

卒業論文

LMS系アルゴリズムの
FPGA実装による
実時間収束特性に関する研究

指導教官 新井 宏之 教授

平成18年2月28日提出

0244031 大場 弘和

要約

無線通信の一形態である陸上移動体通信では，建物による反射・回折・散乱によりマルチパスフェージングが生じることや周波数利用効率などの制限により通信品質が劣化する問題がある．その干渉波や遅延波を除去するために，アダプティブアレーアンテナの研究が盛んに行われている，中でも，MMSE(Minimum Mean Square Error) アダプティブアレーアンテナは構成の簡単さに比べ演算速度が高いことから，移動体通信技術において幅広く用いられている．MMSE アダプティブアレーアンテナの制御アルゴリズムとして，LMS(Least Mean Square), RLS(Recursive Least Mean) アルゴリズムがあり，より高速に環境に適應できるアルゴリズムはどのアルゴリズムであるか検討する必要がある．

本論文では，通常用いられている重み付け(ウェイト) の更新回数で比較するのではなく，一回の更新に要する計算量を考慮した実時間による比較を行う．その処理時間を求めるために，LMS, N-LMS(Normalized - LMS) をデバイスに実装し，その実装結果から動作速度・処理クロック数を調べる．複数のアルゴリズムを実装するにあたり論理仕様を電氣的にプログラムすることで書き換え可能な FPGA(Field Programmable Gate Array) をデバイスに用いることを考える．FPGA に実装する演算方法は，固定小数点演算が適しており，固定小数点数とは、コンピュータにおける実数の近似値の表現方式で，整数部分に用いるビット数と小数部分に用いるビット数をあらかじめ固定して表現する．2進表記，有限桁であるゆえに丸め誤差や打ち切り誤差が生じ，有限桁数つまりワード長が短かいと丸め誤差による影響で計算結果に誤差が生じ，ワード長が長くなると，回路規模の増加に伴い，処理速度が低下する．よって，FPGA に実装を行う前に，入力変数や内部演算変数のワード長の検討を行う．また，N-LMS アルゴリズムでは，ステップサイズの計算に除算があり，その実装に除算器を用いると回路規模，動作速度は劣化する．そこで，除算器を取り除くために分母の値を2のべき乗に近似することでビットシフトを行う．

シミュレーションによるワード長の検討を行い，除算器を取り除いて実装することで，回路規模，動作速度ともに改善された．筆者らのグループで既に FPGA に実装されている RLS アルゴリズムも含め実時間による収束特性比較を行った結果，条件によって，LMS, N-LMS は RLS と同等の早さで収束することが分かった．

目次

第 1 章	序論	1
1.1	背景と目的	1
1.2	アダプティブアレーの概要	4
1.3	MMSE アダプティブアレーの概要	8
1.3.1	LMS アルゴリズム	10
1.3.2	N-LMS アルゴリズム	11
1.3.3	RLS アルゴリズム	11
1.4	FPGA の概要	13
第 2 章	固定小数点演算による LMS, N-LMS アルゴリズムの検討	14
2.1	LMS アルゴリズム	14
2.1.1	演算パラメータのワード長検討	14
2.1.2	ワード長削減	18
2.2	N-LMS アルゴリズム	19
2.2.1	除算演算とビットシフト演算	19
2.2.2	各種パラメータのワード長	20
第 3 章	FPGA 実装と実時間収束特性比較	22
3.1	FPGA 実装	22
3.1.1	LMS アルゴリズム	22
3.1.2	N-LMS アルゴリズム	23
3.1.3	実装結果	23
3.2	実時間収束特性比較	24
3.2.1	収束時間の決定	24
3.2.2	誤調整	26
3.3	比較結果	28
第 4 章	結論	30

謝辞	31
参考文献	32

第 1 章

序論

1.1 背景と目的

無線通信の一形態である移動体通信では，従来の音声のみならずデータ伝送をも考慮した高速デジタル通信システムが可能となっている．移動体通信，特に陸上移動体通信においては，送受信間の電波伝搬路が周囲の建物に遮断されることが多々ある．障害物の反射・散乱・回折により多重伝搬路ができ，それらが干渉することにより，通信品質が劣化する（マルチパスフェージング）．また，割り当てられた周波数を有効に利用するためにセル方式が導入されており，他の同一周波数のセルからの干渉を受ける問題がある（チャンネル間干渉）．これらの問題を解決するために，複数個のアンテナを配列し，指向特性の適応制御を行うアダプティブアレーアンテナは，干渉波の抑圧が可能であることから，盛んに研究が行われている [1] ．

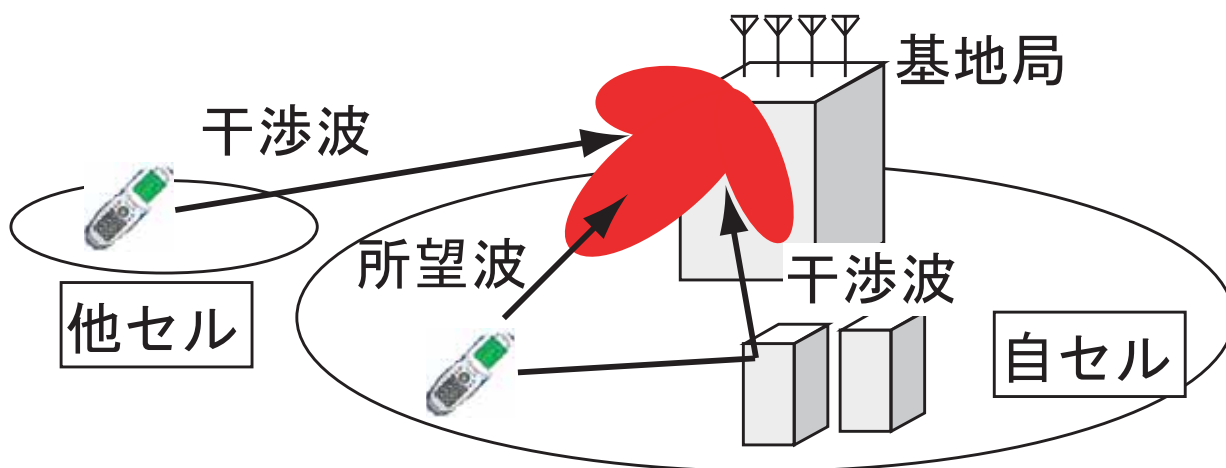


図 1.1: アダプティブアレーアンテナの基地局への適用

その中でも，MMSE(Minimum Mean Square Error) アダプティブアレーアンテナは構成の簡単さに比べ演算速度が高いことから，移動体通信技術において幅広く用いられている．

MMSE アダプティブアレーのデジタル制御による最適化アルゴリズムとして、最急降下法に基づく LMS (Least Mean Square) アルゴリズムと再帰的最小 2 乗法の RLS (Recursive Least-Squares) アルゴリズム、またそれらの拡張・修正されたアルゴリズムがある。LMS・N-LMS(Normalized - LMS) は収束速度は遅いものの、計算負荷が非常に少なく簡単な演算のみで実現でき、実装するハードウェアの規模が小さい場合には有効である。これに対して、RLS アルゴリズムは複数の行列演算を伴い計算負荷が大きくなるが収束が極めて速いことで知られている。これらのアルゴリズムの中で環境の変化により高速に対応できるアルゴリズムがどれかという検討を本論文では、図 1.2 のように行う。

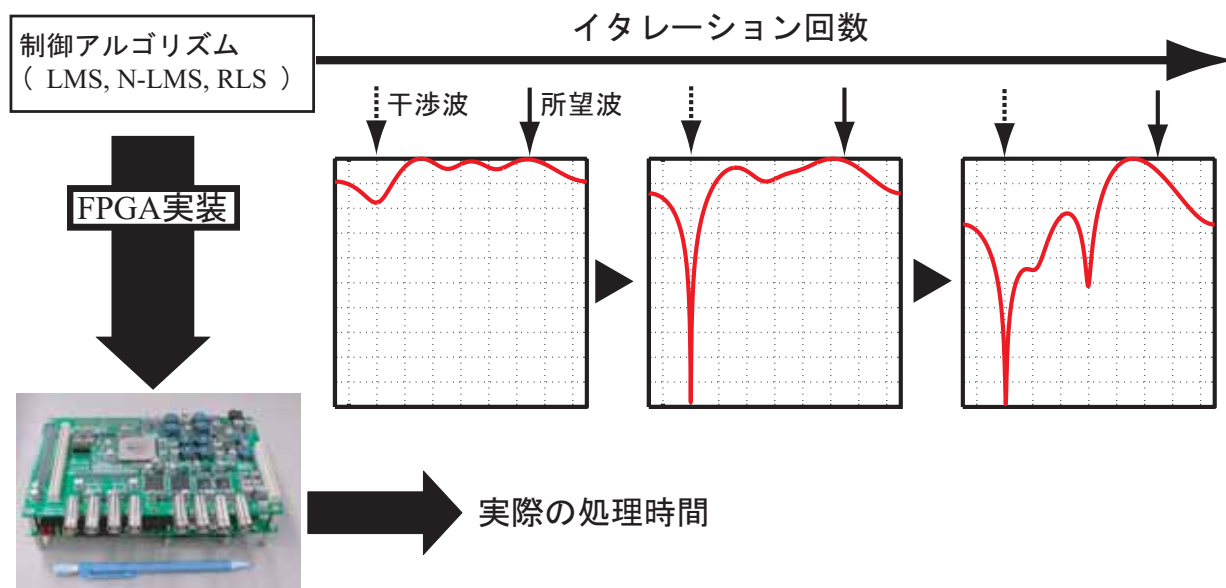


図 1.2: イタレーション回数から実時間への評価

図のように MMSE アダプティブアレーアンテナでは、制御アルゴリズムにより指向特性を逐次的に更新していく。その最適な指向特性を得るまでの過程を収束特性といい、各アルゴリズムの収束特性の比較は通常、収束するまでの制御アルゴリズムの繰り返し回数で行われている。アルゴリズムによって一回の更新に要する計算量が異なる。そのため本論文では、LMS, N-LMS をデバイス上に実装することにより、一回の更新に要する処理時間を求め、実時間による収束特性の比較を検討する。複数のアルゴリズムを実装するので、論理仕様を電氣的にプログラムすることで書き換え可能な FPGA (Field Programmable Gate Array) を用いる。FPGA に実装する演算方法としては固定小数点が適しており、固定小数点数とは、コンピュータにおける実数の近似値の表現方式で、整数部分に用いるビット数と小数部分に用いるビット数をあらかじめ固定して表現する。固定小数点で実数を表現するのに用いるビット数をワード長という。2 進表記、有限桁であるゆえに丸め誤差や打ち切り誤差が生じ、有限桁数が短かいと丸め誤差による影響で計算結果に大きな誤差が生じてしまう。しかし、ワード長が長くなると、回路規模の増加に伴い、処理速度のが低下する。よって、FPGA に実装を行う前に、入力変数や内部演算変数のワード長の検討をシミュ

レーションにより行う。また，N-LMS アルゴリズムでは，ステップサイズの計算に除算があり，その実装に除算器を用いると回路規模，動作速度は劣化する。そこで，除算器を取り除くために分母の値を 2 のべき乗に近似することでビットシフトを行う。

第 2 章では，固定小数点演算におけるワード長決定のために，シミュレーションによる検討と除算器を取り除くために行うビットシフト演算の方法を説明する。第 3 章では，実際に FPGA に実装した回路の説明と，2 章で行った検討によって，回路規模・動作速度がどのくらい改善できたかを説明し，各アルゴリズムの回路規模・動作速度から，実時間による収束特性を比較する。また，収束特性に大きく影響するステップサイズ (LMS, N-LMS), 忘却係数 (RLS) をアダプティブアレーの収束特性を表現する誤調整というパラメータを近似的に用いて選択することで，等しい条件での収束特性の比較を考える。第 4 章で結論とする。

1.2 アダプティブアレーの概要

アレーアンテナを構成するためのアンテナ素子の配列法は直線状、平面状、曲面状などいろいろ考えられるが、ここでは K 素子よりなるリニア (直線状) アレーを考える.

今、電波 (平面波) がブロードサイドから測って角度 θ の方向から到来したとする. ベースライン上の基準点での受信信号を $E_0(t)$ と表し、各アンテナ素子の受信特性は等しいとすると、 k 番目のアンテナ素子に誘起する電圧は次式で与えられる.

$$E_k(t) = E_0(t - \tau_k) \quad (k = 1, 2, \dots, K) \quad (1.1)$$

$$\tau_k = \frac{d_k \sin \theta}{c} \quad (1.2)$$

ここに、 c は伝播速度、 d_k は基準点より測った k 番目の素子の位置である. さらに、受信信号がアレー開口長 $|d_K - d_1|$ に対して十分狭帯域、すなわち帯域幅を Δf とすれば、

$$2\pi\Delta f \frac{|d_K - d_1|}{c} \ll 1 \quad (1.3)$$

であれば、 f を搬送波周波数として、 $E_0(t - \tau_k) \cong E_0(t) \exp(-j2\pi f \tau_k)$ とおくことができる. このとき式 (1.1) の第 k 素子の受信電圧は

$$E_k(t) = E_0(t) \exp(-j2\pi f \tau_k) \quad (1.4)$$

$$= E_0(t) \exp(-j2\pi f \frac{d_k}{c} \sin \theta) \quad (1.5)$$

$$= E_0(t) \exp(-j \frac{2\pi}{\lambda} d_k \sin \theta) \quad (1.6)$$

と表される. ここに、 $\lambda = c/f$ は搬送波の波長である.

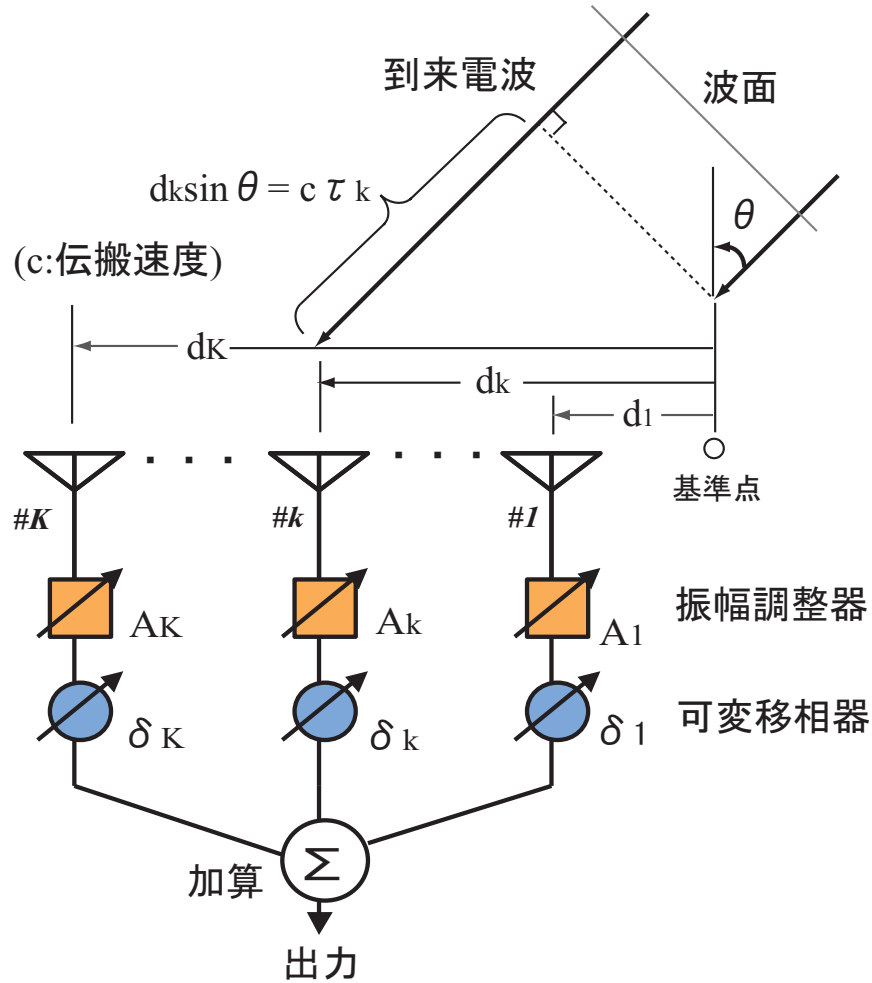


図 1.3: K 素子リニアアレー

図 1.3 のように各素子の出力をそれぞれ振幅調整器 (増幅器または減衰器) と可変移相器を経て加算すると、合成出力 $E_{\text{sum}}(t)$ は

$$E_{\text{sum}}(t) = E_0(t) \sum_{k=1}^K A_k \exp(-j2\pi f \frac{d_k}{c} \sin \theta + j\delta_k) \quad (1.7)$$

$$= E_0(t) D(\theta, f) \quad (1.8)$$

$$D(\theta, f) \triangleq \sum_{k=1}^K A_k \exp\{-j(2\pi f \frac{d_k}{c} \sin \theta - \delta_k)\} \quad (1.9)$$

となる。ここに、 A_k 、 δ_k はそれぞれ k 番目の素子に掛けられる重み (実数) と移相量である。 δ_k は所望の受信信号の到来方向と素子の位置に応じて決められるが、ある角度 θ_0 方向から到来する信号 (所望信号) を受信したい場合は、一般に移相量を

$$\delta_k = 2\pi f \frac{d_k}{c} \sin \theta_0 = \frac{2\pi}{\lambda} d_k \sin \theta_0 \quad (1.10)$$

と選ぶ。すなわち、所望信号に関して移相器の出力での位相が各チャンネルにわたって揃うように定められる。それ以外の方向では、各チャンネルの出力のチャンネルが一致せず、互いにあ

る程度の相殺が行われる. このようにアレーアンテナを用いると所望信号に対する利得が上がる. ただし、 d_k が大きい場合には、

$$2\pi f \frac{d_k}{c} \sin \theta_{gm} - \delta_k = 2m\pi \quad (m = \pm 1, \pm 2, \dots) \quad (1.11)$$

を満足するような角度 $\theta_{gm} (\neq \theta_0)$ でも同相になって加算されるので、大きなアレー応答値を生ずる. これはグレーティングローブ (grating lobe) と呼ばれ、設計の段階で防止策がとられるのが普通である.

式 (1.9) の絶対値 $|D(\theta, f)|$ を角度 θ の関数として表したものは指向性パターンと呼ばれ、その最大値周辺をメインローブ (main lobe) と呼ぶ. その他にも局所的に極大値がいくつも存在するが、これらはサイドローブ (side lobe) と呼ばれる. また、ローブとローブの間の零点をヌル (null) という.

アンテナ素子が等間隔に配置されているときには、式 (1.7) は整次多項式の形となるので、数学的な手段を利用して A_k 、 δ_k を選ぶことにより、指向性パターンにおいて、サイドローブを全体的に低くしたり、あるいは特定の強力な不要波に対してその到来方向の応答値を零にしたりすることが可能である. しかし、到来する電波の到来方向が未知であったり、または変化する場合には、何らかの学習を行って得られた情報を利用し、最適の特性を作りあげることが必要になる. このような思想を背景にして生まれたシステムがアダプティブアレーである.

アダプティブアレーアンテナの主な構成を図 (1.4) に示す. アダプティブアレーの機能は、目的により、アダプティブビームフォーミング (adaptive beamforming) とアダプティブヌルステアリング (adaptive null-steering) に大きく分類できる. アダプティブビームフォーミングは、受信波の到来方向が未知あるいは時間的に変化する場合にも、アレーのビームを自動的に追従させる機能である.

一方、強い妨害波の存在下で微弱な所望電波を受信する場合に一般的な指向性合成法を用いるとすれば、非常に低いサイドローブレベルを設定しなければならない. そこで、指向性パターンのヌル点を自動的に妨害波方向に向ける必要性が生じてくる. これが、アダプティブヌルステアリングである.

アダプティブアレーは、電波環境に関する情報を学習しながら、指向特性および周波数特性を環境に合わせて変えていくので、不要波および雑音を含んだ電波環境から所望信号を抽出するために所望信号に関する予備知識を必要とし、通常、信号の中心周波数 (搬送波周波数)、到来波方向、変調方式、偏波などが利用される. そのため、アダプティブアレーの動作原理は、それら予備知識および評価関数によって次のように大別される.

- 1) 最大 2 乗誤差法 (Minimum Mean Square Error : MMSE)
- 2) 最大 SNR 法 (Maximum Signal-to-Noise ratio : MSN)
- 3) 拘束出力電力最小化法 (Constrained Minimization of Power : CMP)

4) 定包絡線信号用アルゴリズム (Constant Modulus Algorithm : CMA)

移動通信に適してアダプティブアレーの動作原理としては，所望信号の到来方向に関する情報を必要としないもの，かつ，所望波と相関のある多重 (遅延波) の抑圧が可能なものが要求される。

次に上記の動作原理に該当する MMSE について述べる。

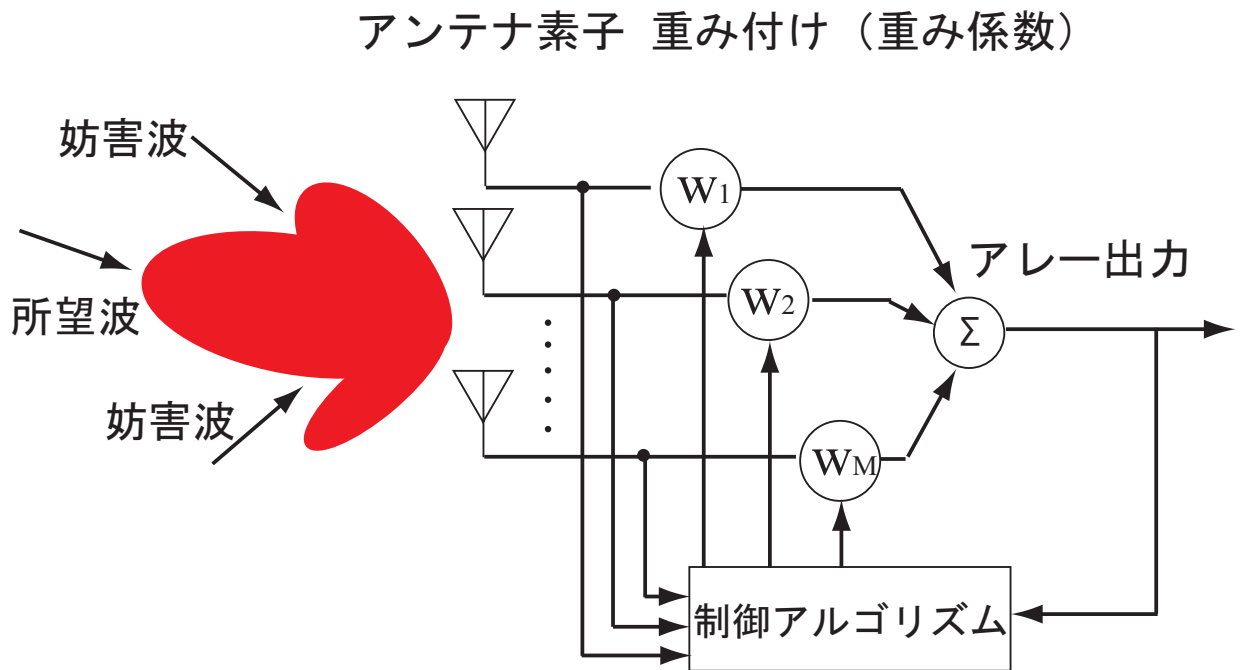


図 1.4: アダプティブアレーアンテナ

1.3 MMSE アダプティブアレーの概要

MMSE アダプティブアレーは所望のアレー応答である参照信号と実際の出力信号との差 (誤差信号) を最小にすることによって最適なウェイトを決定するシステムである (図 1.5) . この方式はアダプティブマルチステアリングと同時にアダプティブビームフォーミングを行う . そのために素子配列やシステム構成に制約を受けないという長所がある反面 , 参照信号として厳密には所望信号そのものを必要とするという矛盾がある . 実際には , 所望信号の性質 (周波数帯域 , 変調方式等) に関する予備知識があるので , 後述するようにアレー出力信号を適当に処理することによって適切な参照信号を得ることが可能である . したがって , 受信側で所望信号のレプリカである参照信号を作るという概念は現実的な手段となり , アダプティブプロセッサが完全な所望信号の性質をあらかじめ既知として特性の解析を進めることができる .

最小化の対象となる誤差信号 $e(t)$, すなわち , 所望のアレー応答 (参照信号) $r(t)$ と実際のアレー出力信号 $y(t)$ との差は次式で与えられる .

$$e(t) = r(t) - y(t) = r(t) - \mathbf{w}^H \mathbf{x}(t) \quad (1.12)$$

上式より , 誤差信号の 2 乗の期待値 (平均 2 乗誤差) は次のように表される .

$$\begin{aligned} E[|e(t)|^2] &= E[|r(t) - y(t)|^2] = E[|r(t) - \mathbf{w}^H \mathbf{x}(t)|^2] \\ &= E[|r(t)|^2] - \mathbf{w}^T \mathbf{r}_{\mathbf{x}r}^* - \mathbf{w}^H \mathbf{r}_{\mathbf{x}r} + \mathbf{w}^H \mathbf{R}_{\mathbf{x}\mathbf{x}} \mathbf{w} \end{aligned} \quad (1.13)$$

ただし ,

$$\mathbf{R}_{\mathbf{x}\mathbf{x}} \triangleq E\{\mathbf{x}(t)\mathbf{x}^H(t)\} \quad (1.14)$$

$$\mathbf{r}_{\mathbf{x}r} \triangleq E[\mathbf{x}(t)r^*(t)] \quad (1.15)$$

であり , $\mathbf{R}_{\mathbf{x}\mathbf{x}}$, $\mathbf{r}_{\mathbf{x}r}$ はそれぞれ信号の相関行列 , 相関ベクトルと呼ばれている . MMSE アダプティブアレーの目的はウェイトベクトル \mathbf{w} を適切に選ぶことによって式 (1.13) の平均 2 乗誤差を最小にすることである .

アンテナ素子 重み付け (重み係数)

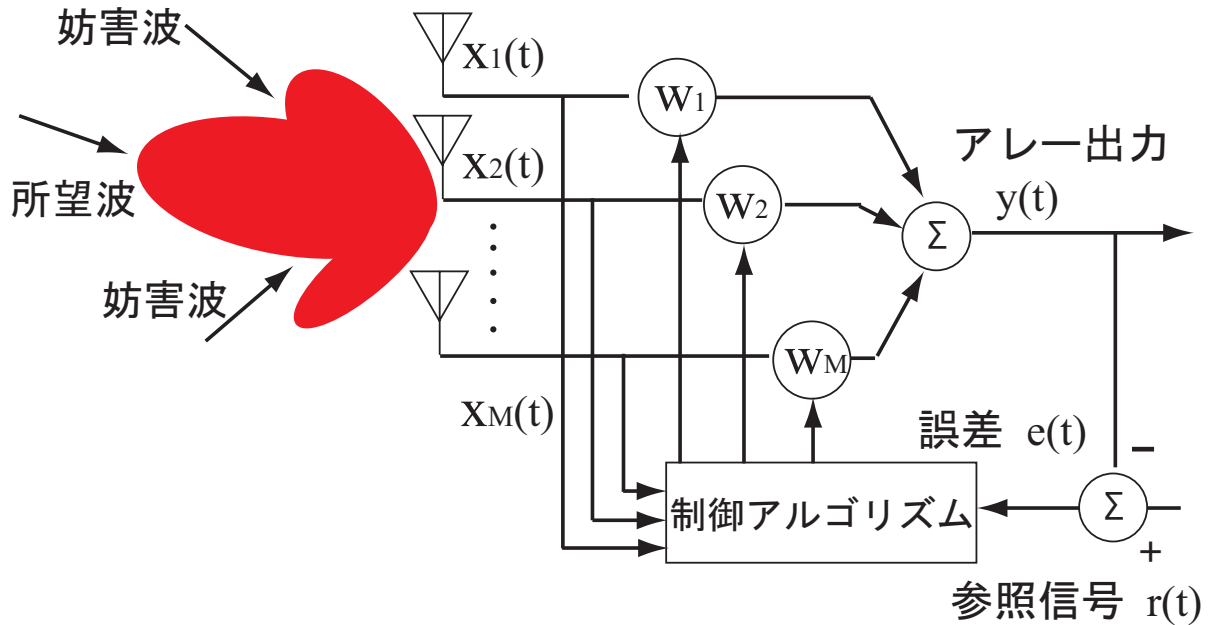


図 1.5: MMSE アダプティブアレー

式 (1.13) はウェイトベクトル \mathbf{w} の 2 次関数であり，相関行列 \mathbf{R}_{xx} が正定値であるので極値が唯一の最小値となる．それ故，平均 2 乗誤差を最小にするウェイトベクトル \mathbf{w} の値 (最適ウェイト) は式 (1.13) のウェイトベクトルに関する勾配を零とおく，すなわち，

$$\nabla_{\mathbf{w}} E[|e(t)|^2] = 0 \quad (1.16)$$

によって求めることができる．上式中の $\nabla_{\mathbf{w}} E[|e(t)|^2]$ は平均 2 乗誤差のウェイトベクトルに関する勾配であり，ベクトルによる微分演算法を用いると

$$\nabla_{\mathbf{w}} E[|e(t)|^2] = -2\mathbf{r}_{rx} + 2\mathbf{R}_{xx}\mathbf{w} \quad (1.17)$$

と得られるので最適ウェイト \mathbf{w}_{opt} は次式で与えられる．

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xr} \quad (1.18)$$

これはウィーナ解と呼ばれる．相関行列 \mathbf{R}_{xx} は正定値であるので正則であり，逆行列が存在する．式 (1.18) を式 (1.13) に代入することによって最適時の平均 2 乗誤差が次式のように得られる．

$$E[|e(t)|_{\text{min}}^2] = E[|r(t)|^2] - \mathbf{w}_{\text{opt}}^T \mathbf{r}_{xr}^* - \mathbf{w}_{\text{opt}}^H \mathbf{r}_{xr} + \mathbf{w}_{\text{opt}}^H \mathbf{R}_{xx} \mathbf{w}_{\text{opt}} \quad (1.19)$$

$$= E[|r(t)|^2] - \mathbf{r}_{xr}^H \mathbf{R}_{xx}^{-1} \mathbf{r}_{xr} \quad (1.20)$$

式 (1.18) により MMSE アダプティブアレーの最適ウェイトベクトルが与えられる。しかし、この式からウェイトベクトルを求めるためには相関行列と相関ベクトルが必要になる。これらは式 (1.14)、式 (1.15) からわかるように集合平均で与えられている。集合平均は数学的に定義された概念であり、その値を現実的に求めることはできない。そこで、観測可能な量からウェイトベクトルを決定する近似的な計算を用いた最適化アルゴリズムが必要となる。

デジタル制御による最適化アルゴリズムには、最急降下法に基づく LMS (Least Mean Square) アルゴリズムと、サンプル値を用いた直接解法の SMI (Sample Matrix Inversion) アルゴリズム、さらには再帰的最小 2 乗法の RLS (Recursive Least-Squares) アルゴリズムなどがある。最急降下法は最も一般的であり、確実に評価関数の最小点にたどりつくこと、計算負荷も小さいことなどが特長である。しかし、最急降下法は入射波の到来角が接近していたり、各波の電力比が大きい場合、収束が極端に遅くなるという欠点がある。これは相関行列の固有値分散の問題 (固有値の広がりが大きくなる問題) と呼ばれている [2]。この問題を克服する方法の一つが SMI 方式である。SMI 方式は計算負荷が大きいことから従来敬遠されがちであったが、近年のコンピュータの発達にともなって注目されるようになった。また、SMI 方式と同様に固有値分散問題を克服するもう一つのアルゴリズムが RLS アルゴリズムである。RLS アルゴリズムはカルマンフィルタの一種と考えられ、最急降下法と SMI の複合的なアルゴリズムとも解釈できる。

1.3.1 LMS アルゴリズム

最急降下法に基づく最適化アルゴリズムは次のように表される。

$$\mathbf{W}(m+1) = \mathbf{W}(m) + \frac{\mu}{2} \nabla_{\mathbf{W}} E[|e(m)|^2]. \quad (1.21)$$

ここに、 μ はステップサイズで、ウェイト更新の割合を調整する。式 (1.17) より $\nabla_{\mathbf{W}} E[|e(m)|^2]$ は次のように表される。

$$\begin{aligned} \nabla_{\mathbf{W}} E[|e(m)|^2] &= -2\mathbf{r}_{xd} + 2R_{xx}\mathbf{W}(m) \\ &= -2E[\mathbf{X}(m)r^*(m)] + 2E[\mathbf{X}(m)\mathbf{X}^H(m)]\mathbf{W}(m) \\ &= -2E[\mathbf{X}(m)\{r^*(m) - \mathbf{X}^H(m)\mathbf{W}(m)\}] \\ &= -2E[\mathbf{X}(m)\{r^*(m) - y^*(m)\}] \\ &= -2E[\mathbf{X}(m)e^*(m)] \end{aligned} \quad (1.22)$$

これを式 (1.21) に代入すると

$$\mathbf{W}(m+1) = \mathbf{W}(m) + \mu E[\mathbf{X}e^*(m)] \quad (1.23)$$

が得られる．上式においてアンサンブル平均演算を除き，一回のサンプルでウエイト更新すると

$$\mathbf{W}(m+1) = \mathbf{W}(m) + \mu E[\mathbf{X}(m)e^*(m)] \quad (1.24)$$

が得られる．

ステップサイズ μ の値は収束条件から

$$0 < \mu < \frac{1}{\lambda_{max}} \quad (1.25)$$

を満たさなければならない．ここに， λ_{max} は相関行列 R_{xx} の最大固有値である．

1.3.2 N-LMS アルゴリズム

Normalized LMS(N-LMS) は、LMS のステップサイズが一定値だったのに対し、入力データに依存した式 (1.26) のステップサイズを用いる [3]．LMS がステップサイズを決める際、発散せずに収束するように相関行列の固有値の評価をしないとイケないが、NLMS はその必要がなくなる．また、収束特性の安定性は劣化するが、速度が速くなる [4]．

$$\mu = \frac{\mu_0}{\|x(m)\|^2} \quad (1.26)$$

実装にあたり、ステップサイズの除算演算には通常、除算器を用いるが回路規模が非常に大きくなってしまったため、ビットシフトを用いる．両者の比較として回路規模，精度，動作速度を調べる．

1.3.3 RLS アルゴリズム

RLS アルゴリズムは入力のサンプル値を用いて相関行列を推定していく点では、SMI 方式と似ている．一方、ウエイトを逐次更新していく点では最急降下法にも似ている．RLS アルゴリズムは現時点までの入力サンプルを用いて、次式のような指数重み付け誤差の 2 乗和を直接最小化する．

$$E(n) = \sum_{i=1}^n \alpha^{n-i} |e(i)|^2 \quad (1.27)$$

$$e(i) = r(i) - y(i) = r(i) - \mathbf{w}^H(n)\mathbf{x}(i) \quad (1.28)$$

ただし， α は $0 < \alpha \leq 1$ の重み付け乗数で忘却係数と呼ばれる．この式 (1.27) の \mathbf{W} に関する勾配ベクトルをゼロとにおいて最小 2 乗解が次式のように得られる．

$$\mathbf{R}_{xx}(n) = \sum_{i=1}^n \alpha^{m-i} \mathbf{x}(i)\mathbf{x}^H(i) \quad (1.29)$$

$$\mathbf{r}_{xr}(n) = \sum_{i=1}^m \alpha^{m-i} \mathbf{x}(i) y^*(i) \quad (1.30)$$

この形は、SMIの直接開放と同じであるが、RLSアルゴリズムでは直接解法を用いず逐次方式で以下のようにウェイトを求める。

$$\mathbf{W}(m+1) = \mathbf{W}(m) + \gamma \mathbf{R}_{xx}(m)^{-1} \mathbf{X}(m+1) e^*(m+1), \quad (1.31)$$

$$\mathbf{R}_{xx}(m) = \alpha \mathbf{R}_{xx}(m-1) + \mathbf{X}(m) \mathbf{X}^H(m), \quad (1.32)$$

ただし、

$$\gamma = \frac{1}{\alpha + \mathbf{X}^H(m+1) \mathbf{R}_{xx}(m)^{-1} \mathbf{X}(m+1)}, \quad (1.33)$$

となり、初期値 $\mathbf{R}^{-1}(0) = I/\delta$ (δ は小さな正定数) とするのが通常である。このようにRLSはウェイトのみならず、相関行列も更新していることが分かる。そのため、計算量は増大するが高速に収束し、近接波や各波の電力比が大きい場合でも収束特性が劣化しない特徴をもつ。

1.4 FPGAの概要

ソフトウェアは書き換えができるが、速度が遅い。ハードウェアは書き換えができないが速度が速い。FPGA(Field Programmable Gate Array) は、ハードウェアでありながらソフトウェアのように書き換え修正ができる特徴を持っている。FPGAには図 1.6のように論理ブロック、入出力ブロック、配線から構成される。すでに用意されている入出力と論理ブロックをどのように組み合わせるかを自由に決めることができ。これにより、ソフトウェアのように動作内容を書き換えることを実現している。

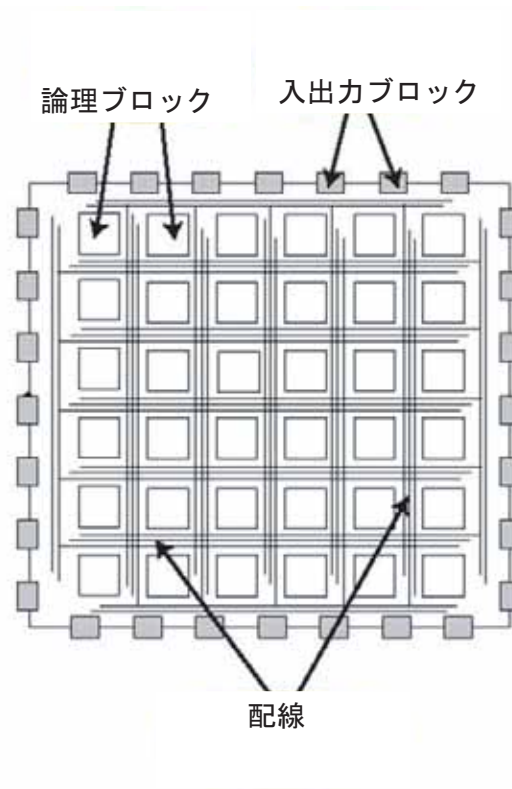


図 1.6: FPGAの構造

第 2 章

固定小数点演算による LMS, N-LMS アルゴリズムの検討

この章では，FPGA に LMS, N-LMS を実装するにあたり，それぞれのアルゴリズムを固定小数点演算で考える必要があり，入力ベクトル，ウェイトベクトルなどの各演算パラメータを固定小数点数で表現する．固定小数点数で表現する際，ビット数の多さがその数の精度を表すことになる．このビット数のことをワード長という．実装を行う前にシミュレーションにより，各演算パラメータのワード長がどのくらい必要かを検討する．

2.1 LMS アルゴリズム

2.1.1 演算パラメータのワード長検討

想定するシステムを図 2.1 に示す．アレーアンテナから受信されたアナログの RF(Radio Frequency) 信号は，DBF 受信機により IF 信号にまで周波数変換され，そのアナログの IF 信号は A/D 変換機を通じてデジタル IF 信号に変換される．そして，FPGA にてデジタルダウンコンバート (DDC) されたベースバンドの I, Q(in-phase, Quadrature-phase) 信号が得られ，CPU(Central Processing unit) 制御することで処理が行われる．実際には変調信号が入力されるので図 2.1 のようなシステムを想定して評価を行わなければならないが，今回は正弦波を用いたので上記のような処理ではなくアダプティブプロセッサ部だけを用いての評価を行った．

図 2.2 は LMS アルゴリズムのブロック図を示しており，入力情報である入力信号ベクトル x と参照信号 r のワード長を決める．想定するシステムである図 2.1 より，信号処理部の A/D 変換器の解像度が 12 ビットなので，そのダイナミックレンジを最大限に利用するために 12 ビットでフルスケールレンジになるように変換する．つまり入力情報の固定ワード

長は 12 ビットとする．それに合わせて参照信号 r のワード長も 12 ビットとした．ただし，入力信号レベルが変動する場合にも A/D 変換器に入力する信号のレベルを一定にするために，ソフトウェアによる AGC (Automatic Gain Control) を行うことを前提としている．次に，ウエイトベクトル w について考えた．

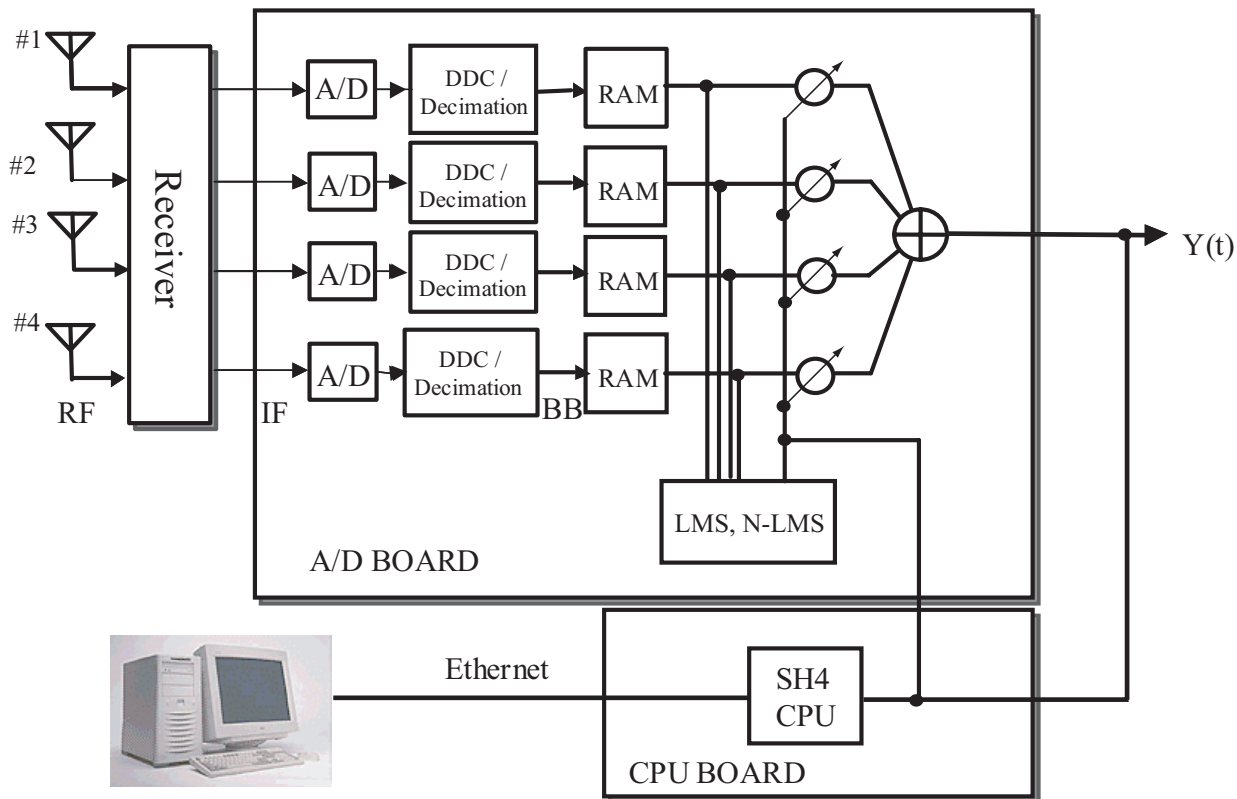


図 2.1: システム構成

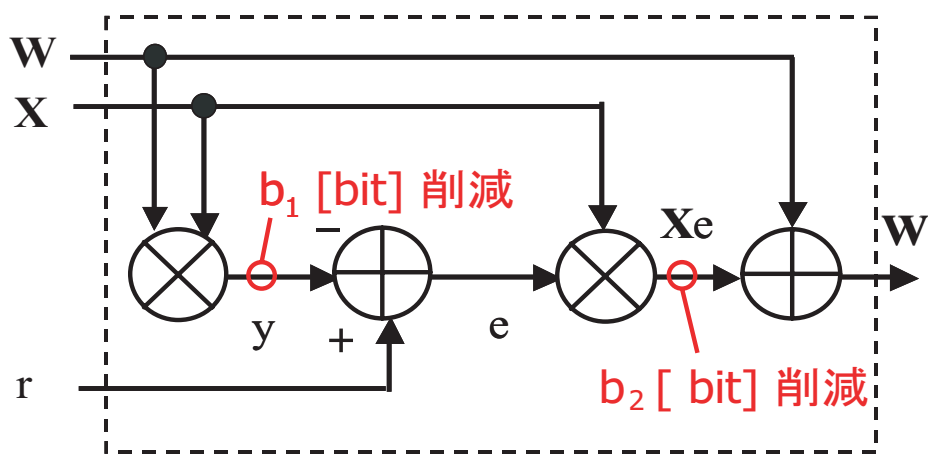


図 2.2: LMS アルゴリズムのブロック図

図 2.3 は，表 3.2 のシミュレーション諸元において，SNR を 5,10,20 のそれぞれの環境で LMS のウエイトベクトルのワード長を変えて計算したときの収束特性の変化である．ただ

し，縦軸は収束後の平均二乗誤差 (MSE) とし，19000 から 20000 サンプル間の MSE の平均を取った値である．

表 2.1: シミュレーション諸元

アレー形状	4 素子等間隔 ($\lambda/2$) リニアアレー
到来波数	無相関 2 波
到来方向	所望波 30° , 干渉波 -60°
D/U 比	0[dB]
SNR	5,10,20[dB]
ステップサイズ	$\mu=0.125, 0.250, 0.500(= \frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^3})$

図 2.3 より，SNR が大きいほど，浮動小数点演算の精度に近づけるためにはワード長を長くする必要があることが分かる．これは，ワード長が短いと切捨て雑音の影響が信号成分に含まれる内部雑音成分よりも大きくなるためにウェイトが本来収束すべきウェイトからずれて収束してしまうためと考えられる．第 3 章で SNR を 5dB から 20dB まで変えて各アルゴリズムの収束特性を比較するので，SNR が 20dB のときに浮動小数点演算の収束時の MSE と誤差が生じないようにワード長を選択する．また，SNR が 20[dB] でステップサイズが $\mu=0.125, 0.250, 0.500(= \frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^3})$ のそれぞれにおいて，ワード長を変化させたときの収束時の MSE を調べたところ，図 2.4 となった．この図より，ステップサイズが小さいほど浮動小数点演算との精度に差が生じることが分かる．するためにはワード長を長くする必要があることが分かる．第 3 章において LMS のステップサイズを 0.250 よりも大きくするので，ステップサイズが 0.250 のときに浮動小数点演算との誤差がないようにウェイトベクトルのワード長を選択しておけば，ステップサイズを 0.250 よりも大きくしても，浮動小数点との誤差はあまり大きくならないことが予想される．よって，ウェイトベクトルのワード長は 17ビットとした．

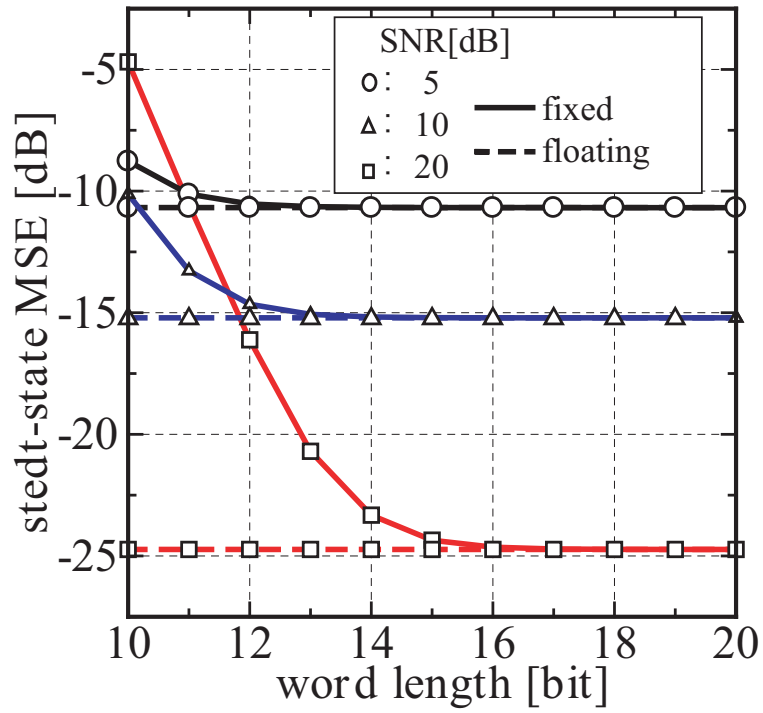


図 2.3: ワード長による浮動・固定小数点演算の収束特性 (SNR)

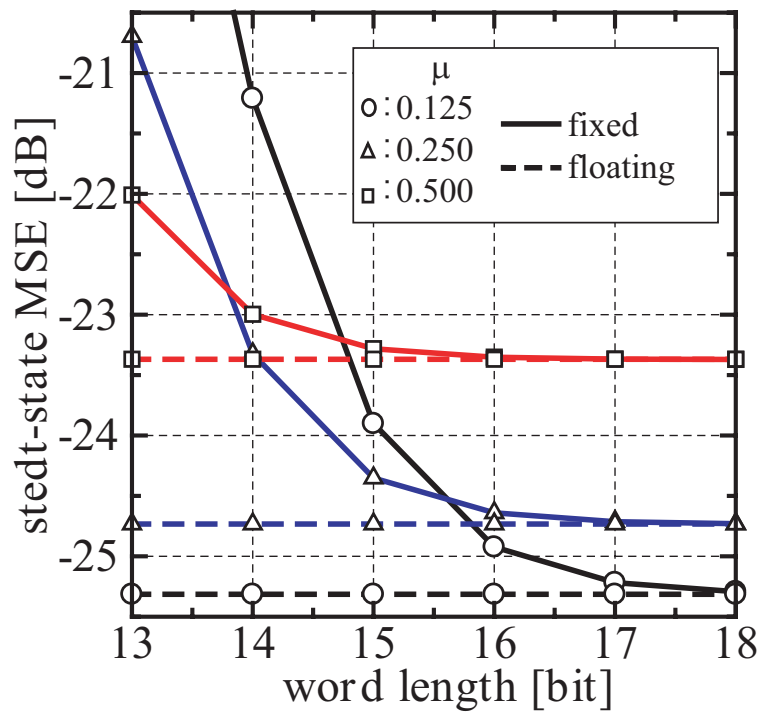


図 2.4: ワード長による浮動・固定小数点演算の収束特性 (ステップサイズ)

2.1.2 ワード長削減

入力ベクトル X, W, r のワード長が決定したので，式 (1.23) を固定小数点で計算することが可能となったが，回路の規模・動作速度の改善を図るために，ワード長の削減を考える．式 (1.23) では，ウェイトベクトル w と入力ベクトル x 、誤差 e と入力ベクトル x の 2 箇所で乗算が行われている．通常，固定小数点演算における乗算では，

$$m[\text{bit}] \times n[\text{bit}] = m \times n[\text{bit}] \quad (2.1)$$

となるので，ビット数が非常に大きくなる．そこで，この長くなったワード長を短くすることを考える．それぞれの出力 y 、 xe の削減ビット数を b_1, b_2 とし，表 3.2 のシミュレーション諸元において，LMS を計算したときの結果を図 2.5 に示す．図の縦軸は図 2.3 と同じで収束後の MSE として，19000 から 20000 サンプル間の MSE の平均値である．図より， b_1 をできるだけ大きくし， b_1 と b_2 の和が最大になるように選んだ結果， b_1, b_2 はそれぞれ，16, 5 ビットとした．このように選択した理由は，出力 y は次の乗算である $x \times e$ への入力になるからである．乗算器は回路規模が大きいため，できるだけ小さくするために入力のワード長を短くする必要がある．

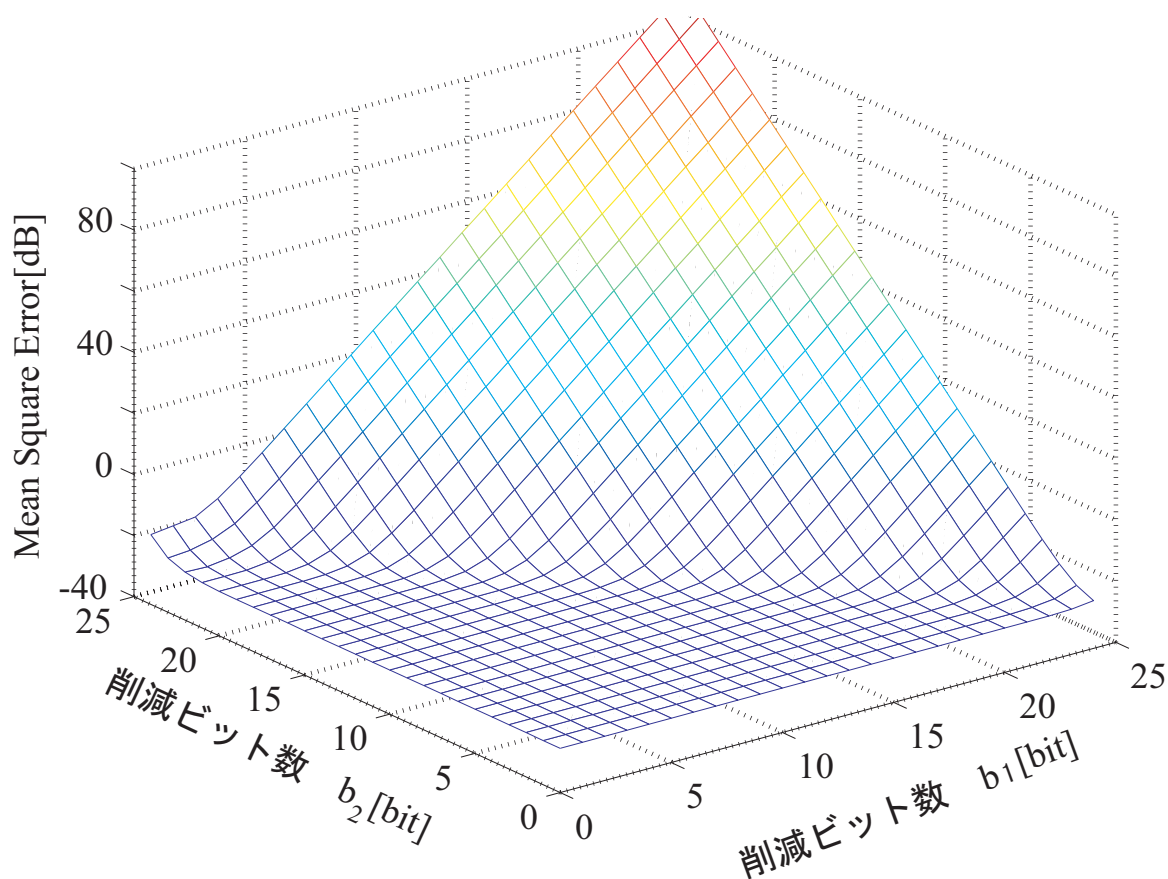


図 2.5: 削減ビットによる固定小数点演算の収束特性比較

2.2 N-LMS アルゴリズム

2.2.1 除算演算とビットシフト演算

N-LMS は、図 2.6 のブロック図が表すことができ、式 (1.26) より除算が存在するため、除算器を用いる必要がある。しかし、除算器は回路規模が大きくなってしまい、それに伴い、動作速度が劣化してしまう。そこで、この問題を取り除くためにビットシフトを用いる。ビットシフト演算は以下のように行う。

$$\|\mathbf{X}\|^2 \simeq \begin{cases} 2^n & (2^{n-1} \leq \|\mathbf{X}\|^2 < 2^n + 2^{n-1}) \\ 2^{n+1} & (2^n + 2^{n-1} \leq \|\mathbf{X}\|^2 < 2^{n+1}) \end{cases} \quad (2.2)$$

ただし、 n が $n < 2^{-9}$ のときは、 2^{-9} へ近似している。このようにすることで、入力ベクトルの絶対値が 0 に近づくことを防ぐため、収束特性が不安定になることを防ぐ。ビットシフト演算を用いた N-LMS のブロック図を図 2.7 に示す。

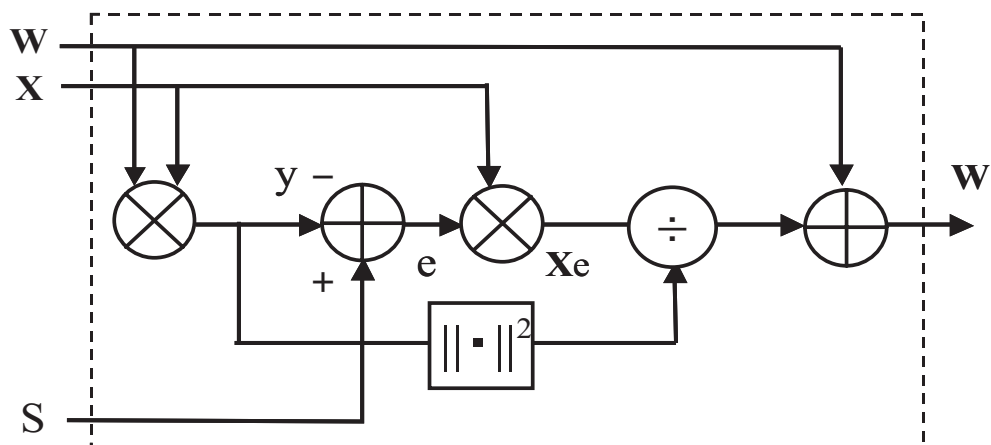


図 2.6: NLMS アルゴリズム (除算器) のブロック図

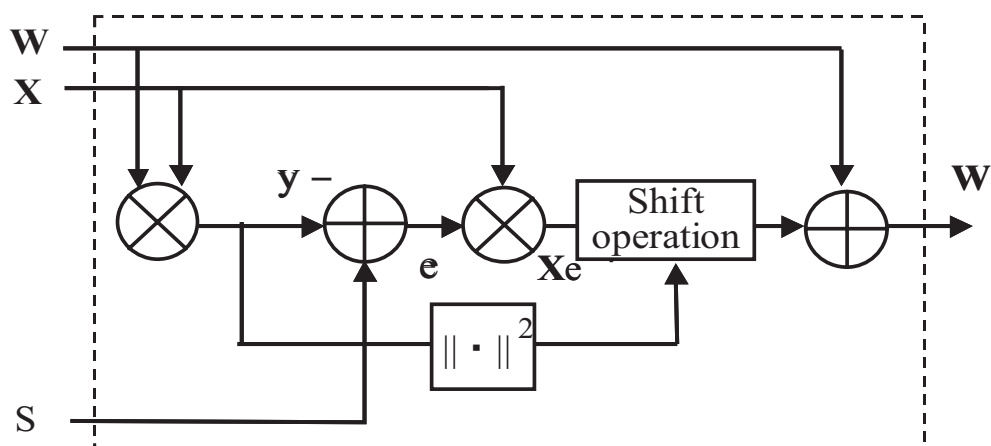


図 2.7: NLMS アルゴリズム (ビットシフト) のブロック図

2.2.2 各種パラメータのワード長

LMSと同様にして、入力情報 X 、参照信号 r の固定ワード長を12ビットとしたときの、内部演算変数であるウェイトベクトル w について検討した。シミュレーション諸元は表3.2においてワード長を一番長く必要とする $SNR = 20dB$ のときで、 $\mu_0 = 0.25(= \frac{1}{2^2})$ として浮動小数点との誤差を調べた。ステップサイズを $\mu_0 = 0.25(= \frac{1}{2^2})$ としたのは、第3章の収束特性の比較の際にステップサイズを $\mu_0 = 0.25(= \frac{1}{2^2})$ 以上の範囲で変化させるからである。図2.4より、同じワード長でもステップサイズが大きくなると、浮動小数点演算との誤差が小さくなるので、 $\mu_0 = 0.25(= \frac{1}{2^2})$ で誤差が十分小さくなるワード長を選択すれば、ステップサイズを大きくしても精度は劣化しないと考えられる。よって、図2.8よりウェイトベクトルのワード長は、17ビットを選択した。

次に、回路規模・動作速度の改善のためにワード長の削減を考える。ウェイトベクトル w と入力ベクトル x 、誤差 e と入力ベクトル x 、入力ベクトル x の絶対値の二乗の3箇所 wx 、 $x e$ 、 $|x|^2$ で乗算が行われている。それぞれの削減ビット数を b_1, b_2, b_3 とする。図2.9は、この3つの出力のワード長の削減のうち、どれを削減することが特性の劣化に一番寄与するかを調べたもので、3つのうち2つを固定して残りの1つの変数のワード長を短くしていったときの収束後のMSEの平均値を調べたものである。図より、除算演算を用いた場合は $|x|^2$ の削減が、特性に変化を与える影響が強いことが分かった。よって、 $|x|^2$ の削減ビット数 b_3 を優先して決めた。今回は、図より11ビットとした。ビットシフトの場合、 $|x|^2$ の削減による影響がないのは、 $|x|^2$ は、比較的大きい値になるので、式(2.2)において、上位ビットで2のべき乗へ近似が行われるため、下位ビットが削減されても影響がないためと考えられる。ビットシフトにおいても、除算器と同様に b_3 を11bitとした。最後に、LMSのときと同様にして、 b_1, b_2 を定めたところ、除算器・ビットシフトを用いた場合ともには16, 4となった。

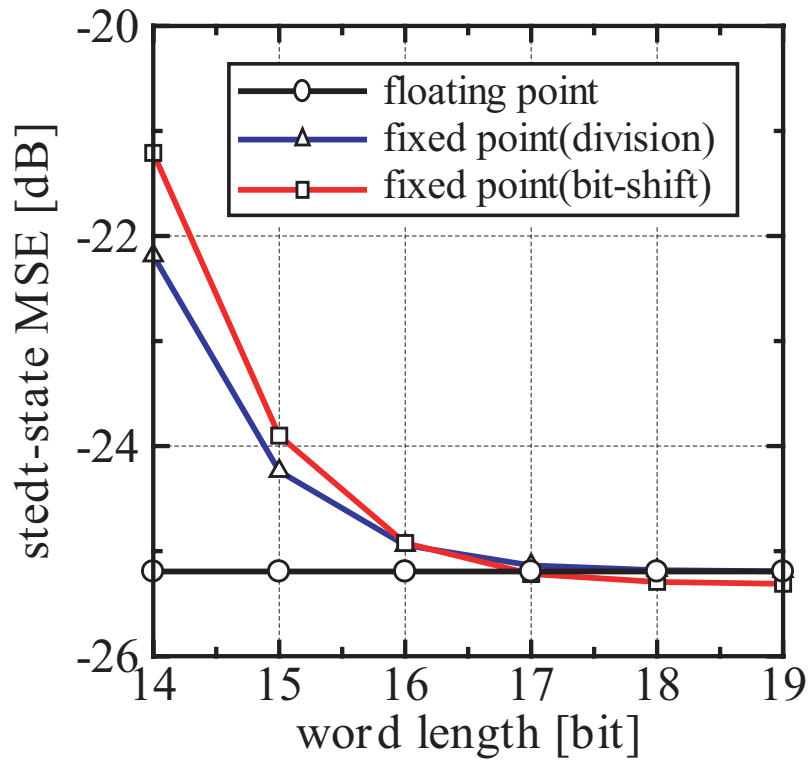


図 2.8: ウェイトベクトルのワード長による浮動・固定小数点演算の収束特性 (N-LMS)

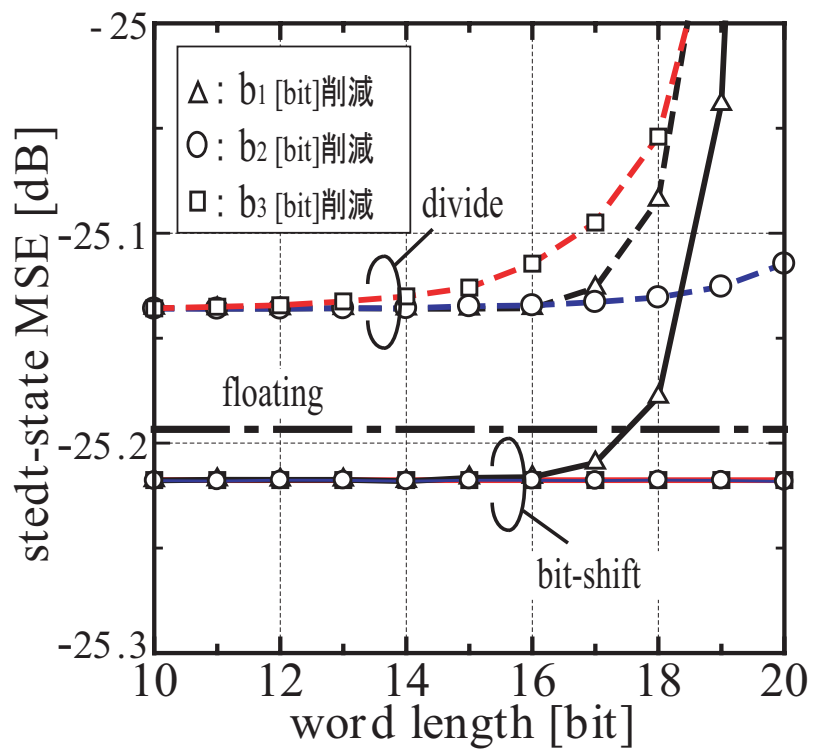


図 2.9: ビット削減による浮動・固定小数点演算の収束特性 (N-LMS)

第 3 章

FPGA 実装と実時間収束特性比較

本章では，2章でシミュレーションにより求めた各種パラメータのワード長を用いて，実際に回路を VHDL で記述し，FPGA に実装する．それによって，一回のウェイト更新に要する処理時間が分かるので，実時間による収束特性を行うことが可能となる．また，比較する際，ステップサイズや忘却係数の値によって，収束特性は大きく変化するので，誤調整と呼ばれる収束特性を特徴づけるパラメータの1つを用いて決定する．

3.1 FPGA 実装

3.1.1 LMS アルゴリズム

2章のシミュレーションにおいて，想定したシステムから，入力ベクトル x ，参照信号 s を 12ビットとしたとき，できるだけ少ないワード長で，浮動小数点演算による計算結果との誤差が生じないようなワード長を検討したところ，ウェイトベクトル w は 17ビットとなった．また，2.1節でワード長を短くすると回路規模を小さくできることから，回路規模削減を目的として乗算器からの出力のワード長を短くすることを行った，シミュレーションより， b_1 と b_2 は，それぞれ 16, 5 となった．よって，そのワード長と図 2.2 をもとに実際に実装した．ワード長を削減する前後の回路規模・動作速度を表 3.1 に示す．また，用いたシステムの仕様を表 3.2 に示す．

表 3.1: 削減による実装結果比較

	削減前	削減後
回路規模 (総 LE に対して) [%]	52	14
DSP Block (全体に対して) [%]	100	80
最大動作周波数 [MHz]	37.5	84
クロック数 [clk/iteration]	12	12

表 3.2: 信号処理ユニットの仕様

AD converter	Resolution	12bit
	Sampling Rates	32MHz
DA converter	Resolution	14bit
	Sampling Rates	40MHz
FPGA	Altera Stratix EP1S25	600,000 Gates
CPU	Hitachi SH4	200MHz
	Operating System	NetBSD
Interface	TCP/IP Ethernet	100 BaseT

表より，ワード長を削減することによって，回路規模は4分の1に，動作速度は2倍早くなっていることが分かる．

3.1.2 N-LMS アルゴリズム

次に，N-LMSの実装を考える．前章で求めたワード長と図 2.6, 2.7をもとに実装した．図 2.2と比較して入力ベクトルの絶対値の計算と除算器またはビットシフト演算部が加わるために回路規模は増加する．特に，除算器を用いた場合は大分特性が劣化すると考えられる．実際に，実装した結果を表 3.3に示す．表より，除算器を用いた場合，LMSと比較して回路規模は，2倍になり，動作周波数も5分の1になり，一回の更新に要する時間は3倍になってしまう．

表 3.3: NLMS 実装結果比較

	除算器	ビットシフト
回路規模 (総 LE に対して) [%]	35	17
DSP Block (全体に対して) [%]	80	80
最大動作周波数 [MHz]	16	76.3
クロック数 [clk/iteration]	13	12

また，ビットシフト演算単体では，60万ゲート全体の3.3%となり，除算器単体では，21.3%となった．除算演算は，回路規模・動作速度ともに今回実装したものよりも改善はできるが，ビットシフト演算ほど特性は良くなれないと考えられるので，本論文では，ビットシフト演算を用いた N-LMS を用いて以降，各種アルゴリズムとの比較を行う．

3.1.3 実装結果

ここまでの実装結果を，既に本研究室で FPGA 実装が行われている RLS アルゴリズムのデータも合わせてまとめると，表 3.4 となる [5]．表のクロック数と所要時間は一回のイタレーションに必要な値であり，所要時間はクロック数と最大動作周波数の逆数の積から求めている．

表 3.4: 各種アルゴリズム実装結果

	LMS	NLMS(除算器)	NLMS(ビットシフト)	RLS
回路規模(総 LE に対して) [%]	14	35	17	40
DSP Block(全体に対して) [%]	80	80	80	
最大動作周波数 [MHz]	84	16	76	80
クロック数 [clk/iteration]	12	13	12	400
所要時間 [μ s/iteration]	0.143	0.813	0.157	5

表 3.4 より、回路規模は LMS, NLMS, RLS の順に大きくなり、それに伴って、1 回のウェイトベクトル更新に要する時間が大きくなっていることが分かる。動作速度を上げようとするとレジスタを挟むことになる、すると、一回の更新に要するクロック数が多くなるためにやはり、所要時間が大きくなる結果となる。このように、各アルゴリズムにおいて、一回更新するのに要する時間が異なるためにイタレーション回数で比較するのではなく実時間で比較を行うべきだと考えられる。そこで、最高動作周波数の逆数と一回の更新に要するクロック数の積は一回の更新に要する時間となるので、この値と収束するまでのイタレーション回数を掛けることによって実時間を算出する。

3.2 実時間収束特性比較

収束時間を比較するにあたり、いくつかのことを考える必要がある。一つ目は、収束をいつと見なすかを定めること。二つ目は、収束特性に大きく影響する LMS, N-LMS のステップサイズ μ, μ_0 と RLS の忘却係数 λ の検討である。まずは収束時間を定める。

3.2.1 収束時間の決定

収束時間は次のように定める。図 3.1 のように、平均自乗誤差が収束した後、1000 サンプルで平均を取り、その値を閾値とする。今回実装した LMS の場合、1000 サンプル数分つまり実時間で 2500 [μ s] で平均ととることになる。次に、図 3.2 のように、平均自乗誤差を一定の間隔で平均を取っていきその値が、先ほど定めた閾値に達したときの時間を収束時間と定めた。今回用いた間隔は、50 サンプルとした [6]。

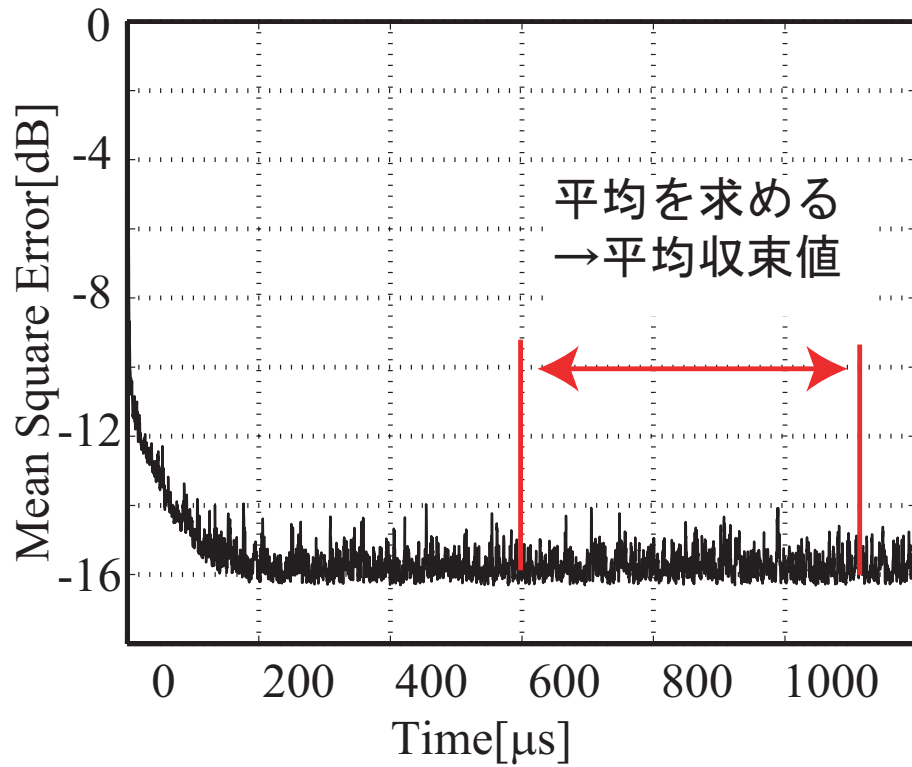


図 3.1: 閾値の算出

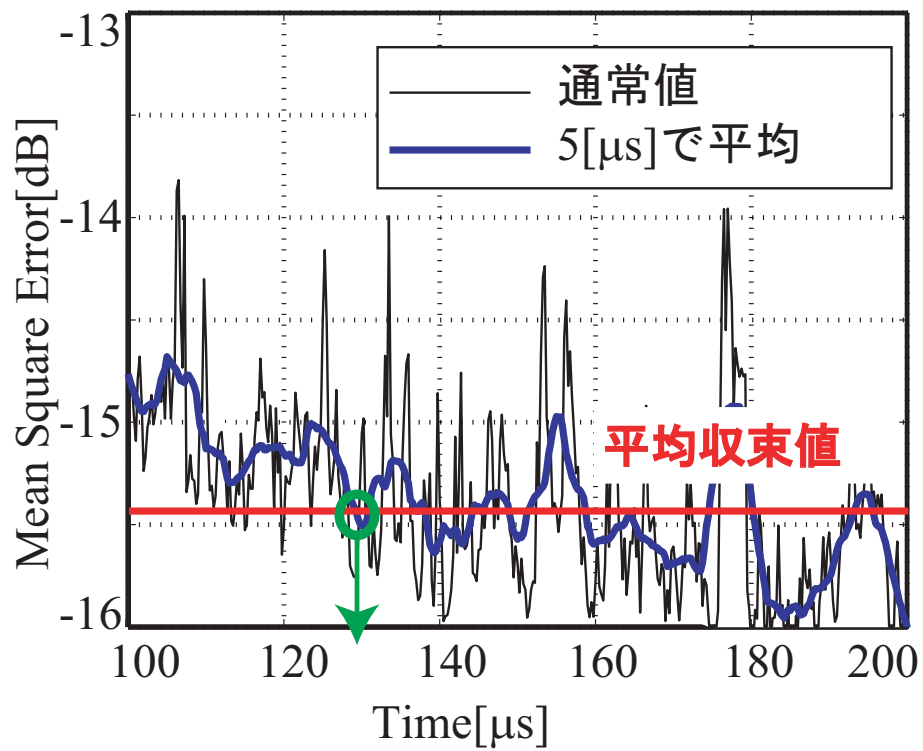


図 3.2: 収束時間

3.2.2 誤調整

各アルゴリズムの収束特性を比較するにあたり、ステップサイズ、忘却定数をどのように選択するかが問題となる。つまり、安定度を等しくする必要がある。これは、以下の式で与えられる、誤調整 M (Misadjustment) が等しくなるように、ステップサイズを選択する [6]。図 3.3 に示すように、十分な回数イタレーションを行っても、ウェイトは式 (3.1) で表される最適ウェイト W_{opt} 付近の近傍で揺らぎを生じる。

$$W_{opt} = R_{xx}^{-1} r_{xr} \quad (3.1)$$

ただし、 R_{xx} , r_{xr} は、信号の相関行列と相関ベクトルである。上記の理由より、平均自乗誤差 MSE と最小平均自乗誤差 MMSE に誤差が生じる。その誤差は過剰平均自乗誤差 (Excess MSE) と呼ばれる。誤調整は、その誤差の大きさを表したもので、アルゴリズムの適応能力の測度として用いられる。収束後、無限サンプルにおける ExcessMSE を steady-state ExcessMSE と言うが、今回、この steady-state MSE を 19000 サンプルから 20000 サンプル間での十分収束している 1000 サンプル間での平均値として定める。その平均値と最適解のときの MSE つまり MMSE との比から誤調整を算出し、以下の図に示す。RLS アルゴリズムの忘却係数を $\lambda = 0.875$ とすると、図 3.6 より、誤調整は 30% となるので、LMS, N-LMS アルゴリズムとも同じ値になるようにステップサイズを定めて比較する [7]。

$$M = \frac{\text{average}(MSE - MMSE)}{MMSE} \quad (3.2)$$

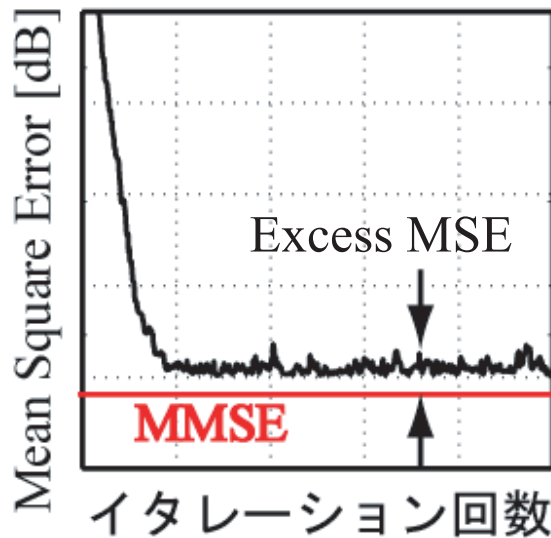


図 3.3: MMSE と MSE

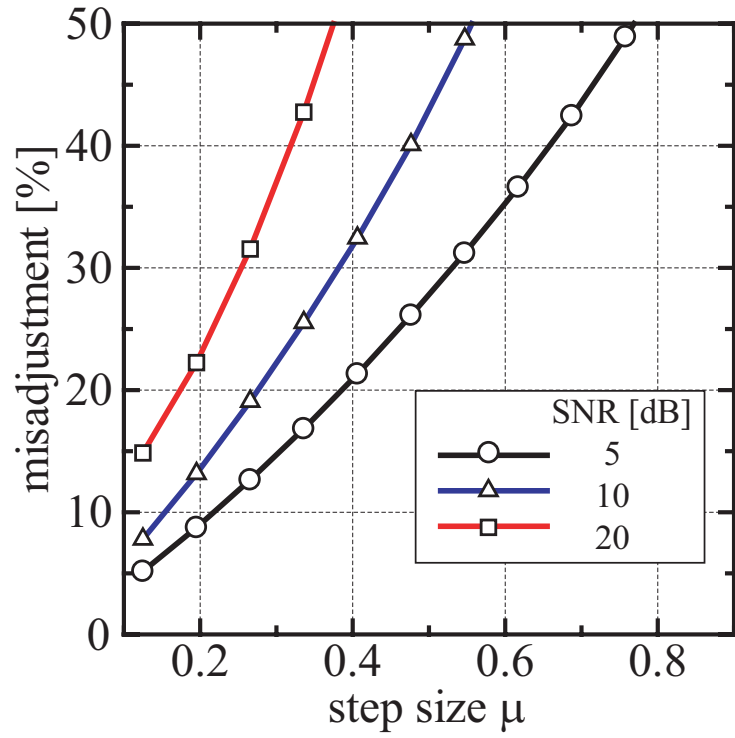


図 3.4: 誤調整 (LMS)

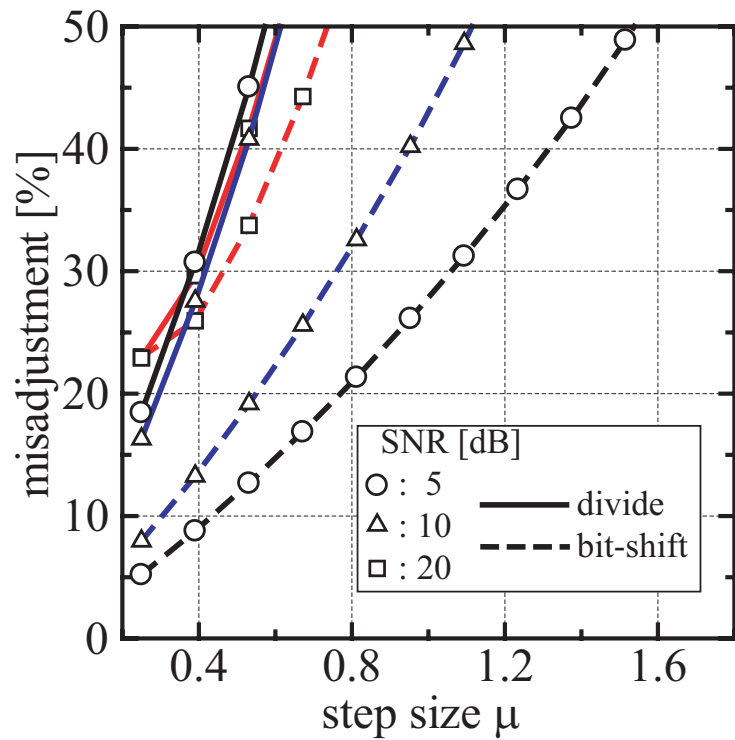


図 3.5: 誤調整 (NLMS)

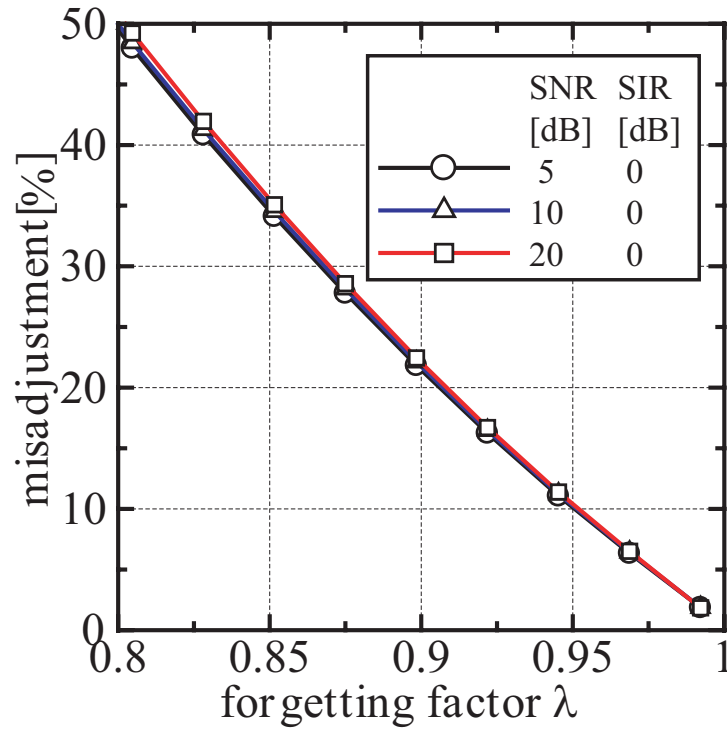


図 3.6: 誤調整 (RLS)

3.3 比較結果

比較する条件としては、表 3.2 を基本的な設定として、 $D/U=0$ [dB] で固定して、 $SNR=5,10,20$ [dB] と変えた。実時間に対する平均二乗誤差と出力 SINR の収束特性をそれぞれ、図 3.7、図 3.8 に示す。平均自乗誤差、出力 SINR どちらも、SNR が低いときは LMS, N-LMS の方がわずかに早く収束し、高いときは RLS の方がだいぶ収束が早いことが分かる。閾値に達した時間を表 3.5 に示す。RLS は、SNR に対して収束時間はあまり変化しない。しかし、LMS は、SNR に対する収束時間に差が生じてしまっている。ただし、ここで比較している RLS アルゴリズムは論文 [5] の実装結果のウェイト更新所要時間を用いた倍精度の RLS アルゴリズムである。

表 3.5: SNR に対する各種アルゴリズムの収束時間

	LMS	NLMS	RLS
5 [dB]	21.3	23.4	145
10[dB]	45.7	50.3	130
20[dB]	412.8	524	215

ただし、単位は $[\mu s]$

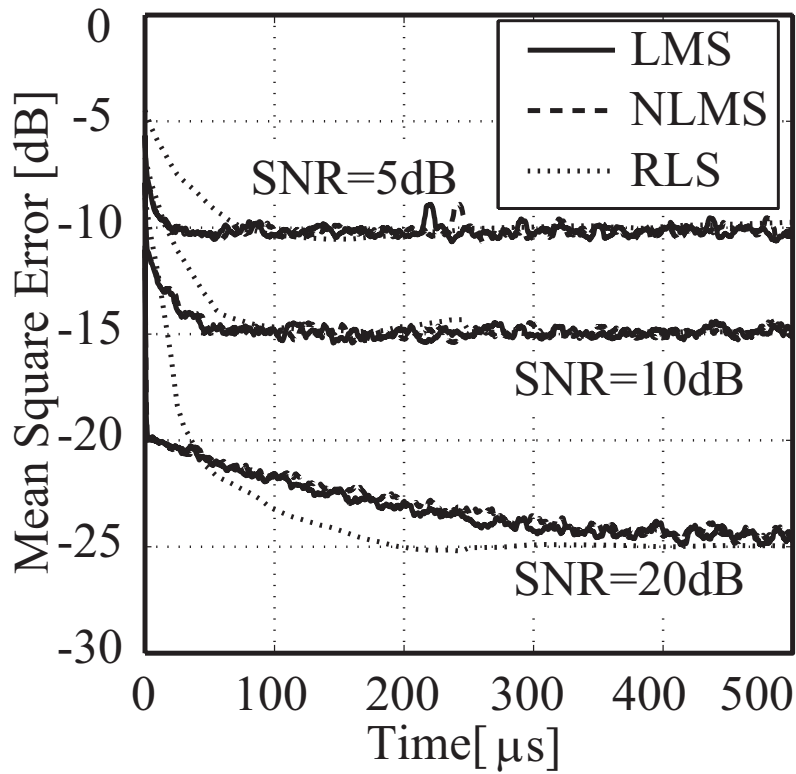


図 3.7: SNR に対する MSE の実時間収束特性

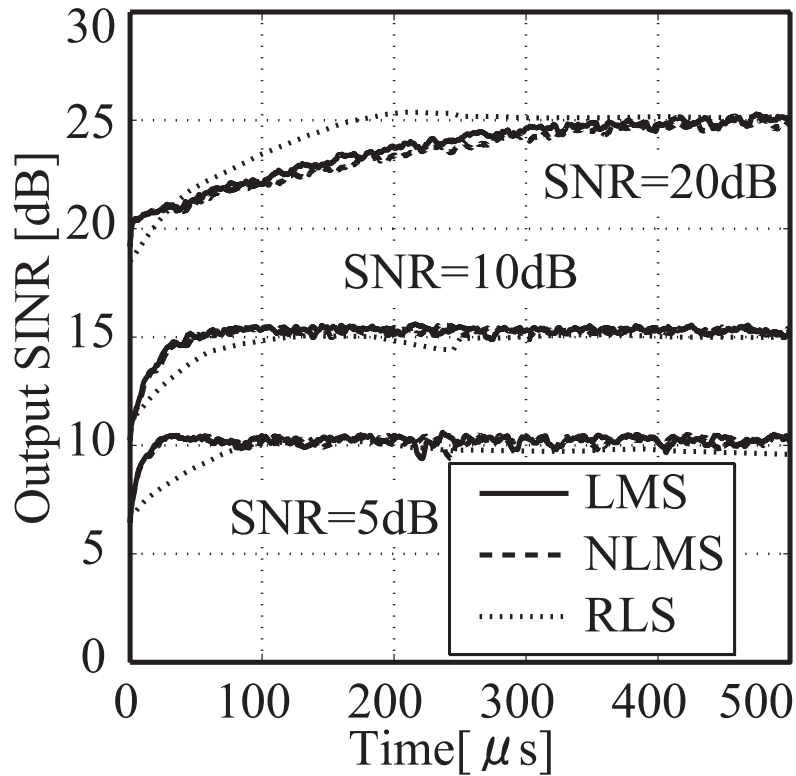


図 3.8: SNR に対する SINR の実時間収束特性

第 4 章

結論

本論文では、MMSE アダプティブアレーの最適化アルゴリズムである、LMS, N-LMS, RLS の収束特性をイタレーション回数でなく、計算量を考慮した実時間による比較を行った。実時間で比較するので、各アルゴリズムのウェイトベクトルを一回更新するのに要する時間を算出する必要がある。そこで、比較対象となる 3 つのアルゴリズムのうち、LMS, N-LMS の 2 つを FPGA に実装した。固定小数点演算による実装を行うので、入力ベクトル、内部演算変数が取り扱う一区切りのデータの大きさであるワード長を決定する必要がある。シミュレーションにより、そのワード長で表現できる精度で計算した固定小数点演算の結果と浮動小数点演算による結果との誤差を比較することでワード長を決定した。また、N-LMS の除算器による回路規模増大の問題をビットシフトを用いることで改善した。実時間の比較の際には、ステップサイズ・忘却係数の選択という問題を解決するために収束特性の特徴を表現するパラメータである誤調整を用いた。

1. シミュレーションから、浮動小数点演算と誤差が生じないようにワード長を短くした結果、回路規模・動作速度が改善された。LMS においては、回路規模は 3 分の 1 になり、動作周波数は 2 倍早くなった。このことから、ワード長を短くし回路を実装することは、回路規模だけでなく、動作速度も改善されることが分かる。
2. N-LMS において、除算演算をビットシフト演算で行うことで、回路規模・動作速度が改善され、回路規模は 2 分の 1 となり、動作周波数は 5 倍となった。
3. 誤調整が等しくなるようにステップサイズ・忘却係数を定め、実時間での比較を行った結果、条件によっては、LMS, N-LMS は、RLS よりも早く収束することもあり、実時間による比較では、LMS, N-LMS は RLS に近い早さで収束することが分かった。このことから、計算量は収束特性に大きく影響を与えることが分かった。

以上のことが、本研究の結果より得られた。

謝辞

本研究を進めるにあたり、厳しくかつ丁寧に御指導下さった新井宏之教授に深く感謝致します。

また研究全般に渡って御指導下さった株式会社ブレインズの金ミン錫様に深く感謝致します。

最後に研究生生活を共に過ごした新井研究室，久我研究室，市毛研究室の皆様に深く感謝致します。

参考文献

- [1] 菊間信良 , アレーアンテナによる適応信号処理 , 科学技術出版 , 1999 年
- [2] N. Kikuma and K.Takao , “Effect of Initial Values od Adaptive Arrays,” IEEE, Trans. Aerosp. & Electron. Syst., vol. AES-22, no. 6, pp.688–694, Nov. 1986.
- [3] M. Tarrab and A. Feuer, “Convergence and Performance Analysis of the Normalized LMS Algorithm with Uncorrelated Gaussian Data, ” IEEE Trans. Signal Processing, Vol. 41, pp.2811–2825, 1993.
- [4] L. Godara , “Applications of Antenna Arrays to Mobile Communications, Part 2: Beam-Forming and Direction-of-Arraival Considerations,” Proc. IEEE, vol. 85, no. 8, pp.1195–1245, Aug. 1997.
- [5] 中島敦, “RLS アルゴリズムを用いた MMSE アダプティブアレーアンテナの FPGA 実装,” IEICE Trans. Commun., Vol. J88-B, NO. 9, pp.1772–1779, Sep. 2005.
- [6] W. A. Gardner and W. A. Brown , “A new algorithm for adaptive arrays,” IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-35, pp.1314–1319, 1987.
- [7] G. Manolakis, K. Ingle and M. Kogon, “Statistical and Adaptive SIGNAL PROCESSING,” Mc Graw Hill, 2000.
- [8] Dimitris G.Manolakis, Vinay K.Ingle, Stephen M.Kogon, Statistical and Adaptive SIGNAL PROCESSING, McGraw-Hill Higher Education pp.274-431.
- [9] J. Winter, “Smart Antennas for Wireless Systems,” IEEE Personal Commun., vol. 5, no. 1, pp. 23–27, Feb. 1998.
- [10] L. Godara , “Applications of Antenna Arrays to Mobile Communications, Part 2: Beam-Forming and Direction-of-Arraival Considerations,” Proc. IEEE, vol. 85, no. 8, pp.1195–1245, Aug. 1997.

- [11] C.Ward, P.Hargrave, J.G.McWhirter, "A novel algorithm and architecture for adaptive digital beamforming," IEEE Trans. Antennas & Propag., vol.AP-34, no.3, pp.338-346, 1986
- [12] M.S.Kim, K.Ichige and H.Arai, "Design of Jacobi EVD Processor Based on CORDIC for DOA Estimation with MUSIC Algorithm", IEICE Trans. Commun., vol.E85-B, no.12, pp. 2648-2655, Dec, 2002.
- [13] M.Kim, "A Study of Implementation of Digital Signal Processing for Adaptive Array Antenna", 横浜国立大学工学研究科 新井研究室 修士論文, 平成 14 年 2 月.
- [14] 村松慎太郎 "アダプティブアレー用送受信機のデジタル処理部の構成法に関する研究" 横浜国立大学工学研究科 新井研究室 修士論文, 平成 15 年 2 月.
- [15] 藤辰男 "アダプティブアレーのリアルタイムデジタル補正に関する研究" 横浜国立大学工学部 新井研究室 卒業論文, 平成 15 年 2 月
- [16] 中島淳 "複数の変調方式に対応するデジタル復調器に関する研究" 横浜国立大学工学部 新井研究室 卒業論文, 平成 16 年 2 月
- [17] 松本直也 "同期処理を含めた RLS アルゴリズムの FPGA 実装に関する検討" 横浜国立大学工学研究科 市毛研究室 修士論文, 平成 16 年 2 月.
- [18] 横山良晃 "RLS シストリックアレーの FPGA 実装法に関する研究" 横浜国立大学工学研究科 横浜国立大学工学部 新井研究室 卒業論文, 平成 17 年 2 月