

長岡技術科学大学大学院  
工学研究科修士論文

題目  
交差項を考慮した一般化スプライン  
非線形適応フィルタに関する研究

指導教員  
杉田 泰則 准教授

著者  
16315690 斎藤 栄二

提出期日  
令和 2 年 2 月 7 日

# ABSTRACT

## A study on generalized spline nonlinear adaptive filter with cross terms

Author : Shuji Saitou

Supervisor : Yasunori Sugita

Estimating the mathematical model of unknown systems based on input-output observation using adaptive filters is very important for applications on signal processing technology such as echo cancellers and noise control. A nonlinear adaptive filter is used to accurately perform estimating when the unknown system has nonlinear characteristics. Adaptive Volterra filters are most well-used which are capable of modeling nonlinear characteristics with very high accuracy. However, the amount of computation increases exponentially when modeling complex nonlinear characteristics. As other known adaptive filter, there is the Functional Link Artificial Neural Network (FLANN) which expands the input elements by using some function like trigonometric functions. The computational complexity of FLANN is lower than that of the adaptive Volterra filter, but its modeling performance is inferior.

Generalized spline nonlinear adaptive filter (GSNAF) that expands the input as a function using an adaptive spline function (ASF) has been proposed as a new nonlinear adaptive filter. As GSNAF is calculated using only some of the coefficients according to the input, the amount of calculation is small. It is also excellent in modeling performance and convergence speed, because frequently chosen coefficients are updated multiple times.

However, the GSNAF has a problem that it cannot express nonlinear characteristics with cross terms because function expansion is performed for each input element. This paper proposes a new nonlinear adaptive filter which is based on GSNAF. In the proposed method, ASF is modified to enable the representation of cross terms while maintaining the advantages of GSNAF. An ASF of GSNAF has one input and one output, while an ASF of the proposed method has two inputs and one output for each combination of input elements. So, the proposed method is possible to express a cross term. However, the proposed method has a very large number of ASFs due to its configuration. It might have many unnecessary ASFs for modeling. Furthermore, many unnecessary ASFs outputs deteriorate modeling performance. Therefore, we removed unnecessary ASFs by considering the output of each ASF

during modeling. Unnecessary ASFs are removed by considering the output of each ASF during modeling.

In the simulation, system identification was performed by using adaptive third-order Volterra, GSNAF, and the proposed method, and their Mean Squared Errors (MSE) were compared. As a result, the proposed method was able to identify nonlinear characteristics with cross terms. Furthermore, it was confirmed that the proposed method is more accurately than the adaptive third-order Volterra filter in the modeling of the system with complex nonlinear characteristics.

# 目次

第 1 章	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
1.3	論文構成	2
第 2 章	一般化スプライン非線形適応フィルタ	3
2.1	スプライン曲線を用いた補間	3
2.2	GSNAF の構造	5
2.3	問題点	8
第 3 章	提案手法	11
3.1	スプライン曲線を用いた補間による曲面表現	11
3.2	提案手法の構造	12
3.3	問題点	14
3.4	解決案	15
第 4 章	シミュレーション	16
4.1	シミュレーション 1	17
4.1.1	結果と考察	18
4.2	シミュレーション 2	19
4.2.1	結果と考察	20
4.3	シミュレーション 3	21
4.3.1	結果と考察	22
4.4	シミュレーション 4	23
4.4.1	結果と考察	23
第 5 章	終わりに	25

謝辞	26
参考文献	27
付録	29
A 提案手法の問題点と解決案の効果	29
B 4.2 のシミュレーション結果について	30

# 第1章

## はじめに

本章では、本論文の研究背景および目的、本論文の構成について述べる。

### 1.1 研究背景

適応フィルタにより入出力関係から未知システムを数学モデル化することはエコーチャンセラー [1] [2] や騒音制御 [3] 等の応用で重要である。その中で未知システムの入出力関係を有限インパルス応答 (FIR : Finite Impulse Response) で表す適応 FIR フィルタは良く知られる手法である。これは未知システム出力と適応 FIR フィルタ出力の誤差を何らかの評価基準に基づき最小化するようにフィルタ係数を更新する。良く知られた係数更新アルゴリズムとして、LMS(Least Means Square)、NLMS(Normalized Least Means Square)、RLS(Recursive Least Square) アルゴリズム等がある。しかし未知システムが非線形特性を含む場合、線形特性しか表現できない適応 FIR フィルタでは精度よくモデル化することは困難である。そこで、非線形特性をモデル化することができる非線形適応フィルタが用いられる。ボルテラ級数展開を利用した適応ボルテラ (Volterra) フィルタ [4] [5] は一般的な非線形適応フィルタである。これは非常に良く非線形特性をモデル化することが可能だが、複雑な非線形特性を表現するために非線形項の次数を増やす場合はその計算量が指数関数的に増加することが問題であり、実環境ではその性能は制限される。FLANN(Functional Link Artificial Neural Network) [6] も良く知られる非線形適応フィルタである。これは入力を三角関数等の関数で展開し、それぞれの信号に対して重みづけを行い、その総和を出力とする手法である。重みは Volterra と同様に LMS 等の更新アルゴリズムで更新される。この手法は適応 Volterra フィルタと比べて非線形項の次数を増やした場合でも計算量の増加が少ないが、非線形特性のモデル化能力が劣る。

近年は、上記の手法の他にも適応 FIR フィルタと適応スpline関数 (ASF : Adaptive Spline Function) から構成されるスpline適応フィルタ (SAF : Spline Active Filter) [7] [8] や複数の ASF を用いて構成した一般化スpline非線形適応フィルタ (GSNAF : Generalized spline nonlinear adaptive filters) [9] も提案されている。これらは前述した適応 Volterra,FLANN の 2 つの非線形適応フィルタと違い、各更新で全てのパラメータを更新せずに出力に寄与したパラメータのみが更新される。このため非線形項の次数を増やした場合でも計算量の増加が抑制される。しかし、その構成から交差項を含む未知システム出力の表現が困難という問題点がある。交差項を含む非線形は騒音制御等においてよく見られる特性である。

## 1.2 研究目的

ASF の構成を変更し、GSNAF の利点を継ぎながら交差項を表現することが可能な GSNAF の新しい非線形適応フィルタの提案を目的とする。本論文では、問題点である交差項の表現力を解決するために活性化関数の構成を変更した。具体的には 1 入力 1 出力であった ASF を 2 入力 1 出力に変更し、それぞれの組み合わせによる ASF で非線形適応フィルタを構成する。さらに、シミュレーションにより交差項を含む未知システムにおける適応 Volterra、GSNAF、提案手法の比較を行う。

## 1.3 論文構成

本論文の構成を下記に示す。第 2 章では GSNAF の説明を行い、次いで問題点を述べる。第 3 章では提案手法の説明とその手法における問題点、その解決法について述べる。第 4 章ではシミュレーションにより適応 Volterra、GSNAF、提案手法の 3 手法を比較し結果を示す。第 5 章では本論文の結論を述べる。

## 第 2 章

# 一般化スプライン非線形適応 フィルタ

本章では、既存手法である一般化スプライン非線形適応フィルタ (GSNAF)について述べる。まず 2.1 節では GSNAF の原理に関する知識としてスプライン曲線による補間について述べる。次に 2.2 節では GSNAF のブロック図や計算方法、LMS による更新方法について述べる。最後に 2.3 節では GSNAF の問題点として交差項の表現が困難である点について述べる。

### 2.1 スプライン曲線を用いた補間

この節では GSNAF の原理に関する知識としてスプライン曲線を用いた離散点の補間手法について述べる。与えられた離散点間を滑らかに接続するために GSNAF ではスプライン曲線を用いた補間が使われている。それは全体を一つの多項式で表現するのではなく、ある特定の与えられた区間のみの区分多項式を用いた補間手法である。今回は B スプライン曲線と Catmul-Rom(CR) スプライン曲線による補間について説明する。これらの 2 つの手法は後述する条件で同じ形式の数式で表すことができる。

x-y 平面上に  $\mathbf{Q}_i = [q_{x,i} \ q_{y,i}]^T$  とする離散点を考える。この時、 $i = 0, 1, \dots, Q$  である。また、 $q_{x,i}$  は式 (2.1) のような条件である。

$$q_{x,0} < q_{x,1} < q_{x,2} < \dots < q_{x,Q} \quad (2.1)$$

$u \in [q_{x,i} \ q_{x,i+1}]$  とした x 軸座標値について考える。スプライン曲線補間は式

(2.2) に示すような  $P$  次のスプライン基底関数のアフィン結合で表現される。

$$\varphi(u) = \sum_{i=0}^{Q-P-1} \mathbf{Q}_i N_i^P(u), \quad u \in [q_{x,n}, q_{x,Q-P}] \quad (2.2)$$

ここで、スプライン基底関数  $N_i^P(u)$  は式 (2.3)、(2.4) のように表される。

$$N_i^0(u) = \begin{cases} 1 & (q_{x,i} < u < q_{x,i+1}) \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

$$N_i^P(u) = \frac{u - q_{x,i}}{q_{x,i+P} - q_{x,i}} N_i^{P-1}(u) + \frac{q_{x,i+P+1} - u}{q_{x,i+P+1} - q_{x,i+1}} N_{i+1}^{P-1}(u) \quad (2.4)$$

例えば、スプライン補間次数  $P = 1$  とした場合は式 (2.5) のように表される。

$$N_i^1(u) = \begin{cases} \frac{u - q_{x,i}}{q_{x,i+1} - q_{x,i}} & (q_{x,i} < u < q_{x,i+1}) \\ \frac{q_{x,i+2} - u}{q_{x,i+2} - q_{x,i+1}} & (q_{x,i} < u < q_{x,i+1}) \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

式 (2.5) より  $N_i^1(u)$  は区間以外は 0 になる線形補間となっていることが分かる。つまり、 $N_i^1(u)$  は線形補間、 $N_i^2(u)$  は 2 次関数による補間、 $N_i^P(u)$  は  $P$  次の関数による補間となる。また、各区分で計算に使用される離散点はそれぞれ  $P + 1$  点である。これらの離散点を制御点と呼ぶ。一般的には  $N_i^P = N_i^0 * N_i^0 * \dots * N_i^0$  が成り立つ。ここで \* は畳み込み積分を示す。これらより、式 (2.2) は式 (2.6) のように簡単に書くことができる。

$$\varphi(u) = \varphi_i(u), \quad \forall i = 0, \dots, Q \text{ and } u \in [0, 1] \quad (2.6)$$

式 (2.6) 中の  $\varphi_i(u)$  は  $P$  次区分多項式を示している。

式 (2.4) を事前に計算することが可能ならば計算量を削減することができる。そこで、事前計算を可能とするために各制御点の x 軸座標間の距離を一定にする。つまり  $\Delta x = q_{x,i+1} - q_{x,i}$  とし、これにより式 (2.4) は式 (2.7) のように簡単に表すことができる。

$$\varphi_i(u) = \mathbf{u}^T \mathbf{C} \mathbf{q}_i \quad (2.7)$$

式 (2.7) の  $\mathbf{C}$  は事前計算されたスプライン基底関数を行列式としたものである。 $\mathbf{u} = [u^P \ u^{P-1} \ u^{P-2} \ \dots \ u \ 1]^T$  であり、 $u$  は 2 つの制御点間の正規化された x 軸座標値である。さらに  $\mathbf{q}_i = [q_i \ q_{i+1} \ q_{i+2} \ \dots \ q_{i+P}]^T$  である。この時  $q_i \equiv q_{y,i}$  である。例えば  $P = 2$  とした時、式 (2.4) は下式のように計算される

$$N_i^2(u) = \begin{cases} \frac{1}{2}(u - q_{x,i}) & (q_{x,i} < u < q_{x,i+1}) \\ \frac{1}{2} - (u - q_{x,i+1}) - (u - q_{x,i+1})^2 & (q_{x,i+1} < u < q_{x,i+2}) \\ \frac{1}{2}(1 - (u - q_{x,i+2}))^2 & (q_{x,i+2} < u < q_{x,i+3}) \end{cases} \quad (2.8)$$

式(2.8)を行列式に変形し、式(2.7)の形にすると式(2.9)が得られる。

$$\varphi_i(u) = [u^2 \ u \ 1] \frac{1}{2} \begin{pmatrix} 1 & -2 & 1 \\ -2 & -2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{bmatrix} q_i \\ q_{i+1} \\ q_{i+2} \end{bmatrix} \quad (2.9)$$

同様に  $P = 3$  として計算した場合、式(2.10)が得られる。

$$\varphi_i(u) = [u^3 \ u^2 \ u \ 1] \frac{1}{2} \mathbf{C} \begin{bmatrix} q_i \\ q_{i+1} \\ q_{i+2} \\ q_{i+3} \end{bmatrix} \quad (2.10)$$

また、ここで  $\mathbf{C}$  は式(2.11)である。

$$\mathbf{C} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ -1 & 4 & 1 & 0 \end{pmatrix} \quad (2.11)$$

Bスプライン曲線による補間は、与えられた制御点を必ずしも通過せずに各区分が滑らかになるように接続される。対して、CRスプライン曲線による補間は全ての制御点を通過する補間方法である。これもBスプライン曲線による補間と同様に、制御点間隔を一定とすれば行列式として表すことが出来る。 $P = 3$  の CRスプライン基底関数の行列式を式(2.12)に示す。

$$\mathbf{C} = \frac{1}{2} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix} \quad (2.12)$$

式(2.11)や式(2.12)をスプライン基底関数行列とよぶ。BスプラインとCRスプラインを比較すると、必ず全ての制御点を制御点を通過するか否かや各制御点付近に対して滑らか(微分関数が連続)などの違いがある。CRスプライン曲線はCAD等のコンピューティンググラフィックスにおいて図形を表示するために用いられている[10][11]。Bスプライン曲線は点群データのフィッティングに用いられている[12]。

## 2.2 GSNAFの構造

この節ではGSNAFの構造について述べる。GSNAFのブロック図を図2.1に示す。GSNAFは入力信号とそのタップ遅延した信号に対してPGUとASFの2種類のブロックが連結されている非線形適応フィルタである。入力とそれをタッ

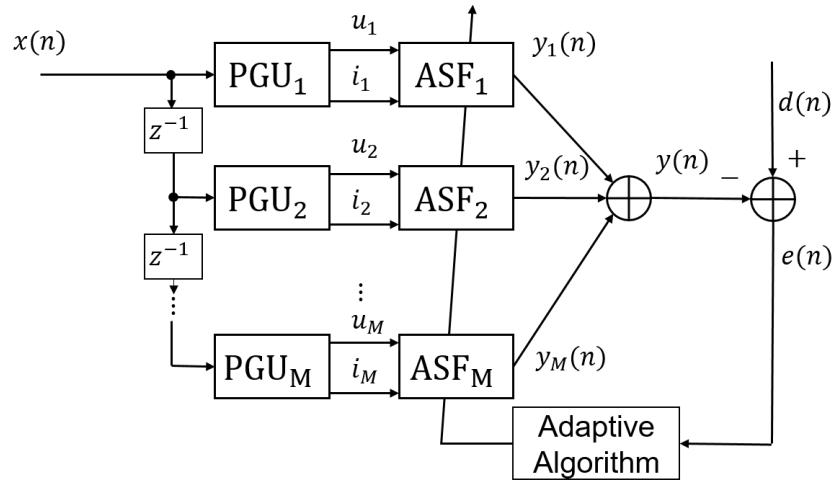


図 2.1 GSNAF のブロック図

プロセス遅延した信号を入力信号ベクトルとして  $\mathbf{x}(n) = [x(n) \ x(n-1) \ x(n-2) \ x(n-3) \ \dots \ x(n-M+1)]^T$  とする。ここで、 $x(n)$  は時間  $n$  での入力信号であり、 $M$  は入力信号ベクトルの長さである。図 2.1 では、入力信号ベクトルの各要素に対して、PGU、ASF が接続されており、その ASF の出力の合計が GSNAF の出力になっていることが分かる。PGU は Parameter Generation Unit の略であり、次に連結された ASF の処理に使用するパラメータを出力する。ASF は Adaptive Spline Function の略であり、横軸に入力、縦軸に出力とした時の適応スプライン関数である。ASF は連続であることが望ましいが、実際には関数を連続で表現することはできないので制御点として用いる離散点で構成されている。

まず、PGU について説明する。式 (2.13)、(2.14) に入力信号ベクトルが入力された PGU の出力  $i_j$ 、 $u_j$  を下式に示す。

$$i_j(n) = \left\lfloor \frac{x(n-j+1)}{\Delta x} \right\rfloor + \frac{Q_j - 1}{2} \quad (2.13)$$

$$u_j(n) = \frac{x(n-j+1)}{\Delta x} - \left\lfloor \frac{x(n-j+1)}{\Delta x} \right\rfloor \quad (2.14)$$

ここで、 $j = 1, 2, \dots, M$  であり入力信号ベクトルの要素のインデックスである。式 (2.13)、(2.14) の  $i_j(n)$ 、 $u_j(n)$  は後述する ASF における処理に使用するパラメータである。 $\Delta x$  は適応スライン関数の制御点間隔であり、 $Q_j$  は  $j$  番目の適応スライン関数の制御点総数である。また、 $\lfloor \cdot \rfloor$  は床関数を表す。

次に、ASF について説明する。各パラメータと適応スライン関数の簡単な図を図 2.2 に示す。 $i_j(n)$ 、 $u_j(n)$  は図 2.2 にしめすような  $j$  番目の適応スライン関数における制御点インデックスとその近傍の制御点からの距離を正規化した値

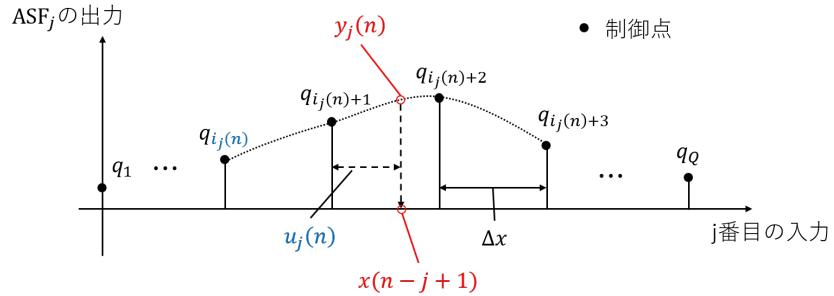


図 2.2 PGU の各パラメータと ASF の処理

である。これらのパラメータから  $j$  番目の適応スプライン関数の出力  $y_j$  が決定される。その式を式 (2.15) に示す。

$$y_j(n) = \mathbf{u}_j(n) \mathbf{C} \mathbf{q}_{i_j(n)} \quad (2.15)$$

ここで  $\mathbf{u}_j = [u_j^P(n) \ u_j^{P-1}(n) \ u_j^{P-2}(n) \ \cdots \ u_j(n), 1]$ 、 $\mathbf{q}_{i_j} = [q_{i_j}(n) \ q_{i_j+1}(n) \ q_{i_j+2}(n) \ \cdots \ q_{i_j+P}(n)]^T$ 、 $\mathbf{C}$  は式 (2.11) や式 (2.12) で表されるスプライン基底関数である。 $\mathbf{q}_{i_j}$  のように制御点インデックス  $i_j(n)$  から選ばれた制御点は活性化制御点と呼ぶ。 $P$  はスプライン曲線による補間次数である。今、 $P = 3$ 、 $\mathbf{C}$  は B スプライン関数、 $\mathbf{f}_j = \mathbf{u}_j \mathbf{C}$  とすると  $\mathbf{f}_j$  は式 (2.16) のように表される。

$$\begin{aligned} \mathbf{f}_j &= \mathbf{u}_j(n) \mathbf{C} \\ &= \frac{1}{6} [1 - 3u_j(n) + 3u_j^2(n) - u_j^3(n), 4 - 6u_j^2(n) + 3u_j^3(n), 1 + 3u_j(n) + 3u_j^2(n) - u_j^3(n), u_j^3(n)] \\ &= [f_1(u_j(n)), f_2(u_j(n)), f_3(u_j(n)), f_4(u_j(n))] \end{aligned} \quad (2.16)$$

となり、これは  $x(n-j+1)$  を非線形関数により拡張した形であることが分かる。

前述した ASF 出力の総和が GSNAF の出力であり、それは式 (2.17) で表される。

$$\begin{aligned} y(n) &= \sum_{j=1}^M y_j(n) \\ &= \sum_{j=1}^M \mathbf{u}_j \mathbf{C} \mathbf{q}_{i_j} \end{aligned} \quad (2.17)$$

さらに、各 ASF の活性化制御点として  $\mathbf{q}_A = [\mathbf{q}_{i_1}^T(n) \ \mathbf{q}_{i_2}^T(n) \ \mathbf{q}_{i_3}^T(n) \ \cdots \ \mathbf{q}_{i_M}^T(n)]$ 、非線形関数ベクトルとして  $\mathbf{f}_A = [\mathbf{f}_1^T(n) \ \mathbf{f}_2^T(n) \ \mathbf{f}_3^T(n) \ \cdots \ \mathbf{f}_M^T(n)]^T$  とすると式 (2.17) は式 (2.18) と書くことが出来る。

$$y(n) = \mathbf{f}_A \mathbf{q}_A \quad (2.18)$$

次に、GSNAF の更新式について説明する。図 2.1 より、誤差信号  $e(n)$  は式 (2.19) と表すことが出来る。

$$e(n) = d(n) - y(n) \quad (2.19)$$

ここで、 $d(n)$  は所望信号である。GSNAF では出力時の活性化制御点のみを最急降下方により更新する。評価関数  $\zeta(n) = E\{e^2(n)\}$  を最小化するように制御点を更新する。ここで、 $E\{\cdot\}$  は期待値を示す。LMS アルゴリズムを適用するために  $E\{e^2(n)\} \approx e^2(n)$  とした時、評価関数を最小化するように制御点を更新する更新則は式 (2.20) のように表される。

$$\begin{aligned} \mathbf{q}_A(n) &= \mathbf{q}_A(n-1) - \frac{\mu}{2} \Delta_{\mathbf{q}_A} \zeta(n) \\ &= \mathbf{q}_A(n-1) + \mu e(n) \mathbf{f}_A \end{aligned} \quad (2.20)$$

ここで  $\mu$  はステップサイズである。式 (2.20) より、前述した通りに出力時の各 ASF の活性化制御点のみが更新される。例えば、スプライン補間次数  $P = 3$  である場合、更新毎に各 ASF で 4 点の制御点が更新される。全ての制御点が更新されるわけではないため、計算量が少ない。前述の例ならば更新の計算量は  $O(4M)$  となる。

### 2.3 問題点

この節では GSNAF の構造上、交差項の表現が困難である点について述べる。交差項とは 2 種類以上の異なる要素の積である。例えば  $y = f(x_1, x_2) = x_1^2 + x_2^2 + 2x_1x_2$  のような関数がある場合、 $2x_1x_2$  の積が交差項、残りの項が純粋項である。

GSNAF で交差項を含むモデルと含まないモデルについてシステム同定のシミュレーションを行い、そのモデル化性能について比較する。システム同定のブロック図を図 2.3 に示す。図 2.3において、 $x(n)$  は入力信号、 $y(n)$  は出力信号、 $d(n)$  は所望信号である。また、 $\delta(n)$  は  $-40[dB]$  のノイズである。誤差信号  $e(n) = d(n) - y(n)$  を小さくするように GSNAF を更新する。以下の 2 つのモ

ルに対してシステム同定を行った。

#### シミュレーション 1: 交差項を含まないモデル

$$d_1(n) = x(n) - 0.5x^2(n-1) + 1.6x^2(n-3) \quad (2.21)$$

#### シミュレーション 2: 交差項を含むモデル

$$d_2(n) = x(n) - 0.5x(n)x(n-1) + 1.6x^2(n-3) \quad (2.22)$$

式(2.21)、式(2.22)において、 $d_1(n)$  は交差項を含まないモデルであり、 $d_2(n)$  は交差項を含むモデルである。GSNAF の各パラメータは、フィルタサイズ  $M = 10$ 、ステップサイズ  $\mu = 0.1$ 、制御点間隔  $\Delta x = 0.1$  である。それぞれのモデルに対して 30 回のシミュレーションを行い、その平均 MSE を図 2.4 に示す。図 2.4 より、交差項を含まないモデルと交差項を含むモデルの MSE を比較すると、含むモデルは  $-40[dB]$  付近まで下がっているのに対して、含まないモデルではその半分程度の  $-20[dB]$  付近までしか下がっていない。式(2.21)、式(2.22)より、2 つのモデルの違いは各式の 2 項目のみである。つまり、図 2.4 のような結果は GSNAF が交差項を含むモデルをモデル化する場合、その性能が低下するためである。

式(2.16)で述べた通り、GSNAF は  $x(n-j+1)$  を非線形関数により拡張した非線形適応フィルタである。ASF<sub>j</sub> の出力  $y_j(n) = f(x(n-j+1))$  とすると、その総和  $y(n)$  は式(2.23)と表すことができる。

$$y(n) = f(x(n)) + f(x(n-1)) + \cdots + f(x(n-M+1)) \quad (2.23)$$

ここで  $f(\cdot)$  は関数を表す。式(2.23)は入力信号ベクトル  $\mathbf{x}(n)$  の要素毎に独立した関数である。つまり要素同士の積である交差項を出力することができない。そ

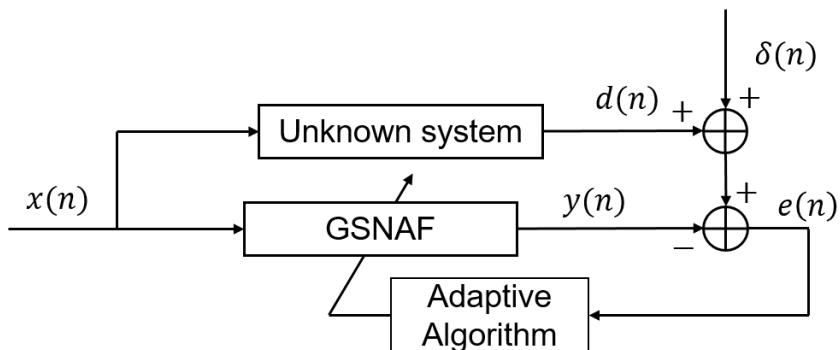


図 2.3 システム同定のブロック図

のため GSNAF で交差項を含む非線形特性をモデル化する場合、多くの場合は性能が大きく低下する。

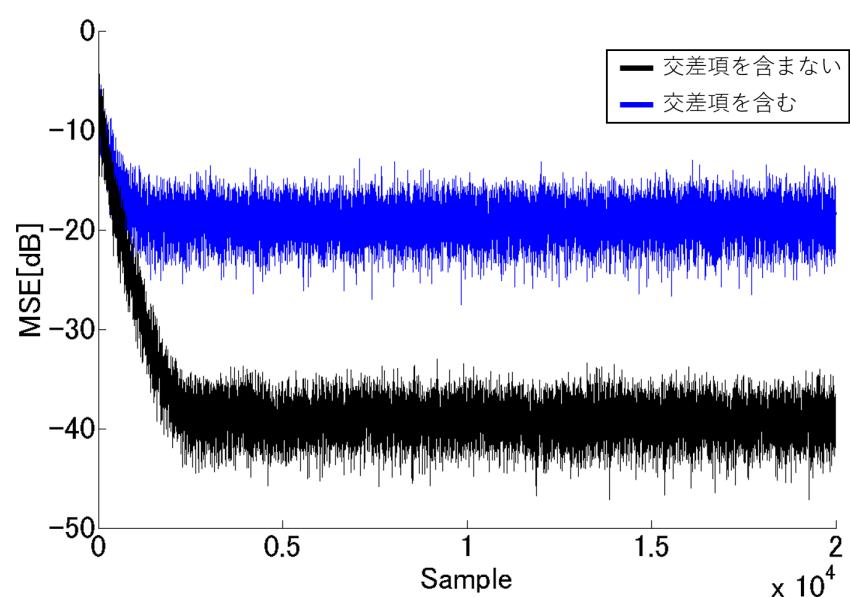


図 2.4 各モデルの 30 回平均 MSE

# 第3章

## 提案手法

本章では、提案手法について述べる。まず、3.1節では提案手法に関する知識として、スプライン曲線を用いた補間による曲面表現について述べる。次に3.2節では提案手法のブロック図や計算方法、更新方法について述べる。次に3.3節では提案手法の問題点について述べる。最後に3.4節ではその問題点の解決案であるASF削除手法について述べる。

### 3.1 スプライン曲線を用いた補間による曲面表現

この節では提案手法の原理に関する知識として、2.1節で示したスプライン曲線を用いた補間による曲面表現について述べる。式(2.7)で示したように制御点に対するスプライン曲線を用いた補間は式(3.1)で表される。

$$\varphi_i(u) = \mathbf{u}^T \mathbf{C} \mathbf{q}_i \quad (3.1)$$

ここで、図3.1のような場合を考える。図3.1は、変数  $k$  で制御点  $\mathbf{q}_i$  が変化する場合のスプライン曲線による補間である。この場合、制御点の位置は  $k$  により変化するので式(3.1)を式(3.2)のように変更する。

$$\varphi_i(u, k) = \mathbf{u}^T \mathbf{C} \mathbf{q}_i(k) \quad (3.2)$$

式(3.2)において、 $\mathbf{q}_i(k) = [q_i(k) \ q_{i+1}(k) \ \dots \ q_{i+P}(k)]^T$  であり、 $P$  はスプライン補間次数である。ここで、 $q_{i,k}(u_k) = q_i(k)$  とすると、

$$q_{i,k}(u_k) = \mathbf{u}_k^T \mathbf{C} \mathbf{q}_{k,i} \quad (3.3)$$

式(3.3)において、 $\mathbf{q}_{k,i} = [q_{k,i} \ q_{k+1,i} \ \dots \ q_{k+P,i}]^T$  であり、 $\mathbf{u}_k = [u_k^P \ u_k^{P-1} \ \dots \ u_k \ 1]$  であり、 $u_k$  は制御点間距離  $\Delta x = q_{k+1,i} - q_{k,i}$  で距離を一定としたときの正規化

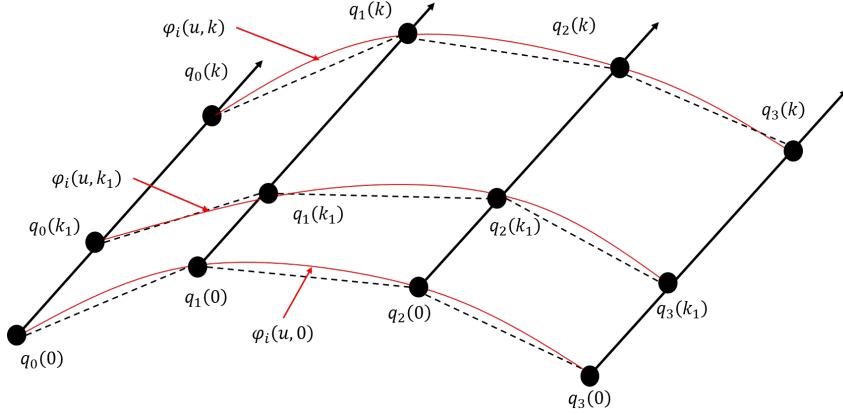


図 3.1 変化する制御点の補間

した距離の横軸座標値である。式 (3.3) は  $q_i(k)$  のスプライン曲線による補間である。式 (3.2)、式 (3.2) をまとめると、式 (3.4) が得られる。

$$\varphi_{i,k}(u_i, u_k) = \mathbf{u}_i^T \mathbf{C} (\mathbf{u}_k^T \mathbf{C} \mathbf{q}_{k,i})^T \quad (3.4)$$

ここで、 $\mathbf{q}_{k,i}$  は式で表される。

$$\mathbf{q}_{k,i} = \begin{pmatrix} q_{k,i} & \cdots & q_{k,i+3} \\ \vdots & \ddots & \vdots \\ q_{k+3,i} & \cdots & q_{k+3,i+3} \end{pmatrix} \quad (3.5)$$

さらに、転置行列において  $(AB)^T = A^T B^T$  が成り立つため、 $\mathbf{f}_i = \mathbf{u}_i^T \mathbf{C}$  とすると、式 (3.4) は式 (3.6) と簡単に表すことができる。

$$\begin{aligned} \varphi_{i,k}(u_i, u_k) &= \mathbf{f}_i (\mathbf{f}_k \mathbf{q}_{k,i})^T \\ &= \mathbf{f}_i \mathbf{q}_{k,i}^T \mathbf{f}_k^T \\ &= \mathbf{f}_i \mathbf{q}_{i,k} \mathbf{f}_k^T \end{aligned} \quad (3.6)$$

式 (3.6) により、 $(P + 1)^2$  個の制御点を用いてスプライン曲線により曲面を表現することができる。

## 3.2 提案手法の構造

提案手法は、既存手法である GSNAF が交差項を表現できるように ASF を変更した非線形適応フィルタである。具体的には GSNAF の ASF が 1 つの PGU のみに連結されていたことに対して、提案手法は 2 つの PGU に連結されている。そのブロック図を図 3.2 に示す。図 3.2 では 1 つの ASF に対して異なる 2 つの PGU から入力されていることが分かる。PGU は GSNAF と同じブロック

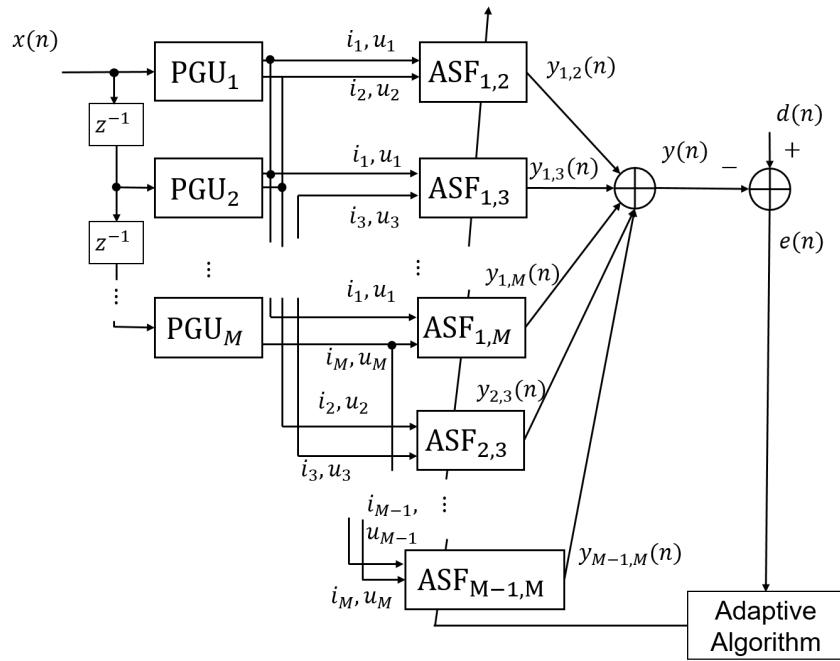


図 3.2 提案手法のブロック図

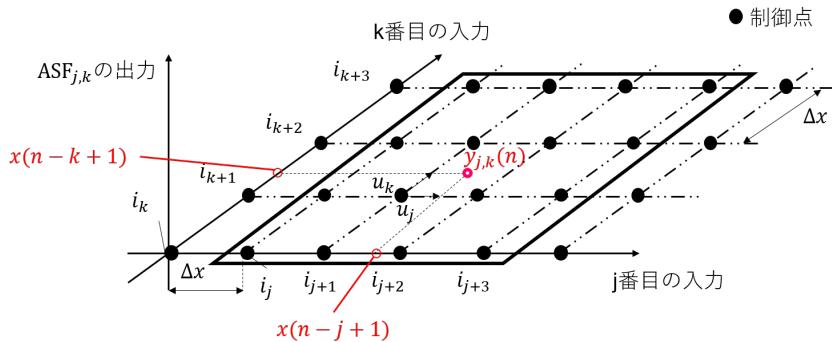


図 3.3 ASF の簡略図

であり、その出力  $i_j$ 、 $u_j$  や計算式も式 (2.13)、(2.14) で同じく、式 (3.7)、(3.8) で計算できる。

$$i_j(n) = \left\lfloor \frac{x(n-j+1)}{\Delta x} \right\rfloor + \frac{Q_j - 1}{2} \quad (3.7)$$

$$u_j(n) = \frac{x(n-j+1)}{\Delta x} - \left\lfloor \frac{x(n-j+1)}{\Delta x} \right\rfloor \quad (3.8)$$

ただし、式 (2.13) 中の  $Q_j$  は変更した ASF の 1 入力方向の制御点数である。

次に ASF について説明する。提案手法の ASF は GSNAF と異なり、2 つの入力方向を持つ 2 次元の適応スプライン関数となっている。 $i_j$ 、 $u_j$  等のパラメータと ASF の簡単な図を図 3.3 に示す。図 3.3 より異なる 2 つの PGU からそれぞ

れのインデックス  $i_j, i_k$  とそれぞれの正規化した各軸座標値  $u_j, u_k$  から  $\text{ASF}_{j,k}$  の出力を決定する。この時の  $\text{ASF}_{j,k}$  の出力式を式(3.9)に示す。

$$y_{j,k}(n) = \mathbf{f}_j \mathbf{q}_{j,k} \mathbf{f}_k^T \quad (3.9)$$

式(3.9)において、 $\mathbf{f}_j = \mathbf{u}_j \mathbf{C} = [u_j^P(n) \cdots u_j(n) 1] \mathbf{C}$  であり、 $\mathbf{q}_{j,k}$  は式(3.10)で表されるそれぞれのインデックス  $i_j(n), i_k(n)$  から選ばれた活性化制御点である。また、 $\mathbf{C}$  はスプライン基底関数行列である。

$$\mathbf{q}_{j,k} = \begin{pmatrix} q_{i_j(n), i_k(n)} & \cdots & q_{i_j(n), i_k(n)+3} \\ \vdots & \ddots & \vdots \\ q_{i_j(n)+3, i_k(n)} & \cdots & q_{i_j(n)+3, i_k(n)+3} \end{pmatrix} \quad (3.10)$$

また、提案手法の出力  $y(n)$  は式(3.9)で表された各 ASF の出力の総和であり、式(3.11)で表される。

$$y(n) = \sum_{j=0}^{Q-P-1} \sum_{k=j+1}^{Q-P-1} y_{j,k}(n) \quad (3.11)$$

ここで、 $Q$  は ASF の 1 入力方向の制御点数、 $P$  はスプライン補間次数である。GSNAF では入力信号の各要素に対する関数展開だったが、提案手法は入力信号の各要素の組み合わせによる関数展開である。

次に、提案手法の更新式について説明する。GSNAF と同様に図より誤差信号  $e(n) = d(n) - y(n)$  として、出力時の各 ASF の活性化制御点のみを最急降下法により更新する。評価関数  $\zeta(n) = E\{e^2(n)\}$  を最小化するように制御点を更新する。LMS アルゴリズムを適用するために  $E\{e^2(n)\} \approx e^2(n)$  として、更新式は式(3.12)で表すことができる。

$$\begin{aligned} \mathbf{q}_{j,k}(n) &= \mathbf{q}_{j,k}(n-1) + \frac{\mu}{2} \Delta_{\mathbf{q}_{j,k}} \zeta(n) \\ &= \mathbf{q}_{j,k}(n-1) + \mu \mathbf{f}_j^T \mathbf{f}_k \end{aligned} \quad (3.12)$$

ここで、 $\mu$  はステップサイズである。更新についても GSNAF と同様に出力時の活性化制御点のみが更新される。補間次数  $P = 3$  であるならば、 $(P+1)^2$  個の制御点、つまり 16 点の制御点が各更新で更新される。

### 3.3 問題点

提案手法では入力要素の組み合わせの数分の ASF で構成されるため、入力信号ベクトルの要素数を  $M$  とすると、ASF は  $\frac{1}{2}M(M-1)$  個生成される。これに

よる問題としてあげられるのが、計算量の増加と無駄な ASF によるモデル化性能低下である。

まず計算量の増加について考える。GSNAF の更新に必要な計算量は 2.2 節で示した通り、スプライン補間次数  $P = 3$ 、フィルタサイズ  $M$  とすると  $O(4M)$  である。これは ASF1 つ辺りに乗算 4 回で済み、ASF が  $M$  個で GSNAF が構成されているためである。対して同条件において、提案手法は前述した通り  $(P + 1)^2$  個、16 点の制御点を更新するので ASF1 つ辺りに乗算 16 回、ASF が  $\frac{1}{2}M(M - 1)$  個で構成されるため、その計算量は  $O(8M(M - 1))$  となる。これは GSNAF と比べると非常に大きい。

次にモデル化性能の低下について考える。提案手法は非常に多くの ASF で構成されている。そのため、モデル化に必要な ASF の他にモデル化に不要な ASF も存在する。不要な ASF が少数ならば問題はないが、多数の不要な ASF が存在する場合、その出力が誤差信号に影響を与え、モデル化性能を低下させる。

### 3.4 解決案

3.3 節で示した問題点を解決するために、必要な ASF のみを残して不要な ASF を削除することが求められる。不要な ASF のみを削除することで、その分の計算量を削減でき、その出力が誤差信号に与える影響も無くすことができる。削除するために、提案手法の各 ASF の出力がモデル化にどれだけ寄与しているかを考える。具体的には、各 ASF 出力に関して、モデル化中にある一定間隔で過去数サンプル分の出力データの絶対値和をそれぞれ計算する。つまり、ASF 出力を  $y_{j,k}(n)$  すると式 (3.13) を計算する。

$$Av_{j,k} = \sum_{l=0}^L |y_{j,k}(n-l)| \quad (3.13)$$

式 (3.13) で  $L$  は考慮する過去のサンプル数である。周期毎に  $Av_{j,k}$  の最大値を基準値として閾値を決め、それを数周期連続で下回っていた ASF を削除する。これにより、ASF 出力の絶対値が他と比べて小さいものが削除される。

提案手法の問題点とその解決案である ASF 削除の効果についての簡単なシミュレーションを付録 A に示す。

## 第4章

### シミュレーション

本章では、シミュレーションによって提案手法の有効性を示す。シミュレーションはシステム同定を行い、3次 Volterra フィルタ、GSNAF、提案手法の3つの非線形適応フィルタを比較する。システム同定のブロック図を図4.1に示す。図4.1において、 $x(n)$ は入力信号、 $y(n)$ は出力信号、 $d(n)$ は所望信号である。また、 $\delta(n)$ はノイズであり、全てのシミュレーションで $-40[dB]$ のノイズを付与している。また、全ての入力信号は $[1 - 1]$ の一様乱数である。

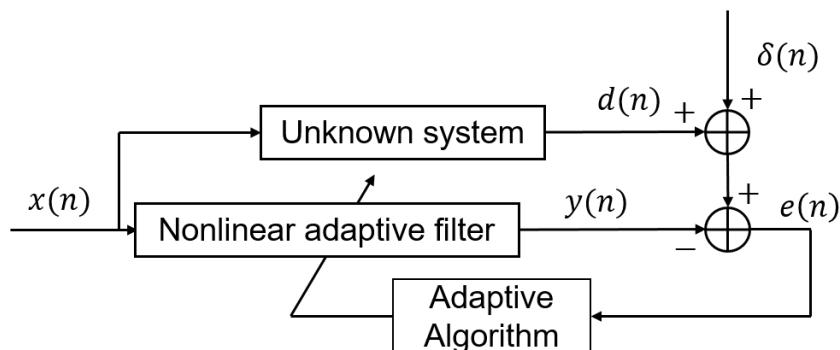


図4.1 システム同定のブロック図

## 4.1 シミュレーション 1

この節では交差項を含むモデルを提案手法がモデル化できることを示す。非線形適応フィルタの各条件を表 4.1 に示す。所望信号  $d(n)$  を式 (4.1) として、 $d(n)$  との誤差が少なくなるようにシステム同定を行った。式 (4.1) で表された  $d(n)$  は交差項を含む非線形出力のモデルである。

$$\begin{aligned} d(n) = & x(n) + 0.8x(n-1) + 0.3x(n-2) + 0.4x(n-3) - 0.8x(n)x(n-1) \\ & + 0.9x(n)x(n-2) + 0.7x(n)x(n-3) - 3.9x^2(n-1)x(n-2) \\ & - 2.6x^2(n-1)x(n-3) + 2.1x^2(n-2)x(n-3) \end{aligned} \quad (4.1)$$

式 (4.1) は線形と非線形が並列に接続された図 4.2 のような特性である。これらの条件で 30 回のシミュレーションを行い、その平均 MSE を比較した。提案手法の ASF 削除は 1000 サンプル毎に行い、各 ASF で過去 500 点の出力の総和を計算し、基準値の 30% 以下の出力が 2 回続いた ASF を削除した。

表 4.1 シミュレーション 1 の各条件

	3 次 Volterra	GSNAF	提案手法
フィルタサイズ $M$	8	8	8
ステップサイズ $\mu$	0.08	0.08	0.08
制御点間隔 $\Delta x$		0.1	0.1

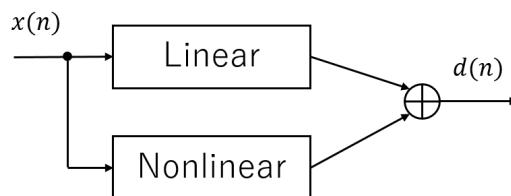


図 4.2 シミュレーション 1 のブロック図

### 4.1.1 結果と考察

シミュレーションの MSE を図 4.3 に示す。さらに、各 MSE の最終点から 1000 点平均した値を表 4.2 に示す。図 4.3 と表 4.2 より、GSNAF と提案手法を比較すると、GSNAF が同定できない非線形出力のモデルに対して提案手法は精度良く同定できていることが分かる。しかし、3 次 Volterra と提案手法を比較すると、3 次 Volterra の方が精度良く同定が行えており、収束速度も速い。これは非線形出力のモデルが Volterra フィルタでモデル化しやすい 3 次の多項式であるためであると考えられる。

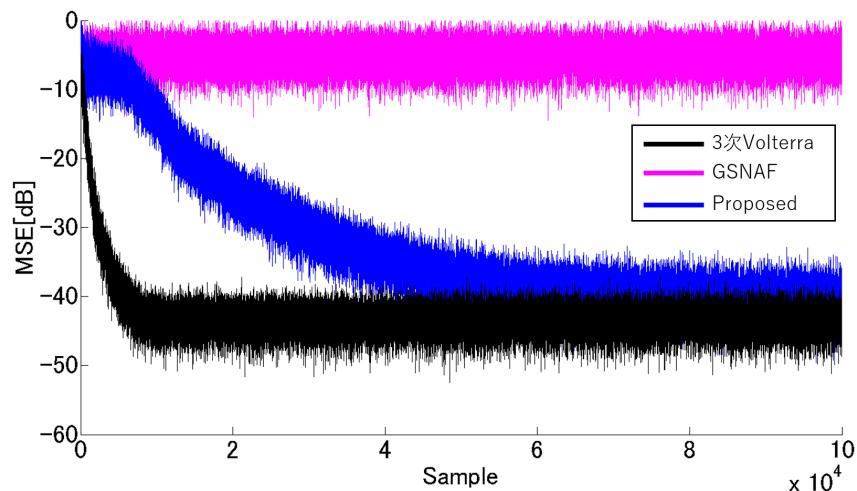


図 4.3 シミュレーション 1 の結果

表 4.2 最終点から 1000 点平均 MSE

	3 次 Volterra	GSNAF	提案手法
MSE[dB]	-43.48	-5.142	-40.15

## 4.2 シミュレーション 2

この節では、非線形項の次数が低い適応 Volterra フィルタでモデル化することが難しい三角関数等を含むモデルに対して、提案手法の有効性を示す。非線形適応フィルタの各シミュレーション条件を表 4.3 に示す。所望信号  $d(n)$  を式 (4.2) として、 $d(n)$  との誤差が少なくなるようにシステム同定を行った。

$$\begin{aligned} y_1(n) &= H_1(n) * x(n) \\ y_2(n) &= 3.3 \tanh(y_1(n)) \\ d(n) &= H_2(n) * y_2(n) \end{aligned} \quad (4.2)$$

式 (4.2)において、 $H_1(n)$ 、 $H_2(n)$  はそれぞれ線形フィルタであり  $H_1(n) = [1, -0.6, 0.05]$ 、 $H_2(n) = [1, 0.6, 0.05]$  である。また、\* は畳み込みである。式 (4.2) は線形、非線形、線形と直列に接続された図 4.4 のような特性である。

提案手法の ASF 削除は 1000 サンプル毎に行い、各 ASF で過去 500 点の出力の総和を計算し、基準値の 30% 以下の出力が 2 回続いた ASF を削除した。

表 4.3 シミュレーション 2 の各条件

	3 次 Volterra	GSNAF	提案手法
フィルタサイズ $M$	8	8	8
ステップサイズ $\mu$	0.08	0.08	0.08
制御点間隔 $\Delta x$		0.1	0.1



図 4.4 シミュレーション 2 のブロック図

### 4.2.1 結果と考察

シミュレーションの MSE を図 4.5 に示す。さらに、各 MSE の最終点から 1000 点平均した値を表 4.4 に示す。図 4.5 と表 4.4 より、GSNAF と提案手法を比較すると、提案手法は GSNAF より精度良く同定できていることが分かる。3 次 Volterra と提案手法を比較すると、収束速度は 3 次 Volterra の方が速いが同定精度は提案手法の方が良いことが分かる。提案手法の ASF 削除は同定がある程度進んでから行われ、始めの内は同定に不要な ASF を更新している。そのため提案手法の収束速度が遅いと考えられる。式 (4.2) で表されるシミュレーションモデルは、三角関数の  $\tanh$  を含んでいる。このため適応 Volterra フィルタにより精度よくモデル化するためには高い次数が必要になる。

今回のシミュレーションにおける入力信号は  $[1 - 1]$  の一様乱数であるが、入力信号の範囲が広くなると適応 Volterra フィルタでは更に同定が困難となり、提案手法との同定精度差も増加する。そのシミュレーション結果は付録 B に掲載する。

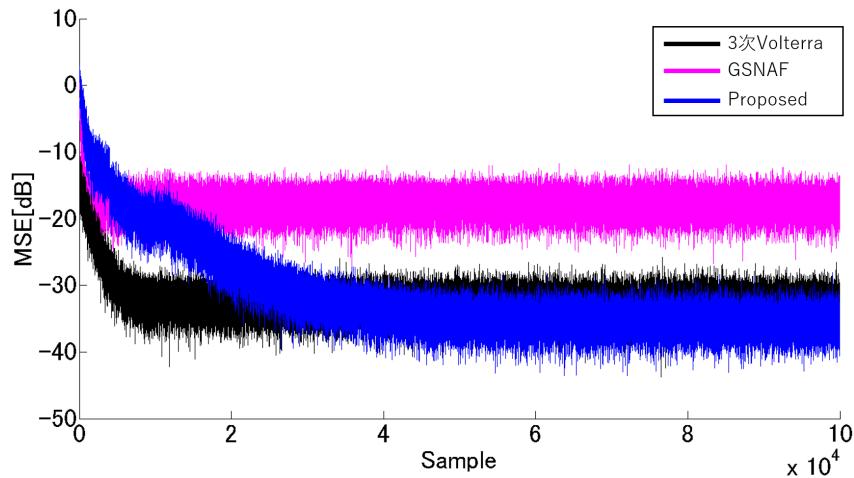


図 4.5 シミュレーション 2 の結果

表 4.4 最終点から 1000 点平均 MSE

	3 次 Volterra	GSNAF	提案手法
MSE[dB]	-32.71	-17.85	-35.37

### 4.3 シミュレーション 3

この説では、4.2 節と同様に三角関数を含むモデルに対してシステム同定を行った。非線形適応フィルタの各シミュレーション条件を表 4.5 に示す。所望振動  $d(n)$  を式 (4.3) として、 $d(n)$  との誤差が少なくなるようにシステム同定を行った。

$$\begin{aligned} y(n) &= \tanh(x(n)) \\ d(n) &= y(n) + 0.35y(n-1) + 0.09y(n-2) - 0.5y(n)y(n-1) + 0.4y(n)y(n-2) \end{aligned} \quad (4.3)$$

式 (4.3) は非線形ブロックに線形ブロックと非線形ブロックが並列に接続された図 4.6 のような特性である。提案手法の ASF 削除は 1000 サンプル毎に行い、各 ASF で過去 500 点の出力の総和を計算し、基準値の 30% 以下の出力が 2 回続いた ASF を削除した。

表 4.5 シミュレーション 3 の各条件

	3 次 Volterra	GSNAF	提案手法
フィルタサイズ $M$	10	10	10
ステップサイズ $\mu$	0.05	0.05	0.05
制御点間隔 $\Delta x$		0.2	0.2

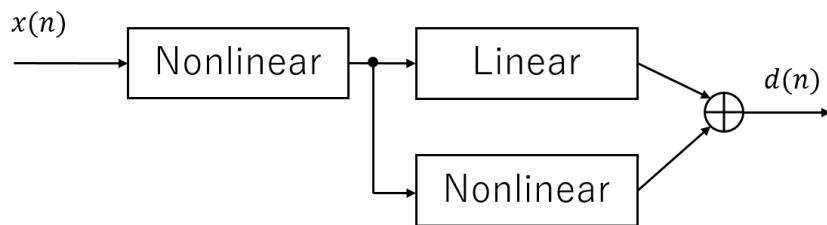


図 4.6 シミュレーション 3 のブロック図

### 4.3.1 結果と考察

シミュレーションの MSE を図 4.7 に示す。さらに、各 MSE の最終点から 1000 点平均した値を表 4.6 に示す。図 4.7 と表 4.6 より、GSNAF と提案手法を比較すると、提案手法は GSNAF より精度良く同定できていることが分かる。また、3 次 Volterra と提案手法を比較すると、収束速度は 3 次 Volterra の方が速いが、同定精度は提案手法の方が良いことが分かる。これらは 4.2.1 の結果から分かること同様である。図 4.7 をみると提案手法の MSE は 6000~9000 サンプルの間で度急峻に上昇し、その後低下していることが分かる。

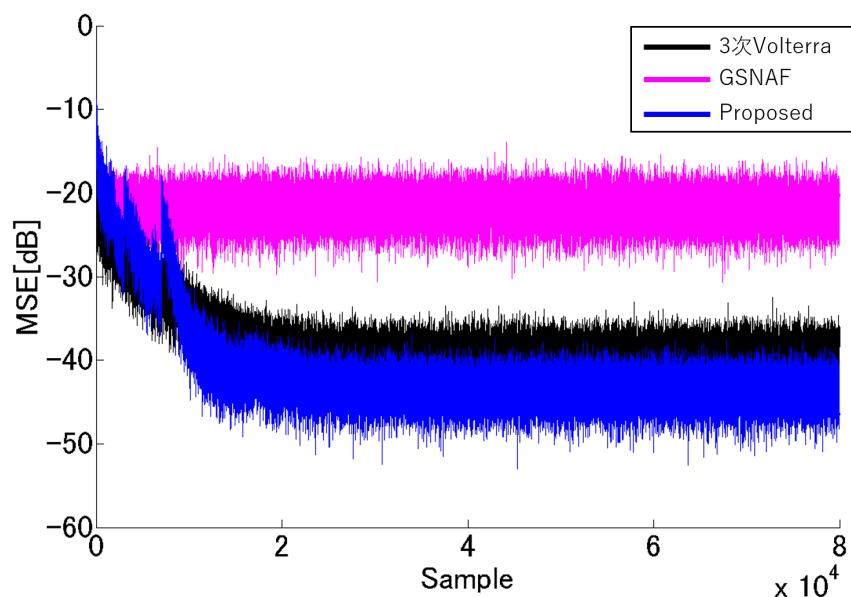


図 4.7 シミュレーション 3 の結果

表 4.6 最終点から 1000 点平均 MSE

	3 次 Volterra	GSNAF	提案手法
MSE[dB]	-39.55	-21.77	-43.58

## 4.4 シミュレーション 4

4.3.1 節の原因を調べるために、4.3 で行ったシミュレーションを同条件で提案手法だけで 1 回だけ行った。そのシステム同定中に削除された ASF と MSE について調査した。

### 4.4.1 結果と考察

シミュレーションの MSE を図 4.8 に示す。図 4.8 は 0~20000 サンプル間を拡大して表示している。また、図中の赤い丸は MSE が急上昇している部分である。さらに各 ASF が最後まで削除されなかつたか、あるいは何サンプル目に削除されたかを表 4.7 に示す。表 4.7 には一部の ASF を掲載している。表 4.7において、○は最後まで削除されなかつた ASF であり、数字は何サンプル目に削除されたかを示す。図 4.8 と表 4.7 を比べると、 $ASF_{1,4} \sim ASF_{1,9}$  が削除された時に大きく MSE が上昇していることが分かる。これは 1 つの ASF で表現できる特性を複数の ASF で表現していたことが原因であると考えられる。ASF 削除で

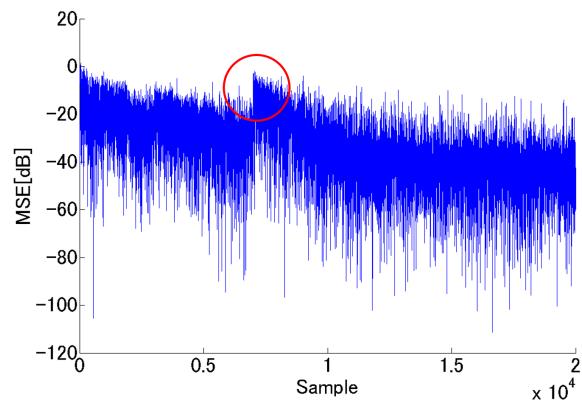


図 4.8 シミュレーション 4 の結果

表 4.7 シミュレーションの削除された ASF

	削除タイミング		削除タイミング
$ASF_{1,2}$	○	$ASF_{1,7}$	7000
$ASF_{1,3}$	○	$ASF_{1,8}$	7000
$ASF_{1,4}$	7000	$ASF_{1,9}$	6000
$ASF_{1,5}$	7000	$ASF_{1,10}$	3000
$ASF_{1,6}$	7000	$ASF_{2,3}$	3000

それらの内一部が削除されたため、削除された出力分 MSE が増加した。

## 第 5 章

### 終わりに

本論文では、従来の GSNAF から ASF の構成を変更した新しい非線形適応フィルタを提案した。第 1 章では、非線形適応フィルタの重要性や既存の非線形適応フィルタについて示した。第 2 章では、既存手法の GSNAF の原理に関する知識として、スプライン曲線を用いた補間について示した。さらに、GSNAF の構造や更新手法、そして交差項を含む非線形特性の表現が難しいという問題点について示した。第 3 章では提案手法の構造や更新手法について示した。さらに、提案手法の問題点とその解決案、その効果を示した。第 4 章では、3 次適応 Volterra フィルタ、GSNAF、提案手法の 3 つの手法で比較を行った。その結果、提案手法は GSNAF と違い、交差項を含む非線形特性をモデル化可能であり、高次の適応 Volterra フィルタでなければ、表現しづらいようなモデルも表現することができることを示した。以上から、ASF の構成を変更した新しい非線形適応フィルタは、交差項を含む非線形特性をモデル化可能であり、高次のモデルに対するモデル化性能も高いことを示した。

## 謝辞

本研究の過程において、終始ご指導ご鞭撻頂き、適切なご指導をしていただいた杉田泰則准教授に深く感謝いたします。また、本論文を執筆するにあたり、副査をご担当いただき多くの助言を賜りました岩橋政宏教授ならびに圓道知博准教授に感謝いたします。さらに、研究生活においてお世話になりました、信号処理応用研究室の皆様に感謝の意を表します。最後にここまで育てて頂いた両親へ深い感謝の意を表します。

## 参考文献

- [1] Merouane Messini, Mohamed Djendi “A new adaptive filtering algorithm for stereophonic acoustic echo cancellation” Applied Acoustics Vol.146, pp.345-354, Mar. 2019.
- [2] S. Hannah Pauline, Dhanalakshmi Samiappan, R. Kumar, Ankita Anand, Asutosh Kar, “Variable tap-length non-parametric variable step-size NLMS adaptive filtering algorithm fofr acoustic echo cancellation” Applied Acoustics Vol.159, Feb. 2020.
- [3] Dong Woo Kim, JunWoong Hur, PooGyeon Park, “Two-stage active noise control with online secondary-path filter based on an adapted scheduled-stepsize NLMS algorithm” Applied Acoustics Vol.158, Jan. 2020.
- [4] TAIHO KOH, EDWARD J. POWERS, “Second-Order Volterra Filtering and Its Application to Nonlinear System Identification” IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Vol.ASSP-33, pp.1445-1455, No.6, 1985.
- [5] 長瀬 隆久, “Volterra 級数展開を用いた非線形時系列のシステム同定” 情報処理学会論文誌 Vol.43, pp.3982-3985, Dec. 2002.
- [6] Jagdish C. Patra, Ranendra N. Pal, B. N. Chatterji, and Ganapati Panda, “Identification of Nonlinear Dynamic Systems Using Functional Link Artificial Neural Networks” IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS, Vol. 29, No. 2, Apr. 1999.
- [7] Michele Scarpiniti, Danilo Comminiello, Raffaele Parisi, Aurelio Uncini “Non-linear spline adaptive filtering” Signal Processing, Vol. 93, pp.772-783, Feb. 2013.
- [8] Jagdish C. Patra, Ranendra N. Pal, B. N. Chatterji, and Ganapati Panda, “Normalised Spline Adaptive Filtering Algorithm for Nonlinear System Idenfitication” Neural Process Lett, Vol. 46, pp.595-607, Aug. 2017.
- [9] Milan Rathod, Vinal Patel, Nithin V. George, “Generalized spline nonlinear

- adaptive filters” Expert Systems With Applications, Vol. 83, pp.122-130, Oct. 2017.
- [10] Manolya Eyiyyrekli, Davic Breen “Localized Editing of Catmull-Rom Splines” Computer-Aided Design and Applicarions, Vol.6,2009.
- [11] 山口 輝, 西尾 考治, 小堀 研一 “点群データを用いた三次元 CAD に関する研究” FIT 2011(第 10 回情報科学技術研究フォーラム), pp.367-368. Aug. 2011.
- [12] Akemi Galvez, Andres Iglesias, “Firefly Algorithm for Explicit B-Spline Curve Fitting to Data Points” Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2013

# 付録

## A 提案手法の問題点と解決案の効果

この節では、提案手法の問題点と解決案である ASF 削除の効果について記したシミュレーション結果を掲載する。提案手法において表の 3 つの条件でシステム同定のシミュレーションを行う。この条件において ASF 削除は 500 サンプル毎に行い、各 ASF で過去 50 点の出力の総和を計算し、基準値の 30% 以下の出力が 2 回続いた ASF を削除した。

$$d(n) = x(n) - 0.9x(n)x(n-1) + 1.6x^2(n-1)x(n-2) \quad (\text{A.1})$$

また、誤差信号  $e(n)$  とし所望信号との誤差が小さくなるように更新した。同条件で 30 回シミュレーションを行い、その平均の結果を示す。評価関数  $\text{MSE} = 20 \log_{10}(e(n))$  とし、その各結果を図 A.1 に示す。

図 A.1 中の青線が条件 1、緑が条件 2、赤線が条件 3 である。この結果をみると、フィルタサイズ  $M$  以外は全て々であるにも関わらず最終的な性能に差が出てしまっていることが分かる。また、解決案の ASF 削除により条件 3 は条件 2 よりも性能が向上し、最初から  $M$  が小さい条件 1 と同程度になっていることが分かる。

	条件 1	条件 2	条件 3
フィルタサイズ $M$	3	8	8
ステップサイズ $\mu$	0.1	0.1	0.1
制御点間隔 $\Delta x$	0.1	0.1	0.1
ASF 削除	△	△	○

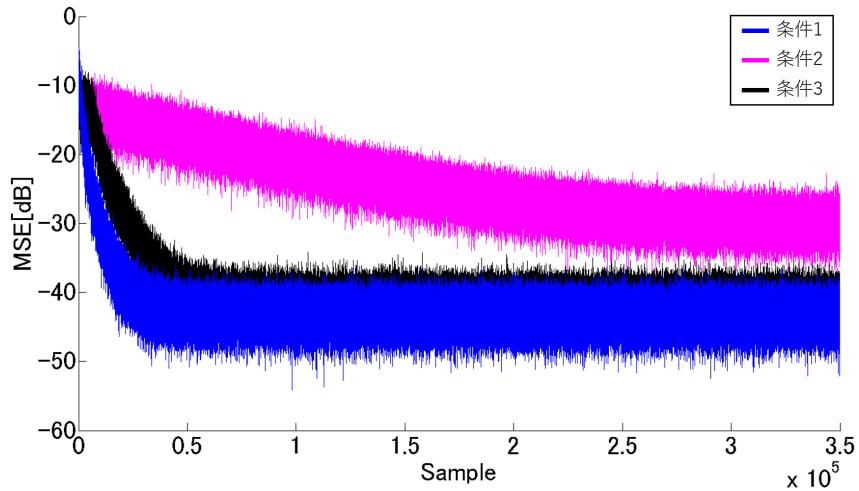


図 A.1 シミュレーションの結果

## B 4.2 のシミュレーション結果について

この節では、4.2 で行ったシミュレーションにおいて入力範囲が広くなることにより、提案手法と他 2 つの手法との MSE の差が増加することを示す。非線形適応フィルタの各シミュレーション条件を表 B.1 に示す。これらの条件にお

表 B.1 シミュレーションの各条件

	3 次 Volterra	GSNAF	提案手法
フィルタサイズ $M$	8	8	8
ステップサイズ $\mu$	0.02	0.1	0.1
制御点間隔 $\Delta x$		0.1	0.1

いて ASF 削除は 1000 サンプル毎を行い、各 ASF で過去 50 点の出力の総和を計算し、基準値の 40% 以下の出力が 2 回続いた ASF を削除した。入力信号は  $[2 - 2]$  の一様乱数である。他の条件は 4.2 節と同じである。シミュレーションの MSE を図 B.2 に示す。さらに、各 MSE の最終点から 1000 点平均した値を表 B.2 に示す。図 B.2、表 B.2 より、4.2 節の結果より提案手法と他 2 手法との MSE 差は増加していることが分かる。これは三角関数の  $\tanh(\cdot)$  への入力値が大きくなると、非線形性が強くなるためである。

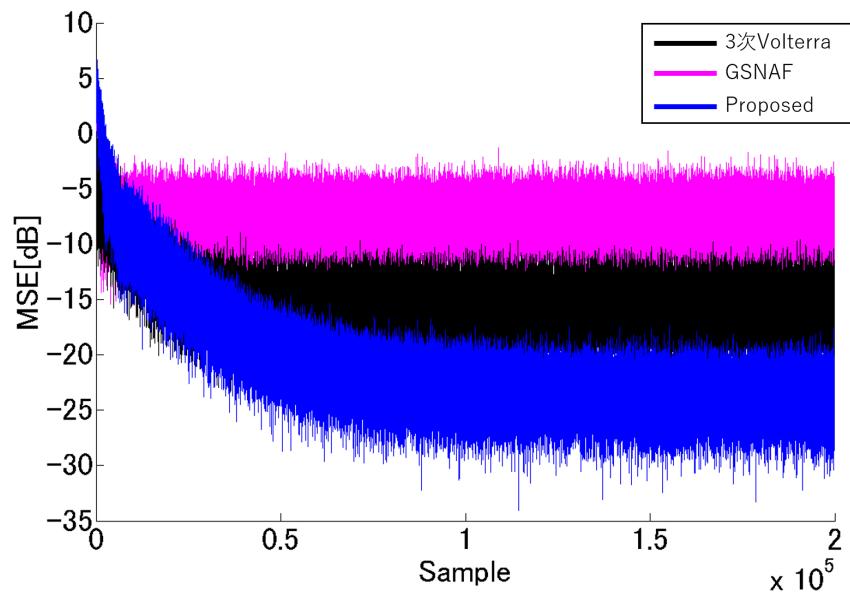


図 B.2 シミュレーションの結果

表 B.2 最終点から 1000 点平均 MSE

	3 次 Volterra	GSNAF	提案手法
MSE[dB]	-15.59	-7.69	-23.86