

自律移動ロボットの沿目標移動について

中井 正幸 森 英雄

山梨大学 工学部 計算機科学科

TVカメラで取り込んだ画像を理解し屋外を自律移動するロボット晴信4号機は、“サインパターンと定形行動連鎖”なる概念に基づいて開発を行っている。視覚による定形行動の1つに、道路境界や白線などをサインパターンとし、それに沿って移動する沿目標移動がある。本報告では沿目標移動のサインパターン抽出と走行制御について述べる。走行実験の結果、最高40cm/secで40mの実験コースを走行できた。また、車軸に取り付けたオプティカルエンコーダによる位置決めは微小区間では正確であるが、長区間になると誤差が増大するのに対し、サインパターンによる位置決めは局所的に見れば誤差は大きい、大局的には正確であることが分かった。

Moving along of mobile robot

Masayuki NAKAI Hideo MORI

Department of Computer Science, YAMANASHI University

A mobile robot strategy "stereotyped motion by sign pattern" is applied to HARUNOBU-4. One of the most important stereotyped motion is Moving-along in which motion the robot moves along sign pattern such as road boundary, lane mark. The robot was successful in moving an asphalt road along a boundary with 40cm/sec in velocity. Because the road has a bank of small slope to the boundary, the robot trace in inner sensor space calculated from odometers attached to wheels showed a curve which bent to the boundary. However the actual robot trace showed a winding but not bending one.

1. はじめに

当研究室では1983年より、TVカメラにより道路環境を理解しながら移動する自律移動ロボット「晴信」の開発を行っている。現在、屋外移動ロボット「晴信4号機」を開発し走行実験を行っている[1][2][3]。

視覚を用いた自律移動ロボットの移動戦略の代表的なものには、予め正確な3次元モデルを与えて、これより予測した画像と実際に取り込んだ画像を比較して位置同定を行うもの[4]や、道路のモデルを与え、画像を3次元空間に変換し、ロボットの道路からの相対位置を求め移動するものがある[5][6]。「晴信」は、蜂などの下等動物の行動をまねた「サインパターンと定形行動連鎖」なる概念に基づく移動戦略をとっており、3次元モデルや複雑な知識を利用しないのでシンプルなシステム構成ができ、1サイクルの処理時間も数秒で行える。サインパターン (sign pattern) とは、移動のガイドや目標となる環境の特徴であり、定形行動 (stereo-typed motion) とは、ロボットがある状態に遭遇したときに、いつも見せる固定行動パターンである。

視覚による定形行動の基本形は、沿目標移動 (Moving Along) と向目標移動 (Moving Toward) である。沿目標移動は白線や段差をサインパターンとし、これに沿って移動する。また、向目標移動は柱や門、街路樹などをサインパターンとし、これに向かって移動する。このようなロボットの定形行動の繰り返しにより、屋外環境を移動する。

本報告では沿目標移動の視覚行動対応、すなわち、画像処理によるサインパターン抽出と、それによる走行制御、そして走行実験とその結果について述べる。

2. 沿目標移動

2.1 サインパターン

沿目標移動のサインパターンは以下の条件を備

えたものである。

- (1) エッジがはっきり現れる。
- (2) 他の領域と区別し易い。
- (3) 直線近似が容易にできる。
- (4) ある程度の長さを持っている。

このような条件を満たすサインパターンは次のようなものがある。

○道路境界 (段差)

道路と歩道、道路と側溝

○道路上の白線、黄色の線

○歩道上の点字ブロック

○横断歩道

○道路や歩道内の直線成分

サインパターンの抽出法は、種類や状況によって異なる。例えば道路境界や白線の様なサインパターンを抽出するためには、まず道路領域を抽出し、その境界を直線近似して得られた境界線をサインパターンとする。また、点字ブロックや横断歩道の様なサインパターンは、まずそれらの領域を抽出する。この場合、領域は複数になるので、マーキングを行い1つの領域にしてから、その境界線をサインパターンとする。

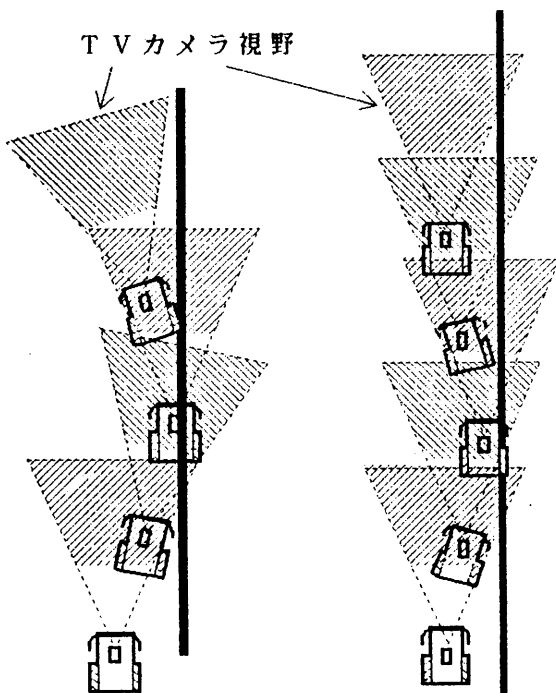
2.2 定形行動

沿目標移動の定形行動は、まずTVカメラをロボット正面に向け、約10度下に向ける。その後、画像処理を開始しサインパターンを探す。サインパターンを発見したら、それが画面の一定箇所に現れるように走行を制御する。それには、車体とサインパターンとの距離を一定に保つように走行制御すれば良い。この制御方式には予測制御を用いる。その理由は、画像処理には時間がかかりフィードバック制御が難しいからである。

2.3 TVカメラ回転台の利用

沿目標移動で用いるサインパターンは、ほぼ直線と見なせる。TVカメラを車体に固定すると、TVカメラの視野は常にロボットの正面を向き、図1(a)の様に車体の向きが変わる度に大きく変わりサインパターンの予測がつけにくくなる。また、

サインパターンに対して車体の向きが大きく傾くとサインパターンが視野から外れてしまう。そこで、図1(b)の様にTVカメラを常にサインパターンと平行になるように制御し、車体の向きが変わってもTVカメラは常にサインパターンを捉えるようにする。これによって、一度サインパターンを捉えたら、その後、見失う事はほとんど無く、円滑な沿目標移動が可能となる。



(a) TVカメラ固定 (b) 車体の向きに適応
図1 TVカメラの視野と車体の向き

3. 画像処理

画像処理には、当研究室で開発した画像解釈言語PILS V-5を用いる[7]。PILSは、エッジ抽出や領域分割、領域探索などの画像操作をツール化したもので、ライブラリはMC68000のアセンブラで記述しており、これらはPascalのサブルーチンとして呼び出すことによって実行する。PILSの特徴として次の点が挙げられる。

(1) カラー濃淡画像が扱える。

(2) 画像処理に特別なハードウェアを必要としない。

(3) エッジに基づく領域分割画像と、その辺縁を表す線画像の二つのデータ構造を有する。

(4) 多種の微分フィルターがあり、それを使い分けることによって様々な領域分割画像が得られる。

(5) 画像要素について、論理和、論理積、否定などの画像演算ができる。

(6) 画像要素の探索が容易にできる。

【画像処理システム】

○ホストコンピュータはMVME133システムでCPUはMC68020(16MHz)である。

○イメージメモリは512(H)×512(V)×3(RGB)で、1画素8Bitであり、ホストコンピュータに組み込まれている。

○TVカメラの映像はRGB分離器を介してイメージメモリに送られる。

3.1 画像領域分割

イメージメモリに取り込んだ画像を原画像とする(図2)。原画像に微分フィルターを作用させてエッジ抽出を行う(図3)。沿目標移動のサインパターンは斜めのエッジを持っているので、ここで用いる微分フィルターはエッジの斜め成分を抽出し易いものを用いている。このエッジによって原画像を領域分割する(図4)。この領域分割を行う際に、各領域の属性値として面積や縦の長さ、横の幅、重心座標、色(RGB)の平均値を計算している。この属性値を使ってサインパターン抽出に必要な領域を抽出する。

3.2 サインパターンの発見

領域分割した画像より、色情報と面積、位置を用いて、道路領域を抽出する(図5)。抽出した道路領域に対して路面の汚れやゴミなどによってできるノイズの除去を行う(図6)。この領域のはずれ値を除いた境界点を、最小二乗法で直線近似する。この境界線が抽出したサインパターンである(図7)。

最初サインパターンが画面に現れる位置は予測

できない。そこでサインパターンを発見するこの処理をFINDモードとする。サインパターンを抽出するのにロボットから10m以上はなれている道路を抽出する必要はないので、取り込んだ画像の下半分の300(H)×250(V)画素のウィンドウで処理をする。FINDモードで、画像取り込みからサインパターン抽出までの処理時間を表1に示す。

表1 処理時間 (300×250画素)

画像転送	0.30秒
エッジ抽出	1.65秒
領域分割	0.65秒
道路抽出	0.55秒
サインパターン抽出	0.35秒
合計	3.50秒

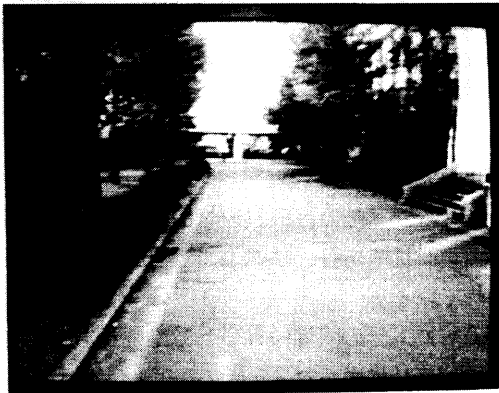


図2 原画像



図5 道路領域

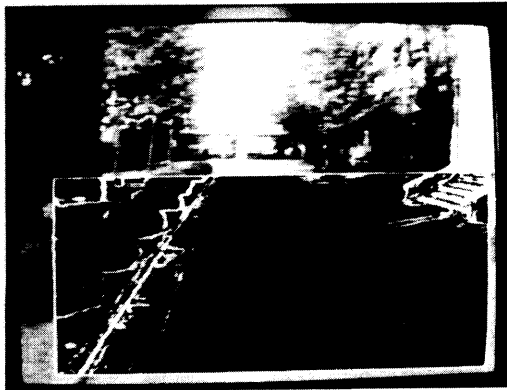


図3 微分画像

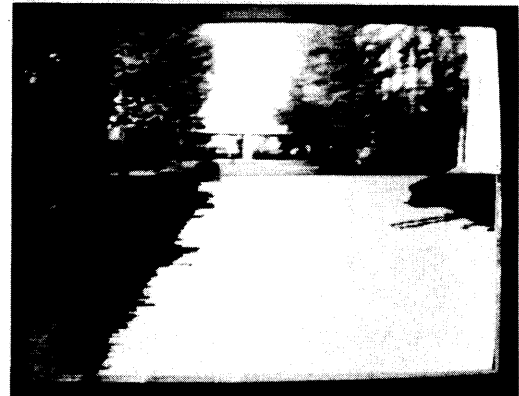


図6 ノイズ除去を行った道路領域



図4 領域分割画像

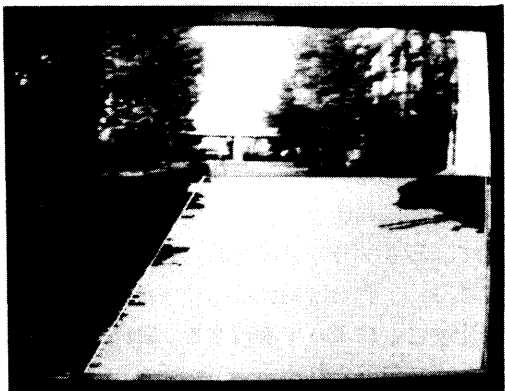


図7 抽出したサインパターン

3.3 サインパターンの追跡

サインパターンを発見できれば、それを予測サインパターンとして、次のサインパターンの画面上の位置が予測できる。また、FINDモードでは画像処理に4秒程かかり、走行速度を上げると1回の処理で数m進んでしまい危険である。そこで、100(H)×100(V)にウィンドウを縮小したものを2個使い、サインパターン付近のみを画像処理し、処理時間を短縮する(図8)。この処理をFOLLOWモードと呼び、その処理時間を表2に示す。そのウィンドウは予測サインパターンの位置と傾きによって画面上を移動する。サインパターンの追跡に失敗したら、FINDモードに戻る。

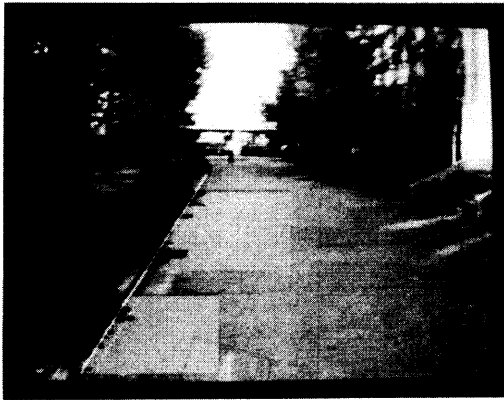


図8 サインパターンの追跡

表2 処理時間(100×100画素×2)

画像転送	0.10秒
エッジ抽出	0.50秒
領域分割	0.25秒
道路抽出	0.35秒
サインパターン抽出	0.15秒
合計	1.35秒

4. 内部センサ空間のサインパターン

画像処理で抽出したサインパターンとロボットの位置関係を示すために3つの座標空間を用意する。画像空間は、画面の中心を原点とし右方向にu軸、上方向にv軸をとる。ロボット空間は、車

軸の中心を原点とし正面方向にRx軸、右横方向にRy軸をとる。カメラ回転台空間は、z軸を中心にカメラの向き(pan)だけ回転したCx軸とCy軸をとる。また、車軸に取り付けたオプティカルエンコーダによりロボットの位置を算出する。この空間を内部センサ空間と呼び、ロボットの出発点を原点とし、右横方向にX、正面方向をYとる。

4.1 座標変換

画像空間の点(u, v)は、変換式(1)(2)を用いて、カメラ回転台空間の点(Cx, Cy)に変換できる。ここで、カメラの高さHと垂直方向の傾き(tilt)は既知で、比例定数αとβはキャリブレーションによって得る。

$$C_x = \frac{H \alpha v}{(\alpha \sin(\text{tilt}) - u \cos(\text{tilt})) \beta} \quad (1)$$

$$C_y = \frac{H (\alpha \cos(\text{tilt}) + u \sin(\text{tilt}))}{\alpha \sin(\text{tilt}) - u \cos(\text{tilt})} \quad (2)$$

カメラの向きを(pan)とすると、カメラ回転台空間の点(Cx, Cy)は、変換式(3)(4)を用いてロボット空間の点(Rx, Ry)に変換できる。

$$R_x = C_x \cos(\text{pan}) + C_y \sin(\text{pan}) \quad (3)$$

$$R_y = -C_x \sin(\text{pan}) + C_y \cos(\text{pan}) \quad (4)$$

内部センサ空間のロボットの現在位置(Px, Py)向きφとすると、変換式(5)(6)によってロボット空間の点(Rx, Ry)を、内部センサ空間上の点(X, Y)に変換する。

$$X = P_x + R_x \cos \phi + R_y \sin \phi \quad (5)$$

$$Y = P_y - R_x \sin \phi + R_y \cos \phi \quad (6)$$

以上の変換を行うと画像上の点が内部センサ空間に投影される。

4.2 サインパターンの補正

画像上のサインパターンを座標変換する場合、次の問題点がある。

(1)実際の位置がロボットから離れるほど、1画素のずれによる誤差が大きくなる。

(2)変換の誤差は5~10%で、距離の二乗に比例して大きくなる。

(3)誤ったサインパターンを抽出することがある。

(4) サインパターンを抽出できない時がある。

そこで、座標変換後サインパターンの補正を行う。

サインパターンの補正は、図9に示すように抽出したサインパターン SP_i と以前の画像処理で抽出したサインパターン $SP_{i-4} \sim SP_{i-1}$ を使う。内部センサ空間で画像を取り込んだ時のロボットの位置 P_i よりサインパターン SP_i に最短距離にある点 Q_i を求める。 $Q_{i-4} \sim Q_i$ を最小二乗法で直線近似する。もし Q_i がはずれ値ならば、その時のサインパターン抽出は失敗したと見なしデータより取り除く。これにより補正サインパターンを内部センサ空間に作成できる。

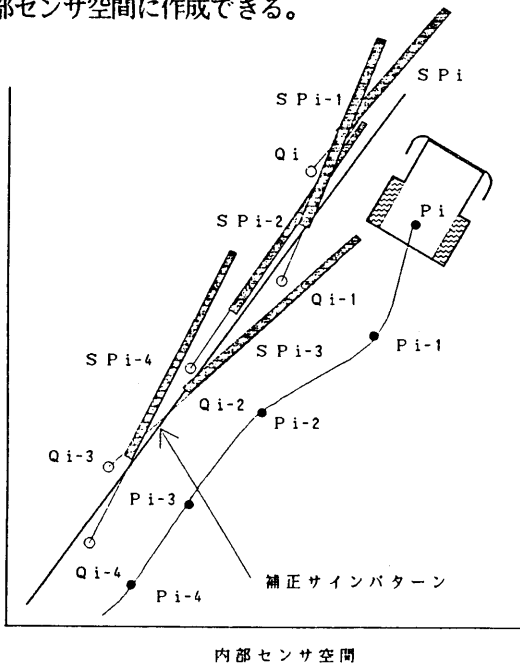


図9 サインパターンの補正

5. 走行制御

5.1 操舵指令作成

操舵指令は左右の車輪の速度 V_r と V_l であり、ホストコンピュータよりRS-232C回線を使って車体に送ることによって実行する。操舵指令を送る間隔を Δt 秒とすると、現在点 P_i より現在の操舵指令で走行した場合の Δt 秒後のロボットの到達点 $P_{i+1} (X_{i+1}, Y_{i+1})$ と向き ϕ_{i+1} を予測する。

図10に示すように補正サインパターンより W_{cm} 離れて目標軌道を設定し、この軌道に ΔW_{cm} の許容範囲をとる。点 P_{i+1} と補正サインパターンとの距離 D を求め、予測誤差 $\Delta D = D - W$ を計算し、走行速度 V から V_r と V_l を次のように決定する。

(1) $|\Delta D| < \Delta W$ (コース上にある) 場合

現在の速度を維持する。

(2) $\Delta D > \Delta W$ (コースを右に外している) 場合

$$V_r = V(1 + \lambda), \quad V_l = V(1 - \lambda)$$

λ は $\Delta D \leq \Delta W$ となるように決定する。また、上限を設定しそれ以上は大きくしない。

(3) $\Delta D < -\Delta W$ (コースを左に外している) 場合

$$V_r = V(1 - \lambda), \quad V_l = V(1 + \lambda)$$

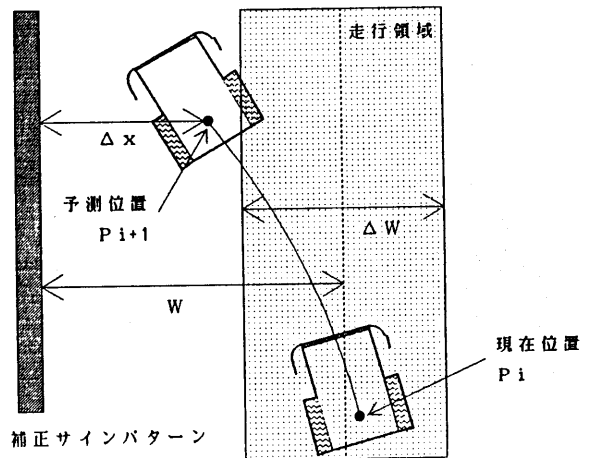


図10 予測制御

5.2 画像処理と車体システムの並列処理

(1) 車体システムよりロボットの位置、方位、状態を読み取る。このとき、センサによる定形行動中ならば終了まで待つ。

(2) 画像をTVカメラより取り込む。

(3) Δt 秒後のロボットの位置と方位を予測し、操舵指令と回転指令を作成する。

(4) 操舵指令を車体システムに送る。

(5) 回転指令をTVカメラ回転台に送る。

(6) サインパターン抽出、補正を行う。

(7) 車体システムは操舵指令を実行する。

(8) TVカメラ回転台は回転指令を実行する。

(1)~(6)は画像処理部であるホストコンピュー

タがシーケンシャルに実行し、(7)(8)は(6)とほぼ同時に実行を開始する。(6)の処理時間が約3~5秒、(7)で走行制御の効果が現れるまでに2~3秒、(8)で回転終了までに0~1秒かかる。

5.3 走行指令書と走行履歴

晴信は地図を持たない代わりに走行指令書を持つ。走行指令書とは目標到達点までの移動法(沿目標、向目標)やサインパターンの情報(種類、長さ)、走行パラメータなどが書かれている。また、走行履歴としてロボットの位置や抽出したサインパターンの距離などのデータを記録する。走行指令書の解析と走行履歴の記録は、端末として使用しているラップトップコンピュータで行う。

6. 走行実験

「晴信4号機」(図11)は、画像処理システムのある胴体部と車体システムのある車体部からなり、車体制御にはMC68000(8MHz)シングルボードコンピュータを用いている。

胴体部は56(W)×88(L)×85(H)cm、重さ約40kgで、直射日光を避けるため全体をアルミ・パネルで覆っており、放熱ファンが12箇所にある。車体部は75(W)×105(L)×41(H)cm、重さ約65kgで、三輪構造をなし、前輪はキャスターで自由に動き、後輪駆動である。

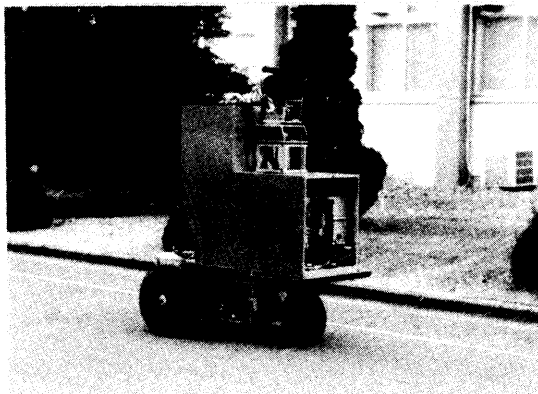


図11 晴信の外観図

6.1 走行コースと制御パラメータ

大学構内の図12に示す環境で沿目標移動の走行実験を行った。その時の条件を以下に示す。

○走行環境

路面：アスファルト 天候：晴れ

時間：13~15時

○サインパターン

種類：1)道路と花壇の境界

2)道路上に引かれた白線

距離：約40mでほぼ直線

○走行パラメータ

速度：V=10~40(cm/sec)

位置：W=60~100(cm)、 $\Delta W=20\sim40$ (cm)

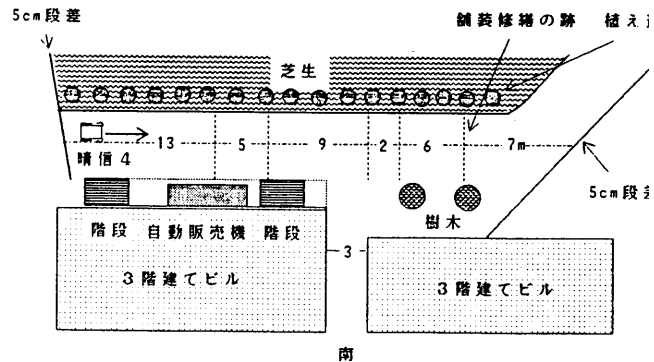


図12 走行実験コース

6.2 実験結果

最高40cm/s(1.4km/h)の速度(「晴信4号機」の現時点での最高速度)での走行に成功した。白線は無くとも走行できたが、白線が有る方がサインパターン抽出の失敗がなくスムーズに走行した。また、緩やかに曲がったサインパターンの沿目標移動は、画面上でほぼ直線に見なせるカーブならば走行が可能であった。図13に速度30(cm/sec)、走行領域 60 ± 20 (cm)、白線有りの内部センサ空間での走行軌跡を示す。この図で●はロボットの位置、そこから伸びている線はロボットの向き、○は抽出したサインパターンの位置を示している。また、直線のサインパターンを追跡したにも関わらず、図では右に円弧を描くように曲がっており、他の走行軌跡も一様に右に曲がっている。また、路面がフラットな状態では正しく走行することが

確認してあるので、機械系やサーボ系の故障ではない。内部センサ空間での走行軌跡がこのように右に曲がってしまう理由は次のように考えられる。

(1)排水のため道路の端は傾いているので、自重で左右のタイヤの径が変わってしまい、それにもよって左の路肩に寄ってしまう。それを修正するために左車輪を多く回転させるので内部センサ空間での走行軌跡が右に曲がる。

(2)路面のパッチやギャップによる路面の凹凸の影響で誤差が発生する。

(3)発進や停止などの急な速度変化を行うと、タイヤがスリップし方向がかわる。

そこで、サインパターンが直線であると仮定して補正を行った走行軌跡を図14に示す。この補正した空間を、サインパターン空間と呼ぶ。この図よりロボットはサインパターンに沿って走行していることがよく分かる。

7. おわりに

内部センサで求めたロボット自身の位置は微小区間ではかなり精度の良いものであるが、長い区間を走行すると計測誤差が累積されて絶対位置の定位には大きな誤差がある。一方、画像処理によ

るロボットの定位は、局所的には誤差が大きいですが、サインパターンからの相対的の位置は正確である。

[参考文献]

- [1] 中井、斉藤、森：視覚移動ロボット「晴信4号機」の開発NO.2, 情報処理学会第38回全国大会1989、151-152
- [2] 湯口、中山、森：視覚移動ロボット「晴信4号機」の開発NO.3、同上、153-154
- [3] 陳、森：視覚移動ロボット「晴信4号機」の開発NO.4、同上、155-156
- [4] 辻：室内環境自律移動ロボットの視覚、日本ロボット学会誌Vol.5, NO.6、1987、470-474
- [5] M.A.Turk, D.G.Morgenthaler, K.D.Greban & M.Marra: VITS - a vision system for autonomous land vehicle navigation, IEEE Trans. on PAMI, Vol.10, No.3, 1988 342-361
- [6] C.T.Martial, H.Hebert & T.Kanade: Vision and Navigation for the Carnegie-Mellon Navlav, ibid, 362-373
- [7] 岩淵、森：画像解釈言語PILS V-5の開発、情報処理学会第38回全国大会、1989、149-150

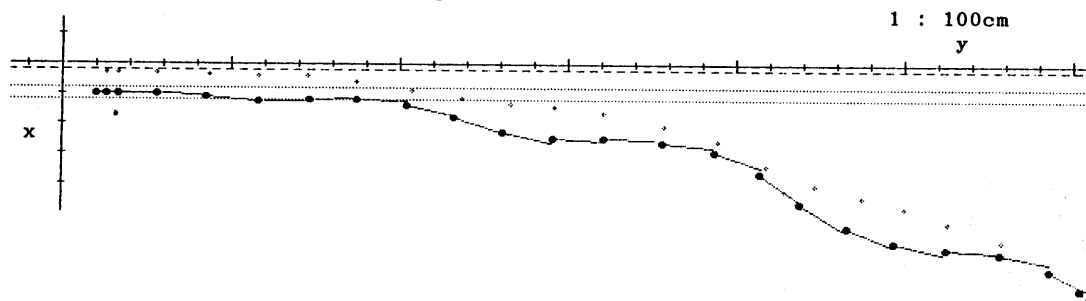


図13 内部センサ空間の走行軌跡

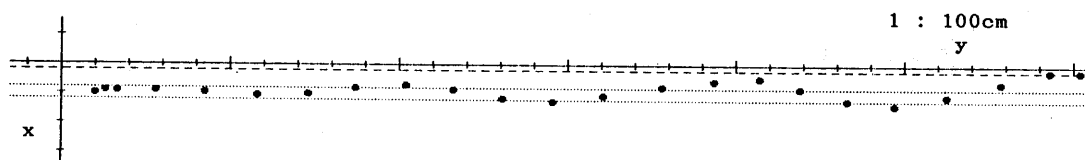


図14 サインパターン空間の走行軌跡