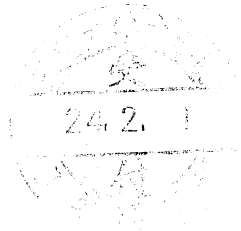


平成23年度修士学位論文

クラスタリング手法を用いた
二輪駆動型ロボットの運転操作モデルに関する研究



三重大学大学院工学研究科博士前期課程
電気電子工学専攻
電気システム工学講座

牧田 克紀

目次

1	序論	1
1.1	本研究の背景	1
1.1.1	人の操作のモデリング	1
1.1.2	ハイブリッド動的システム (HDS)	1
1.1.3	HDS による人の操作のモデリングとその応用	2
1.2	本研究の目的	3
1.3	本論文の構成	3
2	数学的準備	5
2.1	PWARX モデル	5
2.2	データクラスタリング	5
2.2.1	最小二乗推定によるパラメータベクトルの同定	7
2.3	サポートベクターマシン	8
3	PWARX モデルによるロボットの運転操作の表現	10
3.1	データ取得環境	10
3.2	入出力データと人の認識情報	12
3.3	使用データの選定	12
3.4	PWARX モデルのサブモデル数	13
3.5	人の認識情報	14
3.5.1	\mathbf{x}_I	14
3.5.2	\mathbf{x}_{II}	15
4	前任者による人の認識情報の重要性の評価	18
4.1	前任者の評価法	18
4.2	同定用データとモデリング	19

4.2.1	同定用データの質	19
4.2.2	Model I の同定	20
4.2.3	Model II の同定	23
4.3	自動走行結果と重要性の評価	27
4.4	前任者の評価法の問題点	29
5	寄与度を用いた人の認識情報の重要性の評価	32
5.1	寄与度を用いた評価法	32
5.2	寄与度を用いた評価の例	34
5.3	モデル次数低減化への応用	35
5.4	モデル次数低減化の検証	35
6	結論	38
6.1	まとめ	38
6.2	今後の課題	38
	参考文献	40
	関連論文および口頭発表	43
	謝辞	44

図 目 次

2.1	線形識別関数を用いたトレーニングデータの線形分離	9
3.1	データ取得環境	10
3.2	ロボットの外観	11
3.3	d_1, d_2	15
3.4	ε	15
4.1	前任者の評価手順	18
4.2	人の操作による走行軌道	20
4.3	入出力空間 $d_1 - d_2 - \omega$	21
4.4	クラスタリング後の入出力空間 $d_1 - d_2 - \omega$	22
4.5	Model I の各モードの領域	24
4.6	入出力空間 $\varepsilon - \omega$	25
4.7	クラスタリング後の入出力空間 $\varepsilon - \omega$	26
4.8	Model II の各モードの領域	28
4.9	自動走行環境	29
4.10	Model I による走行軌道	30
4.11	Model II による走行軌道	31
5.1	本研究の評価手順	33

表 目 次

3.1	人とモデルの走行軌道の差の平均	16
4.1	走行データの質 μ	19
4.2	走行データの質 σ	19
4.3	Model I のパラメータベクトル	23
4.4	Model I の識別関数パラメータ	23
4.5	Model II のパラメータベクトル	27
4.6	Model II の識別関数パラメータ	27

第1章 序論

1.1 本研究の背景

1.1.1 人の操作のモデリング

近年，人による操作を代替するロボット工学の研究や，その操作を解析し，操作者のアシストシステムなどに展開する研究が盛んに行われており，人の操作のモデリングに関する様々な研究が報告されている [1], [2], [4], [5], [6]．

たとえば，人の操作のモデリングとして，制御工学の知見に基づく単一の線形システム^[1]や，情報処理工学などのニューラルネットワーク等の非線形動特性を内包した動的システムモデルなどの数理モデル^[2]などが提案されてきた．しかしながら，前者においては複雑で多様なタスクを表現することは困難であると考えられ，後者においては入出力関係が非線形となり，人の操作の解析が困難であるという欠点を有している．

一方，人の操作がいくつかの要素的な操作の切替えにより実現しているという考え方^[3]に基づく新たなタスクのモデリング方法が提案されている [4], [5], [6]．そこでは，人の操作が連続的動作と離散的判断により実現されることに注目し，人の操作をハイブリッド動的システム（HDS : Hybrid Dynamical System）^{[7], [8]}として同定している．

1.1.2 ハイブリッド動的システム（HDS）

HDS は，微分方程式や差分方程式などで表現される連続ダイナミクスと，ロジックやルールなどの離散ダイナミクスが混在するシステムである．近年，HDS に関して，モデリング [9], [10]，安定性解析 [9], [11]，制御 [9], [12], [13]，検証 [14]，故障検出 [15] など，様々な観点から研究が進められている．

HDS のモデリングにおいては，線形相補性（LC: Linear Complementarity）システム^[16]，区分的アフィン（PWA : PieceWise Affine）システム^[17]，混合論理動的（MLD:

Mixed Logical Dynamical) システム^[12] など、さまざまなモデルが提案され、また、これらの表現能力が等価であることが示されている^[10]。特に、状態によって動作モードが切り替わる HDS に対しては、PWA システムの同定手法が有効であり、クラスタリングに基づく手法^[18]、MLD システムに基づく手法^[19]、境界誤差法^[20]、ベイズ法^[21]、代数的手法^[22] など、さまざまな手法が提案されている。とりわけ、クラスタリングに基づく手法^[18] は比較的大量のデータを扱うことが可能であり、アルゴリズムが簡単で、実用性が高いという特徴を有しており、非線形電気回路^[18]、工作機械^[23]、人の運転行動^[4] などへの適用例が報告されている。

1.1.3 HDS による人の操作のモデリングとその応用

HDS により得られるモデルは、一見すると複雑そうに見える人の操作を、いくつかの簡単かつ要素的な動作モードの集合とそれらの切り替えで表現できるため、入出力関係が明確であり、表現能力が高いという特徴を有している。したがって、人の行動の予測、アシストシステムや安全システムの構築など、工学的利用への発展が期待できる。

文献^[24]、^[25] では、MLD システムに基づく手法を用いた、人の操作のモデリングと操作の自動化への応用に関する研究が報告されている。しかしながら、MLD システムに基づく手法は、最適化問題の解としてモデルパラメータが得られるものの、同定に必要なデータ数やモデルの構造が複雑になり求めるべきパラメータ数が多い場合、計算量が多くなり、最適解の導出が困難という欠点がある。したがって、モデリングにおいて、データ数やモデル構造に制限が生じ、対象となる操作が限定される。また、少数のデータからモデリングしたとしても、それらのデータから、人の特徴を十分抽出できるかどうかは疑問である。

これに対し、クラスタリングに基づく同定手法は比較的大量のデータを扱うことができ、実用的な手法として注目されている。文献^[4] で、クラスタリングに基づく同定手法を用いて人の車の運転における前方車両回避行動をモデリングし、モデルを用いた人の運転行動の解析、理解に関する研究が報告されている。

1.2 本研究の目的

本研究では、車のレーンキープ時のステアリング操作を想定し、人が一定速度で直進している二輪駆動型ロボットをコースに沿って走行させたときのステアリング操作を考える。そして、人がコースに沿ってロボットを走行させるとき、どのような環境情報を基にステアリングの操作量を決定しているのかということに着目する。人の操作のモデリングにおいて、これらの情報がモデルに組み込まれていなければ、人の操作を表現した好ましいモデルを得ることができない。これまでの先行研究では、2種類の情報、すなわち「ロボットから左右の壁までの距離」と「コース中央とロボットの予測位置との間の距離」の利用が提案されており、前者の情報を用いた Model I と後者の情報を用いた Model II をモデル化し、各々のモデルを利用したロボットの自動走行を実現している。これら二つのモデルの自動走行結果の比較から Model II の方が人の操作が表現されている、つまり、後者の情報の方が人の認識している情報としてより重要なのではないかという可能性が示唆されている。しかしながら、これら認識情報の重要性を評価する従来手法はいくつかの問題点があった。本稿ではこれらの問題点を解決するため寄与度を用いた新しい認識情報の重要性の評価法を提案する。寄与度を定義することで、実際の人の操作量に対して認識情報がどれだけ支配的かを定量的に評価できるようになる。また、得られた評価結果から認識情報間の順位付けを行い、重要でない情報を除去して操作モデルの低次元化を検証する。

1.3 本論文の構成

2章では本研究の数学的準備として、HDSの1つのクラスであるPWARXモデルについて述べ、その同定手法であるクラスタリングに基づく同定手法について述べる。3章では一定速度で直進している二輪駆動型ロボットをコースに沿って走行させたときの人のステアリング操作を対象として、その操作をどのようにしてPWARXモデルとして表現するのかについて述べる。4章では前任者の人の認識情報の評価法について説明し、実際に本稿においても2種類の認識情報に対して前任者の手法を用いて評価する。そして4章の最後で前任者の評価法の問題点について述べる。5章では4章で述べた前任者の評価法の問題点を解決した新しい手法として寄与度を用いた人の認識情報の重要性の評価法について述べる。そして、認識情報間の重要性の評価結果から認識情報に対して順位付けを行い、重要性の低い情報を除去することで操作モデルの次数

低減化を行ったのでそれについても述べる。最後に、6章において本研究の成果をまとめ、今後の課題について述べる。

第2章 数学的準備

本章では、数学的準備として、2.1節では、本論文においてヒトのスキルモデルとして扱う PWA システムの1つのクラスである PWARX モデルについて述べる。また、2.2節と2.3節では、スキルモデルを同定する手法として用いるデータクラスタリングとサポートベクターマシンについて述べる。

以降、実数の集合を、 \mathbb{R} と記述する。また、任意の実行列 A およびベクトル v に対して、 A および v の転置をそれぞれ、 A' および v' と記述する。

2.1 PWARX モデル

区分的 ARX(PWARX: PieceWise affine AutoRegressive eXogeneous) モデルは、状態によって動作モードが切り替わるシステムを表現することができる PWA システムの一つのクラスであり、

$$y = \theta_i' \varphi \quad \text{if } x \in \mathcal{X}_i, \quad (i = 1, \dots, s), \quad (2.1)$$

のように表現する。ここで φ を回帰ベクトルと呼び、 $\varphi = \begin{bmatrix} x' & 1 \end{bmatrix}'$, $x \in \mathbb{R}^n$ と表される。また、 $\{\theta_i\}_{i=1}^s$, $\{\mathcal{X}_i\}_{i=1}^s$ および s はそれぞれ、パラメータベクトル、入力 x の区分、およびシステムのサブモデル数を表す。

2.2 データクラスタリング

データクラスタリングとは、入出力データ (x_t, y_t) ($t = 1, \dots, N$) をデータの類似度に従って、 s 個の部分集合 $\{\mathcal{F}_i\}_{i=1}^s$ に分類する手法である。そこで、各データの特徴量を求めるため、まず、各データに対して、

$$\|x_t - \tilde{x}\|^2 \leq \|x_t - \hat{x}\|^2, \quad \forall (\hat{x}, \hat{y}) \in \mathcal{S} \setminus \mathcal{C}_t, \quad (2.2)$$

を満たす $c-1$ 個のデータ $(\tilde{x}_t, \tilde{y}_t)$ を集め、局所集合 \mathcal{C}_t を作成する。ここで $\|\cdot\|$ はユークリッドノルムを表す。

次に、各データの特徴量を表した特徴ベクトル ξ_t ,

$$\xi_t = \begin{bmatrix} (\theta^{\text{LS},t})' & m_t' \end{bmatrix}', \quad (2.3)$$

$$\theta^{\text{LS},t} = (\Phi_t' \Phi_t)^{-1} \Phi_t' y_{\mathcal{C}_t}, \quad (2.4)$$

$$\Phi_t = \begin{bmatrix} x_1 & x_2 & \dots & x_c \\ 1 & 1 & \dots & 1 \end{bmatrix}', \quad (2.5)$$

$$m_t = \frac{1}{c} \sum_{(x_t, y_t) \in \mathcal{C}_t} x_t \quad (2.6)$$

を求める。ここで、 $\theta^{\text{LS},t}$, m_t および $y_{\mathcal{C}_t}$ はそれぞれ、パラメータベクトル、 \mathcal{C}_t に属するデータの入力の平均値および出力サンプルにより構成されるベクトルを表す。

最後に、G. F. Trecate らによって提案されたクラスタリング手法^[18]に従って、特徴ベクトル ξ_t を s 個の部分集合（以降クラスターと呼ぶ） \mathcal{D}_i に分類する。クラスター \mathcal{D}_i への分類では、評価関数、

$$J(\{\mathcal{D}_i\}_{i=1}^s, \{\mu_i\}_{i=1}^s) = \sum_{i=1}^s \sum_{\xi_t \in \mathcal{D}_i} \|\xi_t - \mu_i\|_{R_t^{-1}}^2, \quad (2.7)$$

$$R_t = \begin{bmatrix} V_t & 0 \\ 0 & Q_t \end{bmatrix}, \quad (2.8)$$

$$V_t = \frac{SSR_t}{c - (n+1)} (\Phi_t' \Phi_t)^{-1}, \quad (2.9)$$

$$SSR_t = y_{\mathcal{C}_t}' (I - \Phi_t (\Phi_t' \Phi_t)^{-1} \Phi_t') y_{\mathcal{C}_t}, \quad (2.10)$$

$$Q_t = \sum_{(x_t, y_t) \in \mathcal{C}_t} (x_t - m_t)(x_t - m_t)' \quad (2.11)$$

を用いて、最適なクラスター \mathcal{D}_i^* およびその中心 μ_i^* を求める。 R_t , V_t , SSR_t および Q_t はそれぞれ、特徴ベクトル ξ_t の分散、 $\theta^{\text{LS},t}$ の共分散行列、残差平方和および局所集合 \mathcal{C}_t 内の x_t のまばらさを表す。また、任意の実行列 A に対して、 $\|A\|_{R_t^{-1}}^2 = A' R_t^{-1} A$ とし、特徴ベクトル ξ_t の分散を考慮したノルムを表す。以下にクラスター \mathcal{D}_i^* およびその中心 μ_i^* の導出手順を示す。

Step1: 中心 $\mu_i^{(0)}$ ($i = 1, \dots, s$) を初期化し、 $r = 0$ と置き、評価関数、

$$J(\{\mathcal{D}_i\}_{i=1}^s, \{\mu_i^{(0)}\}_{i=1}^s). \quad (2.12)$$

を最小化する特徴ベクトル ξ_t のクラスター $\mathcal{D}_i^{(0)}$ を計算する.

Step2: $i = 1, \dots, s$ に対して線形システム

$$\left(\sum_{t: \xi_t \in \mathcal{D}_i^{(r)}} R_t^{-1} \right) \mu_i^{(r+1)} = \sum_{t: \xi_t \in \mathcal{D}_i^{(r)}} R_t^{-1} \xi_t. \quad (2.13)$$

を解くことにより中心を更新する.

ここで, $J(\{\mathcal{D}_i^{(r)}\}_{i=1}^s, \{\mu_i^{(r+1)}\}_{i=1}^s) = J(\{\mathcal{D}_i^{(r)}\}_{i=1}^s, \{\mu_i^{(r)}\}_{i=1}^s)$ が成り立つ場合は $\mathcal{D}_i^* = \mathcal{D}_i^{(r)}$ および $\mu_i^* = \mu_i^{(r)}$ と置き, 終了する. そうでなければ Step 3 に進む.

Step3: 評価関数,

$$J(\{\mathcal{D}_i\}_{i=1}^s, \{\mu_i^{(r+1)}\}_{i=1}^s). \quad (2.14)$$

を最小化する特徴ベクトル ξ_t のクラスター $\mathcal{D}_i^{(r+1)}$ を計算する.

ここで, $J(\{\mathcal{D}_i^{(r+1)}\}_{i=1}^s, \{\mu_i^{(r+1)}\}_{i=1}^s) = J(\{\mathcal{D}_i^{(r)}\}_{i=1}^s, \{\mu_i^{(r+1)}\}_{i=1}^s)$ が成り立つ場合は, $\mathcal{D}_i^* = \mathcal{D}_i^{(r)}$ および $\mu_i^* = \mu_i^{(r+1)}$ と置き, 終了する. そうでない場合は $r = r + 1$ と置き, Step 2 へ戻る.

なお, if $\xi_j \in \mathcal{D}_i^{(*)}$, then $(x(j), y(j)) \in \mathcal{F}_i$ という規則に従って, 部分集合 $\{\mathcal{F}_i\}_{i=1}^s$ を求めることができる.

2.2.1 最小二乗推定によるパラメータベクトルの同定

クラスタリングによって得られた各部分集合 $\{\mathcal{F}_i\}_{i=1}^s$ に対して, 最小二乗推定によりパラメータベクトル $\{\theta_i\}_{i=1}^s$ を同定する. そこで, データ空間上でのアウトライアの影響を軽減するために, 加重最小二乗推定,

$$\min_{\theta_i} \sum_{(x_t, y_t) \in \mathcal{F}_i} w_t \|y_t - \theta_i' [x_t' \ 1]'\|^2, \quad (2.15)$$

$$w_t = \frac{1}{\sqrt{(2\pi)^{n+1} \det(R_t)}} \quad (2.16)$$

によりパラメータベクトル $\{\theta_i\}_{i=1}^s$ を同定する. ここで w_t は重みであり, 平均 ξ_t , 分散 R_t のガウス分布の最大値を表す.

2.3 サポートベクターマシン

サポートベクターマシン (SVM:Support Vector Machine) は, 各クラスのデータと識別面との最小距離を表すマージンが最大になるように識別を行うための, パターン認識問題に対して最も優れた学習手法として知られている. この手法は, 入力データ $x_{\tilde{t}}$ ($\tilde{t} = 1, \dots, \tilde{N}$) ($\tilde{N} \leq N$) と, それに対応するクラスラベル $\eta_{\tilde{t}} = \{1, -1\}$ からなるトレーニングデータ $(x_{\tilde{t}}, \eta_{\tilde{t}})$ を, 線形識別関数,

$$f(x) = w'x + b \quad (2.17)$$

により識別する. Fig. 2.1 にその概念図を示す. ここで w および b は, 重みベクトルおよびバイアス項と呼ばれるパラメータである. (2.17) 式におけるパラメータ w, b の算出方法を以下に示す.

ここで SVM は, マージンが最大となる超平面が汎化能力が最も高いとされているため, マージンを最大にするような超平面を求める問題を考える. マージンは,

$$\gamma = \min_{\tilde{t}=1, \dots, \tilde{N}} \frac{|w'x_{\tilde{t}} + b|}{\|w\|} \quad (2.18)$$

と表せる.

ここで, w, b は定数倍しても得られる超平面が同じであるという冗長性を持っている. このため, 識別面が一意に定まらない. そこで,

$$\min_{\tilde{t}=1, \dots, \tilde{N}} |w'x_{\tilde{t}} + b| = 1 \quad (2.19)$$

と制約を与える. また, (2.19) 式の制約を考慮すると, (2.18) 式は, $1/\|w\|$ と書き換えられる.

このようなマージン最大化問題は, 制約条件付き 2 次計画問題,

$$\min \|w\|^2 \quad (2.20)$$

$$\eta_{\tilde{t}}(w'x_{\tilde{t}} + b) \geq 1 (\tilde{t} = 1, \dots, \tilde{N}) \quad (2.21)$$

で表すことができる.

さらに, ラグランジュ未定乗数法を用いると, 最適化問題,

$$\begin{aligned} \max_{\lambda} Q(\lambda) &= \sum_{\tilde{t}=1}^{\tilde{N}} \lambda_{\tilde{t}} - \frac{1}{2} \sum_{\tilde{t}, j=1}^{\tilde{N}} \eta_{\tilde{t}} \eta_j \lambda_{\tilde{t}} \lambda_j x'_{\tilde{t}} x_j \\ \text{subject to } &\sum_{\tilde{t}=1}^{\tilde{N}} \eta_{\tilde{t}} \lambda_{\tilde{t}} = 0, \lambda_{\tilde{t}} \geq 0, \tilde{t} = 1, \dots, \tilde{N} \end{aligned} \quad (2.22)$$

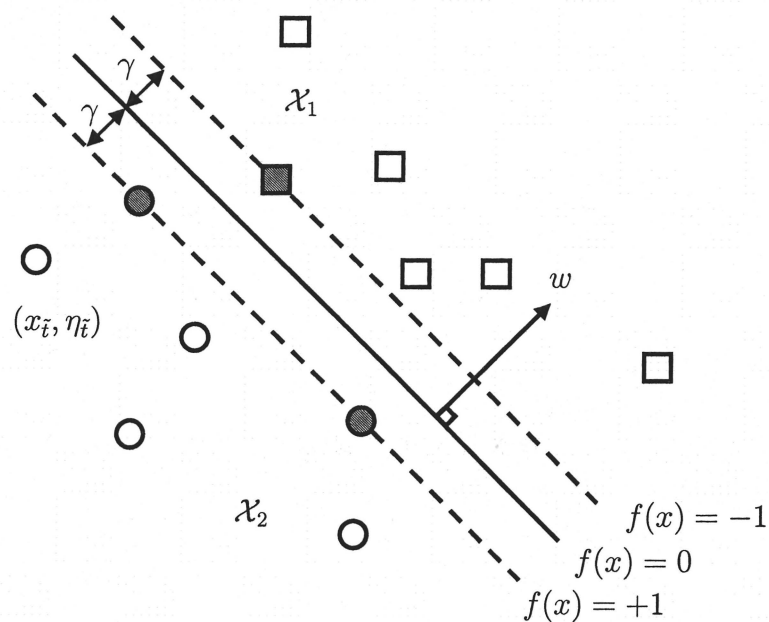


Fig. 2.1 線形識別関数を用いたトレーニングデータの線形分離

に帰着される．ここで，多くのデータに対して $\lambda_i = 0$ となるため， $\lambda_i \neq 0$ を満たすものがサポートベクターと呼ばれ，分離超平面の構成に寄与するデータに対応する．

第3章 PWARXモデルによるロボットの 運転操作の表現

本章では，ロボットの運転操作とその操作を表現する PWARX モデルについて述べる．3.1 節では本研究のロボットの操作データ取得環境，3.2 節では入出力データについて説明し，3.3 節では同定に用いるデータの選定方法，3.4 節では PWARX モデルのサブモデル数の決定，3.5 節では本稿で扱う人の認識情報について述べる．

3.1 データ取得環境

本研究では対象とする操作として，人が一定速度で直進しているロボットを S 字コースの中央に沿って走行させたときのステアリング操作に注目する．操作データ取得のための実験環境は Fig. 3.1 に示されるように，PC，コントローラ，CCD カメラ，ロボット，S 字コースから構成される．操作者は PC に接続されているコントローラを用いて操作する．コントローラからの操作情報は PC に送信され，Bluetooth による無線通信

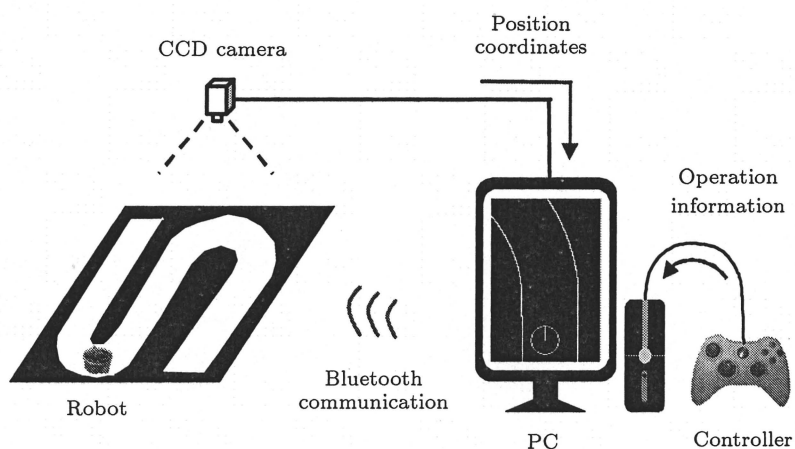


Fig. 3.1 データ取得環境

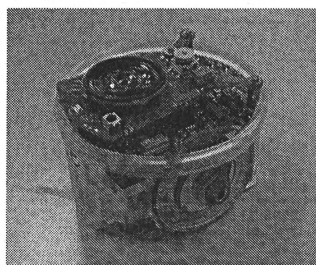


Fig. 3.2 ロボットの外観

を使用して 0.1[s] 間隔でロボットへ送信される。それと同時に、CCD カメラからロボットの位置情報を 0.1[s] 間隔で取得している。操作者の視点や経路の先の情報を限定するため、操作者は Fig. 3.1 の PC に示す画面を見てロボットを操作する。この画面の白い円はロボット、白線は壁を表しており、ロボットの進行方向を常に画面上方向に取るものとしている。これらの環境の実現には、Windows Visual C++. NET ver. 2002 を使用している。

使用するロボットは Fig. 3.2 に示されるような二輪駆動機構を有しており、底面の直径 8 [cm]、高さ 5 [cm] の円柱形である。このロボットは研究および教育用としてスイス連邦工科大学ローザンヌ校の自律システム研究所により開発された e-puck と呼ばれるロボットである。e-puck は、プロセッサとして dsPIC30F6014A、入出力デバイスとして赤外線近接センサや CMOS カメラ、マイクロフォン、Bluetooth 無線リンクなどを搭載している。また、左右二つの車輪は、20 [step/rotation] のモータにより、50:1 の比率の減速ギヤで駆動し、車輪の最大角速度は 2π [rad/s] である。本研究において、e-puck は操作者の意図するロボット動作を高精度に実現することができるハードウェアを備えていると言える。

コントローラーには、Windows PC 向けの xbox360 用コントローラーを使用する。このコントローラーを使用する利点として、ゲーム用に開発されているため、直感的に操作しやすいデザインである、Microsoft 公式のコントローラー用ドライバが用意されているため導入が容易である、Microsoft より XInput と呼ばれる xbox360 用コントローラーのための API が提供されているため、実機実験環境の開発に使用している VC++ との相性がよいことが挙げられる。

座標取得のためのカメラ入力システムには、ライブラリー社のリアルタイムトラッカー Radish を使用している。Radish SDK と呼ばれる画像処理用の C++ クラスライブラリが用意されているため、ユーザ独自のソフトウェアが開発でき、これにより本研

究の座標取得システムを実現している。CCD カメラはコース真上に備え付けられており、解像度は 640×480 [pixel]、最大 30[fps] で位置情報を取得することができる。

3.2 入出力データと人の認識情報

人がロボットを走行させたときに取得される入出力データは、操作者が Fig. 3.1 の画面上から認識している情報 \mathbf{x}_t とそれに伴い決定されるロボットの操作量 \mathbf{y}_t である。 \mathbf{x} がモデルの入力、 \mathbf{y} がモデルの出力を表している。したがって、モデルの入力は人が操作中に認識している情報であり、出力は人が操作する物理量である。

1 回の走行実験より、有限の入出力データ集合 $(\mathbf{x}_t, \mathbf{y}_t)$, $(t = 1, \dots, N)$ を取得する。 t は時刻を表す。本研究の環境から \mathbf{x} はロボットの現在位置や速度、コース壁までの距離などの情報が考えられ、 \mathbf{y} はロボットの回転方向の速度、直進方向の速度が考えられる。しかしながら、コースのスタートからゴールまで走行させるという操作を対象とする場合、最短で走行させる人もいれば、コースから外れないように速度落として走行させる操作者も考えられ、その上、操作者がどのように走行させるかによって操作中の認識情報も異なってくる。本研究では問題を簡単にするため、ロボットの走行に関して「ロボットは一定速度で直進方向に進ませ、回転方向の速度のみを操作し、コースの中央に沿って走行させる」という限定的にすることで、人の認識情報の範囲を制限している。また、人は回転方向の速度のみを操作するので \mathbf{y} は回転方向の速度 ω [rad/s] となり、 $y = \omega$ である。

3.3 使用データの選定

取得された 1 回分の走行データ $(\mathbf{x}_t, \mathbf{y}_t)$, $(t = 1, \dots, N)$ は、そのデータの質の良し悪しを評価し、良いデータのみをモデルの同定に用いることとする。ロボット操作に熟練している人とそうでない人との間には、コントローラーの操作感覚の慣れや画面などの情報を基に操作するかなどの経験の差が存在する。たとえば、操作経験の少ない人はロボットを思うように操作することができず、コースの中央から外れ、蛇行しているような悪い走行データになると考えられる。もしくは、熟練者が操作したとしても人間であるので人為的な操作ミスが生じてしまい、コースの中央から外れた悪い走行データが得られることも考えられる。本稿では良いデータのみを取り扱うことで同定に用いるデータの質をある一定以上のものとし、コースの中央に沿って走行させる

操作モデルに対する熟練していない人の操作データの影響を回避する．ここで，取得データの良し悪しを評価する条件式を以下に記述する．

$$\mu = \frac{1}{N} \sum_{t=1}^N d_t^{\text{center}} < 1 \quad (3.1)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{t=1}^N (d_t^{\text{center}} - \mu)^2} < 2 \quad (3.2)$$

d_t^{center} はコース中央からロボットがどれだけずれているかの距離を表し，コース中央よりも左側にずれている場合を正，右側を負とする．(3.1) 式，(3.2) 式はそれぞれ d_t^{center} の平均値，標準偏差を示している．すなわち，(3.1) 式，(3.2) 式を用いて，ある程度コースの中央に沿ったばらつきの少ない走行データを選別している．本稿では μ が 1， σ が 2 以下と設定しているが，この値は，操作者がある程度ロボットの操作に慣れ，そして操作ミスなく走行させたときに取得できる基準として，4 名の走行実験結果から経験的に決定した．

モデリングには 5 回分の走行実験データを使用する．これは，取得データに含まれるアウトライアの影響を小さくし，人の操作のばらつきを考慮するためである．したがって，(3.1) 式，(3.2) 式により選ばれた 5 回分の良い走行データに対してクラスタリングに基づく手法を適用し，PWARX モデルとして同定する．

3.4 PWARX モデルのサブモデル数

2.2 節で述べたクラスタリングに基づく同定手法は，予め (2.1) 式のサブモデル数 s を決定しなければならない．この s は切り替える ARX モデルの数を表しており，本稿で対象とする操作「人が一定速度で直進しているロボットをあるコースの中央に沿って走行させたときの回転方向速度の操作」がいくつかの要素的な操作から達成しているのかということに対応している．一定速度で直進しているロボットの回転方向速度のみを操作できる場合，直感的ではあるが，人の操作の種類として「左旋回」，「直進」，「右旋回」が考えられる．そこで本研究では $s = 3$ とし，次式のような PWARX モデルとして同定する．

$$y(\mathbf{x}) = \begin{cases} \theta'_1 \varphi & \text{if } \mathbf{x} \in \mathcal{X}_1 \text{ (左折モード)} \\ \theta'_2 \varphi & \text{if } \mathbf{x} \in \mathcal{X}_2 \text{ (直進モード)} \\ \theta'_3 \varphi & \text{if } \mathbf{x} \in \mathcal{X}_3 \text{ (右折モード)} \end{cases} \quad (3.3)$$

以降, $i = 1$ を「左折モード」, $i = 2$ を「直進モード」, $i = 1$ を「右折モード」と呼んでいく.

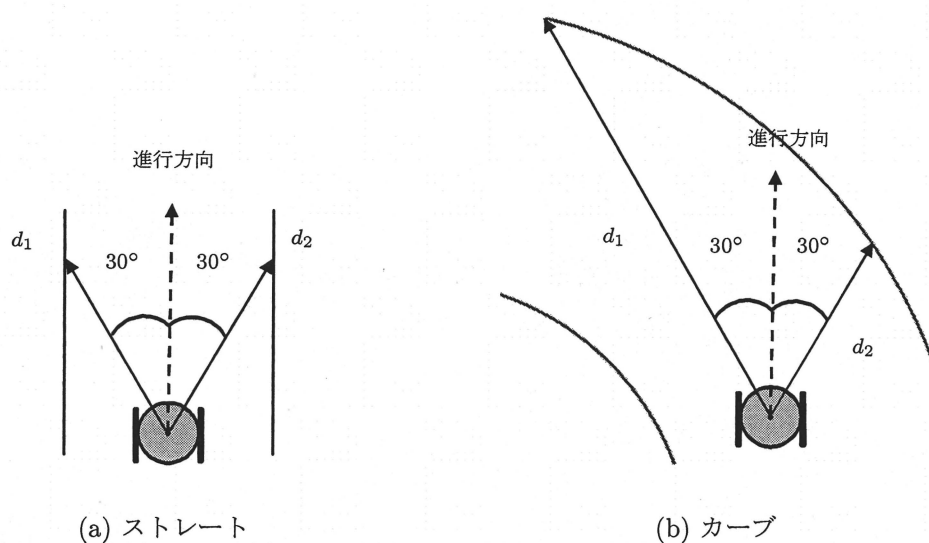
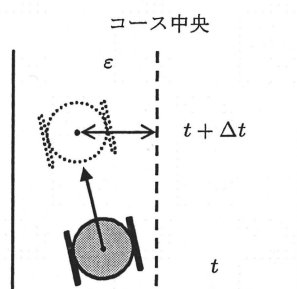
入力 \mathbf{x} と操作時の人の認識情報の関係についても簡単に述べておく. 人の操作量を表す出力 y は (3.3) 式から分かるように, パラメータベクトル $\boldsymbol{\theta}$ と入力ベクトル $\boldsymbol{\varphi} = [\mathbf{x}' \ 1]'$ の積で表される. また, モデルの切り替え条件においても \mathbf{x} がどの領域の \mathcal{R}_i に存在するかに依存している. これより \mathbf{x} は PWARX モデルを決定する重要なものであり, 入力の各要素を操作時の人の認識情報とすることで, PWARX モデルは人の操作を表現することができる. したがって, \mathbf{x} をどのような認識情報で構成するかが, モデルの特性を大きく変える大きな要因となる.

3.5 人の認識情報

これまでの先行研究で提案された 2 種類の認識情報, 「ロボットから左右の壁までの距離」と「コース中央とロボットの予測位置との間の距離」を本稿においても着目する. 前者^[26]と後者^[27]では対象とする操作に違いがあり, 文献^[26]では, 単にロボットをコースから外れないよう走行させる, 文献^[27]では, 本稿と同様, コースの中央に沿って走行させることを目的としていた. これより, 前者の情報は単にコースに沿って走行させる場合には影響はないが, コースの中央を認識している情報ではないため, 中央に沿って走行できるモデルを同定できるかどうか分からない. 認識情報の違いが操作モデルにどのような差異をもたらすのかを明らかにするためにも検討していく. 最終的には, 前者の情報を用いたモデルよりも後者の情報を用いたモデルの方が人の操作を表現しており, 人が認識している情報として後者の方が重要度が高いと言えるような結果を期待して進めていく. 前者の情報で構成された入力を \mathbf{x}_I , 後者の情報で構成された入力を \mathbf{x}_{II} とし, 次に二つの情報がどのようなものなのか詳細を述べる.

3.5.1 \mathbf{x}_I

\mathbf{x}_I では人が認識している情報として, Fig. 3.3 に示される d_1 と d_2 を考える. 前述したように, ロボットをコースから外れないように走行させるとき, 操作者は左右の壁までの距離を見ながら操作していると考えられる. ロボットの進行方向に対して $+30$ [deg] 方向の壁までの距離を d_1 , -30 [deg] 方向の距離を d_2 としている. したがって, $\mathbf{x}_I = [d_1 \ d_2]'$ となる. Fig. 3.3 の (a) のように d_1 と d_2 がほぼ同じ距離のときは直進モード, Fig. 3.3

Fig. 3.3 d_1 , d_2 Fig. 3.4 ε

の(b)のように d_1 が d_2 に比べ大きいときは左折モード, 反対に d_2 が d_1 に比べ大きいときは右折モードと考えることができる。

3.5.2 x_{II}

x_{II} では人が認識している情報として, Fig. 3.4に示される ε を考える. 人はロボットをコースの中央に沿って走行させるために, 予測時間 Δt [s] 先のロボットの位置を予測し, その位置が中央からどれだけずれているかを基に, 操作していると考え. ロボットの予測位置から最短のコース中央までの距離を ε とし, コース中央を基準として正の方向がコース左側, 負の方向がコース右側にずれているとする. したがって,

Δt	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
Driver A	2.48	2.15	2.05	1.99	1.84	1.75	1.75	1.93	1.82	2.06	2.03
Driver B	4.94	4.11	2.46	1.94	1.76	1.71	1.66	1.55	1.61	1.54	1.59
Driver C	2.04	1.88	1.76	1.81	1.69	1.66	1.72	1.64	1.64	1.69	1.71
Driver D	2.49	2.21	2.05	1.93	1.86	1.84	1.79	1.84	1.78	1.81	1.90

Table 3.1 人とモデルの走行軌道の差の平均

$x_H = \varepsilon$ となる. Fig. 3.4 のように ε が正の方向に大きい場合は右折モード, 反対に, 負の方向に大きい場合は左折モード, そして, 0 辺りの場合は直進モードと考えることができる.

文献^[28]において, Land らにより人が車を運転するときに認識している経路情報に関する研究が行われている. この中で, 60[km/h] で走行中の場合, 約 1[s] 先, すなわち自車よりも約 16[m] 前の経路情報を利用していることが報告されている. これは, 自車よりも前の情報を認識するという点で ε と共通しており, ε が考えるべき情報であることを裏付けているが, 車のシートに座った運転者から見た視点と Fig. 3.1 のロボットを真上から見た視点といった操作者の視点の違いや, 車とロボットの走行速度の違いなどにおいて本研究と環境が異なる. したがって, 本研究における予測時間 Δt を決定する必要がある.

本稿では, 人の操作に最も近い Δt を決定するために, 人の走行軌道と操作モデルを用いた自動走行軌道の差を全軌道点に対し計算し, その軌道差の平均をもって人の操作と各モデルの類似度を評価する. すなわち, 差の平均が最も小さい自動走行軌道が, 人の走行軌道に最も近いといえる. モデルを用いた自動走行については 4.3 節にて述べるが, 同定された操作モデルを人の代わりに用いることで, ロボットを自動で走行させることが可能となる^[26]. Δt に関して, 文献^[27]においても言及されており, $\Delta t = 1.5[s]$ が最も人の操作に近いということが報告されている. しかしながら, 時間の都合上, $\Delta t = 0.5, 1.0, 1.5, 2.0$ の 4 点での比較しかなく, 加えて, 検討された操作者が 1 名ということもあり, Δt を決定するために不十分である. 本稿では, 操作者 4 名分の走行データを取得し, それらに対して Δt を 0.1[s] から 2.0[s] の間で 0.1[s] ずつ変化させ, それぞれ操作モデルを同定していき, Fig. 3.1 と同じコースを実機実験により自動走行させる. 操作者はそれぞれ Driver A, Driver B, Driver C, Driver D とする. そのときの変化の最小値周りである $\Delta t = 0.5 \sim 1.5[s]$ の結果を Table. 3.1 に示す. Table. 3.1 より Driver A は 1.0[s], Driver B は 1.4[s], Driver C は 1.3[s], Driver D は

1.3[s] のとき，最も人に近い軌道を走行させるモデルが得られることが分かった．操作者間で最小値にばらつきが生じたため，4 名分の軌道差の平均が最も小さい 1.0[s] を Δt とする．以降，本稿において $\Delta t = 1.0$ であり， ε は $\Delta t = 1.0$ で計算したものとする．

第4章 前任者による人の認識情報の重要性の評価

前任者の研究^[27]では、人の認識情報をロボットの進行方向左右 30° の壁までの距離 $\mathbf{x}_I = [d_1 \ d_2]'$ とした Model I と 1[s] 先に予測されるロボットの位置から中央までの距離 $\mathbf{x}_{II} = \varepsilon$ とした Model II の二つのモデルに着目し、どちらの認識情報がより人のロボット操作を表現できているかを検証していた。その表現性を評価するため得られた操作モデルをロボットの自動走行に応用し、モデルによる走行軌道と人の操作による走行軌道を比較し、その類似性をもって評価していた。本章では、前任者の評価法を思い出してもらうために、4.1 節に前任者の評価法についての説明、4.2 節に Model I と Model II の同定、4.3 節に操作モデルによる自動走行結果と認識情報の重要性評価について、4.4 節に前任者の評価法の問題点を述べる。さらに、前任者では操作者 1 名にのみ対して評価していたが、本研究では 4 名の操作者に対して評価を行い、異なる操作者においても同様の結果が得られるか確認した。

4.1 前任者の評価法

本節では前任者の認識情報の重要性の評価法について述べる。Fig. 4.1 に前任者による評価法の手順を示す。前任者の評価法では、まず、Fig. 3.1 の環境から人の入出力データを取得し、次に、クラスタリングに基づく同定手法により PWARX モデルを得る。そして、得られたモデルを用いてロボットの自動走行へ応用し、操作モデルによる

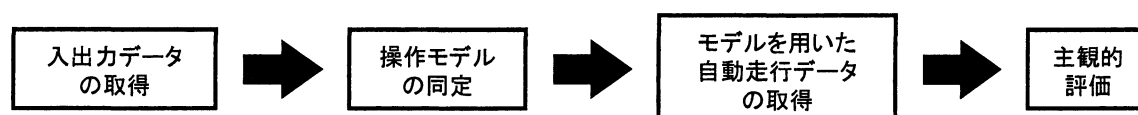


Fig. 4.1 前任者の評価手順

Data	1	2	3	4	5
Driver A	-0.160	-0.085	0.079	0.138	0.604
Driver B	0.310	-0.045	0.277	-0.091	0.165
Driver C	-0.222	0.626	-0.276	-0.058	0.002
Driver D	-0.684	0.203	-0.013	-0.377	0.350

Table 4.1 走行データの質 μ

Data	1	2	3	4	5
Driver A	1.50	1.72	1.21	1.00	1.18
Driver B	1.69	1.40	1.80	1.80	1.42
Driver C	1.99	1.31	1.82	1.53	1.91
Driver D	1.79	1.52	1.66	1.66	1.57

Table 4.2 走行データの質 σ

る走行軌道を取得する．最後に，最初に取得した人の走行軌道とモデルによる走行軌道の結果をプロットし，その見た目の類似性を評価していた．たとえば，二つの軌道が重なっているならば，その操作モデルは人の操作を表現できている，その逆であるなら，表現できていないというように主観的に判断していた．このようにして類似性を評価し，人のロボット操作を表現できるのはどの認識情報かを検証していた．

次節より文献^[27]と同じように人が一定速度で直進しているロボットをS字コースの中央に沿って走行させたときのステアリング操作を対象として，人の認識情報を $\mathbf{x}_I = [d_1 \ d_2]'$ とした Model I と $\mathbf{x}_{II} = \varepsilon$ とした Model II の二つを評価していく．そして，どちらの認識情報のモデルが人の操作を表現できているかを検証していく．

4.2 同定用データとモデリング

4.2.1 同定用データの質

操作モデルを同定するため，人の操作から5回分の良い走行データを取得する．操作者は Driver A, Driver B, Driver C, Driver D の4名を対象とする．Fig. 4.2 と Table. 4.1, Table. 4.2 に操作者4名分のモデル同定に用いる走行データの軌道と走行データの質の評価結果を示す．3.3 節で述べたように， μ が1， σ が2以下の良いデータを取得した．5回分の走行でどの操作者も約2200点のデータが得られ，これらに対してクラスタリングに基づく同定手法を適用し，Model I と Model II を同定する．

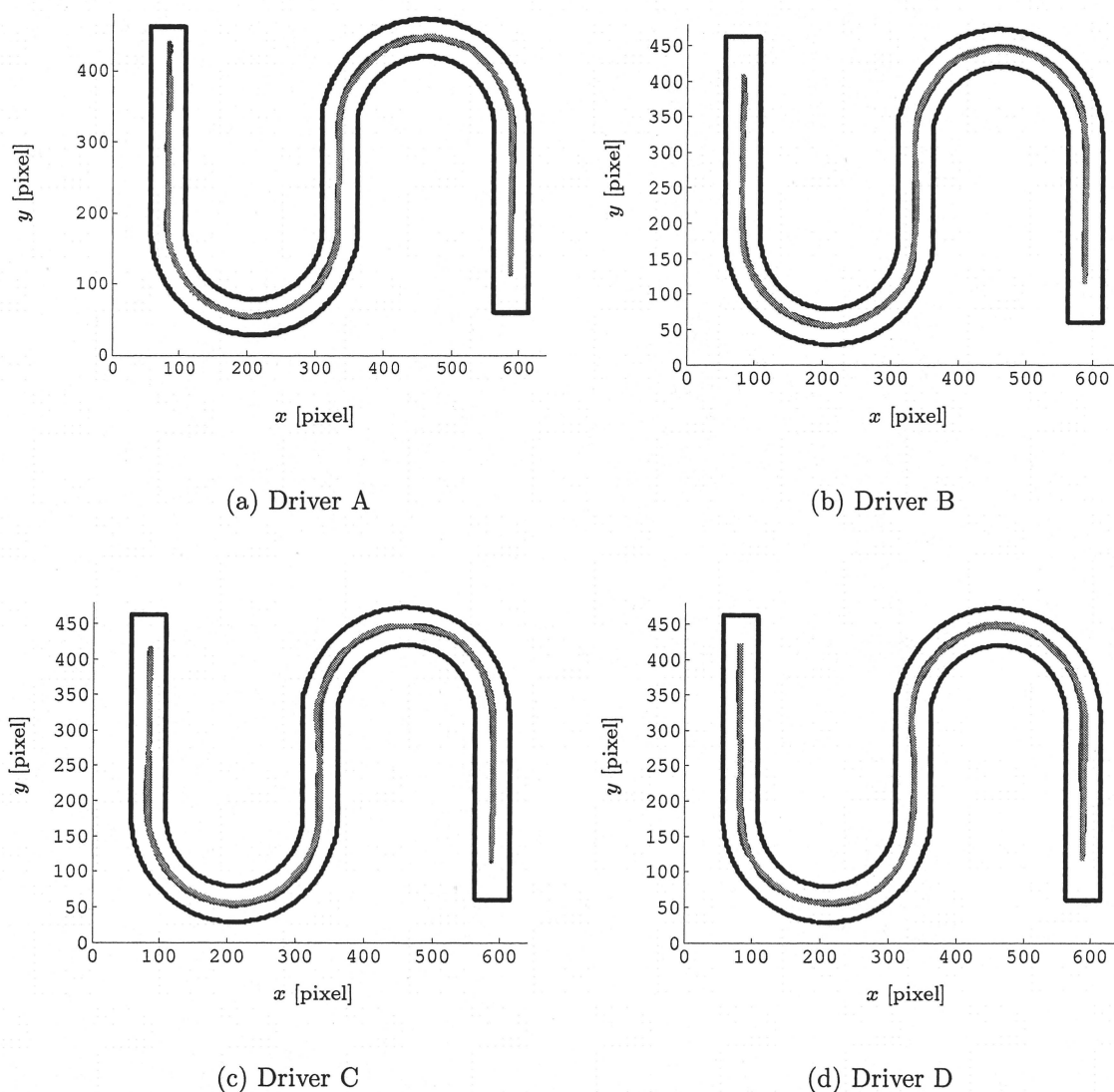
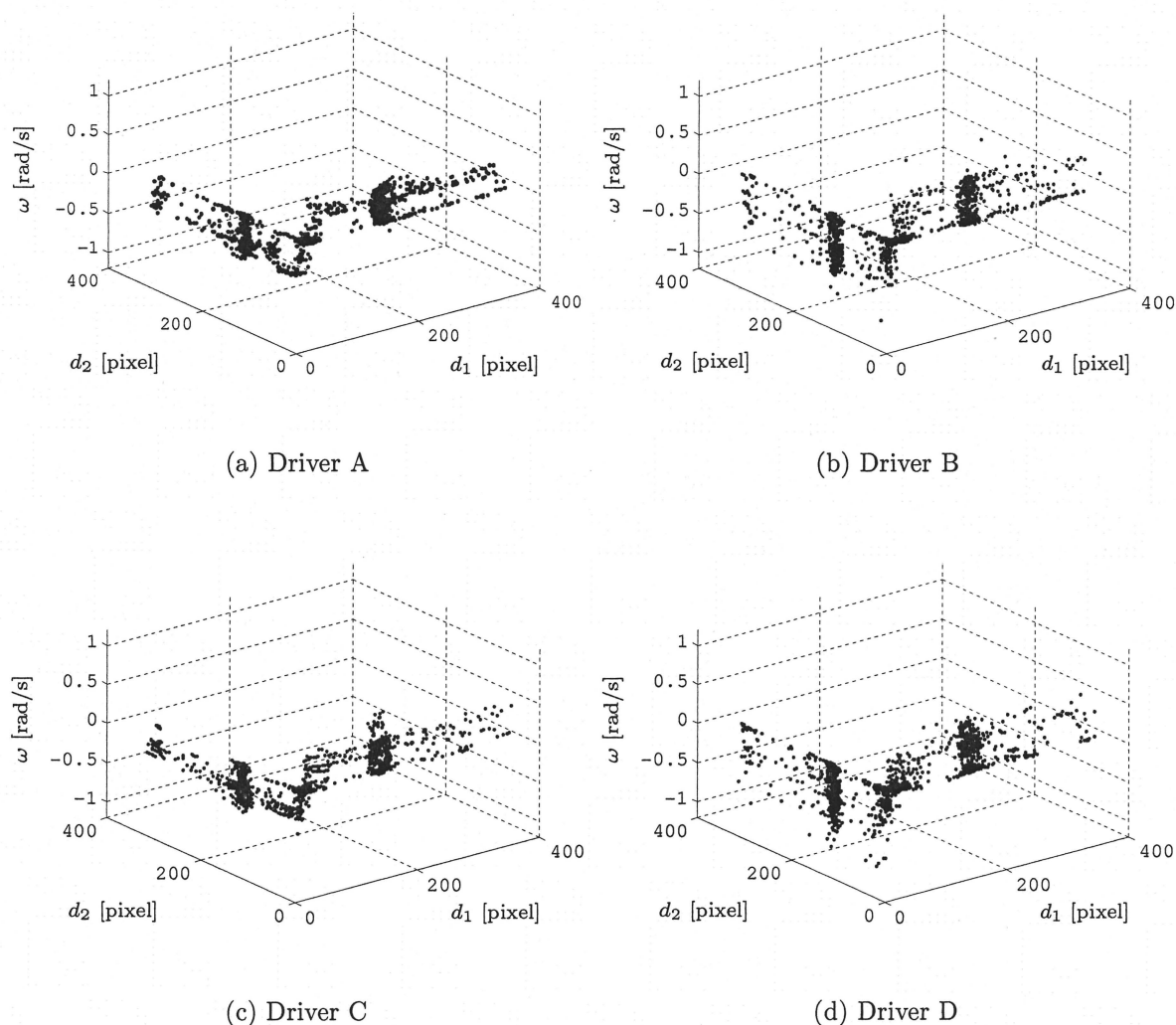


Fig. 4.2 人の操作による走行軌道

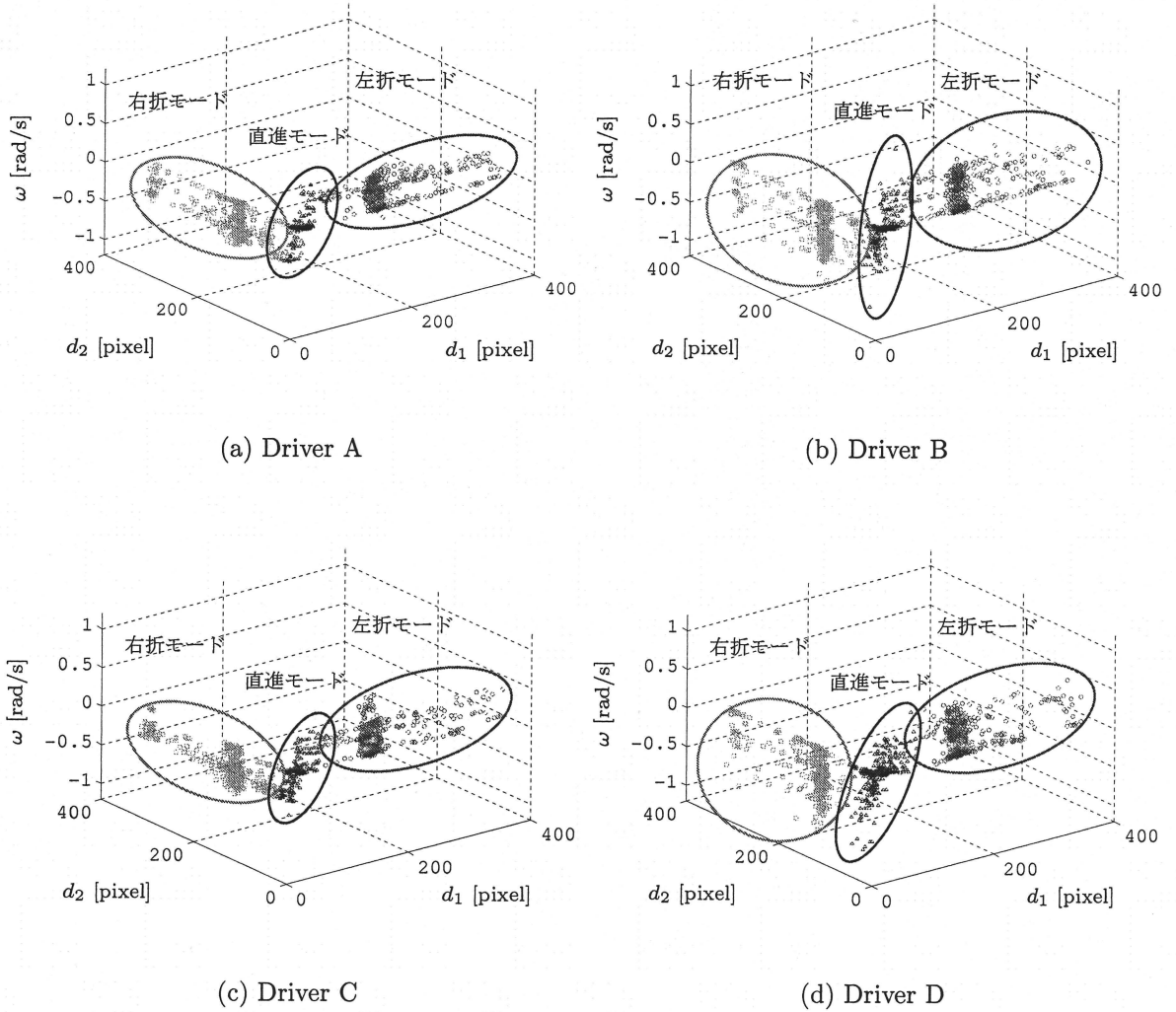
4.2.2 Model Iの同定

Fig. 4.3に操作者4名より得られた5回分の走行時の入出力データ $(\mathbf{x}_t, \mathbf{y}_t)$ を示す. 入出力データはそれぞれ $\mathbf{x} = [d_1 \ d_2]'$, $y = \omega$ であるため, データ空間は3次元となる. d_1 , d_2 の単位は [pixel], ω は [rad/s] である. 1 [pixel] は約 2.8 [mm] であり, ω は, 正方向の場合, 左旋回負方向の場合, 右旋回となる. Fig. 4.3 より, d_1 が d_2 がよりも比較的大きく, ω が $\omega > 0$ の値をとるデータの集合, 反対に, d_2 が d_1 がよりも比較的大きく, ω が $\omega < 0$ の値をとるデータ集合, そして, d_1 と d_2 ほぼ同じで, ω が 0 の値を

Fig. 4.3 入出力空間 $d_1 - d_2 - \omega$

とるデータ集合があることが確認できる。したがって、3.5.1で述べたように、 d_1 と d_2 の関係の変化に伴って、操作者はコントローラーの操作量 ω を変化させていることを確認できた。これらの入出力空間上の操作データを3つの部分集合に分割するために、 $s = 3$ でクラスタリングした結果が、Fig. 4.4 となる。局所集合数は試行錯誤的に決定している。○（赤）は左折モード、△（青）は直進モード、□（緑）は右折モードを示す。Fig. 4.4 より、左折モードを構成するデータ、直進モードを構成するデータ、右折モードを構成するデータがそれぞれ同じ特徴を持つ集合として分類できていることを確認できた。

次に、得られた各モードごとに対して、最小二乗推定によりパラメータベクトル

Fig. 4.4 クラスタリング後の入出力空間 $d_1 - d_2 - \omega$

$\theta_1 = [\theta_{11} \ \theta_{12} \ \theta_{13}]'$, $\theta_2 = [\theta_{21} \ \theta_{22} \ \theta_{23}]'$, $\theta_3 = [\theta_{31} \ \theta_{32} \ \theta_{33}]'$ が得られる. 4名分の θ_1 , θ_2 , θ_3 を Table 4.3 に示す.

最後に, 得られた各モードを切り替えるために, それぞれのモードの領域 \mathcal{X}_i を SVM を用いて求める. 具体的には, 入力空間上の各モードを形成するデータ集合に対して, 上記で得られたクラスタリング結果を教師データとし, 線形識別関数を求めていく. ここで, SVM は二つの集合を分ける 2 クラス分類器であるので, 本来ならば, 三つの集合を分類するには, 多クラス SVM を用いなければならない. しかしながら, 本稿においては, 左折モードから右折モード, 右折モードから左折モードへ直接遷移せず, 直進モードを必ず経由して遷移すると考え, 左折モードと直進モード, 右折モードと

	θ_{11}	θ_{12}	θ_{13}	θ_{21}	θ_{22}	θ_{23}	θ_{31}	θ_{32}	θ_{33}
Driver A	-2.386×10^{-4}	-0.021	1.143	4.159×10^{-4}	2.591×10^{-5}	-0.22	-0.022	-1.914×10^{-5}	-1.198
Driver B	-2.463×10^{-4}	-0.025	1.366	-0.001	-0.002	0.143	0.032	-2.884×10^{-4}	-1.572
Driver C	-2.464×10^{-4}	-0.019	1.038	-0.002	-0.002	0.198	0.0180	-1.776×10^{-4}	-0.961
Driver D	-5.106×10^{-4}	-0.024	1.347	-0.002	-0.006	0.421	0.035	-7.385×10^{-5}	-1.683

Table 4.3 Model I のパラメータベクトル

	w_{11}	w_{12}	b_1	w_{21}	w_{22}	b_2
Driver A	0.73	0.08	-72.17	-0.12	0.54	-34.42
Driver B	1.04	-1.23	-46.81	-0.02	1.07	-91.58
Driver C	0.88	-0.24	-74.99	-0.90	1.07	-41.62
Driver D	0.36	-0.01	-36.82	-0.03	0.17	-17.34

Table 4.4 Model I の識別関数パラメータ

直進モードの二つの線形識別関数を導出する，したがって，本研究では多クラス SVM ではなく 2 クラス SVM を用いる．左折モードと直進モードを分類する線形識別関数 $f_1(\mathbf{x}) = \mathbf{H}_1' \boldsymbol{\varphi}$ のパラメータを $\mathbf{H}_1 = \begin{bmatrix} w_{11} & w_{12} & b_1 \end{bmatrix}'$ ，右折モードと直進モードを分類する線形識別関数 $f_2(\mathbf{x}) = \mathbf{H}_2' \boldsymbol{\varphi}$ のパラメータを $\mathbf{H}_2 = \begin{bmatrix} w_{21} & w_{22} & b_2 \end{bmatrix}'$ とすると，4 名分の識別関数パラメータは Table 4.4 のように求められた．Fig. 4.5 に入力空間上のクラスタリング結果に線形識別関数をプロットした図を示す．Fig. 4.5 より黒の実線 $f_1(\mathbf{x})$ により左折モードと直進モード，黒の点線 $f_2(\mathbf{x})$ により右折モードと直進モードを分類できていることを確認できる．そして， $f_1(\mathbf{x})$ と $f_2(\mathbf{x})$ を用いて，それぞれのモードの領域を次式のように決定した．

$$\mathcal{X}_1 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} > 0 \wedge \mathbf{H}_2' \boldsymbol{\varphi} < 0\} \quad (4.1)$$

$$\mathcal{X}_2 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} \leq 0 \wedge \mathbf{H}_2' \boldsymbol{\varphi} \leq 0\} \quad (4.2)$$

$$\mathcal{X}_3 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} < 0 \wedge \mathbf{H}_2' \boldsymbol{\varphi} > 0\} \quad (4.3)$$

これらの式より Fig. 4.5 の赤色の一点鎖線で囲まれた領域を左折モード，青色の領域を直進モード，緑色の領域を右折モードと決定することができた．以上より，Model I を同定することができた．

4.2.3 Model II の同定

Fig. 4.6 に 5 回分の走行時の操作データ $(\mathbf{x}_t, \mathbf{y}_t)$ の入出力空間上のプロットを示す． $x = \varepsilon$ ， $y = \omega$ であるため，入出力空間は 2 次元となる． ε の単位は [pixel] であり，正の

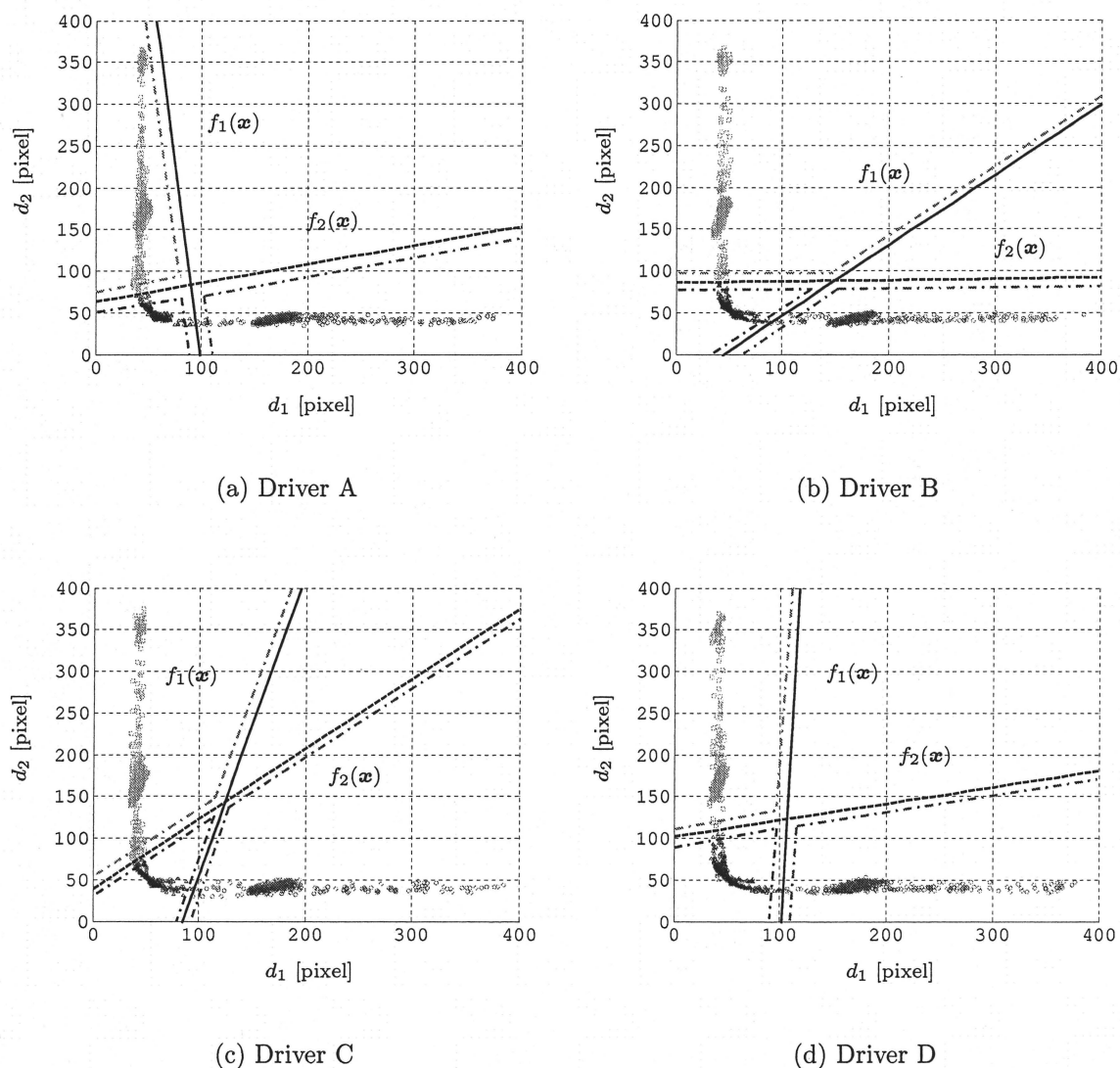
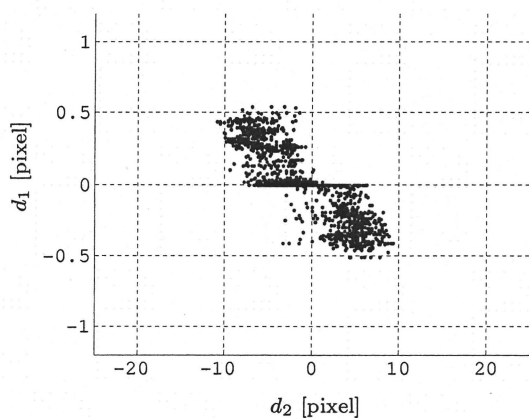
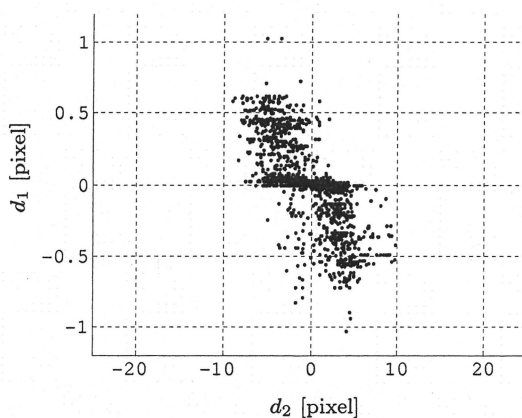


Fig. 4.5 Model I の各モードの領域

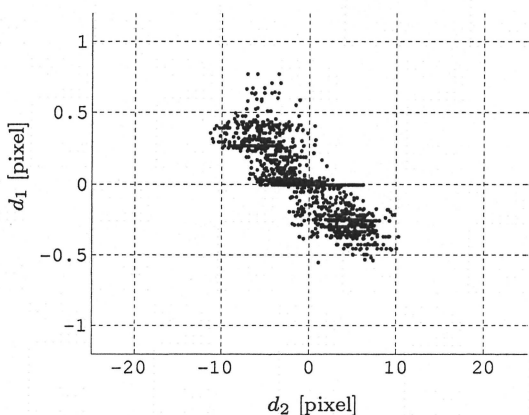
場合は左側，負の場合は右側に中心線からずれている．Fig. 4.6 より，ロボットが左側にいる ε が正の領域では右に旋回するため $\omega < 0$ の値をとるデータ集合，反対に，ロボットが右側にいる ε が負の領域では左に旋回するため $\omega > 0$ の値をとるデータ集合，そして，ロボットがコース中央付近にいる ε が 0 近傍の領域では，直進するため ω がほぼ 0 の値をとるデータ集合があることが確認できる．したがって，3.5.2 で述べたように， ε の変化に伴って，操作者はコントローラーの操作量 ω を変化させていることを確認できた．これらの入出力空間上の操作データ群を $s = 3$ でクラスタリングした結果が，Fig. 4.7 となる．局所集合数は試行錯誤的に決定している．○（赤）は左折モー



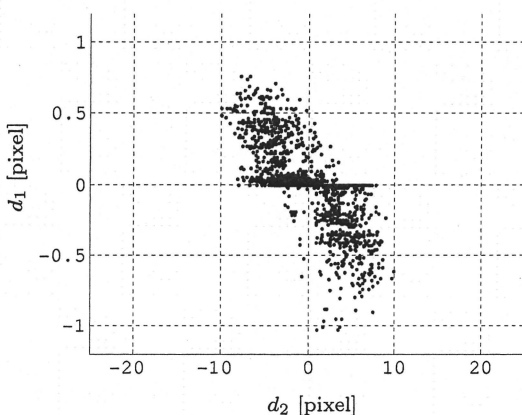
(a) Driver A



(b) Driver B



(c) Driver C



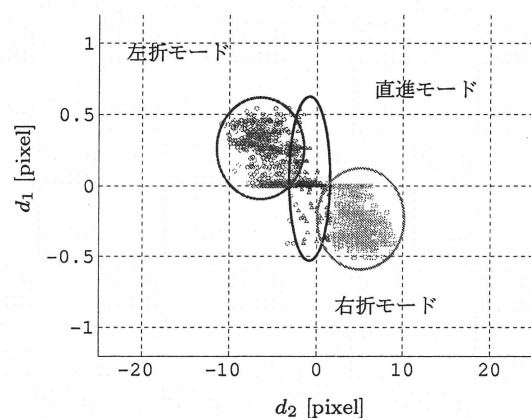
(d) Driver D

Fig. 4.6 入出力空間 $\varepsilon - \omega$

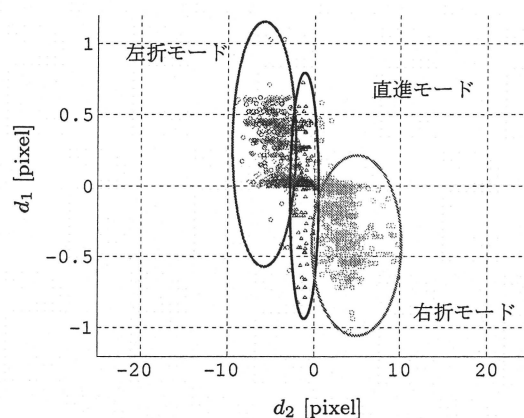
ド, \triangle (青) は直進モード, \square (緑) は右折モードを示す. Fig. 4.7 より, 左折モードを構成するデータ, 直進モードを構成するデータ, 右折モードを構成するデータがそれぞれ同じ特徴を持つ集合として分類できていることを確認できた.

次に, 得られた各モードごとに対して, 最小二乗推定によりパラメータベクトル $\theta_1 = [\theta_{11} \ \theta_{12}]'$, $\theta_2 = [\theta_{21} \ \theta_{22}]'$, $\theta_3 = [\theta_{31} \ \theta_{32}]'$ が得られる. 4 名分の θ_1 , θ_2 , θ_3 を Table 4.5 に示す.

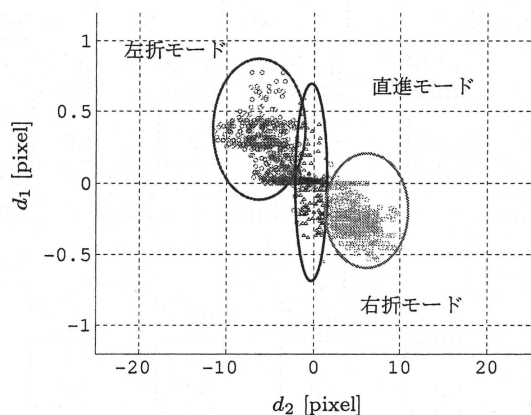
最後に, それぞれのモードの領域 \mathcal{R}_i を求める. 4.2.2 でも述べたように, 左折モードから右折モード, 右折モードから左折モードへの直接モード遷移はないと考え, 左折



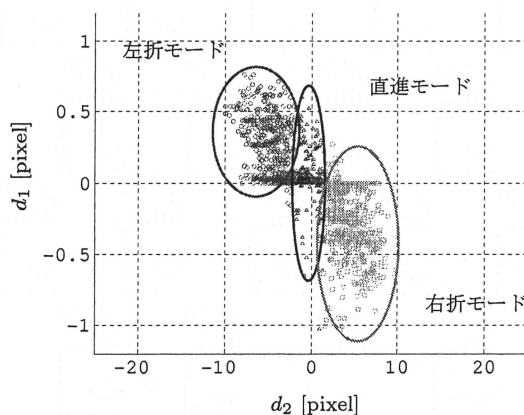
(a) Driver A



(b) Driver B



(c) Driver C



(d) Driver D

Fig. 4.7 クラスタリング後の入出力空間 $\varepsilon - \omega$

モードと直進モード，右折モードと直進モードの二つの線形識別関数を導出する．左折モードと直進モードを分類する線形識別関数 $f_1(\mathbf{x}) = \mathbf{H}_1' \boldsymbol{\varphi}$ のパラメータを $\mathbf{H}_1 = [w_1 \ b_1]'$ ，右折モードと直進モードを分類する線形識別関数 $f_2(\mathbf{x}) = \mathbf{H}_2' \boldsymbol{\varphi}$ のパラメータを $\mathbf{H}_2 = [w_2 \ b_2]'$ とすると，識別関数パラメータはそれぞれ Table 4.6 のように求められた．Fig. 4.8 に，入力空間上のクラスタリング結果に線形識別関数をプロットした図を示す．Fig. 4.8 より，黒の実線 $f_1(\mathbf{x})$ により左折モードと直進モード，黒の点線 $f_2(\mathbf{x})$ により右折モードと直進モードを分類できていることを確認できた．また， $f_1(\mathbf{x})$

	θ_{11}	θ_{12}	θ_{21}	θ_{22}	θ_{31}	θ_{32}
Driver A	-0.050	-0.082	-0.008	-0.003	-0.035	-0.075
Driver B	-0.029	-6.955×10^{-4}	-0.070	-0.074	-0.066	0.039
Driver C	-0.047	-0.045	-0.024	-0.009	-0.024	-0.139
Driver D	-0.060	-0.043	-0.020	0.019	-0.056	0.47

Table 4.5 Model II のパラメータベクトル

	w_1	b_1	w_2	b_2
Driver A	-107	-245	92	-147
Driver B	-363	-823	288	-63
Driver C	-142	-229	47	-83
Driver D	-316	-541	570	-804

Table 4.6 Model II の識別関数パラメータ

と $f_2(\mathbf{x})$ を用いて、それぞれのモードの領域を次式のように決定した.

$$\mathcal{X}_1 = \{\mathbf{x} : \mathbf{H}'_1 \boldsymbol{\varphi} > 0 \wedge \mathbf{H}'_2 \boldsymbol{\varphi} < 0\} \quad (4.4)$$

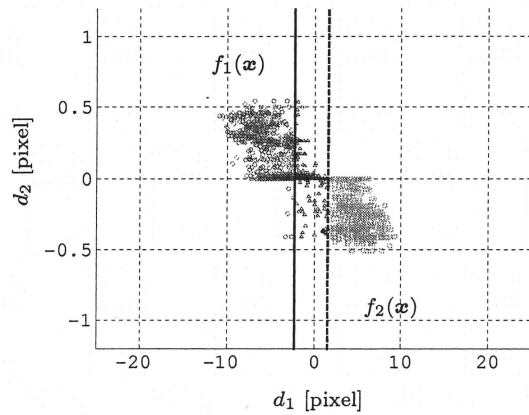
$$\mathcal{X}_2 = \{\mathbf{x} : \mathbf{H}'_1 \boldsymbol{\varphi} \leq 0 \wedge \mathbf{H}'_2 \boldsymbol{\varphi} \leq 0\} \quad (4.5)$$

$$\mathcal{X}_3 = \{\mathbf{x} : \mathbf{H}'_1 \boldsymbol{\varphi} < 0 \wedge \mathbf{H}'_2 \boldsymbol{\varphi} > 0\} \quad (4.6)$$

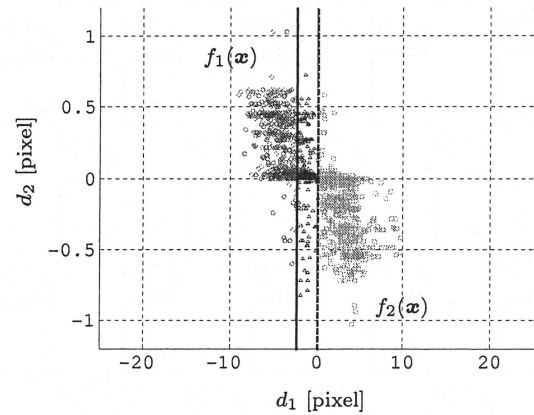
したがって、入力である ε に関して $\varepsilon < -2.3$ のとき左折モード、 $-2.3 < \varepsilon < 1.6$ のとき直進モード、 $\varepsilon > 1.6$ のとき右折モードとなる領域が求められたことが分かる. 以上より、Model II を同定することができた.

4.3 自動走行結果と重要性の評価

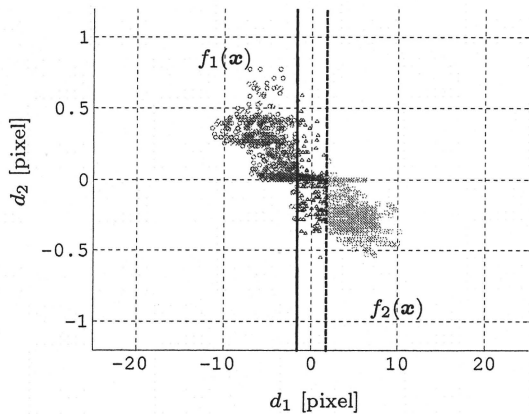
Fig. 4.9 に操作モデルを用いた自動化の環境を示す. Fig. 4.9 のように、人の操作の代わりとしてモデルが操作量 y を決定し、それをロボットに印加することで、ロボットの自動走行を実現する. Model I による自動走行の結果を Fig. 4.10 に、Model II による自動走行の結果を Fig. 4.11 に示す. 5 本の実線が人の操作による走行軌道、点線が操作モデルによる走行軌道を表している. Fig. 4.10 より Model I では最初のストレートや二つのカーブでは人とほぼ同じ軌道を走行させることができているが、コースの 2 番目、3 番目のストレートで人の走行軌道から大きく外れていることが分かる. 4.11 より Model II はスタートからゴールまで人の軌道とほぼ同じ軌道を走行させることができていることが確認できる. 図から判断する限り、どの操作者においても人の走行軌道により類似しているモデルは Model II である. 人であればコースの中央を走行させ



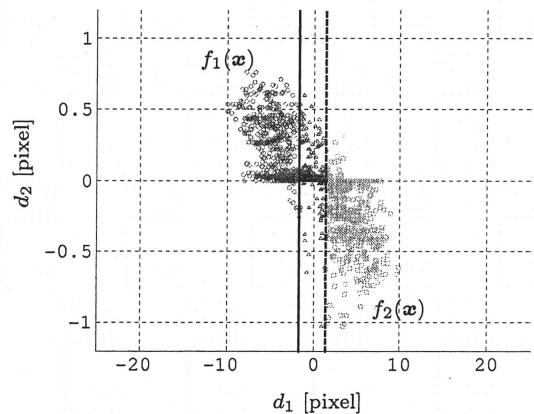
(a) Driver A



(b) Driver B



(c) Driver C



(d) Driver D

Fig. 4.8 Model II の各モードの領域

るという目的のため外れようとする前に中央へ戻すような操作量をロボットに加えると考えられるが、認識情報 d_1 , d_2 ではそのように操作を行うモデルが得られなかった、つまり ε に比べ d_1 , d_2 の情報では人の操作を表現できなかったと言える。したがって、一定速度で直進するロボットをコースの中央に沿って走行させるステアリング操作を行う上で d_1 , d_2 よりも ε の方が認識情報として重要性が高いと考えられる。

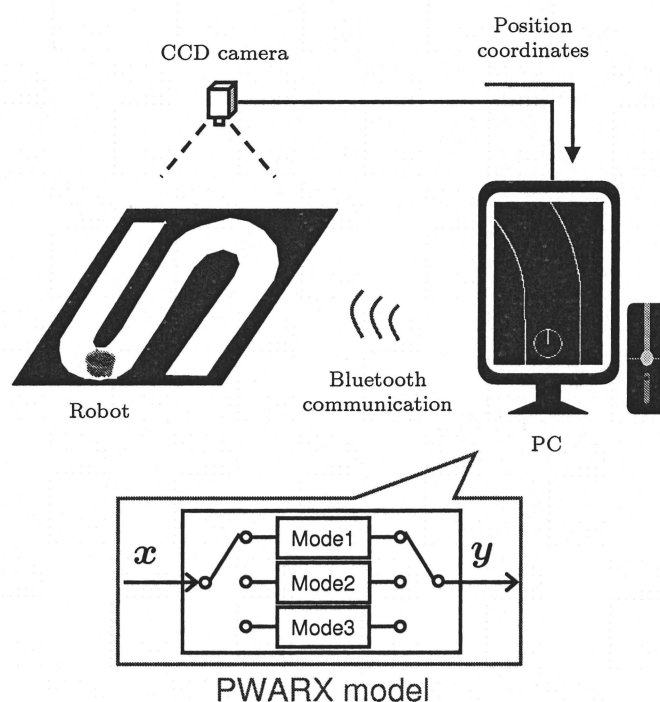


Fig. 4.9 自動走行環境

4.4 前任者の評価法の問題点

前節でモデルによる自動走行結果から d_1 , d_2 よりも ε の方が人の認識情報として重要性が高いという結果が得られた。しかしながら、この評価法には以下の四つの問題点がある。

- (1) 比較したい認識情報ごとに異なるモデルを用いているため、認識情報間の重要性を純粋に比較できない
- (2) モデルの表現性の評価が主観的である
- (3) モデルによる自動走行結果が必要であり煩雑である
- (4) 人の操作の表現性を評価するのに人とモデルの出力を考えていない

(1) について、たとえば4.3節のように2種類の認識情報間の重要性の優劣を決定したい場合、これらの二つの異なるモデルを用いて評価していた。そのため認識情報の違いだけでなく、二つのモデルの次数やSVMの結果、モデルの表現性などの違いも自動走行結果に影響しており、異なるモデルを用いることで認識情報間の対等かつ純粋な

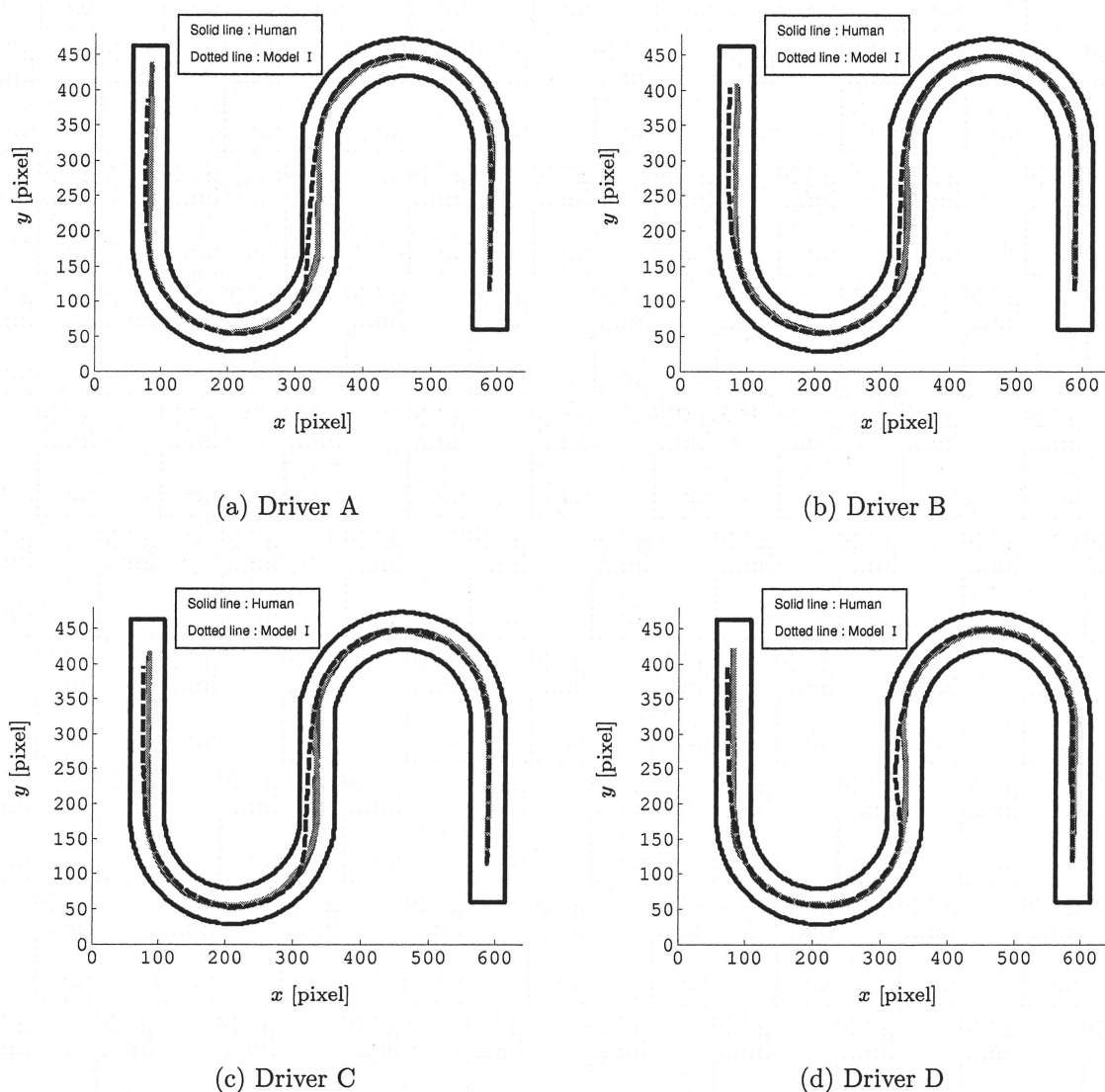


Fig. 4.10 Model Iによる走行軌道

比較ができない。(2)について、前任者の手法では評価者がその見た目より軌道の類似性を主観的に評価しており、評価者が異なった場合でも同じ評価となるという客観性がない。したがって、何か定量的にモデルの表現性を評価できる方法が望まれる。(3)について、モデルの表現性を評価するために、そのモデルを用いた自動走行の実機実験結果を取得する必要がある、さらに比較する認識情報数が多くなればなるほど必要とする実験回数も増加し、煩雑な評価法といえる。(4)について、前任者の手法では軌道が似ていればモデルの表現性は高い、軌道が異なればモデルの表現性は低いというように、人とモデルの走行軌道の類似性をもってモデルの表現性を評価していた。しか

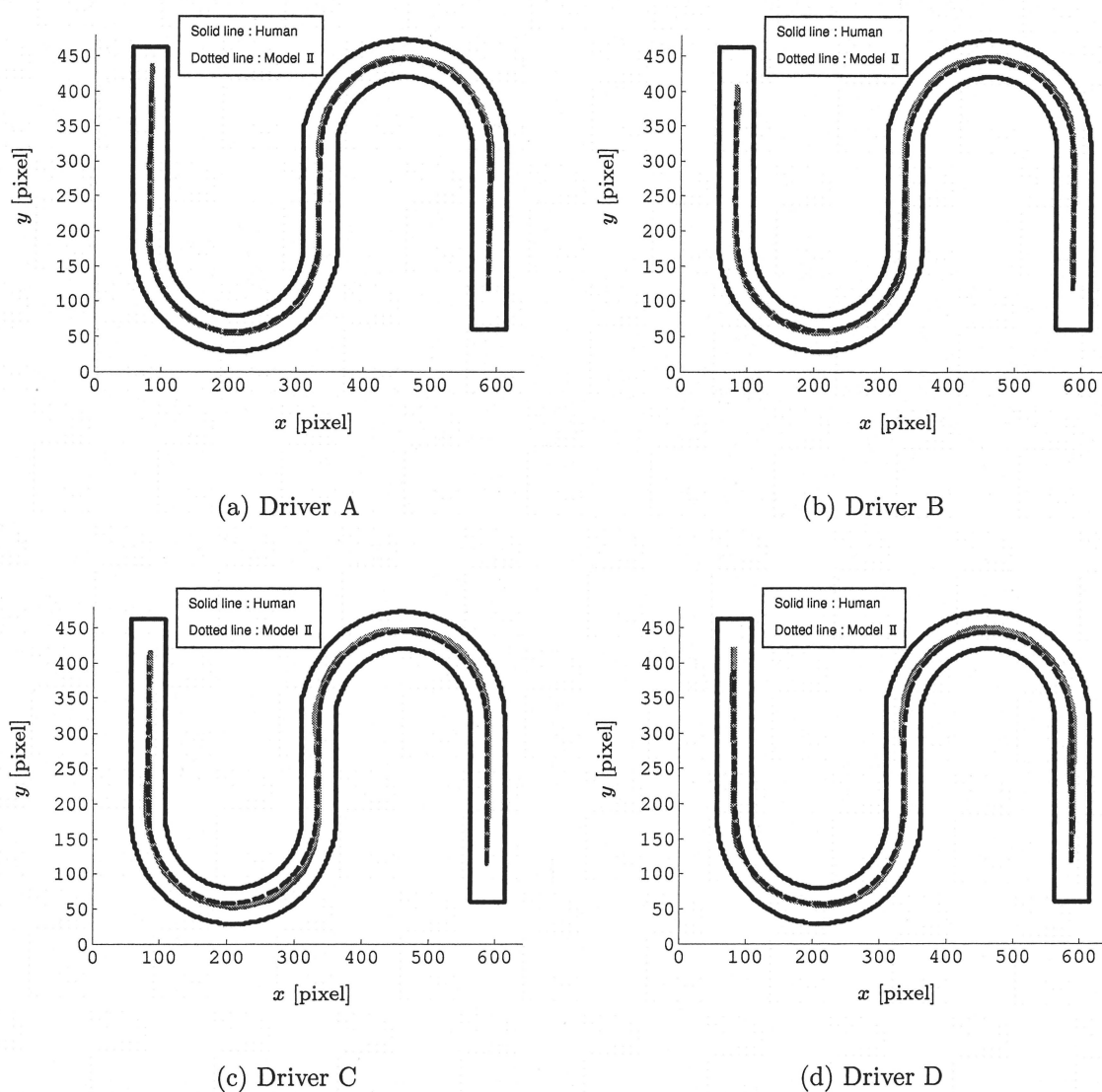


Fig. 4.11 Model IIによる走行軌道

しながら、走行軌道はロボットに操作量を加えた結果生じる軌跡であるため、走行軌道の類似性の評価では直接的にモデルの表現性を評価できていない。したがって、モデルの人の操作の表現性を評価するためには、人の操作量（出力）とモデルの操作量（出力）がどれだけ近いかを考えなければならない。

次章にて、以上の問題点を解決した人の認識情報の重要性を評価する手法を提案する。(1) に対しては異なる認識情報を一つのモデルとして同定し、(2)、(3)、(4) に対しては、モデルの表現性ではなく各認識情報のモデルへの寄与度から人の認識情報の重要性を定量的に評価することでそれぞれ解決している。

第5章 寄与度を用いた人の認識情報の重要性の評価

本章では、前章で述べた前任者による認識情報の重要性の評価方法の問題点を解決するため各認識情報のモデルへの寄与度を定義する．そして求められた寄与度の比を考えることで、人の認識情報間の重要性の順位付けを行う．実際に d_1 , d_2 , ε の三つの認識情報に対して寄与度を計算し、重要性の評価を行った．さらに、得られた順位付けの結果を利用して操作モデルの低次元化を検証した．5.1 節では寄与度を用いた認識情報の重要性の評価方法についての説明、5.2 節では実際に d_1 , d_2 , ε の三つの認識情報の重要性を評価、5.3 節では5.2 節の結果を利用したモデルの低次元化の手順を説明、5.3 節では実際に d_1 , d_2 , ε の3次元のモデルに対して低次元化できるかを検証した．

5.1 寄与度を用いた評価法

本節では本研究で提案する寄与度を用いた人の認識情報の重要性の評価法とどのようにして前任者の問題点を解決したかについて述べる．

Fig. 5.1 に寄与度を用いた人の認識情報の重要性の評価法の手順を示す．まず、比較したい認識情報を入力ベクトルの要素として入出力データを取得し、PWARX モデルを同定する．たとえば n 個の認識情報 $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]'$ を考えた場合、PWARX モデルは以下のように得られる．

$$y_t = \theta_{0,i} + \theta_{1,i} x_{1,t} \dots + \theta_{n,i} x_{n,t}, \quad \text{if } \mathbf{x}_t \in \mathcal{X}_i \quad (i = 1, \dots, s), \quad (5.1)$$

ここで (5.1) 式より出力 y_t は定数項 $\theta_{0,i}$ と n 個の各パラメータベクトルの要素と人の認識情報の積の項の和で表されることが分かる．これら一つ一つの項が実際の操作量(出力)に対してどれだけ寄与しているかを計算することで、重要性を定量的に評価で

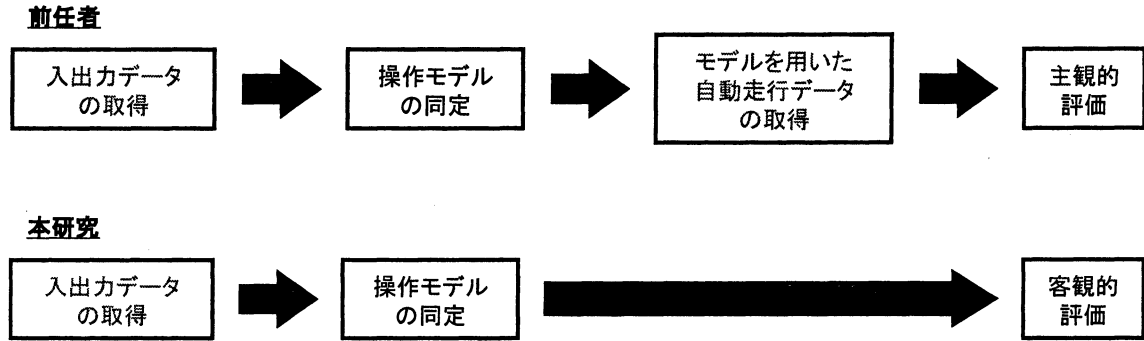


Fig. 5.1 本研究の評価手順

きるのではないかと考え、寄与度を以下の式として定義する.

$$e_1 = \sum_{t=1}^N \|(y_t - \theta_{0,i}) - \theta_{1,i} x_{1,t}\|^2 \quad (5.2)$$

$$e_2 = \sum_{t=1}^N \|(y_t - \theta_{0,i}) - \theta_{2,i} x_{2,t}\|^2 \quad (5.3)$$

$$\vdots$$

$$e_n = \sum_{t=1}^N \|(y_t - \theta_{0,i}) - \theta_{n,i} x_{n,t}\|^2 \quad (5.4)$$

e_1, e_2, \dots, e_n はそれぞれ認識情報 x_1, x_2, \dots, x_n の寄与度を表しており, (x_t, y_t) は人の操作から取得された N 個の同定用データである. 寄与度は認識情報と実際の出力 y_t との二乗誤差の総和で計算され, この値が 0 に近いほど実際の出力に対して認識情報の項が支配的であり寄与度が高く, 誤差が大きければ大きいほど実際の出力に対して認識情報の項が支配的でなくなり寄与度が低い. そして, このようにして得られた n 個の寄与度の比を考えることで人の認識情報の重要性の順位付けができる. 寄与度の値が小さいほど重要な認識情報であるため, 比の値が小さい順から重要度が高い.

4.4 節の問題点 (1) に対して, 比較したい認識情報を入力ベクトルとして一つのモデルしか同定していないためモデルの次数や SVM の結果の違いなどのモデルの違いによる影響が存在せず, 認識情報を純粋に比較することができる. 問題点 (2) に対して, 寄与度という定量的な評価式を導入することで主観的ではなく客観的に評価できるようになった. 問題点 (3) に対して, 寄与度を計算するには同定された操作モデルと同定に用いた入出力データを用いるので Fig. 5.1 の下段に見られるように実験回数が減り, 評価方法の煩雑さを解決することができた. 問題点 (4) に対して, 寄与度を計算

することで人の表現性を評価する必要がなくなった．このように前任者の四つの問題点を解決した評価法となっている．

次節にて入力ベクトルを $\mathbf{x} = [d_1 \ d_2 \ \varepsilon]'$ とし, d_1, d_2, ε の三つの認識情報の重要性を実際に評価する．

5.2 寄与度を用いた評価の例

本節では, d_1, d_2, ε の三つの認識情報の重要性を評価する．取得データは 4.2.1 と同じものを使い, Driver A に対してのみ評価する．入力ベクトルを $\mathbf{x} = [d_1 \ d_2 \ \varepsilon]'$ とすると次のように 3 次元の PWARX モデルのパラメータベクトルと識別関数パラメータが得られる．

$$\boldsymbol{\theta}_1 = [-8.490 \times 10^{-5} \quad -0.014 \quad -0.008 \quad 0.842]'$$
 (5.5)

$$\boldsymbol{\theta}_2 = [-0.005 \quad -1.716 \times 10^{-5} \quad -0.045 \quad 0.139]'$$
 (5.6)

$$\boldsymbol{\theta}_3 = [5.800 \times 10^{-4} \quad -6.367 \times 10^{-5} \quad 3.837 \times 10^{-4} \quad -0.027]'$$
 (5.7)

$$\mathbf{H}_1 = [0.057 \quad -0.191 \quad -0.462 \quad 0.281]'$$
 (5.8)

$$\mathbf{H}_2 = [-0.351 \quad 0.065 \quad 0.456 \quad 10.055]'$$
 (5.9)

それぞれのモードの領域は

$$\mathcal{X}_1 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} > 0 \wedge \mathbf{H}_2' \boldsymbol{\varphi} < 0\}$$
 (5.10)

$$\mathcal{X}_2 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} \leq 0 \wedge \mathbf{H}_2' \boldsymbol{\varphi} \leq 0\}$$
 (5.11)

$$\mathcal{X}_3 = \{\mathbf{x} : \mathbf{H}_2' \boldsymbol{\varphi} > 0\}$$
 (5.12)

と決定され, PWARX モデルが得られた．このモデルに対して寄与度を計算する． e_1, e_2, e_3 はそれぞれ認識情報 d_1, d_2, ε の寄与度である．その結果, $e_1 = 2.35 \times 10^3$, $e_2 = 4.94 \times 10^3$, $e_3 = 2.03 \times 10^3$ と求められた．これより寄与度の比を求めると

$$e_1 : e_2 : e_3 = 0.48 : 1 : 0.41$$

となる．したがって, ε, d_1, d_2 の順に寄与度が高い認識情報であることが分かった．

5.3 モデル次数低減化への応用

前節で d_1, d_2, ε の三つの認識情報の重要性を評価するため入力ベクトルを $\mathbf{x} = [d_1 \ d_2 \ \varepsilon]'$ としてモデル化し、そのモデルを用いて寄与度を計算することで ε, d_1, d_2 の順に寄与度が高いことが分かった。本節では認識情報の重要度の順序を利用して操作モデルの低次元化に応用する。重要度の低い情報は操作量に対して重要なパラメータでないということから、その情報をモデルから除去してもモデルの妥当性が変わらないのであれば低次元化できると考えられる。モデルの低次元化のメリットとして、操作モデルを用いて人の操作の解析を行う場合、操作モデルの次元が低いほどモデルの理解が容易となることが挙げられる。

モデルの妥当性の評価にはシステム同定の分野で用いられるクロスバリデーションと損失関数を用いた手法を用いる。クロスバリデーションとはシステム同定の分野においてモデルの構造を決定するために用いられており、取得した入出力データを同定用と検証用との2種類に分け、モデルの妥当性を調べる方法である。同定用データから PWARX モデルを同定し、検証用データからモデルの妥当性を以下の式より定量的に評価する。

$$V = \sum_{t=1}^N \|\mathbf{y}_t - \mathbf{y}_t^{\text{model}}\|^2 \quad (5.13)$$

$$\mathbf{y}_t^{\text{model}} = \theta_{0,i} + \theta_{1,i} x_{1,t} \dots + \theta_{n,i} x_{n,t} \quad \text{if } \mathbf{x}_t \in \mathcal{X}_i \ (i = 1, 2, 3) \quad (5.14)$$

V を損失関数といい、実際の入力から計算されるモデル出力 $\mathbf{y}_t^{\text{model}}$ と実際の出力 \mathbf{y}_t の間の二乗誤差の総和である。 $\mathbf{y}_t, \mathbf{x}_t$ は実際の入出力データであり、検証用データを用いる。このようにして実際の人の出力とモデルの出力が近いかを計算することでモデルの妥当性を定量的に評価することができる。

次節において、前節で得られた重要性の順序結果を用いて実際にモデルの次数低減化をできるのか検証する。

5.4 モデル次数低減化の検証

5.2 節の結果より最も重要度の低い情報は d_2 であるためこの情報を除去した d_1 と ε の2次元の操作モデル $\text{Model}_{2\text{nd}}$ を同定する。さらに、2番目に重要度の低い情報 d_1 を除去した ε のみの1次元のモデル $\text{Model}_{1\text{st}}$ も同定する。 $\text{Model}_{1\text{st}}$ は4.2.3において既に

得られているため、ここでは省略する。5.2節で得られた3次元のモデルはModel_{3rd}とする。

入力ベクトルを $\mathbf{x} = [d_1 \ \varepsilon]'$ とし、Model_{2nd} のパラメータベクトルと識別関数パラメータは次のように得られた。

$$\boldsymbol{\theta}_1 = [-1.912 \times 10^{-4} \ -0.030 \ 0.165]'$$
 (5.15)

$$\boldsymbol{\theta}_2 = [0.010 \ -0.037 \ -0.517]'$$
 (5.16)

$$\boldsymbol{\theta}_3 = [0.002 \ 0.004 \ -0.089]'$$
 (5.17)

$$\mathbf{H}_1 = [0.091 \ -1.257 \ -14.593]'$$
 (5.18)

$$\mathbf{H}_2 = [-0.083 \ 3.187 \ 2.381]'$$
 (5.19)

それぞれのモードの領域は

$$\mathcal{X}_1 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} > 0 \ \wedge \ \mathbf{H}_2' \boldsymbol{\varphi} < 0\}$$
 (5.20)

$$\mathcal{X}_2 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} \leq 0 \ \wedge \ \mathbf{H}_2' \boldsymbol{\varphi} \leq 0\}$$
 (5.21)

$$\mathcal{X}_3 = \{\mathbf{x} : \mathbf{H}_1' \boldsymbol{\varphi} < 0 \ \wedge \ \mathbf{H}_2' \boldsymbol{\varphi} > 0\}$$
 (5.22)

と決定され、Model_{2nd} が得られた。

Model_{1st}, Model_{2nd}, Model_{3rd} のモデルの妥当性を評価する。それぞれの損失関数を計算した結果、 $V_1 = 30$, $V_2 = 33$, $V_3 = 41$ となった。 V_1 はModel_{1st}, V_2 はModel_{2nd}, V_3 はModel_{3rd} の損失関数を表しており、モデルの次数が高いほど妥当性がよい結果が得られた。システム同定の分野では、入出力データに対するフィッティングはモデル次数の高い方が良いことが知られているため、Model_{3rd} から順に損失関数の値が小さく妥当性が良いのは尤もらしい結果と言える。そのため、モデルの低次元化はモデルの妥当性とモデルの次数のトレードオフを考えながら行わなければならない。

モデルを低次元化できるのは低次元化を行った前後でモデルの妥当性がほぼ変わらない、または妥当性が悪くなったとしても大きな影響がないときである。今回得られた結果では、Model_{3rd} からModel_{2nd} への低次元化を考えたとき、損失関数は3大きくなり、Model_{2nd} からModel_{1st} へは8大きくなる。したがって、この「3」や「8」といった損失関数の変化を妥当性がほぼ変わらない、または大きな影響がないと言ってよいかわからないため、ここでの結論としては低次元化できる可能性あるとしか言えない。もし、「3」の変化は損失関数の変化を妥当性がほぼ変わらない、または大きな影響がないというのであれば、Model_{3rd} はModel_{2nd} まで低次元化でき、逆に、「3」の変化

は損失関数の変化をモデルの妥当性が変わった, または大きな影響があったというのであれば, Model_{3rd} は低次元化できない.

ここで明らかとなったことは, d_2 を除去したときは損失関数が「3」, d_1 を除去したときは損失関数が「8」大きくなったことである. つまり, 重要度の低い情報 d_2 を除去してもモデルの妥当性の変化は小さく, d_2 よりも重要度の高い情報 d_1 を除去するとモデルの妥当性は d_2 のときよりも大きく変化するということである. 重要度の高い認識情報である d_1 は d_2 よりも操作モデルに対して支配的なパラメータであることが確認でき, 寄与度によって正しく順位付けされていることの裏づけとなった.

第6章 結論

6.1 まとめ

本研究では人の車の運転時のステアリング操作など HDS として同定し，得られたモデルを用いて人の操作の解析やアシストシステムへの応用を考えており，特に車のレーンキープ時のステアリング操作を想定し，人が一定速度で直進している二輪駆動型ロボットをコースに沿って走行させたときのステアリング操作に着目した．前任者の研究では，より人間に近い操作モデルを得るためこのとき人がどの認識情報を見ながら操作しているかを得られた操作モデルを用いて評価していたが，「比較したい認識情報ごとに異なるモデルを用いているため認識情報間の重要性を純粋に比較できない」，「モデルの表現性の評価が主観的である」，「モデルによる自動走行結果が必要であり煩雑である」，「人の操作の表現性を評価するのに人とモデルの出力を考えていない」といった問題点があった．本稿ではこれに対して寄与度を用いた認識情報の重要性の評価法を提案し，以上の問題点を解決した．さらに，認識情報間の順位付けを行い，その結果を利用した操作モデルの低次元化を検証した．その結果，本稿においてはモデルの低次元化できる可能性があることが分かった．

6.2 今後の課題

今後の課題として，考慮すべき認識情報を増やし評価していくことである．本稿においては先行研究で用いられていた d_1 , d_2 , ε の三つの情報のみを考えていた．しかしながら，Driver A から Driver D の 4 名に対してロボットの操作中にどこを見ていたかアンケートをとった結果，ロボット左右真横方向の壁までの距離の比やロボットの進行方向と経路角度の差が挙げられた．これらの情報を含め，実際に人が認識している情報を今回提案した評価法を用いて調査していきたい．また，本稿において d_1 , d_2 , ε のモデルの次数低減化を検証した結果，次数を減らせるかもしれないという可能性を

示すのみとなったが、これはこれらの認識情報間の重要性があまり変わらないためであると考えられる。したがって、故意に人が認識していないような情報を含めてモデル化し、その情報の寄与度が悪く、その情報を除去してモデルの妥当性もほぼ変わらない結果が得られれば、寄与度を用いてモデルの低次元化ができることが確認できる。

参考文献

- [1] T. Pilutti and A. G. Ulsoy, "Identification of Driver State for Lane-Keeping Tasks," *IEEE Trans. Syst. Man and Cyber.*, vol. 29, no. 5, pp. 486–502, 1999.
- [2] K. S. Narendra and K. Pathasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [3] D. D. Vecchio R. M. Murray and P. Perona, "Decomposition of human motion into dynamics-based primitives with application to drawing tasks," *Automatica*, vol. 39, no. 12, pp. 2085–2098, 2003.
- [4] J. H. Kim, S. Hayakawa, T. Suzuki, K. Hayashi, S. Okuma, N. Tsutida, M. Shimizu and S. Kido, "Modeling of Human Driving Behavior based on Expression as Hybrid Dynamical System," *Trans. of SICE*, vol. 40, No. 2, pp. 180–188, 2004.
- [5] 鈴木達也, "ハイブリッドシステムモデルに基づく行動情報処理," 計測と制御, vol. 44, No. 7, pp. 452–457, 2005.
- [6] 鈴木達也, "ハイブリッドシステムと高次モーション," 電気学会論文誌 D, vol. 128, No. 6, pp. 819–825, 2008.
- [7] *Hybrid systems* I, II, III, IV, V, Lecture Notes in Computer Science, 736, 999, 1066, 1273, 1567, New York, Springer-Verlag, 1993 to 1998.
- [8] P. J. Antsaklis and A. Nerode, "Hybrid control systems: an introductory discussion to the special issue," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 457–460, 1998.

- [9] M. S. Branicky, V. S. Borkar and S. K. Mitter, "A unified framework for hybrid controll: Model and optimal control theory," *IEEE Trans. Automat. Contr.*, vol. 43, no. 1, pp. 31–45, 1998.
- [10] W. P. M. H. Heemels, B. D. Schutter and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.
- [11] M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 555–559, 1998.
- [12] A. Bemporad and M. Morarri, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [13] J. Lygeros, C. Tomlin and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.
- [14] A. Chutinan and B. H. Krogh, "Computational techniques for hybrid system verification," *IEEE Trans. Automat. Contr.*, vol. 48, no. 1, pp. 64–75, 2003.
- [15] A. Bemporad, D. Mignoe and M. Morarri, "Moving horizon estimation for hybrid systems and fault detection," *Proc. American control conference 1999*, pp. 2471–2475, 1999.
- [16] W. P. M. H. Heemels, J. M. Schumacher and S. Weiland, "Linear complementarity systems," *SIAM J. Appl. Math.*, vol. 60, no. 4, pp. 1234–1269, 2000.
- [17] E. D. Sontag, "Nonlinear regulation: the piecewise linear approach," *IEEE Trans. Automat. Contr.*, vol. 26, no. 2, pp. 346–357, 1981.
- [18] G. F. Trecate, M. Muselli, D. Liberati and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.
- [19] J. Roll, A. Bemporad, and L. Ljung, "Identification of piecewise affine system via mixed-integer programming," *Automatica*, vol. 40, no. 1, pp. 37–50, 2004.

- [20] A. Bemporad, A. Garulli, S. Paoletti and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Trans. Automat. Contr.*, vol. 50, no. 10, pp. 1567–1580, 2005.
- [21] A. L. Juloski, S. Weiland and W. P. M. H. Heemels, "A Bayesian approach to identification of hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 50, no. 10, pp. 1520–1533, 2005.
- [22] R. Vidal, S. Soatto Y. Ma and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," *Proc. of 42nd IEEE Conf. Decis. Contr.*, pp. 167–172, 2003.
- [23] A. L. Juloski, W. P. M. H. Heemels, G. F. Trecate, R. Vidal, S. Paoletti and J. H. G. Niessen, "Comparison of Four Procedures for the Identification of Hybrid Systems," *HSCC 2005*, LNCS 3414, pp. 354–369, 2005.
- [24] S. Ma, T. Zanma and M. Ishida, "Identification of human skill and its application to an automatic driving system –an approach from hybrid dynamical system," *Proc. IEEE Industrial Electronics, IECON 2006*, pp. 400–405, 2006.
- [25] S. Ma, T. Zanma and M. Ishida: "Automatic driving system using identification of switched systems with unknown switch points," *IEEJ Trans. on Electrical and Electronic Engineering*, vol. 1, no. 4, pp. 426–437, 2006.
- [26] H. Sato, T. Zanma and M. Ishida, "Identification of human operation using data clustering and its application to automated system," *Proc. of the 10th IEEE International Workshop on Advanced Motion Control*, pp. 411–416, 2008.
- [27] 坂東宏記, 残間忠直, 石田宗秋, "データクラスタリング手法を用いた操作データの解析に関する研究," 三重大学卒業論文, 2011.
- [28] M. Land, and J. Horwood, "Which parts of the road guide steering," *Nature*, Vol 377 · 28, pp. 339–340, 1995.

関連論文および口頭発表

- (1) 牧田克紀, 残間忠直, 弓場井一裕, 石田宗秋 : “クラスタリング手法を用いた操作データの解析に関する研究,” 平成 23 年度東海支部連合大会, CD-ROM M3-8, 2011.
- (2) K. Makida, H. Bando, M. Ishida, T. Zanma : “Identification of human driving operation of a two-wheeled mobile robot using the clustering technique,” *Proc. of International Symposium for Sustainability by Engineering at MIU*, pp.132-135, 2011.

謝 辞

本研究の遂行ならびに本論文の作成に際し，終始並々ならぬ御指導と御鞭撻を賜りました三重大学大学院工学研究科電気電子工学専攻教授 石田宗秋先生ならびに千葉大学大学院工学研究科人工システム科学専攻電気電子系コース准教授 残間忠直先生に心より御礼申し上げます．大学院2年生からの1年間，残間先生に代わり数々の御指導，御助言を賜りました三重大学大学院工学研究科電気電子工学専攻准教授 弓場井一裕先生にも心より御礼申し上げます．また検討会において助言を賜りました三重大学大学院工学研究科電気電子工学専攻教授 平井淳之先生ならびに三重大学大学院工学研究科電気電子工学専攻准教授 山村直紀先生に心より御礼申し上げます．実機環境構築の際に数々の御援助を賜りました三重大学工学部電気電子工学 技官 廣田栄毅氏に心より御礼申し上げます．

本研究の走行データの取得のため長時間御手伝い頂きました Driver B, Driver D の三重大学大学院工学研究科電気電子工学専攻博士前期課程萩野道久氏，劉鋒氏，Driver C の三重大学工学部電気電子工学科渡邊涼氏，本稿では記載できませんでしたが，Driver E, Driver F の三重大学大学院工学研究科電気電子工学専攻博士前期課程山口隆志氏，迫間泰樹氏に心より感謝します．

以降は人名を略させていただきますが，本研究の進行に際し，数々の御助言を頂きました平成 20, 21 年度石田研究室卒業の先輩方に心より感謝します．また私と同時期に制御システム研究室およびエネルギーシステム研究室に在籍し共に学生生活を過ごした皆様には大変お世話になりました．

最後に，博士前期課程に進学し，4年間何不自由なく勉学，研究に励めたのは家族の理解と協力のお陰であります．末筆ながら，家族，とりわけ父母に対する深い感謝の念をもって本論文を締めくくりたいと思います．