

Abstract: *central vision*: 視野中心視, *peripheral vision*: 視野周辺視, *self-controlled robot*: 自律移動ロボット  
**中心視と周辺視機能を有した自律移動ロボット**の効率を比較する実験を行った。自律移動ロボット「ケペラ」(AAI社製)を用いて、周辺視機能を付加した場合と周辺視機能がない場合の特性を一つの迷路の通過時間で評価した。この結果、周辺視機能を付加したケペラは周辺視機能がないケペラに比べ迷路通過時間が有意に短縮した。  
 山ノ井 高洋\*・大西 真一\*・エリー・サンシェーズ\*\*

**A Self Controlled Robot with the Central and the Peripheral Visions**

*central vision*: 視野中心視, *peripheral vision*: 視野周辺視, *self-controlled robot*: 自律移動ロボット  
**Takahiro YAMANOI\*, Shin-ichi OHNISHI\* and Elle SANCHEZ\*\***  
 自律移動ロボット「ケペラ」(AAI社製)を用いて、周辺視機能を付加した場合と周辺視機能がない場合の特性を一つの迷路の通過時間で評価した。この結果、周辺視機能を付加したケペラは周辺視機能がないケペラに比べ迷路通過時間が有意に短縮した。  
**Abstract**

The human vision has a function of the central and the peripheral visions. We applied this function to a self controlled robot *Kaepera*. In order to compare with and without the function, we measured passing time of each robot through a labyrinth. *Kaepera* with this function proved to be effective on passing through a labyrinth.

人間の視覚には中心視と周辺視の機能がある。我々が自律移動ロボット「ケペラ」(AAI社製)にこの機能を適用することで、効率の良い自律走行が可能となるか検討した。本研究では小型自律移動ロボット「ケペラ」(AAI社製)を用いて、周辺視機能を付加した場合と周辺視機能がない場合の特性を一つの迷路の通過時間で評価した。この結果、周辺視機能を付加したケペラは周辺視機能がないケペラに比べ迷路通過時間が有意に短縮した。

**1. はじめに**

我々は、ヒトの視覚機能の一つである中心視機能と周辺視機能をロボットの視覚センサに適用することで、効率の良い自律走行が可能となるか検討した。本研究では小型自律移動ロボット「ケペラ」(AAI社製)を用いて、周辺視機能を付加した場合と周辺視機能がない場合の特性を一つの迷路の通過時間で評価した。この結果、周辺視機能を付加したケペラは周辺視機能がないケペラに比べ迷路通過時間が有意に短縮した。

**2. 周辺視機能**

**2.1 視野**

視野 (visual field) とは頭部と眼球を固定して1点を凝視 (fixation) したときに見え得る外

\* 北海道大学工学部電子情報工学科  
 \*\* Division of Electronics and Information Engineering, Faculty of Engineering, Hokkai Gakuen University  
 \*\*\* Laboratory of Medical Information, Faculty of Medicine, Aix-Marseille II

界の全範囲であり、単眼視野 (monocular visual field) と両眼視野 (binocular visual field) に分類することができる。両眼視野のうち、視野がオーバーラップしない左右両端の実質上の単眼視野は耳側半月視野 (temporal half-moon field) と呼ばれている。また、頭部を固定して眼球のみを動かして見え得る範囲は動眼視野といわれる。これからの説明では、視野はすべて頭部と眼球を静止させた状態で中央の凝視点を注視したときに右眼単眼視で見え得る外界の全範囲、即ち右眼単眼視野をさしている。

## 2.2 周辺視野と周辺視

周辺視野 (peripheral visual field) は中心視野 (central visual field) に対応する概念であり、周辺視野や中心視野における視機能をそれぞれ周辺視 (peripheral vision) や中心視 (central vision) という。周辺視は間接視 (indirect vision)、中心視は直接視 (direct vision) といわれることもある。周辺視—中心視という表現は視野—網膜部位の対応関係からきているのに対し、間接視—直接視という表現は主体的な観察や注意の仕方にポイントをおいている。また、系統発生的、個体発生的にも周辺視系は中心視系に先行するといわれる。

## 2.3 構造的レベルにおける周辺視と中心視

単眼視野の中で、広義の中心視野の全視野に占める割合は、個人差を考慮しても、たかだか2.5%程度にすぎない。それに対して周辺視野は、およそ97.5% (面積比で約40倍) の広い視野の範囲を占有しているといわれる。視野内の視空間の表面反射特性をもった対象 (distal object) は、ほぼ2次元的広がりを持ち、切手大の面積を有する眼底網膜 (retina) 上に結像される。そして外界と網膜との間には眼の光学系が存在する。したがって外界は近似的には左右反転 (reverse)、上下逆転 (invert) して網膜上に写像されることになる。右眼右半視野 (temporal visual field) と左半視野 (nasal visual field) はそれぞれ右眼鼻側網膜 (nasal retina) と耳側網膜 (temporal retina) に、右眼上半視野 (superior visual field) と下半視野 (inferior visual field) は、それぞれ右眼下半網膜 (inferior retina) と上半網膜 (superior retina) に投影されることになり、視野と網膜の位置の相互関係は少なくとも光学的には1対1対応の関係にあることになる。

Polyak は、網膜視細胞層の組織学的分類に基づいて、視野を3つの中心視野と4つの周辺視野に分割した。厳密に Polyak は網膜上の分割を行ったのであるが、それに対応した視角 (visual angle) に換算した。図2.1に Polyak に従って描いた概念分割図を示す。視軸から、半径視角2.5° (中心窩, fovea), 4.2° (傍中心窩, parafovia), 9.2° (遠中心窩, perifovia) に対応する3つの中心視野と、14° (近周辺部, near periphery), 24.2° (中周辺部, middle periphery), 45~53° (遠周辺部, far periphery), 53°~ (鋸状縁, ora serrata, 外周辺部, extreme

periphery)に対応する4つの周辺視野である。

このような視野の分割の組織学的・生理学的根拠は錐体 (cones)、杆体 (rods)、及びそれらとの層的結合関係における神経節細胞 (ganglion cells) のトポグラフィカルな密度分布に関するデータや、von Kries 以来の2重機能説にその拠所が認められ、いわば構造的、末梢的レベルからながめた生理的視野といてよい。しかし、これらの視野は中枢過程を含む機能的レベル、具体的には精神物理学的データに基づく心理的視野や等感覚視野とは必ずしも一致しない。

本研究では、視力データを勘案して、視角約2.5°(直径5.0°)の中心窩網膜、すなわち狭義の中心視野で注視することを中心視、それ以外の網膜部位でみることを周辺視ということにする。

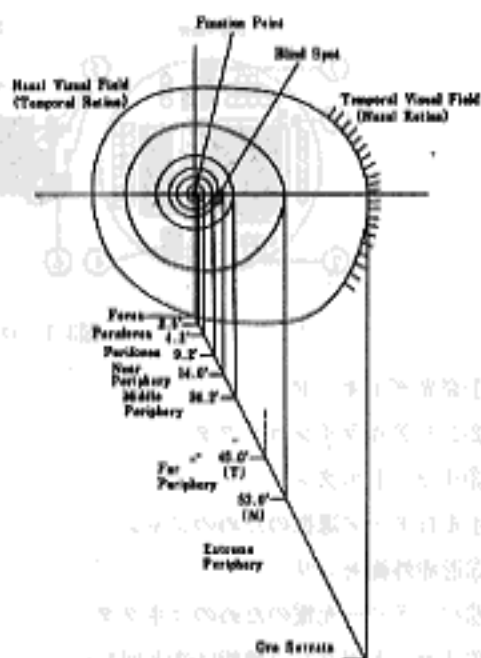


図2.1 右眼単体での視野

#### 2.4 受容野レベルにおける中心視と周辺視

受容野レベルで考えると、中心視機能での受容野は単体では小さいが細かく集まっており、周辺視機能での受容野は単体で大きい。すなわち、中心視機能では物体を細かく明確に見ることができのに対し、周辺視機能では移動物を判断するのに優れている。本システムでは、受容野の大きさをセンサレベルの大きさとみなした。

### 3. 小型自律移動ロボット「ケベラ」

#### 3.1 「ケベラ」とは

ケベラは本来研究用として設計されたものである。ケベラは、軌跡追跡や障害物の回避や生態(動物など)の過程の仮説などのシミュレーションが発達したアルゴリズムを実世界に実現する。

#### 3.2 ケベラとその付属品

##### 3.2.1 ケベラの概要

ケベラの概要を図3.1に示す。

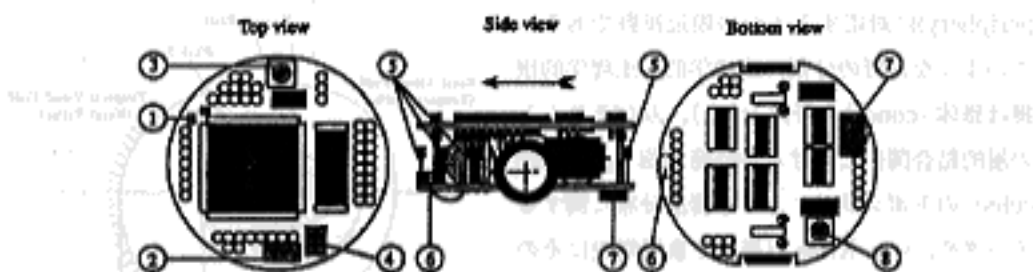


図3.1 ロボットの概要図

- ①発光ダイオード
- ②シリアルラインコネクタ
- ③リセットボタン
- ④走行モード選択のためのジャンパー
- ⑤近赤外線センサ
- ⑥バッテリー充電のためのコネクタ
- ⑦リセットボタン（機能は③と同じ）

### 3.2.2 ON-OFF バッテリースイッチ

ロボットのバッテリーのスイッチは図3.2にあるようにON（“1”）、OFF（“2”）である。ONのときロボットはNi-Cd（ニッケル-カドミウム）の内蔵バッテリーによって動力を得る。OFFのときバッテリーは切断され、ロボットはSシリアルラインコネクタによって電力を得ることができる。

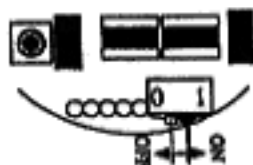


図3.2 ON-OFFスイッチの物理制御と回路図

### 3.2.3 ジャンパー、リセットボタンと設定

ジャンパーにより走行モードの選択をする（図3.3）。以下のモードをセットすることができる。

MODE 0：デモンストレーションモード

Braitenberg による障害物の回避のアルゴリズム。

MODE 1：通信速度が 9600 baud のシリアルリンクを用いたシリアルコミュニケーションプロトコルによる、ロボットのコントロールのためのモード。

MODE 2：MODE 1 と同じだが通信速度が 19200 baud のもの。

MODE 3: MODE 1 と同じだが通信速度が 38400 baud のもの。  
 MODE 4: ユーザアプリケーションモード(ユーザーアプリケーション)モード。EPROM に格納されたアプリケーションをスタートする。

MODE 5: ダウンロードモード。ロボットは移動するためのプログラムを待ち受ける。(S.フォーマット, 9600 baud)

MODE 6: MODE 5 と同じだが通信速度が 38400 baud のもの。

MODE 7: ロボットの機能上のテスト。相次ぐテストの結果はシリアルリンクで得られる。(9600 baud)

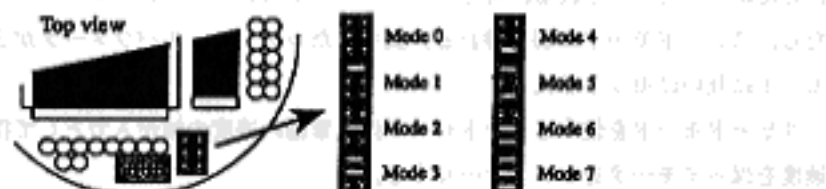


図3.3 ジャンパーの位置と設定

シリアルリンクのセットアップは常に8ビットである。1つ目がスタートビットで2つ目がストップビットである。パリティはない, baud レートを変えるだけである。ジャンパーによるセットアップはいつでも変更することができる。もし、ロボットが動いていると、セットアップの効果を得るためにリセットが必要である。リセットボタンはロボットをリセットするためにいつでも使うことができる。

### 3.2.4 Sシリアルライン

Sシリアルラインは TTL の動作レベル (0-5 V) を伴った非同期のシリアルラインである。インタフェースは標準的な RS 232 C のポートに接続する。このインタフェースのパッケージにはインタフェース/チャージャーモジュールが含まれている。線の長さは、2 m が限界である。

### 3.3 モータとモータの制御

どの車輪も 25:1 の減衰ギヤを通した 2つの DC モータにより動かされる。インクリメンタルエンコードがモータの軸にあり、モータ 1 回転で 24 パルスを与える。車輪が 1 回転するのに 600 パルスに分解されていて、ロボットが 1 mm 進むために 12 パルスとなっている。ケベラのメインプロセッサは、モータのパワーをコントロールしたり、インクリメンタルエンコードのパルスを読んだりできる。割り込みルーチンは、インクリメンタルエンコードのすべてのパル

スを見つけ出し、車輪のポジションカウンタを最新のものにする。

2つのDCモータは、メインプロセッサの割り込みルーチンにより実行されたPIDコントローラによって制御される。このコントローラのすべての条件（比例、積分、微分係数）は、定数として結びつけられる。条件に対応するコードを次のようにする。

$K_p$  : 比例

$K_i$  : 積分

$K_d$  : 微分係数

モータコントローラは、2つのコントロールモードを使用することができる。スピードモードとポジションモードである。活動中のコントロールモードは、受け取られたコマンドの種類にしたがってセットされる。もしコントローラがスピードコントロールのコマンドを受け取ったら、スピードモードに切り替わる。異なるコントロールパラメータが2つのコントロールモードに互いにセットされる。

スピードモードを使うとコントローラには車輪の速度の値が入力として得られ、この車輪の速度を保ってモータをコントロールする。

ポジションモードを使うとコントローラには車輪による目標の位置を入力として得られ、加速して最高速度となる。この値を使うと、コントローラは最高速度になるまでモータを加速させる。そして、目標の位置に達しようとする減速させる。この移動は図3.4に描かれている。このコントローラによる入力の値とコントロールモードはどのような動きをしていても変更することができる。コントローラはポジションモードの新しいプロファイルを実行したり最新のものに換えたりする。また、スピードモードの新しい値を更新し、車輪のスピードをコントロールする。コントローラの状態はアクティブコントロールモードで示される。

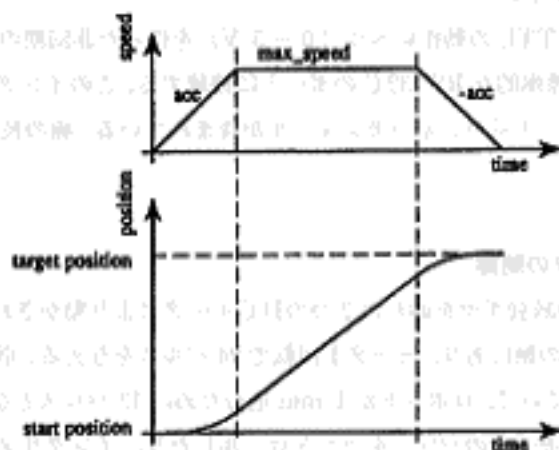


図3.4 ポジションモードにおけるモータの動作

### 3.4 近赤外線センサ

#### 3.4.1 近赤外線センサについて

8つのセンサがロボットのマわりに配置されており、図3.5に示されるように番号がついている。これらのセンサは赤外線光線の発光と受光の機能を組み込んでいる。より多くの情報をセンサによる調査で参照する。正確な部品の名前はSIEMENSのSPH 900である。このハードウェアは2つの測定法をもっている。普通の環境での光（このときは受光のみ）と、光が障害物によって反射される場合である。どのセンサの出力も10ビットのA/Dコンバータによってアナログの値を変換する。次の2つのセクションでは、この10ビットの値の意味するところを図解する。

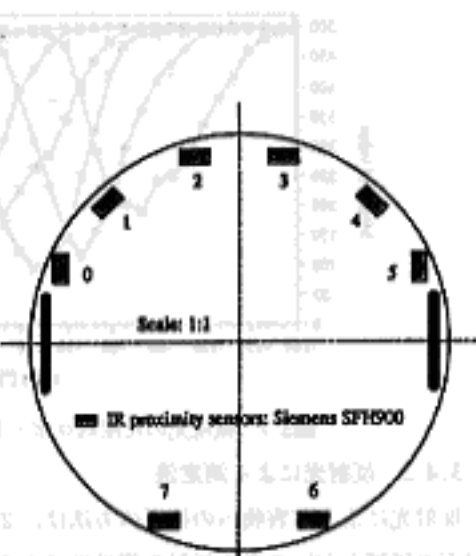


図3.5 近赤外線センサの配置図

#### 3.4.2 環境光の測定法

図3.6は環境光の尺度対光源からの距離の図である。図を見ると光源の強度が強くなると測定される値は減少するということがわかる。図3.7は環境光の尺度対ロボットの前向き方向と光の方向とのなす角の図である。これらの測定はすべての光源の距離、色、強度、位置などのいろいろな要素から非常に強く影響を受ける。これら2つの図はセンサの応答の全体的な形だけを示している。

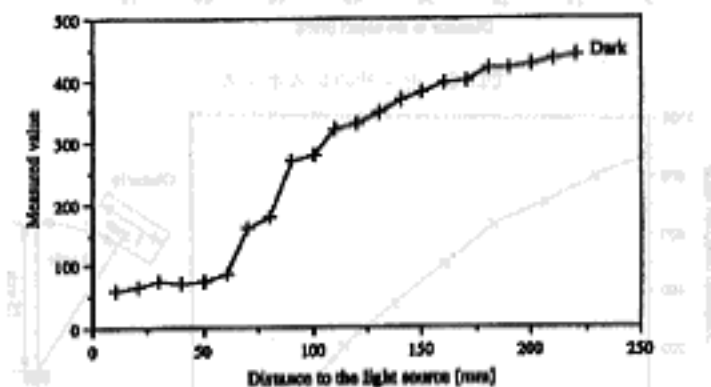


図3.6 環境光の尺度対光源の距離

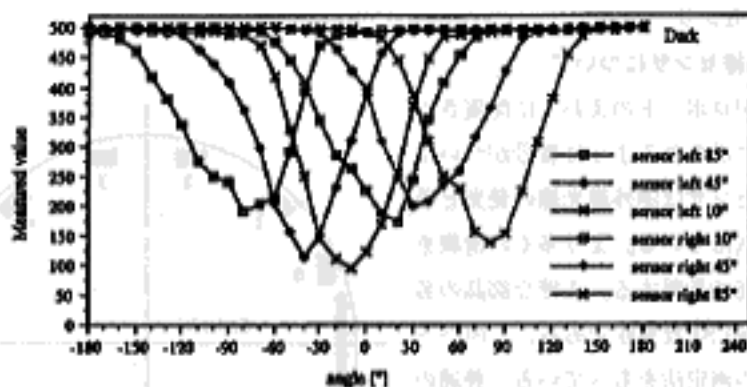


図3.7 環境光の尺度対ロボットの前向き方向と光の方向のなす角

### 3.4.3 反射光による測定法

反射光による障害物への接近の方法は、2つの主要な要素による。それは、障害物（色や反射面の種類など）の光の反射や環境光である。図3.8はセンサのレスポンス測定の実例を示したものである。センサの指向性は図3.9に描かれている。これらのセンサは、視野の範囲がとて

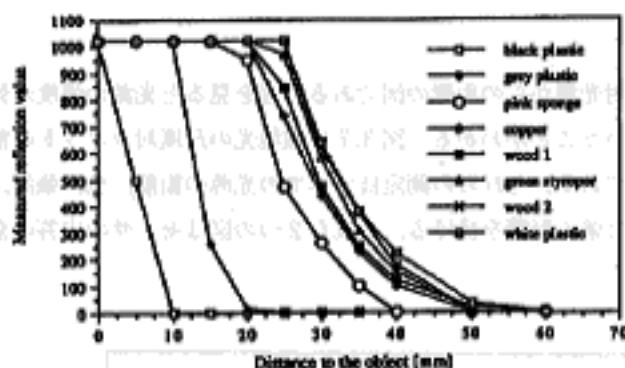


図3.8 センサのレスポンス

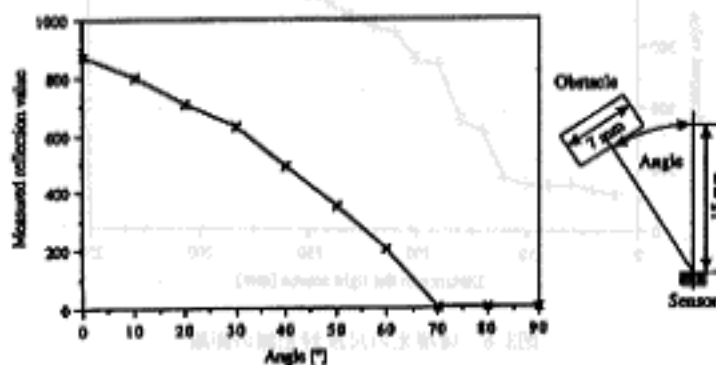


図3.9 センサの指向性



も広い。センサの特性は図 3.10 に示すように大きなレンジで変えることができる。

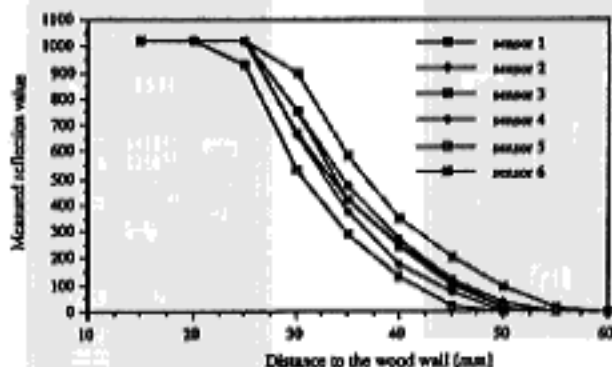


図3.10 センサの特性

## 4. LabVIEW

### 4.1 LabVIEW とは

LabVIEW は、最新の C、BASIC、ナショナルインストルメンツ社の LabWindows/CVI などの開発環境とよく似たプログラム開発環境だが、LabVIEW とこれらのアプリケーションとの間には、重要な相違点がある。他のプログラミングシステムではテキストベースの言語を使用してコード行を作成するのに対し、LabVIEW ではグラフィック的なプログラミング言語 G を使用してブロックダイアグラム形式のプログラムを生成する。

LabVIEW は、C や BASIC と同様、汎用のプログラミングシステムであり、どのようなプログラミングタスクにも対応できるように、総合的な関数ライブラリが用意されている。LabVIEW には、データ収録、GPIB とシリアル計測器の制御、データ解析、データ表示、データ記憶などのためのライブラリがある。このほか、従来のプログラム開発ツールも含まれているので、ブレークポイントを設定したり、実行状態を動画化してプログラム内のデータフローをチェックしたり、シングルステップでプログラムを動作させることができ、プログラムの開発やデバッグがいっそう容易に行える。

LabVIEW は、C や BASIC と同様、汎用のプログラミングシステムであり、どのようなプログラミングタスクにも対応できるように、総合的な関数ライブラリが用意されている。

### 4.2 LabVIEW の動作

LabVIEW は汎用のプログラミングシステムであるが、データ収録や計測器制御用に設計された専用の関数ライブラリや開発ツールも備えている。LabVIEW プログラムは、外観と動作状態が実際の計測器と似ているため、バーチャルインストルメンツ (仮想計測器) VI と呼ばれているが、VI は従来の言語プログラムの関数によく似ている。VI は、対話式ユーザインタ

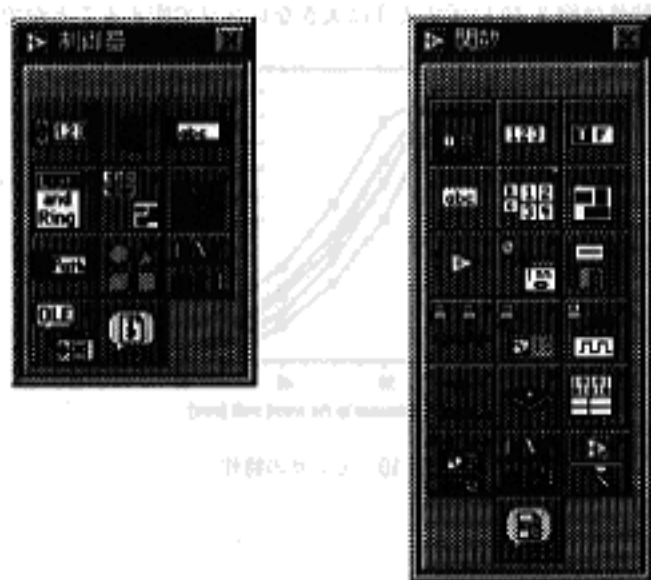


図4.1 Labview フロントパネル

フェース、ソースコードの役割を持つデータフローダイアグラム、上位 VI から VI を呼び出せるようにするためのアイコン接続から構成されている。VI の構成をさらに詳しく説明すると次のようになる。

VI の対話式ユーザインタフェースは実際の計測器のパネルを模倣しているため、フロントパネル(図 4.1)と呼ばれている。フロントパネルには、ノブ、押しボタン、グラフ、その他の制御器や表示器がある。マウスやキーボードでデータを入力すると、コンピュータ画面に結果が表示される。

VI はユーザが G で作成したブロックダイアグラムから指示を受け取る。ブロックダイアグラムは、プログラミングすべきテーマに対する解答を図式的に表現したものである。またブロックダイアグラムは VI に対するソースコードとなる。

VI は階層的なモジュール構造になっている。そして、VI は最上位プログラムとしても他のプログラムのサブプログラムとしても使用できる。他の VI に属する VI はサブ VI と呼ばれる。VI のアイコンとコネクタはグラフィカル・パラメータ・リストと同様に機能するので、他の VI からサブ VI にデータを渡すことができる。

このような特徴をもつ LabVIEW は、モジュール式プログラミングの概念を採用し、この概念にしたがって作られている。アプリケーションを一連のタスクに分割し、さらにこれらを分割していくと、複雑なアプリケーションも一連の単純なタスクに分解できる。分解されたそれ

それぞれの部分タスクを実行するVIを作成し、これらのVIを別のブロックダイアグラムに組み込むことで、大規模なタスクを実現できる。最終的には、アプリケーションの機能を表現したサブVIの集合体が最上位VIの中に含まれた形となる。各サブVIは、アプリケーションの他の部分とは別に単体でも実行できるので、デバッグは非常に簡単になる。さらに、下位のサブVIで実行するタスクは複雑なアプリケーションに共通な場合が多いので、ある種のサブVI群を開発しておく、ユーザが作成する複雑なアプリケーションに対応できる。

## 5. 「ケベラ」へのヒトの視覚の応用

### 5.1 実験方法

前述のとおりロボットには全部で8個のセンサがついているが(図3.5参照)、この実験ではそのうちの前6個のセンサ(0~5)を用いる。便宜上、センサをそれぞれ次のように呼ぶ。

2と3のセンサ : front

1と4のセンサ : 45 degree

0と5のセンサ : 90 degree

このセンサの組み合わせはBraitenbergの組み合わせと同様に対称である。本研究ではセンサの感度をヒトの視覚機能と関連付け、“front”を中心視機能として考え、“45 degree”、“90 degree”を周辺視機能として考える。

次に、LabVIEWを用いて作成したプログラムについて述べる。4章で述べたとおり、プログラミング言語はGを用いているのでプログラム自体はダイアグラム形式で書かれる。ここで

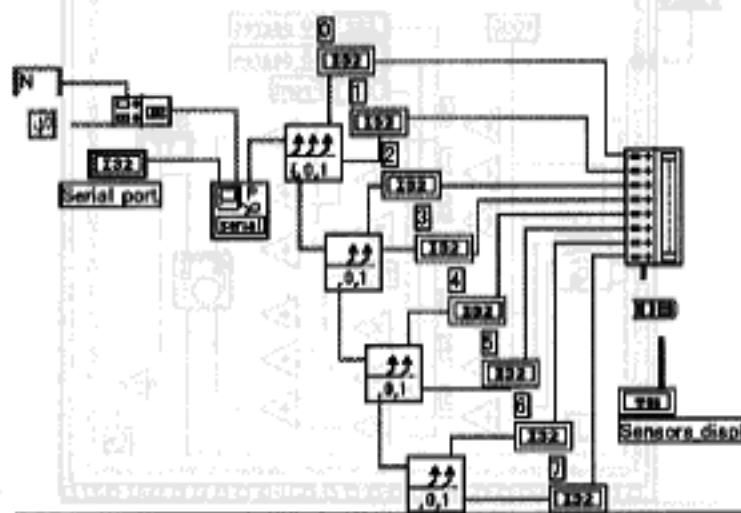


図5.1 センサVIのダイアグラム

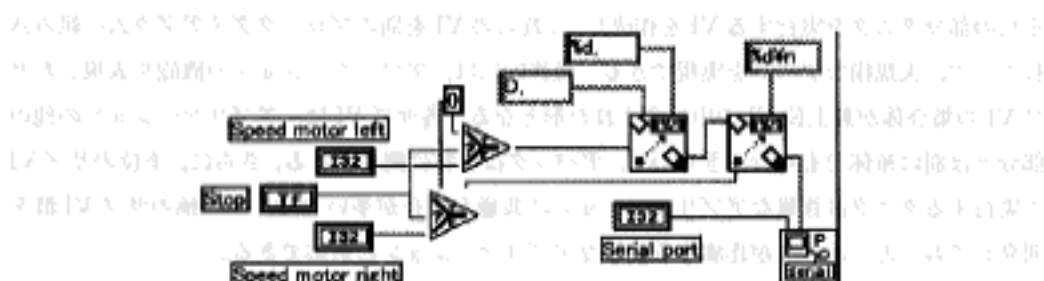


図5.2 モータ VI のダイアグラム

はいくつかある VI のうち、センサ VI、モータ VI、Braitenberg 衝突回避アルゴリズム VI について説明する。センサ VI は図 5.1 となるが、これは近赤外線センサからの入力値を処理するものである。具体的に言うと、センサの値はそのままでは処理できないので文字列として配列に書き込んでいくのである。

次にモータ VI であるが(図 5.2)、これは左右のモータを動かすための VI でモードはスピードモードを用いている(3.3 モータとモータの制御参照)。

最後に本実験の障害物回避アルゴリズムであるが、これは Braitenberg のアルゴリズムを用いる。そのアルゴリズムを実現したものは Braitenberg VI であり(図 5.3)、センサからの値を

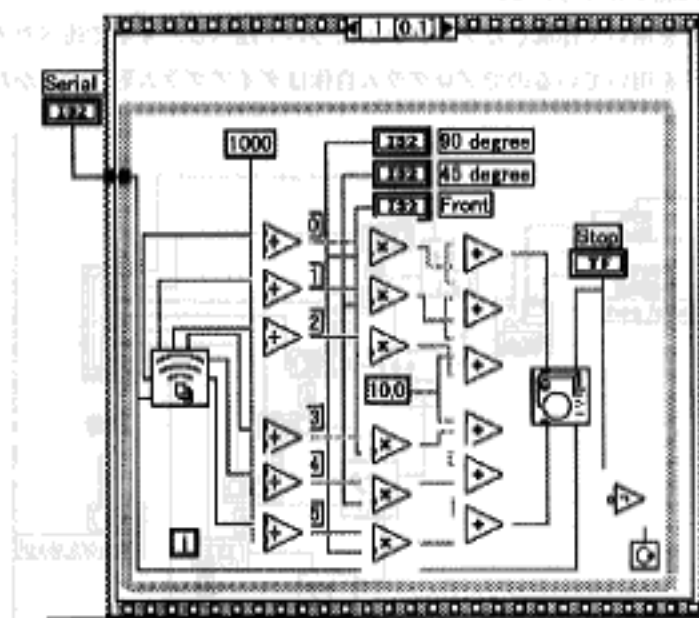


図5.3 Braitenberg の衝突回避アルゴリズムの VI ダイアグラム

処理してその結果をモータへ送っているものである。また、全てのVIのVI階層を図5.4に示す。

ここで、センサの感度をそれぞれ以下のようにする。センサの感度は実験1 VIのコントロールパネル(図5.5)で設定する。以後、本研究で取り扱うレベルの値は衝突回避を目的としているのですべて負の値である。

周辺視なしA	周辺視あり	周辺視なしB
front : レベル7	front : レベル7	front : レベル10
45 degree : レベル7	45 degree : レベル10	45 degree : レベル10
90 degree : レベル7	90 degree : レベル7	90 degree : レベル7

ここで、「周辺視なしB」は「周辺視なしA」のセンサレベルを上げたものとした。ただし、90 degreeをレベル10とすると本実験で設計した迷路の構造上から迷路走行に支障があるのでレベルを7に下げた。これらの条件の下、図5.6に示す迷路を攻略させてみてそのスタートからゴールまでの時間を計測した。Braitenbergのアルゴリズムは本来障害物回避を目的としている。したがって後方のセンサを考慮していない。本研究でも回避アルゴリズムはこれを利用しているので後方の2つのセンサは迷路走行になんら関与していない。さらに、今回は

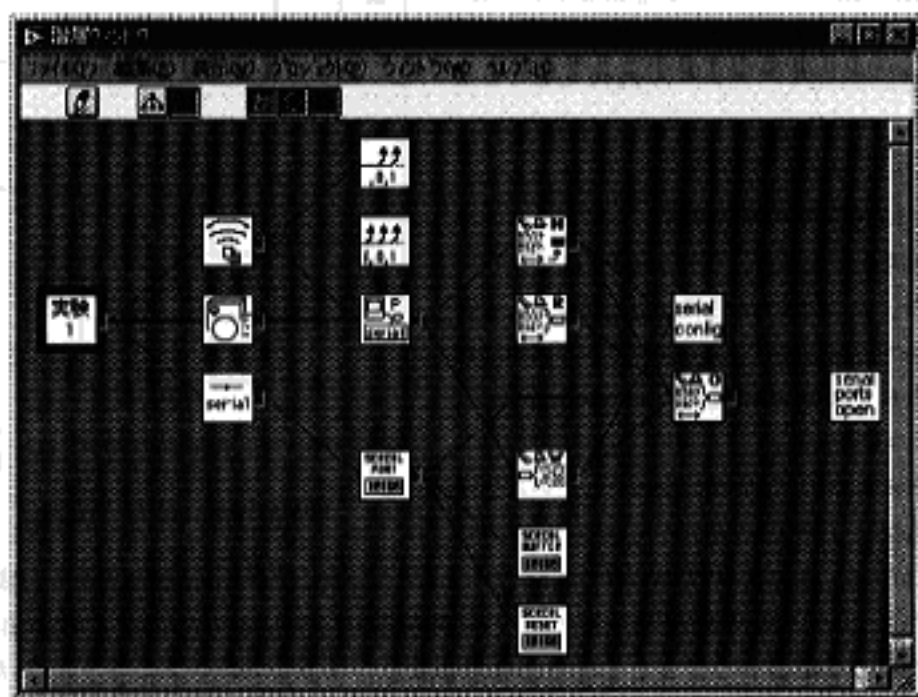


図5.4 本実験のVIの階層構造

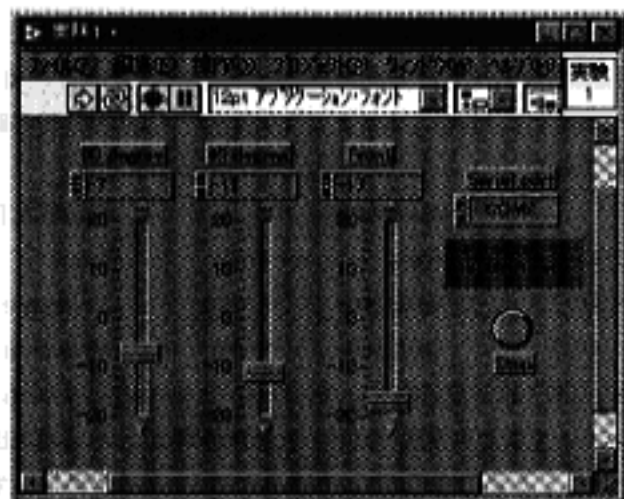


図5.5 実験1 VIのコントロールパネルのスクリーンショット

Braitenbergのアルゴリズムを流用したため、迷路走行のために必要な本質的なアルゴリズムを考慮していない。このため迷路もロボットが無限ループに陥ることのないものを設計した。

## 5.2 迷路攻略の実験結果

実験を行った結果、データは表5.1のようになった。この結果をもとに、周辺視なしAと周辺視あり、周辺視なしBと周辺視ありの対に対してそれぞれ片側検定を行うと、表5.2、表5.3のようになった。

それぞれのデータが同じ分布をしていると仮定するなら、このような差は1%以下の確率でしか起こらない。したがって非常に稀であるのでもととの分布に差があると結論できる。(表5.2)

見た目には周辺視の機能を付加したほうの通過時間の方が短いようであるが、統計的には、分布が同じ母集団のサンプルからのデータだと仮定して16.5%の確率で起こりうる。したがって、統計的に差があるとは言えない。(表5.3)

これらの実験結果を簡単にまとめると図5.7のようになる。

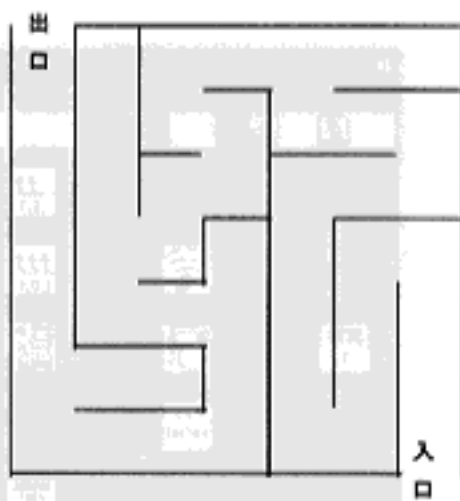


図5.6 実験に使用した迷路

表 5.1 周辺視機能の有無の下での迷路の通過時間

周辺視	センサレベル	通過時間 [秒]									
なし (A)	7,7,7	131	126	122	126	127	126	124	129	126	127
		127	126	131	124	123	130	124	124	120	122
あり	7,10,7	112	123	117	114	112	125	109	109	109	108
		106	108	106	107	110	106	105	107	121	105
なし (B)	10,10,7	118	118	107	120	118	117	106	105	105	119

表 5.2 「周辺視なしA」と「周辺視あり」に対する通過時間の平均の差の検定結果

	周辺視なし(A)	周辺視あり
件数	20	20
平均[秒]	125.75	110.95
標準偏差	2.99	6.06
統計量 (z)	9.7998	
P 値	0.0000	
t (0.01)	2.4671	
判定	有意	

表 5.3 「周辺視なしB」と「周辺視あり」に対する通過時間の平均の差の検定結果

	周辺視なし(B)	周辺視あり
件数	10	20
平均[秒]	113.40	110.95
標準偏差	6.43	6.06
統計量 (t)	1.0026	
P 値	0.1651	
t (0.01)	2.5669	
判定	有意差なし	

通過時間：周辺視機能なしA > 周辺視機能あり < 周辺視機能なしB  
 センサレベル：(7,7,7) < (7,10,7) < (10,10,7)

図5.7 周辺視機能の有無とセンサレベルとの関係

周辺視機能を持たせるとセンサレベルを上げた場合と同様な短い時間で迷路を通過することが出来るのがわかる。この結果、この迷路を効率よく攻略するために、周辺視機能は有効である。ロボットの動きを観察してみると、周辺視機能を持たない場合は角を発見できないのに対し、周辺視機能を持つ場合は角を発見しスムーズに曲がっていた。

### 5.3 考察

実験の結果より、周辺視機能を持っている方が効率よく迷路を攻略できるということが確認できた。この理由の一つは、ロボットが迷路を直進しているとき、どのセンサの値も不変であるが、左右に曲がる角に来たときは前方のセンサよりも周辺視機能として考えたセンサの方が先に値が変化するためである。別の見方をすれば、ロボットの走行を相対化して考え、ロボット自身を静止させて考えるとこのときに移動するのは迷路の壁ということになる。周辺視機能というのは移動物に敏感に反応するので、この感度が良いほど角を見つけることができる。

#### 5.4 今後の課題

本研究ではヒトの視覚機能のうち周辺視・中心視機能をロボットのセンサに応用した。この際に Braitenberg の障害物回避のアルゴリズムを流用した。迷路走行の汎用的なアルゴリズムでの検討は今後の課題である。

#### 6. おわりに

本研究の一部は、著者の一人山ノ井高洋の平成 11 年度後半から平成 12 年度前半のフランス国エクス・マルセイユ第 2 大学医学部医療情報研究室における在外研修中に行なわれた。在外研修を支えてくれた本学工学部の教職員ならびにエクス・マルセイユ第 2 大学の教職員に感謝する。予備実験に際し、協力してくれた平成 10 年度の山ノ井研究室の 4 年生、佐藤祐児君、直田智義君に感謝する。

#### 参 考 文 献

- [1] 井上泰典: LabVIEW グラフィカルプログラミング, 森北出版社, 1998
- [2] LabView ユーザマニュアル, NATIONAL INSTRUMENTS, 1998
- [3] 宇坂直行: 周辺視機能の精神物理学的研究, 風間書房, 1983
- [4] K-Team SA: Khepera USER Manual, Lausanne, 1995
- [5] E. Sanchez, P. Pierre and J. P. Aurrand-Lions: VITAU: Computerized Mobile Vehicle based on Fuzzy Logic Reasoning, for Integration and Autonomy of Disabled Persons, 5th Int. Symp. ORIA '94: "From Telepresence to Virtual Reality, Marseille, 349-356, 1994.