- [5] 金沢竜也,山下淳,淺間一,嘉悦早人,遠藤勲,新井民夫,佐藤一省:段差乗り越 え能力を有する全方向移動ロボットの開発,日本ロボット学会誌 vol17,pp.913-914
- [6] 山田隆明,韓福華,渡辺桂,木口量,泉清高:アクテブ双輪キャスタを有する全方向移動ロボットの実験的考察,日本ロボット学会 vol17,pp915-915
- [7] 池浦良淳,河北洋平,水谷一樹:人間に心理的脅威を与えないロボットの動作予告提示法,日本ロボット学会誌 vol17, pp.719-720
- [8] 中村芳晃,森武俊,吉本諭治,佐藤知正:床センサと天井センサの協調による人の 追跡,日本ロボット学会誌 vol15, pp.357-358
- [9] 平澤康宏,山田陽滋,梅谷陽二:心理的安全確保を目的とした人間に恐怖を与えないロボットの運動条件の定量評価,日本ロボット学会誌,vol16,pp.945-946
- [10] 総務省統計局統計センター:人口推移についての調査結果,HP: http://www.stat.go.jp/data/jinsui/index.htm

# 謝辞

本論文は,筆者が高知工科大学基盤工学専攻修士課程において行った研究をまとめた ものであります.

本研究を行うにあたって終始ご指導ご鞭撻をくださった,高知工科大学知能機械システム工学科王碩玉教授に対して深く感謝いたします.そして,本論文を丁寧に査読し,詳細なご指導くださった高知工科大学社会システム工学科藤澤伸光教授に深く感謝いたします.

また,ロボット制御に関して様々なご助言,ご指導下さった高知工科大学知能機械システムコース博士課程の陳貴林氏,修士課程の溝渕宣誠氏,浜口和洋氏,野村和寿氏, 菅野正人氏,隅田由紀氏,伊藤淳氏に深く感謝いたします.

そして,実験補助手伝いなどをして下さった,高知工科大学知能機械システム工学科, 松浦純氏,細川圭介氏,田中秀明氏,長瀬啓志氏,吉本孝司氏に深く御礼申し上げます.

最後に,筆者のために大学院での2年間の研究生活を支えて下さった両親に深く感謝 いたします.

2002年2月18日