## 屋外型警備ロボットの開発

知能機械システム工学科 知能ロボティクス研究室 学籍番号 1020148 細川 圭介 

# 目次

第1章	緒言	
	1. 1	はじめに・・・・・・・・・・・・・・・1
	1. 2	警備ロボット・・・・・・・・・・・・・・・ 2
	1. 3	「屋外型」の必要性・・・・・・・・・・・・・・2
	1. 4	本研究のねらい・・・・・・・・・・・・・・ 3
	1. 5	本文の構成・・・・・・・・・・・・・・3
<i>™</i> . ○ <del>**</del>	- 12	1 0 1+ 0 14% 41
第2草	ロホッ	小の持つ機能 ・・・・・・・・・・・・・・4
	2. 1 遠	隔操縦機能・・・・・・・・・・・・・5
	2. 2 衝	突の回避機能 ・・・・・・・・・・・・・ 7
	2. 3 C	CDカメラによる画像処理機能 ・・・・・・・・・8
第3章	メカニ	ズム・・・・・・・・・・・・・・・9
	3. 1 移	動機構の構成 ・・・・・・・・・・・・・10
	3. 2 実	'装・・・・・・・・・・・・・・・・11
<b>쑠</b> ⊿辛	生は谷門さん	7-1

<u>目次</u> ii

4. 1 制御システムの構成・・・・・・・・・・・・15
4. 2 CPUボード・・・・・・・・・・・・・・17
4. 3 ロボット用インターフェイスボード・・・・・・・・ 18
4. 4 ロボットドライバ・・・・・・・・・・・・・・・ 24
4. 4. 1 TITECH ROBOT DRIVER について・・・・24
4. 4. 2 ロボットドライバの実装 ・・・・・・・・ 28
4. 5 モータ制御・・・・・・・・・・・・・・・32
4.5.1 モータについて・・・・・・・・・・33
4. 5. 2 モータの実装・・・・・・・・・・・35
4. 6 環境認知センサ・・・・・・・・・・・・ 35
4. 6. 1 超音波センサ・・・・・・・・・・ 35
4. 6. 2 超音波センサの実装・・・・・・・・ 38
4. 6. 3 赤外線センサ・・・・・・・・・・ 40
4. 6. 4 赤外線センサの実装・・・・・・・・・41
4. 6. 5 カラートラッキングビジョンPCI・・・・・・ 45
4. 6. 6 カラートラッキングビジョンPCIの実装 ・・・・ 49
4. 7 遠隔操縦・・・・・・・・・・・・・・・53

目次iii4. 7. 1 サーバクライアント方式について・・・・・・ 53

4. 7. 1 7 7.79 1751 751 201C 20 C
・ソケット通信について ・・・・・・・・ 56
・ソケット通信の実装・・・・・・・・・・ 57
4. 7. 2 USBジョイスティックについて・・・・・・・59
·USBについて・・・・・・・・・・60
・USBジョイスティックの実装 ・・・・・・・・ 64
4. 7. 3 映像通信 ・・・・・・・・・・・ 66
・ネットミーティングについて ・・・・・・・ 66
4. 7. 4 WEBカメラのサーボ制御 ・・・・・・・ 67
・実装 ・・・・・・・・・・・・・・70
4.8 電源・・・・・・・・・・・・・・・・71
第5章 まとめ・・・・・・・・・・・・・・・・・・73
5.1 研究成果と考察・・・・・・・・・・・・・・73
5.2 今後の展開・・・・・・・・・・・・・・ 74
参考文献 ・・・・・・・・・・・・・・・・・・・・・ 75
謝辞 ・・・・・・・・・・・・・・・・・・・・・・・76

第1章 緒言 1

## 第1章 緒言

本文は,新しい屋外型警備ロボットについて述べたものである.

### 1.1 はじめに

20世紀初期においてロボットという概念が生まれ,1980年以降,社会は大量消費を前提とした,大量生産を目的とする社会構造へと変化を遂げる.それに伴い産業用ロボットの需要は,ますます増加の一途をたどることとなる.今日の日本においては,世界中から「ロボット大国日本」と呼ばれるように,世界中の産業用ロボットの約6割が設置され,製造の現場における手段の一つとして定着しており,その技術も成熟段階に入ったと言っても過言ではない.こういった背景化において,今後ロボットの需要は,オフィスで働く,高齢者看護の補助などといった人間の仕事の補佐や,危険な場所での作業を人間に代わって行うといった,人間の身近なところで働くものが高くなると考えられる.

平成13年9月11日,アメリカにおいて同時多発テロが起こった.これをうけて,日本国内でも緊急対応措置として空港ほか国内重要施設の警備体制強化を打ち出し,警察庁は米国支援を表明した国の大使館を含め,重点警備対象施設を拡大した,在日米軍基地の警備ができるよう自衛隊法の改正を検討するという声も挙がった.これは,日本もテロの対象になり得ることを想定したからである.国内でも小学校襲撃殺傷事件といった従来の危機管理意識を超える犯罪が増えている.つまり,現在,一層の安全確保が強く求められている.

ここ数年、「ホームセキュリティ」という言葉をよく耳にするようになった.これは、日本警備保障(現在のセコム)が先駆となった家庭用機械警備システムサービスのことである.警備会社というものは、外国では比較的古くから存在し、19世紀中頃のアメリカ、西部開拓時代に列車強盗に対する警備と捜査を行うことを業務としたピンカートン社が警備会社のはじまりといわれている.しかし、日本において初めて専業の警備保障会社が設立されたのは昭和37年であった.そして、昭和39年に行なわれた東京オリンピックにおいて広く世間に知られることとなった.このように、日本において警備会社は、戦後にでてきた歴史の浅い事業である.これは、「水と安全はただ」という根強い国民意識が存在するとともに、かつては、宿直・夜警・守衛・巡視・保安員といった名称で、自社員をもって、自主警備に当たることが多かったからである.また、地域的な防犯力も存在した.地域交流による団結力が不審者の浸入を防いでいたからある.現在は、プライバシーの尊重や核

家族化,女性の社会進出などにより地域的な防犯力は著しく低下した.

警備において,人的警備には以下のような問題点がある. 人命の危険, 人材の教育, 人件費, 労務管理.これらの解決策としてはじめられたのが,「ホームセキュリティ」のような機械警備である.機械警備は,人命を危険にさらすこともなく,長い目で見れば,初期投資とメンテナンスを必要とするだけで,人的警備より安く済む.警備ロボットは,このような機械警備の一つとしてあげられる.

## 1.2 警備ロボット

警備ロボットといえば,最近では総合警備保障の「ガードロボC3型」,富士重工やセントラル警備保障と富士重工の共同開発した「RFS2」などが上げられる.

「ガードロボ C 3型」は,自律移動,自動充電,障害物回避,位置認識,遠隔監視,煙・炎・人体・漏水センサによる異常の検出と通報,同時通話方式の遠隔通話や音声警報といった機能を搭載している。自律移動は,あらかじめロボットに記憶させておく地図によって走行するため,専属のオペレータが常時操作する必要がない。自動充電は,ロボットが充電装置に自動的にドッキングし充電することが可能であり,タイムスケジュールによる自動運用ができる。遠隔監視は,内蔵しているカメラにより現地の情報を画像で確認することで行う。異常検出機能には,遠赤外線センサ・紫外線センサ・近赤外線センサ・光距離センサなどを用いて,人体検出・炎検出・煙検出・漏水検出を行うことができる。

「RFS2」は警備清掃ロボットで,清掃業務と合わせて,警備業務も同時にこなす.このロボットも自律移動,自動充電,障害物回避,遠隔監視,異常の検出と通報ができる.自律走行・障害物回避は,ジャイロによる方向制御,車輪からの走行距離検出,障害物センサにより行う.この「RFS2」も自動充電機能により,24時間休むことなく働くことができる.

## 1.3 「屋外型」の必要性

1.2で挙げた警備ロボットは「屋内型警備ロボット」であり、現在ある警備ロボットと呼ばれるものの多くは、そのほとんどが屋内を巡回・監視するものである。屋外の機械警備には監視カメラ・センサーの設置などがあるが、どうしても死角ができ、カメラやセンサだけでは不審者に対しての対応(追い払う・追跡するといった行動)ができず、建物などに設置された装置では範囲が限定されてしまう。警備員の場合、夜間の警備には危険が伴う、とくに、近年は異常気象ため夏場は夜間だけでなく昼間でも、熱中症といった危

第1章 緒言 3

険もある.よってこれらに替わる屋外の警備手段として屋外を巡回・監視できる「屋外型 警備ロボット」が有効であると思われる.また,屋外を巡回・監視できるロボットは,作 業環境の悪い場所,例えば原子力発電所などでの活躍も期待できる.本研究では,人(不 審者・不法侵入者など)に対応するため,人を追尾する機能をもたせた,屋外を警備でき るロボットの開発を目指す.

### 1.4 本研究のねらい

本研究は,作製したロボットを用い,警備ロボットに必要な動作とその制御システムの 説明をメインにおき,大まかなメカニズムを記載する.

## 1.5 本文の構成

本節は本文の構成とその内容についての説明する.第二章では本ロボットの持つ機能についての説明をする.第三章ではメカニズムについて,おもに移動機構に重点をおいて説明する.第四章では制御システムの構成,各部の説明を行う.最後に第五章では本研究の結果と今後の課題についてまとめる.

## 第2章 ロボットの持つ機能

本章では,人間の警備員をサポートするために必要なロボットの機能について述べる.



図 2.1

本研究のロボットを図 2.1 に示す.これは屋外型警備ロボットであり,人間の警備員の負担を減らすための二つの機能を持っている.一つは遠隔操縦機能である.人間が操作をすることで,屋外での警備をサポートすることができる.もう一つの機能は自動的な監視(自律モード)を行う.その内容は,以下のとおりである.ロボットを警備させる場所ま

で移動させた後,ジョイスティックのボタン操作により,静止状態時に,カメラ(画像処理用)が動く物体をとらえると,操縦者側のPCに警報(警告音)がながし,異常を報せる.

これらの機能によりロボットを遠隔操縦によって警備をおこなう場所まで移動させ,自 律モードに切り替えることで人間の手を離れ自動的に周辺の警備をおこなうことができる.

### 2.1 遠隔操縦機能

遠隔操縦機能は、図 2.1.1 ネットワーク構成図に示すように、操縦者側 PC、警備ロボット側 PC 2台の合計 3台のPCで行う、警備用ロボットに、制御PC・カメラ用PCを用いることにより、制御と処理速度の重いマルチメディア処理であるコミュニケーションツールを区別する事で、システムの安定化を図っている、具体的に、この 3台の各PCには、無線LANカードが搭載されており、TCP/IPを利用した、IPアドレス間通信をこの 3 者の間で行う、操縦者側のPCと警備ロボット側の制御PC間はソケット通信を行い、操縦者側のPCと警備ロボット側の制御PC間はソケット通信をした映像通信を行う。

ネットワーク環境は、図 2.1.1 に示すにように、グローバルネット内に、NATサーバを兼ねた無線 LAN中継機器(ルータ)を設置することにより、グローバルネット側から見て閉鎖的なローカルネットエリアを構築している。これにより、グローバルネット内からローカルネット内へのアクセスを不可能とした実験エリア環境を築く、具体的には、操縦者側 PC・警備ロボット側制御 PC・映像転送 PCの3者に対して、固定 IPアドレスを割り振る、操縦者側 PCアドレスを 192.168.0.3、警備ロボット側制御 PCアドレスを192.168.0.4、映像転送 PCアドレスを192.168.0.5とする。この3者の IPを固定とすることで、今後、ロボットの移動範囲拡大のために無線 LAN中継機器を増設したとしても、そのローミング環境内ではロボットの遠隔操作を行うことができる。

図 2.1.2 に示すように,操縦者はロボットに搭載されたウェブカメラによる映像を,操縦者側のPCで見ながら,操縦者側のPCにつながれているUSBジョイスティックを動かし操縦する.その方法は,無線LANカードを通じて受信した命令を元に,ロボット用インタフェースボードに指令を送る方法を用いる.操縦者側のPCのジョイスティックはゲームポートを介して,ジョイスティックX軸・Y軸座標,ボタン1,ボタン2の有無の情報をリアルタイムにて検出し,その情報をソケット通信によって警備ロボット側制御PCへ送信する.警備ロボット側制御PCは受け取った情報を元に動作へと反映する.ジョイスティック情報の検出をゲームポートから行う理由としては,汎用性の高いデバイスを用いることによって,家庭・学校等から,遠隔操作を行う際の便利を考慮したものである.

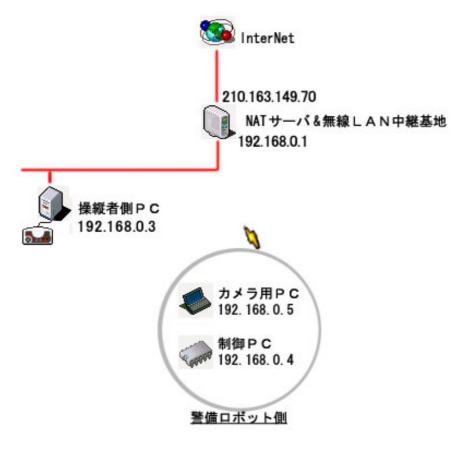


図 2.1.1 ネットワーク構成図

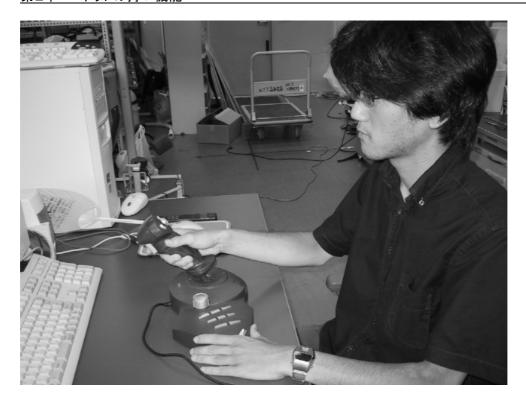


図 2.1.2 USB-Joystick による操縦

#### 2.2 衝突の回避機能

図 2.2.1 に示すように,赤外線センサ,超音波センサをロボットの前面に取り付けることによって,遠隔操縦時の衝突を未然に回避する.具体的には,赤外線センサの前方約15cm以内に物が接近すると,警備ロボットは自動的に,赤外線センサ後方に対して,反応しない場所まで移動する.

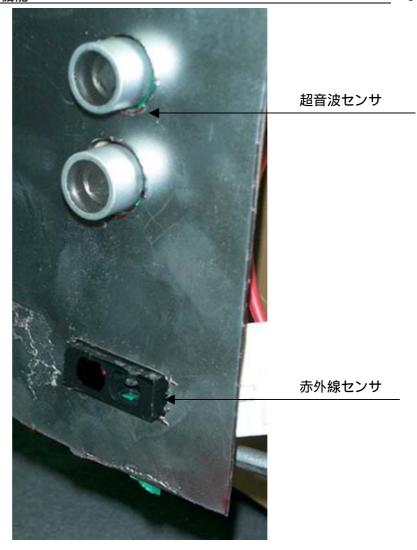


図 2.2.1

#### 2.3 C C Dカメラによる画像処理機能

画像処理機能は、物理的対応のために用いる.具体的に、ロボットが自律モード時に侵入者(動く物体)を検知・認識し、その物体に対し接近する際に目標を見失わないために使用する.超音波センサや赤外線センサなどのセンサのみでも、物体を検知して接近していくことはできるが、動くものだけを検知し、追うことは困難である.また、カメラ(画像処理用)は、超音波センサなどよりも、遠くの物体を検知することができるため、ロボットの監視範囲も広がる.

## 第3章 メカニズム

本章では本研究における警備ロボットの機械構造とその動き,そして実際に作製された警備ロボットを写真を交えて説明する.





正面図 側面図

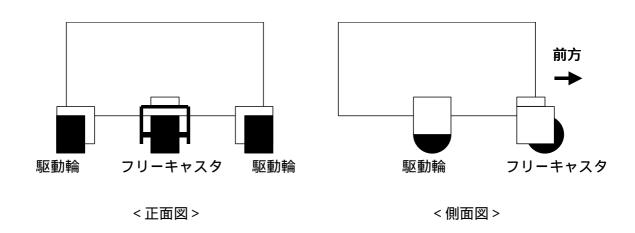


上面図

10

## 3.1 移動機構の構成

本研究での警備ロボットの移動機構は図 3.1.1 に示すように,本体の中心に対して左右に並列に配置した,それぞれが独立して動く2つの駆動輪と,支えとして本体前方に配置した360度フリーに動くキャスタで構成されている.両駆動輪の回転方向と回転速度をそれぞれ制御する事により,前進,後進,その場での回転による方向転換,カーブを描いて進むことができる.



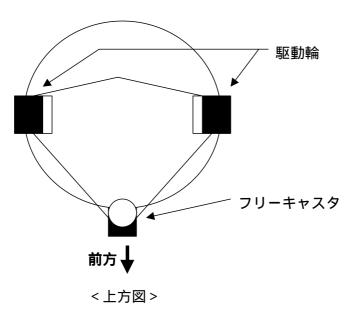


図 3.1.1 移動機構の構成

11

#### 3.2 実装

駆動輪,キャスタに使用しているタイヤは,本ロボットの屋外での使用を可能にするために,直径120mmのオフロードラジコン用大型タイヤを採用.本体の重量を支えるために,ホイールにはウレタン制のタイヤを削りだしてはめ込むことで軸に対する剛性を高めている.また中空となっているタイヤとホイールの間には発砲ウレタンを注入することで,タイヤがつぶれて沈んでしまう事を解消した.図3.2.1 左はオフロードラジコン用大型タイヤ,右はホイールにはめ込む前のウレタン製のタイヤである.



図 3.2.1 タイヤ

モータから駆動輪への動力の伝達は傘歯ギアによって垂直に伝えている.これにより駆動輪部をコンパクトにまとめ,本体への取付,位置の調整等を容易にしている.図3.2.2 は駆動輪部の設計寸法図である.

12

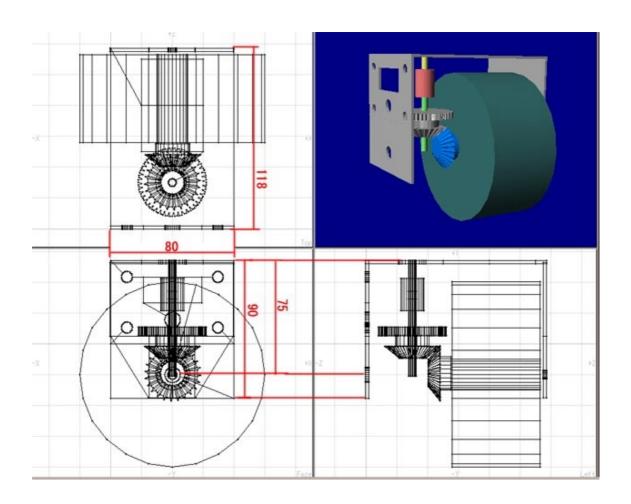


図 3.2.2 駆動輪部設計寸法図

図 3.2.3 , 図 3.2.4 はそれぞれ本体に取り付けられた駆動輪部とキャスタである.また , 図 3.2.5 は足回り全体を示している .



図 3.2.3 駆動輪部

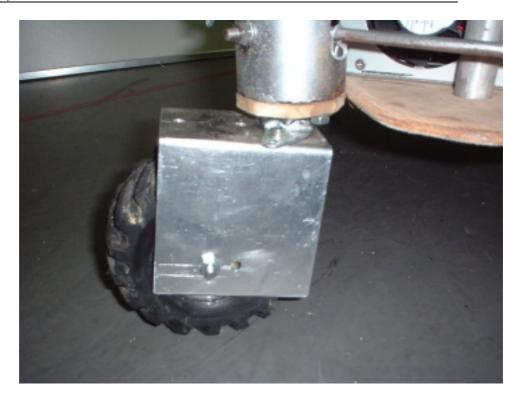


図 3.2.4 キャスタ



図 3.2.5 足回り全体(本体前方)

## 第4章 制御システム

本章では図 4.1.1 に示す制御システム仕様書を元に , 制御システムの構成から制御のながれにはじまり , 制御に関する各部の詳しい説明を記載する .

## 4.1制御システムの構成

## 制御・PC

ロボット用インタフェースボード	富士通製 A/D D/AUPP ボード
MotherBord	GIGABYTE GA-6VA
CPU	Intel Cleron 633Mhz
Memory	DIMM 128M × 2
HardDisk	東芝製 NotePC 8G
Sound Card	Creative SoundBlaster128
無線LANカード	MELCO 製

## センサ系

赤外線センサ	ベストテクノロジ製 × 4

### 画像PC

ノート型 P C	TOSHIBA 製
カメラ	Creative USB CAMERA
無線LANカード	MELCO 製

図 4.1.1 制御システム仕様

図 4.1.1 に示すように,本ロボットの制御システムは制御ボード,赤外線リモコンセンサ,モータ×2,超音波センサ,そして電源部によって構成されている.制御ボードは図 4.1.2,図 4.1.3 に示すように,演算用 C P U ボードとアナログ信号の入出力用 A D - D A ボードで構成されており,モータはドライバによって制御する.図 4.1.2 は制御システムの構成図であり,矢印向きは入出力する信号の流れを表している.

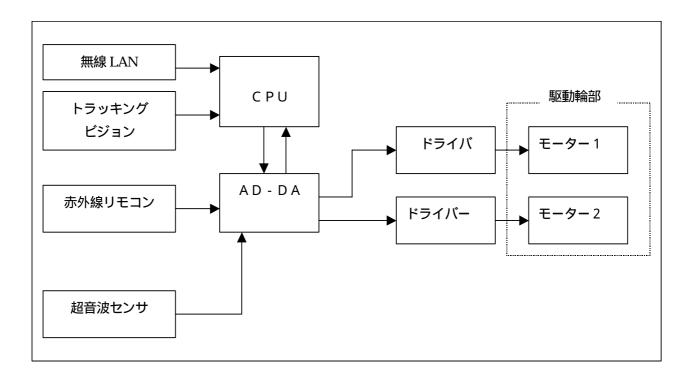


図 4.1.2

制御ボードの構成はCPU,メモリを有するCPUボードと拡張ボードであるグラフィックカード,LANカード,AD-DAボード,主記憶装置のハードディスク,入力機器のキーボード,マウス,グラフィックカードからの画像情報を表示するCRTで構成される.

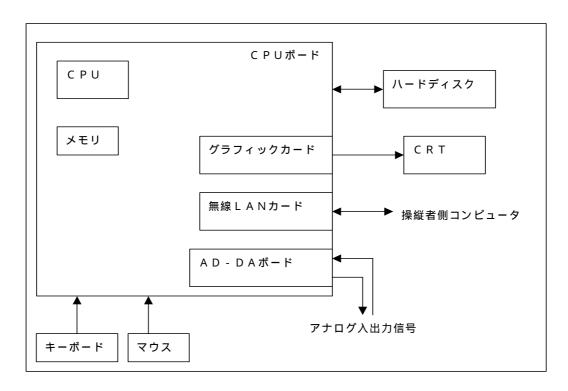


図 4.1.3

## 4.2 CPUボード



図 4.2.1

CPUボードは産業用のものではなく,市販されている組み立てパソコン用のものを使用した.その意義は,処理速度が高速,ISAバス,PCIバスを有し拡張性が高い,安価の3点である.よって本研究のCPUボードの構成は一般的なパソコンとなんら変わりは無い.ロボットへの実装を図4.2.1,図4.2.2へ,主なスペックを図4.1.1で示す.



図 4.2.2

#### 4.3 ロボット用インターフェイスボード

AD-DAボードには富士通口ボット用インターフェイスボード "RIF-01"を使用(図4.3.1). このボードは、図14に示すようにISAバス規格のハーフサイズ・ボードに、アナログ入力36チャンネル、アナログ出力16チャンネル(-10v~+10)
デジタル入出力32チャンネルを搭載したロボット用インタフェース・ボードである.入

出力制御チップ部としてD/A変換器 1 2 ビット/8 チャンネルのMP 7 6 1 3 (EXAR 社製)を 2 個搭載している。A/D変換器は 1 2 ビット/8 チャンネルのAD 7 8 9 1 (アナログ・デバイス製)を 2 個搭載している。さらに、日立製 Universal Plus Processor 以下UPPも 2 個搭載している。インタフェース部としてベース・アドレスはジャンパピンにて000H~07F8 Hの範囲でISAバスのアドレス空間のうち、8 バイト分の領域を占有し、任意に設定できる。本ロボットは図 4.3.2 が示すようにRIF-01ロボット用インタフェースのD/A・A/D・UPPの機能を使用する。



図 4.3.1

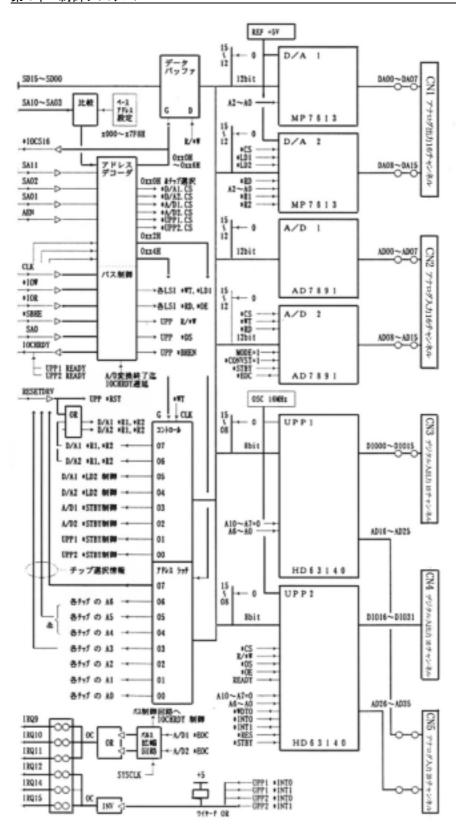


図 4.3.2

D/A は駆動用モータを直接制御するロボットドライバへの命令伝達を行う. A/Dについては外部センサからのアナログ電圧信号の読み取りを行っている. UPPについては赤外線リモコンからのパルス信号を解析している. 基本的にRIF-01は電圧・パルス信号の読み取り,測定を行う. 測定を行い具体的なデータの形で読み取りを行うために下記に示す4つのC言語サブルーチンが必要である.

```
int regst(unsigned baseadr)
  reg_data=0xcf;
  _outpw(baseadr+4, reg_data); //ジャンパーとの認証
  int regstoutpw(unsigned baseadr, unsigned int regstno)
               reg_data=regstno;
               outpw(baseadr, reg data);
                      return 0:
       }
  double daoutpw(unsigned baseadr,int portno,double
                                               value)
  {
       int 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");
       }
               }
  double adinpw(unsigned int baseadr, unsigned int portno)
       double
                      volt;
                      ad read data;
       int
       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;
   ad_read_data=(signed)_inpw(ba \(^1\) Mseadr );
   if(ad_read_data & 0x0800) ad_read_data |= 0xf000;
   volt = (double)ad_read_data / 2048 * 10;
   return(volt);
}
else{
   erron=(1);
   return(erron);
}
```

#### ・ボード初期化

RIF - 0 1 ボードの初期化をするにあたって regst(base\_adr)のサブルーチンを使用する.\_outpw(baseadr+4, reg\_data)によってあらかじめ図 4.3.2 に示すようにISAバス上のジャンパピンの設定状態と reg\_data=0xcf=base\_adr という状態を一致させる.この命令をメイン文に記載することによってこのボードを待機状態に維持することができる.
尚,下記に述べる機能は全てボードの初期化を必要とする.

#### • D/A

このボードのD/A部の機能を使用する場合,上記の初期化手順によってこのボードへのアクセスを行う.その後,例えば図14に示すようにCN1アナログ出力のチャンネル1に対して+5vの出力を行う場合,daoutpw(base\_adr,1,5.0)となる.また,チャンネル5に対して-7vの出力を行う場合,

daoutpw(base\_adr,1,-7.0)となる.

#### · A/D

A/Dを使用するにあたって adinpw()のサブルーチンを使用する. A/D部はアナログ電圧の入力であり, D/A部はアナログ電圧の出力である. 故にC言語上で読み取った電圧値

を測定するためにサブルーチンの最後で下記の命令を実行する必要がある.

volt = (double)ad\_read\_data / 2048 \* 10; return(volt);

#### ·UPP

UPPの使用にあたって下記の特別な初期化設定が必要である.

#### 

図4.3.2 に示す C N 3 U P P 1 6 チャンネルのデジタル入出力端子の 1 ~ 8 チャンネルを下位ビット,8~16 チャンネルを上位ビットと定める.上記ソースの 1 の行によって下位ビット 8 に対して,M P U (直接入力)モードに設定を行う.次に,2 によりM P U ポートに指定する端子を 1 6 ビットの形で指定を行う.この場合,f f という指定により全ての端子がM P U端子として設定されることとなる.各チャンネルのモード設定を行った後,3 により上位ビットからの信号を判断するのか下位ビットからの信号を判断するのかを指定する.つまり U P P はマイクロコンピューターである故に,A/Dなどのような入力信号を直接受け渡すのではなく.そのまま使用できるデータの形に変換する処理を施すことができる.よってここでは下位 8 ビットを指定することにより入力されたチャンネル

をビット変換した形を4のような命令により取得することができる.

#### 4.4 ロボットドライバ

モータの制御用ドライバには , 岡崎産業製 " Titech Robot Driver Ver.1 " を使用する . ここでは , ドライバの特性と実装について述べる .

#### 4.4.1 TITECH ROBOT DRIVER について

TITECH ROBOT DRIVER の特徴として,小型,軽量,高出力(750W),低自己消費型(PWM 駆動),であり,電流制御,速度制御,位置制御の3つのモードがある.また,タコジェネ無しでの速度制御や同じくタコジェネ無しの即応性の良い位置制御を実現している.

項目	仕様	備考
定格出力電圧	± 30V	
定格出力電流	± 3.8V	
最大出力電圧	± 45V	
最大出力電流(連続)	± 16.7V	
最大出力(連続)	750W	十分な放熱板使用時
主電源	D C 12 ~ 48V	
指令入力電圧	± 10V	
外形寸法(LWH)	100 * 90 * 35	

重量	130g	放熱板を除く
	電流制御	
制御モード	速度制御	タコジェネ不要
	位置制御	
保護機能	電流制限	
	電気ブレーキ停止	モータ端子短絡
モータ・ブレーキ機能	フリー・停止	モータ端子開放
	正逆転限界停止	リミット・スイッチ使用

図 4.4.1.1 駆動回路の諸仕様

前述したように,動作モードは,それぞれ電流制御,速度制御,位置制御モードの三種類ある.本研究では,このモードの中から速度制御モードを設定した.速度制御モードは(図 4.4.1.2)でフィードバックに用いるモータの回転速度情報は,擬似タコジェネレータ回路によって演算する方法と,外部の機械的タコジェネレータによって測定する方法の二つがある.

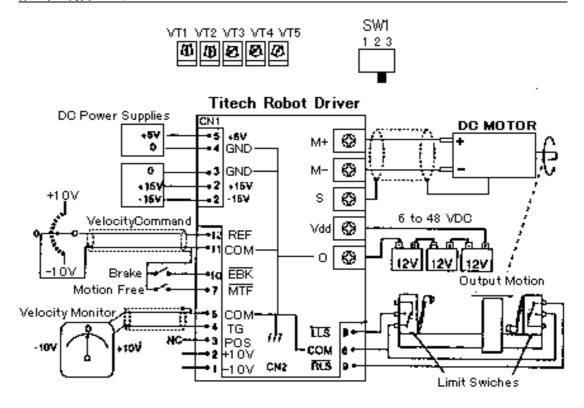


図 4.4.1.2 速度制御モード

PIN	信号名	備考
5	+ 5 V	+ 5 V ( 5 ~ 6 V ), 0.1 A 電源
4	GND	5 V電源の 0 V端子
3	GND	± 15 V電源の0 V端子
2	+ 1 5 V	+ 1 5 V ± 1 0 % , 0.1 A電源
1	- 15 V	+ 1 5 V ± 1 0 % , 0.1 A電源

図 4.4.1.3 CN1

PIN	信号名	備考
12	REF	指令信号 , 入力インピーダンス: 100 V
11	СОМ	グラウンド端子
10	EBK	電機ブレーキ(モータ端子短絡)
9	RLS	正回転リミットスイッチで電機ブレーキ作動
8	LLS	逆回転リミットスイッチで電機ブレーキ作動
7	MTF	フリー停止(モータ端子解放)
6	СОМ	グラウンド端子
5	СОМ	グラウンド端子
4	T G	タコジェネ入力 / 電子ガバナ出力
3	NFB	位置制御用ポテンションメータ・フィードバック
2	+ 1 0 V	最大 1 0 m A
1	- 1 0 V	最大 - 10mΑ

図 4.4.1.4 CN2

モータ電機子抵抗値Raの補償,電子ガバナ回路を正常に動作させるには,使用するモータの電機子抵抗値Raに合わせて調整する必要がある.そのためには,電機子抵抗値Raをモータの規格書から,もしくは実際に測って調べ,VT1を調整するVT1resを計算する.

 V T 1 は 0 ~ V T 1 max[]の範囲で調整でき、その回転量 V T 1 pos は次式のように計算できる。

$$V T 1 pos = V T 1 res / V T 1 max$$

ただし、標準仕様では電流センサー次側巻数はNturns = 3 , 半固定抵抗 V T 1 は V T 1max = 2 0 0 k である. なお , V T 1 の設定は , V T 1 上に刻まれた印をもとに目測で

おこなうため、設定値に多少のずれが生じる.ここで注意しなくてはならないことは、この設定値を計算値より大きくした場合、回路の動作が不安定になることである.そのため計算値より少なめに設定しなければならない.モータ回転速度の計算、電子ガバナ回路使用状態では、コネクタCN2の4番ピン(TG)からモータの回転速度 pをモニターできる. p[rpm]は、TG端子の出力電圧Vp[V]に次のように直線的に比例する.

$$p = (Vrated / 3 0 0) * free * V p$$

ただし, Vrated[V]はモータの定格電圧, free[rpm]は定格電圧での無負荷回転数であり, 対象とするモータの説明書から調べたものを使う.

速度フルスケール調整(V T 4). 最大入力指令電圧に対するモータ速度 fn をフルスケール速度と呼ぶ.

fn は , V T 4 の回転量 V T 4 pos(0~100%)によって , 次式のように計算する .

$$fn = (47 / 100 V T 4 pos + 47)* pmax[rpm]$$

速度制御モードの電流制限の調整については, VT5の回転量を70%以下に設定する. 以上の特性から,モータへの実装について述べる.

#### 4.4.2 ロボットドライバの実装

本ロボットの駆動用モータ電圧を直接制御しているロボットドライバを図 4.4.2.3 に示す .また ,その構成図を図 4.4.2.4 に示す。このロボットドライバは図 4.4.2.4 に示す 1 1、1 2 ( C O M、R E F )にD/Aからの信号を入力することにより V d d、0 に接続された D C バッテリーをエネルギー源として駆動用モーターの制御を行う。ロボットドライバ自体の制御モードとして「速度制御モード」・「電流制御モード」があり、図 4.4.2.4 に示す S W 1 の切り替えによりモードを選択することが可能ととなる。本ロボットは速度制御モ

ードを利用してモーター制御を行う。具体的に制御対象となるモーターは4つ存在するのでロボットドライバは各々4つ必要となる。故にD/AチャンネルはCH1~4チャンネルを使用する。各チャンネルの信号端子とグラウンド端子を図4.4.2.4に示す11、12(COM、REF)に接続し、下記に示すC言語プログラムによりロボットドライバへの命令を行う。

daoutpw(base\_adr,1,0.0); //出力 CH1 ロボットドライバ1

daoutpw(base\_adr,2,0.0); //出力 CH2 ロボットドライバ 2

daoutpw(base\_adr,3,0.0); //出力 CH3 ロボットドライバ3

daoutpw(base\_adr,4,0.0); //出力 CH4 ロボットドライバ4

上記Cソースはロボットドライバに対して0 vを入力している。故にドライバに接続されているモーターは停止状態を維持する。速度、制御モードを用いることにより4つのドライバに対して例えば、下記のようなプログラムを実行すると

daoutpw(base\_adr,1,10.0); //出力 H1 ロボットドライバ 1

daoutpw(base\_adr,2,10.0); //出力 H2 ロボットドライバ 2

daoutpw(base\_adr,3,10.0); /出力 CH3 ロボットドライバ3

daoutpw(base\_adr,4,10.0); //出 CH4 ロボットドライバ4

ロボットドライバは - 10 v ~ + 10 vまでの勾配により接続されているDCバッテリーにより速度制御を行うので、例えば接続されているDCバッテリーがDC24 vだとするとMAXの24 vをモータに流し制御を行う。また、D/Aからの値がこれと逆の - 10 だとするとドライバは - 24 vをモータに供給し制御を行うことになる。つまり、モータはD/Aからの電圧信号が0 vだと停止、 + 信号を正回転、 - 信号を逆回転となる。また、4つのモータ全て、例えば + 4 v というD/Aからの信号が入力されると。図 4.4.2.4 に示

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

図 4.4.2.1

すVT1~VT5を調節する事によりほど同回転での制御が実現され。ロボットは直進や斜め移動を簡単に実現することが可能である。その入出力の関係を図 4.4.2.4 に示す.また,ロボットドライバ使用にあたって注意として、図19に示す Vdd、0 に対して大容量のアンペアを持つDC電源を使用する際、図 4.4.2.4 に示す11、12(COM、REF)に対して D/A からの入力電圧がまったくない場合、先に Vdd、0 にDC電源を接続してしまった場合、ボードの暴走を引き起こしてしまう可能性が十分にあるといことである。この点に注意して使用すればボード、モーターの破損はまずないと考えられる。実際、本ロボットに積んだ場合。図 4.4.2.4 に示す Vdd、0 にスイッチを取り付けることによって上記の危険性から回避することができた。しなしながら最終的にそのスイッチを押すのはプログラムではなく人間であるので、D/A 入力電圧を確認した後に入電するよう心がけなければならない.

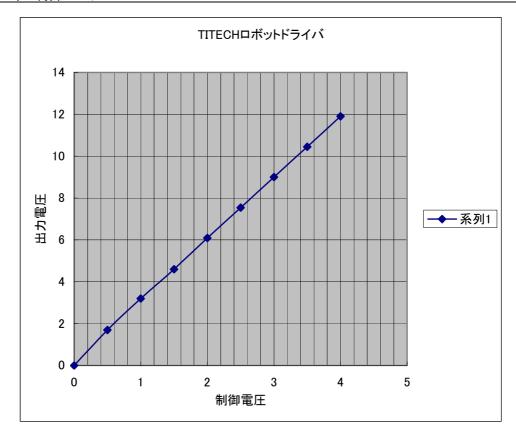


図 4.4.2.2



図 4.4.2.3

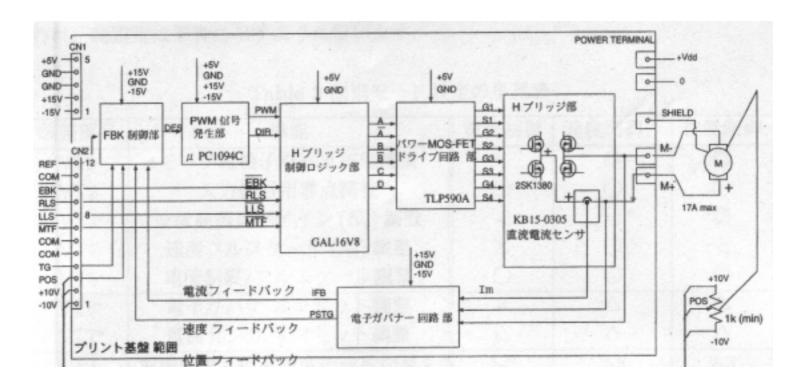


図 4.4.2.4

## 4.5 モータ制御

本研究のロボットの駆動には,日邦電機株式会社製の型式 MM - 26EH HIGH SPEED MOTOR (図 4.5.1.1) を 2 個使用した.

## 4.5.1 モータについて



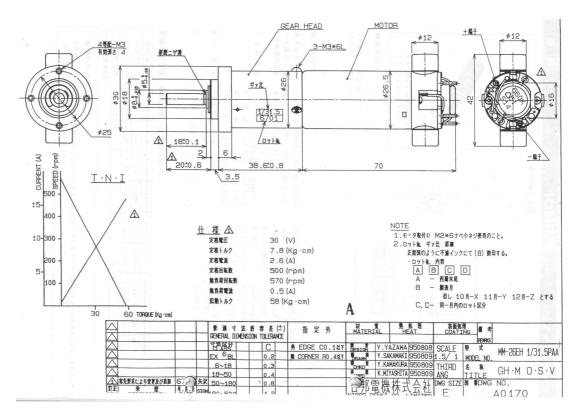


図 4.5.1.1

MM - 26EH HIGH SPEED MOTOR は高速,高出力で頻繁な起動・停止動作をおこなう電動工具用モータのため,苛酷な使用に耐えられるようにつくられている.また,このモータは,減速機付きのモータであるため,大トルクを出すことができる.

ここで使用したモータは,ブラシ付きDCモータという種類のものである.DCモータとは,直流で駆動するモータを(広義で)総称していう.DCモータには,ブラシ付きDCモータ以外にも,ブラシレスDCモータ,ステッピングモータ,アクチュエータなどがある.

ブラシ付きDCモータ(DCモータ)は,ブラシ,整流子,そしてコイル(電磁石)を 組み合わせたもので,電流を力に変換し動作する(図 4.5.1.2).ブラシは,電流を整流子 に送る(整流子と接触することにより、その先にある電磁石に電流を流す)役割をしてい る.整流子とは,モータの回転子に取付けてブラシに接触する部分のことで,コンミュテ ーターともいう.

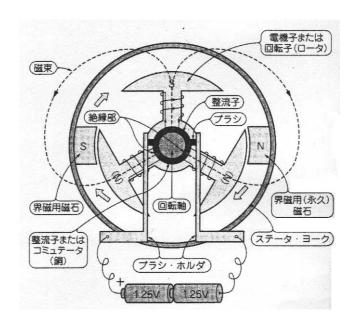
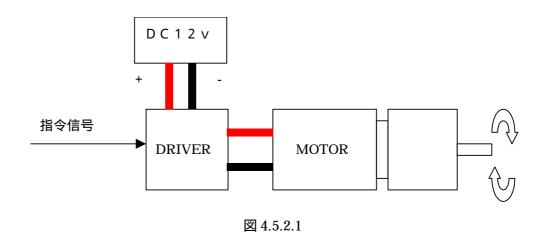


図 4.5.1.2

ブラシ付きDCモータの長所は,起動トルクが大きく,出力効率が高いこと,他のモータと比べると安価であることが挙げられる.また,短所としては,整流子の寿命に限界があることや整流時に電気的ノイズが発生することが挙げられる.

#### 4.5.2 モータの実装

モータに対する指令信号は図 4.5.2.1 に示すロボット用インタフェースボードのD/A 出力チャンネル0 ,1 より出力し ,それぞれのモータに接続されたドライバに信号を送る . ドライバは指令された信号に応じて電源からモータへと電流を流す . 図 4.5.2.1 はドライバによって制御されるモータのイメージ図である .



## 4.6 環境認知センサ

本ロボットは,環境認知センサとして,超音波センサ,赤外線センサ,カラートラッキングビジョンを用いる.

## 4.6.1 超音波センサ

超音波とは,人間の可聴範囲以上の(約16kHz)の音波をいう.超音波は音であるから常温の空気中を約340m/sの速度で伝播する.よって,音は1cmの距離を約28 μ s e c だけの時間をかけて進む.すなわち超音波を発射してから物体に反射して戻っ

てくるまでの時間を測定すれば,超音波の送受波器から物体までの距離を知ることができる.超音波を発射してから,反射波が検出されるまでの時間を T µ s e c とすると,物体までの距離 d は,

$$d = 0.5T * 340 * 0.001 = 0.17T [mm]$$

で計算できる.このため超音波センサは,障害物の検知や距離測定,自機座標の確認などによく使用される.しかし,硬い物(金属,木材,コンクリート,ガラス,ゴム,紙など)は超音波をほぼ100%反射するのでこれらの物体の有無は十分検知することが出来るが,布,グラスウール,綿のやわらかく空気を含んでいる物体は超音波を吸収するため物体の有無を検知し辛い.また,物体の表面の起伏が大きいものは,超音波が乱反射するため検知しにくいことがある.

超音波センサと呼ばれるものは,電気信号を超音波に変換した後,空中に発射する送波器と空中を伝搬してきた超音波を受けてそれを電気信号に変える受波器の二つをあわせたものを指す.今回使用する空中用超音波センサは,電気信号を超音波に変換するための振動子として圧電セラミックを用いたものである.

電極をもうけて分極された圧電セラミックにおいて,電極間に電圧を加えると,この電圧に対応した機械的な変形が生じ,また圧電セラミックに外部から応力を加えると電極間に電荷が発生する.これを圧電現象といい,今回使用した超音波センサは,この圧電現象を利用している.構造は,圧電セラミックに金属板を貼り合わせたユニモルフ構造の振動子で形成されている.

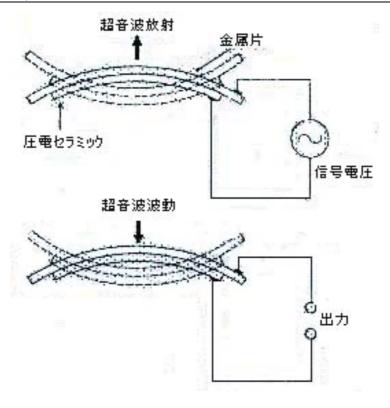


図 4.6.1.1 振動子原理図

(図 4.6.1.1)この振動子に信号電圧を加えると屈曲振動を起こし,この機械振動周波数と信号電圧の周波数を一致させたときに最も効率的に超音波を空中に放射する.また,振動子に空中からの超音波の波動が加わると,振動子に屈曲振動が生じ振動子の電極間に波動に応じた電気の出力が発生する.前者が送波器で,後者が受波器として用いられる.

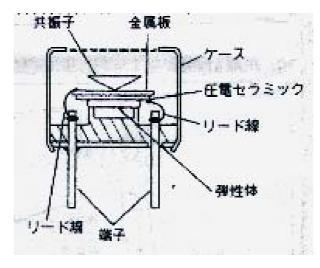


図 4.6.1.2 振動子構成図

この超音波センサは,開口型超音波センサといい,ユニモルフ振動子と共振子を結合して構成されている開口型のため非常に感度が高いのが特徴である.サイズは,外形10~24mmで,主な周波数として,25kHz・32kHz・40kHzの3種類がある.

## 4.6.2 超音波センサの実装

図 4.6.2.1 に示すのは ,ベストテクノジー製 8 方向超音波センサボードである .図 4.6.2.2 にその構成図を示す .このセンサボードは基本的に A/D からの入力によって対象物の検知を行うことができる .しかしながら 8 チャンネルのチャンネル切り替え時において D/A からの 5 v 信号を必要とする .

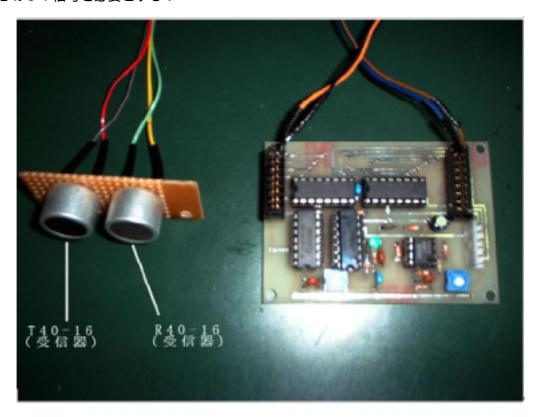


図 4.6.2.1

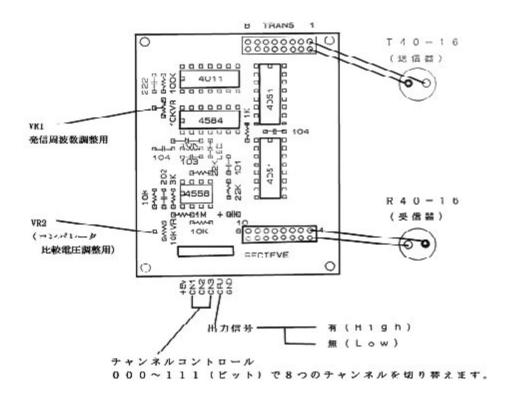


図 4.6.2.2 超音波構成図

具体的に、図 4.6.2.2 に示す「チャンネルコントロール」部においてCN1~3に対して5 v信号を入力すると接続されている超音波8チャンネルが切り替わる構成になっている. 例えば,1チャンネルに接続されている超音波センサを使用したい場合0 v・0 v・5 vをCN3・CN2・CN1に対してA/Dから入電を行う. これによりボード回路構成がチャンネル1に対する処理を開始する.

超音波センサは対象物までの距離をリアルタイムに測定することができるものもあるが, このセンサボードに関しては予め設定した距離まで近づくと応答がある.つまり,その距離に対して対象物が有るか無いかのみの判断を行う.

#### 4.6.3 赤外線センサ

赤外線センサは,可視光のセンサでは実現できない特徴を持っている.一つは,人間の視覚を刺激しないで対象物の検出が可能である.もう一つは,対象物の温度を遠方から非接触で瞬時に測定できることである.このような特徴から,高炉・焼成炉・重合反応器などの温度制御技術,材料の均一性・異種材料の接合部の良否・応力分布などを温度分布から測定する技術や病気の原因となっている炎症箇所検知,腫瘍検知などの新しい医療技術などにも使われはじめている.赤外線センサは動作原理により,2種に大別することができる.1つは熱型赤外線センサ,もう1つは量子型赤外線センサである.熱型赤外線センサは,感度,応答速度は低いが,波長帯域が広く常温で使用でき,使いやすいという特徴があり,量子型赤外線センサについては,検出感度が高く,応答速度が速いなどの特徴を持っている.本ロボットに使用する赤外線センサは,ロボット自身の反応速度をあげるため,検出感度が高く,応答速度の早い量子型赤外線センサを使用した.

#### 4.6.4 赤外線センサの実装

本ロボットに使用した赤外線センサは光変調回路を搭載しており,外乱光に影響されにくいセンサの発光部から最大約800mm離れた物体を検出することができる(図4.6.4.1,図4.6.4.2).また,検出レベルは,発光素子のフィルタ窓の開閉を調整することで変更できる(図4.6.4.3,図4.6.4.4).



図 4.6.4.1 光センサ写真

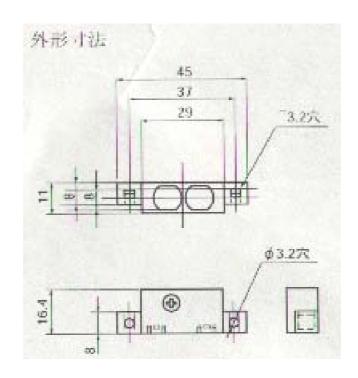


図 4.6.4.2 寸法図



図 4.6.4.3 感度調整ネジの写真

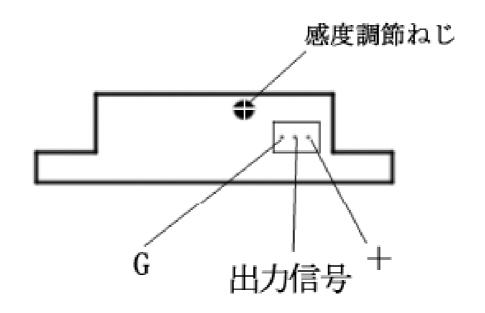


図 4.6.4.4 かんどちょうせいネジの絵

このセンサはA/Dへの入力において使用することができる.図 4.6.4.4 に示す構成図より右より+・出力信号・Gとなっている。+、Gに対してDC5 vを入電することによりこのセンサは作動し、図 4.6.4.1 に見られる2つレンズ,発光レンズと受信レンズにより対象物を判断し、図 4.6.4.4 に示す出力信号より、その判断結果を電圧信号により出力を行う.出力された電圧信号は図7に示すように、RIF-01のA/D部へ接続により読み取ることが可能となる.本ロボットはこのセンサを8方向に取りつけることによりロボットの周囲に対物があるかないかを判断できる.具体的な使用として下記のC言語ソース

により, S1~S8にセンサ出力値が代入される.

S1=adinpw(base\_adr,8); //入力 CH\_\_8

S2=adinpw(base\_adr,9); //入力 CH\_\_9

S3=adinpw(base\_adr,10);//入力 CH\_\_10

S4=adinpw(base\_adr,11);//入力 CH\_\_11

S5=adinpw(base\_adr,12);//入力 CH\_\_12

S6=adinpw(base\_adr,13);//入力 CH\_13

S7=adinpw(base\_adr,14);//入力 CH\_\_14

S8=adinpw(base\_adr,15);//入力 CH\_\_15

前節で記載した超音波センサと赤外線センサはどちらも非接触にて物の有無の検出を行う ものであるが、超音波センサは布などの超音波を吸収する物体の検知が難しい.また,赤 外線センサは光の反射を利用しているため,ガラスなどの透明な物体の検知ができないと いう欠点があるため,2種類のセンサを併用する事でそれぞれの欠点を補い,より正確な 物体の検出が可能となる.

#### 4.6.5 カラートラッキングビジョンPCI

画像処理は富士通株式会社のカラートラッキングビジョンPCI(図 4.6.5.1)を用いて行う。ここでは、トラッキングビジョンの主となる「動きベクトル計算」・「累積相関」の2つの説明を行う.

このボードは専用ハードウェアを用いることにより, RGBカラーでの相関演算を33 msの間に500回以上計算することで複数のターゲットを追跡することが可能となり, 1テンプレート最大128×128画素による複数のテンプレートとの比較を可能とし ている.トラッキングビジョンの柔軟なメモリアクセス機能は,2つの画像メモリ間での 高速DMA転送機能によってすばやくデータのやり取りを行えるようになっており,動き ベクトル・オプティカルフローの計測や,パターンマッチング処理をビデオレートで実現 している.警備ロボットにこの動きベクトルの測定機能を用いることによってロボットは 任意の作業を画像によって判断することが可能となる.



図 4.6.5.1

#### 動きベクトルの計算

動きベクトルとは,特定のターゲットが,どちらにどれがけ動いたかを表す量である. 例えば図 4.6.5.2 のようにフレーム f で p f の位置にあったターゲットが次のフレームに おいて p f + 1 の位置に移動したとすると, p f と <math>p f + 1 を結ぶベクトルの動きベクトルが v となる.

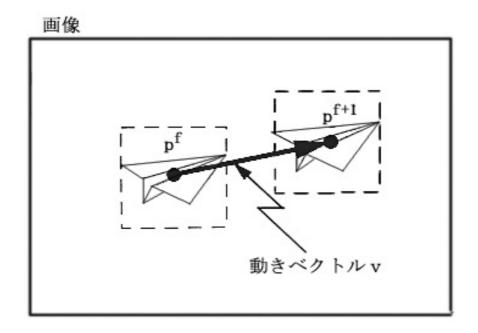


図 4.6.5.2

カラートラッキングビジョンはカメラからの入力画像を時間を追って画像メモリに記憶し、高速DMAにより相互比較することで、特定のターゲット領域が次の瞬間(時間)どこへ移動したのかを判断することが可能である.具体的にターゲットとなる物のポイントを囲む矩形領域を考え、この画像ブロックに対して述べる相関演算と呼ぶ処理を実行し動きの対応を求めるものである.相関演算とは図4.6.5.3に示すように追跡対象とする矩形領域を参照ブロック(テンプレート)として抽出した上で、次のフレームの候補ブロックとの間で、画像の相関値を計算するものである.候補ブロックを2次元的にスキャンしながらこれを繰り返し、相関が一番高い位置を求める.これによって特定の目標が次のフレームでどこに動いたかを示す動きベクトルを知ることができる.尚、候補ブロックがスキャンする領域全体を候補ブロックと呼ぶ.具体的に相関演算では、以下の式で示すブロック内の画像ごとの濃度差の絶対値による総和を求めている.

$$D(u, v) = \sum_{x,y=0}^{7} | R(x, y) - S(x + u + 8, y + v + 8) |$$

R(×,y)は参照プロック,S(×,y)は探索プロックの画像データである.D(u,v)を動きベクトルに対応するディストーションと呼び,ディストーション値がもっとも小さくなる位置をもっとも相関の高い位置とみなし,動きベクトルを求めている.カラートラッキングビジョンでは,参照プロックサイズは8×8画素を基本としており,動きベクトルの探索範囲(u,vの値)は,-8~+7画素に固定される.その結果,探索プロックサイズは23×23となっている.参照プロックの位置や探索プロックの位置は,パラメータとして指定することが可能である.また,複数のターゲットに対する動きベクトルの追跡・測定は1フレーム(33ms)の間に,異なる参照・探索プロックに対して複数回の相関演算を行うことによって可能となる.

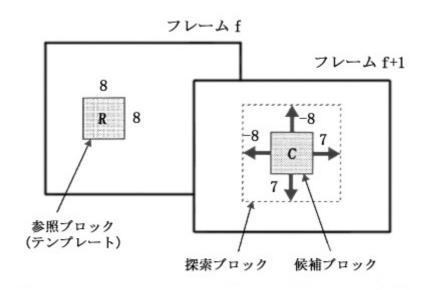


図 4.6.5.3

#### ・累積相関

カラー画像を参照した追跡処理・大型テンプレートでの追跡処理を可能とするため,カ ラートラッキングビジョンでは、累積相関と呼ぶ方式を用いている.

累積相関とは下記に示す式のように , 直前の相関値 D k - 1 (u,v)に今回の相関演算結果を加算することで , 新しい相関値 D k (u,v)を検出できる演算である .

$$D_k(u, v) = D_{k-1}(u, v) + \sum_{x,y=0}^{7} | R(x, y) - S(x+u+8, y+v+8) |$$

この演算を実行するためにカラートラッキングビジョンでは図 4.6.5.4 に記するように 相関演算回路の後段に 2 ポートの画像メモリを搭載している .

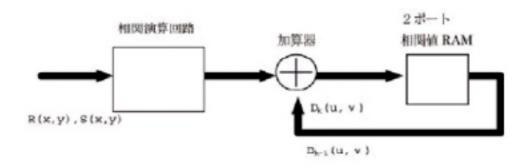
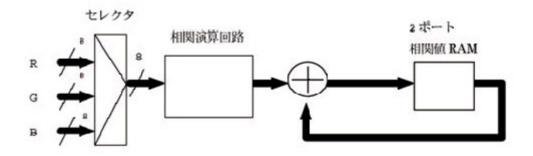
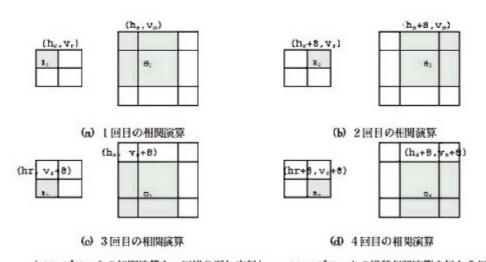


図 4.6.5.4

累積相関を用いることによって、図4.6.5.3に示すようにRGB各プレーンごとに相関演算を行わない結果を累積することで、カラー画像を用いた追跡処理が可能となる.また、複数の部分テンプレートでの演算結果を累積することによって大型テンプレートを等価的に実現している.例えば、図4.6.5.5に示すように16×16画素のテンプレートで相関演算を行いたい場合、参照プロックを4つの8×8テンプレートに分割し順次相関演算を行う.この際、探索プロックの位置も図のように順次ずらす.そして、4回の演算で得たデイストーションを全て累積することにより等価的に16×16画素テンプレートに対するディストーションを得ることが可能となる.しかしながらこの方法においてはRGB3つのテンプレートに対する処理を行うことになるので1つのプレーンで処理を比較した場合、3倍の処理時間を必要とすることになる.また、16×16画素のテンプレートに対する処理は8×8画素のテンプレートと比較した場合、4倍の処理時間を必要とすることになる.また、16×16画素のテンプレートに対する処理は8×8画素のテンプレートと比較した場合、4倍の処理時間を必要とすることになる.プログラミングを行う際、機能と処理速度のトレードオフを行い、用途に応じた適切な処理を選択しなければならない.





(B×8 ブロックの相関演算を 4 回繰り返し実行し、16×16 ブロックの累積相関演算を行なう例)

図 4.6.5.5

#### 4.6.6 カラートラッキングビジョンPCIの実装

以下のプログラム trv()を元に,警備ロボットへの実装について述べる.

下記のプログラム trv()は,1個のウインドウに対して追跡処理を実行する最も基本的なプログラムである.追跡はカラー画像を参照し,カラーでの相関演算を繰り返し行いながら実行する.追跡ポイントは画像の中央に固定されている.また,追跡に必要なテンプレート画像を取得するために,最初に1枚静止画を取り込んでから,追跡処理を開始する.

```
void trv(){
 int size, mag, roff, soff, h0, v0, h, v, rh, rv, sh, sv, I, I2=0, I3=0;
 int dh, dv, mdist, dhz=0, dvz=0, mdistZ=0, devn=0, H=0, V=0;
 int X=0, Y=0, z, Z=0;
 double volt, a, b, c, d, e, f, g, i, j, k, S1, S2, S3, S4, S5, S6;
 /*初期パラメータ*/
 f=300:
                      //追跡時間をフレーム数で指定
 bsize=32:
                      //参照ブロックサイズ
                      //拡大率(間引き率)
 mag=2;
 h0=320;
                      //参照ブロックの中心座標(H)
 v0=240;
                      //参照ブロックの中心座標(V)
 h=h0:
                      //探索ブロック初期中心座標(H)
 v=v0;
                      //探索ブロック初期中心座標(V)
////参照・探索ブロックの中心から左上座標を計算するオフセット
                      //32
 roff = bsize*mag/2;
 soff = roff + 8*mag;
                      //48
TrvInut();
TrvSetSyncMode(EXTERNAL);
TrvVideoThrough();
TrvSetSample(devn, mag-1, mag-1, mag-1, mag-1);//間引き率指定
TrvFetchImage(devn,0);
                                      //参照画面取り込み
TrvVideoInput(devn,DOUBLE_BUF,0);
                                      //画像入力開始
 for(;;){
       TrvWaitNextFrame(0);
       ///////カラー相関演算///////
       rh = h0 - roff;
                      //320-32 = 288
       rv = v0 - roff;
                      //240-32 = 208
       sh = h - soff;
                      //320-48 = 272
       sv = v - soff;
                      //240-48 = 192
TrvColorCorAny(
       devn,
       bsize,//参照ブロック水平サイズ(8の倍数を指定)
       size, //参照ブロック垂直サイズ(8の倍数を指定)
       VRAM_REF, //参照ブロックを取り出す VRAM 番号
       VRAM_SER, / /探索ブロックを取り出し VRAM 番号
       rh, //298 参照ブロック水平アドレス (左上角)
       rv, //208 参照ブロック垂直アドレス (左上角)
       sh,//探索ブロック水平アドレス(左上角)
       sv.//探索ブロック垂直アドレス(左上角)
       0,//累積指定0:累積しない1:累積する
       0,//参照ブロックマスクデータID番号
       0,//探索ブロックマスクデータID番号
       &dh,//(出力)動き水平ベクトル
       &dv,//(出力)動き垂直ベクトル
       &mdist//(出力)最小相関値
        TrvDrawBox( h-roff, v-roff, h+roff, v+roff, 0);
       h += (dh * mag); //新しいカーソルを書く
       v += (dv * mag);
       TrvDrawBox(h-roff,v-roff,h+roff,v+roff, VL_RED);
       TrvVideoStop(0); //画像入力停止
```

}

trv()では,最初に幾つかのパラメーター設定行う、f=300 は300フレームの間追跡を行うことを指定している。bsizze=32,mag=2 は32×32画素の参照ブロックを用い,拡大率2で追跡を行うことを指定している。拡大率を2としているので見かけ上64×64 画素のテンプレートで追跡をしているように見える。h0=320、v0=240 は,参照ブロック(テンプレート)の中心座標であり,h,vは探索ブロックの中心座標である。このプログラムは最初に取り込んだテンプレートを固定的に使うのでh0,v0は一定である。一方,h,vはターゲットの動きに応じて順次更新されることになる。尚,参照ブロックと探索ブロックの実際の指定は,中心座標でなく左墨の座標で行うことになっている。このために中心から左隅を算出するための定数 roff とsoff を下記の用に事前計算する必要がある。

```
roff = bsize * mag / 2;
soff = roff + 8 * mag;
```

カラートラッキングビジョンは TrvInit()に始まる3つの関数, TrvSetSybcMode(), TrvVideoThrough()によっては初期化される.

期化後 TrvSetSample(0,mag-1,mag-1,mag-1,mag-1)

によって間引き率をセットする .TrvSample()を一旦実行すると ,それ以降の全ての相関演算でこの間引き率に従った処理が実行される .尚 ,TrvSetSample()で指定する間引き率は , 0 が間引き無しを意味するものであるので拡大意率m a g から 1 を引いて与える必要がある . また , 間引き率は ,参照ブロックと探索ブロックで独立に指定でき , かつ水平方向と垂直方向の間引き率を独立に指定することができる . 次に TrvFetchImage()を実行し , 図 4.6.6.1 に示すように画像メモリ#0 に静止画を取り込む . カラー・トラッキングビジョンを用いて追跡処理等を実行するためにはビデオ画像の入出力を正しく制御し , これに同期

して処理を行う必要がある.

具体的に、TrvVideoInput(int devn, int mode, int mem)を実行することによって,図 4.6.6.1 に示すようにカメラからの動画像が順次画像メモリに保存される.この時,引数 mode と mem で指定したモードに従って,メモリローテーションが行われる.ローテーションには2つのモードがあり,ここでは2画面切り替えモード(DOUBLE\_BUF)を用いる.また, 画像入力中はカメラ画像がモニタへスルー表示される.

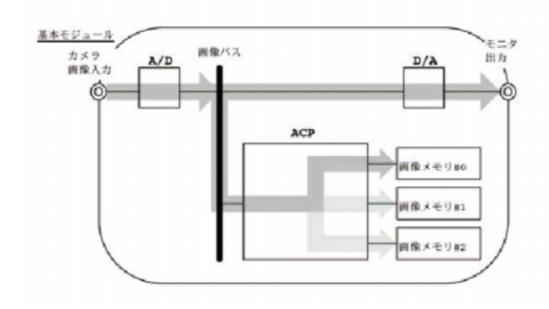


図 4.6.6.1

画像入力を開始した後, for 文の中で毎フレームごとに行閧処理を記述する.

TrvWaitNextFrame()はフレームの開始地点を待ち合わせる関数である。この関数により, 画像信号に同期しながら,処理を繰り返すことができる.一連の追跡処理は原則として1

フレーム (33ms)の間に終了するように設計されなければならない.尚,2フレーム 毎(66ms毎)に処理したり,逆に1/2フレーム毎に処理したりというようにフレーム 周期を変えることも可能である.for文の中では実際以下の関数が相関演算を行っている.

```
TrvColorCorAny(
```

```
0,
 bsize,
 bsize,
 VRAM REF.
 VRAM_SER,
 rh,
 r٧,
 sh,
 SV.
 0,
 0,
 0,
 &dh,
 &dv,
 &mdist
);
```

TrvColorCorAny()は、任意の参照ブロックサイズでカラー相関を実行する関数であり、RGB3つのプレーンについて順次相関演算を行いその結果を累積した後、動きベクトルを求める。相関演算の結果、dh、dvに動きベクトル(dh、dv=-8~7)がmdistに最小ディストーション値がセットされる.TrvDrawBox()は、オーバーレイプレーンにロックカーソル(四角の枠)を描画する関数である.前フレームで描画したカーソルを消すために、値0のカーソルを描画し、次に新しい座標でカーソルを描画する.

TrvVideoStop()は,動画像の入力を停止するためのものである必ずしも実行する必要はないが省エネのためにアイドリング状態に戻す方が好ましいと考えられる.

下記の式において,具体的な制御パラメーターh,vに着目して考えると.

```
h += (dh * mag);
v += (dv * mag);
```

magは拡大率であり.dh,dvはTrvColorCorAny()の関数によって,求められた対象

物の動きベクトルである.故にh, vはリアルタイムに更新されたモニターに映る対象物の現座標である.この座標値を元に,ロボット制御を行う.具体的に,h, vの値をロボット用インターフェースボードD/A 出力に反映させ,モータ制御を行う.

#### 4.7 遠隔操縦

本研究は,インターネット通信でよく使われる.TCP/IPを利用した,サーバ・クライアント方式をとった遠隔操縦を行う.ここでは,そのサーバクライント方式におけるソケット通信の説明,および,ソケット通信を利用した遠隔操縦への実装と応用について述べる.

#### 4.7.1 サーバクライアント方式について

サーバとは、サービスを提供するためのプログラム、またはネットワークの中心となるコンピュータの略称を指す。サーバがサービスを提供する側に対して、サービスを提供される側をクライアントと呼び、ネットワークに接続するためのアプリケーションソフト(WEB ブラウザなど)やOSのことを指す。多くのネットワークを介した通信は、サービスを提供する側・受ける側に分けて考え、サーバ・クライアント型の通信を利用する。サーバ・クライアント型の通信の例えるとするならば、PCを通してホームページの閲覧を行う場合のHTTPサーバーと、そのサービスを利用するWEBブラウザ(クライアント)の関係である。

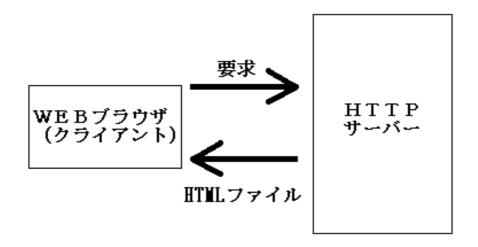


図 4.7.1.1

(図4.7.1.1) HTTP というプロトコル(PC同士のデータ通信の際の規約)に基づいてサーバに対して「このページが見たい」という要求を出し、サーバ側が該当した html ファイルをその応答として送信してくる、これがサーバ・クライアント型の通信である。

TCP/IPは、Internet の標準プロトコルであり、現在最も普及している。複数のPCを接続する際によく利用されるが、必ずしもTCP/IPを使う必要はなく、独自のプロトコルをもって対応することもできる。しかし、PCやPDAなどすでにサポートしている共通のプロトコルを利用することでPC側のソフト開発が省略できるため、TCP/IPが用いられることが多い。TCP/IPプロトコルの構造は、DARPAモデルと呼ばれる4階層の概念的なモデルに表わすことができる。(図4.7.1.2)

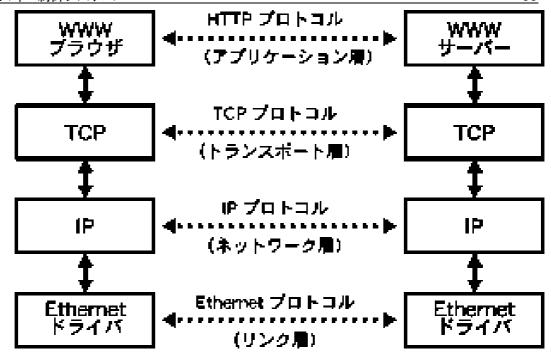


図 4.7.1.2

この層はそれぞれリンク層 (ネットワーク インターフェイス層)・ネットワーク層・トランスポート層・アプリケーション層と呼ばれている.

リンク層は,オペレーティングシステムのデバイスドライバなどを含み,ネットワーク層に対して統一的なインターフェイスを提供する利用する通信機器等のハードウェアで,モデムや,イーサネットにおける物理的な接続を行なう部分である.

ネットワーク層は,IPプロトコルの属する層で,何に対して通信をおこなうのかといったデータの送り先を明確にする部分.トランスポート層は,通信を行う両者の間でポートの情報交換を管理する事と,データ転送の方法決める階層.この層にはTCPとUDPという機能の異なる2つのプロトコルが存在し,データの信頼性と速度の面で使われ方が異なる.TCPプロトコルは,送受信するデータの信頼性を高めるため,データの転送速度が遅い.これに対してUDPプロトコルは,送受信するデータの信頼性は低いが,データの転送速度が速い.

アプリケーション層は ,TCP/IPにおけるユーザーに対して直接的なインターフェイスを提供する部分 . この層に属する代表的なプロトコルとして ,HTTP ,FTP ,SM

TP, POP3, TELNET, DNSなどがあげられる.

#### ソケット通信について

カリフォルニア大学バークレイ校で AT&T 社ベル研究所が開発したオペレーティングシステムである UNIX を拡張しようという研究が行われ,便利で高機能の UNIX (BSD UNIX)が開発されていた.その研究の一環として UNIX から通信ネットワーク,特に TCP/IP を利用した通信プログラム を容易にするとして開発されたもので、ネットワークを介して他のコンピュータに接続するときの接続の端点,データの入出力口をソケットと呼ぶ.プログラムから TCP/IP を利用する際に利用されるため TCP/IP のプログラミング上の大きな特徴といえ,TCP/IP 通信をソケット通信と呼ぶこともある.ソケットは,コンピュータ間の通信もファイルへの読み書きとしてプログラミングができないかということを目標にして開発された.

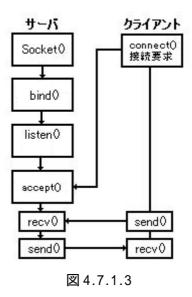
電話には電話番号あるため他の特定の電話機につなげることができる.これがPCの場合,ネットワーク上の複数のPCの中の特定のPCに接続するためには"電話番号"が必要である.このPCの"電話番号"に相当するのが,IPアドレスとポート番号である. IPアドレスとは,ネットワーク上に存在する複数の各PCに割り当てられた番号で,同じ番号をもったPCは,ネットワーク上に一つしか存在しない.このIPアドレスは32ビットの数字(例:192.168.0.2)で表記され,通常はネットワーク名やマシン名をドットで区切って表記される.また,ネットワークの大きさに応じてクラスAからクラスCまで三つのクラスに分類されている.その他にもクラスD(マルチキャスト用),クラスE(実験用)といったものも存在する. ポート番号は,IPアドレスを電話番号だとするとポート番号は内線番号に相当し,一つのPCで複数の通信を行う場合,どのプログラムのた

めの通信なのかを区別し仕分けするための16ビットの整理番号のことを指す.

遠隔操作時に通常の無線操縦ではなく、インターネット(無線 LAN)を用いたのは、以下のような利点があるためである。一つめは法的規制のない通信可能であること。無線 LAN は電波を発する立派な無線局にもかかわらず、電波免許不要な周波数帯の電波を使用するため免許取得の必要がない。二つめは、遠隔地からの操作が可能であること。インターネットを介しているため、かなりの遠隔地からでも操縦ができる。三つめは、WEB カメラなどを通して操作するロボットの状況を知ることができることである。

#### ・ソケット通信の実装

図4.7.1.3のサブルーチン見取り図を参考に以下のプログラム Socket(),connect()の警備ロボットへの実装について述べる.



```
result=listen(svSocket,1);
if(result==SOCKET_ERROR)
                    printf("Server : listen error¥n");
Break();
exit(-1);
          }
          do{
                    int adlen=sizeof(clAddress);
printf("待機中");
                    clSocket=accept(svSocket,(struct sockaddr
             *)&clAddress,&adlen);
                     if(clSocket==INVALID_SOCKET)
                              printf("Server: accept error¥n");
Break();
exit(-1);
                    }
  do{
  resul=trecv(clSocket,(char*)&temp,sizeof(double),0);
  if(result==0){
    printf("Server:clSock:closed\n");
          Break();
                                                                                            break;
  }
  else if(result==SOCKET_ERROR){
    printf("Server:clSock:recv error\u00e4n");
                                                                                  Break();
                                                                        break;
     }
  }
void connect() {
          int à1:
          a1=100;
WSADATA Data;
int result = WSAStartup (MAKEWORD(1,1),&Data);
  if(WSAStartup (MAKEWORD(1,1),&Data)!=0)
          printf("Client: winsock DII load error¥n");
exit(-1);
  }
          SOCKADDR_IN svAddress;
          SOCKET
                               svSocket;
          syAddress.sin_family = AF_INET;
       svAddress.sin_port = htons(45678);
u_long_svaddr=inet_addr("192.168.0.11")
memcpy(&svAddress.sin_addr,&svaddr,sizeof(u_long));
svSocket=socket(AF_INET, SOCK_STREAM,0);
if(svSocket==INVALID_SOCKET)
          printf("Client: socket error\u00e4n");
```

サーバ側のプログラムルーチンである,Socket()は図 4.7.1.3 に示すように,bind(),listen(),accept()の手順を踏むことで,サーバとしての待機状態を確保できる.サーバをある一つの店として,例えると bind()は営業登録,listen()は開店,accept()はお客を待つ,接待係りということになる.また,クライアント側はサービスを受けるお客だと例えると,connect()は接続要求を欲している客となる.この,クライアント側があるIPアドレスで運営を行っているサーバに対して接続要求を行うことで,recv(),send()の情報のやり取りが行われる.本研究では,サーバ側が警備ロボット,クライアント側が遠隔操縦を行う操縦者となる.具体的に,クラインアン側から要求があるまで,警備ロボットは待機を持続する.もし,クライアント側から,接続要求があれば,図 4.7.1.3 に示すように各ルーチンの手順を踏んだ後で,recv(),send()関係が成り立つ.ここで具体的に,やり取りされる信号は,クライアント側に接続されたUSBジョイスティックの情報である.この情報をもとに,遠隔操縦を行う.

#### 4.7.2 USBジョイスティックについて

#### ・USBについて

本研究のロボットの遠隔操縦に用いているジョイスティックコントローラは,USBによって接続されている.USBは,Universal Serial Busの略で,Windows98以降標準でサポートされるようになったパラレルポートやシリアルポートに変わる新しいインターフェイスである.USBの特徴は,電源供給ピンを備えているため,プリンタポートなどと違い,バッテリ,ACアダプタなどの別途電源を必要としないことと,伝送速度がシリアルポートよりも速いことである.USBで,PCとの直接接続やセルフパワー・ハブに接続する場合,USBインターフェースに接続する周辺機器は,最大5V・500mAの電力を使うことができる.バスパワー・ハブのように USBバスから電源供給される場合は,最大5V・100mAまでとされている.また,転送速度に関しては,データの通信といった高速通信が必要な周辺機器には,最大12Mbpsまでの高速データ転送が可能であり,キーボードやマウス,ジョイスティックといった周辺機器に対しては1.5Mbpsまでの低速通信をおこなうこともできる.

複数のデバイスを接続することを前提に開発されたため,使い終われば,すぐに切り離すことができる点(プラグ&プレイ)も特徴の一つであるといえる.ハブ(Hub)と呼ばれる分岐点を追加することで最大 127 個の機器を接続することもできる. USB ケーブルは,PC などルート側(ダウンストリーム)に接続する A コネクタとハブなど USB 周辺機器側(アップストリーム)に接続する B コネクタを持っており,通常 USB ケーブルは,データ通信用とバス電源供給用の2本で構成されており,ケーブル自身の長さは最長で5mまでのばせる.

USBの転送処理は,単一のホストが複数のターゲットポーリングするという伝送のやりかたをとっている.物理的な接続方法は,10BASE-Tなどと同じようにハブを使ってツリー状に接続しているが,論理的にはホストと各ターゲットの間が1対1の接続と

なるような形態をとっている(図 4.7.2.1). USBの転送はISAバスでいうところの, CPUからメモリ I/O アクセスのみでおこなわれているようなもので,ホストがターゲットのエンドポイントと呼ばれる一種の FIFO メモリを読み書きするといった動作しかおこなわない.

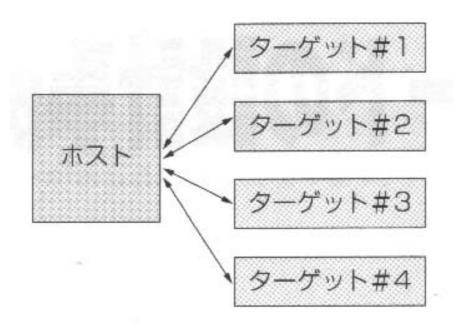


図 4.7.2.1

ホストがエンドポイントにデータを書き込んだり,エンドポイントのデータを読み出したりすると,ターゲットの CPU に割り込みが発生し,ターゲット側が次の動作をおこなうという仕組みになっている.また,エンドポイントにはデバイス制御用にもうけられているコントロールエンドポイントと呼ばれるもの以外はすべて単方向であるため,OUT(ホストからターゲット)と IN(ターゲットからホスト)のデータ転送が両方必要なときは,それぞれ別のエンドポイントを用意しなくてはならない.

USBの転送モードには、バルク転送、インタラプト転送、アイソクロナス転送などがあるが、データのやりとりは、すべてエンドポイントの読み書きだけでなりたっている。 転送モードの違いというのは、データエラー検出をおこなうか否か、ホストがそのエンドポイントを一定周期でアクセスするようにスケジュールを組むことが必要とされているか 否かといった違いだけである.

USBのことを理解するうえで必要となってくるものとして,パイプとエンドポイントというものがある.パイプとは,ホストとターゲットの間で仮想的につくられた1対1の 伝送路のことを指し,エンドポイントとは各パイプに対応してターゲットに用意されているアクセス権制御付きのバッファメモリのようなものをいう.これらのことをふまえて, 先ほどの図 4.7.2.1 をかくと図 4.7.2.2 のようになる.

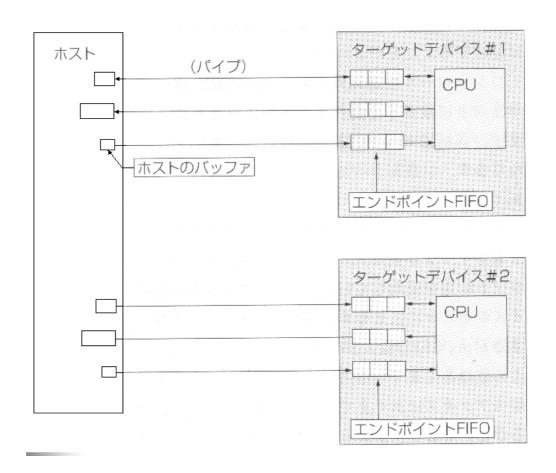


図 4.7.2.2

USBコントローラには, いくつものエンドポイントが存在し, それぞれのバッファサイズが何バイトで, どのような転送モードに対応できるかということは, USBコントローラにより異なるが 規格上はハイスピードデバイスでは1デバイスにつき最大15個 コ

ントロールエンドポイントを除く),ロースピードデバイスでは2個(同)のエンドポイントをもつことができる.

エンドポイントを使って実際にホストからターゲットに転送がおこなわれる場合,図 4.7.2.3 に示すようなことになる.まず,ホストはターゲットのエンドポイントにOUTのトランザクションを発行,続いてデータトランションが発行される.ターゲットのエンドポイント FIFO にデータが正常に入ると,ACK が返ってくる.

書きこみが終了すると,エンドポイントのアクセス権が自動的にターゲットのCPU側に渡る.ここでターゲットは,エンドポイントに入ったデータをすべて引き取り,エンドポイントのアクセス権をホストに返す.エンドポイントにデータが詰まっていたり,アクセス権がターゲットCPU側に渡されているときに,ホストがさらにデータを書きこもうとした場合には,データトランザクションのあとでターゲットからNAKが返り,ホストはNAKが返ってきたときにリトライする.

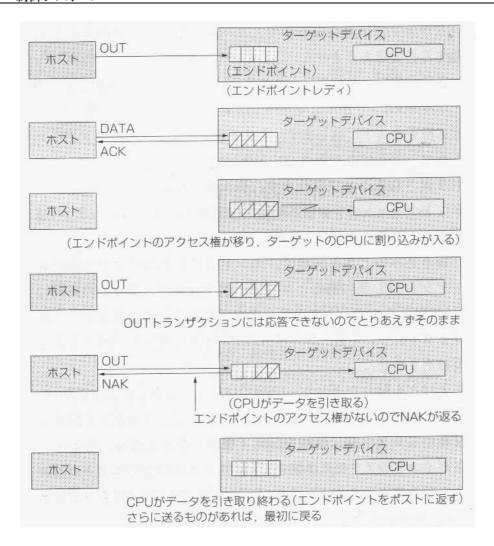


図 4.7.2.3 データ転送 (OUT) 模式図

## ・USBジョイスティックの実装

本研究は、図 4.7.2.4 に示す、JUSTY 製ゲーム用 USB ジョイスティックを使用する.まず、C 言語上からUSBデバイスへアクセスを行うために、マイクロソフト社が提供するDirectX SDKを用いる.DirectXとは開発者向けに提供されている開発キットのサンプル集である.このサンプル集上に新たに設けた、サブルーチン main()内において、下記のソースを追記するだけで、USB からのジョイスティック情報を抜き取ることができる.

g\_pJoystick->Poll(); g\_pJoystick->GetDeviceState(sizeof(DIJOYSTATE2),&js );

x=js.lX;
y=js.lY;
z=js.lRz;
cam=js.rgdwPOV[0];
S=js.rglSlider[0];

上記ソース上x, y, zの値は, 図 4.7.2.5 に示すジョイスティックx軸, y軸, z軸の実数値になる また、ソース上 cam,S は USB ジョイスティックの各ボタンに相当する . この実数値を4. 7. 1 で述べたソケット通信 send(svSocket,(char\*)&temp,sizeof(double),0);

の temp に反映することによって,警備ロボットへの指令を行う.



図 4.7.2.4

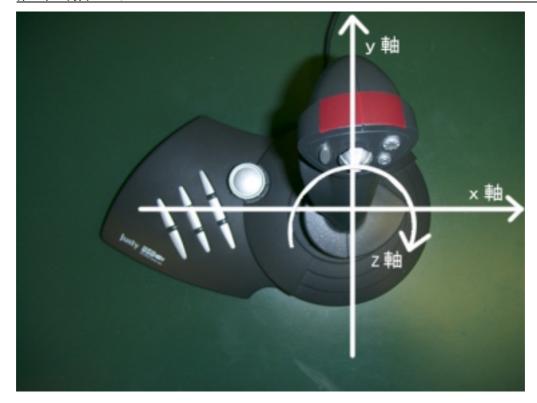


図 4.7.2.5

## 4.7.3 映像通信

カメラからの情報は、図6に示すように、警備ロボット側制御PCとは、独立した映像転送PC(ノート型PC)を搭載し、無線LANカードを通じて、映像転送PCと操縦者側PC間で映像伝達を行う、映像通信は、アプリケーションソフトであるマイクロソフト社、「ネットミーティング」を用いる。

## ・ネットミーティングについて

「ネットミーティング」を用いる理由として,IPアドレス間での通信が可能であり, 図 4.7.3.1 に示すように,お互いのアプリケーション上で映像と音声を同時に送受信できる.また,誰でも手に入るために,ソフトウェアの汎用性が高い点が挙げられる.故に本 研究では,このアプリケーションを,コミュニケーションツールとして用い,警備中の映像をリアルタイムで操縦者側の転送事に利用する.



図 4.7.3.1 ネットミーティングの様子

#### 4.7.4 WEBカメラのサーボ制御

本研究は、図 4.7.4.1 に示すように、ロボットを遠隔操縦する際の操縦者の"目"となる Web カメラの首を動かすためにラジコン用のサーボを使用した・サーボ(Servo)とは、ラテン語で奴隷を意味する Servus が語源であり、指示された命令通りに動くように構成された制御機構のことを指す・このサーボは、小型DCモータとポテンションメータ(可変抵抗器)、ギヤ、制御回路を小さくまとめたもので、このユニット自体が一つの完結したサ

# ーボシステムとなっている.

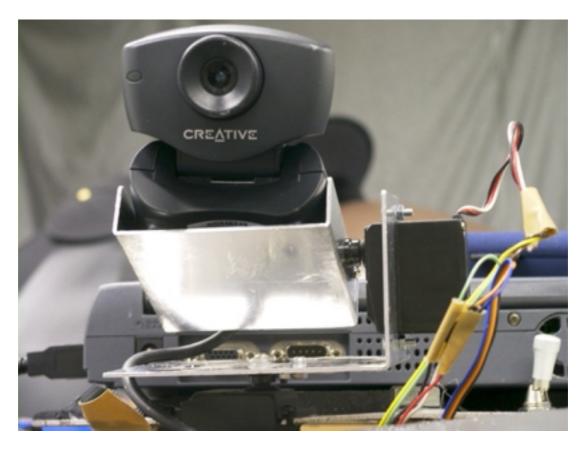


図 4.7.4.1



サーボを動かすには , 制御信号のパルス波を流す必要がある (図 4.7.4.2). サーボの標準的なものは , 制御信号のパルスの周期は約 20[ms]で , サーボの標準的なものは , 制御信号のパルスの周期は約 20[ms]で , パルス幅

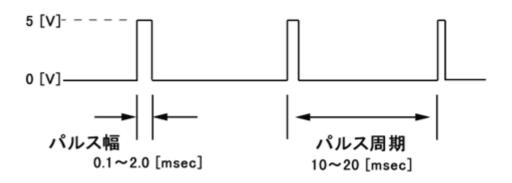


図 4.7.4.2

が約 1.5 ms のときに中立の位置  $, 1.5 \pm 0.5 \text{ms}$  で時計回り , もしくは反時計回りに振り切

れる.この回転の位置情報は,パルス幅のみに依存するため,たとえパルス周期が長くなっても,サーボは中立の位置の戻ろうとするだけとなる.

## ・実装

本研究ではこのサーボをロボット用インタフェースボードの D/A の出力チャンネルを 利用して制御を行う .

具体的に下記のソースに,制御の仕方を述べる.

## 4.8 ロボットの電源



図 4.8.1

ロボットの電源は,ユアサコーポレーションのユアサ小型シール鉛蓄電池・(NPH5-12) (図 4.8.1)を制御ボード用電源として2つと駆動用モータ電源地して1つの計3つを使用.公称電圧は12[V],定格容量は5[Ah](10時間率で),重量は約2[kg].

鉛蓄電池は,正極に二酸化鉛,負極に金属鉛,電解液に希硫酸を使用した蓄電池で,アルカリ蓄電池に比べて比較的安価でバランスのとれた特性を持つ.

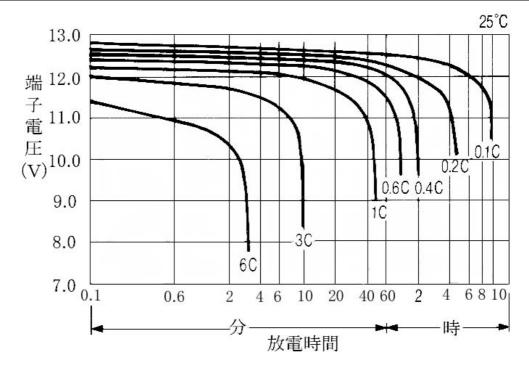


図 4.8.2 放電時間と電圧特性

図 4.8.2 は , 各放電電流の放電時間と電圧特性を示すグラフである . グラフ中の「C」は蓄電池の定格容量を A h を表わした数値で , 例えば , 放電率が 0.05C の場合は ,

0. 05 \* 5 = 0.25[A]

となり,放電率1Cは5[A]での放電を示す.

第五章 まとめ 73

# 第五章 まとめ

本文は,屋外警備における人間のサポートを目的とした警備ロボットを作製し,そのロボットの機能と制御システムの構成についてまとめたものである.本章では,この作製したロボットの機能についての動作実験の結果と今後の課題,及び,展開について述べる.

#### 5.1 研究成果と考察

人間の屋外警備のサポートを目的に,二つの機能を持たせたロボットを使って実験を行なってきた.その結果,警備を行なう場所までの移動や巡回に使用する遠隔操縦に関しては,本システムで行なうことができたが,実際の警備を行なう際には,以下の問題を解決する必要があると思われる.遠隔操作時に於ける問題として一つめは,遠隔に置ける操縦範囲が狭いということである.現在の操縦範囲でも,建物(例えば住宅など)の中から,その周りを警備・巡回することは可能だが,大きな施設(工場など)や完全な遠隔地からの操縦となると現在の操縦範囲では,それに遠く及ばない.二つめは,屋外走行時の操縦者の(ウェブカメラを通しての)視界の揺れが激しく,見づらかったことである.ロボットを移動させるたびに激しく揺れる画面を見せていては,サポートすべき人間を疲れさせてしまいかねない.しかし,これは関しては,駆動輪,及びキャスターにダンパーなどのクッションを入れるなどの対処法がとることができる.三つめは,ウェブカメラを通しての視界が狭かったことである.ロボットの移動を行うときや遠隔での監視を行なう場合,視界は広いほうが良いと思われる.これに関する対処として,より広角のカメラを使用するなどの方法がとれる.

自律モードにおける自動警備の問題としては、移動する物体を人かどうかを判断する

第五章 まとめ

74

すべがないため,実際の警備に実用的であるとは言い難かった.

## 5.2 今後の展開

今後の展開としては、遠隔操作の範囲拡大などの前節 5 . 1 で対処法が上がらなかったものを解決し、より人が扱いやすく、稼動範囲の広い警備ロボットをめざす.

参考文献 75

# 参考文献

Interface 1998 年 10 月号 特集インターネット家電時代の Ethernet と TCP/IP 活用法

> Interface 2000 年3月号 特集 USBターゲット機器の設計法

極める Visual C++ 基礎にて初歩にあらず 吉田弘一郎・著 技術評論社

トランジスタ技術 2000年11月号 特集 メカトロニクス・センサ活用法

謝辞 76

# 謝辞

本論文は筆者が、高知工科大学知能機械システム工学科に在学中において行なった研究をまとめたものである。本研究を行なうに当たり、ご指導下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深く感謝致します。また、お忙しい中、わざわざ筆者のために時間を割いて、ご助力いただいた同研究室の大学院生の浜口和洋氏、野村和寿氏にも深く御礼申し上げます。そして、研究中に様々なアドバイスをして下さった、同研究室の皆様に厚く御礼申し上げます。