

非線形システムの同定

茨城大学工学部 楊子江

この資料は、非線形システムの同定に関する解説です。ご自由に使ってください。ご質問、或いは改訂のご要望は yoh@mx.ibaraki.ac.jp にご連絡ください。

2014.3.7

内容要旨

- (1) 非線形システム同定の歴史と現状
歴史を振り返る
- (2) ブラックボックスモデルの同定
各種の基底関数による三層NNを用いたブラックボックスモデルの同定の統一的解釈
- (3) 非線形同定のための最適化手法
構造決定とパラメータ推定のアルゴリズム
- (4) 線形パラメータ時変 (LPV) モデルの同定
形式的には線形であるが、パラメータが状態変数に依存

- (5) NFIRモデルの同定
非線形インパルス応答モデルの同定, ボルテラモデルを含む
- (6) ブロック指向モデルの同定
Hammersteinモデル, Wienerモデル, Hammerstein-Wiener (sandwich) モデル
- (7) 非線形メカニカルシステムのパラメータ推定
簡単な応用事例
- (8) 非線形システムの適応制御
モデルを同定しながら, 制御器をオンライン調整する
- (9) まとめ

(1) 非線形システム同定の歴史と現状

(1) 非線形システム同定の歴史と現状

1960-1980年代まで

非線形状態空間モデルのパラメータと状態変数の同時推定:

相良節夫, 沖田豪: 非線形システムの同定と推定, 計測と制御, 26-9, 793-790 (1987).

拡張カルマンフィルタ
(EKF)

ボルテラモデルの同定:

渡辺敦: ボルテラ級数とシステム同定, 非線形システム論, 計測自動制御学会, 第7章(1993).

非線形インパルス応答
モデル

多項式モデルに対するGMDH(Group Method of Data Handling)手法:

A. G. Ivakhnenko: Polynomial theory of complex systems, IEEE Trans. on Systems, Man and Cybernetics, 1-4, 364-379 (1971).

信号を二つずつ束ねていくという階層型モデル

(1) 非線形システム同定の歴史と現状

ブロック指向モデル:

R. Haber and H. Unbehauen: Structure Identification of nonlinear dynamic systems-A survey on input-output approaches, Automatica, 26-4, 651-677 (1990).

線形モデルの前に非線形ブロック: Hammersteinモデル

線形モデルの後に非線形ブロック: Wienerモデル

線形モデルの前後ともに非線形ブロック: Sandwichモデル

(1) 非線形システム同定の歴史と現状

1990年代

シグモイド型NN及びBP学習アルゴリズムによる非線形システムモデルの同定:

K.S.Narendra and K.Parthasathy: Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks, 1-1, 4/27(1990).

全くのブラックボックスモデルであり、
解釈が困難。

(1) 非線形システム同定の歴史と現状

RBFNによる非線形システムモデルの制御:

R. M. Sanner and J. J. E. Slotine: Gaussian networks for direct adaptive control, IEEE Trans. Neural Network, NN-6, 837/863(1992).

非線形適応制御, スライディングモード制御と結びつけた。
2012年4月までの引用回数1317回



S. S. Ge, C. C. Hang, T. H. Lee and T. Zhang, Stable Adaptive Neural Network Control, Kluwer Academic Publishers(2002).

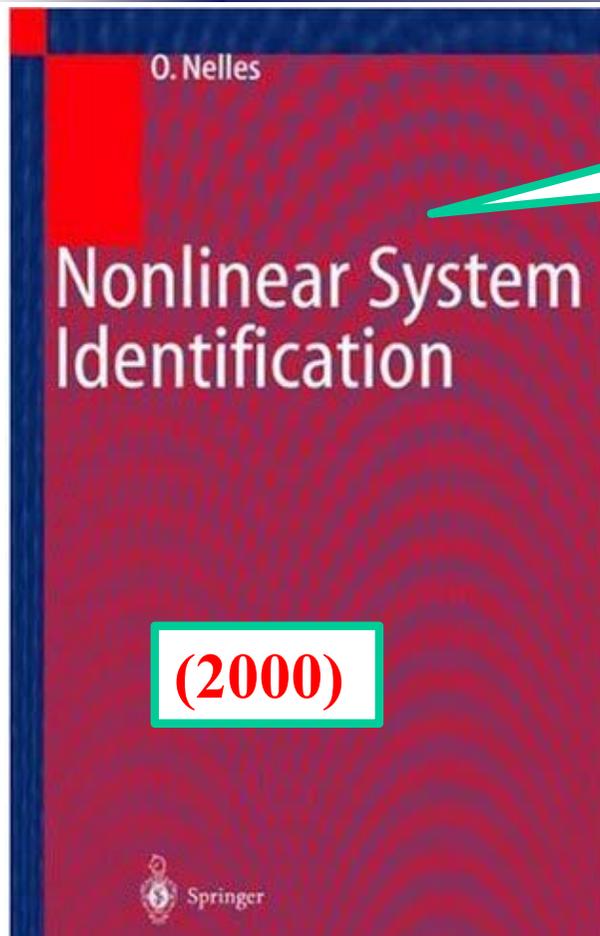
Y. H. Kim and F. L. Lewis: High-Level Feedback Control with Neural Networks, World Scientific (1998).

非線形適応制御の分野では

連続時間非線形状態変数モデルが主流

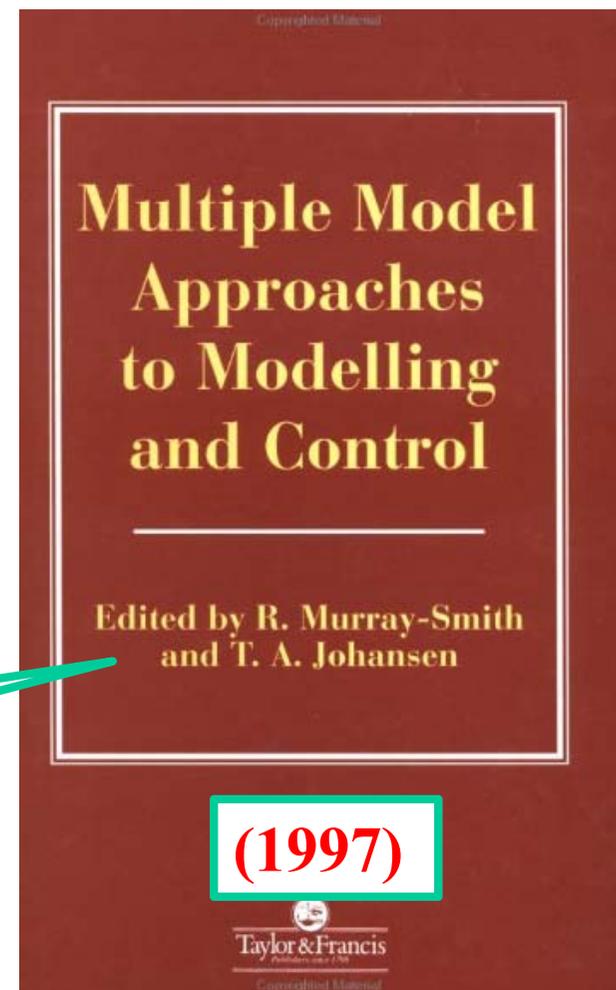
適応制御, 適応非線形オブザーバ,
ハイゲインオブザーバ,
スライディングモードオブザーバなど

(1) 非線形システム同定の歴史と現状



非線形ARX(NARX)モデルの同定全般
自動車エンジン同定の応用例掲載

局所線形モデルによる補間モデル



(1997)

(1) 非線形システム同定の歴史と現状

2000年前後に一段落:

構造決定とパラメータ推定の最適化手法:

遺伝アルゴリズム, 非線形最小二乗法, 直交化最小二乗法,

種々の離散時間NARXモデル:

シグモイドNN, RBFN, 多項式モデル, ファジィモデル, LPVモデル

(1) 非線形システム同定の歴史と現状

2000年以降の注目話題

ハイブリッドシステムモデルの同定:

S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal,
“*Identification of hybrid systems: a tutorial,*” *European Journal of Control*, vol. 513(2-3), pp. 242 – 260, 2007

山本茂, ハイブリッドシステムの制御-VI—同定手法, システム/制御/情報, Vol. 52, No. 3, pp. 103-109, 2008

難しくなってきた!

ハイブリッドシステムって, 何?

例えば: システムの状態によって, 切り替わる(スイッチング)線形モデルの集まり
なめらかな補間モデルではない

PWARX(Piece-Wise affine Auto Regressive eXogenous)

区分的アファインな外部入力つき自己回帰モデル

(1) 非線形システム同定の歴史と現状

Hammersteinモデルの部分空間法による同定

K.S.Narendra and K.Parthasathy: Subspace Identification of Hammerstein Systems Using Least Squares Support Vector Machines, IEEE Transactions on Automatic Control, 50-10, 1509/1519(2005).

部分空間法: 入出力データから状態変数モデルを直接同定する方法

Wienerモデルの部分空間法による同定

R. Raich, R. G. T. Zhou, M. Viberg, Subspace based approaches for Wiener system identification, IEEE Transactions on Automatic Control, 50-10, 1629/1634 (2005).

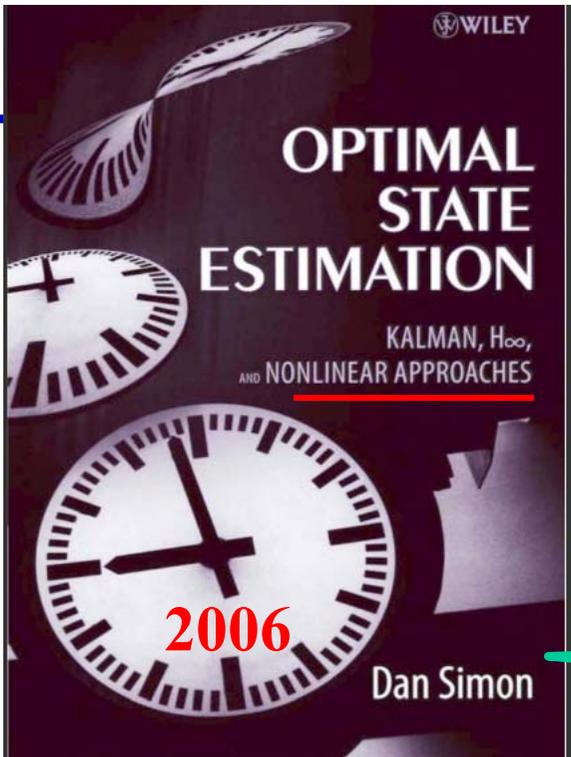
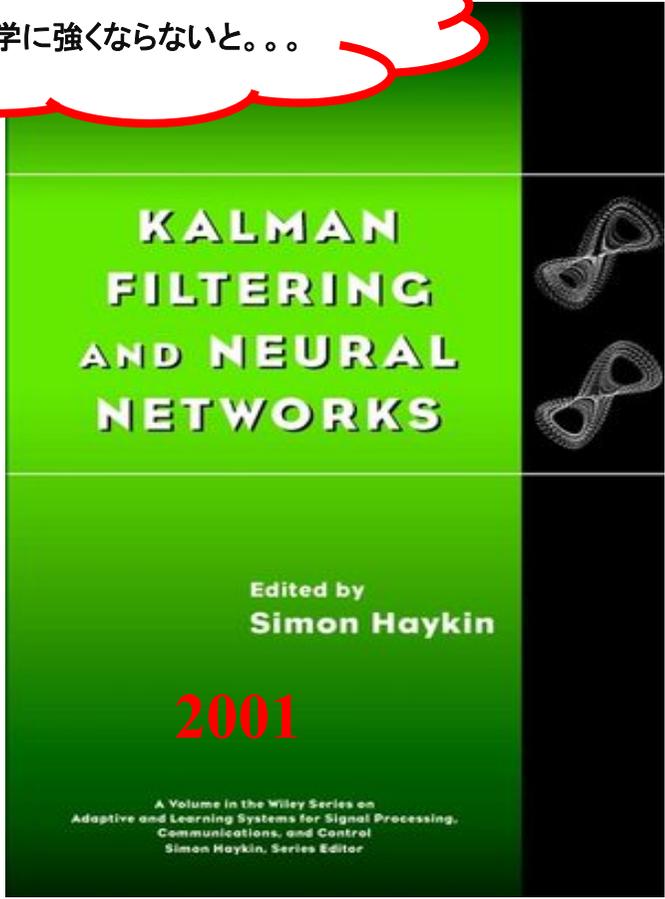
(1) 非線形システム同定の歴史と現状

統計学に強くないと。。。

粒子フィルタ(Partical Filter)
ベイズ推定, 逐次Monte Carlo法

無香料カルマンフィルタ(Unscented Kalman Filter)
Unscented変換による事後確率分布の計算

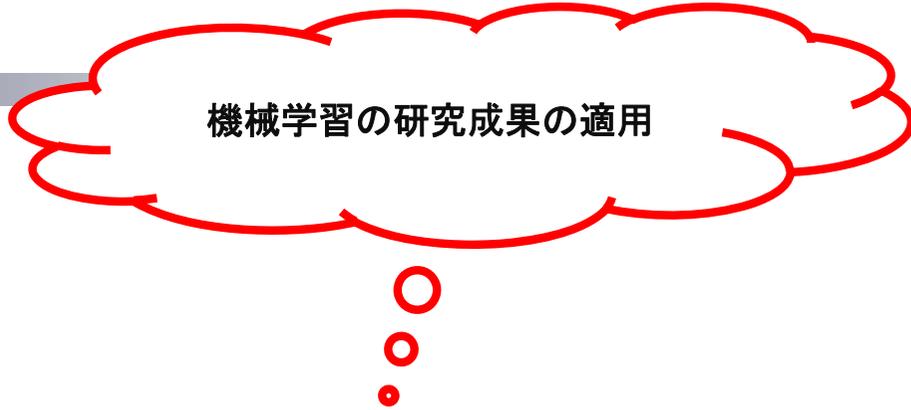
Expectation-Maximization Algorithm



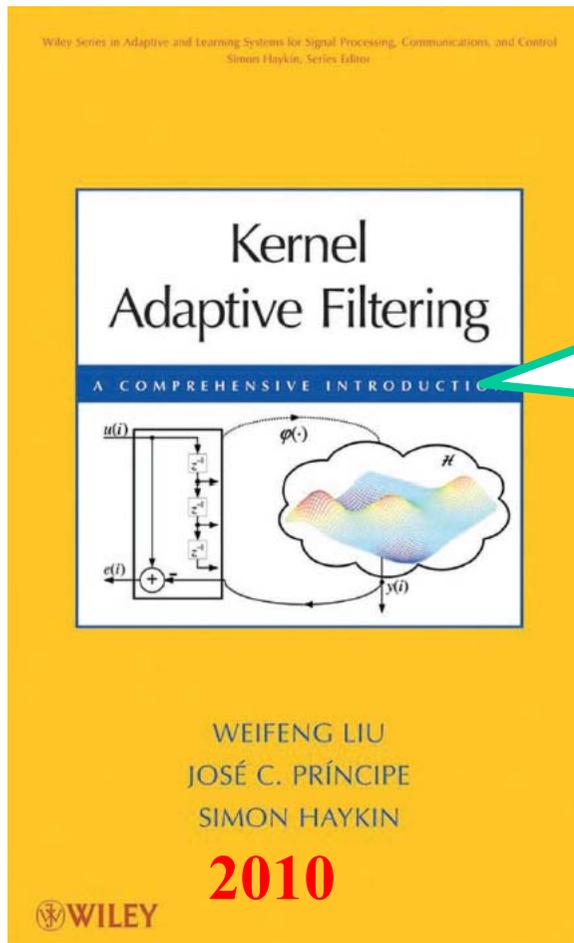
非線形状態変数モデルの状態推定とパラメータ推定
従来の拡張カルマンフィルタ(EKF)よりよさそう

制御関係者にとっては, この本は入門しやすい

(1) 非線形システム同定の歴史と現状



機械学習の研究成果の適用



静的非線形関数を動的な非線形関数へ拡張
従来の線形システムに対する推定手法にカーネル化を行う

井出剛, 矢入健久:
機械学習技術の最近の発展と
システムモデリングへの応用,
計測と制御, Vol. 49, No. 7, 2010

(1) 非線形システム同定の歴史と現状

Gaussian Processes for
Machine Learning



2006

Carl Edward Rasmussen and Christopher K. I. Williams

非線形システムを正規過程
(ガウシアンプロセス)でモデル化;

- ガウシアンプロセスの分散行列はカーネル行列;
- パラメータは事後確率の最大化によって推定
- 少ないパラメータ(ハイパーパラメータ)で済む.
- 分散行列から予測値の信頼区間サイズは学習データに依存計算可能
- 分散行列のサイズはデータ数に依存.

Gregor Gregorcic and Gordon Lightbody: Nonlinear system identification: From multiple-model networks to Gaussian processes, Engineering Applications of Artificial Intelligence 21 (2008) 1035–1055.

(2) ブラックボックスモデルの同定

(2) ブラックボックスモデルの同定

2.1 NARXモデルの導入

2.2 シグモイド型NN

2.3 多項式モデル

2.4 ローカル基底関数ネットワーク

2.5 パラメータ推定と正則化

2.1 NARXモデルの導入

NARX (Nonlinear Auto Regressive eXogenous) モデル

$$y(t) = g_0(\phi(t)) + e(t) \quad \text{—入出力システム}$$

$$\phi(t) = [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]^T$$

$u(t)$: 入力信号 $y(t)$: 出力信号 $e(t)$: ゼロ平均の白色雑音.

g_0 : 未知な連続関数 (ブラックボックス) $\phi(t)$: 回帰ベクトル.

例:

$$\bullet \bullet \bullet \quad y(t) = g_0(\phi(t)) + e(t)$$

ただし,

$$g_0(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5) = \frac{\phi_1 \phi_2 \phi_3 \phi_5 (\phi_3 - 1) + \phi_4}{1 + \phi_2^2 + \phi_3^2}$$

$$\begin{aligned} \phi(t) &= [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] \\ &= [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \end{aligned}$$

2.1 NARXモデルの導入

シミュレーションと予測(用語の統一)

$\hat{y}(t) = \hat{g}(\phi(t), \theta)$	$\hat{y}(t) = \hat{g}(\hat{\phi}(t), \theta)$
<u>一段先予測モデル</u>	<u>シミュレーションモデル</u> 無限段先予測モデル
予測誤差モデル, NARXモデル	出力誤差モデル
直並列モデル	並列モデル
フィードフォワードNN	リカレントNN

$$\hat{\phi}(t) = [\hat{y}(t-1), \dots, \hat{y}(t-n_y), u(t-1), \dots, u(t-n_u)]^T$$

$$\phi(t) = [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]^T$$

2.1 NARXモデルの導入

- シミュレーションモデルの同定は困難.
- 予測モデルに基づいた同定が普通.
- 予測誤差が十分小さくても, シミュレーション誤差が必ずしも小さいとは限らない. 不安定な場合もありうる.
- 予測モデルに基づいた同定の後, 異なる入力データをシミュレーションモデルに入力してモデル検証を行うべき.
- 出力のシミュレーション値がそれに対応するシステムの観測出力に近ければ, よいだろう.

2.1 NARXモデルの導入

NARXモデルの統一表現

三層NNともいう

回帰ベクトル ϕ は第一層
基底関数 $g_k(\phi)$ は第二層
線形結合は第三層

$$\hat{g}(\phi(t), \theta) = \sum_{k=1}^M \alpha_k g_k(\phi)$$

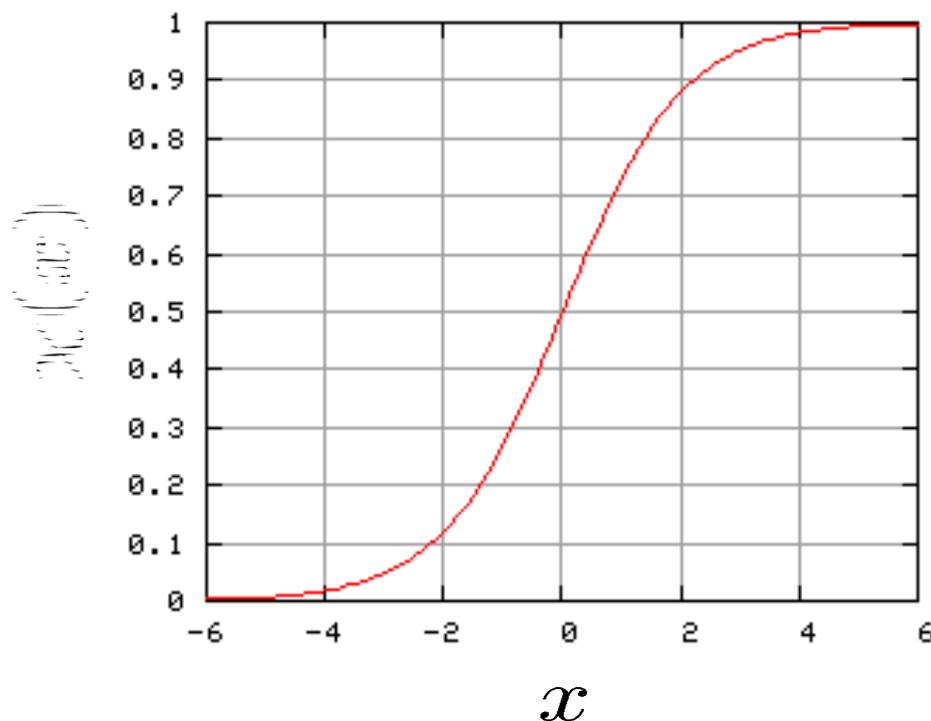
$g_k(\phi)$: 基底関数, $\theta \in R^m$: パラメータベクトル.

基底関数の中にも非線形なパラメータが含まれることがよくある. ここでは, 簡単のため明記していない.

2.2 シグモイド型NN

シグモイド型NN

$$g_k(\phi) = \chi(\mathbf{a}_k^T \phi + b_k), \quad \underline{\mathbf{a}_k} \in R^d, \quad \underline{b_k} \in R$$



非線形パラメータ

標準シグモイド関数

$$\chi(x) = 1/(1 + e^{-x})$$

2.2 シグモイド型NN

- システムの事前情報が十分でなくてもシステムのモデルを精度よく近似できる.
- 勾配法による学習は反復計算回数が多い, 局所的最適解に陥りやすい.
- ネットワークの最適な構造や汎化能力の検討が困難.
- ただのブラックボックス, 物理的解釈が困難で, 制御関係者はそれほど好まない.

2.3 多項式モデル

q 次の Kolmogorov-Gabor 多項式 :

$$\begin{aligned} \hat{g}(\phi(t), \boldsymbol{\theta}) = & \theta_0 + \sum_{i_1=1}^d \theta_{i_1} \phi_{i_1}(t) + \sum_{i_1=1}^d \sum_{i_2=i_1}^d \theta_{i_1 i_2} \phi_{i_1}(t) \phi_{i_2}(t) \\ & + \cdots + \sum_{i_1=1}^d \cdots \sum_{i_q=i_{q-1}}^d \theta_{i_1 \dots i_q} \phi_{i_1}(t) \cdots \phi_{i_q}(t) \end{aligned}$$

$\phi_i(t)$: $\phi(t)$ の i 番目の要素.

単項式の総数

$$M = \sum_{i=0}^q n_i$$

$$n_i = \frac{n_{i-1}(n_y + n_u + i - 1)}{i}, \quad n_0 = 1$$

何千何万個にもなる!

i 次の単項式の数

2.3 多項式モデル

- 必要最小限の項を効率よく選び出すことがポイント.
- パラメータに関して線形であるので, 線形最小二乗法適用可能.
- 非線形プロセスの予測や制御系設計などに使えそう.

E. Hernandez and Y. Arkun, Control of nonlinear systems using polynomial ARMA models, AIChE Journal, 39-3, 446/349 (1993).

J. W. Glass and M. A. Franchek, NARMAX modeling and robust control of internal combustion engines, International Journal of Control, 72-4, 289/304 (1999).

2.3 多項式モデル

例として、以下のシステムについて考える。

$$y(t) = g_0(\phi(t)) + e(t)$$

ただし、

$$g_0(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5) = \frac{\phi_1 \phi_2 \phi_3 \phi_5 (\phi_3 - 1) + \phi_4}{1 + \phi_2^2 + \phi_3^2}$$

$$\begin{aligned}\phi(t) &= [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] \\ &= [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)]\end{aligned}$$

シグモイドNNって、
大変そうだ！

K.S.Narendra and K.Parthasathy: Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks, 1-1, 4/27(1990).

$n_y = 3, n_u = 2$ 既知, 入力層ノード5個, 第1中間層ノード20個, 第2中間層ノード10個, 出力層ノード1個からなる4層NNで, パラメータ総数341個.

2.3 多項式モデル

単項式の選定に遺伝的アルゴリズムを使ってみよう

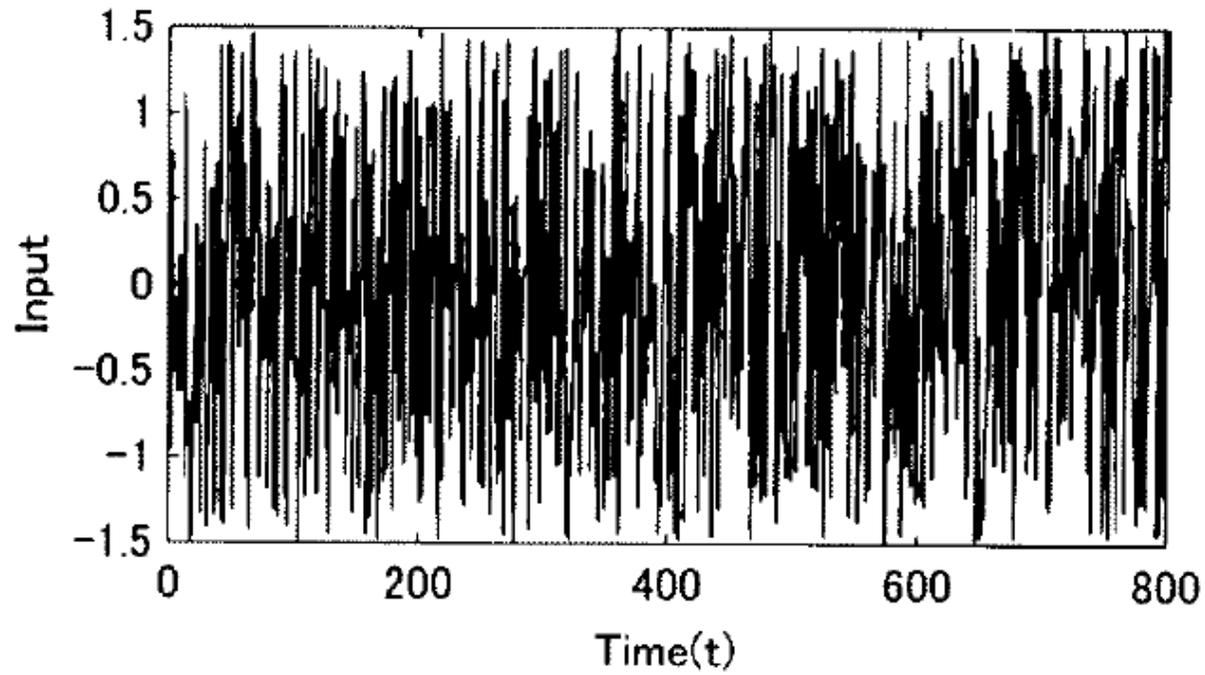
10000001000100000000000100010...

従来法では、染色体の長さは何千何万？
無理！

特殊なコーディング手法なら、大丈夫

楊子江, 藤本, 熊丸: 遺伝的アルゴリズムを用いた非線形多項式モデルの同定, 電気学会論文誌, Vol.120-C, No.8, 1208/1217(2000)

2.3 多項式モデル



確率入力で同定してみよう

2.3 多項式モデル

多項式モデルでは, $n_y = 4$, $n_u = 3$, $q = 8$ と必要以上に設定してしまった場合, 単項式の総数 : 6435.

しかし, 以下の多項式モデルで十分 :

$$\begin{aligned} y(t) = & 0.867108u(t-1) - 0.28615y^2(t-3)u(t-1) \\ & - 0.24383y(t-2)u(t-1)u(t-3) + 0.0886779y(t-2)u^2(t-2) \\ & - 0.424154y(t-1)y(t-2)y(t-3)u(t-2) \end{aligned}$$

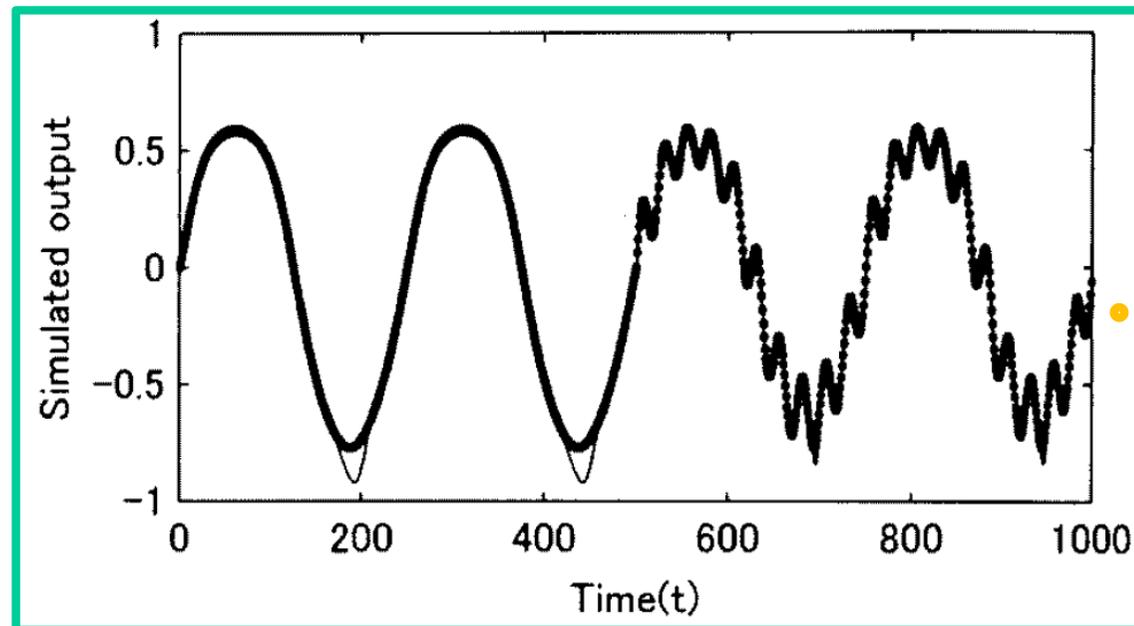
意外と近似能力が強い!

数万個の候補からも大丈夫.

2.3 多項式モデル

別の入力でシミュレーション(予測ではない)してみよう

$$u(t) = \begin{cases} \sin(2\pi t/250), & t \leq 500 \\ 0.8\sin(2\pi t/250) + 0.2\sin(2\pi t/25), & t > 500 \end{cases}$$



よさそう

LBFN (Local Basis Function Network)

基底関数の構成

マザー (mother) 基底関数 : $\chi(x), x \in R$.

LBF : $\chi(\beta^{-1}(x - \gamma))$.

β : dilation パラメータ.

γ : translation パラメータ.

スケール

中心位置

2.4 ローカル基底関数ネットワーク

引数がベクトルである場合の構成

$$\phi(t) = [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]^T$$

ラジアル (radial) 構成

$$g_k(\phi) = \chi \left(\|\phi - \gamma_k\|_{\beta_k}^2 \right)$$

$$\|\phi - \gamma_k\|_{\beta_k}^2 = (\phi - \gamma_k)^T \beta_k^{-2} (\phi - \gamma_k)$$

$$\beta_k^{-1} = \text{diag}(\beta_{k,1}^{-1} \cdots \beta_{k,d}^{-1})$$

$$\gamma_k = [\gamma_{k,1} \cdots \gamma_{k,d}]^T$$

距離に依存

スケール

中心位置

重み付き距離の2乗

2.4 ローカル基底関数ネットワーク

具体例: RBFN (Radial Basis Function Network)

$$\exp \left\{ -\frac{1}{2} \left(\frac{x - \gamma}{\beta} \right)^2 \right\}$$

もっとも代表的なRBF

引数がスカラーの場合

ラジアル構成

$$g_k(\phi) = \exp \left\{ -\frac{1}{2} \left[\frac{(\phi_1 - \gamma_{k,1})^2}{\beta_{k,1}^2} + \dots + \frac{(\phi_d - \gamma_{k,d})^2}{\beta_{k,d}^2} \right] \right\}$$

簡単のため, $\beta_{k,1} = \beta_{k,2} = \dots = \beta_{k,d} = \beta_k$ (single scaling)
とすることが多い.

2.4 ローカル基底関数ネットワーク

ガウス関数の特徴：滑らかな補間

ガウシアン関数のフーリエ変換はまたガウシアン関数：

$$\begin{aligned} G(\omega) &= \int_{-\infty}^{\infty} \frac{1}{\beta\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x}{\beta}\right)^2\right] e^{-j\omega x} dx \\ &= \exp\left[-\frac{\beta^2\omega^2}{2}\right] \end{aligned}$$

- β が大きいと、ガウス関数 $g(x)$ が広がる(滑らか)。
- β が大きいと、そのフーリエ変換は狭まる(低周波特性)。
- 補間できる関数の周波数帯域を見積もれる。

2.4 ローカル基底関数ネットワーク

具体例: FBFN (Fuzzy Basis Function Network)

$$w_k(\phi) = \frac{g_k(\phi)}{\sum_{k=1}^M g_k(\phi)}$$

$$0 \leq w_k(\phi) \leq 1, \quad \sum_{k=1}^M w_k(\phi) = 1,$$

正規化RBFともいう。

RBFを正規化すると、
FBFになる。

Partition of Unity

2.4 ローカル基底関数ネットワーク

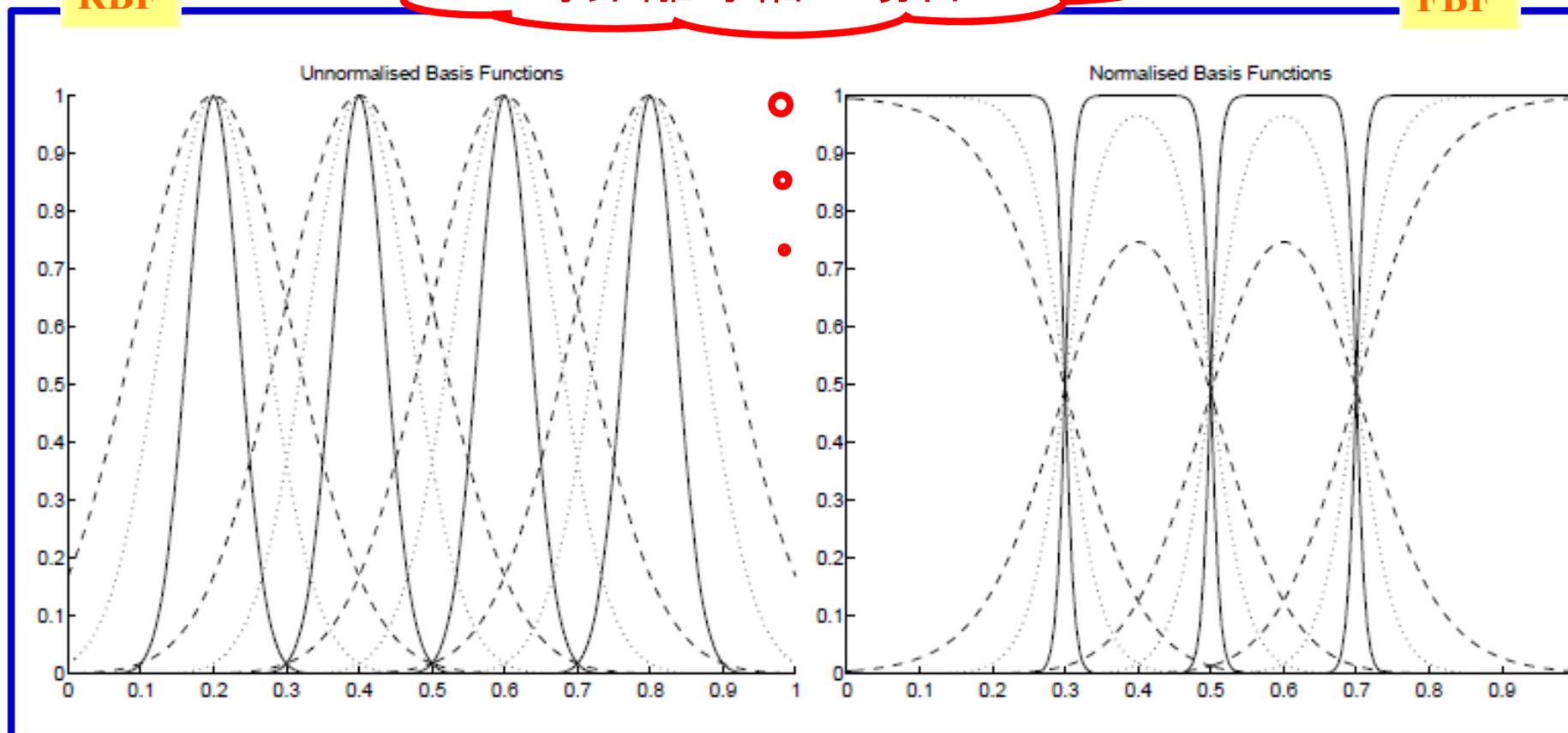
FBFとRBFはどう違う？

Robert Shorten and Roderick Murray-Smith : Side Effects of Normalising Radial Basis Function Networks, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.1261>

RBF

等距離等幅の場合

FBF

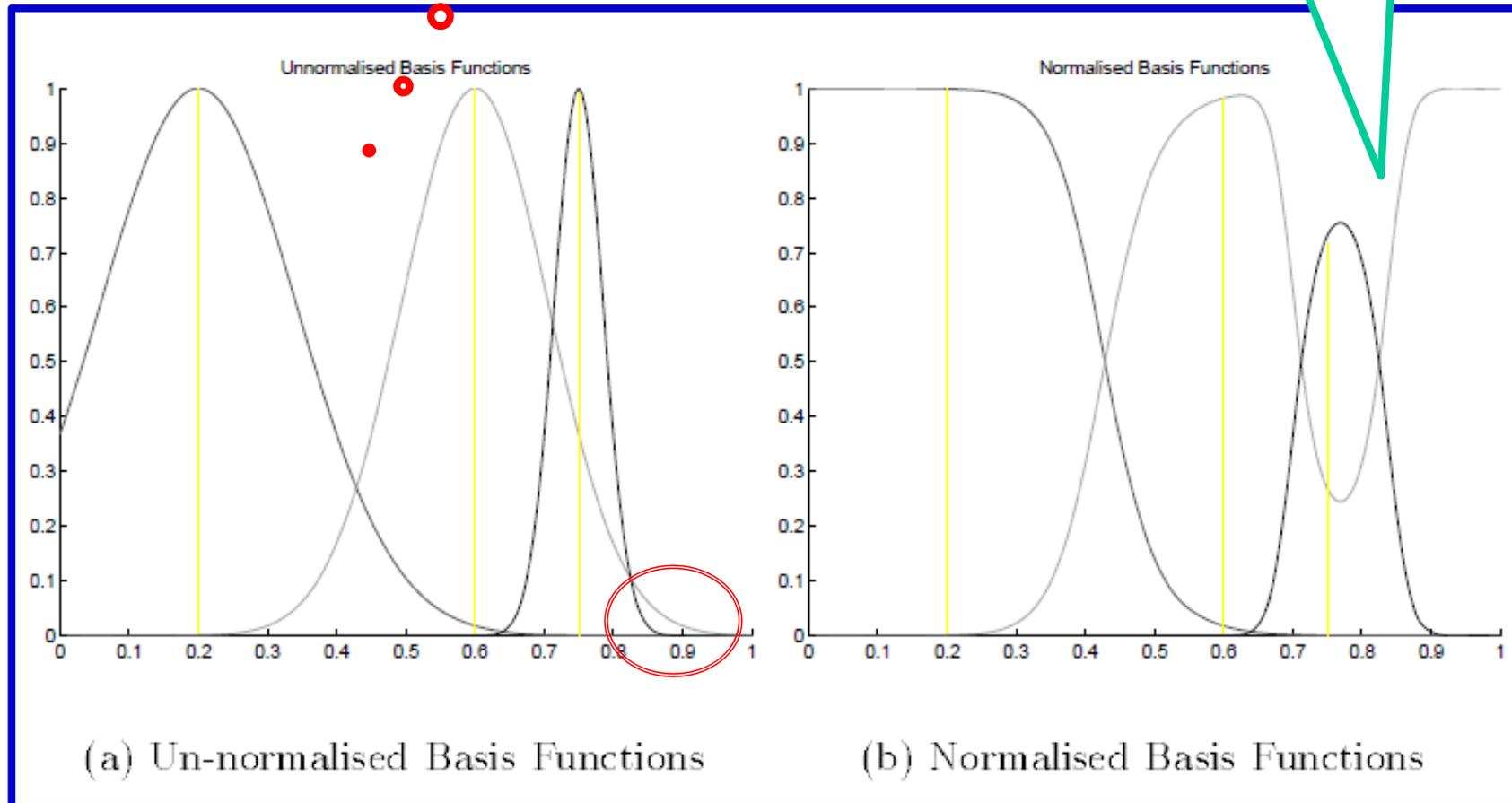


2.4 ローカル基底関数ネットワーク

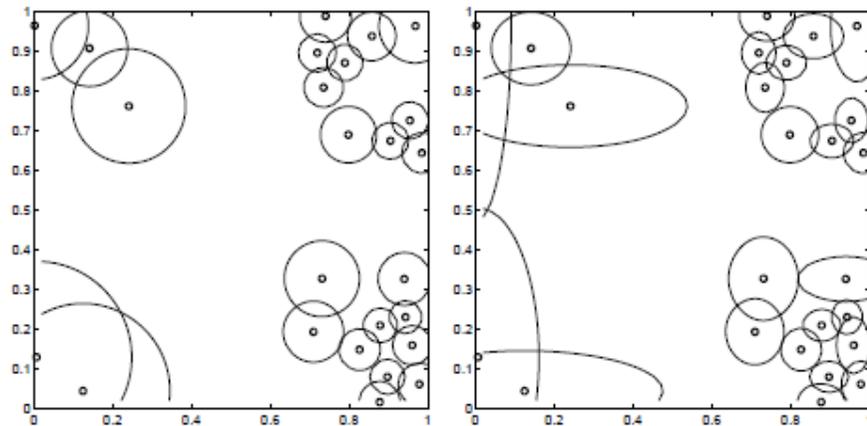
もうファジィではないよ！

等距離等幅でない場合

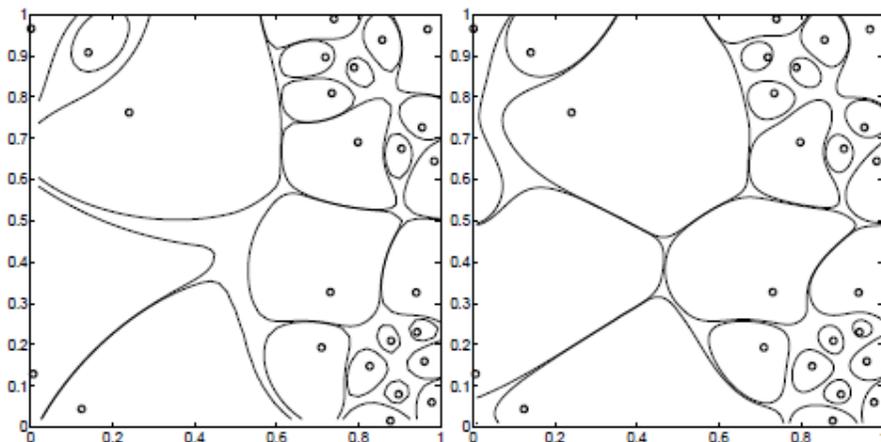
多峰性？



2.4 ローカル基底関数ネットワーク



(a) Un-normalised basis functions

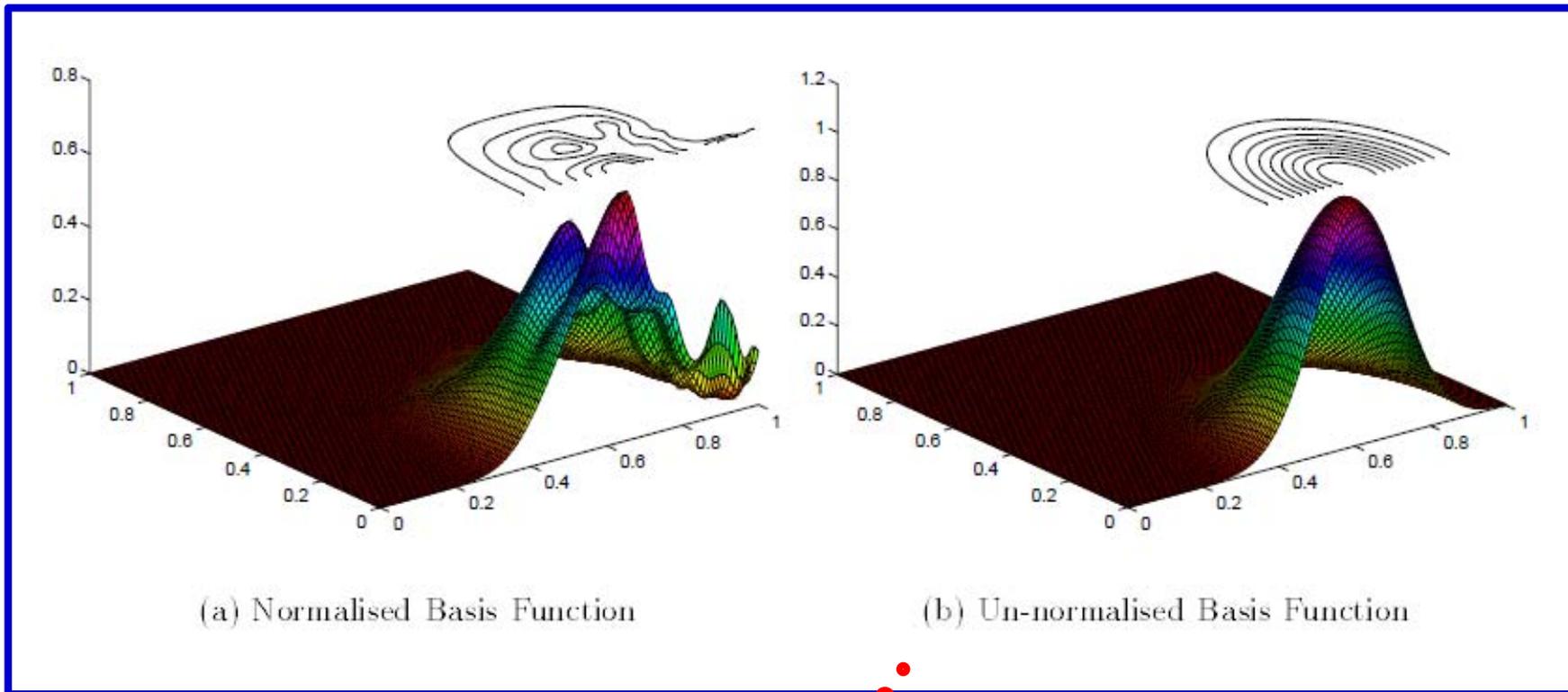


(b) Normalised basis functions

正規化基底による分割の
ほうが全領域をよく覆う。

2.4 ローカル基底関数ネットワーク

正規化基底の形状はちょっと奇妙.



データがフィッティングしても、中身の形状を知るべき！

2.4 ローカル基底関数ネットワーク

具体例: WLBFN (Wavelet Basis Function Network)

メキシカン・ハット・ウェーブレット:

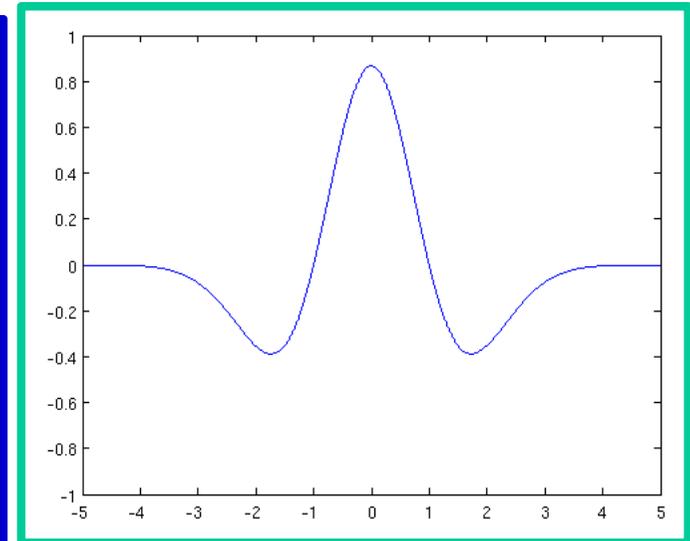
$$\psi(x) = (1 - x^2)\exp(-x^2)$$

ラジアル構成:

$$\Psi(\mathbf{x}) = (d - \|\mathbf{x}\|^2) \exp(-\|\mathbf{x}\|^2), \quad \mathbf{x} \in R^d.$$

$$\hat{g}(\phi(t), \theta) = \sum_{k=1}^M \alpha_k g_k(\phi) + \alpha_0$$

$$g_k(\phi) = \det\left(\beta_k^{-\frac{1}{2}}\right) \Psi\left[\beta_k^{-1}(\phi - \gamma_k)\right]$$



直流分をつける.

Q. Zhang, Using Wavelet Network in Nonparametric Estimation, IEEE Transactions on Neural networks, 8-2, 227/236 (1997)

2.4 ローカル基底関数ネットワーク

ローカル基底関数ネットワークの学習

$$\hat{g}(\phi(t), \theta) = \sum_{k=1}^M \alpha_k g_k(\phi)$$

$$g_k(\phi) = \chi \left(\|\phi - \gamma_k\|_{\beta_k^{-2}}^2 \right)$$
$$\|\phi - \gamma_k\|_{\beta_k^{-2}}^2 = (\phi - \gamma_k)^T \beta_k^{-2} (\phi - \gamma_k)$$
$$\beta_k^{-1} = \text{diag}(\beta_{k,1}^{-1} \cdots \beta_{k,d}^{-1})$$
$$\gamma_k = [\gamma_{k,1} \cdots \gamma_{k,d}]^T$$

2.4 ローカル基底関数ネットワーク

LBFN のパラメータベクトル： $\theta = [\alpha, \beta, \gamma]^T$.

LBFN 学習の特徴：

勾配法などの非線形学習アルゴリズムを用いるとき、非線形パラメータ β, γ は事前情報に基づいて初期値を設定できる.

β, γ を固定すれば、LBFN の学習は線形最小二乗法によって行える.

2.4 ローカル基底関数ネットワーク

(A) LBFN の入力空間を分割して、基底関数の有効に補間できる領域がシステムの動作領域を含むように β, γ を事前に設定しておく。そして、数多くの基底関数をから直交化最小二乗法を含む種々の 直交変換の手法により、必要最小限の基底関数を選ぶ。

M. Setnes, R. Babuska, Rule base reduction: some comments on the use of orthogonal transforms, IEEE Transactions on Systems, Man and Cybernetics, 31-2, 199/205 (2001).

(B) システムの動作領域はその空間上の一部だけを占めることが多いので、クラスタリングの手法によって求められた ϕ の空間分布にしたがって、基底関数を配置し、 β, γ を設定する手法。

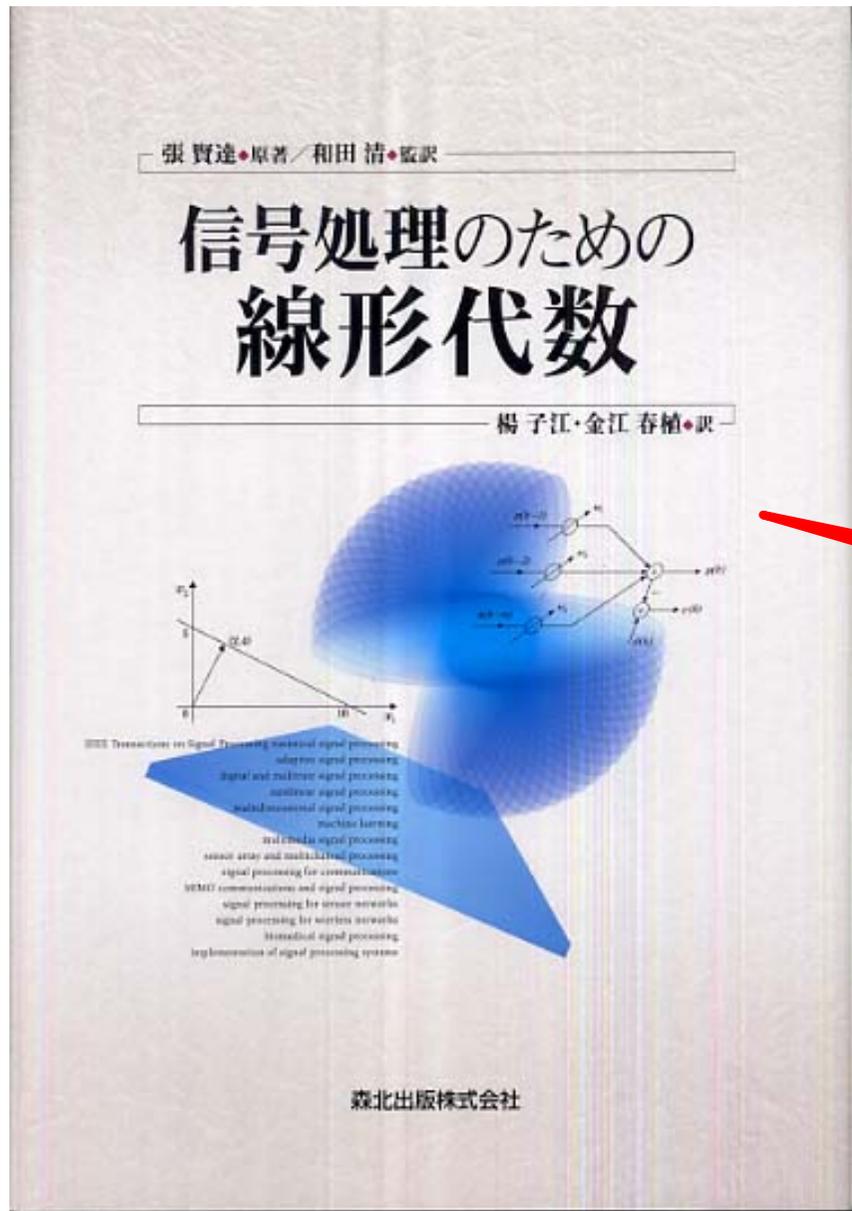
(C) 遺伝的アルゴリズムを用いて β, γ を設定する手法。

(D) パラメータのオンライン学習アルゴリズムに基底関数の数を適応的に増減させる機能を付加した手法。

J. B. Gomm and D. L. Yu, Selecting radial basis function network centers with recursive orthogonal least squares training, IEEE Transactions on Neural networks, 11-2, 306/313 (2002)

直交変換は有用！

2.4 ローカル基底関数ネットワーク



直交変換の勉強なら

2.5 パラメータ推定と正則化

システム :

$$y(t) = g_0(\phi(t)) + e(t)$$

(予測誤差の) 最小二乗評価 :

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta)$$

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \left(y(t) - \hat{g}(\phi(t), \theta) \right)^2$$

モデルの「良さ」評価 :

$$\begin{aligned} \bar{V}(\theta) &= E \left(y(t) - \hat{g}(\phi(t), \theta) \right)^2 \\ &= \lambda + E \left(g_0(\phi(t)) - \hat{g}(\phi(t), \theta) \right)^2 \end{aligned}$$

$E[\cdot]$ はサンプル平均で, λ は白色雑音 $e(t)$ の分散である.

最良のパラメータ :

$$\theta^* = \arg \min_{\theta} \bar{V}(\theta)$$

2.5 パラメータ推定と正則化

モデルの予測誤差が白色になるときのパラメータを θ^* とする。

同定されたモデル $\hat{g}(\phi(t), \hat{\theta}_N)$ の評価関数表現：

$$\begin{aligned} E[\bar{V}(\hat{\theta}_N)] &= \lambda + E\left(g_0(\phi(t)) - \hat{g}(\phi(t), \hat{\theta}_N)\right)^2 \\ &\approx \lambda + E\left(g_0(\phi(t)) - \hat{g}(\phi(t), \theta^*)\right)^2 + \lambda \frac{m}{N} \end{aligned}$$

m はパラメータ数, N はデータ数

近似誤差, 基底関数
が増えると, 小さくなる.

雑音による分散誤差, 基底
関数が増えると, 大きくなる.

オーバーフィッティング: 近似誤差が小, 分散誤差が大

2.5 パラメータ推定と正則化

オーバーフィッティングの回避: 有効パラメータ数 m を低減

正則化手法: パラメータのペナルティーを加えた評価関数

$$W_N(\boldsymbol{\theta}) = V_N(\boldsymbol{\theta}) + \delta \|\boldsymbol{\theta}\|^2$$

前頁の結果の中の m に相当する部分は以下となる.

$$r(m, \delta) = \sum_{k=1}^m \frac{\sigma_i^2}{(\sigma_i + \delta)^2} \leq m$$

σ_i はヘシアン $\bar{V}''(\boldsymbol{\theta})$ (\bar{V} を $\boldsymbol{\theta}$ で2階微分) の i 番目の固有値, $r(m, \delta)$ は近似的に δ より大きい固有値の数と見なせる. 有効なパラメータ数が削減される.

J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson and A. Juditsky: Nonlinear black-box modeling in system identification: a unified overview, *Automatica*, 31-12, 1691/1724(1995).

基底関数の数を減らす手法: 基底関数のライブラリーから必要最小限の基底関数を選ぶ場合に, AIC や BIC といった評価関数を用いることによって基底関数の数にペナルティーを課す.

繰り返し学習を早めに打ち切る方法: 同定データの近似に有効なパラメータ数を減らすことと等価.

J. Sjöberg and L. Ljung: Overtraining, regularization and searching for a minimum, with application to neural networks, *International Journal of Control*, 62-6, 1391/1407(1992).

2.5 パラメータ推定と正則化

AICとBIC

システム：

$$y(t) = g_0(\phi(t)) + e(t)$$

雑音 $e(t)$ が平均値ゼロの正規分布に従う。

パラメータ推定は最尤推定法で求められる（雑音が正規分布の時、線形最小二乗法は最尤推定）とする。

赤池情報量規範：

$$AIC = N \ln \hat{\tau}_e^2 + 2m$$

MDL (minimum description length) または BIC (Bayesian Information Criterion)

$$BIC = N \ln \hat{\tau}_e^2 + m \ln(N)$$

$\hat{\tau}_e^2$ は予測誤差の分散推定値：

$$\hat{\tau}_e^2 = \frac{1}{N} \sum_{t=t_s+1}^{t_s+N} [\hat{y}(t) - y(t)]^2$$

データ数が多いとき、BIC のほうがよりけちな (parsimonious) モデルを選ぶ。

仮定を無視して使う方もいる。
複雑さのペナリティがあれば、
いいだろう？

モデル複雑さとデータ適合の
trade-off

(3) 非線形同定のための最適化手法

(3) 非線形同定のための最適化手法

3.1 ソフトコンピューティングによる最適化手法

3.2 直交化最小二乗法

3.3 非線形最小二乗法

3.1 ソフトコンピューティングによる最適化手法

遺伝的アルゴリズム(GA)

- 遺伝的アルゴリズムは解の候補を遺伝子で表現した「個体」を複数用意し、適応度(評価関数)の高い個体を優先的に選択して交叉(組み換え)・突然変異などの操作を繰り返しながら解を探索する。
- 評価関数の可微分性や単峰性などの知識がない場合であっても適用可能なことである。
- また、遺伝子の表現の仕方によっては組合せ最適化問題やNP困難な問題などのさまざまな問題に適用可能である。

計算機の腕力に頼れば、構造決定と非線形パラメータの学習は可能

前述の基本原則を暗黙的に理解すべき。

3.1 ソフトコンピューティングによる最適化手法

粒子群最適化 (Particle Swarm Optimization, PSO)

- 昆虫の大群や魚群において、一匹がよさそうな経路を発見すると(すなわち、食料を発見したとか安全であるという場合)、群れの残りはどこにいても素早くそれに倣うことができる。
- これは多次元空間において位置と速度を持つ粒子群でモデル化される。これらの粒子はハイパー空間を飛びまわり、最善な位置を探す。位置の評価は適応度で行う。群れのメンバーは良い位置について情報交換し、それに基づいて自身の位置と速度を調整する。

非線形パラメータの学習に利用される

3.1 ソフトコンピューティングによる最適化手法

学習効率向上の工夫

- 非線形パラメータだけをGA或いはPSOで決め、線形パラメータを最小二乗法で求める。
- GA或いはPSOは、できれば、勾配法などの局所探索法とハイブリッド化したほうがよい。

探索空間次元の
低減

局所探索の加速

3.2 直交化最小二乗法

Gram-Schmidtの直交化に基づいた直交化最小二乗法

S. Chen, S. A. Billings and W. Luo, Orthogonal least squares methods and their application to non-linear system identification, International Journal of Control, 50-5, 1873/1896 (1989)

S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, Practical identification of NARMAX models using radial basis functions, International Journal of Control, 52-6 1327/1350 (1990)

S.Chen,X.Hong,B.L.Luk,C.J.Harris, Construction of tunable radial basis function networks using orthogonal forward selection ,IEEETrans.Syst. Man Cybern. PartB, 39-2, 457/466 (2009).

楊子江:直交化最小二乗法による階層型ニューラルネットワークの中間層ニューロン数削減法, 計測自動制御学会論文集, 33-3, 216/223(1997).

直交変換を用いて, 出力ベクトルを近似するのに必要な信号ベクトルを, 候補ライブラリーから選び出す.

3.2 直交化最小二乗法

信号 $d(t)$ の合成を考える.

$$\begin{aligned}d(t) &= \mathbf{z}^T(t) \mathbf{w}^o \\ \mathbf{z}^T(t) &= [z_1(t), z_2(t), \dots, z_M(t)] \\ \mathbf{w}^o &= [w_1^o, w_2^o, \dots, w_M^o]^T\end{aligned}$$

M は冗長であることが多い

$t = 1, \dots, N$ に対して, データベクトルを構築する.

$$\begin{aligned}\mathbf{d} &= \mathbf{P} \mathbf{w}^o \\ \mathbf{d} &= [d(1), d(2), \dots, d(N)]^T \in R^{N \times 1} \\ \mathbf{P} &= [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M] \in R^{N \times M} \\ \mathbf{p}_j &= [z_j(1), z_j(2), \dots, z_j(N)]^T \in R^{N \times 1} \quad (j = 1, 2, \dots, M)\end{aligned}$$

データを縦に並べる

データベクトル $\mathbf{p}_j (j = 1, 2, \dots, M)$ を, それぞれが採用された場合の $d(t)$ の自乗和に対する寄与が大きい順番 (仮に分かったとして) で並べ替える.

$$\begin{aligned}\mathbf{d} &= \mathbf{P}' \mathbf{c}_o \\ \mathbf{c}_o &= [c_1, c_2, \dots, c_M]^T \\ \mathbf{P}' &= [\mathbf{p}'_1, \mathbf{p}'_2, \dots, \mathbf{p}'_M] \\ \mathbf{p}'_j &= [p'_j(1), p'_j(2), \dots, p'_j(N)]^T \quad (j = 1, 2, \dots, M)\end{aligned}$$

3.2 直交化最小二乗法

回帰行列 P' の列ベクトル p'_j ($j = 1, \dots, M$) を直交基底ベクトル r_j ($j = 1, \dots, M$) に変換する.

$$P' = R A_o \quad \underline{P' = [p'_1, p'_2, \dots, p'_M]}$$

$$A_o = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1M} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2M} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & & \ddots & 1 & \alpha_{MM} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \in R^{M \times M}, \quad R = [r_1, r_2, \dots, r_M] \in R^{N \times M}$$

$r_j = [r_j(1), r_j(2), \dots, r_j(N)]^T$ ($j = 1, 2, \dots, M$) は互いに直交.

直交

Gram-Schmidt 法 :

$$\left. \begin{aligned} r_1 &= p'_1 \\ \alpha_{ij} &= \frac{r_i^T p'_j}{r_i^T r_i} \quad (1 \leq i < j) \\ r_j &= p'_j - \sum_{i=1}^{j-1} \alpha_{ij} r_i \end{aligned} \right\} (j = 2, \dots, M)$$

前のベクトルとの相関係数

前のベクトルとの相関成分を除去

3.2 直交化最小二乗法

直交データ

$$d = P' c_0 = R \underline{A_0 c_0} = R g_0$$

ただし, $g_0 = [g_1, g_2, \dots, g_M]^T$ は直交変換後の新しいパラメータベクトル.

$R^T d = R^T R g_0$ より, パラメータ推定値

$$\hat{g}_j = \frac{r_j^T d}{r_j^T r_j} \quad (j = 1, 2, \dots, M)$$

基底ベクトル r_j は互いに直交であるので, いくつかの項を削除したり, 新たな項を加えたりしても, すでに計算された \hat{g}_0 のほかの要素に影響を与えない.

また, 各基底ベクトルによる $d(t)$ の自乗和への寄与の計算が可能となる.

3.2 直交化最小二乗法

M_s 個のベクトルが選ばれたとすると、パラメータベクトル \mathbf{c} の推定値は次の関係で簡単に求められる。

$$\mathbf{A} \hat{\mathbf{c}} = \hat{\mathbf{g}}$$

ただし、

$$\hat{\mathbf{c}} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_{M_s}]^T$$

$$\hat{\mathbf{g}} = [\hat{g}_1, \hat{g}_2, \dots, \hat{g}_{M_s}]^T$$

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1M_s} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2M_s} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & & \ddots & 1 & \alpha_{M_s M_s} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

3.2 直交化最小二乗法

実現

実際に、列ベクトルの総数 M は数百以上. 大きいサイズのマトリクスに関する計算を直接行う必要がなく、 P の M 個の列ベクトルから、 $d(t)$ の自乗和に対する寄与がもっとも大きいものを選び、直交変換を行いながら、マトリクス P' と R の列ベクトルを順次構築していく.

M_s 個の項だけを選び、ほかの項を削除した場合、 $d(t)$ の自乗和は

$$d^t d = \sum_{j=1}^{M_s} g_j^2 r_j^T r_j + \underline{E^T E}$$

第1項は採用されたベクトル r_j による $d(t)$ の自乗和への寄与、第2項は $d(t)$ を再構成する場合の自乗誤差.

$E = [e(1), e(2), \dots, e(N)]^T$ は $x(t)$ を再構成するときの近似誤差ベクトル.

$$e(t) = \sum_{j=M_s+1}^M r_j(t) g_j = d(t) - \sum_{j=1}^{M_s} r_j(t) g_j = d(t) - \sum_{j=1}^{M_s} p'_j(t) c_j$$

3.2 直交化最小二乗法

$$d^t d = \sum_{j=1}^{M_s+1} g_j^2 \mathbf{r}_j^T \mathbf{r}_j + \mathbf{E}^T \mathbf{E}$$

$g_j^2 \mathbf{r}_j^T \mathbf{r}_j$ はベクトル \mathbf{r}_j を取り入れることによる $d(t)$ の自乗和の増加分.

ベクトル \mathbf{r}_j の導入による 誤差減少率 (error reduction ratio) :

$$[\text{err}]_j = \frac{g_j^2 \mathbf{r}_j^T \mathbf{r}_j}{d^T d} \quad \text{寄与率}$$

マトリクス \mathbf{P} の列ベクトルのうち、 $[\text{err}]_j$ を最も大きくするベクトルを逐次選んで、直交変換を行っていくという Feedforward regression アルゴリズムを用いて、AIC や BIC に基づき、必要最小限の項の数で $d(t)$ を再構成すればよい。

3.3 非線形最小二乗法

パラメータ τ に関して非線形, θ に関して線形のモデル:

$$y(t) = \varphi^T(t, \tau)\theta + e(t)$$

二乗誤差:

$$V_N(\theta, \tau) = \frac{1}{N - k_s} \sum_{k=k_s+1}^N \frac{1}{2} \epsilon^2(t, \theta, \tau)$$

$$\epsilon(t, \theta, \tau) = y(t) - \varphi^T(t, \tau)\theta$$

非線形最小二乗問題:

$$\hat{\Theta}_N^T = \left[\hat{\theta}_N^T, \hat{\tau}_N \right]^T = \arg \min_{\theta, \tau} V_N(\theta, \tau)$$

3.3 非線形最小二乗法

非分離型非線形最小二乗法(ガウスニュートン法)

$$\hat{\Theta}_N^{(j+1)} = \hat{\Theta}_N^{(j)} - \mu_N^{(j)} \left[\mathbf{R}_N(\hat{\Theta}_N^{(j)}) \right]^{-1} \mathbf{V}'_N(\hat{\Theta}_N^{(j)})$$

$\mathbf{V}'_N(\Theta)$ と $\mathbf{R}_N(\Theta)$ はそれぞれ評価関数の勾配とヘッシャンの推定値：

$$\mathbf{V}'_N(\Theta) = -\frac{1}{N - k_s} \sum_{k=k_s+1}^N \zeta(t, \Theta) \epsilon(t, \Theta)$$
$$\mathbf{R}_N(\Theta) = \frac{1}{N - k_s} \sum_{k=k_s+1}^N \zeta(t, \Theta) \zeta^T(t, \Theta)$$

$$\zeta(t, \Theta) = -\frac{\partial \epsilon(t, \Theta)}{\partial \Theta}$$

射影操作という

$\mu^{(j)}$ はステップ・サイズで、 $V_N(\Theta)$ が毎回減少するように、パラメータ推定値が拘束条件を満足するように決める。満足されない場合は、 $\mu^{(j)}$ を半減して、パラメータ推定値を再計算、満足されるまで繰り返す。

射影操作がないと、最適解の近傍からスタートしても、遠くまで跳んでしまう。

3.3 非線形最小二乗法

仮に非線形パラメータ τ が既知であるとする、 θ は線形最小二乗法で推定される:

$$\begin{aligned}\hat{\theta}_N(\tau) &= \mathbf{R}^{-1}(N, \tau) \mathbf{f}(N, \tau) \\ \mathbf{R}(N, \tau) &= \frac{1}{N - k_s} \sum_{k=k_s+1}^N \varphi(t, \tau) \varphi^T(t, \tau) \\ \mathbf{f}(N, \tau) &= \frac{1}{N - k_s} \sum_{k=k_s+1}^N \varphi(t, \tau) y(t)\end{aligned}$$

評価関数 $V_N(\theta, \tau)$ に代入すると,

$$\check{V}_N(\tau) = \frac{1}{N - k_s} \sum_{k=k_s+1}^N \frac{1}{2} \check{\varepsilon}^2(t, \tau)$$

$$\check{\varepsilon}(t, \tau) = \xi_{0\bar{y}}(t) - \varphi^T(t, \tau) \mathbf{R}^{-1}(N, \tau) \mathbf{f}(N, \tau)$$

パラメータ分離

G. Golub and V. Pereyra, Separable nonlinear least squares: the variable projection method and its applications, *Inverse Problems*, 19-2 (2003)

3.3 非線形最小二乗法

分離型非線形最小二乗法

分離型非線形最小二乗問題：

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_N^T \\ \hat{\boldsymbol{\tau}}_N^T \end{bmatrix}^T = \underset{\boldsymbol{\theta}, \boldsymbol{\tau}}{\operatorname{argmin}} V_N(\boldsymbol{\theta}, \boldsymbol{\tau}) = \underset{\boldsymbol{\tau}}{\operatorname{argmin}} \check{V}_N(\boldsymbol{\tau})$$

分離型非線形最小二乗法：

$$\hat{\boldsymbol{\tau}}_N^{(j+1)} = \hat{\boldsymbol{\tau}}_N^{(j)} - \mu^{(j)} \left[\check{\mathbf{R}}_N(\hat{\boldsymbol{\tau}}_N^{(j)}) \right]^{-1} \check{\mathbf{V}}'_N(\hat{\boldsymbol{\tau}}_N^{(j)})$$

$\check{\mathbf{V}}'_N(\boldsymbol{\tau})$ と $\check{\mathbf{R}}_N(\boldsymbol{\tau})$ はそれぞれ評価関数の勾配とヘッシャン。

$$\check{\mathbf{V}}'_N(\boldsymbol{\tau}) = -\frac{1}{N - k_s} \sum_{k=k_s+1}^N \boldsymbol{\psi}(t, \boldsymbol{\tau}) \check{\varepsilon}(t, \boldsymbol{\tau})$$

$$\check{\mathbf{R}}_N(\boldsymbol{\tau}) = \frac{1}{N - k_s} \sum_{k=k_s+1}^N \boldsymbol{\psi}(t, \boldsymbol{\tau}) \boldsymbol{\psi}^T(t, \boldsymbol{\tau})$$

$$\boldsymbol{\psi}(t, \boldsymbol{\tau}) = -\frac{\partial \check{\varepsilon}(t, \boldsymbol{\tau})}{\partial \boldsymbol{\tau}}$$

$\hat{\boldsymbol{\tau}}_N$ が得られると、 $\boldsymbol{\theta}$ は線形最小二乗法で推定される。

逆行列のサイズ
が小さいので、数
値的条件がよい。

結果の導出は面倒。

3.3 非線形最小二乗法

グローバル非線形最小二乗法

分離型：

$$\hat{\tau}_N^{(j+1)} = \hat{\tau}_N^{(j)} - \mu^{(j)} \left[\check{\mathbf{R}}_N(\hat{\tau}_N^{(j)}) \right]^{-1} \left[\check{\mathbf{V}}_N'(\hat{\tau}_N^{(j)}) - \beta^{(j)}\eta \right]$$

非分離型：

$$\hat{\Theta}_N^{(j+1)} = \hat{\Theta}_N^{(j)} - \mu_N^{(j)} \left[\mathbf{R}_N(\hat{\Theta}_N^{(j)}) \right]^{-1} \left[\mathbf{V}_N'(\hat{\Theta}_N^{(j)}) - \beta^{(j)}\eta \right] \begin{matrix} \bullet \\ \bullet \\ \bullet \end{matrix}$$

勾配に確率的摂動を加えると、
グローバル探索は可能

η は分散 1 の正規分布確率雑音。

グローバル最適解が得られる

β^j はスカラーで、初期値が大きく、反復計算が進むにつれ、小さくなるように設定される。例えば、評価関数に比例するように減少していく：

3.3 非線形最小二乗法

応用例: 未知なむだ時間(非線形)と未知な線形パラメータを持つ2入力1出力線形連続時間システムの同定

$$\ddot{x}(t) + a_1\dot{x}(t) + a_2x(t) = b_{11}\dot{u}_1(t - \tau_1) + b_{12}u_1(t - \tau_1) \\ + b_{21}\dot{u}_2(t - \tau_2) + b_{22}u_2(t - \tau_2)$$

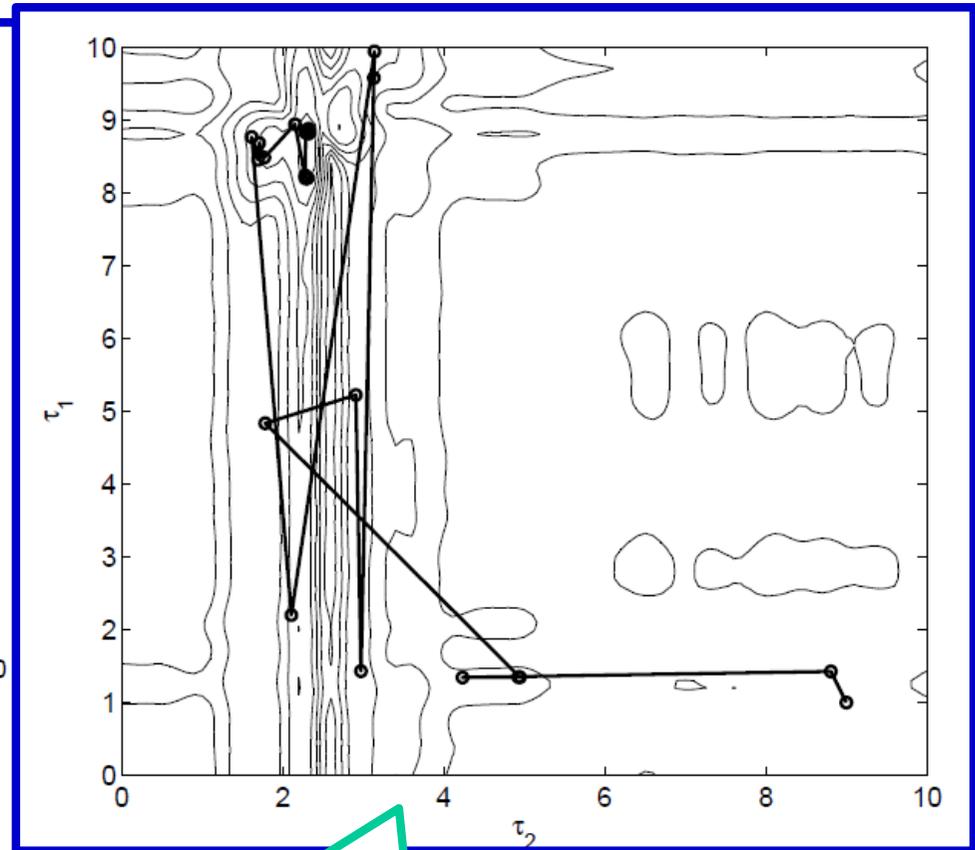
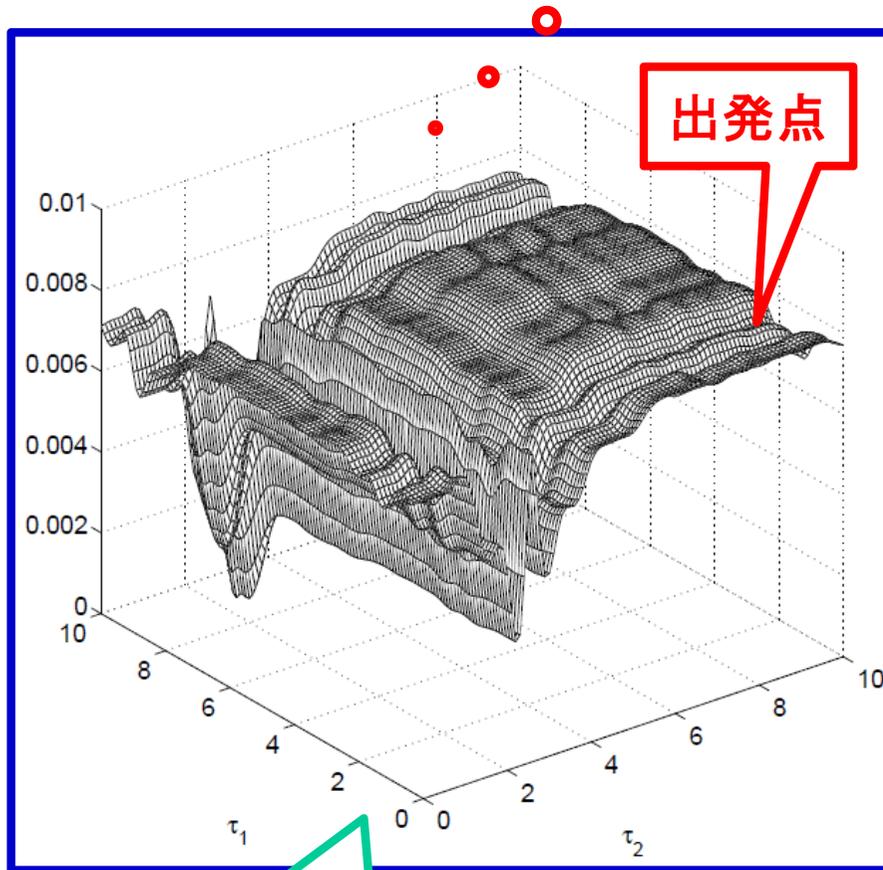
$$a_1 = 3.0, a_2 = 2.0, b_{11} = 1.0, b_{12} = 2.0$$

$$b_{21} = 2.0, b_{22} = 2.0, \underline{\tau_1 = 8.83}, \underline{\tau_2 = 2.32}$$

Zi-Jiang Yang, H. Iemura, S. Kanae and K. Wada: Identification of Continuous-Time Systems with Multiple Unknown Time Delays by Global Nonlinear Least-Squares and Instrumental Variables Methods, Automatica, 43- 7, 1257/1264 (2007).

3.3 非線形最小二乗法

局所解はいっぱいありそう。



評価関数の形状(分離型)

グローバルな最適解にたどり着く

(4) 線形パラメータ時変 (LPV) モデルの同定

(4) 線形パラメータ時変 (LPV) モデルの同定

4.1 LMN (Local Model Network) モデルの導入

4.2 LMNモデルの同定

4.3 線形パラメータ時変 (LPV) モデルと制御系設計

4.4 応用事例

4.1 LMNモデルの導入

LMNモデルの導入

$$y(t) = g_0(\phi(t)) + e(t)$$

$$\phi(t) = [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]^T$$

$u(t)$: 入力信号 $y(t)$: 出力信号 $e(t)$: ゼロ平均の白色雑音.

g_0 : 未知な連続関数 (ブラックボックス) $\phi(t)$: 回帰ベクトル.

動作点 ϕ_k^* の近傍における局所近似線形モデル:

テーラー展開

$$y(t) = \psi^T(t)\theta_k + O(\|\phi_k - \phi_k^*\|^2) + e(t)$$

$$\psi(t) = [-y(t-1), \dots, -y(t-n_y), u(t-1), \dots, u(t-n_u), \underline{1}]^T$$

$$\theta_k = [a_{k,1}, \dots, a_{k,n_u}, b_{k,1}, \dots, b_{k,n_y}, \underline{d_k}]^T$$

局所モデルネットワーク

システムが広い範囲にわたって動作する場合

- (a) 動作範囲の分割,
- (b) 局所モデルの構築,
- (c) 局所モデルの滑らかな補間.

複数の局所モデルによる補間: LMNモデル

ウィーン工科大のグループ <http://mbc.tuwien.ac.at/publikationen/>

ジーゲン大学のグループ

<http://www.mb.uni-siegen.de/mrt/mitarbeiter/ehemalige/hartmann/index.html.en?lang=en>

4.1 LMNモデルの導入

動作範囲 Φ を M 個の局所動作領域 $\Phi_k (k = 1, \dots, M)$ に分割.

LMN モデル :

重み関数が決定されると,
パラメータに関して線形

$$\hat{g}(\phi(t), \theta) = \sum_{k=1}^M \underline{w_k(\phi)} [\psi^T(t) \theta_k] = z^T(t) \theta_g$$

$$z(t) = [\psi^T(t) w_1(\phi), \dots, \psi^T(t) w_M(\phi)]^T$$

ベクトルの次元拡大

$$\theta_g = [\theta_1^T, \dots, \theta_M^T]^T$$

$w_k(\phi)$: 重み関数 (前述の FBF と同じ) :

$$w_k(\phi) = \frac{g_k(\phi)}{\sum_{k=1}^M g_k(\phi)}$$

$$0 \leq w_k(\phi) \leq 1, \quad \sum_{k=1}^M w_k(\phi) = 1,$$

4.1 LMNモデルの導入

重み関数による影響

- LMNモデル: Takagi-Sugenoのファジィモデル, 一次テララー展開.
- FBFN: シングルトンファジィモデル, ゼロ次テララー展開.
- ガウス関数の幅がすべてゼロに近づくと, 重み関数は矩形関数になり, LMNモデルは区分的線形モデルとなる.

PieceWise Affine (PWA) モデル

- ガウス関数の幅がすべて無限大に近づくと, LMNモデルは一つのグローバルな線形モデルになる.

4.2 LMNモデルの同定

領域分割手法

プラントの操業経験などの事前情報に基づいて動作範囲を適切に分割する.

動作範囲の各座標軸を二分割していくというバイナリーツリー(binary tree)構造表現を用いる.
local linear model trees(LOLIMOT)

データの空間分布に注目したクラスタリングの手法.

最適化手法で重み関数の中心や幅を求める.

4.2 LMNモデルの同定

パラメータ推定

グローバル評価：

$$J_g(\boldsymbol{\theta}_g) = \sum_{t=1}^N \left(y(t) - \mathbf{z}^T(t) \boldsymbol{\theta}_g \right)^2$$

(1)：同定データに対する予測性能がよい。

(2)：各局所モデルが実際の非線形システムの局所近似線形モデルとかなり異なるものになってしまう可能性があり、汎化能力が優れていない。

ローカル評価：

$$J_k(\boldsymbol{\theta}_k) = \sum_{t=1}^N w_k(\phi) \left(y(t) - \boldsymbol{\psi}^T(t) \boldsymbol{\theta}_k \right)^2$$

(1)：各局所モデルは非線形システムの局所近似線形モデルと見なすことができ、モデルの汎化能力や物理的解釈などの面で有利。正則化の観点からも解釈できる。

(2)：グローバル評価 $J_g(\boldsymbol{\theta}_g)$ に関しては不利。

局所モデルパラメータをローカル評価、重み関数の中心位置や幅などをグローバル評価にそれぞれ基づいて反復計算によって求めるアプローチが推奨される。

4.3 線形パラメータ時変(LPV)モデルと制御系設計

LMNモデルによる並列分散モデル

$$y(t) = \hat{A}(q^{-1}, \mathbf{w}(\phi))y(t) + \hat{B}(q^{-1}, \mathbf{w}(\phi))u(t) + \hat{D}(\mathbf{w}(\phi)) + e(t)$$

ただし, q^{-1} は遅延演算子 : $q^{-1}y(t) = y(t-1)$.

$$\hat{A}(q^{-1}, \mathbf{w}(\phi)) = \sum_{k=1}^M w_k(\phi) \hat{A}_k(q^{-1})$$

$$\hat{B}(q^{-1}, \mathbf{w}(\phi)) = \sum_{k=1}^M w_k(\phi) \hat{B}_k(q^{-1})$$

$$\hat{A}_k(q^{-1}) = \sum_{i=1}^{n_y} \hat{a}_{k,i} q^{-i}$$

$$\hat{B}_k(q^{-1}) = \sum_{i=1}^{n_u} \hat{b}_{k,i} q^{-i}$$

$$\hat{D}(\mathbf{w}(\phi)) = \sum_{k=1}^M w_k(\phi) \hat{d}_k$$

ファジィモデルでは、この項を省くこともある。

4.3 線形パラメータ時変(LPV)モデルと制御系設計

LMNモデルを真のシステムと信じていいのか？

- 大体的場合, そうであると信じ込む. うまくいく場合も多い.
- LMNモデルに対してシミュレーションを行うと, モデルの出力がまったく予想の付かない挙動をする可能性がある.
- (重み関数の影響で) LMNモデルをテーラー展開して導出した局所モデルが同定対象の局所モデルとまったく異なるものである可能性がある.
- 以下の条件が合えば, うまくいくだろう.
 - (a) システムの状態が滑らかに変化する.
 - (b) 領域分割が十分細かい.
 - (c) 隣同士の局所モデルのパラメータの値が近い.
 - (d) 重み関数が滑らかである.

4.3 線形パラメータ時変 (LPV)モデルと制御系設計

以下の微妙な違いに注意したほうがよい

- システム同定は普通予測モデルを同定している.
- (MBDによる)制御系の設計は普通シミュレーションモデルに基づいている？
- 実装された制御器は, 予測モデルを制御している？



うまくいけば, ラッキー.
うまくいかなければ, 上記の違いが原因かもしれない.

4.3 線形パラメータ時変 (LPV)モデルと制御系設計

LMNモデルの制御

スイッチング :

$$u(t) = u_k(t)$$
$$k = \arg \max_i \{w_i(\phi), i = 1, \dots, M\}$$

補間 :

$$u(t) = \sum_{k=1}^M w_k(\phi) u_k(t)$$

K. J. Hunt and T. A. Johansen, Design and analysis of gain-scheduled control using controller networks, *International Journal of Control*, 66-5, 619/651(1997).

B. A. Foss, T. A. Johansen and A. V. Storenson, Nonlinear predictive control using local models-applied to a batch fermentation process, *Control Engineering and Practice*, 3-3, 389/396 (1995).

K. Tanaka and H. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, Wiley-Interscience (2001)

4.3 線形パラメータ時変(LPV)モデルと制御系設計

見方を変えると: パラメータが状態変数によって変化する線形モデル

$$y(t) = \hat{A}(q^{-1}, \mathbf{w}(\phi))y(t) + \hat{B}(q^{-1}, \mathbf{w}(\phi))u(t) + \hat{D}(\mathbf{w}(\phi)) + e(t)$$

$$\hat{A}(q^{-1}, \mathbf{w}(\phi)) = \sum_{k=1}^M w_k(\phi) \hat{A}_k(q^{-1}) = \sum_{i=1}^{n_y} \left(\sum_{k=1}^M w_k(\phi) \hat{a}_{k,i} \right) q^{-i}$$

$$\hat{B}(q^{-1}, \mathbf{w}(\phi)) = \sum_{k=1}^M w_k(\phi) \hat{B}_k(q^{-1}) = \sum_{i=1}^{n_u} \left(\sum_{k=1}^M w_k(\phi) \hat{b}_{k,i} \right) q^{-i}$$

$$\hat{A}_k(q^{-1}) = \sum_{i=1}^{n_y} \hat{a}_{k,i} q^{-i}, \quad \hat{B}_k(q^{-1}) = \sum_{i=1}^{n_u} \hat{b}_{k,i} q^{-i}$$

$$\hat{D}(\mathbf{w}(\phi)) = \sum_{k=1}^M w_k(\phi) \hat{d}_k$$

状態に依存する一つのパラメータとみなす。

4.3 線形パラメータ時変(LPV)モデルと制御系設計

より一般化すると: 状態変数依存モデル

この項を省くこともある.

$$y(t) = \hat{A}(q^{-1}, \phi)y(t) + \hat{B}(q^{-1}, \phi)u(t) + \hat{D}(\phi) + e(t)$$
$$\hat{A}(q^{-1}, \phi) = \sum_{i=1}^{n_y} \hat{a}_i(\phi)q^{-i}, \quad \hat{B}(q^{-1}, \phi) = \sum_{i=1}^{n_u} \hat{b}_i(\phi)q^{-i}$$

各パラメータはまた前述のシグモイド
NN,RBF,FBF,多項式などで近似できる.

Jinglu Hu, [Research on Hybrid Black-Box Modeling for Nonlinear systems and Its Applications](#), Doctoral Dissertation, Kyushu Institute of Technology (1997)
<http://www.hflab.ips.waseda.ac.jp/~jinglu/publication.html>

4.3 線形パラメータ時変 (LPV)モデルと制御系設計

時変パラメータに注目すると, Linear Parameter-Varying (LPV)モデル

$$y(t) = \hat{A}(q^{-1}, \phi)y(t) + \hat{B}(q^{-1}, \phi)u(t) + \hat{D}(\phi) + e(t)$$

$$\hat{A}(q^{-1}, \phi(t)) = \sum_{i=1}^{n_y} \hat{a}_i(\phi(t))q^{-i}, \quad \hat{B}(\phi(t)) = \sum_{i=1}^{n_u} \hat{b}_i(\phi(t))q^{-i}$$

LPV モデルでは, パラメータの範囲と変化速度の上限を見積もることができると, ゲインスケジューリング (Gain Scheduling) 制御系の安定性解析に役立つ.

$$\underline{a}_i \leq \hat{a}_i(\phi(t)) \leq \bar{a}_i$$

$$\underline{b}_i \leq \hat{b}_i(\phi(t)) \leq \bar{b}_i$$

$$|\hat{a}_i(\phi(t)) - \hat{a}_i(\phi(t-1))| \leq \bar{v}_{ai}$$

$$|\hat{b}_i(\phi(t)) - \hat{b}_i(\phi(t-1))| \leq \bar{v}_{bi}$$

しかし, ゲインスケジューリング制御系の設計は, 連続時間モデルに基づくものが多い.

4.4 応用事例(電気通信大中野先生と新居浜高専豊田先生による)

豊田 幸裕, 尾崎 統, 中野 和司, Valerie Haggan-Ozaki, コンバインドサイクル発電所におけるNO_x低減用SCR制御システムへの非線形モデルベース予測制御の適用, 計測と制御 第42巻 第1号 (2003)

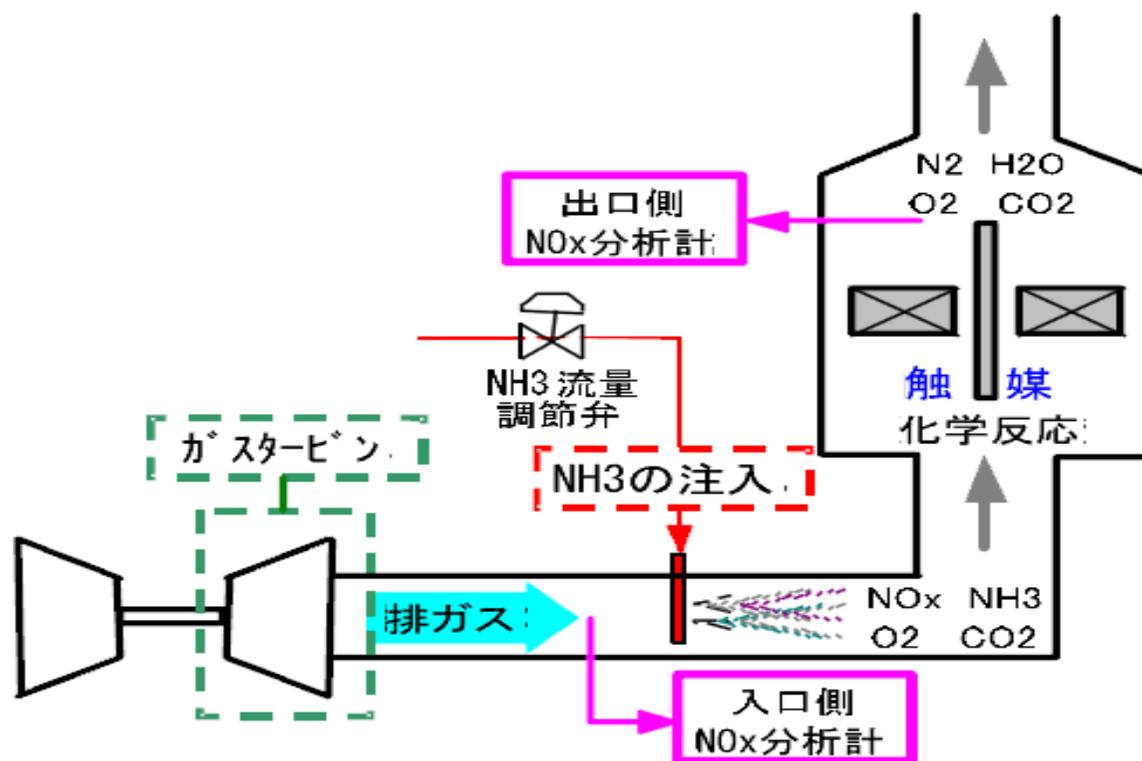


図 1 SCR システムの概要 : SCR(Selective Catalytic Reduction)システムは、NO_x を含む排ガスにアンモニア (NH₃) を注入し、脱硝触媒層を通過させて NO_x を窒素と水とに分解するもので、脱硝反応は図中の化学反応式で表わされる。

4.4 応用事例

RBF-ARX モデルの導入

$$y(t) = \Phi_0(W(t-1)) + \sum_{i=1}^{n_y} \Phi_{y,i}(W(t-1)) \cdot y(t-i) \\ + \sum_{i=1}^{n_u} \Phi_{u,i}(W(t-1)) \cdot u_{pid}(t-i) \\ + \sum_{i=1}^{n_v} \Phi_{v,i}(W(t-1)) \cdot v(t-i) + \xi(t) \quad (1)$$

$$\Phi_0(W(t-1)) = C_0^0 + \sum_{k=1}^m C_k^0 \exp\left\{-\lambda_k^0 \|W(t-1) - Z_k^0\|^2\right\} \\ \Phi_{j,i}(W(t-1)) = C_{j,0}^j + \sum_{k=1}^{n_j} C_{j,k}^j \exp\left\{-\lambda_k^j \|W(t-1) - Z_k^j\|^2\right\} \quad (2)$$

状態依存パラメータ

$$j = y, u, v$$

(1)式中の右辺第一項(オフセット項)は時間的に変化する局所平均値であり、動作点 $w(t-1)$ に依存している。ここで、 $y(t)$ 、 $u(t)$ 、 $v(t)$ 、 $\xi(t)$ は、それぞれ出力、入力、既知の外乱、平均値ゼロの白色雑音である。

4.4 応用事例

パラメータ推定

(1)、(2)式中の各パラメータ

$$C_0^0, C_{i,k}^j, \lambda_k^j, z_k^j, k = 1, 2, \dots, n_j, \dots, m : j = y, u, v$$

線形

非線形

非線形部の未知パラメータを推定するときは、線形部の既知パラメータを固定として求める。

線形部の未知パラメータを推定するときは、非線形部の既知パラメータを固定として求める。

交互に繰り返す。

全ての未知パラメータを非線形最適化計算でいきなりに求める場合に比して1/5程度の少ない繰り返し回数で、収束解が得られた。

4.4 応用事例

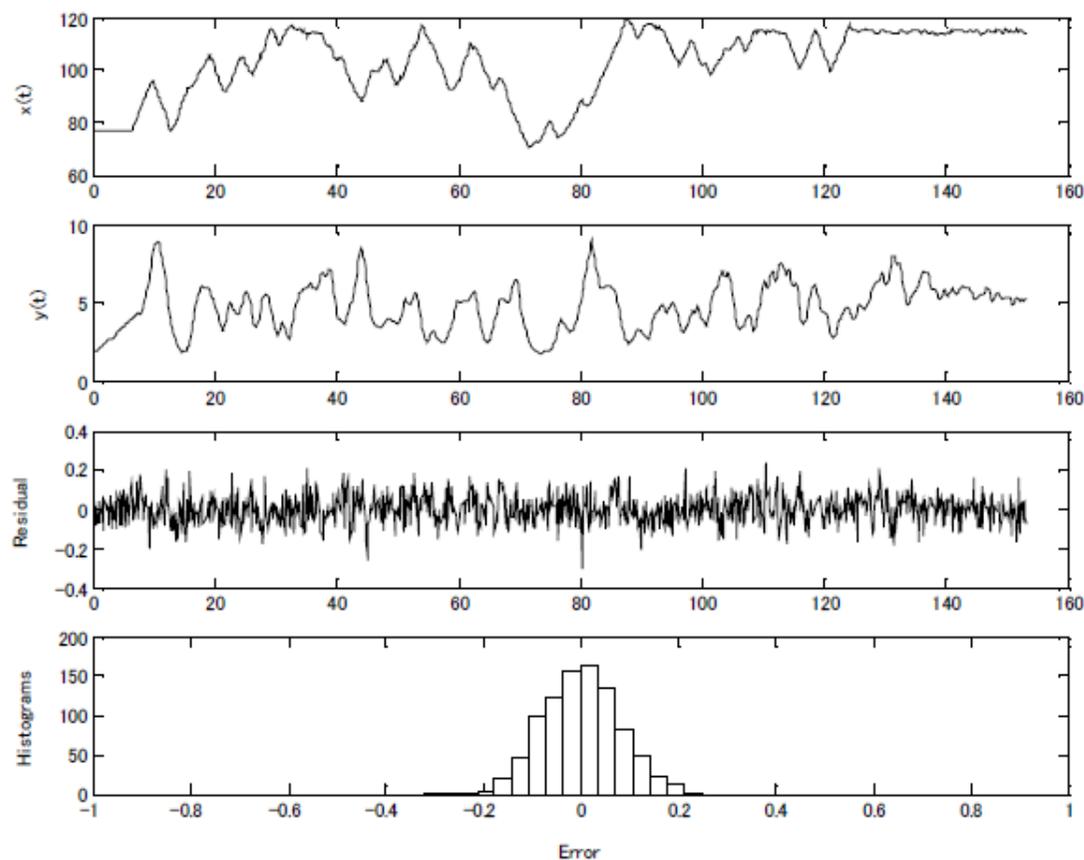


図3 負荷(MWD)変化時のNO_x予測：75MWから115MWの範囲を変動する運転条件下のプロセス変数の時系列を用いたNO_xの予測と残差および残差のヒストグラムを示している。図より良好な同定精度が得られたことが確認できる。

4.4 応用事例

RBF-ARX モデルベース予測制御(NPC)

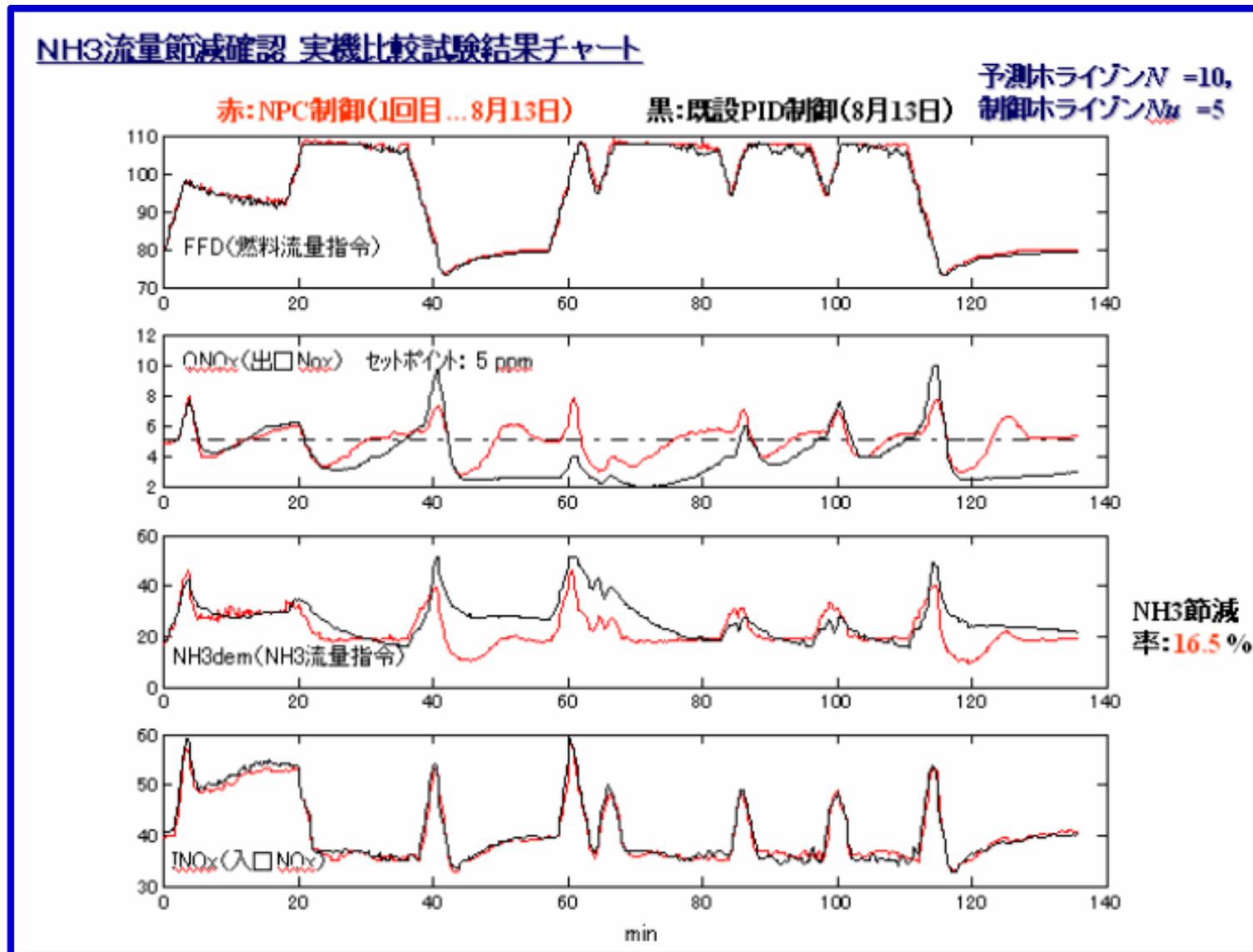
$$J = \sum_{i=N_1}^{N_p} \left\| \hat{y}(k+i|k) - y_r(k+i|k) \right\|_Q^2 + \sum_{i=0}^{N_u} \left\| \hat{u}(k+i|k) \right\|_R^2 \quad (3)$$

$\hat{u}(k+i|k)$ は時刻 k で仮定される入力の時刻 $k+i$ での未来値を表わし、

$\hat{y}(k+i|k)$ は $\hat{u}(k+j|k)$, ($j = 0, 1, 2, \dots, i-1$) が発生するという仮定の下での、 y の時刻 $k+i$ における予測値であり、これらの予測には、RBF-ARX モデルを用いる。

4.4 応用事例

制御性能の向上



(5) NFIRモデルの同定

(5)NFIRモデルの同定

5.1 線形FIR (Finite Impulse Response)モデルの復習

5.2 ボルテラモデルの導入

5.3 正規直交関数フィルタの利用

化学プロセス, 河川, 生体システム, 通信チャンネルなど:
無駄時間や入出力の最大遅れ次数などが未知で, しかも動作範囲などによって変化. インパルス応答モデルはこのような構造指数が不要.

5.1 線形FIRモデルの復習

線形Finite Impulse Response モデルとは？

連続時間線形システムの伝達関数を $G(s)$ とすると、その逆ラプラス変換 $g(t)$ はインパルス応答であり、入出力関係は

$$x(t) = \int_0^t g(\tau)u(t - \tau)d\tau$$

離散化すると（以下では、 t は整数）

$$x(t) = \sum_{k=1}^N h(k)u(t - k)$$

N はインパルス応答の有効な長さに依存、膨大な数になる可能性がある。

5.1 線形FIRモデルの復習

正規直交関数フィルタの利用

B. Wahlberg, System identification using Laguerre models, IEEE Trans. Automatic Control, 36-5, 551/562 (1991)

V. Nalbantoglu, J. Bokor, G. Balas and P. Gaspar, System identification with generalized orthonormal basis functions: an application to flexible structures, Control Engineering Practice, 11-3, 245/259 (2003)

Laguerre Filter:

$$L_i(q^{-1}) = \frac{\sqrt{1-\alpha^2}}{1-\alpha q^{-1}} \left(\frac{q^{-1}-\alpha}{1-\alpha q^{-1}} \right)^{i-1}$$

α をシステムの代表的時定数に近いように選ぶか, 最適化で計算するか.

直交関数フィルタ :

$L_1(q^{-1}), L_2(q^{-1}), \dots, L_L(q^{-1})$ のインパルス応答は正規直交関数系をなす.

顕著な振動モードがある場合, Laguerre Filter を複素極の場合に拡張できる: Kautz Filter

5.1 線形FIRモデルの復習

元の線形 FIR モデル

$$x(t) = \sum_{k=1}^N h(k)u(t-k)$$

パラメータ推定はしやすい

は以下となる.

$$x(t) = \sum_{i=1}^L \theta_i [L_i(q^{-1})u(t)]$$

利点：

$u(t-1), u(t-2), \dots, u(t-N)$ よりは, $L_1(q^{-1})u(t), L_2(q^{-1})u(t), \dots, L_L(q^{-1})u(t)$ のほうが信号同士の独立性が強い.

L のほうが N よりはるかに小さくできる.

$L_1(q^{-1}), L_2(q^{-1}), \dots, L_L(q^{-1})$ のインパルス応答を $b_1(t), b_2(t), \dots, b_L(t)$ とすると, 線形モデルのインパルス応答は正規直交基底関数によって近似される.

$$h(t) = \sum_{i=1}^L \theta_i b_i(t)$$

別の解釈

5.2 ボルテラモデルの導入

q 次の離散時間ボルテラモデル :

$$\begin{aligned}\hat{g}(\phi_u(t), \theta) = & h_0 + \sum_{i_1=1}^{N_u} h_1(i_1)u(t - i_1) \\ & + \sum_{i_1=1}^{N_u} \sum_{i_2=1}^{N_u} h_2(i_1, i_2)u(t - i_1)u(t - i_2) + \dots \\ & + \sum_{i_1=1}^{N_u} \dots \sum_{i_q=1}^{N_u} h_q(i_1, \dots, i_q)u(t - i_1) \dots u(t - i_q)\end{aligned}$$

NFIRモデル

ここまでは線形!

$\phi_u(t) = [u(t-1), \dots, u(t-N_u)]^t$: 回帰ベクトル.

N_u : システムの有限な記憶長.

$h_i(i_1, \dots, i_i)$: i 次の離散値ボルテラ核.

対称性に注意 : $h_2(i, j) = h_2(j, i)$.

$$y(t) = g_0(\phi(t)) + e(t)$$

NARXモデルと見比べよ!

5.2 ボルテラモデルの導入

利点：

- ♠ 予測モデルとシミュレーションモデルが同一．安定性の心配はない．
- ♠ 雑音が有色であっても，回帰ベクトル $\phi_u(t)$ と無相関．パラメータの最小二乗推定は，雑音による偏りが無い．

問題点： パラメータの数が非常に多い．

だからあまり使われていない

5.2 ボルテラモデルの導入

ブラックボックスモデルの立場で考えると、ボルテラモデルは、三層 NN モデル

$$\hat{g}(\phi_u(t), \theta) = \sum_{k=1}^M \alpha_k g_k(\phi_u)$$

の多項式展開近似. $\hat{g}(\phi_u(t), \theta)$ にシグモイド関数やローカル基底関数, LMN モデルなども適用可能.

●
●
○

どんなに頑張っても、パラメータの数が多い！

5.3 正規直交関数フィルタの利用

正規直交関数フィルタの利用

ボルテラ核を正規直交関数で近似：

$$\begin{aligned}h_1(i_1) &= \sum_{j=1}^L \theta_{j_1} b_j(i_1) \\h_2(i_1, i_2) &= \sum_{j_1=1}^L \sum_{j_2=1}^L \theta_{j_1 j_2} b_{j_1}(i_1) b_{j_2}(i_2) \\&\vdots \\h_q(i_1, \dots, i_q) &= \sum_{j_1=1}^L \cdots \sum_{j_q=1}^L \theta_{j_1 \dots j_q} b_{j_1}(i_1) \cdots b_{j_q}(i_q)\end{aligned}$$

$b_1(t), \dots, b_L(t)$: 正規直交基底関数 (Lauerre, Kautz など) .

$B_j(q^{-1})$ は $b_j(i)$ をインパルス応答とする線形フィルタ, フィルタの出力：

$$v_j(t) = B_j(q^{-1})u(t)$$

5.3 正規直交関数フィルタの利用

ボルテラモデルは以下のように書き直せる：

$$\begin{aligned}\hat{g}(\phi_u(t), \boldsymbol{\theta}) &= h_0 + \sum_{j_1=1}^L \theta_{j_1} v_{j_1}(t) + \sum_{j_1=1}^L \sum_{j_2=1}^L \theta_{j_1 j_2} v_{j_1}(t) v_{j_2}(t) + \cdots \\ &+ \sum_{j_1=1}^L \cdots \sum_{j_q=1}^L \theta_{j_1 \dots j_q} v_{j_1}(t) \cdots v_{j_q}(t)\end{aligned}$$

注意：パラメータ推定の際， $\theta_{j_1 j_2} v_{j_1}(t) v_{j_2}(t)$ と $\theta_{j_2 j_1} v_{j_2}(t) v_{j_1}(t)$ を合併する。
形式上，以下になる。

$$\begin{aligned}\hat{g}(\phi_u(t), \boldsymbol{\theta}) &= h_0 + \sum_{j_1=1}^L \theta_{j_1} v_{j_1}(t) + \sum_{j_1=1}^L \sum_{j_2=j_1}^L \theta_{j_1 j_2} v_{j_1}(t) v_{j_2}(t) + \cdots \\ &+ \sum_{j_1=1}^L \cdots \sum_{j_q=j_{q-1}}^L \theta_{j_1 \dots j_q} v_{j_1}(t) \cdots v_{j_q}(t)\end{aligned}$$

V. Z. Marmarelis and X. Zhao: Volterra models and three-layer perceptrons, IEEE Trans. Neural Networks, 8-6, 1421/1433(1997).

5.3 正規直交関数フィルタの利用

ブラックボックスモデルの立場で考えると、前式は $\phi_v(t)$ を入力とする三層 NN

$$\hat{g}(\phi_v(t), \theta) = \sum_{k=1}^M \alpha_k g_k(\phi_v)$$

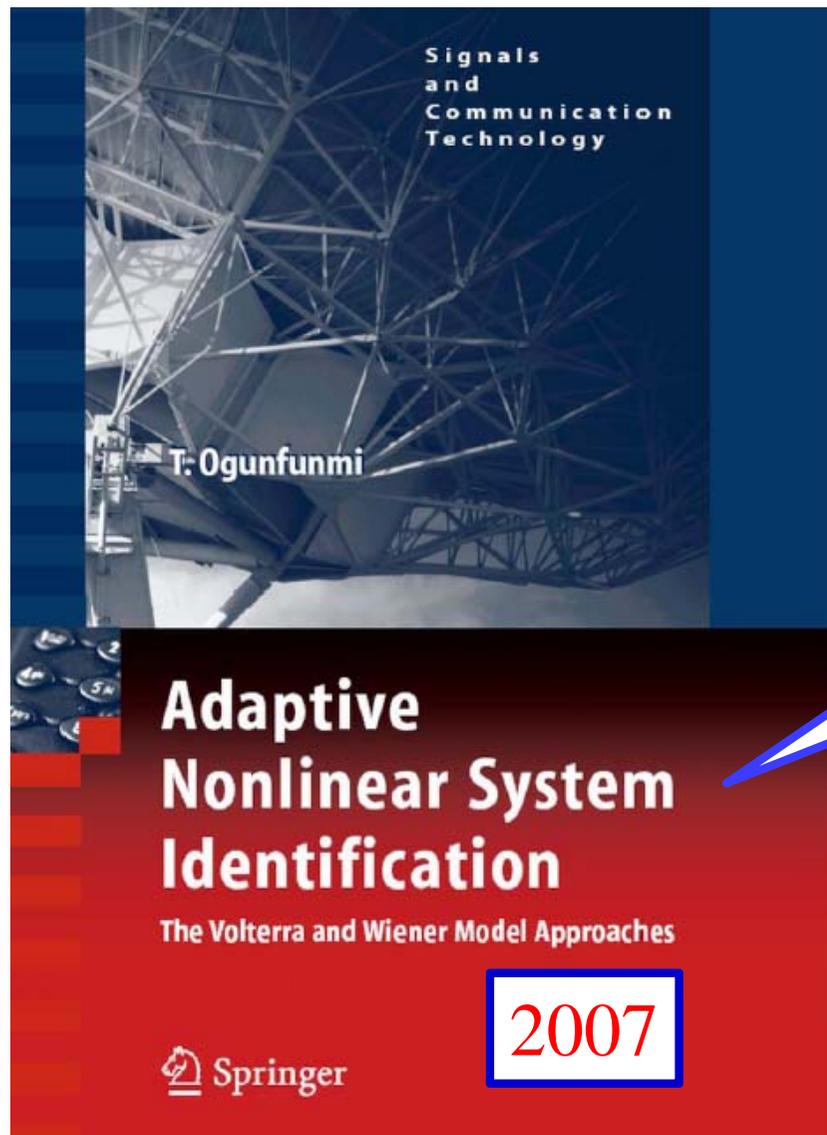
の多項式展開近似.

$\phi_v(t) = [v_1(t), \dots, v_L(t)]^T$: 線形フィルタの出力による回帰ベクトル.

シグモイド関数やローカル基底関数, LMN モデルなども適用可能.

D. Sbarbaro, T.A. Johansen, Multiple local Laguerre models for modeling nonlinear dynamic systems of the Wiener class, IEE Proceedings on Control Theory and Applications, 144-5, 375/380 (1997)

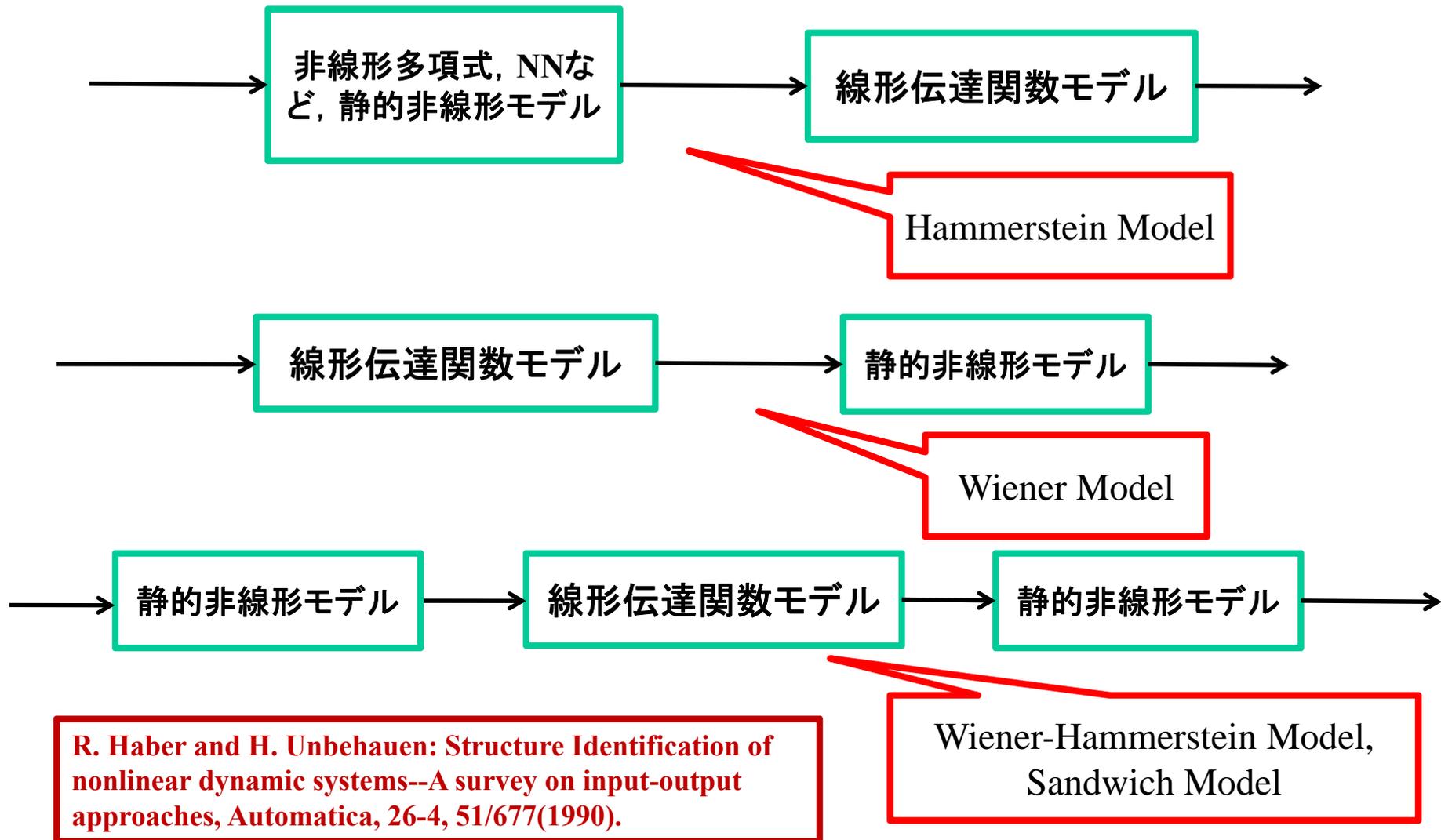
5.3 正規直交関数フィルタの利用



NFIRモデルの同定は
デジタル通信の分野でも注目されている。

(6) ブロック指向モデルの同定

(6) ブロック指向モデルの同定



(6) ブロック指向モデルの同定

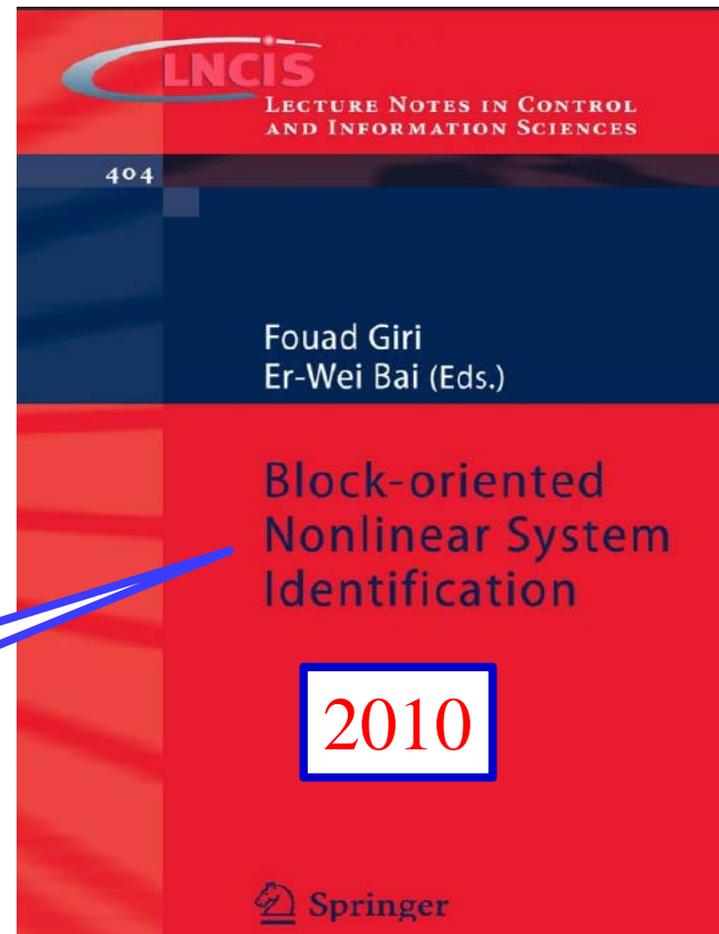
もっとも単純な同定手法:

(1) 非線形な部分を多項式やRBFなどで近似し、線形の部分を伝達関数モデルでパラメータ化する。

(2) 最適化手法でパラメータを推定する。

一部のパラメータに関して線形、一部のパラメータに関して非線形という特徴を利用する。

J. Voros, Iterative algorithm for parameter identification of Hammerstein Systems with two-segment nonlinearities, IEEE Transactions on Automatic Control, 44-11, 2145/2149 (1999)

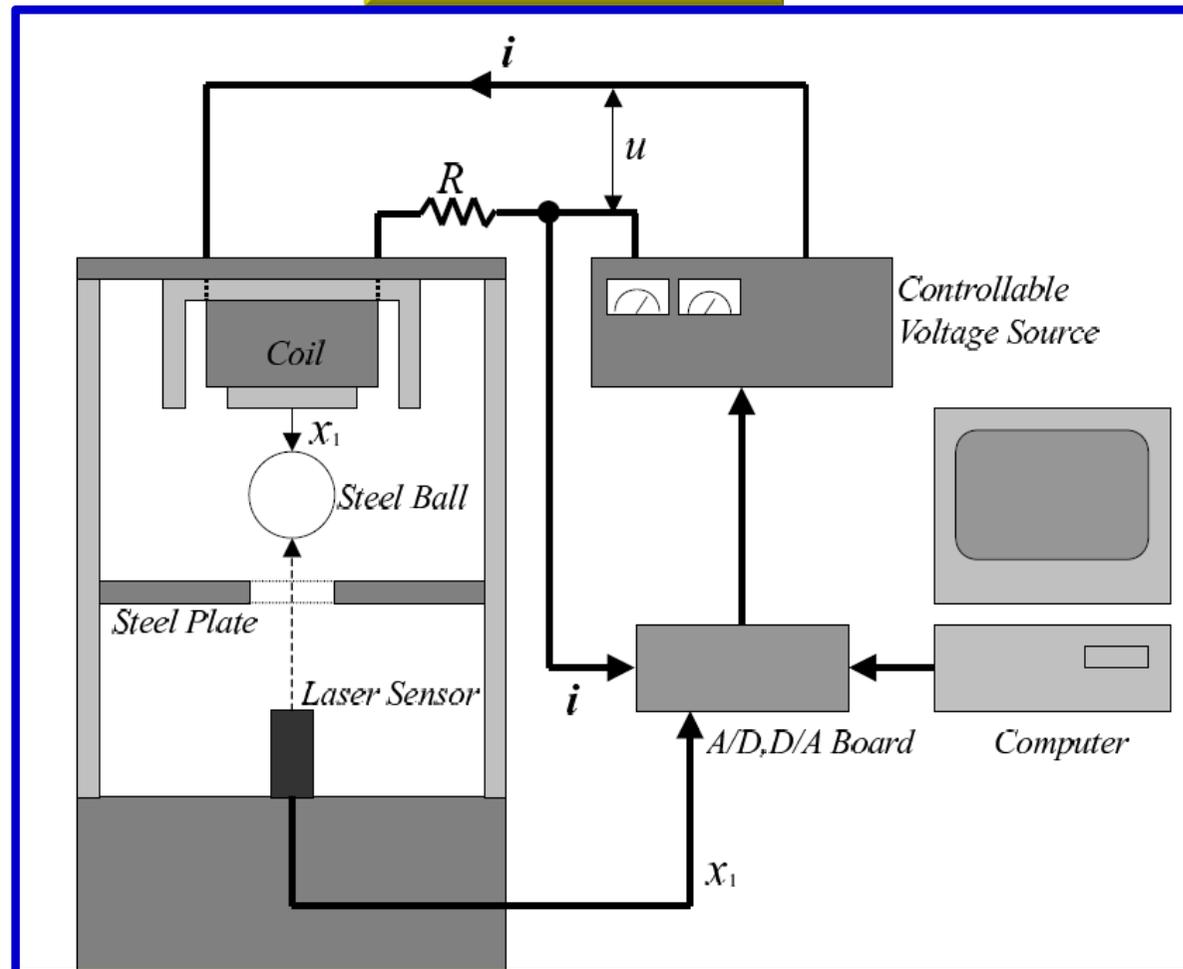


最新の研究

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

(7) 非線形メカニカルシステムのパラメータ推定

応用事例



(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

機械的運動方程式

$$\dot{x}_1 = \dot{x}_2$$

$$\dot{x}_2 = -\frac{Qx_3^2}{2M(\underline{X}_\infty + x_1)^2} + g$$

$$\dot{x}_3 = \frac{x_3\{Qx_2 - R(\underline{X}_\infty + x_1)^2\}}{Q(\underline{X}_\infty + x_1) + L_\infty(\underline{X}_\infty + x_1)^2} + \frac{\underline{X}_\infty + x_1}{Q + L_\infty(\underline{X}_\infty + x_1)}u$$

電気回路の方程式

x_1 は位置, x_2 は速度, x_3 は電流, u は電圧入力

m は鋼球の質量 (既知), g は重力加速度 (既知)

同定は難しそう。。。

Q, X_∞, L_∞, R は電磁石あるいは電気回路の未知なパラメータ。

システム同定の教科書は直接利用できないが、考え方を活用できる。

最小二乗法で同定しやすいデータを作ることがポイント。

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

(貧乏な大学研究室での)

モデルベース制御系設計

- 適当なパラメータ公称値でロバストな制御器を設計(制御だけが目的なら、厳密な同定しなくても何とかなる)。
- 設計した制御器で同定しやすいデータを作成。
- パラメータ推定
- シミュレーション結果と実験結果の一致を確認。
- 同定できたパラメータを使って制御器を更新し、シミュレーションで制御性能の改善を確認(MBD?)。
- 必要があれば、実機実験を実行。

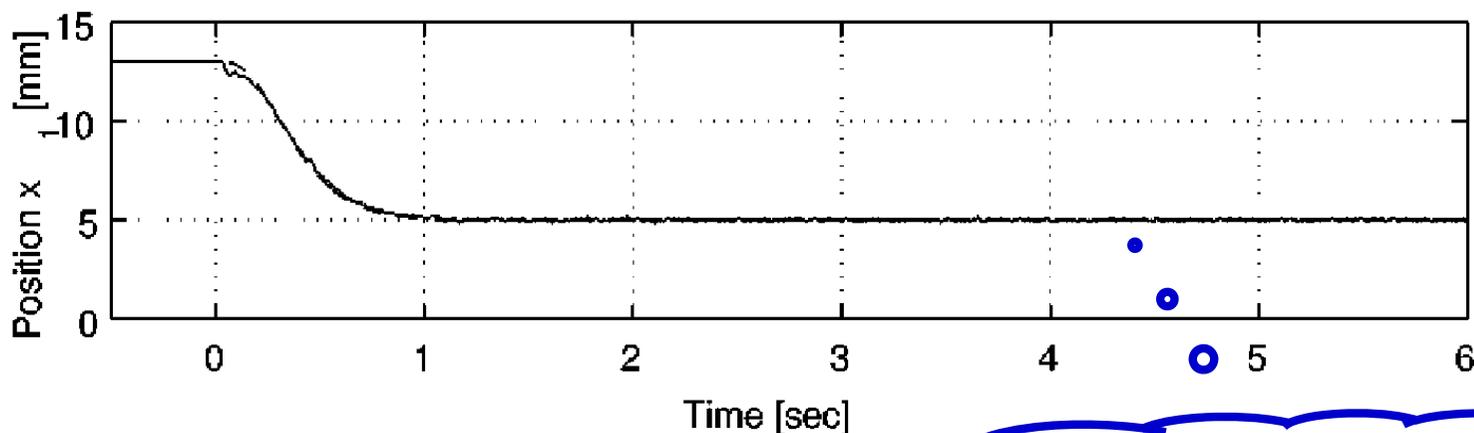
(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

機械運動方程式の同定

$$\dot{x}_2 = -\frac{Qx_3^2}{2M(\underline{X}_\infty + x_1)^2} + g$$

えっ、非線形？

しかし、鋼球が動かなければ、 \dot{x}_2 はゼロになる。



一定の位置に制御

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

$$2Mg(\underline{X_\infty} + x_1)^2 = \underline{Q}x_3^2$$

平方根を施して，整理すると，

$$x_1 = \underline{\sqrt{Q}} \frac{x_3}{\sqrt{2Mg}} - \underline{X_\infty}$$

$$x_1 = \psi^T \theta$$

$$\psi^T = \left[\frac{x_3}{\sqrt{2Mg}}, -1 \right]$$

$$\theta = \left[\sqrt{Q}, X_\infty \right]^T$$

線形最小二乗法が使える！

鋼球位置 x_1 を12.5, 12.0, ..., 3.5, 3.0[mm]の位置に固定して，電流値 x_3 も計測すれば，パラメータは線形最小二乗法で推定できる。

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

電気回路方程式の同定

$$\frac{Qx_2x_3 + (X_\infty + x_1)^2u}{X_\infty + x_1} - Q\dot{x}_3 = \underline{L_\infty}(X_\infty + x_1)\dot{x}_3 + \underline{R}(X_\infty + x_1)x_3$$

既知

$$\eta_2(t) = \eta_1^T(t)\theta$$

$$\theta = [L_\infty, R]^T$$

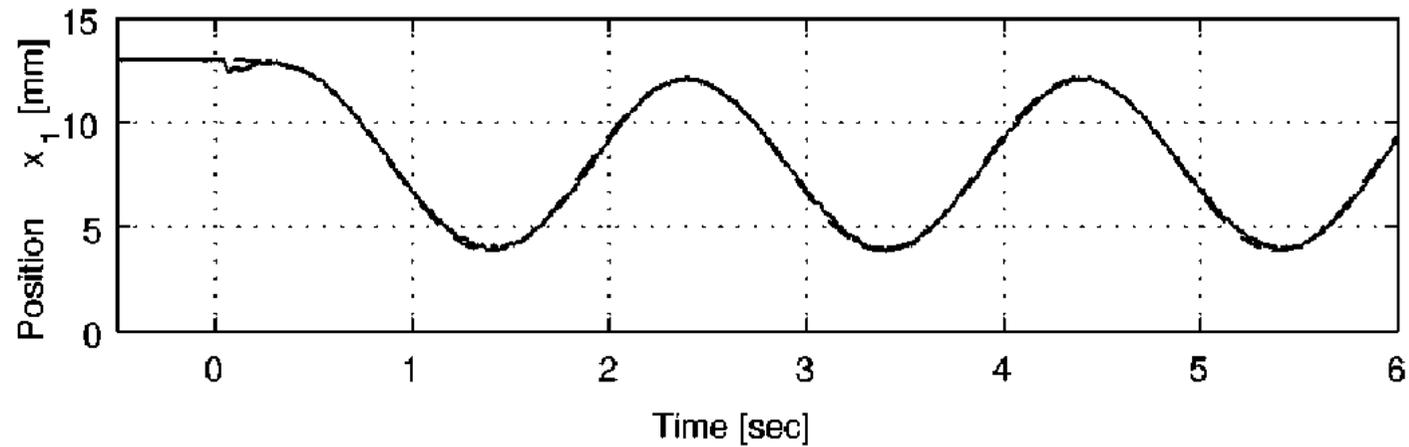
$$\eta_1^T(t) = [(X_\infty + x_1)\dot{x}_3, (X_\infty + x_1)x_3]$$

微分信号があるので、ローパスフィルタで処理:

$$\tilde{\eta}_1^T(t) = \frac{\eta_1^T(t)}{(\lambda s + 1)^2}, \quad \tilde{\eta}_2(t) = \frac{\eta_2(t)}{(\lambda s + 1)^2}$$

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

微分信号があるので, “動く信号データ”を作成する:



同定しやすいデータを作成するために, 上手に制御できたらいいなあ。

(7) 非線形メカカルシステムのパラメータ推定(応用事例)

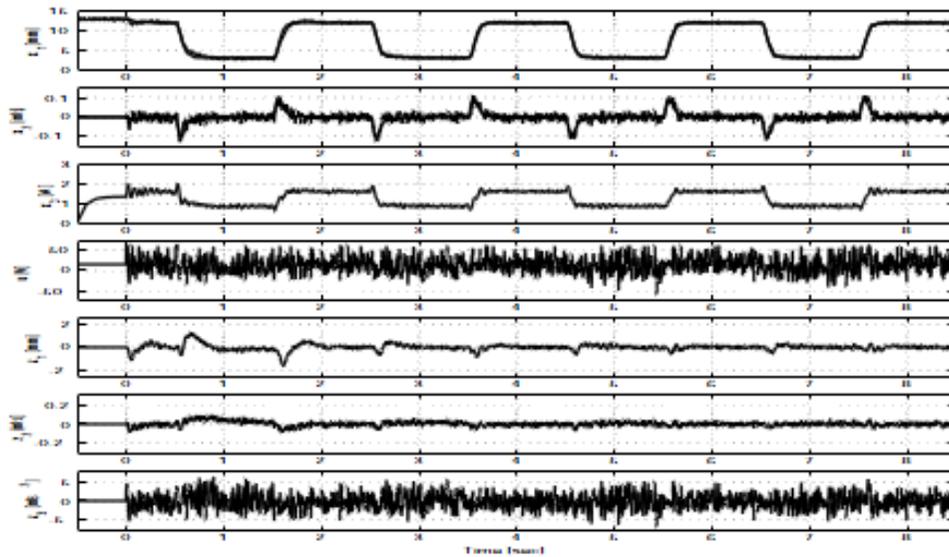


Figure 6: Experiment results of an adaptive robust nonlinear controller.

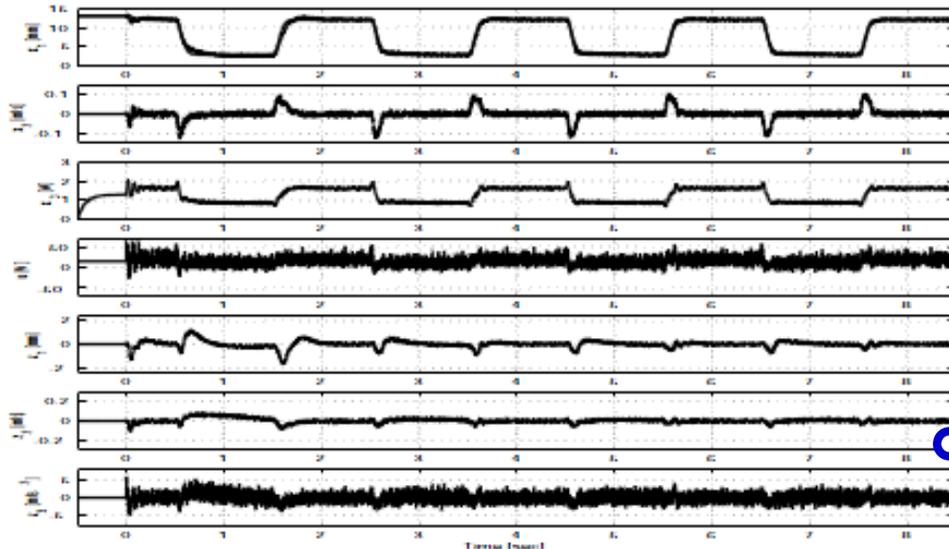


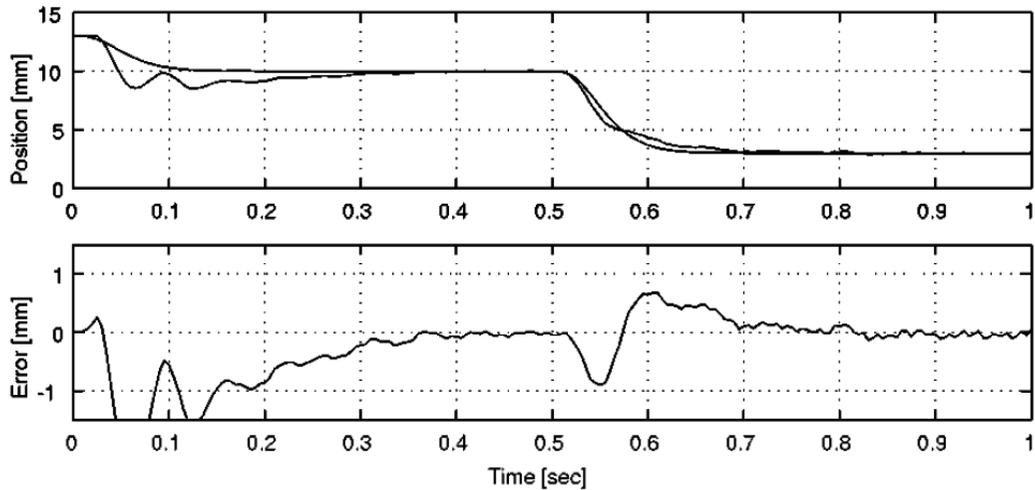
Figure 7: Simulation results of an adaptive robust nonlinear controller.

適応制御の実験
結果

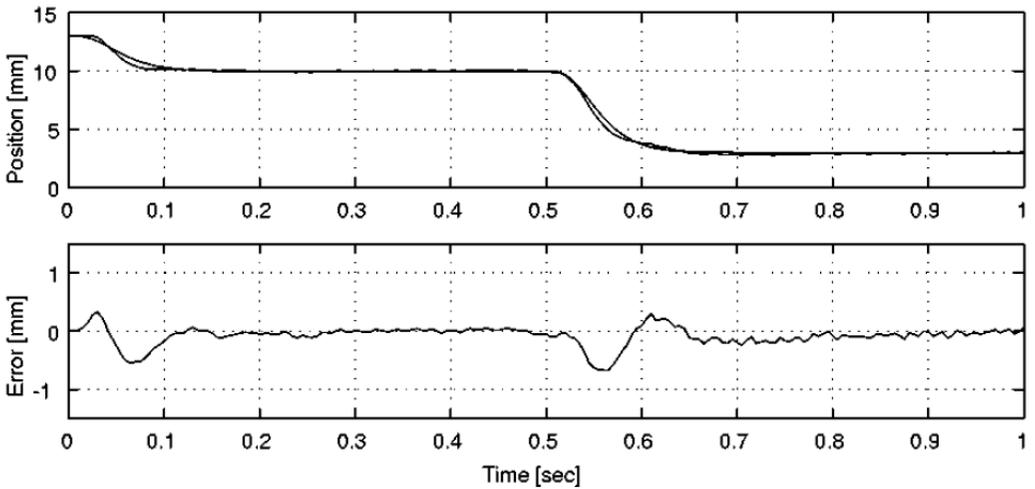
シミュレーション
でも同じ結果

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

(適応なし) ロバスト制御の性能向上の確認



同定の前



同定の後

(7) 非線形メカニカルシステムのパラメータ推定(応用事例)

感想

- **制御対象の特徴, システム同定, 制御系設計**という三つの事柄をすべて熟知しないと, 上手くいかない。
- 制御器を利用して, 同定しやすいデータを得るとよい。
- モデルベース制御系設計のために, 同定モデルが必要; 同定しやすいデータを得るために上手な制御が必要 (**ニワトリと卵はどちらが先?**)

(8) 非線形の適応制御

(8) 非線形システムの適応制御

一入出力システム :

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

⋮

$$\dot{x}_n = f(\mathbf{x}) + g(\mathbf{x})u$$

$$y = x_1$$

$\mathbf{x} = [x_1, x_2, \dots, x_n]^T = [x, \dot{x}, \dots, x^{(n-1)}]^T$: 測定可能な状態変数ベクトル.

$y = x_1$: システム出力.

$f(\mathbf{x}), g(\mathbf{x})$: 未知な連続関数, $g(\mathbf{x}) > \underline{g} > 0$ とする.

Jay A. Farrell, Marios M. Polycarpou:

ADAPTIVE APPROXIMATION BASED CONTROL--Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches, Wiley-Interscience (2006)

(8) 非線形の適応制御

n 階連続微分可能な目標値 y_m , 制御誤差 $e = y_m - y$.

線形化制御入力 u :

$$u = \frac{1}{g(\boldsymbol{x})} [-f(\boldsymbol{x}) + y_m^{(n)} + \boldsymbol{k}^T \boldsymbol{e}]$$

$$\boldsymbol{k} = [k_n, \dots, k_1]^T, \quad \boldsymbol{e} = [e, \dot{e}, \dots, e^{(n-1)}]^T$$

ただし, 線形化された誤差システム

$$e^{(n)} + k_1 e^{(n-1)} + \dots + k_n e = 0$$

が安定であるとする。

問題点 : $f(\boldsymbol{x}), g(\boldsymbol{x})$ は未知.

(8) 非線形の適応制御

あるコンパクトな領域 $X \subset R^n$ において, 十分小さな ε に対して,

$$|\hat{f}(x, \theta_f) - f(x)| \leq \varepsilon, \quad |\hat{g}(x, \theta_g) - g(x)| \leq \varepsilon$$

を満足する θ_f^* , θ_g^* が存在するように **NN** を設計する.

適応制御入力 u :

$$u = u_c + u_s$$

適応線形化フィードバック制御項 :

$$u_c = \hat{g}(x, \hat{\theta}_g)^{-1} [-\hat{f}(x, \hat{\theta}_f) + y_m^{(n)} + \mathbf{k}^T \mathbf{e}]$$

モデリング誤差に対処するロバスト制御項 : u_s .

適応則によるパラメータのオンライン推定値 : $\hat{\theta}_f, \hat{\theta}_g$.

(8) 非線形の適応制御

適応ロバスト非線形制御系の設計のキーポイント：

- (1) ネットワークの設計（線形パラメータが望ましい）.
- (2) ロバスト適応則の設計（推定パラメータが発散しないように、及び $\hat{g}(x, \hat{\theta}_g)$ がゼロにならないように）.
- (3) ロバスト制御項 u_s の設計（スライディングモード制御などが代表的）.

発展的研究：

- ♠ 制御対象のクラスの拡大（座標変換や、バックステッピング設計との関連、多入出力システムなど）；
- ♠ 出力フィードバック制御など.

(8)まとめ

(9)まとめ

- 非線形システムの同定は，多様性に富む．どれが一番よいのか，結論付けが困難．
- 類似した手法でも，違う名称で宣伝されることがある．
- ヨーロッパと比較して，日本では大学における応用研究が少ない？
- 現場を知らないと，ブラックボックスモデル同定は無理．
- 予測モデル，シミュレーションモデル，実システムの違いを意識すべき．
- 機械学習や統計学分野との交流も重要．