

修士論文

リアルタイム同期処理器を含めた RLSアダプティブアレーのFPGA実装

FPGA Implementation
of RLS Adaptive Array
with real-time Synchronization Process

指導教官 新井 宏之 教授

平成18年2月7日提出

横浜国立大学大学院 工学府

物理情報工学専攻 電気電子ネットワークコース

04GD160 中島 淳

要約

高速移動通信における伝搬路の環境では，マルチパスフェージングや周波数利用効率などの制約があるために通信容量や通信品質が劣化してしまう．この問題に対して，干渉波や遅延波の除去を目的としたアダプティブアレーアンテナの研究が盛んに行われており，陸上移動体通信環境では到来波の到来方向や到来波数を正確に把握することが困難であるため，これらの情報を必要としない MMSE (Minimum Mean Square Error) アダプティブアレーアンテナが非常に有効であると考えられる．MMSE アダプティブアレーアンテナは構成の容易さに比べて演算速度が優れていることから，移動体通信技術において幅広く用いられている．

本論文では，MMSE アダプティブアレーアンテナの高速処理を目的として，そのアルゴリズムの FPGA (Field Programmable Gate Array) への実装法について検討を行う．並列処理を得意とする書き換え可能なフレキシブル DSP (Digital Signal Processor) として位置付けられる FPGA を用いることで処理速度を向上させることが可能であるが，FPGA は固定小数点での演算が適しているためアルゴリズムを構成する変数の固定ワード長によって演算精度が変化する．そこで，MMSE アダプティブアレーアンテナの最適化アルゴリズムの中で，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く演算精度が良いとされる RLS (Recursive Least Squares) アルゴリズムを FPGA に実装して処理の並列化およびパイプライン化を実現して高速処理を実現し，同期処理を含めた送受信系システムを構築することで実験的に固定小数点演算を用いた RLS アルゴリズムの干渉波抑圧能力を評価することを本論文の目的としている．

アレーアンテナの素子数を少なく限定した場合において入力情報及び内部演算パラメータの固定ワード長を決定し，固定小数点演算と浮動小数点演算を用いた場合の RLS アルゴリズムの演算精度の比較や，シフト演算による近似的な除算を用いた場合にアルゴリズムに与える精度誤差についてシミュレーションによって検証した．また実際に固定小数点演算を用いた RLS アルゴリズムを FPGA に実装し，同期処理部を含めた送受信系システムを構築して電波暗室にて伝搬実験を行いその性能を評価した．同期処理部ではリアルタイム性を向上させるために，M 系列符号を含めたトレーニングシンボルやフレーム構造を変更して精度向上をねらった同期処理器を提案し FPGA に実装させることで高速なリアルタイム処理を実現した．

目次

第 1 章	序論	1
1.1	背景と目的	1
1.2	FPGA の概要	4
1.3	アダプティブアレーアンテナの概要	5
1.4	MMSE アダプティブアレーの概要	8
1.4.1	LMS アルゴリズム	10
1.4.2	RLS アルゴリズム	12
第 2 章	固定小数点演算における RLS アルゴリズム	14
2.1	入力情報及び内部演算パラメータのワード長	14
2.2	乗算器と除算器の構成	20
2.3	角度依存性	25
2.4	考察	30
第 3 章	RLS アルゴリズムの FPGA 実装と実験系の検討	31
3.1	RLS アルゴリズムの処理の構成	31
3.2	FPGA 実装結果	33
3.2.1	複素数乗算コンポーネント	33
3.2.2	デジタル信号処理ユニットの仕様	35
3.2.3	RLS アルゴリズムの FPGA 実装結果	36
3.3	システムの構成	37
3.3.1	提案システムの処理の流れ	37
3.3.2	フレーム構成と同期処理の概要	38
3.4	実験による評価	40
第 4 章	同期処理器の FPGA 実装とリアルタイム実験	45
4.1	M 系列符号を用いたフレーム構成	45
4.1.1	M 系列符号	46
4.1.2	ダブルフレーム構成	47

4.2	シミュレーションによる同期処理器の検討	49
4.2.1	提案同期処理器の構成	49
4.2.2	M 系列を用いたトレーニングシンボルの効果	54
4.2.3	アップサンプリングレートによる精度比較	56
4.3	同期処理器の FPGA 実装	60
4.3.1	FPGA 実装結果	60
4.3.2	システムの構成	61
4.4	同期処理器を用いたリアルタイム実験	62
4.4.1	実験の構成	62
4.4.2	実験結果	64
	第 5 章 結論	67
	謝辞	69
	参考文献	70
	発表文献	73

第 1 章

序論

1.1 背景と目的

高速移動通信を実現するためには、図 1.1 に表されるような、建物などにより電波が反射、回折、散乱する多重伝搬路を通過するために生じるマルチパスフェージングや、同一周波数を利用している他のセルのユーザーによる同一チャネル干渉 (CCI: Co-Channel Interference)、伝搬遅延時間差などによって生じるシンボル間干渉 (ISI: Inter Symbol Interference) によって通信容量や精度が劣化してしまう。この問題に対して一般的に提案されている手法として、送信電力制御や誤り制御符号、ダイバーシチ技術が挙げられるが、近年では第 4 世代移動体通信の実現に向けて、アダプティブアレーアンテナを用いて干渉波や遅延波の除去することを目的とした研究が盛んに行われている [1]。

陸上移動体通信環境では、到来波の到来方向や到来波数を正確に把握することが困難であるため、これらの情報を必要としない MMSE (Minimum Mean Square Error) アダプティブアレーアンテナが適していると考えられる [2]。図 1.2 に示すように、MMSE アダプティブアレーアンテナでは、各アンテナ素子で受信された信号の重み付け和と、あらかじめ与えられた参照信号との平均 2 乗誤差を最小化するように重みが決定される。これは、見通し伝搬環境ならば複数の到来波のうち所望波方向にメインビームを向け、干渉波方向にヌルを向ける操作に等しく、これにより所望信号のデータのみを取り出すことが可能となる。このような MMSE アダプティブアレーアンテナを用いることで、チャネル容量増大や周波数利用効率の向上が期待できる。さらに、多数のマルチパスが存在する環境においても、アレー自由度以内の干渉波を除去し、所望信号については残りのアレー自由度を使ってダイバシティ合成するように働く [3]。

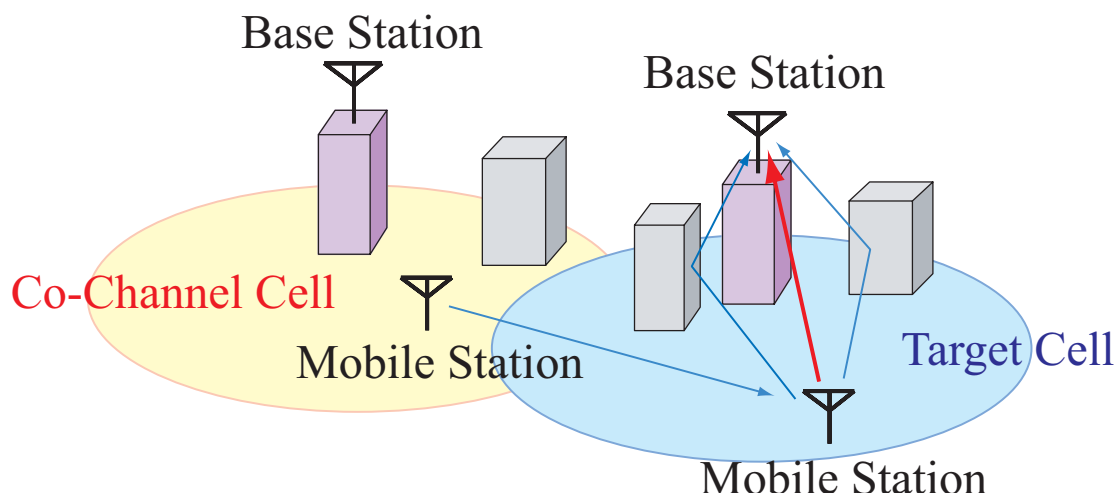


図 1.1: マルチパスフェージング環境

MMSE アダプティブアレーアンテナにおいて、最適ウェイトを近似計算するアルゴリズムとして、主に最急降下法に基づく LMS (Least Mean Square) アルゴリズム、サンプル値を用いた直接解法の SMI (Sample Matrix Inversion) アルゴリズム、再帰的最小 2 乗法の RLS (Recursive Least Squares) アルゴリズムの 3 つが挙げられ、またそれらを改良したのも提案されている [4]。LMS アルゴリズムは計算負荷が非常に少なく簡単な演算のみで実現できるものの、収束速度は遅く、到来波の到来方向が近い場合や到来波の強度比が大きい場合など、特定の条件において収束が極めて遅くなるという問題がある [5]。また、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 [6]。さらに、限られたイタレーション回数では、RLS アルゴリズムの方が LMS アルゴリズムよりも優れた収束特性を持っており、電波環境が変動する環境において通信品質を向上させるためには RLS アルゴリズムを用いることが望ましい。

MMSE アダプティブアレーアンテナのウェイト導出の計算では RLS アルゴリズムがよく用いられているが、計算機シミュレーション計算による報告が多い。専用デバイス上での実装例としては、ASIC (Application Specific Integrated Circuit) でシストリックアレーを用いて実装した報告が挙げられる [7]。シストリックアレーは並列パイプライン処理を行うことで高速処理を実現しているが、その一方で回路規模は増大してしまっている。また、RLS シストリックアレーの実装に関する詳細には触れられていない。

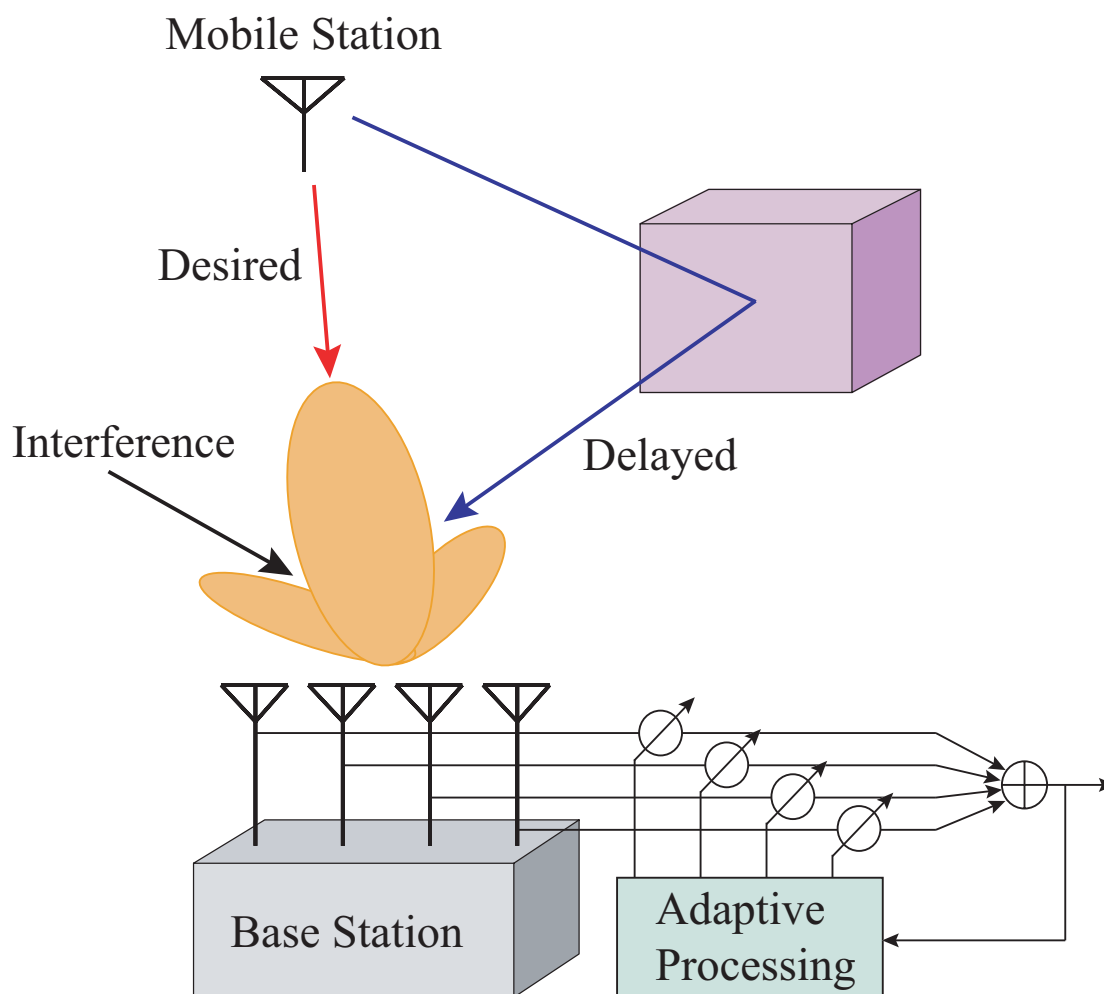


図 1.2: MMSE アダプティブアレーアンテナの概略図

本論文では RLS アルゴリズムの理論的な式からシフト演算による除算の近似などの固定小数点演算における工夫について取り上げ、それを実際に FPGA に実装するまでの詳細な過程を述べる。ASIC が特定用途向けの高速演算処理装置であるのに対し、書き換え可能なフレキシブル固定小数点演算 DSP (Digital Signal Processor) として位置付けられる FPGA (Field Programmable Gate Array) は、並列処理が可能であるのでコンパクトな構成で高速演算を実現するのに適しており、筆者達のグループでは既に様々な FPGA 実装に関する研究を行ってきている [8][9][10]。

本研究では、アレーアンテナの素子数を限定し、実装デバイスに FPGA を用いることで、高速かつ高精度に RLS アルゴリズムを計算することを目的としており、本論文の構成は第 1 章で FPGA や MMSE アダプティブアレーアンテナの概要について取り上げ、第 2 章では FPGA 実装に適した固定小数点演算を用いた RLS アルゴリズムについての検討を行う。固定小数点演算と浮動小数点演算との演算精度の比較や、入力情報及び内部演算パラメータのワード長についても検討する。第 3 章では実際に固定小数点演算を用いた RLS アルゴリ

ズムを FPGA に実装し，そのシステムの構成や実装結果，実験結果などを評価する．第 4 章で，リアルタイム処理を実現するためには同期処理器が FPGA 実装される必要を挙げ，その同期処理器の構成及び実装結果について述べ，実際に FPGA 実装された同期処理器を用いて RLS アルゴリズムのリアルタイム処理を行った結果を述べる．最後に第 5 章で結論とする．

1.2 FPGA の概要

本研究で用いる受信系ハードウェアのデジタル信号処理部は FPGA によって回路構成されている．FPGA は内部の論理回路を設計者が自由に変更することができるフレキシブルデバイスであるが，論理ブロックとそれらを接続する配線および接続スイッチで構成されていて，配線の結線情報は内部の SRAM などに蓄えられており，SRAM の内容を変更することにより内部結線を変更して任意の論理回路を作ることが可能となる．その構成を図 1.3 に示す．FPGA を設計するプログラムは HDL (Hardware Discription Language) が用いられ，本研究では VHDL (Very high-speed integrated circuit HDL) を用いている．

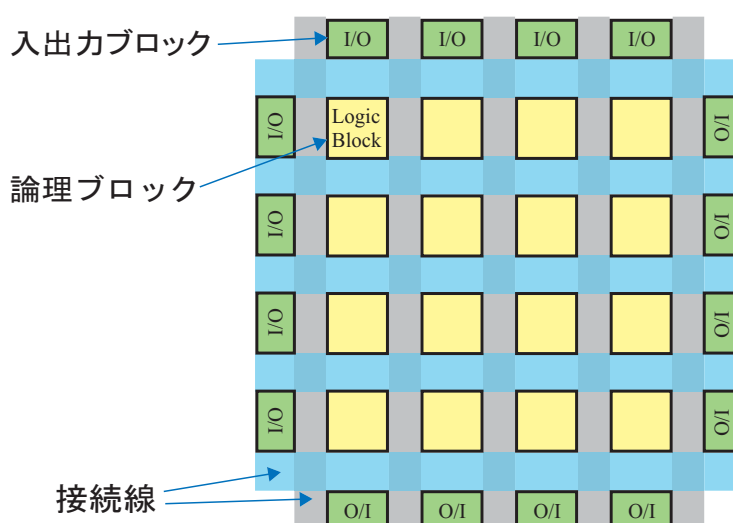


図 1.3: FPGA の構成

FPGA とは別に，ASIC などデジタル信号処理を行う専用 LSI として挙げられる．ASIC は信号処理のパフォーマンスは高いが，内部の論理回路は一度製作されると固定され，製作される費用も高くなり一般には大量生産に適している．FPGA は ASIC に比べると回路速度は劣り，実装可能な論理回路の容量も小さくなるが，その分比較的安価に抑えることができる．FPGA は，近年になって大容量かつ高速なモデルのもの登場してきており，試作や市場即応性が必要な分野に必須なデバイスとなってきている．

1.3 アダプティブアレーアンテナの概要

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，曲面状など様々な方法が考えられるが，ここでは K 素子よりなるリニア（直線状）アレーについて考える．

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 は基準点より測定した 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．ここで， $\lambda = c/f$ は搬送波の波長である．

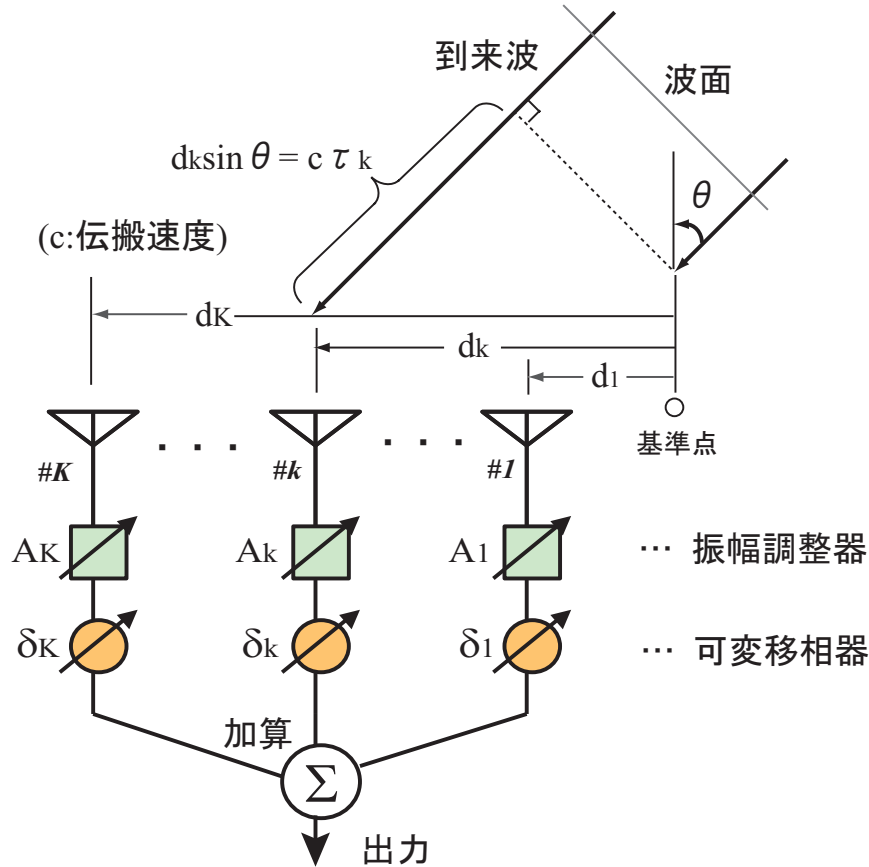


図 1.4: K 素子リニアアレー

図 1.4 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.

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

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て、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。

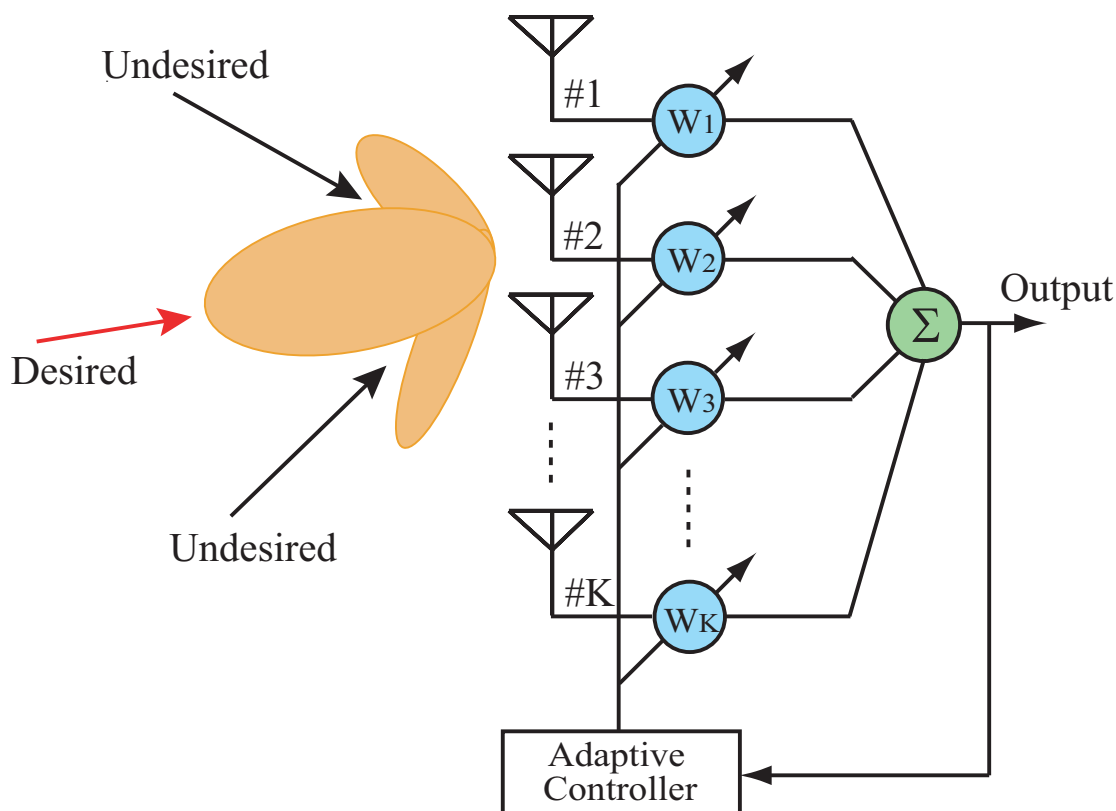


図 1.5: アダプティブアレーアンテナ

アダプティブアレーアンテナの主な構成を図 1.5 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) に大きく分類することができる。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プ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) 拘束付出力電力最小化法 (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。

1.4 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アレー出力信号との誤差(誤差信号)を最小にすることによって、ウエイトウエイトを最適化するシステムである。この方式はアダプティブヌル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い、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一方、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った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 .

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

ただし ,

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

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

$$= [E[x_1(t)r^*(t)], E[x_2(t)r^*(t)], \dots, E[x_M(t)r^*(t)]]^T \quad (1.16)$$

であり , \mathbf{R}_{xx} , \mathbf{r}_{xr} 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エイトベクトル \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 . 式 (1.13) はウエイトベクトル \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エイトベクトル \mathbf{w} の値 (最適ウエイト) は式 (1.13) のウエイトベクトルに関する勾配を零とおく , すなわち ,

$$\nabla_{\mathbf{w}} E[|e(t)|^2] = 0 \quad (1.17)$$

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 乗誤差のウエ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

$$\nabla_{\mathbf{w}} E[|e(t)|^2] = -2\mathbf{r}_{xr} + 2\mathbf{R}_{xx} \mathbf{w} \quad (1.18)$$

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

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xr} \quad (1.19)$$

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 (1.19) 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 .

$$E[|e(t)|_{min}^2] = E[|r(t)|^2] - \mathbf{w}_{opt}^T \mathbf{r}_{xr}^* - \mathbf{w}_{opt}^H r_{xr} + \mathbf{w}_{opt}^H \mathbf{R}_{xx} \mathbf{w}_{opt} \quad (1.20)$$

$$= E[|r(t)|^2] - \mathbf{r}_{xr}^H \mathbf{R}_{xx}^{-1} \mathbf{r}_{xr} \quad (1.21)$$

式 (1.19) により MMSE アダプティブアレーの最適ウエイトベクトルが与えられる . しかし , この式からウエ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エ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。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 アルゴリズムの複合的なアルゴリズムとも解釈できる。

1.4.1 LMS アルゴリズム

LMS アルゴリズムを拡張したアルゴリズムが RLS アルゴリズムであると解釈できるので, ここで LMS アルゴリズムの基本的な手法について述べる。

LMS の基本的な考えは, まず各アンテナ素子で受信された重み付け和と所望のアレー応答 (参照信号) を比較することで誤差推定を行うということと, アレーの重み係数 (ウエイト) を制御してこの誤差を最小化することで出力を所望の応答に近づけるということである。つまり, アレーに入力されてから現在までの全ての平均 2 乗誤差を最小化するようにウエイトを制御することが LMS の基本的な考えである。そこで, これらの定式化を行う。

図 (1.6) において, ある時点 i におけるアレーの出力は式 (1.22) のように示される。

$$e(i) = y(i) - \mathbf{w}^H(n)\mathbf{x}(i) \quad 1 \leq i \leq n \quad (1.22)$$

ただし,

$$\mathbf{x}(i) = [x_1(i), x_2(i), \dots, x_M(i)]^T \quad (1.23)$$

$$\mathbf{w}(n) = [\omega_1(n), \omega_2(n), \dots, \omega_M(n)]^T. \quad (1.24)$$

ここで, ウエイトベクトル $\mathbf{w}(n)$ が $1 \leq i \leq n$ 間で一定であると仮定する。 $n \times M$ の入力信号を行列で表すと,

$$\mathbf{X}^H(n) \triangleq [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)]$$

$$= \begin{bmatrix} x_1(1) & x_1(2) & \dots & x_1(n) \\ x_2(1) & x_2(2) & \dots & x_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ x_M(1) & x_M(2) & \dots & x_M(n) \end{bmatrix} \quad (1.25)$$

となり (M:アンテナの素子数), $n \times 1$ の参照信号ベクトルは次式で示される.

$$\mathbf{y}^H(n) \triangleq [y(1), y(2), \dots, y(n)] \quad (1.26)$$

また, 誤差信号は次式のように示される.

$$\mathbf{e}(n) \triangleq [e(1), e(2), \dots, e(n)]^H \quad (1.27)$$

式 (1.25),(1.26),(1.27) より式 (1.22) は次式のように定義できる.

$$\mathbf{e}(n) = \mathbf{y}(n) - \mathbf{X}(n)\mathbf{w}(n) \quad (1.28)$$

忘却係数 $\lambda^2 (0 < \lambda \leq 1)$ を用いて次式で示される指数関数的に重み付けされた $n \times n$ 行列を定義する.

$$\mathbf{\Lambda}(n) \triangleq \text{diag}[\lambda^{n-1}, \lambda^{n-2}, \dots, 1] \quad (1.29)$$

忘却係数は 1 以下の正定値であるが, たいていは 1 よりも少し小さい値を用いる. $1 - \lambda$ の逆数はそのアルゴリズムの記憶を測定することができ, λ が 1 の時はアルゴリズムはただの最小 2 乗法であるが, その記憶は無限大となる. 忘却係数を定義すると古いデータサンプルは忘れてしまい, そのアルゴリズムは時間的に変化していく非定常な電波環境の場合に適する.

以上より, 指数重み付誤差の 2 乗和は次式で示される. ただし, $\|\cdot\|$ はノルムを示す.

$$\begin{aligned} E(n) &= \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \\ &= \sum_{i=1}^n \lambda^{n-i} |y(i) - \mathbf{w}^H \mathbf{x}(i)|^2 \\ &= \|\mathbf{\Lambda}(n)\mathbf{e}(n)\|^2 \end{aligned} \quad (1.30)$$

$E^2(n)$ を最小にするようなウェイト $\mathbf{w}(n)$ を求めるアプローチ法として前節で述べた最急降下法に基づく LMS アルゴリズムや, サンプル値を用いた直接解法の SMI アルゴリズム, そして再帰的最小 2 乗法 (RLS アルゴリズム) がある.

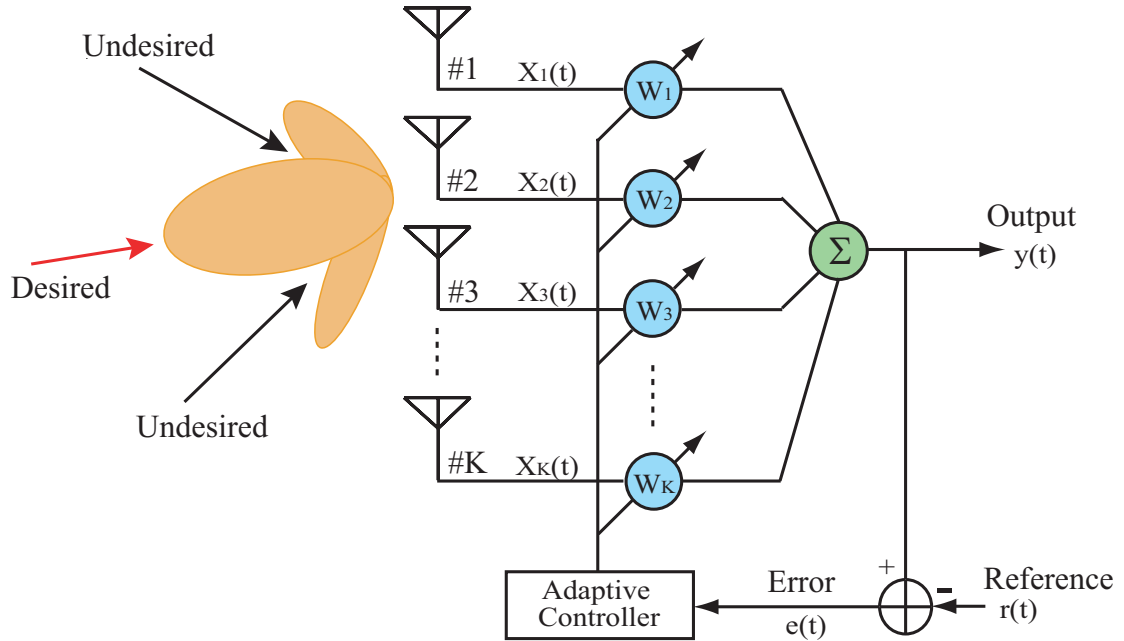


図 1.6: MMSE アダプティブアレー

1.4.2 RLS アルゴリズム

各アンテナ素子で受信された入力ベクトルを x , 出力信号を y , 忘却定数を q とおく . 出力信号 y は , ウェイトベクトル w と入力信号 x との内積

$$y(n) = \mathbf{w}^H(n-1)\mathbf{x}(n) \quad (1.31)$$

により求められる . ただし , H は複素共役転置を , 引数 (n) は n 番目のサンプル値を意味している . ここで , 各ベクトルや行列の次数はアレーアンテナの素子数 k に等しくなる . なお , w の初期値は 4 素子の場合において $[1,0,0,0]$ であり , これを更新して最適化を行うことで , 受信信号から所望波成分を取り出すことができる .

RLS アルゴリズムでは , まず以下の手順でカルマンゲインを計算する .

$$\mathbf{k}_n(n) = \mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n) \quad (1.32)$$

$$k_d(n) = q + \mathbf{x}^H(n)\mathbf{k}_n(n) \quad (1.33)$$

$$\mathbf{k}(n) = \frac{\mathbf{k}_n(n)}{k_d(n)} \quad (1.34)$$

ただし , T は転置 , \mathbf{R}_{xx}^{-1} は入力信号ベクトル x の相関行列の逆行列を表し , その初期値は 4 素子の場合で 4×4 の単位行列である .

このカルマンゲインを用いて，ウエイトベクトル \mathbf{w} は以下のように更新される．

$$\mathbf{w}(n) = \mathbf{w}(n-1) + e(n)^* \mathbf{k}(n) \quad (1.35)$$

ただし， $*$ は複素共役を表し， e は次式で表されるような参照信号 r と出力信号 y の誤差である．

$$e(n) = r(n) - y(n) \quad (1.36)$$

また，相関行列の逆行列 \mathbf{R}_{xx}^{-1} は逆行列演算を行うことを避け，以下のように再帰的に近似計算することができる [24] ．

$$\mathbf{R}_{xx}^{-1}(n) = \frac{1}{q} \{ \mathbf{R}_{xx}^{-1}(n-1) - \mathbf{k}(n) \mathbf{M}_a(n) \} \quad (1.37)$$

$$\mathbf{M}_a(n) = \mathbf{x}^H(n) \mathbf{R}_{xx}^{-1}(n-1) \quad (1.38)$$

式 (1.31) ~ (1.38) の計算を繰り返すことで，ウエイトベクトルは最適値に更新されていく．

第 2 章

固定小数点演算における RLS アルゴリズム

RLS アルゴリズムは、前章で述べたとおり、ウエイトや相関行列を更新していく過程に除算を含んだ計算が必要である。なので、一般的には浮動小数点演算によって計算されるが、RLS アルゴリズムを FPGA に実装するためには、全ての計算を固定小数点演算を用いて処理することが望ましい。除算を固定小数点演算で計算すると、スケーリングや丸めによる誤差が生じ、浮動小数点演算によって求めたウエイトや相関行列とは異なった値を持つことになり、精度劣化につながることを考えられる。

本章では、はじめに本研究で提案するシステムの概略について述べ、浮動小数点演算による RLS アルゴリズムと固定小数点演算による RLS アルゴリズムとの演算精度の比較を行う。計算の規模を小さくするため、アンテナ素子数はあらかじめ 4 素子と決定し、固定小数点演算で計算する場合の内部演算パラメータのワード長について検討し、RLS アルゴリズムを固定小数点演算で構成する際に必要となる乗算器や除算器について述べ、固定小数点演算及び浮動小数点演算における RLS アルゴリズムの干渉波抑圧性能について定量的な検討を行う。

2.1 入力情報及び内部演算パラメータのワード長

式 (1.31) ~ (1.38) より、RLS アルゴリズムは演算パラメータとして、入力信号ベクトル x と参照信号 r の 2 つの入力情報と、ウエイトベクトル w 、相関行列の逆行列 R_{xx}^{-1} の 2 つの内部演算パラメータから成り立っていることがわかる。固定小数点演算で RLS アルゴリズムを計算する場合、この入力情報 x, r と内部演算パラメータ w, R_{xx}^{-1} の固定ワード長が RLS アルゴリズムの精度に影響を及ぼすため、以下それぞれの変数のワード長について述べる。

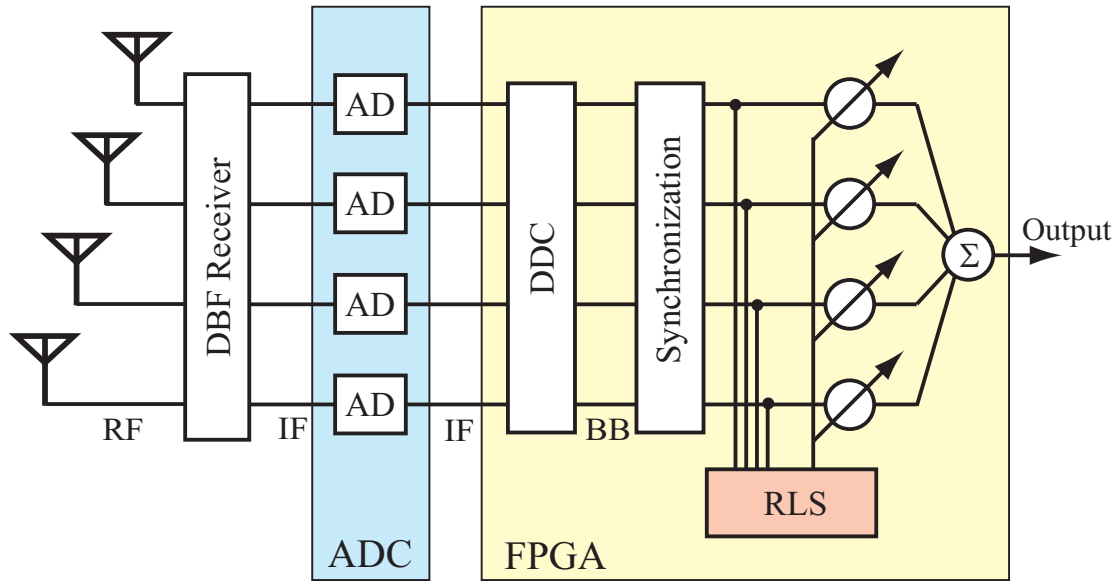


図 2.1: 提案するシステムの概略

図 2.1 に本研究が提案する受信系システムの概略を示す．RF (Radio Frequency) 帯より到来してきた波はアレーアンテナの各素子から受信され，DBF (Digital Beam Forming) 受信機にて IF (Intermediate Frequency) 帯の信号へと変換され，AD (Analog to Digital) 変換の後，信号はFPGA へと送られる．FPGA では DDC (Digital Down Conversion) にてさらに BB (Base Band) 信号に変換され，同期処理器による同期検波，RLS プロセッサによるウエイト更新が行われ，干渉波や遅延波の影響が取り除かれたアレー出力をアウトプットする構成になっている．信号処理の詳細については次章で詳しく述べる．

ここで，RLS アルゴリズムの入力情報である入力信号ベクトル x と参照信号 r は，図 2.1 で想定するシステムで用いる AD 変換器の解像度が 14 ビットなので最大で 14 ビットまでワード長を持つことが可能であるが，DDC の入力レンジが 12 ビットと固定となっているため，ベースバンド I,Q 信号を用いて同期処理や RLS アルゴリズムを行うことを考慮して，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．内部演算パラメータであるウエイトベクトル w と相関行列の逆行列 R_{xx}^{-1} については，FPGA のローカルデータバスの幅が 32 ビットで固定となっており，本研究では 32 ビットのデータバスにて I,Q 成分を 16 ビットのデータを 2 つ合わせることで，I,Q 成分の信号を同時に送ることを可能にしている．RLS プロセッサから出力されたウエイトベクトルも同様に 16 ビットデータを組み合わせることでデータのやり取りを行うため，内部演算パラメータのワード長は 16 ビット以内におさめる必要があるため，内部演算パラメータのワード長は 15 ビットとして検討を行っている．ただし，想定しているシステムでは，DBF 受信機からの入力信号レベルが変動する場合にも AD 変換器に入力する信号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．

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 0° , 干渉波 -40°
変調方式	$\pi/4$ - shift DQPSK
SIR	0 [dB]
SNR	0, 4, 8, 10, 20, 30 [dB]
忘却係数	$q = 0.875 (= \frac{1}{2} + \frac{1}{4} + \frac{1}{8})$

アレーアンテナの素子数を 4 素子と限定した場合において，入力情報である入力信号ベクトル x ，参照信号 r の固定ワード長は，フルスケールで AD 変換を行ったとしてそれぞれ 12 ビットとし，内部演算パラメータであるウエイトベクトル w ，相関行列の逆行列 R_{xx}^{-1} の固定ワード長は 15 ビットとして，固定小数点演算による RLS アルゴリズム及び浮動小数点演算における RLS アルゴリズムの特性をシミュレーションにより比較した．シミュレーションの諸元は表 2.1 に示す通りである．アレー形状は 4 素子等間隔リニアアレーを用いており，そのアレー間隔は $\lambda/2$ として，互いに相関のない 2 波が 0° 方向 (所望波) と -40° 方向 (干渉波) から到来している状況を想定する．2 波とも $\pi/4$ shift DQPSK 変調されており，SIR (Signal to Interference power Ratio: 所望波電力対干渉波電力比) = 0dB，つまり所望波と干渉波は等電力とし，SNR (Signal to Noise power Ratio: 所望波電力対雑音電力比) を 10, 20, 30dB と変化させている．忘却定数は，FPGA に実装する際に 2 進数で表現する必要があることを考慮して $q = 0.875 (= \frac{1}{2} + \frac{1}{4} + \frac{1}{8})$ とした．また，入力情報と雑音は全ての固定小数点演算及び浮動小数点演算において等条件である．

図 2.2，図 2.3 は，RLS アルゴリズムを浮動小数点演算と固定小数点演算を用いて計算した場合において，イタレーション回数に対するアレー出力と参照信号との平均二乗誤差 (MSE: Mean Square Error) によって示される収束特性である．図 2.2 は SNR を 0, 4, 8dB とレベルが比較的低い範囲で変化させた場合の収束特性を示し，図 2.3 は SNR を 10, 20, 30dB とレベルが高い範囲で変化させた場合の収束特性である．

図 2.2 より，固定小数点演算を用いた RLS アルゴリズムと浮動小数点演算を用いた RLS アルゴリズムから得られた収束特性を比較すると，収束速度は若干浮動小数点演算の場合に比べて固定小数点演算の方が遅くなっているが，干渉波抑圧能力は固定小数点演算の場合と浮動小数点演算の場合でほぼ同等な結果が得られたことが分かる．また，図 2.3 から，固定小数点演算を用いた RLS アルゴリズムにおける MSE 特性について，SNR が高くなるにつれて収束レベルが飽和して浮動小数点演算を用いた RLS アルゴリズムによる結果に比べて劣化してしまっていることが確認できるが，SNR が 10dB 程度の場合は浮動小数点演算と固定小数点演算とで同程度の抑圧能力を確認することができる．これは，SNR が高

くなることによって雑音電力が小さくなり， $SNR=30dB$ といった所望波と干渉波の電力が大きい場合には，干渉波に対する雑音の影響が小さくなり，固定小数点演算による RLS アルゴリズムの場合では干渉波を平均二乗誤差の収束値に達するまで抑えることができなくなっているためだと考えられる．また，収束に必要とされるイタレーション回数については，浮動小数点演算の RLS アルゴリズムの場合は数回のイタレーションで収束値に近いところまで収束しているが，固定小数点演算を用いた RLS アルゴリズムの場合は収束回数が多くなっていることが確認できる．

ここで， SNR が $0dB$ ， $20dB$ の時にアレーアンテナで形成されるビームパターンをそれぞれ図 2.4，図 2.5 に示す．図 2.4，図 2.5 共に所望波の方向にビームを向け干渉波の方向にヌルを向けていることが分かる．固定小数点演算による RLS アルゴリズムの方が，やや干渉波方向に向けるヌルのレベルが高いことが確認できるが，いずれの場合も干渉波を抑え所望波を受けようとする様子が分かる．

以上の結果をまとめると，MSE 特性の比較より，浮動小数点演算と固定小数点演算の RLS アルゴリズムの動作は， SNR が高くなるにつれて収束速度や干渉波抑圧能力に差が現れてくるものの， $SNR=10dB$ 以下の場合に関しては浮動小数点演算と固定小数点演算で若干収束速度に誤差は見られるものの大きな変化は見られなかった．また，ビームパターン特性の比較より，どちらの場合も干渉波を抑えて所望波を受けように働いていることが確認できた．

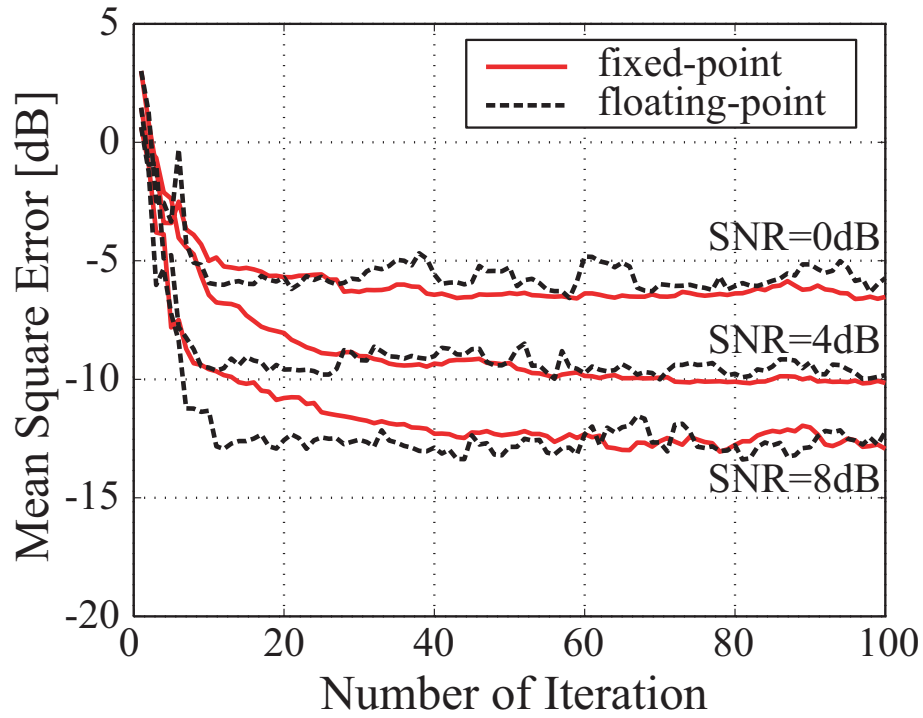


図 2.2: 浮動小数点演算と固定小数点演算の収束特性の比較 (SNR=0, 4, 8dB)

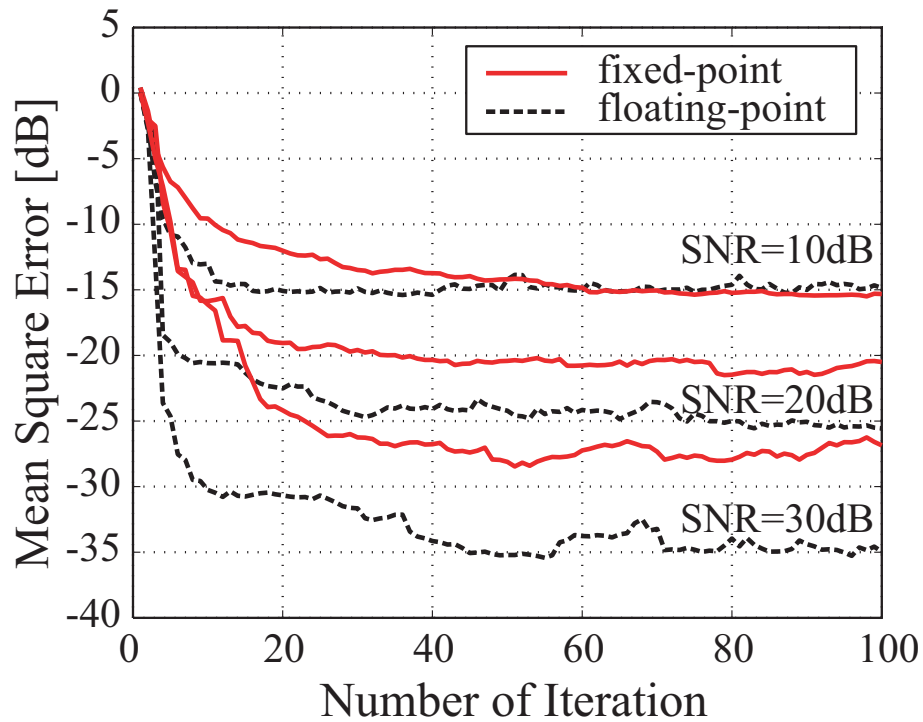


図 2.3: 浮動小数点演算と固定小数点演算の収束特性の比較 (SNR=10, 20, 30dB)

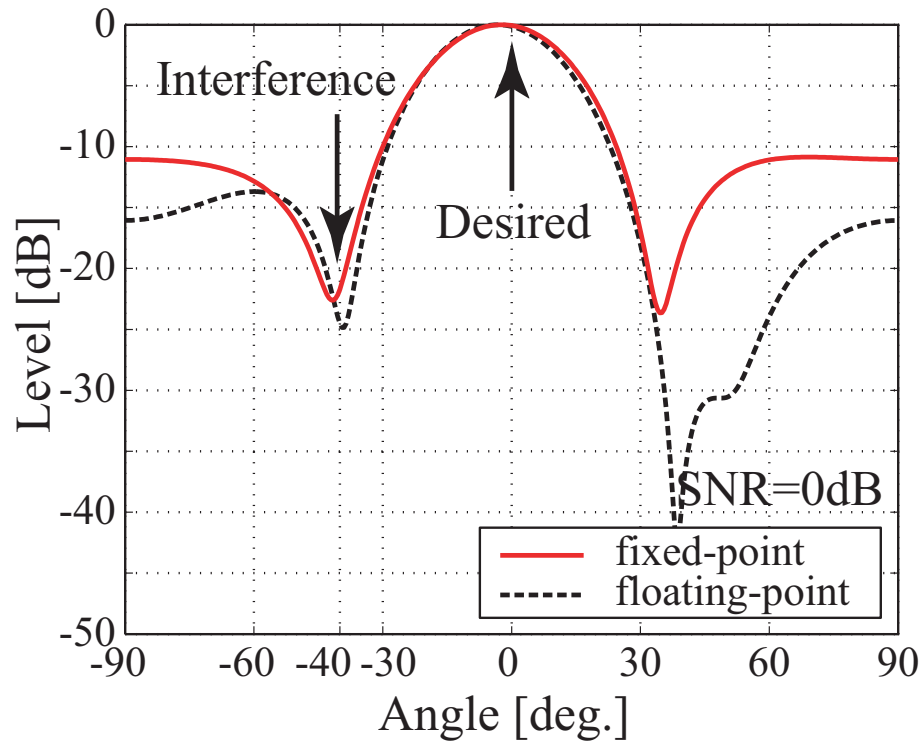


図 2.4: 浮動小数点演算と固定小数点演算のビームパターンの比較 (SNR=0dB)

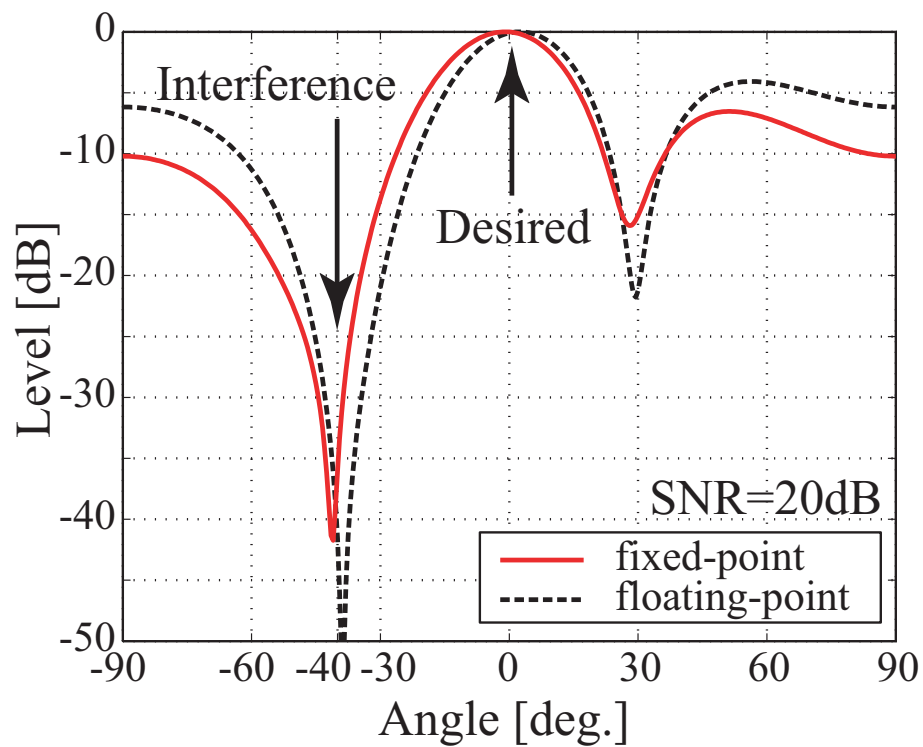


図 2.5: 浮動小数点演算と固定小数点演算のビームパターンの比較 (SNR=20dB)

2.2 乗算器と除算器の構成

2.1 節にて固定小数点演算と浮動小数点演算の精度評価を行ったが、本節では実際に固定小数点演算を用いた RLS アルゴリズムを FPGA に実装した場合に、入力ベクトル x 、参照信号 r 、ウエイトベクトル w 、相関行列の逆行列 R_{xx}^{-1} の固定ワード長を用いた場合に必要とされる乗算器、除算器の規模、及び乗算、除算の計算回数を表 2.2 に示す。表内の乗算回数における () 中の数字は複素数を考慮した場合の計算回数である。 $k(n)$ を求める計算は、式 (1.34) を見ればわかるように除算のみであるが、 $k_n(n)$ 、 $k_d(n)$ がそれぞれ複素数であり、この除算をするために式 (1.34) を次式のように有理化して乗算を行っている。

$$\frac{k_n(n)}{k_d(n)} = \frac{k_n(n) \cdot \overline{k_d(n)}}{|k_d(n)|^2} \quad (2.1)$$

RLS アルゴリズムの乗算回数については、一般にアレーアンテナの素子数 M に対して、 $(3M^2 + 3M)$ オーダーの乗算回数が必要となること知られており [12]、表 2.2 の乗算回数は理論値と一致している。

ところで、加減算および乗算は FPGA を用いて高速処理が可能であるが、除算の場合は様々な問題が生じる。まず除算器に浮動小数点演算を用いた場合は、FPGA に実装する際に固定小数点演算に比べて著しく回路規模が増大し計算速度も遅くなる。除算器に固定小数点演算を用いた場合でもその演算は複雑であり、除算器の計算過程がボトルネックとなり動作周波数が低下することが分かっており、固定小数点演算の場合でも FPGA に実装するには不適である [13]。本研究では、除算器にあたる計算をビットシフトを用いたシフト演算による近似的な除算で処理する。シフト演算による除算は、2 進数で表された値を右に n ビットシフトすることで、シフト後の値をシフト前の値の 2^{-n} にするという手法であり、加減算器及び乗算器以上の動作速度が期待できる。しかし、シフト演算では 2 の累乗の除算でしか処理できないため、除数を 2 の累乗に近似的に置き換えることが必要となる。ここで、除数を d としたときに、 $2^n < d \leq 2^{n+1}$ を成り立たせるような n の値を求め、 d の値を 2^{n+1} に置き換えることで除数を 2 の累乗に近似している。除数の近似的な置き換え方法の一例を表 2.3 に示す。表 2.3 より、近似前の除数の値が大きくなれば近似誤差が大きくなっていくことが分かる。4 素子の場合、表 2.2 より除数の最大ワード長は 55 ビットであり、このときの近似誤差は非常に大きな値になり、ビットシフト後に得られる商の値の誤差も大きくなると考えられる。商の誤差が大きくなると、結果として RLS アルゴリズムの精度劣化につながる可能性があるため、以下シフト演算における近似計算の精度について検討する。

表 2.1 に示される諸元で、入力情報や内部演算パラメータの固定ワード長は 2.1 節にて行った検討より、入力ベクトル x 、参照信号 r をそれぞれ 12 ビット、ウエイトベクトル w 、相関行列の逆行列 R_{xx}^{-1} をそれぞれ 15 ビットとして、シフト演算による除算の場合と除算

表 2.2: 乗算器の規模

	乗算器の規模	乗算回数	ワード長
$y(n)$	15×12	4 (16)	25 bit
$\mathbf{k}_n(n)$	15×12	16 (64)	25 bit
$k_d(n)$	12×25	4 (16)	37 bit
$\mathbf{k}(n)$	36×36	0 (20)	8 bit
	59/55	4 (-)	
$\mathbf{M}_a(n)$	15×12	16 (64)	25 bit
$\mathbf{R}_{xx}^{-1}(n)$	8×25	16 (64)	15 bit
$\mathbf{w}(n)$	8×25	4 (16)	15 bit

表 2.3: シフト演算のための除数の近似例

近似前の除数	近似後の除数
1, 2	2
3, 4	4
5, 6, 7, 8	8
9, 10, 11, 12, 13, 14, 15, 16	16
17, 18, 19, 20, 21, ... ,28 ,29 ,30, 31, 32	32

器を用いた場合の平均二乗誤差による収束特性をシミュレーションによって計算したものを図 2.6, 図 2.7 に示す. 図 2.6 より, SNR が低い場合に関してはシフト演算を用いた除算と除算器を用いた場合で MSE 特性を比較してもほぼ同程度の精度が得られていることが分かる. しかし, 図 2.7 で表されるように SNR が高い場合に関しては, シフト演算を用いた除算は除算器を用いた場合に比べやや収束速度が遅くなっているところが見られるが, 最終的な抑圧能力はいずれの場合も同程度であることが確認できる.

ここで, 浮動小数点演算と固定小数点演算の特性比較の時と同様に, SNR が 0dB, 20dB の時にアレーアンテナで形成されるビームパターンをそれぞれ図 2.8, 図 2.9 に示す. 図 2.8, 図 2.9 共に所望波の方向にビームを向け干渉波の方向にヌルを向けていることが確認でき, 除算器を用いている方がシフト演算を用いた除算を行っている場合よりもわずかに干渉波方向に向けるヌルのレベルが低く抑えられていることが確認できるが, いずれの場合も干渉波方向にヌルを向けて所望波方向にビームを向けている様子が分かる.

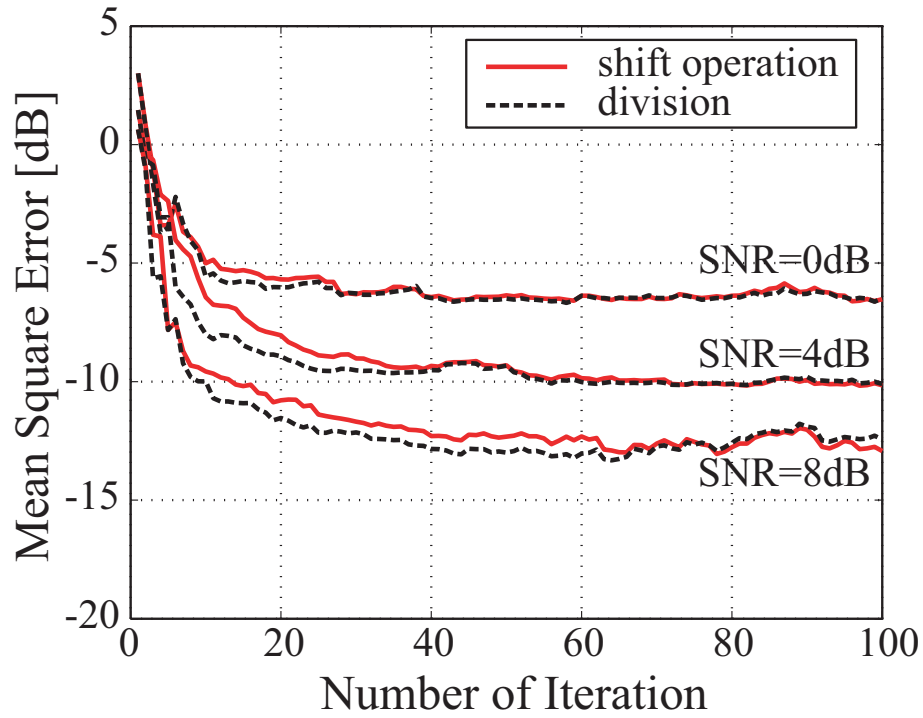


図 2.6: シフト演算による除算と除算器を用いた場合の収束特性の比較 (SNR=0, 4, 8dB)

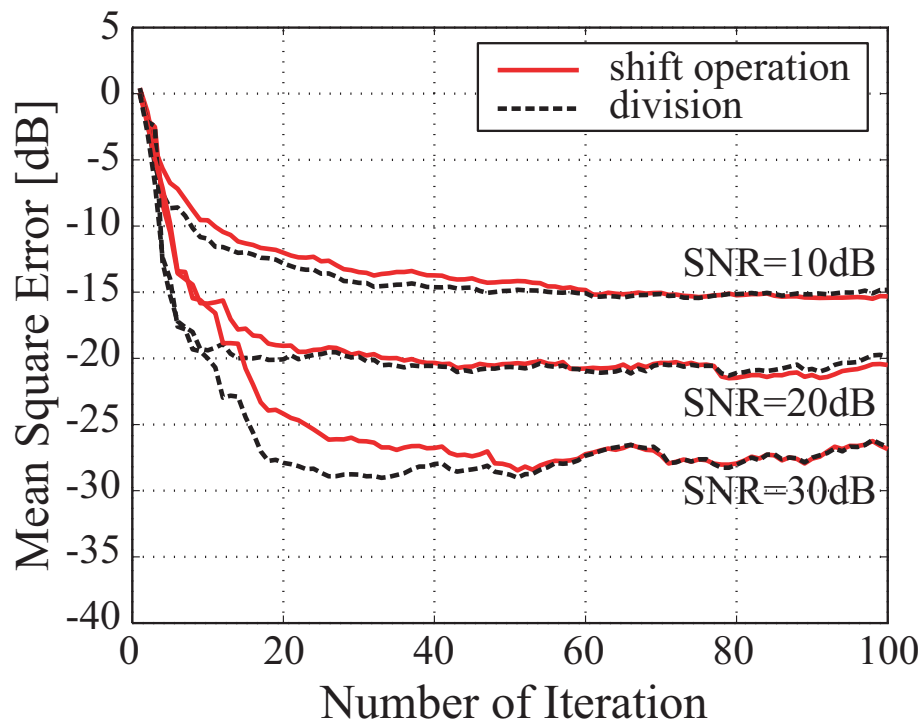


図 2.7: シフト演算による除算と除算器を用いた場合の収束特性の比較 (SNR=10, 20, 30dB)

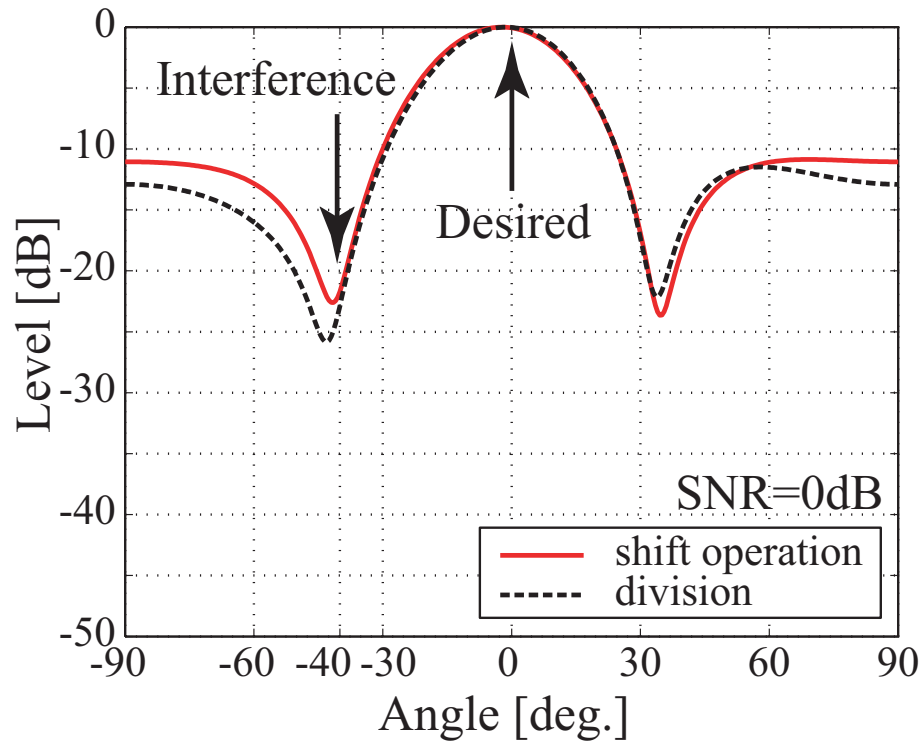


図 2.8: シフト演算による除算と除算器を用いた場合のビームパターンの比較 (SNR=0dB)

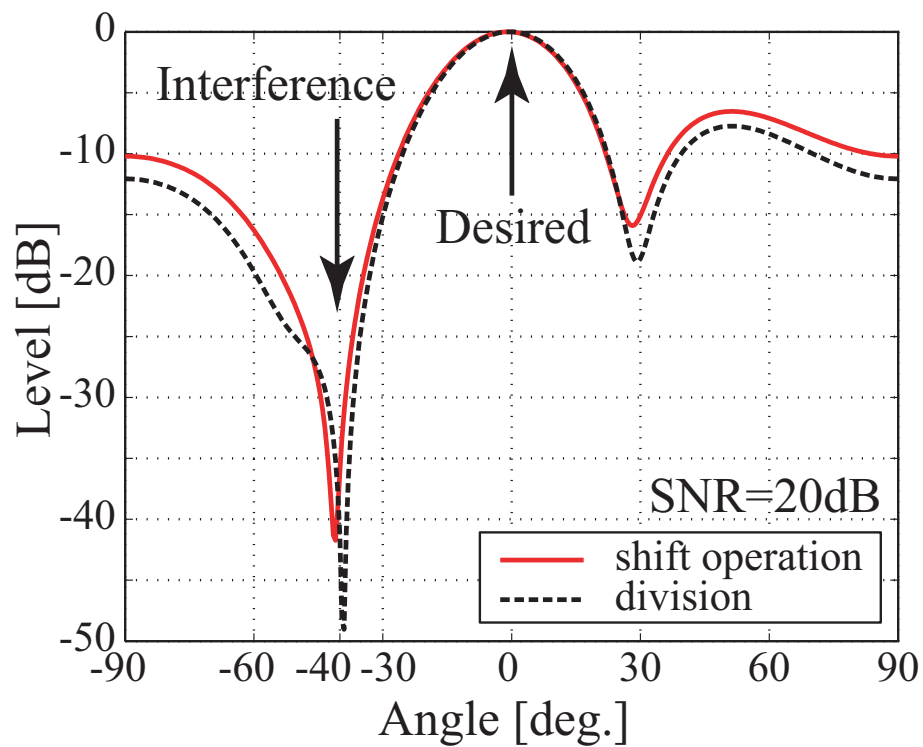


図 2.9: シフト演算による除算と除算器を用いた場合のビームパターンの比較 (SNR=20dB)

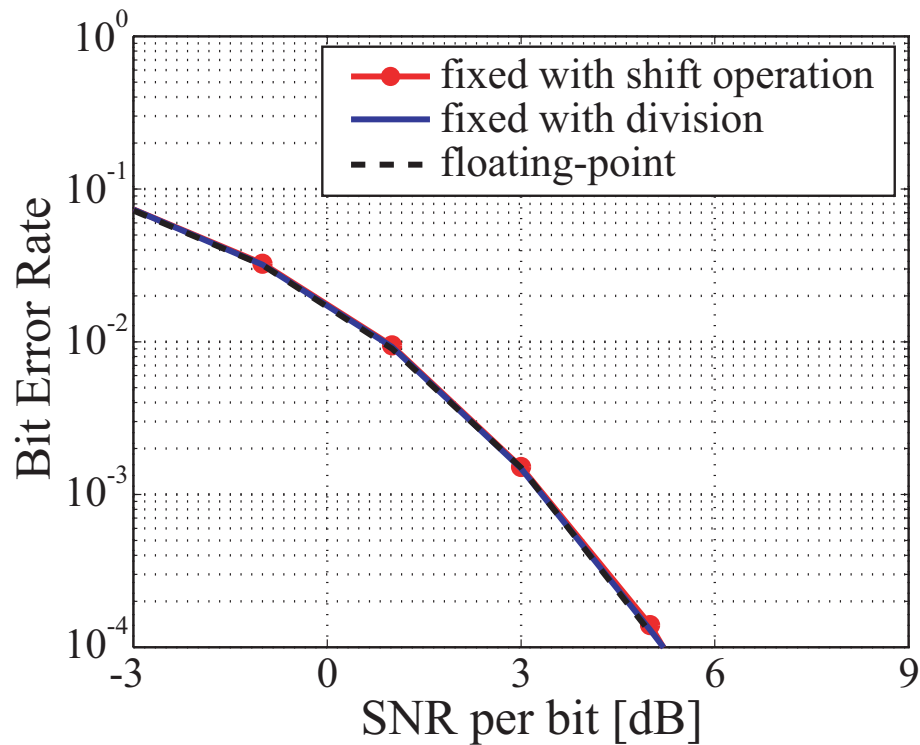


図 2.10: 浮動小数点演算とシフト演算及び除算器を用いた固定小数点演算の BER 特性

図 2.10 は、表 2.1 に示される諸元で、浮動小数点演算による RLS アルゴリズムと固定小数点演算でシフト演算を用いた場合と固定小数点演算で除算器を用いた場合の BER (Bit Error Rate) 特性である。図 2.10 より、浮動小数点演算を用いた RLS アルゴリズムの場合も固定小数点演算を用いた RLS アルゴリズムの場合も BER 特性はほぼ一致していることが確認できる。これより、固定小数点演算においてシフト演算を用いた場合でも浮動小数点演算の場合と同等の精度で RLS アルゴリズムを計算することができることが示される。

以上の考察より、収束特性やビームパターンによる精度評価に加えて BER 特性による精度評価を行った結果、RLS アルゴリズムは浮動小数点演算を用いて計算された方が収束特性の精度は良いが、固定小数点演算を用いて計算した場合においてもその精度誤差は大きくないことが確認できた。さらに固定小数点演算で計算する際に問題となる除算について、除算器を使用しないでビットシフトのみで実現可能なシフト演算を用いることで処理速度の向上及び回路規模の削減を実現可能であり、得られた結果より RLS アルゴリズムを FPGA に実装する際には固定小数点演算で計算してかつ除算をシフト演算で代用できることを示した。

2.3 角度依存性

MMSE アダプティブアレーの最適化アルゴリズムには角度依存性があり，波が 90° 方向などのボアサイドから到来した場合や到来波が近接した場合などでは干渉波抑圧精度が劣化してしまう．図 2.11 に，所望波と干渉波がそれぞれ 0° ， 5° と近接して到来した場合の一例を示す．ビームパターンのヌルを干渉波方向に正確に向けることができず，メインビームも所望波方向に向けることができていない様子が確認できる．このような状況における固定小数点演算と浮動小数点演算による RLS アルゴリズムの精度を定量的に考察するために，本節では固定小数点演算を用いた RLS アルゴリズムの角度依存性の影響を浮動小数点演算を用いた場合と比較して検討する．

シミュレーションの諸元を表 2.4 に示す．まず表 2.4 の case #1 より所望波が 0° 方向より到来し，干渉波の到来方向が変化する場合の角度依存性を BER 特性により評価する．図 2.12 は表 2.4 の case #1 の条件における干渉波の到来方向に対する BER 特性の変化を示している．図 2.13 は図 2.12 を BER 特性の収束レベルについて着目するために拡大した図となっている．いずれも 2° 間隔で干渉波の到来方向を -90° から 90° まで変化させた時の BER をプロットした図であるが，図 2.13 を見ても分かるように，シフト演算を用いた固定小数点演算による RLS アルゴリズムと浮動小数点演算を用いた RLS アルゴリズムは同程度の精度となっていることが分かる．

同様に図 2.14 は表 2.4 の case #2 の条件における所望波がボアサイドから到来した場合の BER 特性を示している．図 2.15 は図 2.14 の BER 特性を拡大して BER の変化を詳細に確認できる図となっている．図 2.15 より $-50^\circ \sim 30^\circ$ 以外から干渉波が到来した場合に BER が劣化していることが分かる．さらに，干渉波が $-10^\circ \sim 20^\circ$ の間で到来した場合において，固定小数点演算による RLS アルゴリズムで計算した場合に BER 特性がわずかに劣化していることが確認できる．

このシミュレーションでは入力情報や内部パラメータの初期値，雑音ともに固定小数点演算と浮動小数点演算で等条件にしているので，固定小数点演算によって生じた演算誤差の影響により干渉波方向に鋭いヌルを形成するようなビームパターンの形状からずれ，干渉波抑圧能力が低下してしまうことで BER 特性が劣化してしまったと考えることができる．図 2.16 は所望波が 60° ，干渉波が 0° より到来したときの浮動小数点演算とシフト演算を用いた場合の固定小数点演算のビームパターンである．図 2.16 より，浮動小数点演算は 0° 方向に正しくヌルを向けているのに対し，シフト演算を用いた固定小数点演算の場合は -5° 付近にヌルが形成されている．しかし，BER の誤差量は浮動小数点演算と固定小数点演算で小さく，また干渉波が $-10^\circ \sim 20^\circ$ 以外から到来する場合は BER 特性はほぼ同程度となっている．

表 2.4: シミュレーション諸元

	case #1	case #2
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)	0°	60°
到来方向 (干渉波)	-90° ~ 90°	
変調方式	$\pi/4$ -shift DQPSK	
SIR	0 [dB]	
SNR	5 [dB]	
忘却定数	$q = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = 0.875$	

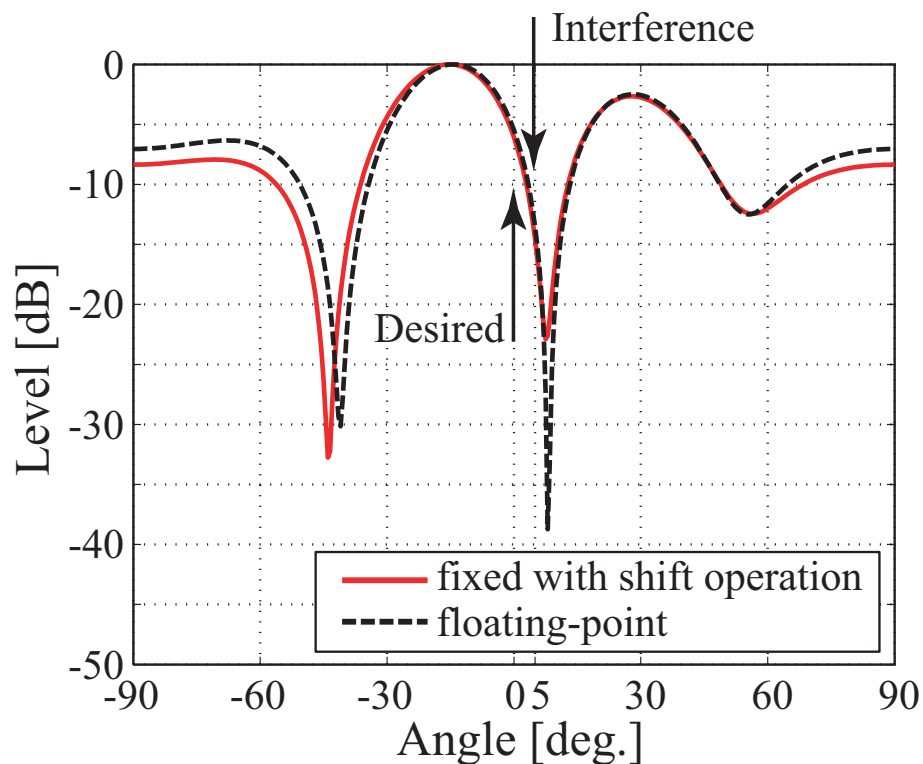


図 2.11: 到来波 2 波が近接した場合のビームパターンの一例

以上より，到来波が近接した場合の角度依存性は固定小数点演算と浮動小数点演算とで同精度であることが分かる．また，所望波が 60° のようなボアサイドから到来した場合には，固定小数点演算による RLS アルゴリズムでは干渉波の到来方向によりヌルが数度ずれてしまい BER がわずかに劣化してしまう部分があるが，極端な精度の劣化は見られない．

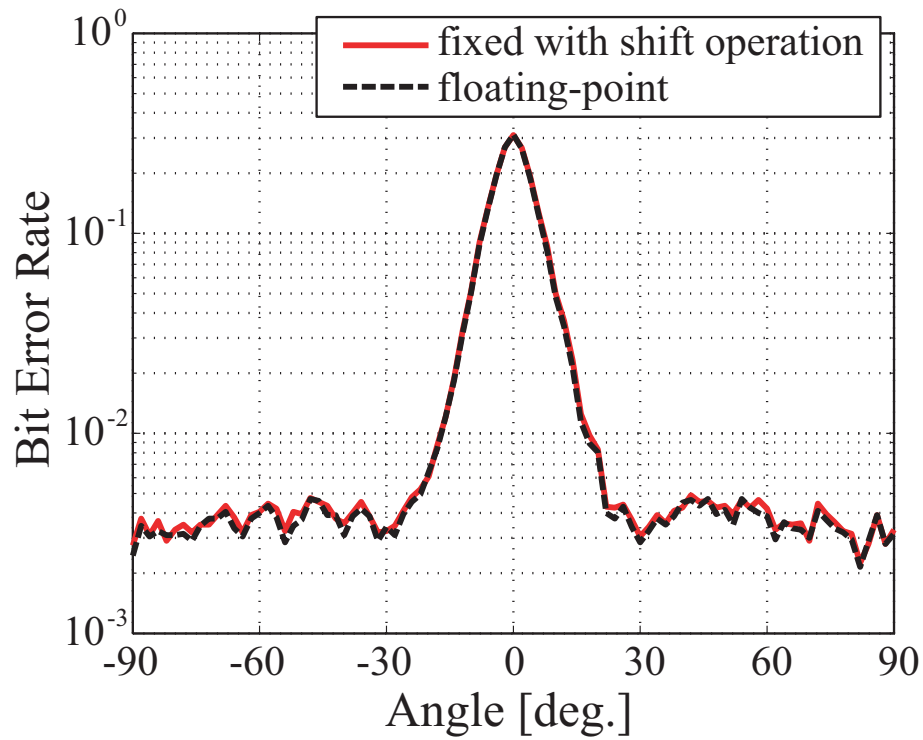


図 2.12: 浮動小数点演算と固定小数点演算を用いた場合の BER 特性 (所望波: 0° より到来)

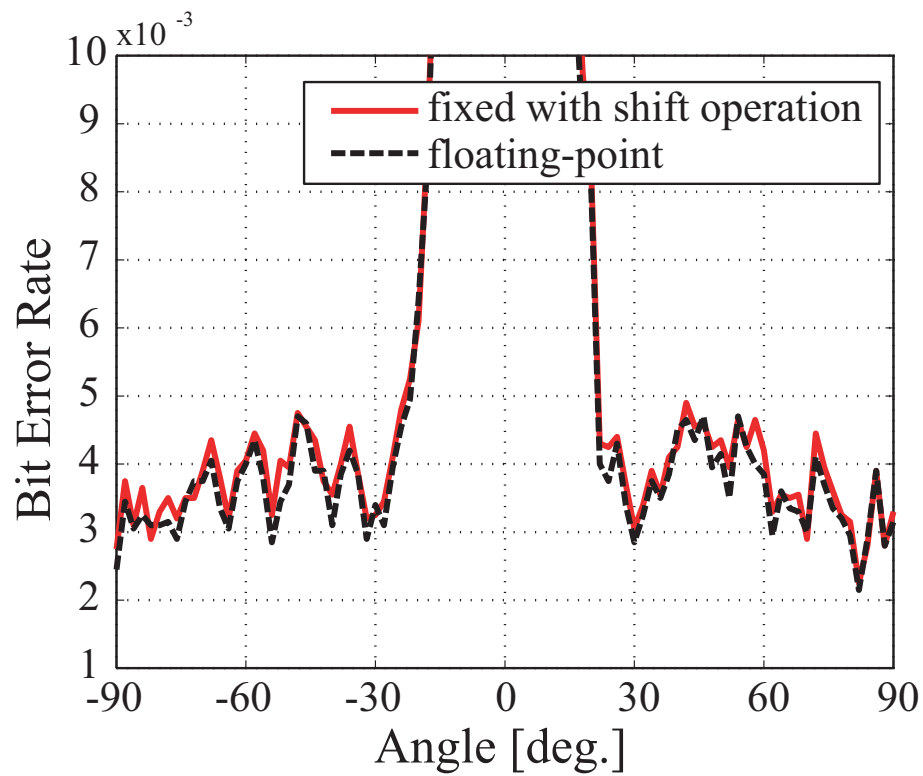


図 2.13: 浮動小数点演算と固定小数点演算を用いた場合の BER 特性 (所望波: 0° より到来)

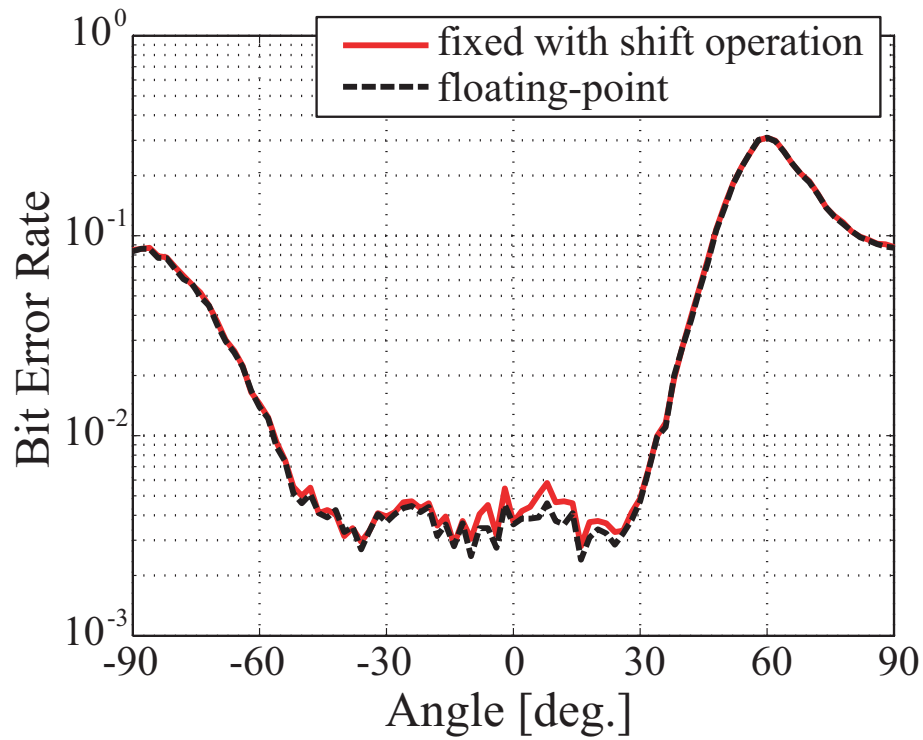


図 2.14: 浮動小数点演算と固定小数点演算を用いた場合の BER 特性 (所望波: 60° より到来)

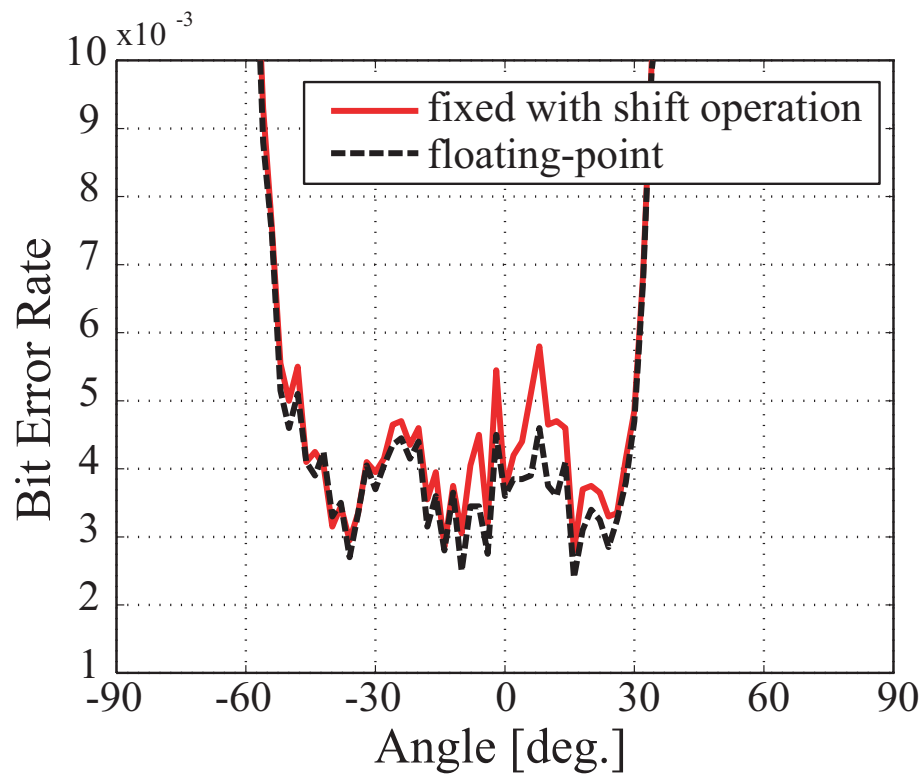


図 2.15: 浮動小数点演算と固定小数点演算を用いた場合の BER 特性 (所望波: 60° より到来)

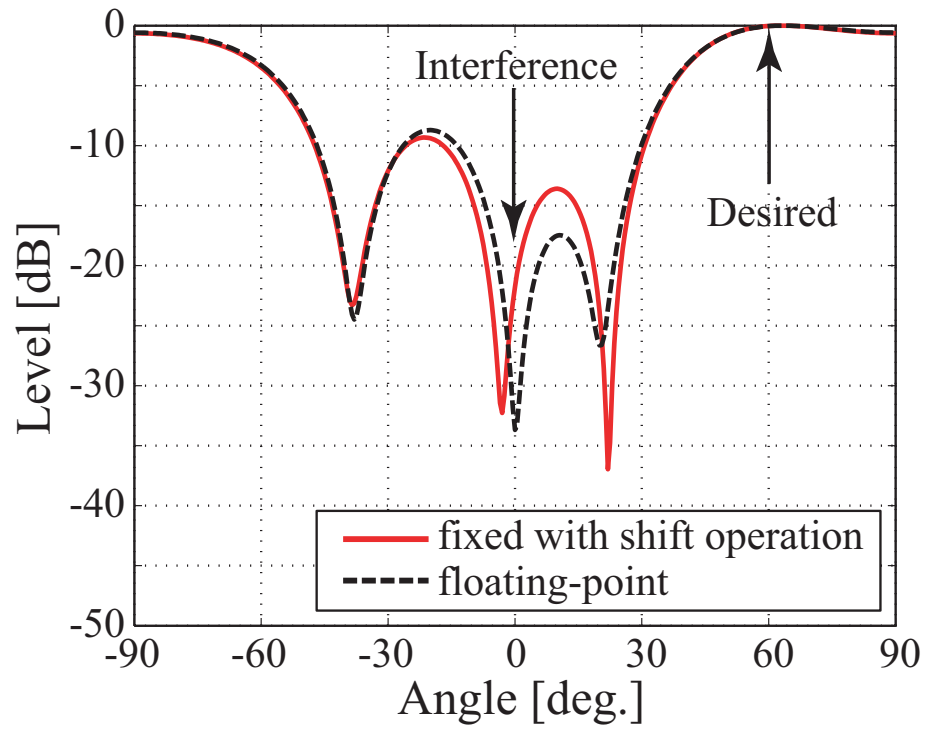


図 2.16: 固定小数点演算と浮動小数点演算を用いた場合のビームパターンの一例

2.4 考察

本章では，固定小数点演算を用いた RLS アルゴリズムを FPGA に実装するために，固定ビット長や固定小数点演算の精度についてシミュレーションによる検討を行った．アンテナ素子数をあらかじめ 4 素子と少なく決定することで固定小数点演算でも浮動小数点演算と同程度の精度が得られることを確認した．

入力信号ベクトル x と参照信号 r の 2 つの入力情報と，ウエイトベクトル w と相関行列の逆行列 R_{xx}^{-1} の 2 つの内部演算パラメータの固定ワード長を設定し，浮動小数点演算による RLS アルゴリズムと固定小数点演算を用いた RLS アルゴリズムの干渉波抑圧能力の検証を行った．固定小数点演算と浮動小数点演算の MSE 特性の比較結果より，RLS アルゴリズムの動作は SNR が高くなるにつれて固定小数点演算の場合の収束速度及び干渉波抑圧能力は劣化が見られるものの，SNR=10dB 以下の場合に関しては浮動小数点演算と固定小数点演算で若干の収束速度に誤差が見られたが干渉波抑圧能力には大きな変化は見られなかった．

また，除算器についての検討も行った．FPGA では除算の動作速度及び回路規模は極めて遅かつ大きな容量を必要とするので，本論文ではシフト演算を用いて除算を行うことを検討し，通常除算器を用いた場合と精度について収束特性，BER 特性からの評価を行った．SNR が低い場合はシフト演算を用いた除算と除算器を用いた場合で MSE 特性を比較してもほぼ同程度の精度が得られていることが確認できたが，SNR が高い場合に関しては，シフト演算を用いた除算は除算器を用いた場合に比べやや収束速度が遅くなっていることが分かった．しかし，最終的な抑圧能力はいずれの場合も同程度であり，BER 特性からも大幅な特性劣化は見られなかった．このことから，シフト演算による除算を用いて高速かつ低規模で演算精度も維持できることを確認した．

固定小数点演算は角度依存性があり，到来波の到来方向によってはアレーアンテナのビームパターンのヌル形成が干渉波方向に向かずにはずかになぜかになってしまう状況があった．ただし，浮動小数点演算においてもその依存性は確認でき，浮動小数点演算と固定小数点演算で精度を比較してみても極端な誤差は見られなかった．

以上のシミュレーションによる検討から，アンテナ素子数が 4 素子の場合において固定小数点演算を用いても高精度に RLS アルゴリズムを計算することができることを検証した．次章にて，固定小数点演算を用いた RLS アルゴリズムの FPGA 実装を行い，実際のシステムに組み込んで動作させた場合を検討する．

第 3 章

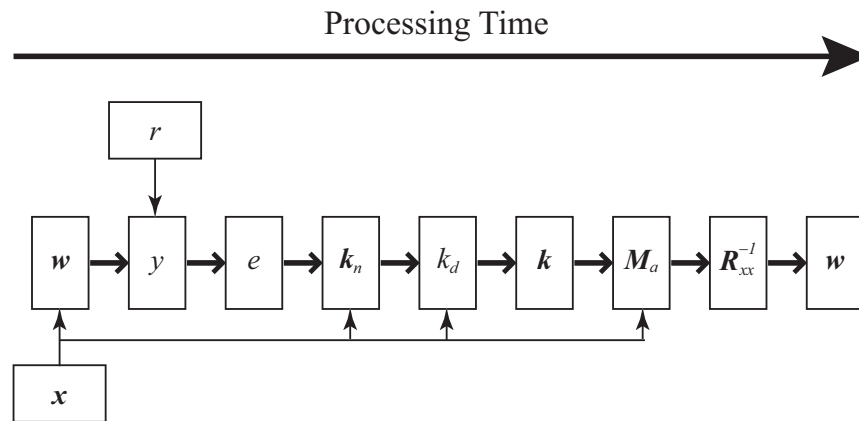
RLS アルゴリズムの FPGA 実装と実験系の検討

本章では，前章での検討から固定小数点演算を用いた RLS アルゴリズムのが浮動小数点演算を用いた RLS アルゴリズムと同等の精度で処理が可能であるので，実際に固定小数点演算の RLS アルゴリズムを FPGA に実装する．はじめに固定小数点演算による RLS アルゴリズムの処理の詳細について述べる．RLS アルゴリズムが FPGA で実装される際の各パラメータの信号処理の流れやワード長の変化について説明し，必要とされる乗算器の個数や複素数の乗算器の構成についても触れ，RLS アルゴリズムの FPGA 実装結果を示す．さらに，実装された RLS アルゴリズムを用いたシステム全体の構成についても述べ，伝搬実験を行うための実験モデルについても述べる．電波暗室にて実際に実験を行い，その結果より提案システムの有効性を評価する．

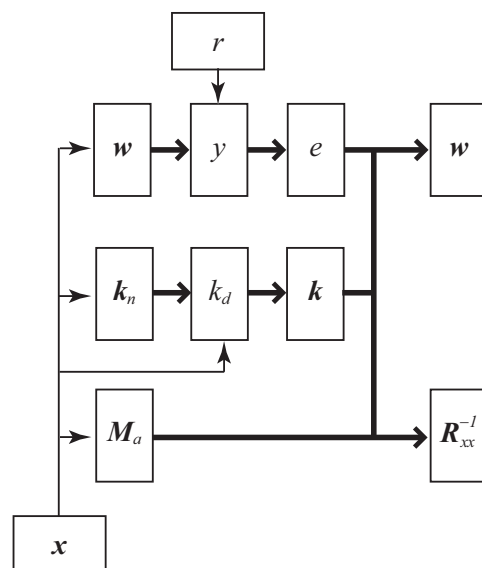
3.1 RLS アルゴリズムの処理の構成

RLS アルゴリズムは式 (1.31) ~ (1.38) に示すように，行列演算を複数回含んでいるので計算量が多くなっており，そのまま逐次演算で高速に処理するためには FPGA の動作周波数を向上させる等の工夫が必要となる．FPGA はフレキシブルな固定小数点演算を得意とするデバイスであり，パイプラインなどの並列処理が可能であるので，固定小数点演算で表現された RLS アルゴリズムを高速に効率よく計算するのに適していると言える．図 3.1 に固定小数点演算を用いた RLS アルゴリズムの逐次処理及び並列処理を行った場合の処理の流れを示す．図中の変数は式 (1.31) ~ (1.38) 及び表 2.2 で用いられている変数に対応している．図 3.1(a) の逐次処理では，1 つのプロセスが終了すると次のプロセスに遷移する構成になっており，乗算器などのコンポーネントを再利用することで回路規模を小さく抑えることができるが多くの処理時間が必要となる．一方，図 3.1(b) の並列処理では複数のプ

ロセスを同時に処理することができるので演算時間は短縮されるが，逐次処理のように乗算器などのコンポーネントを再利用するのではなく複数個並列に配置することで並列処理を実現するので回路規模は増大する．



(a) Serial Processing



(b) Parallel Processing

図 3.1: 固定小数点演算演算を用いた RLS アルゴリズムの処理の流れ

本論文では，FPGA のメリットである並列処理を十分に活用するために，図 3.1(b) の並列処理で表されるプロセス用いて固定小数点演算を用いた RLS アルゴリズムを実装する．図 3.1(b) に示す固定小数点演算を用いた RLS アルゴリズムを FPGA に RLS アルゴリズムを実装する際の信号処理の構成を図 3.2 に示す．図中の赤と青で記された数字はそれぞれ信号のビット数及び乗算器の規模を表しており，変数は式 (1.31) ~ (1.38) 及び表 2.2 で用い

られている変数に対応している．また，破線で囲まれたブロックが今回実装した RLS プロセッサとなっており，このプロセスを各入力サンプルごとに繰り返すことで逐次ウエイト w 及び相関行列の逆行列 R_{xx}^{-1} の更新を行っている．図 3.2 の中央に示される $|\cdot|^2$ では，入力される複素数の絶対値の 2 乗を求める計算を行い，次のプロセスの shift operation では前章で述べたシフト演算を用いてビットシフトによる近似的な除算を行っている．なお，変数のワード長は乗算器や加算器の入力前及び出力後に適宜ビットを削っている．

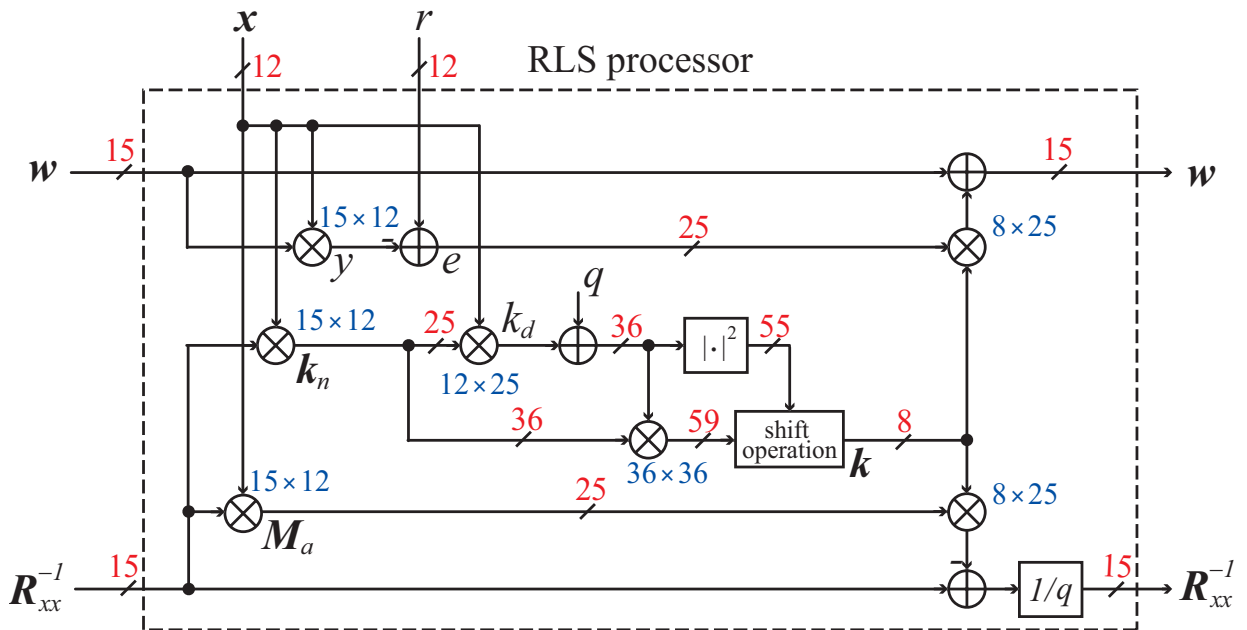


図 3.2: 固定小数点演算演算を用いた RLS アルゴリズムの処理の構成

3.2 FPGA 実装結果

前節ではコンポーネントという表現を使用した，本節ではまず本論文におけるコンポーネント構成について説明する．次に実装する FPGA を搭載したデジタル信号処理ユニットについてその詳細な仕様について述べ，最後に固定小数点演算を用いた RLS アルゴリズムの FPGA 実装結果を示す．

3.2.1 複素数乗算コンポーネント

本研究では，RLS アルゴリズムを FPGA に実装する際に VHDL というハードウェア言語を使用しており，一般的に VHDL では実装する対象を階層構造にする場合が多く，その

一つ一つの階層においてコンポーネントと呼ばれる構成要素を作成してプログラミングする．固定小数点演算 RLS アルゴリズムにおいても，複数のコンポーネントを用いて階層構造にすることで FPGA 実装を実現している．

ここで，本論文で用いている乗算器のコンポーネントについて説明する．固定小数点演算を用いた RLS アルゴリズムでは異なるワード長を持った変数による乗算が頻繁に行われるので，入力する変数のワード長によって乗算器の規模や構成も変わってくる．さらに，RLS アルゴリズムの入力情報及び内部演算パラメータである入力信号ベクトル x ，参照信号 r ，ウエイトベクトル w ，相関行列の逆行列 R_{xx}^{-1} は全て複素数で構成されるので，実際に行われる乗算のほとんどは複素数の乗算となっている．例として，式 (1.31) で表される $y(n) = w^H(n-1)x(n)$ という計算について注目してみると，

$$y = w^H x \quad (3.1)$$

$$= \begin{bmatrix} w_1^* & w_2^* & w_3^* & w_4^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (3.2)$$

$$= w_1^* * x_1 + w_2^* * x_2 + w_3^* * x_3 + w_4^* * x_4 \quad (3.3)$$

ここで，式 (3.3) の乗算は全て複素数同士の乗算である． $w_1^* * x_1$ に着目して実数同士の乗算にするために以下のように式変形する．

$$w_1^* * x_1 = (w_{1,re} + jw_{1,im})^* * (x_{1,re} + jx_{1,im}) \quad (3.4)$$

$$= (w_{1,re} - jw_{1,im}) * (x_{1,re} + jx_{1,im}) \quad (3.5)$$

$$= (w_{1,re} * x_{1,re} + w_{1,im} * x_{1,im}) + j(w_{1,re} * x_{1,im} - w_{1,im} * x_{1,re}) \quad (3.6)$$

このように，複素数同士の乗算を行う場合は実数の乗算を 4 回と加減算を行うことで同等の処理が可能となる．本論文では式 (3.6) で表されるような複素数の乗算を行うコンポーネントを複素数乗算コンポーネントと呼び，複素数乗算コンポーネント内に含む実数の乗算器を内部乗算器と呼ぶことにする．表 3.1 に各変数における乗算器の規模と複素数乗算コンポーネント数，複素数乗算回数を示す．回路規模の増大を抑えるために，乗算器の規模が 15×12 と等しい $y(n)$ ， $k_n(n)$ ， $M_a(n)$ を同じ複素数乗算コンポーネントにて処理している．計算回数の多い 15×12 や比較的規模が小さい 8×25 で構成される複素数乗算は，処理時間を短縮するためにそれぞれ複素数乗算コンポーネントを 4 つ，または 2 つ並列に用いている．表 3.1 の複素数乗算コンポーネント数は並列に用いた複素数乗算コンポーネントの数を示してある．

表 3.1: 複素数乗算コンポーネントと内部乗算器の数

	乗算器の規模	複素数乗算コンポーネント数	複素数乗算回数
$y(n), \mathbf{k}_n(n), \mathbf{M}_a(n)$	15×12	4	36(144)
$k_d(n)$	12×25	1	4(16)
$\mathbf{k}(n)$	36×36	1	0(20)
$\mathbf{R}_{xx}^{-1}(n), \mathbf{w}(n)$	8×25	2	20(80)

3.2.2 デジタル信号処理ユニットの仕様

ここで、RLS アルゴリズムを実装する FPGA を含むデジタル信号処理を行うデバイスを搭載したボードについて説明する。DA 変換, AD 変換, DDC, RLS, 同期処理などの信号処理は全て図 3.3 に示されるデジタル信号処理ユニットにて行われる。仕様の詳細を表 3.2 に示すが、デバイスは大きく ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), FPGA, CPU (Central Processing Unit) から構成されており、他に外部クロックの入出力ポート及び TCP/IP の Ethernet 100 Base-T に対応した LAN ポートを 2 つ持っている。ADC, DAC はそれぞれ 16 チャンネルのポートに対応して 16 個搭載されており、それぞれ 14 ビットの解像度を持っている。ADC, DAC の最大の動作周波数は 80MHz, 165MHz となっているが、本研究での検討ではいずれも 40MHz で動作させている。FPGA は送信系, 受信系でそれぞれ 1 つずつ用意されており、その仕様は表 3.2 に示す通りで同一のものとなっている。ADC に入力した信号から RLS アルゴリズムの処理を行うので、今回 RLS アルゴリズムを実装するのは受信系 FPGA とした。使用する FPGA には DSP ブロックが搭載されており、複雑な加減算及び乗算などの算術演算をより高速に処理することが可能であり、DSP ブロックを使用することで LE (Logic Element) リソース消費を低減できる [13]。また図 3.3 に示すように、デジタル信号処理ユニットにはメモリサブボードとして組み込み式の外部メモリが搭載されているが、本論文においては未使用である。オンボード CPU は NetBSD という UNIX 互換 OS (Operating System) によって制御され、ユーザーが FPGA とデータのやり取りを行う際にはインターフェースとして CPU を介す必要がある。

表 3.2: デジタル信号処理ユニットの仕様

ADC	Resolution	14 bits
	Max sampling rates	80 MHz
	Channels	up to 16
DAC	Resolution	14 bits
	Max data rates	165 MHz
	Channels	up to 16
FPGA	Part	Altera Stratix EP1S40×2
	Logic Elements	41,250 (1 Mega Gates)
	Embedded Memory	3,423,744 bits
CPU	Part	HITACHI SH4
	Operation frequency	200 MHz
OS	NetBSD 1.5	
User I/F	TCP/IP Ethernet 100 Base-T	

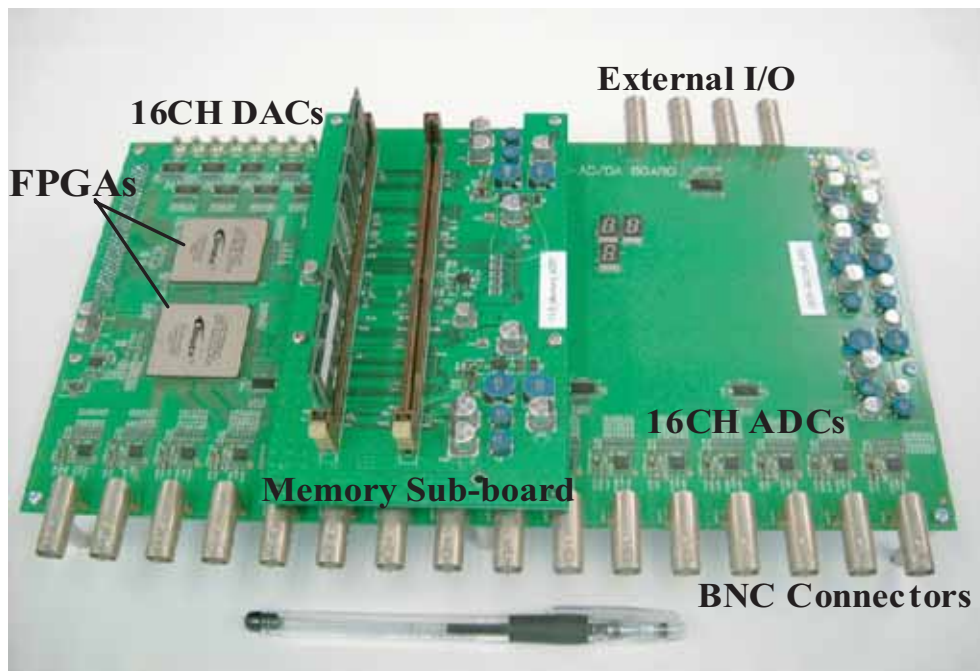


図 3.3: デジタル信号処理ユニットの概観

3.2.3 RLS アルゴリズムの FPGA 実装結果

RLS アルゴリズムを FPGA に実際に実装した結果を表 3.3 に示す．総ゲート数が約 100 万ゲートの FPGA に対して，固定小数点演算を用いた RLS アルゴリズム単体が占める回路容量は約 35% となり，ADC や DDC，その他のコンポーネントやレジスタを含めた全体

表 3.3: RLS アルゴリズムの実装結果

	RLS 単体	全体
回路容量 (総ゲート数に対する占有率)	35 %	62 %
最大動作周波数	80 MHz	80 MHz
クロック数	174 clock	
処理時間 / イタレーション	2.2 μ sec	

の回路容量は約 62% となった。RLS アルゴリズムにおいて、ウエイトベクトルを 1 回更新するのに要するクロック数が 174 クロックであるので、今回実装した FPGA が構成する回路の最大動作周波数 80MHz で動作させた場合、1 イタレーション当たりの処理時間は約 2.2 μ sec となる。本研究では、RLS プロセッサは ADC や DDC の動作クロックに同期させて処理させているので、実際の動作周波数は 40MHz である。この場合の 1 イタレーション当たりの処理時間は約 4.4 μ sec となる。

3.3 システムの構成

本節では、FPGA に実装された RLS アルゴリズムを用いた実験的検討を行うためのシステム全体の構成について説明する。システムの構成の概略図は図 2.1 に示したようになるが、ここではさらに具体的な信号処理の流れとシステムの詳細について述べる。

3.3.1 提案システムの処理の流れ

本研究が提案する受信系システムにおける信号処理のブロックチャートは図 3.4 のように表される。送信アンテナより出力された信号は RF 帯より到来波が 4 素子リニアアレーアンテナの各素子へと入射され、DBF 受信機にて RF 帯から IF 帯へダウンコンバートされ、デジタル信号処理ユニット内の ADC にて AD 変換される。デジタル IF 信号は FPGA へと送られ、各素子に対して DDC されてベースバンド I,Q 信号へと変換されて、その動作クロックに同期して FIFO (First-In First-Out) と呼ばれるメモリに 1 フレームデータが一時的に保存される。FIFO にためられたデータを CPU で読み出して同期処理を行ってフレームデータから最適シンボルを取り出し、RLS アルゴリズムにてウエイトベクトルの計算を行う。RLS アルゴリズムに入力したデータとウエイトベクトルとのアレー出力を再び CPU で読み出すことで、干渉波の影響が除去された所望信号を取り出すことができる。ウエイトベクトルから RLS アルゴリズムによって干渉波に向けるビームパターンを計算することができ、ウエイトを乗算後のアレー出力を復調することで元の信号との BER を計算することができるので、ビームパターン特性及び BER 特性にてシステムを評価する。

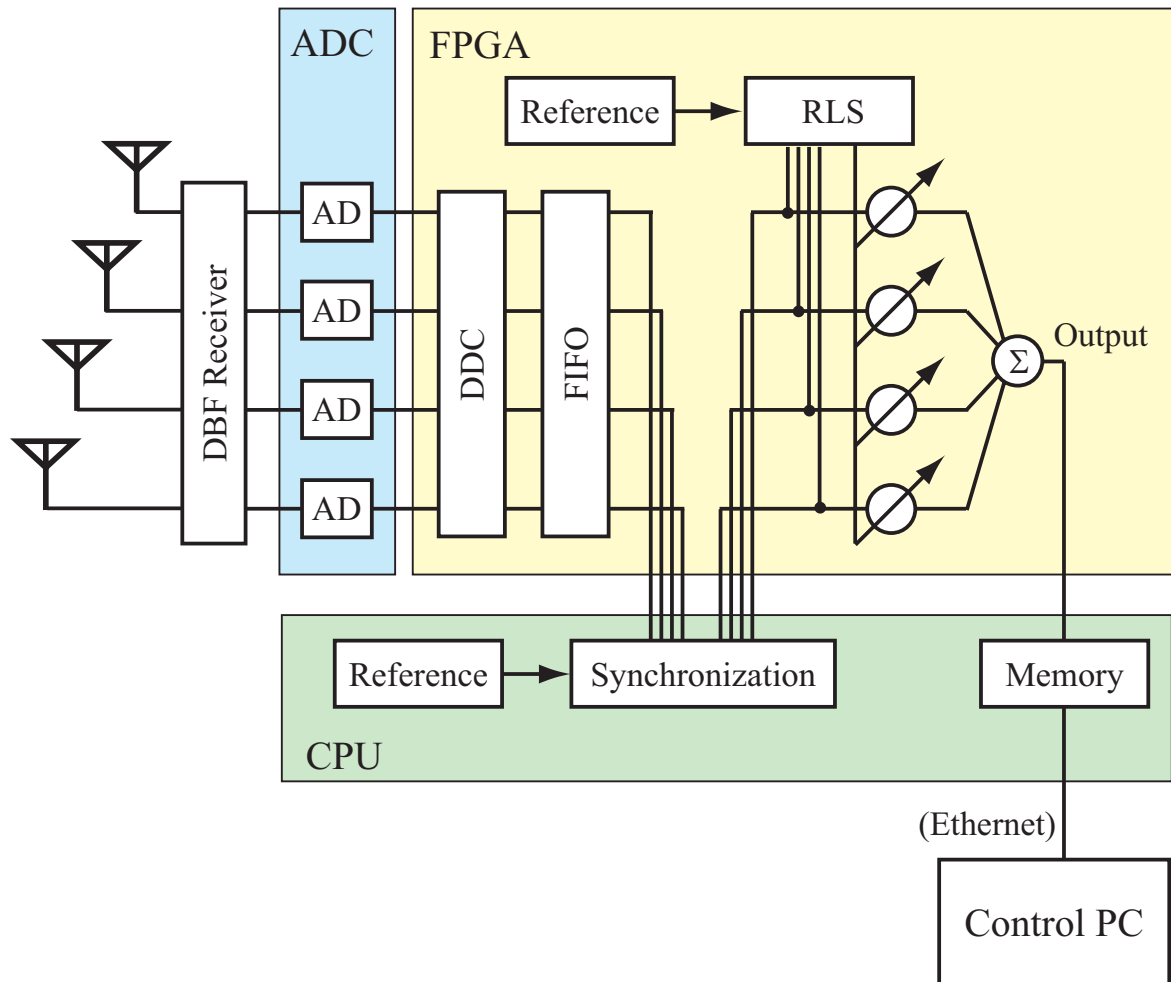


図 3.4: 提案システムの構成

3.3.2 フレーム構成と同期処理の概要

ここで、図 3.4 に Synchronization として表記されている同期処理器について説明する。DDC から出力されるベースバンド信号は、伝搬遅延や送受信機のシステム熱雑音等の影響により、受信したフレームの先頭が必ずしも所望のフレームの先頭になっているとは限らない。所望フレームの先頭からデータを抽出することができない場合、RLS アルゴリズムによって干渉波の影響を取り除こうとしてもヌルが干渉波の到来方向に向けられないどころか、所望波方向にビームを向けることもできなくなる可能性がある。

そこで本研究ではスライディング相関を同期処理器に用いており、あらかじめ送信シンボルの先頭に付与された既知のトレーニングシンボルで構成される参照信号を、DDC 後のベースバンド受信信号に対して 1 サンプルずつスライディングさせて逐次相関係数を計算してその大きさを比較することで、相関係数の大きさが最大になるようなサンプリングポ

イントより所望のフレーム先頭位置を調べている．ここで，フレームの先頭にトレーニングシンボルが 16 シンボル，続いてデータシンボルが 120 シンボルというフレーム構成としており，図 3.5 のように表される．また，スライディング相関の概略図を図 3.6，相関係数とスライディング回数関係をプロットした図を図 3.7 に示す．図 3.6 において， i 回スライディングさせたときの受信信号 x_i と参照信号 r の相関係数 cor_i は次式のように表される．

$$cor_i = \frac{x_i \cdot r^*}{\sqrt{(x_i \cdot x_i^*) \cdot (r \cdot r^*)}} \quad (3.7)$$

スライディング相関によって得られる相関係数は，受信信号 x_i と参照信号 r の内積を受信信号と参照信号のノルムで除算することで得られる．式 (3.7) より相関係数 cor_i を最大にするような i 番目のサンプル点を探す，つまり受信信号中のトレーニングシンボルと参照信号の区間が一致するようなサンプル点を探すことで，参照信号とトレーニングシンボルの相関が最も高い状態を検出することができ，その点からデータを取り出すことでフレームの同期が取れたと判断することができる．

式 (3.7) で表される計算は複雑な除算を含み，FPGA で実現するには回路規模の増大や動作周波数の低下を招き，本研究で使用する FPGA に実装するには不適であるので，図 3.4 に示すように DDC より得られたベースバンド I,Q 信号は一度 CPU にて読み出し，相関係数の最大値を求めて最適なサンプル点よりデータをサンプリングしてからまた FPGA に戻すようなプロセスを踏んでいる．このようにすることで除算を含む複雑な計算が処理可能となるが，CPU の DSP プロセッサを用いた計算なので計算時間は sec オーダーと遅くなってしまっている．

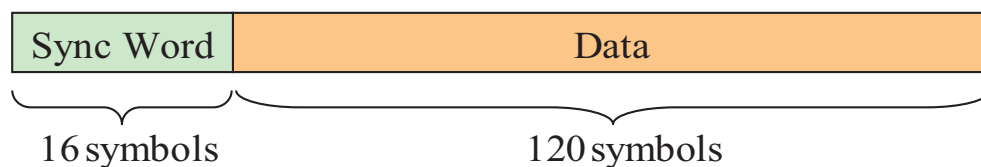


図 3.5: フレームの構成

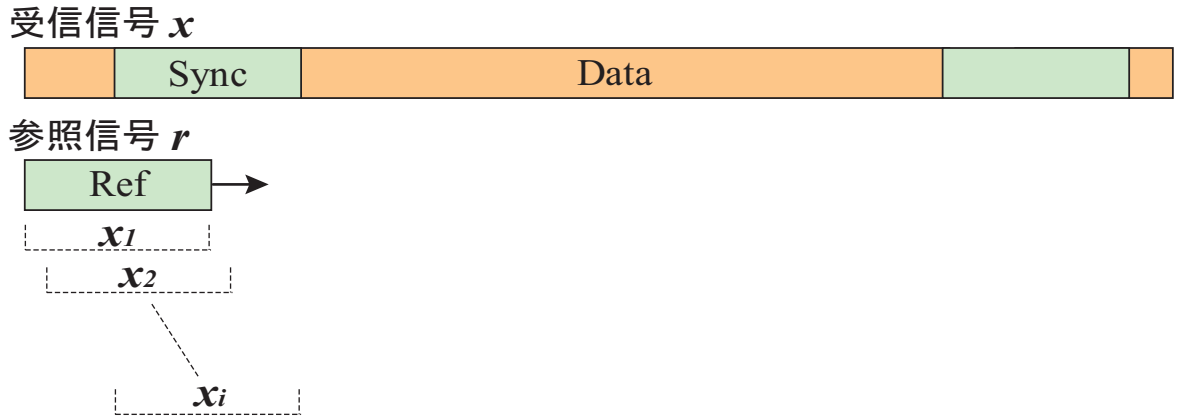


図 3.6: スライディング相関の概略図

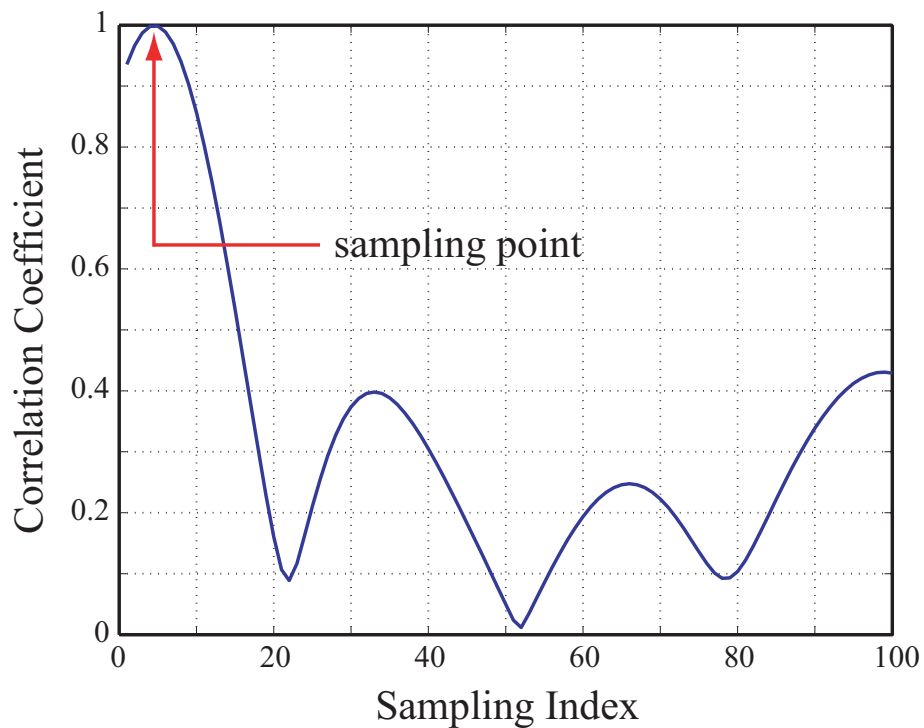


図 3.7: 相関係数とスライディング回数との関係

3.4 実験による評価

前節まで説明してきた提案システムの有効性を評価するために電波暗室にて実際に伝搬実験を行った。図 3.8 は、デジタル信号処理ユニット及び送受信機、S.G.(Signal Generator)、C.S.(Clock synthesizer) を用いた電波暗室における実験のモデルを示しており、表 3.4 に実験の諸元を示す。

表 3.4: 実験の諸元

アンテナ素子	4 素子スリーブアンテナ
アレー配置	等間隔 ($\lambda/2$) リニアアレー
到来波数	無相関 2 波 (INR=3dB)
到来方向	所望波 0° , 干渉波 -40°
変調方式	$\pi/4$ -shift DQPSK
フレーム構成	136 symbols (272 bits)
送信バースト数	100 frames (every level)
DA サンプルレート	40 MHz
AD サンプルレート	32 MHz
シンボルレート	2 Msymbol/s
RF キャリア周波数	5 GHz
送信 IF 周波数	10 MHz
受信 IF 周波数	40 MHz
忘却係数	$q = 0.875$

実験の送信信号にはランダムに生成したバイナリのデータを $\pi/4$ -shift DQPSK 変調したものをを用いており, Matlab にてフィルタリングやアップサンプリングした IF デジタル信号を生成し, 図 3.8 の DSP ユニットの DAC からアナログ信号に変換して出力する. アナログ IF 信号は 10MHz の中心周波数を持ち, 送信機にて S.G. のローカル信号を受けて 5GHz の RF 信号へと変換され, 送信用スリーブアンテナから電波暗室内に出力される. 受信アンテナは素子間隔 $\lambda/2$ の 4 素子リニアアレーで送信アンテナと同じスリーブアンテナを用いており, 受信された信号は受信機にて 40MHz の IF 信号へとダウンコンバートされる. 受信アナログ IF 信号は, ADC に入力され DDC されてベースバンド I,Q 信号へと変換される. 得られたベースバンド信号は, スライディング相関器によって元のベースバンド信号との相関値が最も高くなるようなタイミングで間引かれ, その I,Q コンスタレーションパターンよりデマッピングされると元のバイナリ信号が再生される.

以上が信号処理の流れであるが, 所望波と干渉波の条件によっては 2 波を同時に送信した場合に同期を確立することができない可能性が考えられる. 例えば仮に所望波と同電力程度でかつ同一系列のトレーニング信号を持つ干渉波が到来してきた場合には, スライディング相関を用いた方法では必ずしも同期を確立できないと予想される.

そこで, 今回の実験では送信源から所望波・干渉波それぞれ別々のタイミングで 1 波のみ送信することを想定して同期処理を行っている. 実装した RLS アルゴリズムを評価するためには, 所望波と干渉波を 2 波到来させてその干渉波抑圧能力について考察する必要があるため, 2 波到来を想定するためにローテータを用いてアレーアンテナを回転させ, 仮想的に異なる角度からの到来波として信号を受信することで 2 波が同時に到来した環境を仮定している. 所望波と干渉波の到来方向は, 所望波を 0° , 干渉波を -40° からの 2 波と

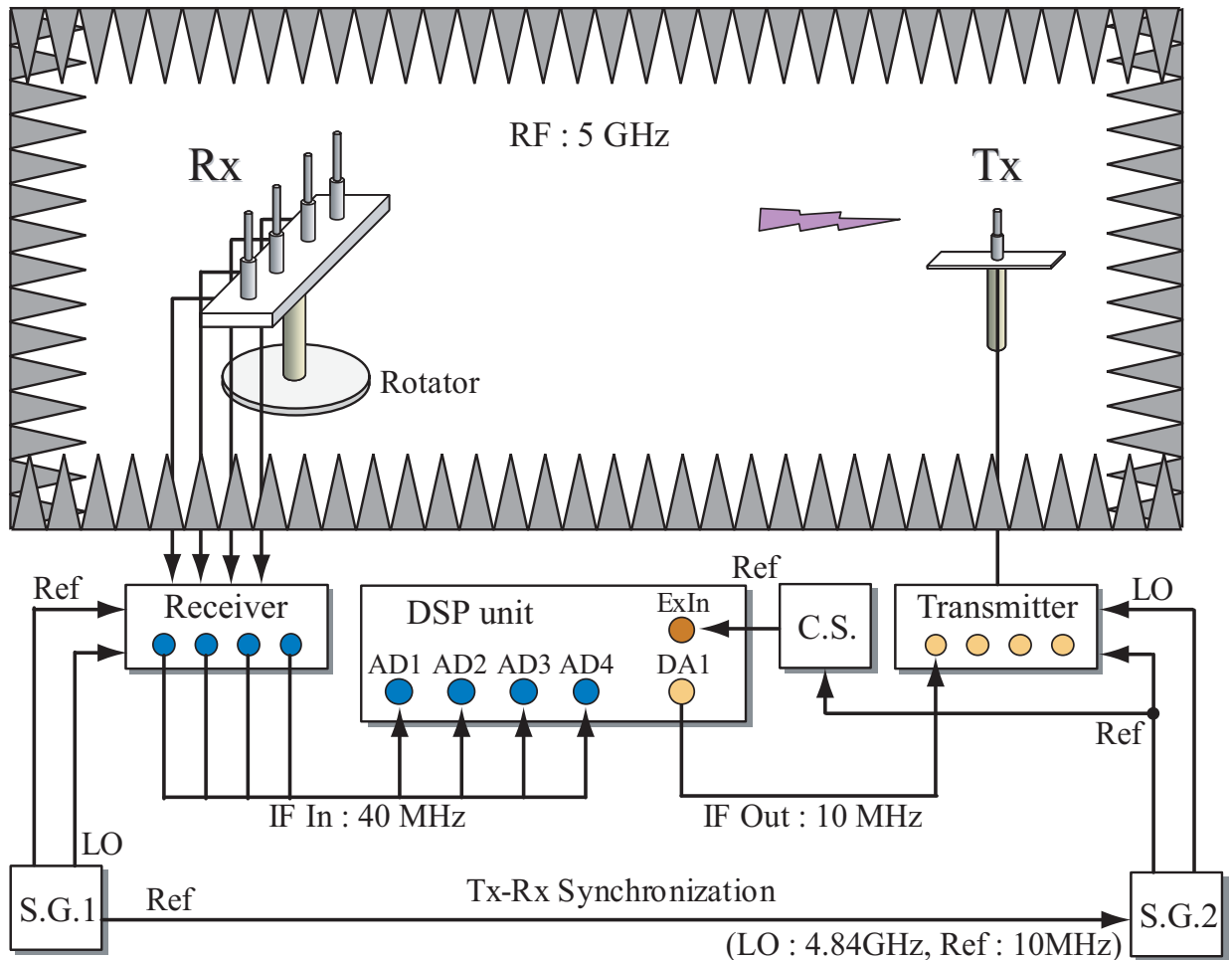


図 3.8: 電波暗室における実験モデル

し、それぞれの場合においてアンテナ素子から得られた受信信号のデータを合成して、その合成信号を 2 波到来した時の受信信号と仮定して検討を行う。

RF 周波数は 5GHz とし、IF 周波数は、受信側では 40MHz の IF 信号を 32MHz でアンダーサンプリングし、送信側では 10MHz の IF 信号を 40MHz で 4 倍のオーバーサンプリングを行っている。ここで、実験で用いる受信機が 40MHz の IF 信号を出力する仕様になっており、40MHz の IF 信号を 32MHz でアンダーサンプリングすることで対応している [14]。また、送受信機及びデジタル信号処理ユニットは S.G. と C.S. からリファレンス信号を入れているので完全に同期が取れている環境となっている。

以上に述べた手段で電波暗室にて実験して得られた結果を図 3.9 と図 3.10 に示す。図 3.9 は、4 素子リニアアレーアンテナのビームパターンを表しており、所望波方向にメインビームを向け干渉波方向にヌルを向けていることが確認でき、RLS アルゴリズムが干渉波の影響を取り除くように動作していることが分かる。また、図 3.10 は、干渉波が存在しない場

合の AWGN (Additive White Gaussian Noise) チャネルにおける $\pi/4$ -shift DQPSK の同期検波を行った際のアンテナ 1 素子に対する理論曲線と, RLS アルゴリズムを FPGA に実装する場合を想定した実装と同じ条件での固定小数点演算及び浮動小数点演算によるシミュレーションによる結果と, 実際に伝搬実験を行った際の単位ビット当たりの SNR に対する BER 特性の比較となっている. AWGN チャネルにおいて 1 素子同期検波の QPSK 及び $\pi/4$ -shift DQPSK に対する BER の理論値は次式のように与えられる [15].

$$P_{N,QPSK}(\gamma) = \frac{1}{2} \operatorname{erfc}(\sqrt{\gamma}) \quad (3.8)$$

$$P_{N,\pi/4DQPSK}(\gamma) = \operatorname{erfc}(\sqrt{\gamma}) \quad (3.9)$$

ここで γ は 1bit 当たりのエネルギーに対する雑音電力密度比を表し, erfc は相補誤差関数で次式で表される.

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt \quad (3.10)$$

アンテナ素子数が 4 素子で 2 波到来する場合の BER の理論値は, 干渉波方向に正しくヌルを向け干渉波を除去することで 4 素子 1 波の場合と同じとなり, 最適合成により 1 素子 1 波の場合に比べて 6dB の利得改善があることから求めることができる.

図 3.10 の横軸の SNR per bit は, 今回の実験において AWGN を発生させる際に送信出力を一定に保つことにより電力レベル一定の受信信号に対して雑音レベルを変化させることで定義している. 実験では, 送信側から何も信号を出力していない状態において発生するシステムの熱雑音を十分長いサンプル測定し, その平均値を取って雑音電力レベルの固定値としている. 求めた雑音電力の固定値に対して, 送信の出力レベルは SNR が約 3dB 程度の間隔で変化するように調整し, 受信信号電力から平均を取って決めた雑音電力レベルを引いた値を所望信号の電力レベルとしている. BER 特性を見てみると, 電波暗室内実験より得られた BER 特性は理想的な曲線を描いているシミュレーション結果に良く似た実験結果となっていることが分かるので, BER 特性からも RLS アルゴリズムの有効性を確認することができる.

以上の結果より, 固定小数点演算にて実装された RLS アルゴリズムを用いて, 伝搬実験においても干渉波の影響を十分に取り除き所望信号を取り出すことで良好な特性が得られていることが確認でき, 提案システムの有効性を実証することができる.

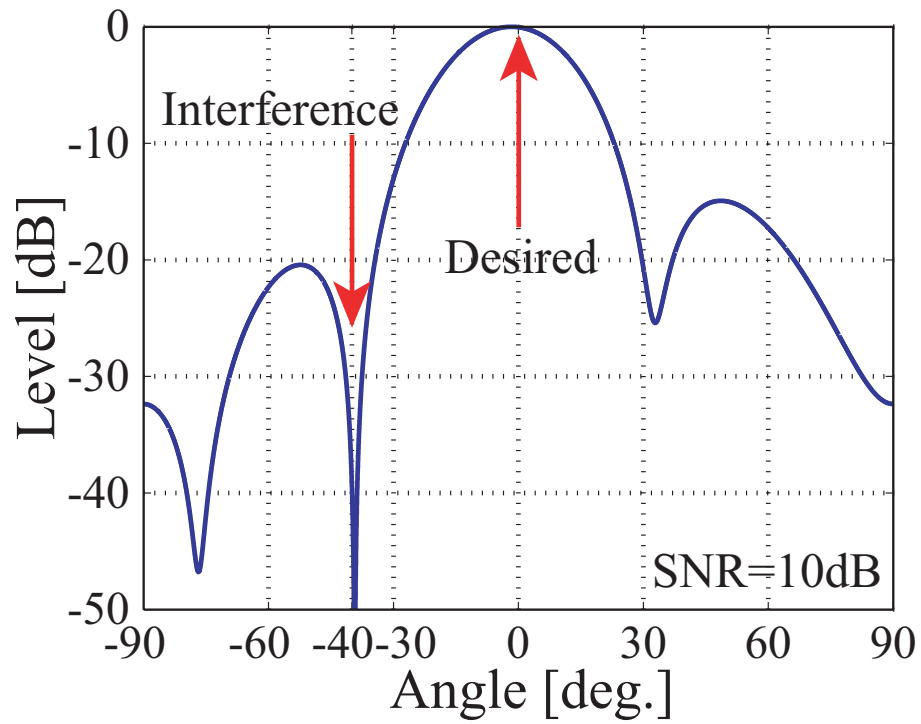


図 3.9: 暗室実験から得られたビームパターン

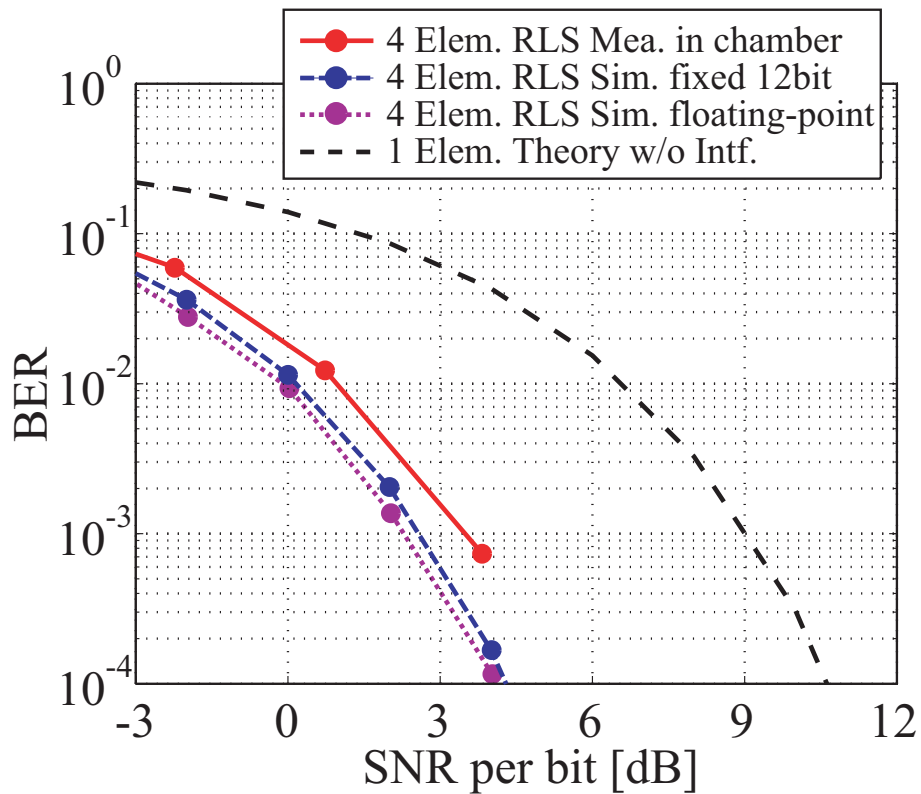


図 3.10: 暗室実験とシミュレーションから得られた BER 特性

第 4 章

同期処理器のFPGA実装とリアルタイム実験

本章では、従来まで数 sec オーダーの計算時間を要したオンボード CPU 実装された同期処理器を FPGA で実装することで一連の信号処理を FPGA 内で処理することが可能となり、同期処理を含めた固定小数点演算 RLS アルゴリズムの計算が μsec オーダーで実現でき、リアルタイム処理が可能であることを述べる。従来までのスライディング相関は CPU にて乗算や除算を行っていたために処理が複雑化して多大な計算時間を要していたが、FPGA で実装される提案するスライディング相関器は並列処理を行うことで高速処理を実現できる。

はじめに同期処理において重要となる所望波と干渉波の相関関係を M 系列符号を用いて改善し、送信信号のフレーム構成をダブルフレーム構成にすることでリアルタイム処理実現に対応することについて述べ、シミュレーションにおいて提案する同期処理方法が従来までのものよりも高精度に計算できることを示す。さらに FPGA に実装した場合の回路規模及び計算時間について述べ、実際に IF 帯を直接接続する実験を行うことでリアルタイム同期処理器が正常に動作していることを実証する。

4.1 M 系列符号を用いたフレーム構成

前章で述べたように、所望波と干渉波が混在し、かつ干渉波の信号電力が所望波と同程度であるような伝搬環境の場合は、必ずしもスライディング相関を用いて同期を確立することができない。そこで、所望波と干渉波の相関関係を改善させるために、フレームの先頭に配置するトレーニングシンボルに M 系列 (Maximal-length sequences) 符号を用いて所望波の自己相関を高くすることで検討を行う。

4.1.1 M 系列符号

CDMA (Code Division Multiple Access) システムでは複数ユーザを分離するために拡散符号が重要な役割を果たしている．一般に拡散には PN (Pseudo Noise) 系列と呼ばれる拡散符号が用いられており，PN 系列はシフトレジスタとフィードバックを用いた回路によって人工的にある規則に基づいて生成される．本研究では，最も良く知られた PN 系列のうち，優れた相関特性を持つ符号系列が少ないという欠点からシングルユーザのスペクトル拡散に用いられている M 系列について検討する．

M 系列とは，ある長さのシフトレジスタとフィードバックによって生成される符号系列のうち，その周期が最長になる系列である． n をシフトレジスタの段数とすると，M 系列の符号長 L は次式で与えられる．

$$L = 2^n - 1 \quad (4.1)$$

図 4.1 に M 系列発生回路を示す．M 系列符号発生器は何段かのシフトレジスタとその複数の段の状態の論理結合をシフトレジスタの入力へフィードバックする論理回路とで構成される．ここで図中に示される論理和は排他的論理和であり， f_j はシフトレジスタにかける係数 (0 または 1)， a_i はシフトレジスタの各段の内容である．

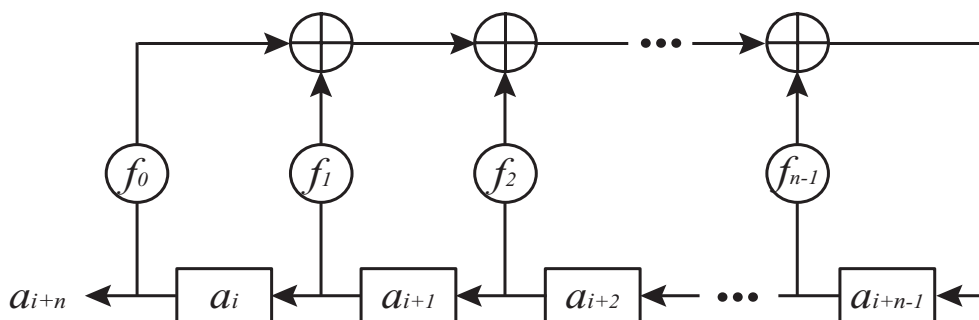


図 4.1: M 系列発生回路

図 4.1 の回路で発生される系列は次式によって与えられる．

$$a_{i+n} = \sum_{j=0}^{n-1} f_j a_{i+j} \quad (4.2)$$

式 (4.2) は， $f_n = 1$ として次のようにも表せる．

$$\sum_{j=0}^n f_j a_{i+j} = 0 \quad (4.3)$$

これらの式は系列を発生させる線形漸化式と呼ばれ，ここで $a_{i+j} = x^j a_i$ となるような遅延演算子を導入すると式 (4.3) は，

$$\left(\sum_{j=0}^n f_j x^j \right) a_i = 0 \quad (4.4)$$

となる .

$$f(x) = \sum_{j=0}^n f_j x^j \quad (f_0 \neq 0, f_n = 1) \quad (4.5)$$

ここで、式 (4.5) の多項式は特性多項式と呼ばれ、発生する系列の性質を決める重要な多項式であり、生成される系列が M 系列である。つまり系列長が最長であるためには、特性多項式は原始多項式と呼ばれる多項式でなくてはならない。原始多項式はシフトレジスタのタップが特定の位置の場合にのみ作られる。今回は 5 次 (シフトレジスタの段数 $n=5$) の M 系列を使用し、所望波と干渉波のトレーニングシンボルに用いる原始多項式をそれぞれ $f_d(x) = x^5 + x^3 + 1$, $f_u(x) = x^5 + x^2 + 1$ として M 系列を生成させている。所望波の場合を例にとると、式 (4.5) より $(f_0, f_1, f_2, f_3, f_4, f_5) = (1, 0, 0, 1, 0, 1)$ となるから、初期条件として $(a_1, a_2, a_3, a_4, a_5) = (0, 0, 0, 0, 1)$ とすると M 系列発生回路は図 4.2 に示すようになる。 $n=5$ としていることから、式 (4.2) より符号長が 31 ビットの M 系列符号が得られる。

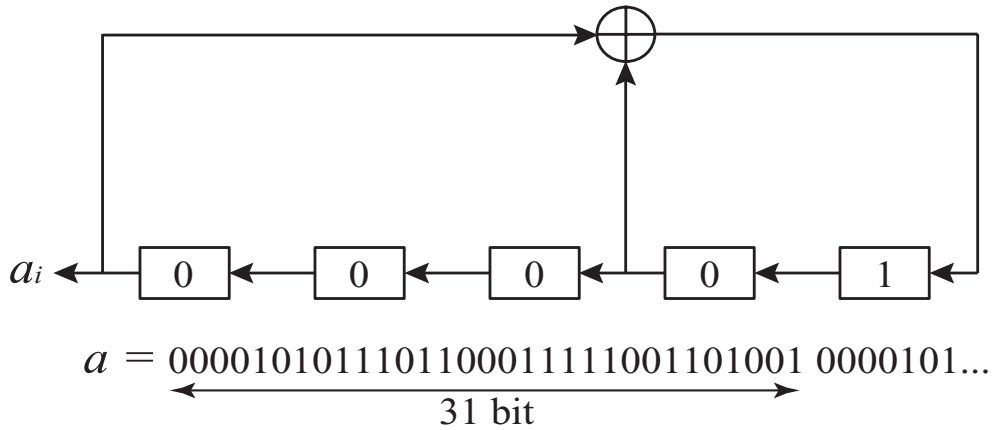


図 4.2: 5 次 M 系列発生回路の一例

4.1.2 ダブルフレーム構成

前章での電波暗室における伝搬実験では、送信フレームをバーストで 1 フレームだけ送信してそれを受信側で処理する流れになっていたが、この場合 DAC から出力するタイミングと ADC で受信するタイミングをデジタル信号処理ユニット内のトリガーによって合わせているので、受信フレームの先頭付近にトレーニングシンボル部分が現れる。このような場合、数十回程度スライディングさせて相関係数の大きさを調べれば、その範囲内に最大値が得られることが分かっている。しかし、フレームを連続して出力し続けて受信側においてもリアルタイムで同期処理を確立できる環境を実現するためには、必ずしも受信フレームの先頭にトレーニングシンボル部分が現れるとは限らない状況を想定し、トレーニングシンボル部分がフレームの途中に現れる場合についても対応できなければならない。

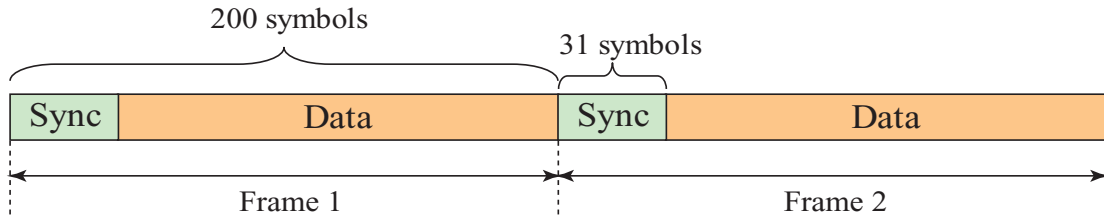


図 4.3: ダブルフレームの構成

そこで送信フレームを図 4.3 に示すような構成に変更することを提案する。図 3.5 に示す従来まで用いていたフレーム構成とは、1 フレームに存在するトレーニングシンボル部分とデータ部分の領域の数が異なっている。

トレーニングシンボル及びデータ部分を 2 つずつ設けることにより、図 4.4 に示すように、受信信号のフレーム内のどの位置にトレーニングシンボルがきても 2 つのトレーニングシンボル部分のいずれかで必ず相関係数を最大にするサンプリング点を得られる。図 4.4 の Case #1 は受信信号のちょうど先頭にトレーニングシンボルが現れた場合であり、この時は一番最初のスライディングによって最大相関係数が得られる。Case #2 は、先頭付近に現れたトレーニングシンボルが何らかの原因で欠落または破損などによって失われてしまった場合もしくはフレームの最後にシフトしてしまった場合について示しているが、この場合は 2 番目に現れるトレーニングシンボル部分より同期を取ることができる。Case #3 では、トレーニングシンボル部分がフレーム先頭より遅れて受信された場合であり、この場合は最初に現れるトレーニングシンボルによってサンプリング点を得られる。これら 3 つの Case から、受信フレームに対してスライディングを行う範囲は、フレームの先頭から全フレーム長の半分までとなることが分かる。

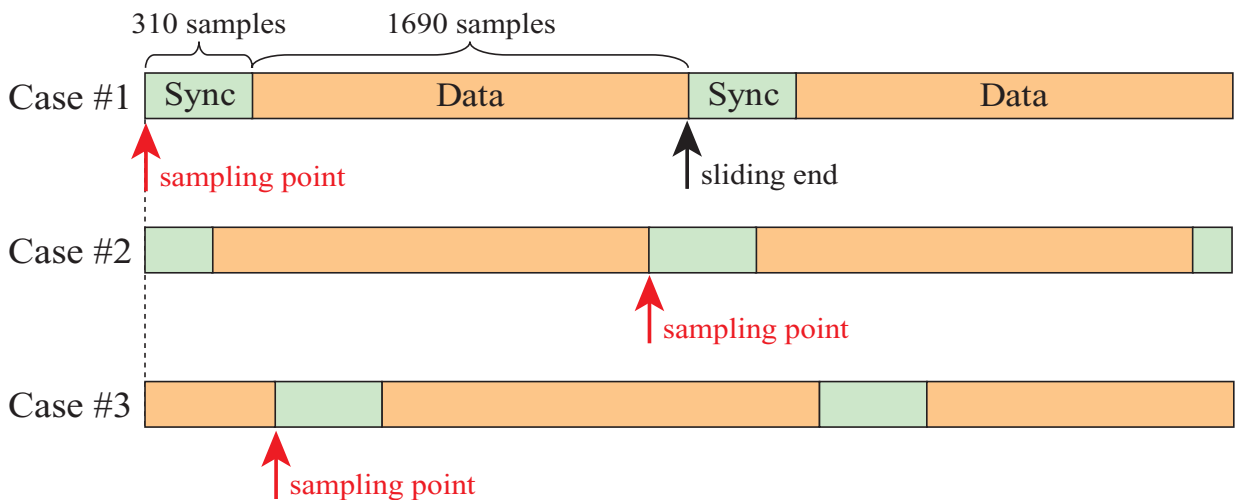


図 4.4: 受信信号に対するダブルフレーム構成の効果

4.2 シミュレーションによる同期処理器の検討

本節では，新たなスライディング相関器の構成を提案して，従来法より高速化，回路規模の縮小，同期処理の精度向上を実現できることを示す．シミュレーションでは，提案するスライディング相関器と従来までの相関器について，所望波と干渉波が2波混在する環境における同期処理能力について述べる．また，トレーニングシンボルにM系列符号を用いた場合の同期処理器の精度の確認やアップサンプリングレートを変化させた場合の相関係数のピークの変化を検討する．さらに提案したスライディング相関器をFPGAに実装した場合の処理速度及び回路規模を従来までの構成のスライディング相関器のものと比較する．最後に実験系のシステムにて動作確認実験を行い，提案システムの有効性を評価する．

4.2.1 提案同期処理器の構成

3.3.2節で同期処理の概略について述べたが，スライディング相関では式(3.7)で表される相関係数 cor_i の大きさを最大にするようなサンプル点を調べることでフレームの同期を確立し元の信号を復元することができる．参照信号に対して受信信号を1サンプルずつスライディングさせて相関係数を求めていくことからスライディング相関と呼ばれており，概略図は図3.6に示した通りである．

式(3.7)で表される計算を行う従来までの同期処理器は，図4.5に示すように，仮にトレーニングシンボル数を31と決めた場合，連続した受信信号 $x_1 \sim x_{31}$ に対して参照信号 $r_1 \sim r_{31}$ を乗算した総和を求め， x を1サンプルずらして同様に逐次相関係数を求めていく．この計算には12ビット×12ビットの乗算や除算を含んでいるのでその演算量は非常に大きくなっている．この方法の場合，本研究における実験系ではベースバンド信号はアップサンプリングしてからIF信号へと変換しているため，31シンボルに決めたトレーニングシンボルもアップサンプリングのレートによってサンプル数が変化する．例えば10倍のアップサンプリングさせた場合，トレーニングシンボルは10倍で零補間されてフィルターに通されて310サンプルのデータに拡張される．したがって，31サンプルずつ受信信号と参照信号の相関係数を求めても，M系列等の特殊な系列のトレーニングシンボルを用いている場合その全てのシンボルを有効に利用することができない．参照信号のサンプル数を310サンプルにして受信信号と相関を求めればトレーニングシンボルの影響を反映することは可能だが，サンプル数が多くなればなるほど計算負荷が増大してしまう．

そこで本研究で提案する同期処理器の構成は図4.6に示すようになっており，アップサンプリングレートに合わせて受信信号を間引いて参照信号と乗算するため，M系列等のトレーニングシンボルの情報を全て考慮することが可能である．ここでもトレーニングシンボル数を31，アップサンプリングレートを10倍と仮に決めた場合，受信信号 $x_1, x_{11}, x_{21}, \dots, x_{301}$ から31サンプルを取り出してそれぞれ参照信号と乗算することで相関係数の大きさを計算

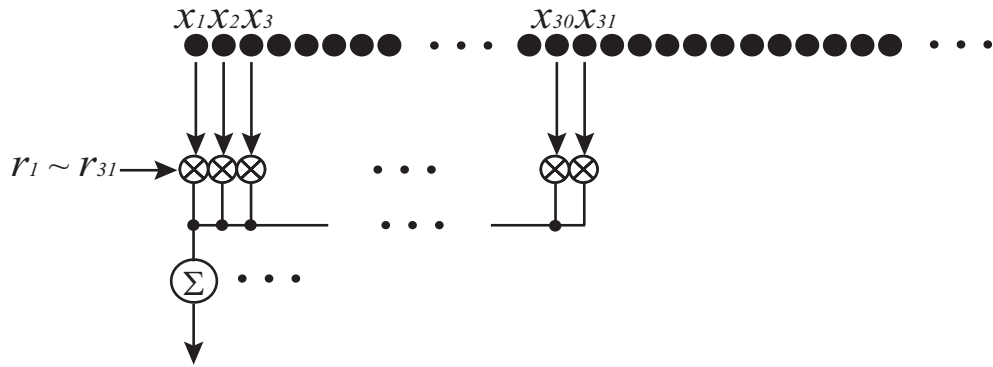


図 4.5: 従来までのスライディング相関器の構成

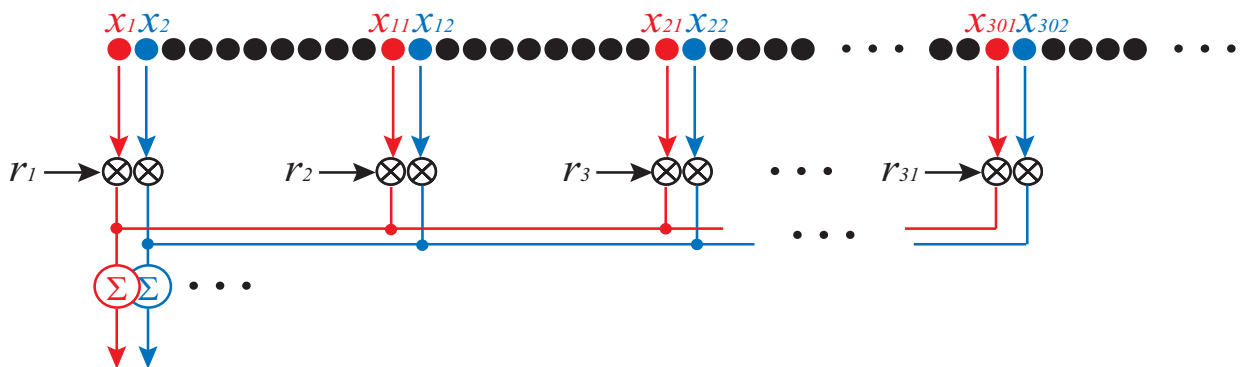


図 4.6: 提案するスライディング相関器の構成

する．ここで，同期処理器を FPGA に実装するに当たって計算速度と回路規模の増大を抑えるために，12ビットで表現される参照信号 r に着目する．参照信号 r は変調方式が QPSK の場合は必ず 1, -1 の値で表現することが可能であり，受信信号と参照信号の乗算の際に符号反転を行うかどうかだけで演算が可能で，式 (3.7) におけるノルムの除算も不必要となる．参照信号の値を 1, -1 で構成する場合そのワード長はわずか 1 ビット済むので，計算時間及び回路規模の大幅な削減が見込める．しかしビット削減することで演算精度が劣化してしまう可能性が考えられるので，従来法と提案法のスライディング相関器を用いた場合の精度評価を行う必要がある．

表 4.1 に示す諸元にてシミュレーション諸元を行い，従来法及び提案法によるスライディング相関の精度を評価した．トレーニングシンボルには 5 次の M 系列符号を使用しており，フレームもダブルフレーム構成にしており，最適なサンプリング点はフレームの先頭から 500 サンプル遅らせて 501 番目としている．結果を図 4.7～図 4.10 に示す．いずれの図も従来法と提案法のスライディング相関器の特性を比較しており，横軸がスライディング相関のサンプルのインデックスを表し，縦軸は相関係数の大きさとなっている．

図 4.7, 図 4.8 は SNR=0dB のときの従来法と提案法の比較結果で, 図 4.7 において相関係数のピーク値付近を拡大した図が図 4.8 となっている. 図 4.7 では最適サンプル点付近で相関係数がピーク値となり他のサンプル点の場合は相関係数は小さい値を取っていることが確認できる. また, 提案法は従来法に比べてピーク値以外の抑圧レベルが小さく抑えられていることが分かる. また図 4.8 においては, いずれの方法においても最適サンプル点にピークがきていることが確認できる.

また, SNR=20dB という雑音が所望信号に与える影響が小さくなった場合のシミュレーション結果を図 4.9, 図 4.10 に示す. 図 4.9 から, SNR=0dB の場合より提案法が従来法より相関係数の抑圧レベルをより小さく抑えられていることが分かる. また図 4.10 は図 4.9 の拡大図であるが, SNR が高い場合は提案法のプロットのばらつきが抑えられていることが確認でき, 従来法と提案法でいずれも最適サンプル点にピークがきていることが分かる.

以上より, 従来法の結果に比べて提案法の結果は相関係数のピーク値以外のサンプル点では抑圧レベルを低く抑えることができていることが良好となっており, 提案したスライディング相関器の構成でも精度劣化なく用いることができることが分かった. これは従来法が受信信号を連続して取り出して参照信号との相関を計算するのに対し, 提案法はアップサンプリングレートに対応して間引く間隔で受信信号を取り出すので, より自己相関を高めることができたからであると考えられる.

表 4.1: 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 0° , 干渉波 -40°
変調方式	QPSK
SIR	0 [dB]
SNR	0, 20 [dB]
トレーニングシンボル	31 シンボル (5 次 M 系列符号)
フレーム長	400 シンボル (ダブルフレーム構成)
アップサンプルレート	10 倍
参照信号のサンプル数	従来法: 100, 提案法: 31
忘却係数	$q = 0.875$

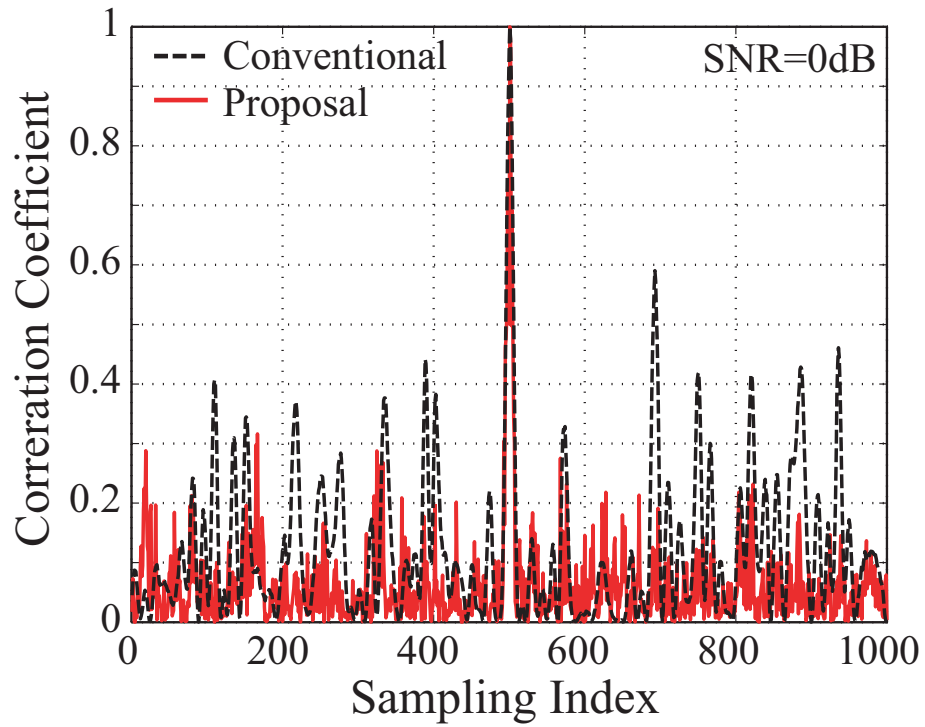


図 4.7: 従来法と提案法における相関係数の比較 (SNR=0dB)

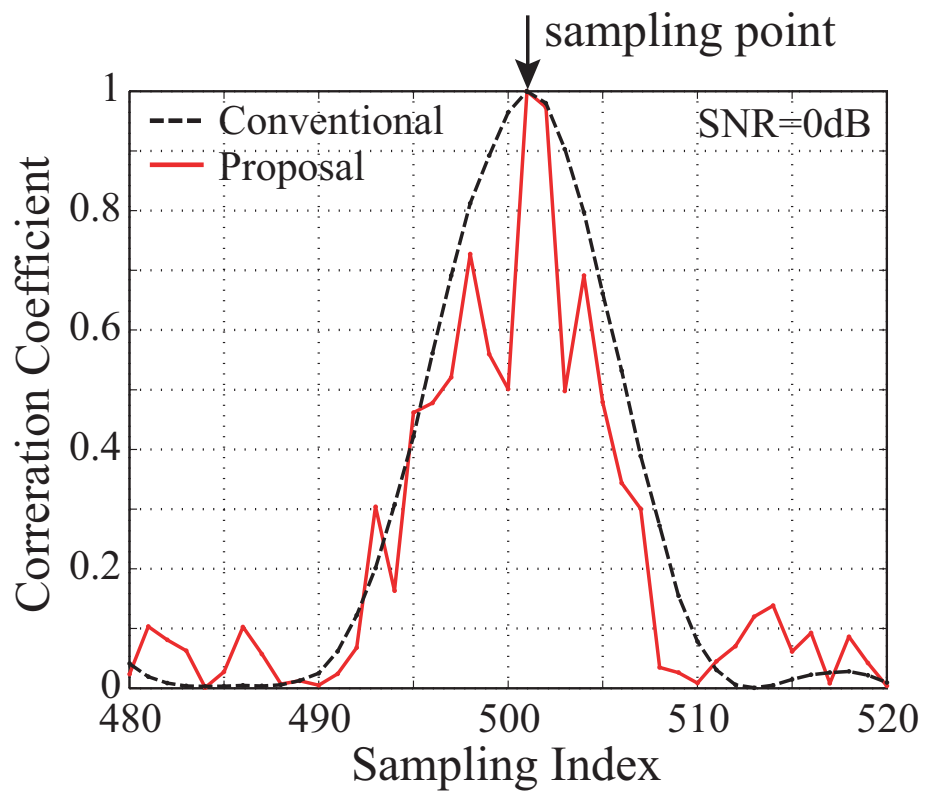


図 4.8: 従来法と提案法における相関係数の比較の拡大図 (SNR=0dB)

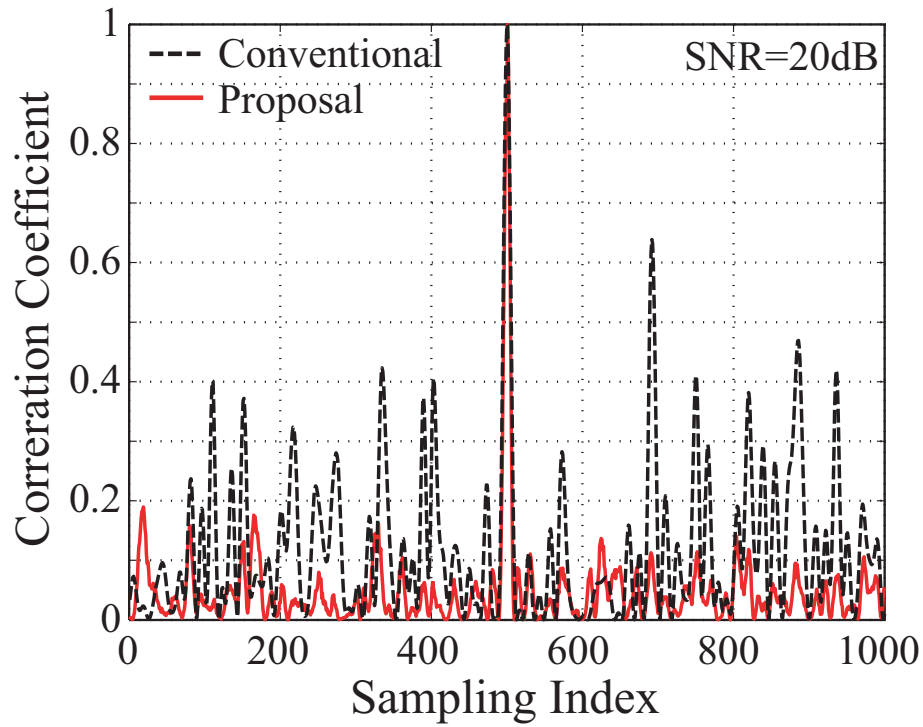


図 4.9: 従来法と提案法における相関係数の比較 (SNR=20dB)

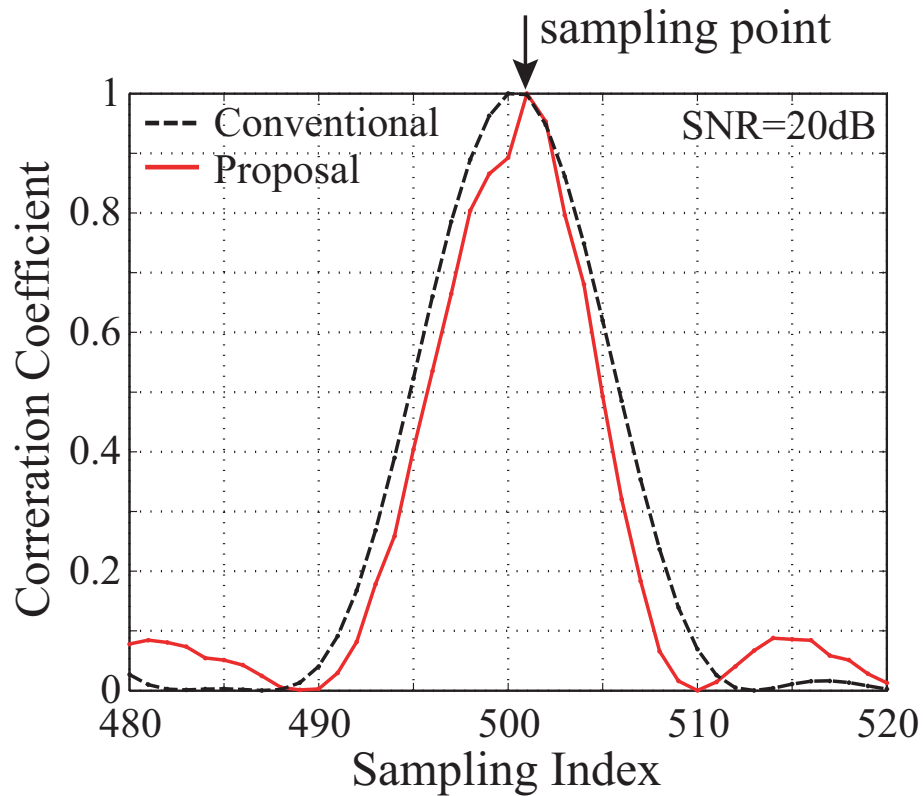


図 4.10: 従来法と提案法における相関係数の比較の拡大図 (SNR=20dB)

4.2.2 M 系列を用いたトレーニングシンボルの効果

4.2.1 節では提案したスライディング相関器の有効性を検証した．ここからの検討には図 4.6 で表される提案法のスライディング相関器を用いてシミュレーションを行う．M 系列の詳細は 4.1.1 節に述べた通りであるが，M 系列を用いることで実際にどの程度スライディング相関に改善をもたらすのかを検証するために，トレーニングシンボルに 5 次の M 系列符号 31 シンボルを用いた場合とランダム符号を 31 シンボル用いた場合でサンプル点に対する相関係数の分布を調べる．

図 4.11 は，M 系列を用いた場合（提案法）とランダム符号を用いた場合（従来法）における，スライディングさせたサンプルのインデックスに対する相関係数のプロットとなっている．この時のシミュレーション諸元は表 4.2 に示す通りである．提案法と従来法の相違点はトレーニングシンボルに M 系列符号を用いているかランダム符号を用いているかだけであるが，図 4.11 を見ると分かるように，相関係数のピークの鋭さも，ピーク付近以外の抑圧能力も M 系列を用いた場合の方が良い結果になっている．図 4.12 は図 4.11 を相関係数のピーク値付近で拡大したプロットであるが，ランダム符号を用いた場合は最適サンプリング点に相関係数のピーク値がきていないことが確認できる．

このように，トレーニングシンボル部分に M 系列符号を用いた場合とそうでない場合とを比較すると，スライディング相関後の結果からランダム符号を用いた場合は自己相関を効果的に高めることはできず，M 系列符号を用いた方が良い結果が得られたことが確認できた．

表 4.2: 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 0° , 干渉波 -40°
変調方式	QPSK
SIR	0 [dB]
SNR	0 [dB]
トレーニングシンボル	31 シンボル (5 次 M 系列符号 or ランダム符号)
フレーム長	400 シンボル (ダブルフレーム構成)
アップサンプルレート	10 倍
忘却係数	$q = 0.875$

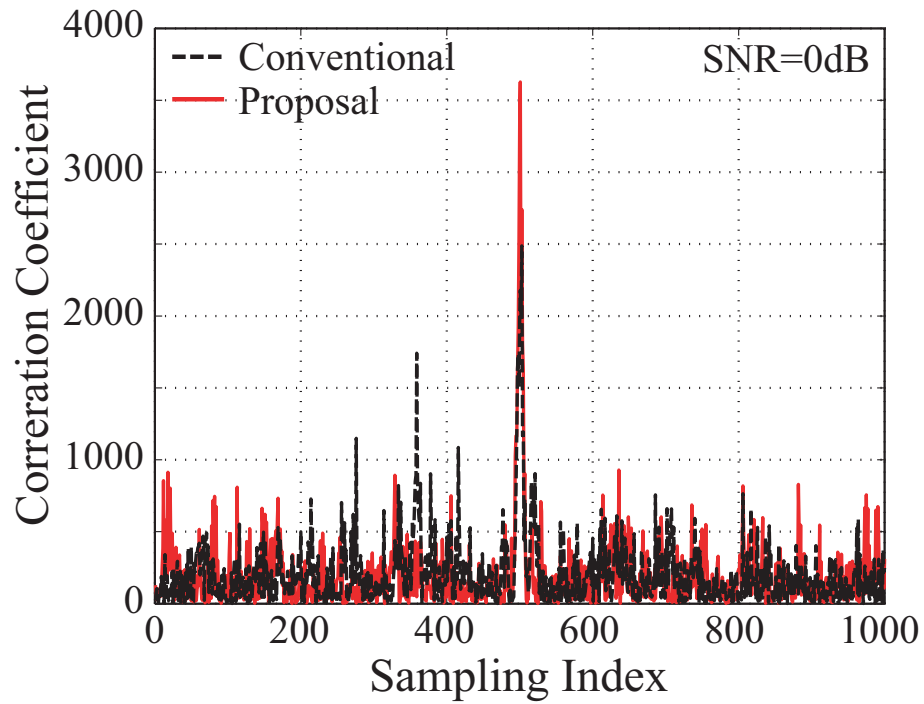


図 4.11: M 系列適用における精度比較 (SNR=0dB)

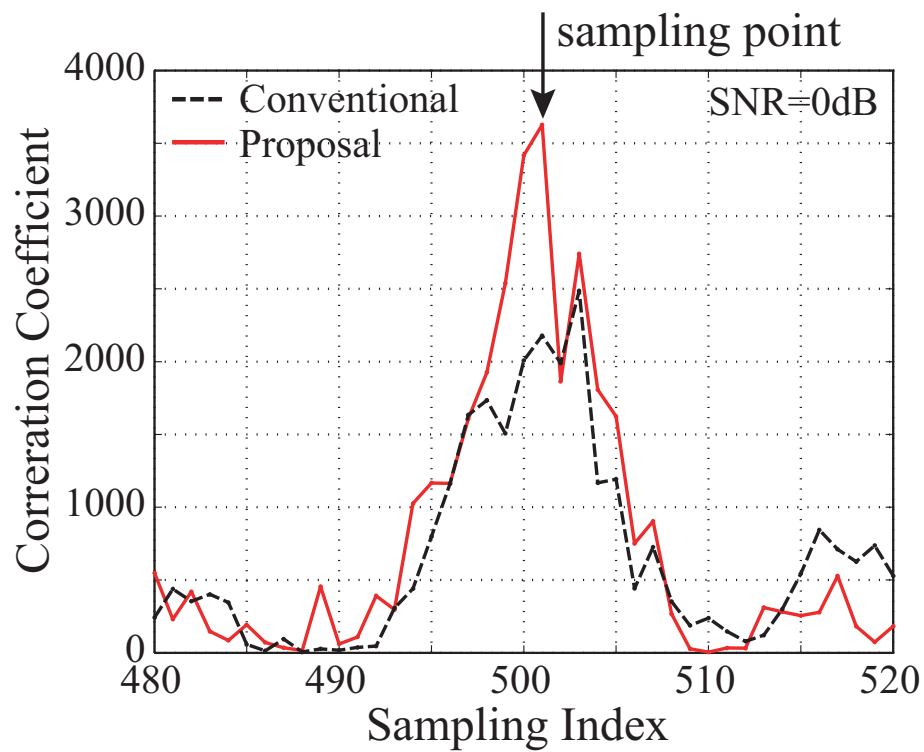


図 4.12: M 系列適用における精度比較の拡大図 (SNR=0dB)

4.2.3 アップサンプリングレートによる精度比較

これまでのシミュレーションではアップサンプリングレートを仮に10倍として検討してきたが、ここでアップサンプリングについての詳細を述べる。本研究で用いているシステムでは、送信側のDACで4倍のオーバーサンプリングを想定しており、DACのサンプリングレートは40MHzであるので、DACから出力するIF信号は10MHzとする必要がある。故に元のシンボルのデータを最低4倍以上でアップサンプリングする必要があり、ここではアップサンプリングレートを変更することで同期処理に及ぼす影響を検証する。

本研究におけるアップサンプリングとは、元のシンボルデータに零補間してシンボル数を増やし、さらにフィルタリングされることでシンボル間隔を増加させることを意味する。アップサンプリングレートを調節することでデータのシンボルレートを変更することができる。図4.13はアップサンプリングレートを10倍としたときのシンボル補間の一例である。ここで、提案する受信系システムにおいては受信機のIF出力が40MHzで固定となっていてデジタル信号処理ユニットにて32MHzでアンダーサンプリングを行う場合が考えられるので、シンボルレートによっては同期処理の際にデータを正確に間引くことができなくなってしまう。例として10倍でアップサンプリングを行う場合を考える。10MHzのIF信号に対して4倍のオーバーサンプリングを行うので送信アップサンプルは2.5倍となり、シンボルレートは $10\text{MHz}/2.5=4[\text{Msymbol/s}]$ となる。受信側で40MHzに対して32MHzでアンダーサンプリングされる場合、同期処理でデータを間引くレートが $32\text{MHz}/4[\text{Msymbol/s}]=8$ 倍となり、この場合は受信信号を8倍の間隔で間引けば元のシンボルデータが得られる。しかし、仮にアップサンプリングレートが8倍の場合を考えると、シンボルレートが $5[\text{Msymbol/s}]$ となり間引くレートが $32\text{MHz}/5[\text{Msymbol/s}]=6.4$ 倍となり、整数で表される受信信号のサンプル数に対して間引くレートが整数倍でないので間引くことができない。以上より、受信信号を間引くことのできる範囲内でアップサンプリングレートを変更して、アップサンプリングレートが及ぼす影響を調べる。

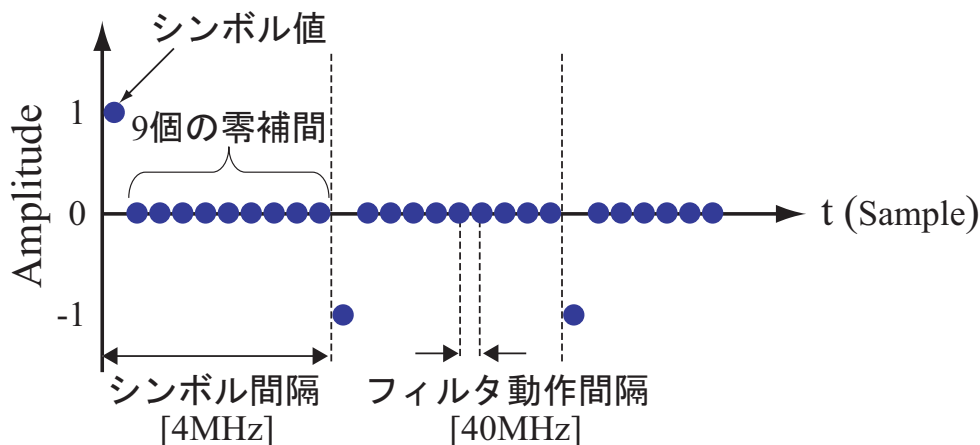


図 4.13: シンボルの零補間の例

表 4.3 に示すシミュレーション諸元にて，アップサンプリングレートを 5 倍，10 倍，20 倍と変化させた場合の同期処理に及ぼす影響を検証した．図 4.14，図 4.15 にそれぞれ結果を示す．図 4.14 は 3 通りのアップサンプリングレートにおける相関係数のプロットとなっている．図からアップサンプリングレートが大きくなるにつれて相関係数のピーク値付近に存在するサンプル点が増加していることがわかる．つまり，アップサンプリングレートが小さい場合の方がより鋭いピークが立っていることが確認できる．最適サンプル点は，アップサンプリングレートが 5 倍，10 倍の場合は正確に 501 番目のサンプルインデックスを示しているが，20 倍の場合については 1 サンプル誤ってピークが立っており，この場合正確なサンプル位置から間引くことができない．図 4.15 はスライディング相関によって得られたサンプルインデックスから受信信号を間引いた後に得られたベースバンド I,Q 信号のマッピングを示している．アップサンプリングレートが 5 倍の場合のプロットに注目すると，他の 2 つの場合に比べてコンスタレーションパターンが広がってしまっていることが確認できる．これは，アップサンプリングレートが少ない場合，同期処理後の最適なコンスタレーションパターン自体が広がってしまっているために，相関係数の大きさが最大となるインデックスで間引いても広がったパターンが得られてしまうことを意味する．図 4.16 にアップサンプリングレートが 3 通りの場合で変化する際のアイパターンを示す．図中の円で示されるサンプリングポイントの広がりを見ると，アップサンプリングレートが低いほど広がっていることが確認でき，この広がりがある限り最適なサンプリング点にて同期を確立してもコンスタレーションパターンは広がってしまう．

表 4.3: 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 0° , 干渉波 -40°
変調方式	QPSK
SIR	0 [dB]
SNR	20 [dB]
トレーニングシンボル	31 シンボル (5 次 M 系列符号)
フレーム長	400 シンボル (ダブルフレーム構成)
アップサンプルレート	5, 10, 20 倍
忘却係数	$q = 0.875$

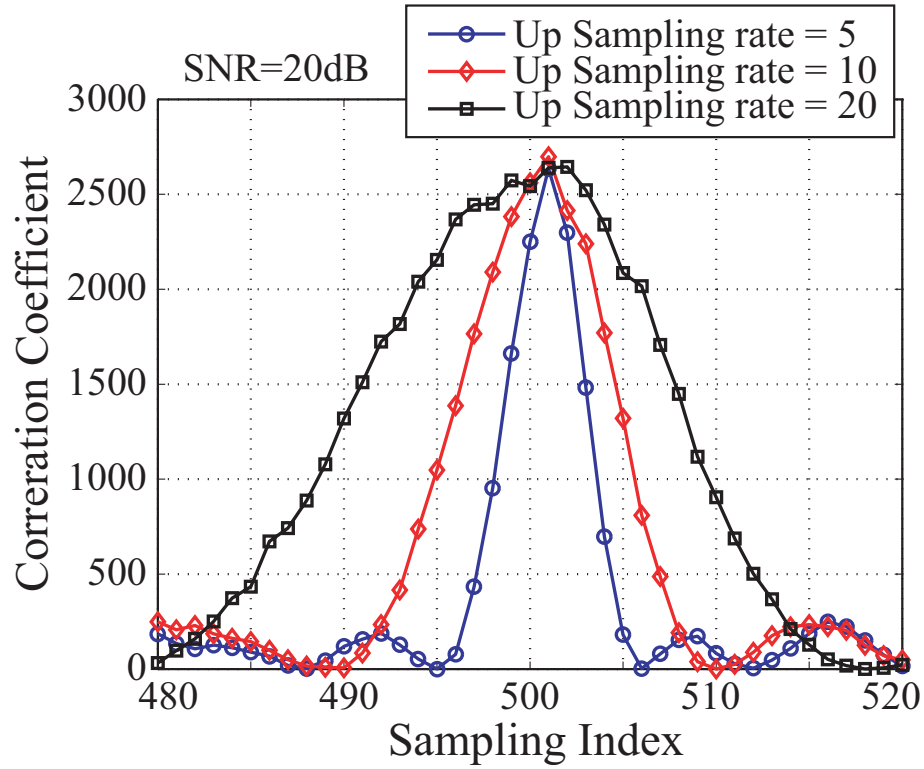


図 4.14: アップサンプリングレート変化時の相関特性の比較 (SNR=20dB)

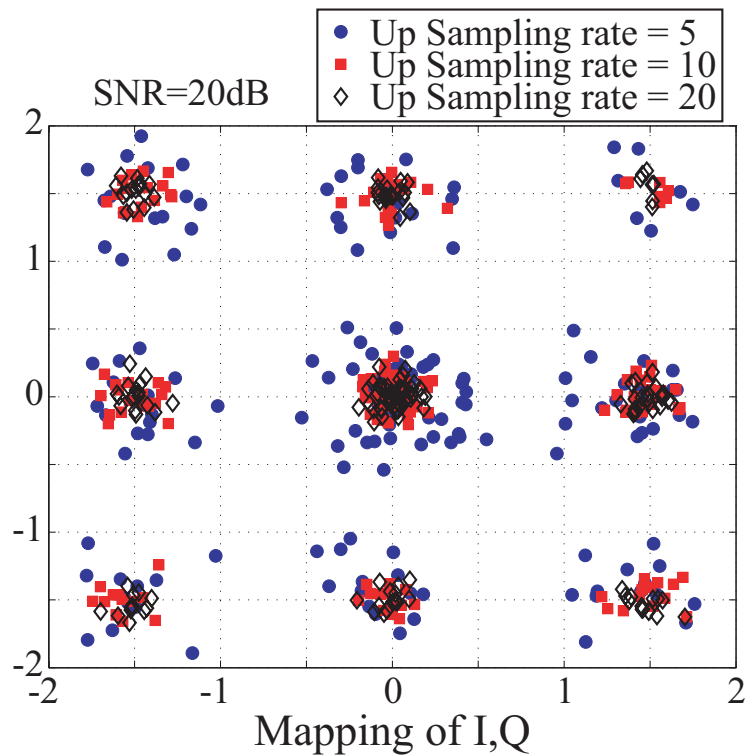


図 4.15: アップサンプリングレート変化時の I,Q パターンの比較 (SNR=20dB)

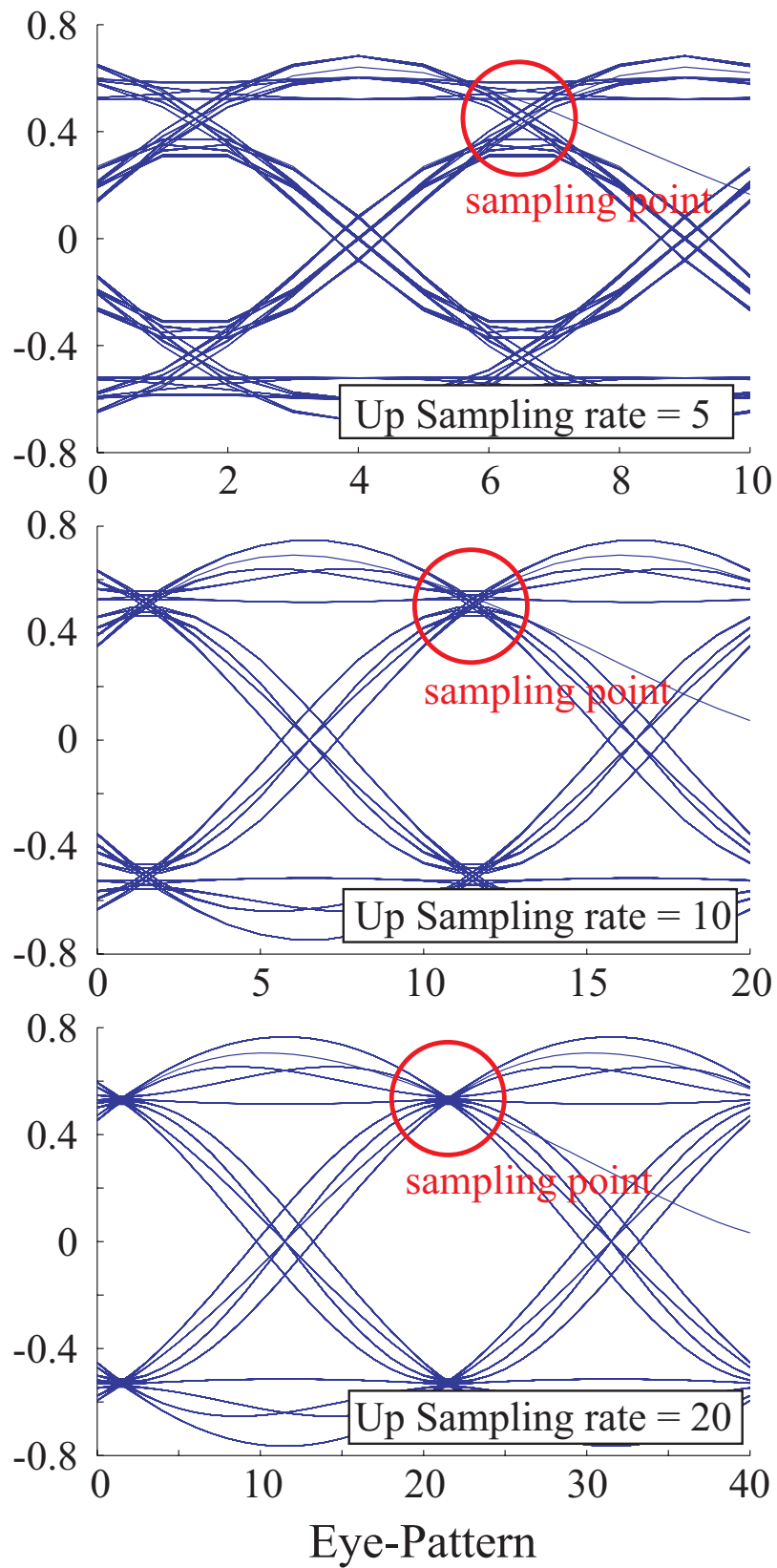


図 4.16: アップサンプリングレート変化時のアイパターンの比較 (SNR=20dB)

以上の検討よりアップサンプリングレートについて考察すると、アップサンプリングレートが大きくなると相関係数のピーク値付近のサンプル点が増加して正しいインデックス位置でピークが立たない可能性が生じ、その場合は最適サンプルインデックスより間引いた場合に比べてコンスタレーションパターンは広がってしまう。また、アップサンプリングレートが小さい場合、正確に相関係数のピーク位置を検出できても最適なコンスタレーションパターン自体が広がってしまっているために、求めた最適インデックスより間引いてもコンスタレーションパターンは広がってしまい結果的に BER の低下を招く恐れがある。本検討では、アップサンプリングレートを大きくするとサンプル数が増加してその分計算負荷が増大してしまうことも考慮して、コンスタレーションパターンの広がりも大きくなく相関係数のピーク値付近のサンプル点も多すぎない、アップサンプリングレートが 10 倍の場合が最適値であると考察した。

4.3 同期処理器の FPGA 実装

前節では提案する同期処理器の精度について検証したが、本節では提案したスライディング相関器を用いてトレーニングシンボルには M 系列符号を使用しアップサンプリングレートも 10 倍として実際に同期処理器を FPGA に実装し、実装結果や新たに提案するシステムの構成について説明する。

4.3.1 FPGA 実装結果

4.2 節で検討してきた提案法による同期処理器を新たに構成し、その FPGA 実装結果を従来法を用いた同期処理器の FPGA 実装結果と比較した結果を表 4.4 に示す。図のような構成を持つ従来法の同期処理器のスライディング相関では、相関係数は式 (3.7) で与えられるが、分母のノルムの除算は相関係数の大きさを求めて比較する際には省略可能な計算なので、表 4.4 の従来法の結果は計算を省略した場合の FPGA 実装結果である。従来法と提案法の結果を比較すると、提案法による同期処理器の方が約 3~4 倍の回路容量及び 1 サンプル当たりのクロック数を削減できていることが分かる。最大動作周波数は共に約 80MHz であるので、1 サンプル当たりに必要とするクロック数より 1 フレーム当たりに要する計算時間を計算した結果、提案法は従来法の約 1/4 の計算時間で処理が可能なが確認できる。実際に動作する周波数は ADC や DDC と同じ 40MHz のクロックを使用するので、そのときの計算時間は表 4.4 に示す処理時間の 2 倍となる。

表 4.4: 同期処理器の FPGA 実装結果

	従来法同期処理器	提案法同期処理器
回路容量 (総ゲート数に対する占有率)	25.1 %	7.6 %
DSP ブロック使用量	100 %	14 %
最大動作周波数	80 MHz	80 MHz
クロック数 / サンプル	35 clock	9 clock
処理時間 / フレーム	875 μ sec	225 μ sec

4.3.2 システムの構成

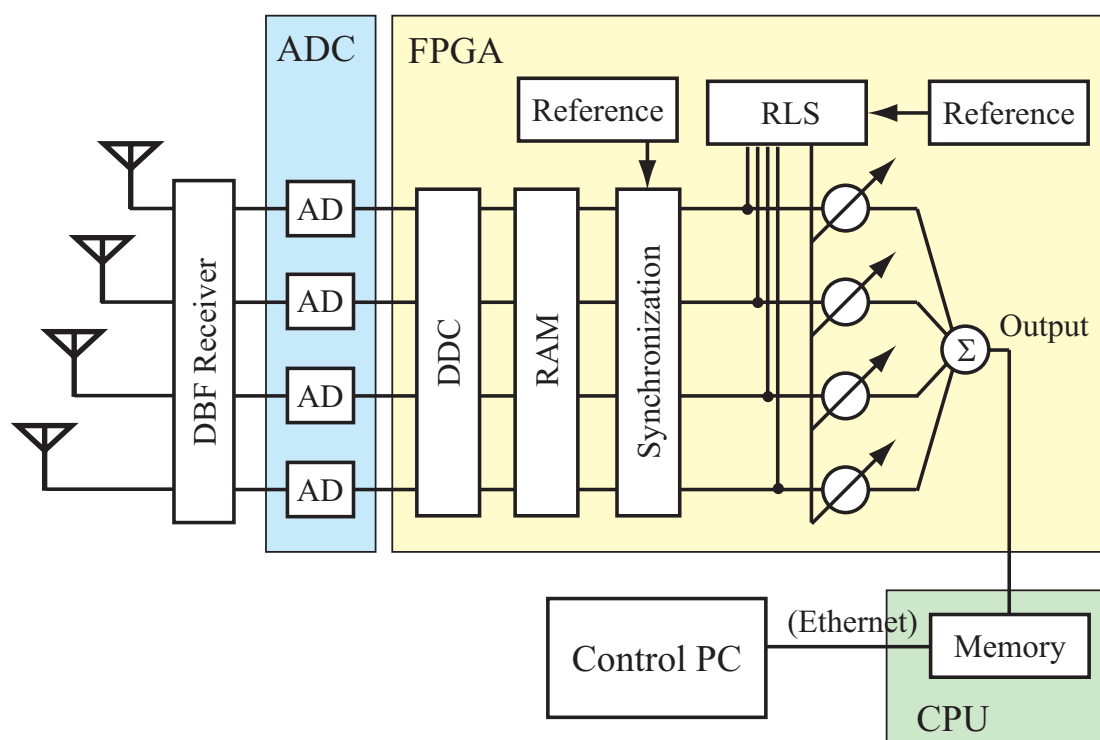


図 4.17: 同期処理器が FPGA 実装された場合のシステム構成

前章では，図 3.4 に示すような受信系システムを提案して実験を行ったが，ここでは提案した同期処理器を FPGA で実装することでまた新たに受信系システムの構築を行う．従来まで同期処理はデジタル信号処理ユニットのオンボード CPU で行っていたが，FPGA で実装したことによって信号処理のブロックチャートは図 4.17 のように表すことができる．

RF 帯より到来した信号がデジタル信号処理ユニット内の FPGA にて各素子に対して DDC されてベースバンド I,Q 信号が得られるまでの処理は従来法と相違ないが，提案するシステムではその後信号は RAM (Random Access Memory) へと書き込まれ記憶される．FIFO を用いた方が書き込みや読み出しのアクセスが容易であるが，一度メモリから取り出してしまうとメモリ内には次々と別のデータが保存されるような構成になっているため

に，FPGA 内で同期処理を行うために FIFO から受信データを読み出しても間引く際には受信データは FIFO には存在しないので，一度読み出しても保存が可能な RAM を用いている．RAM は FIFO と違って書き込みと読み込みにメモリのアドレスを指定することが可能であり，スライディング相関によって求めた最適なサンプルインデックスからアップサンプリングレートに基づく間隔でデータを間引くことも容易に実現できる．

従来では CPU で同期処理を行ってそのデータを再び FPGA に戻して RLS アルゴリズムへと入力するような構成であったが，FPGA で同期処理器を実装することによって一連の流れが全て FPGA 上でのやり取りとなるのでより高速な演算が期待できる．ここで，同期処理器に入力する 1 ビットで表現される参照信号は，あらかじめ FPGA 内の ROM (Read Only Memory) にて書き込まれており，同期処理器に入力される信号は受信信号のみとなっている．

4.4 同期処理器を用いたリアルタイム実験

前章にて行った実験は，所望波と干渉波が 2 波混在する伝搬環境ではスライディング相関を用いて必ずしも同期確立できるとは限らず，所望波と干渉波を異なるタイミングで出力してデータを取り，後に合成して 2 波同時送信された状況を想定していた．また，同期処理もオンボードの CPU にて計算させており演算時間も数 *sec* を必要とした．本節では，前節にて提案した受信系システムにおいて FPGA 実装された同期処理器を用いて，上記問題を解消しリアルタイムで一連の流れを計算して結果を出力するリアルタイム実験を行う．

4.4.1 実験の構成

図 4.17 に示すようなシステムを用いて実験を行うためには，周波数オフセットの影響を考慮しなければならない．周波数オフセットとは，準同期検波において発振回路の精度や安定性による問題から送信搬送波と受信機の準同期検波用基準信号との間に生じる周波数のオフセットのことであり，周波数オフセットが生じると所望信号と参照信号との相関値が低下する原因となってしまいうため [16]，所望信号と干渉信号の 2 波を同時に送信するような伝搬環境では同期を確立することが難しいと考えられる．周波数オフセットの影響を改善する手法として様々な方式が提案されているが [17][18]，シミュレーションによる検討に留まっており実際に FPGA などのハードウェアで実装して検証している例は少ない．本研究で用いる FPGA にこれら提案方式を実装させるには，FPGA の回路容量が大きな問題となり実装が困難であることが挙げられ，またこのような検討は本論文の主旨からは外れるので，ここでは周波数オフセットが無視できるような環境での実験にてリアルタイム同期処理器の動作を確認する．

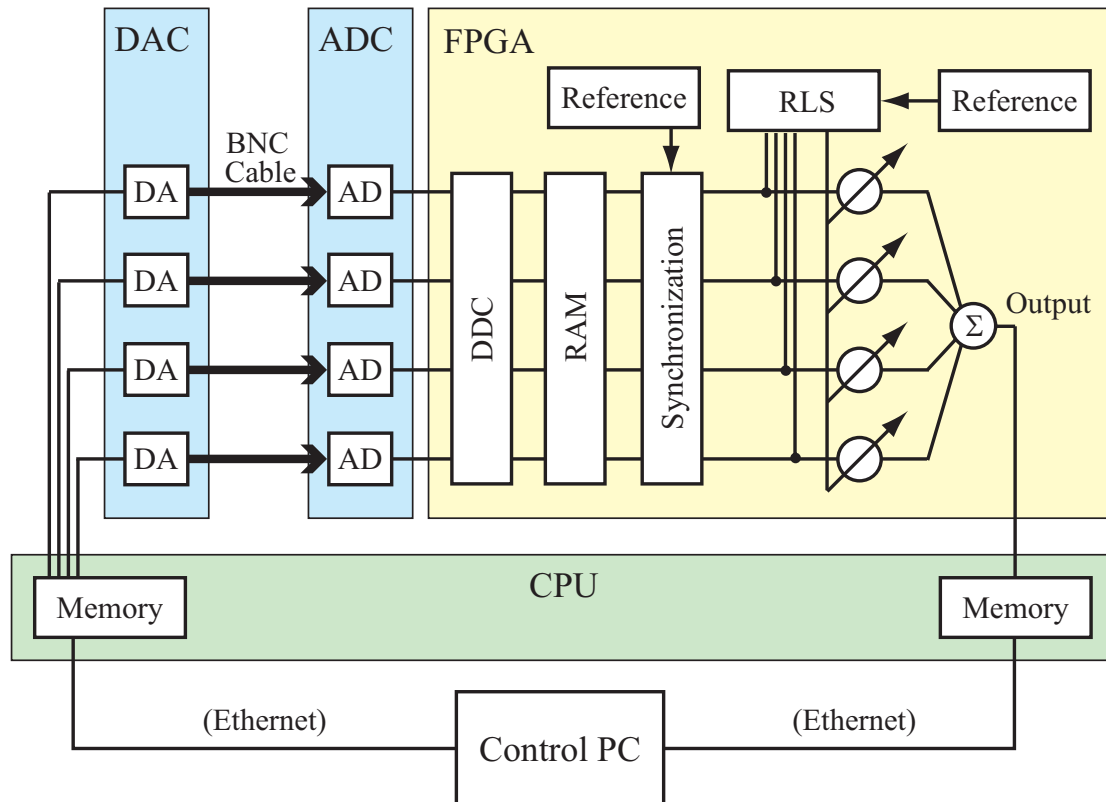


図 4.18: IF 帯直結実験の送受信システム構成

実験系システムの構成は図 4.18 に示すようになる。図 4.17 とは RF 帯を使用せずに IF 帯でデータを送受信している点で異なっており、このような場合、前述した周波数オフセットの影響は無視できるほどわずかなものとなり、所望信号と干渉信号が同時に到来する環境を想定して実験することができる。信号処理の流れとしては、まず制御用 PC にて 14 ビットのデジタル IF 信号を生成して、デジタル信号処理ユニットのオンボード CPU 内のメモリに読み込ませることで、DAC からアナログ IF 信号が出力される。出力された信号は 4 素子リニアアレーを想定して所望信号と干渉信号の 2 波がそれぞれの方向から到来するかという角度情報を含めて合成されており、出力された信号は BNC (Bayonet Neill Concelman) ケーブルを通じて ADC へと入力され、受信側では 2 波が同時に角度を持って到来した場合と同等の環境を想定することができる。その後の信号処理は 4.3.2 節にて述べた通りである。

また、実験の諸元を表 4.5 に示す。IF 帯を直結しての実験の場合は IF 周波数が送受信共に 10MHz であるので、受信側ではアンダーサンプリングする必要がなく、AD サンプルレートを 40MHz として 4 倍のオーバーサンプリングを行う。

表 4.5: 実験の諸元

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 0° , 干渉波 -40°
変調方式	QPSK
フレーム長	400 symbols (ダブルフレーム構成)
トレーニングシンボル	31 symbols (5 次 M 系列符号)
SIR	0 [dB]
SNR	変化
アップサンプルレート	10 倍
DA サンプルレート	40 MHz
AD サンプルレート	40 MHz
シンボルレート	4 Msymbol/s
送信 IF 周波数	10 MHz
受信 IF 周波数	10 MHz
忘却係数	$q = 0.875$

4.4.2 実験結果

図 4.18 に示すような構成で実験を行った．実際に IF 帯で DAC の出力ポートからを BNC ケーブルで直接 ADC の入力ポートに接続して実験を行っている様子を図 4.19 に示す．また，実験とシミュレーションによって得られた BER 特性を図 4.20 に示す．実験値は IF 帯を直接接続して提案した同期処理器にて同期を取って固定小数点演算を用いた RLS アルゴリズムにて干渉波を抑圧して得られた BER 特性であり，シミュレーション結果は実験の場合と同様の処理を行うシミュレーションにて求めており，RLS アルゴリズムを固定小数点演算を用いて計算した場合と浮動小数点演算を用いて計算した場合の 2 通りを示している．また，理論値は 1 素子で 1 波到来した場合の同期検波 QPSK 変調の BER の理論値であり，4 素子で 2 波到来する場合の理論値はこの曲線より 6dB 利得が改善された値となる．図 4.20 の横軸のビット当たりの SNR については，3.4 節にて述べられている AWGN の定義に基づいた同様の手段にて決定している．

図 4.20 を見ると，実験値とシミュレーションより得られた結果は大きな誤差もなく良く似た特性を示していることが確認できる．わずかに実験値の方が BER が劣化している原因として考えられることは，実験系による雑音が AWGN 以外にも何らかの形で加わってしまい SNR を決める際に誤差が生じてしまっていることや，AWGN についても十分長いサンプルにて平均電力を決定しているがわずかな誤差は生じてしまうこと等が考えられる．また，IF 帯を直接接続する際に送信データを固定小数点化する際に精度が劣化している可能性も要因として挙げられる．シミュレーションの結果が理論曲線から 6dB 利得



図 4.19: IF 帯直結実験の様子

改善された値よりも悪くなっていることに対して考えられる原因としては、BER を求める際に SNR に対してフレームを何回も受信してその都度同期処理を行っているが、相関係数の大きさの最大値は必ずしも常に最適なサンプリング点で現れるとは限らず数回に 1 度は最適なサンプリングインデックスより前後したサンプル点より間引いてしまうことがある。その時のコンスタレーションパターンはわずかに広がってしまうため、BER 特性にも当然影響が出てきてしまうと考えられる。

以上の結果より、FPGA で実装した同期処理器を用いて実験した結果得られた BER 特性がシミュレーションで得られた結果と傾向が良く似ていることから、その動作は正常に行われていると判断できる。また、動作速度についても理論上 μsec オーダーでの計算が実現できていることから、CPU を用いて同期処理を行った場合よりもリアルタイム処理が可能となっていると言える。

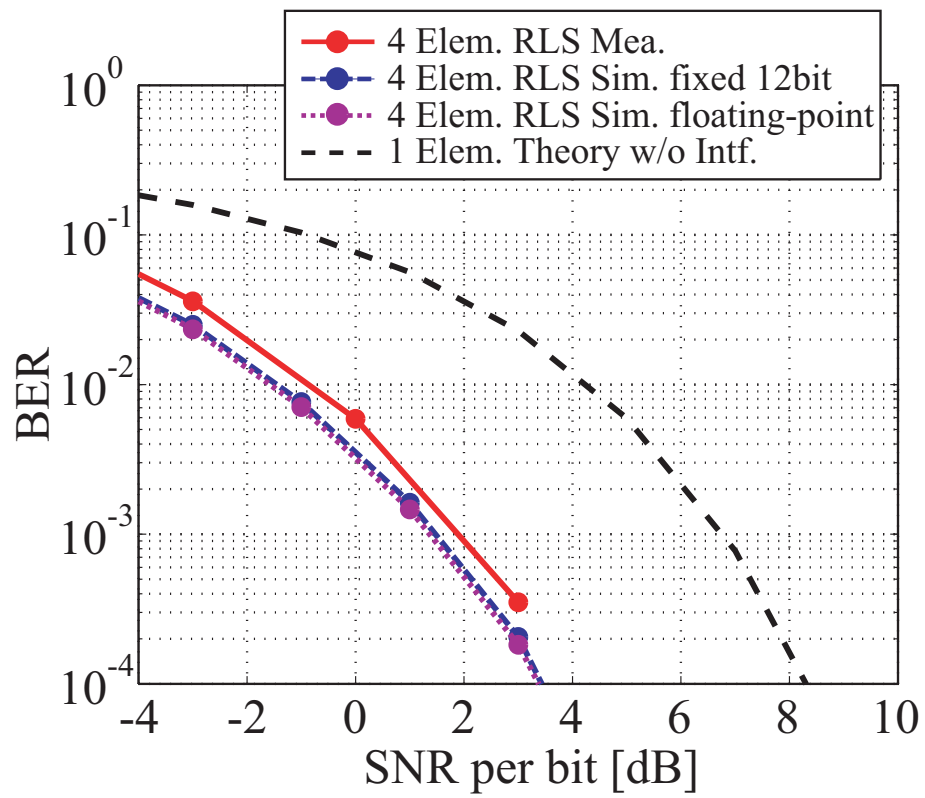


図 4.20: IF 直結実験とシミュレーションから得られた BER 特性

第 5 章

結論

本研究では、MMSE アダプティブアレーアンテナのリアルタイム処理を実現するために、固定小数点演算を用いた RLS アルゴリズムの FPGA 実装についてシミュレーションおよび実験による評価を行った。並列処理を得意とする FPGA を用いることにより処理速度の向上という利点が得られるが、FPGA では固定小数点演算が適しているため固定小数点演算を用いた RLS アルゴリズムの精度に関して検討を行った。

- 固定小数点演算における RLS アルゴリズムの入力情報や内部演算変数のワード長を実装上の制約を考慮して決定し、シミュレーションにより得られた収束特性から RLS アルゴリズムを浮動小数点演算と固定小数点演算でそれぞれ計算した場合、RLS アルゴリズムの動作は SNR が高くなるにつれて固定小数点演算を用いた場合には劣化が見られるものの、SNR=10dB 以下の場合に関しては浮動小数点演算を用いた場合と若干収束速度に誤差が見られたが干渉波抑圧能力には大きな違いはなかった。
- 除算においてシフト演算を用いた場合と除算器を用いた場合において、SNR が低い場合はシフト演算を用いた除算と除算器を用いた場合でほぼ同程度の精度が得られたが、SNR が高い場合に関してはシフト演算を用いた除算の方がやや収束速度が遅くなっていることが分かった。しかし、最終的な抑圧能力は同程度であり、BER 特性からも大幅な特性劣化は見られなかった。
- 角度依存性については、到来 2 波が近接している場合に精度劣化が見られたものの、浮動小数点演算と固定小数点演算の場合で BER 特性による比較の結果極端な誤差は見られず同精度に計算できることを確認した。

以上のシミュレーションによる検討から、アンテナ素子数が 4 素子の場合において固定小数点演算を用いても高精度に RLS アルゴリズムを計算することができることを検証した。

また、実際に固定小数点演算を用いた RLS アルゴリズムを FPGA に実装した結果を示す。

- RLS アルゴリズムの回路容量は FPGA 全体の約 35% を占め、1 イタレーション当たりの処理時間は約 $2.2\mu\text{sec}$ となった。
- ビームパターン特性から、所望波方向にメインビームを向け干渉波方向にヌルを向けていることが確認でき、RLS アルゴリズムが干渉波の影響を取り除くように動作していることが確認できた。
- BER 特性から、電波暗室内実験より得られた結果は理想的な曲線を描いているシミュレーション結果によく似た傾向を示していることが分かるので、BER 特性からも RLS アルゴリズムの有効性を示すことができた。

以上の結果より、固定小数点演算を用いて FPGA 実装された RLS アルゴリズムは、伝搬実験においても干渉波の影響を十分に排除し所望信号を取り出すことで良好な特性が得られることが確認でき、提案システムの有効性を実証することができた。

最後に、リアルタイム性を向上させるために同期処理器の構成を変更して FPGA で実装した結果を示す。

- 同期処理器において、フレームの構成でトレーニングシンボルに M 系列符号を用いることで相関係数の特性を改善することができ、またスライディング相関においてアップサンプリングレートに応じて間引いて相関係数を計算する方が回路規模が大幅に削減できることを示した。
- 提案した同期処理器の回路容量は全体の約 7.6% となり、1 フレームの同期を取るのに要する処理時間は約 $225\mu\text{sec}$ となった。
- 2 波を同時に送信した IF 帯直接接続における実験から得られた BER 特性より、実験値とシミュレーションより得られた結果は大きな誤差もなく良く似た特性を示していることから、FPGA 実装した同期処理器は正常に動作していることが実証された。その動作速度は μsec オーダーでの計算が実現できていることから、リアルタイム処理を実現することができた。

今後の展望としては、周波数オフセットの影響を改善して電波暗室や屋外環境におけるリアルタイム伝搬実験などが考えられる。

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、合同輪講など様々な場面において的確な御指摘を下さいました市毛弘一助教授に深く感謝致します。

アダプティブアレーアンテナやFPGAを始め、研究生活全般に渡って御指導下さいました株式会社ブレインズの金ミン錫 (Minseok Kim) 様に深く感謝致します。

最後に研究生活を共に過ごした新井研究室、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] Y. Ogawa, M. Ohmiya and K. Itoh, “An LMS adaptive array for multipath fading reduction,” IEEE Trans. Aerosp. Electron. Syst., vol.AES-23, no. 1, pp. 17–23, Jan. 1987.
- [3] J. Winter, “Smart Antennas for Wireless Systems,” IEEE Personal Commun., vol. 5, no. 1, pp. 23–27, Feb. 1998.
- [4] 藤元美俊, 堀俊和, “移動通信環境における LMS アダプティブアレーの収束特性の改善,” 信学論 (B), vol. J87-B No.9 pp.1240-1248, Sept. 2004.
- [5] N. Kikuma and K. Takao, “Effect of initial values of adaptive arrays,” IEEE Trans. Aerosp. Electron. Syst., vol.AES-22, no. 6, pp. 688–694, Nov. 1986.
- [6] L. Godara , “Applications of Antenna Arrays to Mobile Communications, Part 2: Beam-Forming and Direction-of-Arrival Considerations,” Proc. IEEE, vol. 85, no. 8, pp. 1195–1245, Aug. 1997.
- [7] T. Asai and T. Matsumoto, “A Systolic Array RLS Processor,” IEICE Trans. Commun., Vol. E84-B, NO. 5, pp.1356–1361, May. 2001.
- [8] S. Muramatsu, M. Kim, K. Ichige and H. Arai, “A Prototype Transmitter Using FPGA and Adaptive Antenna Array for High-Speed Mobile Communication System,” Technical Report of IEICE, Wireless Communication Workshop, AP2003-14, YRP, JAPAN, Mar. 2003.
- [9] M. 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.

- [10] M. Kim, K. Ichige and H. Arai, "Implementation of FPGA based Fast Unitary MUSIC DOA Estimator," IEICE Trans. Electron., Vol. E87-C, no. 9, pp. 1485–1494, Sept. 2004.
- [11] 小川恭孝, "アンテナ・伝搬における設計・解析手法ワークショップ (第 23,24 回) アダプティブアンテナ中級コース," Nov. 2002.
- [12] 神谷幸宏, 唐沢好男, "時間及び周波数領域信号処理を行う適応型アレーアンテナの種々の構成における特徴比較と収束特性改善," 信学論 (A), vol. J82-A, No. 6, pp. 867–874, Jun. 1999.
- [13] Stratix Device Handbook, Altera Corp., Web document is available at http://www.altera.com/literature/hb/stx/stratix_handbook.pdf.
- [14] A. Kiyono, M. Kim, K. Ichige and H. Arai, "Jitter Effect on Digital Downconversion Receiver with Undersampling Scheme," in Proc. The 2004 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS2004), Hiroshima, Japan, July 2004.
- [15] 三瓶政一, "デジタルワイヤレス伝送技術," 科学技術出版
- [16] 浜口清, "陸上移動通信用 QAM の複数パイロットシンボルを用いた周波数オフセット補償方式," 信学論 (B2), vol. J79-B2, No. 7, pp. 426–428, July 1996.
- [17] 尾保手茂樹, 市川佳弘, 鹿子嶋憲一, "アダプティブアレーアンテナにおける参照信号同期法に関する一検討," 2005 信学ソ大, B-1-212, Sept. 2005.
- [18] 市川佳弘, 尾保手茂樹, 鹿子嶋憲一, "MMSE-SMI アダプティブアレーアンテナにおける周波数オフセット補償法," 信学論 (B), vol. J86-B, No. 9, pp. 1914–1924, Sept. 2003.
- [19] M. Kim, "Hardware Implementation of Signal Processing in Smart Antenna Systems for High Speed Wireless Communication," Doctoral Dissertation, Department of Electrical and Computer Engineering, Yokohama National University, Yokohama, Japan, Dec. 2004.
- [20] 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.
- [21] 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.
- [22] 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.

- [23] 米澤, 小川, 西村, 大鐘, 土居, “MMSE 基準に従うアダプティブアンテナにおける参照信号同期法に関する検討,” Technical Report of IEICE, AP2000-131, Oct. 2000.
- [24] Y. Ogawa, T. Ohgane, “Adaptive Antennas for Future Mobile Radio,” IEICE Trans., Vol. E79-A, No.7 July 1996.
- [25] Y. Ichikawa, “A Computation Reduced MMSE Adaptive Array Antenna Using Space-Temporal Simultaneous Processing Equalizer,” IEICE Trans. Commun., Vol. E85-B, No.12 Dec. 2002.
- [26] S. Tsukamoto, “A Complex Baseband Platform for Spatial-Temporal Mobile Radio Channel Simulation,” IEEE Trans. VT., Vol. 51, No.6, Nov. 2002.
- [27] Sandeep Mukthavaram, “Design and FPGA Implementation of an Adaptive Demodulator,” B.S.E.E Osmania University, Hyderabad, India, 1997
- [28] 関清三, “デジタル変復調の基礎,” オーム社出版, 2001 年
- [29] 高木直史, “算術演算の VLSI アルゴリズム 並列処理シリーズ (5),” コロナ社
- [30] 並木秀明, “VHDL によるデジタル回路入門,” 技術評論者, 2001 年
- [31] 仲野巧, “VHDL によるマイクロプロセッサ設計入門,” CQ 出版社, 2002 年
- [32] 長谷川裕恭, “VHDL によるハードウェア設計入門改訂版,” CQ 出版社, 2004 年
- [33] 斉藤洋一, “デジタル変復調,” 和歌山大学, MWE 2003 Microwave Workshop Digest.
- [34] 中村尚五, “デジタルフィルタ,” 東京電機大学出版局

発表文献

- [1] 中島淳, 金ミン錫, 新井宏之, "FPGA を用いた MMSE アダプティブアレーの性能評価," 2004 信学ソ大, B-5-78, Sept. 2004.
- [2] Atsushi Nakajima, Minseok Kim, Hiroyuki Arai, "Performance Evaluation of MMSE Adaptive Array using FPGA," Korea-Japan Joint Conference on AP/EMC/EMT, Seoul, Korea, Oct. 2004.
- [3] Atsushi Nakajima, Minseok Kim, Hiroyuki Arai, "FPGA Implementation of MMSE Adaptive Array Antenna using RLS Algorithm," Antennas and Propagation Society, Washington DC, USA, July 2005.
- [4] 中島淳, 金ミン錫, 新井宏之, "FPGA を用いた RLS アダプティブアレーによる室内伝搬実験," 2005 信学総大, B-1-276, March 2005.
- [5] 中島淳, 金ミン錫, 新井宏之, "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," 信学論 (B), vol. J88-B No.9 pp.1772-1779, Sept. 2005.
- [6] 中島淳, 新井宏之, "リアルタイム同期処理器の FPGA 実装," 2005 信学ソ大, B-1-213, Sept. 2005.