

はじめに

自動車産業などでモデルベースによる開発

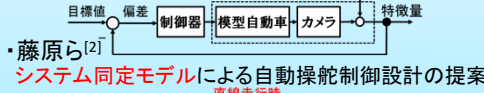
・モデルベース開発
コンピュータ上に仮想的なモデルを作成し、検証開発を行う

・モデル設計の手法

- 第一原理による手法
-物理原理に基づき制御対象の振る舞いを表現する簡易的な数式
モデル→運動方程式、回路方程式、etc..
モデル表現には必ずモデルで表せない特性を含む
- システム同定理論による手法
-制御対象へ加えるテスト信号のデータ列とその出力データ列より
設計するモデル→状態方程式、伝達関数
第一原理では表現できない制御環境を考慮したモデル構築が可能

カメラ付き模型自動車の自動操舵制御設計

・小森ら[1]
第一原理モデルによる自動操舵制御設計の提案
コース変動による外乱やダイナミクスの影響

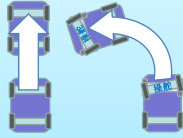


・藤原ら[2]
システム同定モデルによる自動操舵制御設計の提案

システム同定によりコース変動による外乱やダイナミクスの影響を考慮したモデルの設計

・研究目的
ビジュアルフィードバックを用いた各モデル設計
直線走行データからモデル構築
カーブ走行データからモデル構築

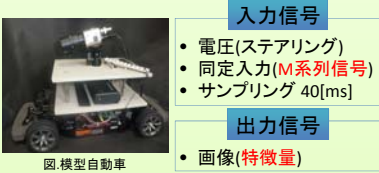
走行経路を考慮した
制御器の設計が可能



・検証方法
直線走行モデルに基づく制御器と各モデルに基づく
制御器のシミュレーションおよび実機により性能比較

システム構成

実験装置	定格
コンピュータ	CPU:Pentium4 3GHz メモリ:1GB
DA変換器	DS1104(dSPACE社製)
信号変換器	制御入力電圧(0-10V) 走行電圧(0-4.5[V])
送信機	周波数:27.145MHz 電圧:12V
ラジコンカー	TAMIYA TT-1



直接法-出力インターサンプリング

□ 入力のサンプリング周期
 $T = p\Delta = 2 \times 0.02$

□ 出力のサンプリング周期
 $\Delta = T/p = 0.04/2$

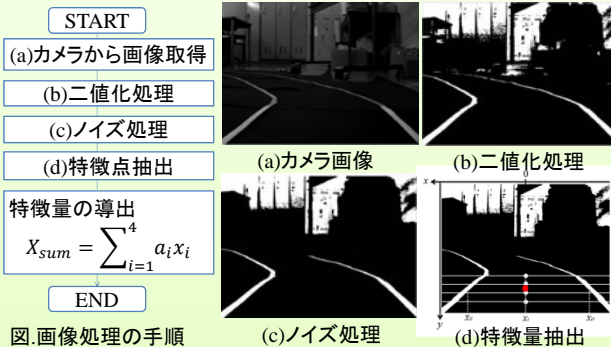
$u(kT)$ と $d(k\Delta)$ は無相関

間接法-two-stage法

□ 2段階によるモデル構築
($r = r_1 + Cr_2$), (d) から (u, y) の伝達関数
 $\begin{cases} u(kT) = Sr(kT) - SCHd(kT) \\ y(kT) = GSr(kT) + SHd(kT) \end{cases}$

$Sr(kT)$ と $d(kT)$ は無相関

・道路形状の特微量をカメラ画像上から導出



直接法の入出力データ

□ PI(Past Input)-MOESP
- 入力と入力雑音が無相関時に有効

□ PO(Past Output)-MOESP
- 入力雑音と出力雑音ありでも有効

間接法の入出力データ

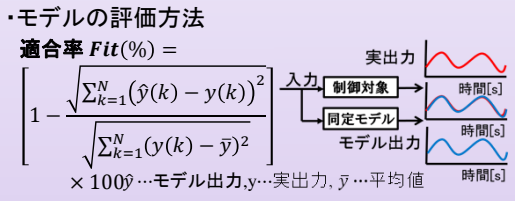
□ CL(Closed Loop)-MOESP
- 閉ループ同定の有効な手法として提案

MOESP=Multivariable Output Error State sPace

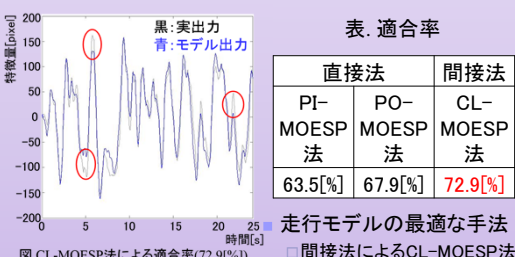
直線走行モデルの設計

・モデル次数の決定
運動特性による次数決定
進行方向の運動 + ヨ一回転の運動 + 特微量の動き

モデルの次数: 3次



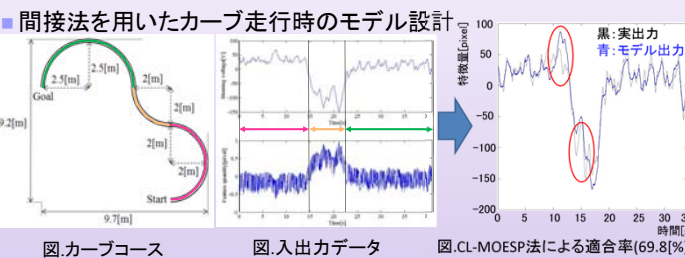
モデルを適合率により評価



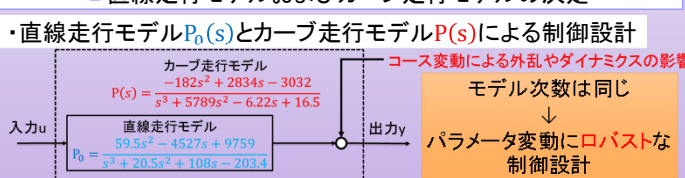
走行モデルの最適な手法
□ 間接法によるCL-MOESP法

カーブ走行モデルは
間接法を用いたCL-MOESP法によりモデル設計

カーブ走行モデルの設計



直線走行モデルおよびカーブ走行モデルの決定



制御器設計

・モデル変動を考慮した制御設計
□ 変動をフィードバック型と定義

P:カーブ走行モデル
P0:直線走行モデル

外乱 → モデル化誤差 Δ → 出力 y

■ 走行経路にロバストな制御設計
□ モデル化誤差 Δ を低減する制御器を H_∞ 制御理論により設計

H_∞ 制御理論
■ 入力と出力外乱を低減するために混合感度関数問題により定式化

$$\|W_S S\|_{\infty} < \gamma$$

■ 周波数重みの選定
モデル化誤差 Δ(s) = (P(s) - P0(s)) / P(s)

周波数重み W_S(s) = (1.5s + 2.5) / (s + 4)

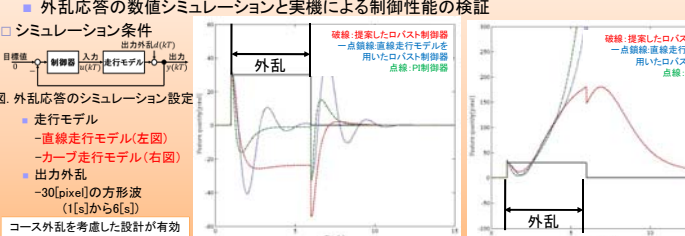
■ 設計した制御器 K(s)

$$K(s) = \frac{1.72 \times 10^7 s^4 + 8.55 \times 10^8 s^3 + 1.34 \times 10^{10} s^2 + 7.78 \times 10^{10} s + 7.02 \times 10^{10}}{s^5 + 2.88 \times 10^8 s^4 + 7.94 \times 10^8 s^3 + 3.10 \times 10^8 s^2 + 3.28 \times 10^{10} s + 2.97 \times 10^{10}}$$

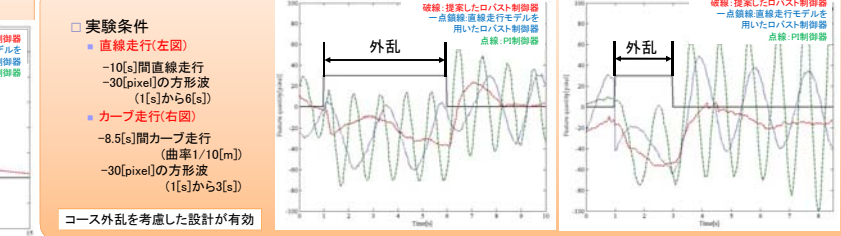
■ 比較用の直線走行モデルを用いたPI制御器設計
 $K_{PI}(s) = \frac{2.3s + 1.7}{s}$

■ 比較用の直線走行モデルを用いたロバスト制御器設計
カーブ走行モデルを直線走行モデルに対して2倍の変動と考える
 $K(s) = \frac{4.67 \times 10^8 s^4 + 2.42 \times 10^7 s^3 + 3.84 \times 10^8 s^2 + 2.21 \times 10^8 s + 1.68 \times 10^9}{s^5 + 6942 s^4 + 49.03 \times 10^3 s^3 + 3.94 \times 10^7 s^2 + 5.57 \times 10^8 s + 5.53 \times 10^7}$

シミュレーション



実機実験



まとめ

■ モデル設計
□ 間接法により直線走行モデルおよびカーブ走行モデルを設計した。
□ 直線走行モデルとカーブ走行モデルは3次のモデルで表現した。

■ 制御設計
□ 各モデルを用いた走行経路にロバストな制御器の設計をした。
□ カーブ走行モデルを考慮して制御設計を行うことは有効であることがわかった。