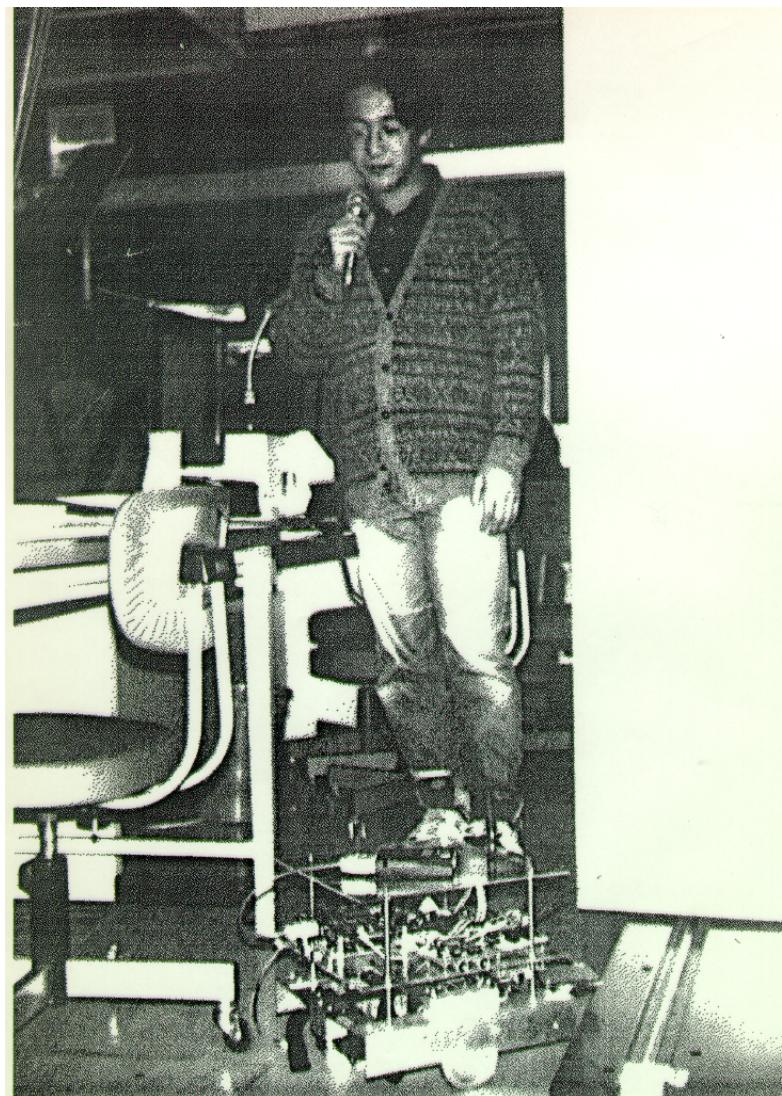


知的ロボットインターフェース製作ガイド

慶應義塾大学理工学部情報工学科3年



1 演習の目的

ロボットは、動く体を持った計算機としても捉えることができ、情報通信の分野でも重要な研究課題となりつつある。本演習では、ロボットの構築および、音声を用いた知的インターフェースの実装を通して、計算機のハードウェア、ソフトウェア、センサ、人工知能の基本技術の習得を目的とする。

2 演習の体制

期間 秋学期の半期を通して実験

班分 5名で一つの班とし、四つの班を作る。

3 演習課題

演習では、ロボットの作成、センサの作成、知的インターフェースの作成を行なう。最終的には、音声命令を受けとて行動するロボットを構築する。

実製作業は、大きく分けて三つに分かれる。ロボットのハードウェアの構築および、基本ソフトウェアの構築、知的インターフェースの構築である。各部を実現するにあたって必要となる作業を次に挙げる。

1. ロボットのハードウェアの構築

- マザーボードおよび、AD（アナログデジタル変換）ボード、DIO（デジタル入出力）ボードのセットアップ。OS(Linux) のインストール。ロボットのボディの組み上げ。駆動系の組み上げ。センサの作成。電源系の組み上げ。無線 LAN のセットアップ。

2. 基本ソフトウェアの構築

- モーターコントローラ用のプログラムの作成。エンコーダ（モータの回転数）の値を読み込むプログラム作成。DIO を用いてセンサの値を読み込むプログラムの作成。AD ボードを用いてセンサの値を読み込むプログラムの作成。マザーボードの音声入出力のセットアップ。

3. 知的インターフェースの構築

- 音声認識ソフトウェアのセットアップ。音声合成ソフトウェアのセットアップ。合成音声データの作成。音声認識ソフトおよび合成音声データを用いて音声対話プログラムを構築。ロボットの基本行動の構築。命令セットの作成。

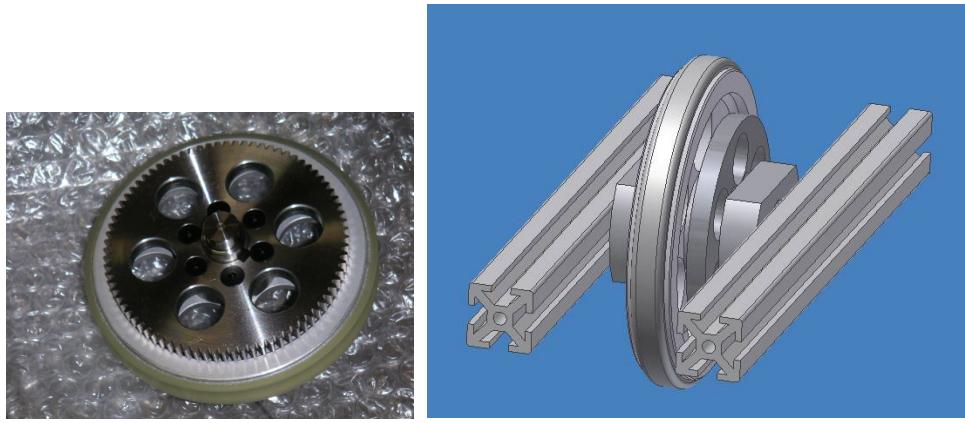


図 1: タイヤとタイヤの取り付け

最終的には、音声命令として必修命令とオリジナル命令を実装する。必修命令は 4.4.5 節で説明する。またオリジナル命令は各班独自で命令を考えロボットに実装する。これらの命令は、人間がロボットに音声で命令する際に用いる。

4 各種設計

4.1 ロボットのハードウェアの設計

4.1.1 ロボットのボディの組み上げ

ロボットは、二輪の駆動輪と一つのキャスターによる三輪移動ロボットを基本とする。筐体は、アルミのフレームと板の組み合わせで構築し筐体の形状などは、設計者の自由とする。図 1 にタイヤの取り付け例を示す。タイヤには既にギアが固定されている。モータは、タイヤの横の位置に固定する。

4.1.2 駆動系の組み上げ

モータの回転速度は、図 2 のモータコントローラおよびモータドライバによって制御する。モータコントローラはマザーボードと USB 経由で接続される。またモータコントローラには 24V 電源も接続される。モータコントローラとモータドライバ間は専用のケーブルによって接続する。

ここで、モータの回転速度の制御について簡単に触れる。モータの回転速度は、モータドライバが作り出す電圧によって変化する。モータに与えられる電圧は、PWM という手法によって変化する。PWM とは、Pulse Width Modulation の略である。名前の通り PWM は、パルスの幅を変化させることである。パルスの幅を変化させることにより、時間あたりの平均電圧が変化し(図 3)、モータの回転を制御することができる。モータドライバは、



図 2: 左からモータコントローラ、ドライバー、コントローラとドライバーの接続の写真

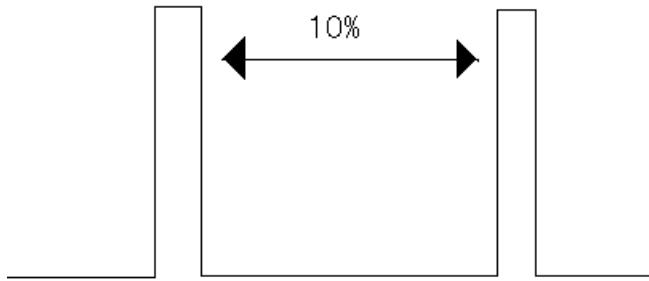


図 3: PWM

モータコントローラから与えられる命令に従って、モータへの PWM を変化させる。また、モータドライバは、モータへの電圧を 0V にすることによりブレーキをかける。

4.1.3 センサの作成

実験では、ロボットのセンサとして超音波距離センサのキットを四セット用意してある(図 4)。このセンサによって、ロボットの前後左右にある物体までの距離を測定することができる。マザーボードに距離の値を入力するためには、センサキットを改造し、デジタル信号を取り出す。もしくは、センサキットが output するアナログ電圧を取り出す方法でも可能である。

また必要に応じて、赤外線センサ、温度センサ自分で購入しセンサを取り付けることも可能である。

4.1.4 マザーボード、AD ボード、DIO ボードのセットアップ

ロボット上の計算機環境は、PC-104 バスを持つマザーボードに AD ボードと DIO ボードを装着したものである(図 5)。マザーボードには、電源 $\pm 5V$ と $\pm 12V$ を入力する。また、各種ケーブル(ディスプレイ、キーボード、マウス、シリアル、USB)を接続する。

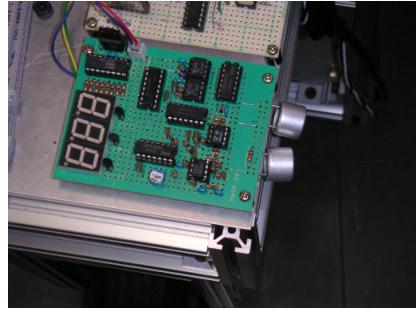


図 4: 超音波距離センサ



図 5: 左からマザーボード、DIO ボード、AD ボード

また AD ボードおよび DIO ボードのアドレスを設定しておく。各ボードは、このアドレスによりマザーボードからアクセスされる。具体的には、各ボードへ命令を送ったり、データの読み書きに使用される。設定方法は各ボードのマニュアルを参照のこと。

アドレスを設定した各ボードは、マザーボードの PC104 バスに接続する。PC104 バスは、マザーボード上に各ボードを積み上げて接続していくバスである。例、AD のアドレス 0x300、DIO のアドレス 0x340。

4.1.5 OS(Linux) のインストール

組み上げたマザーボードに Linux(kernel 2.4.20-8, RedHat 9.0) をインストールする。インストールは、フロッピーディスクでマザーボードを起動し、ネットワーク経由で行う。大まかなインストールの流れを次に示す。

1. BIOS の設定を変更し、ロボットのマザーボードを A: ドライブから Boot するように設定。
2. マザーボードに FDD と HDD を接続し、Boot 用のフロッピで起動する。
3. HTTP、FTP、NFS の内、サーバーで用意しているプロトコルを選択する。
4. ロボットの IP アドレスを設定する（各班毎に与える）。



図 6: DC-DC コンバータ

5. server とパッケージの場所を指定
6. インストーラの指示に従う。

4.1.6 無線 LAN のセットアップ

無線 LAN の設定を行う。マニュアルに従って、無線 LAN を Infrastructure モードにセットアップする。ネットワークケーブルを使うとき、クロスケーブルとストレートケーブルを間違えないこと。

4.1.7 電源系の組み上げ

ロボットの電源を構築する。ロボットには、12V のバッテリーを直列に接続し、24V を用いる。また、 $\pm 5V$ と $\pm 12V$ も必要なので、DC-DC コンバータ (図 6) を使って、24V から各電圧が出力されるよう配線する。また、非常停止ボタンでモータへの電力 24V 供給がカットできるようにする。

4.2 ハードウェア構成

最終的なハードウェアの構成を図 7に示す。

4.3 基本ソフトウェアの構築

4.3.1 モータコントローラ用のプログラムの作成

USB を介してモータコントローラボードを制御する。制御プログラムは、サンプルプログラムを参考に開発する。最初は、入力に対してモータの回転速度が変化するプログラムを作る。また制御パラメータは、ロボットがある程度完成してから調整する。

4.3.2 エンコーダの値を読み込むプログラム作成

USB を介してモータの回転数を読み取る。制御プログラムは、サンプルプログラムを参考に開発する。タイヤの回転数は、モータの回転数とタイヤの回転数が違うので計算して求める。

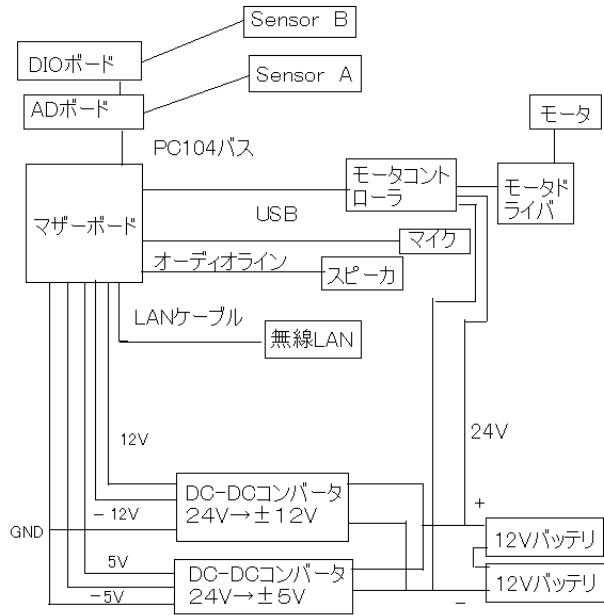


図 7: ハードウェア構成

4.3.3 DIO ボードを用いてセンサの値を読み込むプログラムの作成

DIO ボードは、01(具体的には 0V と 5V) のビット列で出力されたデータをコンピュータに読み込んだり、逆に、01 のデータを出力することができる。DIO が読み込んだデータをマザーボードに取り込んだり、あるデータを DIO から出力させるためには、ボードに対して読み込みコマンドや書き込みコマンドを送り、データの読み書きを行う必要がある。

DIO ボードのマニュアルには、ボードへのコマンドおよび、コマンドを書き込むアドレス、データを読み書きするアドレスおよび、データの読み書きの手順が書かれている。マニュアルを参考に C 言語でプログラムを作成する。

また、バス上のアドレスに対して直接読み書きするのでプログラムの実行は root 権限で行う必要がある。さらに、ボードの初期化の際に以下のコードを追加し、バス上のアドレスへ読み書きできるようにする。

```
{
int i;
i=iopl(3);
if(i) {printf("iopl(3) failed\n"); exit(1);}
}
```

4.3.4 AD ボードを用いてセンサの値を読み込むプログラムの作成

アナログ電圧出力のセンサを用いる場合は、AD ボードを用いてセンサからの出力電圧を読み取る必要がある。AD ボードは、入力されたアナログ電圧をデジタルデータに変換し、コンピュータで扱えるようにする。

AD ボードも DIO ボードと同様に、マザーボードから AD ボードにコマンドを送り、変換されたデータを読み取る。ただし、AD ボードは DIO と違い値を読み取るのみなのでデータの書き込みは存在しない。

AD ボードのマニュアルを参考にして、データを読み取るプログラムを作成する。この実行も root 権限で行う必要がある。また、前節で示したコードは一度実行されていれば再度行う必要はない。

4.3.5 マザーボードの音声入出力のセットアップ

マザーボードは、サウンドブラークスター互換のハードウェアが実装されており、音声の入出力が行える。Linux から音声を使うためには、複数の方法があるが、本実験では、音声認識ソフトウェアの都合上、ALSA(Advanced Linux Sound Architecture) (alsa driver 0.9.6 alsound 0.9.6, alsound tools 0.9.6) をインストールし使用する。

ALSA のパッケージを ftp でローカルディスクを持ってきて、tar (tar xvzf filename) で展開したあと、Readme にしたがってインストールを行う。

インストール後、録音再生を行って、動作確認をすること。

4.4 知的インターフェースの設計

4.4.1 音声認識ソフトウェアのセットアップ

音声認識ソフトは、JULIUS(julus free 3.3p4) を用いる。JULIUS のインストールは、JULIUS のセットに Release.txt.ja というファイル名のマニュアルがあり、手順に従ってインストールする。

認識辞書は、認識して欲しい単語と認識の結果画面に表示されるキャラクタの対で構成される。例えば、”Ohayo おはよう”と記述すると、Julius がおはようを認識したとき画面に Ohayo と表示される。認識の結果表示されるキャラクタは後に対話プログラムで使用するので半角英数字を用いるのが望ましい。

4.4.2 音声合成ソフトウェアのセットアップ

音声合成は、Windows 上の音声合成ソフトで合成音声のファイルを作成し、そのファイルをプログラムから再生する形で実現する。音声合成ソフトが生成するファイルは、WAV 形式のフォーマットであるのに対し、ALSA 上で扱えるフォーマットは RAW 形式である。WAV 形式から RAW 形式に変換するプログラムは配布するので変換プログラムを用いてファイルフォーマットを変換すること。

4.4.3 音声対話プログラムを構築

人間に命令されたら、返事を返したり、命令に従ってロボットを動かしたりするプログラムがこの部分である。Julius の認識結果が Ohayou だったら、合成音声のファイルを再生し「おはよう」とロボットを喋らせたりする。また、MAE NI SUSUME だったら、ロボットのタイヤを回し前進させる。

4.4.4 ロボットの基本行動の構築

ロボットが命令に従って前進する際に、ただ単に前に進むだけではロボットが机や人間にぶつかってしまう。通常このような事態を避けるためにセンサ情報を確認しながらロボットを行動させる。例えば、前進する場合には、前面の超音波距離センサの値が安全であるか確認しながら前進する。もし障害物が近づいたら停止する。もしくは避けて移動する。つまり、センサを確認しながら行動する基本行動セットを構築しておく必要がある。人間からの音声命令に従って基本行動セットが実行されるのが望ましい実装形態である。

4.4.5 命令セットの作成

以上説明したとおり、知的インターフェースの設計の中心的課題となるのが、命令と、ロボットの行動と、センサ情報をどのように組み合わせるかである。つまり、次の対応関係の設計が重要となる。

- 音声命令と行動の対応
- センサ情報と行動の対応
- 音声命令やセンサ情報と音声発話の対応

以上の対応関係は班毎に自由に設計してかまわない。また、音声とセンサ情報を絡めてロボットが人間に話しかけたり行動したりしてもかまわない。音声命令の実行中にセンサ情報の変化があった場合にロボットが何か他のことを実行しても構わない。ただし、次に挙げる必修命令は必ず実装すること。各命令に対するロボットの反応は各自考えること。

必修命令

1. 「こんにちは」
2. 「50cm 前進」(50cm が音声認識できない場合は、「前へ進め」でも構わない。ただしできるだけ正確に 50cm 進める。)
3. 「とまれ」
4. 「右に進め」, 「左に進め」
5. 「回れ」

6. 「もうすこし」
7. 「回れ右」（右に 90 度回転）
8. 「回れ左」（左に 90 度回転）

5 開発環境

マザーボード上の開発環境は、以下の通りである。

- OS: Linux Red Hat 9.0
- 言語: c 言語
- コンパラ: gcc

6 使用パーツおよび予算

品名	個数
マザーボード MPC6890B	各班 1 台
CPU Pentium III 850Hz	各班 1 個
AD ボード T104-AD16S	各班 1 枚
DIO ボード MPC104-96DIO	各班 1 枚
有線マイク	各班 1 個
スピーカ	各班 1 個
キーボード	各班 1 台
マウス	各班 1 台
ハードディスク	各班 1 台
モータードライバ	各班 2 個
モーター-コントローラ	各班 1 個
超音波距離センサ	各班 4 個
タイヤ	各班 2 個
筐体ブザイ (アルミフレーム)	各班基本セット
エンコーダ付きモーター	各班 2 個
キャスター	各班 1 個
鉛バッテリ	各班 2 個
KENWOOD 電源装置	各班 1 個
KENWOOD DC-DC コンバータ 24V → ±5V	各班 1 個
KENWOOD DC-DC コンバータ 24V → ±12V	各班 1 個
非常停止スイッチ	各班 1 個
無線 LAN NetHawk	各班 1 台

足りないもの各自購入、予算 3 万円／班

7 日程

1. 9/29 ガイダンス、班分け、基本説明、概念設計、役割分担
2. 10/6 設計発表、部品手配、ボディ作成、ボードセットアップ、回路製作、インストール作業
3. 10/20 モータテスト、回路製作、電源作成
4. 10/27 基本モジュール作成 ALSA インストール
5. 11/10 基本モジュール作成 音声認識インストール
6. 11/17 基本モジュール作成 音声認識インストール
7. 11/26(Wed) 各基本プログラムのテスト
8. 12/1 知的インターフェースの構築、センサの読み込み、行動の設計、命令の設計
9. 12/8 知的インターフェースの構築、センサの読み込み、行動の設計、命令の設計
10. 12/15 知的インターフェースの構築、センサの読み込み、行動の設計、命令の設計
11. 12/22 知的インターフェースの構築、センサの読み込み、行動の設計、命令の設計
12. 1/8 知的インターフェースの構築、試走
13. 1/19 デモンストレーション、最終報告会

8 最終報告

- 製作した知的インターフェースの命令の説明書を作り提出する。どうゆう命令をするとどういう動きをするのか。
- また、製作したロボットの特徴（売り込む点）を各班で報告する。
- 必修命令と、オリジナル命令のデモンストレーションを行なう。

打ち上げをやる。