ロボット言語による全方向移動台車の自律誘導
Autonomous Guidance of an Omni-Directional Vehicle using the Robot Language

杉井 春夫* 浅岡 武之* 谷口 正志*
Haruo Sugii *Takeyuki Asaoka* *Masashi Taniguchi*

大川 裕蔵* 朴 忠植*
Yuzou Okawa *Chunshuku Paku*

(1997年2月17日 受理)

The current AGV (Automated Guidance Vehicle) in a factory moves by sensing the guidance line which is laid down on the floor. This may be good for safety and reliablity, but the guidance line needs to be laid down along with the route, so that it is difficult to change layout in the factory. We proposed the method of changing layout flexibly by using a developed omni-directional vehicle which can move forward, backward, sideward and turn with two driving motors and a steering motor.

This paper describes the developed programming language for the omni-directional vehicle to move autonomusly, and a operating exsample of the vehicle using the language.

キーワード：全方向移動台車, ロボット言語, 自律誘導, 移動ロボット, AGV

1．はじめに

工場などにおいて規定のルートを走行する無人搬送車（ＡＧＶ）の誘導には，誘導ラインを床に埋設する方法、磁気テープや光学テープを床面に貼付する方法などが利用されている．これらの方法はＡＧＶの作業空間が明確で，安全性や信頼性の面から広く用いられているが，誘導ラインの設置工事を伴うため，設備のレイアウト変更を困難にしている．当研究所では，２個の駆動輪と操舵機構による前後進，横行，斜行，その場での方向転換などの移動機能を持つ全方向移動台車[1)]を開発し，これらの機能を生かしたレイアウト変更に柔軟に対応できる運用方法[2) 3)]を提案してきた．

この提案を具現化するため，全方向移動台車の自律運転に必要な動作記述レベルのロボット言語を開発し，移動台車を規定ルート（誘導ライン）から一時的に離脱させ，台車が自ら作業目標点を探索，接近し，作業終了後すみやかに規定ルートに復帰する自律誘導を試みたので報告する．

2．全方向移動台車

（1） 全方向移動台車の概要

全方向移動台車は，左右の２輪を独立して駆動するための２台のＡＣサーボモータと，操舵のためのＡＣサーボモータを備えている．左右の駆動モータによるＰＷＳ運動および，操舵モータによる斜行運動の組み合わせにより，全方向への種々の移動パターンが選択可能である．図１に開発した全方向移動台車の概要を示す．

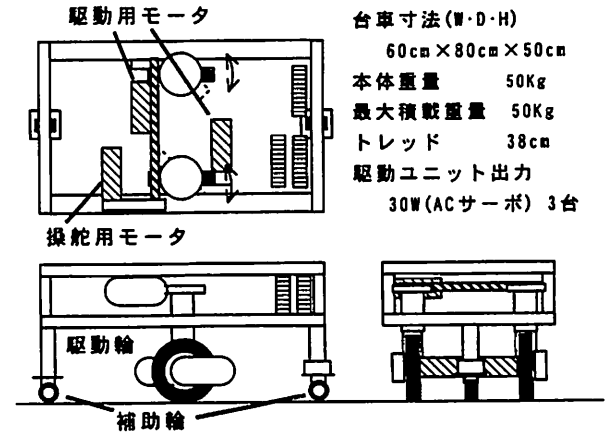


図１ 全方向移動台車の概要
Outline of an omni-directional vehicle

（2） 計測制御系の構成と制御ソフトウェア

本移動台車は，
（1）パソコンによるリモート運転モード
（2）操作ボックスによるマニュアル運転モード
（3）ロボット言語による自動運転モード
の３つのモードで制御することができる．自動運転のためのセンサとして，走行速度や移動距離計測のための内

---

* システム技術部ロボティクスグループ

界センサおよび障害物検知や対象物との距離計測，誘導ラインを検出するための外界センサを備えている．計測用および駆動系の制御用にそれぞれワンボードマイコンを使用した．図２は移動台車の計測制御系の構成である．

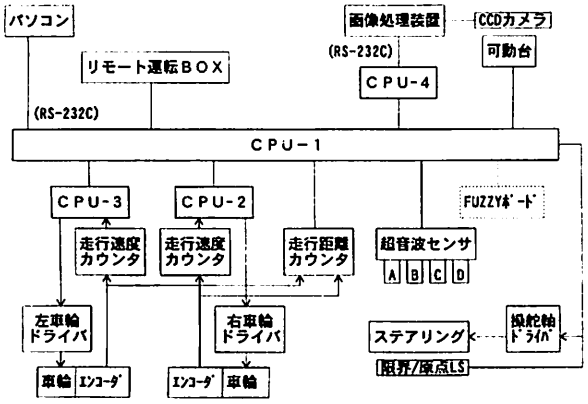


図２ 移動台車の計測制御系の構成

Block diagram of measurement & control system

左右駆動輪はＰＩＤ速度制御され，メインのＣＰＵ-１から与えられる目標速度に追従する．目標速度は符号付き12ビットデータで表現されており，誘導ラインによる進路補正時には，必要に応じて増減させる．進路補正のための目標速度増減量（左右輪の速度制御量）は，画像処理装置から読み取った台車進行方向と誘導ラインとの角度偏差量(θ)および画像のピクセル値から換算した左右方向の変位偏差量(dx)から次式により算出した．

$$v_x = \frac{(v_L + v_R)}{2} \cdot K_1 \cdot dx$$

$$v_\theta = \frac{(v_L + v_R)}{2} \cdot K_2 \cdot \tan(90 - \theta)$$

$$v_d = v_x + v_\theta$$

$$V_L = v_L + v_d,\ V_R = v_R - v_d$$

ただし，

v_x:横ズレに対する補正速度

v_θ:角度ズレに対する補正速度

v_d:速度制御量

K_1:台車中心と視野中心との距離およびトレッドより決まる定数

K_2:トレッドおよび補正半径より決まる定数

v_L, v_R:現在の左右車輪速度

V_L, V_R:新しい左右車輪速度

なお，誘導ラインのトレース時の進路補正は，タイマ割込みにより一定時間間隔で自動的に実行される．

自律移動時の作業台までの距離の検出は，図３に示すように，作業台に取り付けたマーカ（２個のＬＥＤを上下に６cm離して配置．上側ＬＥＤはカメラ水平面軸と一致）の画像上の重心位置がＡＧＶの接近に伴い変化することから求めた．ただし，作業台との距離が，移動台車に搭載している超音波距離センサの有効範囲内（約１m）に入ると，感度の良い超音波センサを利用して距離を計測する．

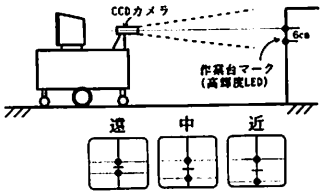


図３ 移動台車と作業台の距離検出方法

Distance detection method

３．ロボット言語

（１） ロボット言語の必要性

組立作業に用いる産業用ロボットの動作プログラムを記述するものとして，ロボット言語ＳＬＩＭ[1]およびその内部コードＳＴＲＯＬＩＣ[5]がＪＩＳにより標準化されている．これは各社が独自に使用してきた言語に共通性を持たせることが第一の目的であるが，部品設計のＣＡＤ化により対象物の形状データを得やすくなったことや，複雑な部品形状を扱うための３次元軌跡の生成は人手による教示に限界のあることにも起因している．

自律的な行動を目的とした移動ロボットでは，刻々変化するロボットの状態をセンサにより監視し，その時の状態に応じた動作を選択する必要があるため，その行動はシーケンス的な「動作コマンド列」では表現できない．また移動ロボットの教示をマニピュレータと同様な手段で行うことも現実的でない．従って，移動ロボットの誘導には，定まったルートの走行を記述する移動速度，移動距離，停止位置，分岐方向などの他に，センサ情報を監視しながら制御量を調整したり，動作を変更するための判断機能を持たせる仕組み（ロボット言語）が必要となる．

自律移動ロボットの行動表現として，ロボットの動作とセンサ情報の集合の組合せにより定義する方法が考案され，この表現方法によりロボットの行動をプログラムするためのロボット言語ＲＯＢＯＬ/0[6]が提案されている．

筆者らは，開発した全方向移動台車の自律誘導のため，シーケンス的な動作コマンド列の中に環境センサ情報に基づく行動の切り替え機能を組み込んだ処理系（ロボット言語ＲＬ/1）を開発し，移動台車に実装した．

（２） ロボット言語ＲＬ/1の仕様と組込関数

開発したロボット言語ＲＬ/1は，移動台車に搭載した

コンピュータに簡単な命令を与えれば，その命令を解釈して制御情報を生成する．表１にＲＬ/1の仕様を示す．

表１ ロボット言語（ＲＬ／１）の仕様
Specification of the robot language(RL/1)

数　値	２バイト整数（負数は補数表現）
定　数	10進数(ex.「1200」)または16進数(ex.「$1234」)
変　数	「A」-「Z」、「a」-「z」の５２個、ただし「s」～「z」は特殊予約変数
演算子	+(加算)，-(減算)，++(ｲﾝｸﾘﾒﾝﾄ)，--(ﾃﾞｸﾘﾒﾝﾄ) >(右ｼﾌﾄ)，<(左ｼﾌﾄ)，&(AND)，|(OR)，^(XOR)
式	変数＝変数(or定数)［演算子　変数(or定数)］ ex. A=B+C-d, H=C+$12+F&$AA55, C++, L=k<3 など
ラベル	ジャンプ先ラベルは「:00」～「:99」 ただし、「:96」～「:99」は異常時ｼﾞｬﾝﾌﾟ先
分岐命令	無条件ジャンプ #10　(goto :10) 条件付ジャンプ ?d=#20 (if d=0 goto :20) ?A>#30 (if A>0 goto :30)
その他	(END)プログラム終了。'(コメント)など

表２ 動作制御関数
Control function of RL/1

コマンド	意　味	動　作	引　数
(FRE)	Free	駆動軸を解放	
(HLD)	Hold	駆動軸を固定	
(ORG)	Origin	ステアリングを原点に復帰	
(MVI xx, yy, zz)	Move Imidiate	指定速度で、指定距離移動	xx 左車輪速度(±12) yy 右車輪速度(±12) zz 移動距離(±1900cm)
(MOV xx, yy)	Move	指定速度で走行 (※非同期コマンド)	xx 左ﾊﾟﾙｽ速度(±12) yy 右ﾊﾟﾙｽ速度(±12)
(STR xx)	Steer Right	ステアリングを右に回転	xx 回転角度(0-180°)
(STL xx)	Steer Left	ステアリングを左に回転	xx 回転角度(0-180°)
(STX xxx)	Steer X	ステアリングを回転	xxx ×0.1°回転(±1800)
(ROT xx)	Rotate	ＰＷＳによる旋回	xx 旋回角度(°)+左,-右
(TRC x)	Trace	白線追尾モードの設定	x 追尾ﾓｰﾄﾞ(0, 1, 2, 3)
(TIM xx)	Time	0.1秒単位で動作を停止	xx × 0.1秒(1-100)

表３ センシング関数
Sensing function of RL/1

コマンド	意　味	動　作	引　数
(KYC)	Kyori Clear	距離ｶｳﾝﾀをｸﾘｱ	
(KYO)	Kyori	距離ｶｳﾝﾀの読込み 台車中心の移動量を戻り値として、 右車輪の移動量を変数「u」に、 左車輪の移動量を変数「v」に格納	
(SCN x)	Scene	視覚ｾﾝｻのシーン切り替え	x シーン番号(0-7)
(IMG)	Image	視覚ｾﾝｻによる計測 面積(pixel)は特殊変数「w」に、(戻り値=面積) X座標(pixel)は特殊変数「x」に、 Y座標(pixel)は特殊変数「y」に、 角度(°)　は特殊変数「z」に格納	
(CAM Z)	Camera Zero	カメラを原点へ	
(CAM U) (CAM D)	Camera Up Camera Down	カメラを上に向ける カメラを下に向ける	
(CAM A xxx)	Camera swing Angle	指定された角度まで回転	xxx 角度(0-180°)
(CAM P xxx)	Camera swing Pulse	指定されたパルス位置まで カメラを回転	xxx ﾊﾟﾙｽ数(0-784)
(CAM R)	Camera Read	カメラの現在位置の読込み	
(ULT x)	Ultra	超音波ｾﾝｻの距離データ	x センサ選択(A, B, C, D)
(LMP x)	Lamp	照明の 消灯/点灯	x 0:消灯, 1:点灯
(INP xx)	Input	入力ﾎﾟｰﾄからの入力	xx 入力ﾎﾟｰﾄ1/07ﾄﾞﾚｽ
(OUT xx, yy)	Output	出力ﾎﾟｰﾄへのﾃﾞｰﾀ出力	xx 出力ﾎﾟｰﾄ1/07ﾄﾞﾚｽ yy 出力ﾃﾞｰﾀ
(FZO xx, yy)	Fuzzy Out	ファジーボードへの出力	xx データ(1) yy データ(2)
(FZI)	Fuzzy In	ファジーボードからの入力 特殊変数「s」,「t」に格納	

表４ 各トレースモードの処理内容
Process in a tracing-mode

モード	(MOV)コマンドの動作	(MVI)コマンドの動作
0	トレースしない	
1	コマンドを続行(ラインが見つかるまで直進、トレース再開)	
2	コマンド中断(停止)後 :98へジャンプ	:99へジャンプ
3	コマンド続行のまま　:96へジャンプ	:97へジャンプ

この言語はＢＡＳＩＣ言語に似た，ロボットの動作を記述するためのコマンドを持つインタプリタであり，処理速度を速めるため，中間言語に変換した後，実行するようにした．現段階ではサブルーチン・コールを持たず，ラベルの数や変数名に制限はあるが，ロボット言語ＲＬ/1は，環境センサの情報を調べながら行動を決定する動作を記述できるため，自律的な動作の指令を与えることに適している．

ＲＬ/1の組込関数として，制御系に制御量を与える動作制御関数（表２）と計測系から得られる情報を読みとるためのセンシング関数（表３）を定義した．一般的に制御コマンドはロボットの構造や機能に大きく依存し，ＲＬ／１の関数も本移動台車に特有なものとなっている．

動作制御関数の「MVI」命令は台車を指定速度で指定距離だけ移動させる命令で，移動が完了するのを待つ同期コマンド（実行が終了するまで次の命令に移らない）である．一方，「MOV」命令は非同期コマンドで，指定速度での移動を開始すれば直ちに次の命令の実行に移るため，移動中に何らかのセンシングを行い，動作の変更あるいは停止させるようにプログラムを作成する必要がある．誘導ラインを離れた自律移動時には，並列的に処理される進路補正は不要なため，表４に示すトレースモードで補正処理の有無を選択する．またこれは誘導ラインのトレースエラーに対する例外処理としても適用される．

（３）ＲＬ/1を用いた全方向移動台車の誘導例

全方向移動台車の機能を利用し，レイアウト変更に柔軟に対応できるように，開発したロボット言語ＲＬ／１を用いて，誘導ラインから一時的に離脱した後，ライン

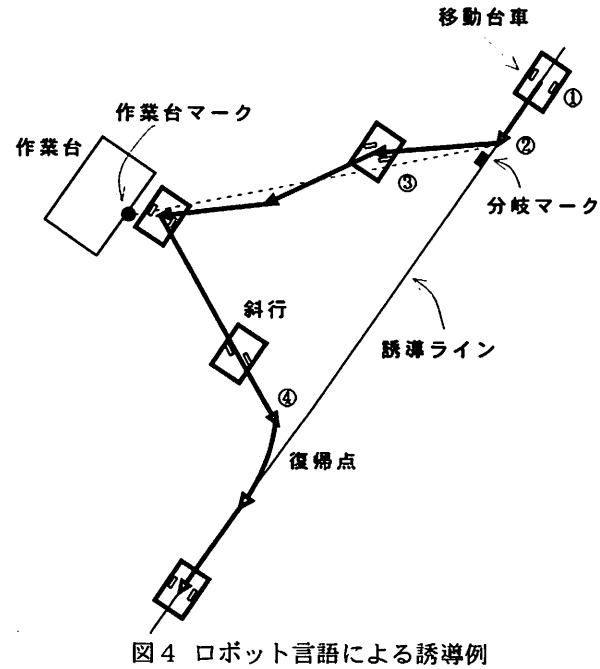


図４ ロボット言語による誘導例
Example of guidance method using the robot language(RL/1)

```
'SAMPLE.ROB
(CAM Z); (CAM A 90); (CAM D)
(SCN 0); (TRC 3); (MOV 3,3)
:10 A=(IMG)-180; ?A<#10
(MOV 0,0); (TRC 0)
(SCN 4); (IMG); L=x
(SCN 3); (IMG); U=x
A=U-L; ?A>#12
    c=205; d=-450; #13
:12 c=579; d=450;
:13
(CAM U); (SCN 2); (CAM P c)
@=5; C=c; D=d
:14 (IMG)
  a=x-128
  b=a<4+a; C=a<3+b>6+C
  c=a<7; d=a<3+c+a>7; D=D+d
  (CAM P C)
@--; ?@>#14

(STX D); (MOV 1,1)
:20 (IMG)
  Y=y-160; ?Y>#29
  X=x-60; ?X<#21
  X=x-200; ?X>#21
#20
:21 (MOV 0,0); @=5; D=0
:22 (IMG);
    a=x-128;
    b=a<4+a; C=a<3+b>6+C;
    c=a<7; d=a<3+c+a>7
    (CAM P C); D=D+d; @--
?@>#22
(STX D); (MOV 1,1)
#20
'
:29 (MOV 0,0); (HLD)
      .....
(CAM D); (SCN 0); (STL 75)
(TRC 0); (MOV 1,1)
:40 A=(IMG)-100; ?A<#40
(MOV 0,0); (STR 75); (TRC 3);
(MVI 1,1,50); (MOV 3,3)
      .....
:96 (MOV 0,0)
:99
(FRE); (END)
```

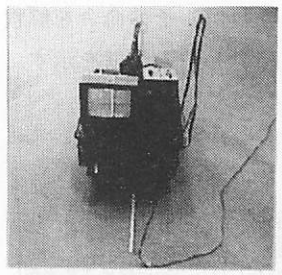
リスト１　プログラム例

Example of guidance program by RL/1

に復帰する自律走行を試みた．誘導にはＣＣＤカメラを用い，走行路を示す誘導ラインと，作業台の設置方向を示す分岐マークおよび作業台に取り付けた作業台マークの画像認識を行った．図４の例で，移動台車は誘導ラインに沿って走行中に分岐マークを発見して停止し，作業台を探す．作業台の検出は，移動台車に取り付けたカメラを左右に振ることで作業台マークの位置を確認する．移動台車は作業台マークの画面上での重心位置が設定値に達するか，あるいはあらかじめ設定した移動限界距離に達するまで作業台に接近する．作業台での作業終了後，誘導ラインの方向に斜行し，ラインを見つけて誘導ラインに復帰する．リスト１は，この一連の行動を記述したプログラム例である．また図５a～８aは，それぞれ誘導ライン走行，離脱箇所，作業台への接近，誘導ラインへの復帰の状況を示し，図５b～８bはその時点での画像処理画面である．

（４）　ロボット言語シミュレーション

移動台車のコンピュータに組み込まれた制御ソフトウエアは，ハードウエアと１対１に対応させて機能毎にパーツ化した下位レベルのものであり，使用法を間違わなければ与えられたコマンド通り確実に動作する．しかし，移動台車の動作を任意に変更できるロボット言語を介在させることで，プログラムエラーによる暴走の危険性が発生する．文法的な記述間違いや論理的な間違いは，事前のチェックである程度防止できるが，プログラム作成者の思い違いによるエラーには対処できない．



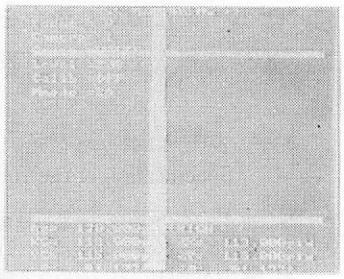
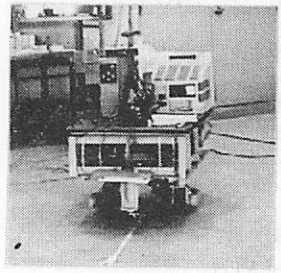


図５ａ　誘導ライン走行　　図５ｂ

Running along with a guidance line



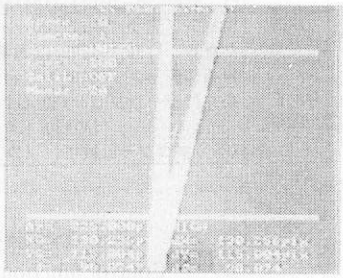
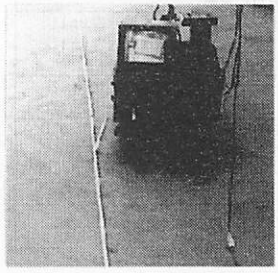


図６ａ　離脱箇所　　図６ｂ

At a derailing point



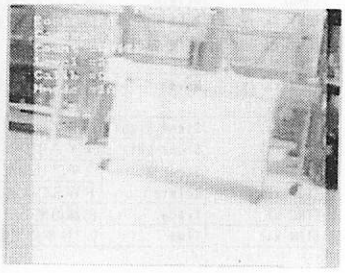
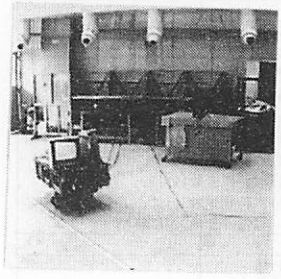


図７ａ　作業台への接近　　図７ｂ

Approach to a workstand



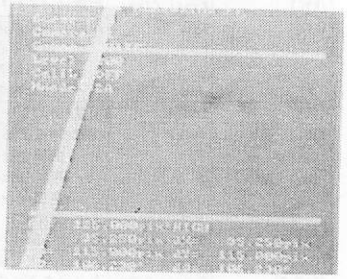


図８ａ　誘導ラインへの復帰　　図８ｂ

Return to a guidance line

そこで，誘導用プログラムの作成を支援するため，ロボット言語ＲＬ/1が解読でき，あらかじめ画面上で動作の確認ができるシミュレーション・プログラムを開発した．図９は，移動台車が作業台に接近する様子のシュミレーション結果である．このシミュレータは，作成した誘導プログラムの動作確認用だけでなく，ＡＧＶの効率的な走行経路を求めるためにも利用できる．

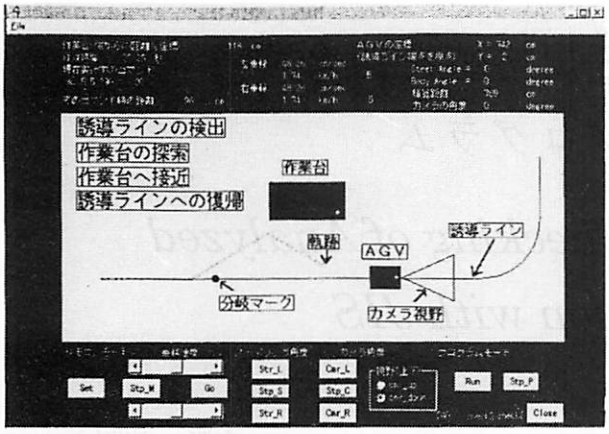


図９ 移動台車の誘導シミュレーション
Guidance simulation of an omni−directional vehicle

４．おわりに

当研究所で開発した全方向移動台車の自律的な誘導を行うため，環境センサの情報を逐次参照しながら行動を決定する動作を記述できるロボット言語ＲＬ/1を開発し，移動台車の自律的制御を行った．また，ロボット言語の採用により，プログラムエラーによる暴走の危険性を回避するために必要なシミュレータも開発した．

開発したＲＬ/1は動作記述レベルの言語であるため，１つの作業を複数の動作に分解して記述しなければならず，また一部の命令は，開発した移動台車特有の構造に依存しているため，汎用性に欠ける．

今後，自律移動ロボットの誘導には，作業指令から直接行動パターンが生成できる，作業記述レベルの言語が望まれよう．

誘導プログラム作成時におけるロボット言語の課題は，マンマシンインターフェイスをどのように設計するかであり，そのためには，ＧＵＩ手法を積極的に取り入れたプログラミング支援ツールが必要である．また，計測制御系の課題としては，ロボット言語による動作の制御と，行動中に発生する外乱の監視および修正処理を実時間でかつ並列的に処理する必要があり，そのためのリアルタイムＯＳの導入も必要である．

［参考文献］

1) 大川ほか，第７回アドバンティシンポジウム講演論文集，1994
2) 大川ほか，第８回アドバンティシンポジウム講演論文集，1995
3) 大川ほか，産業技術総合研究所報告，No. 4，1994
4) 日本工業規格，産業用ロボット：プログラム言語SLIM，JIS B 8439，日本規格協会，1992
5) 日本工業規格，産業用ロボット：中間コードSTROLIC，JIS B 8440，日本規格協会，1995
6) 鈴木ほか，”移動ロボットのセンサに基づく行動の記述とプログラミング”，日本ロボット学会誌，Vol. 10，No. 2，pp254～265，1992