

飽和型の未知システムと適応フィルタを 有する適応信号処理システムの解析

理 工 学 研 究 科 シ ス テ ム 理 工 学 専 攻
電 気 電 子 情 報 工 学 分 野 情 報 数 理 工 学 研 究 室

23M6331 伊原 大地

指 導 教 員

印

論文要旨

飽和型の未知システムと適応フィルタを有する適応信号処理システムの解析

23M6331 伊原 大地

適応信号処理は係数ベクトルが可変である適応フィルタを使用し、周辺の環境や対象となる信号の性質およびそれらの時間的変動に応じて処理方法を変化させて適応フィルタの出力が目的の出力に近づくようにすることで実現される。この技術はアクティブノイズコントロールや音響エコーチャンセラ、システム同定など、幅広い分野で利用されている。適応フィルタの更新則の一つとして最小二乗平均(LMS)アルゴリズムがある。これは確率的勾配降下法の一つであり、適応フィルタの代表的な更新則でもある。

一方、物理学の中に統計力学と呼ばれる分野がある。これを情報分野に応用したものは情報統計力学と呼ばれている。情報統計力学はシステムサイズが大きい極限を考えることにより問題の個別詳細によらない普遍的性質を巨視的に議論することをその精神としている。これまで情報統計力学の枠組みで、構成要素が線形な場合の適応信号処理の解析が進められてきた。しかし、実環境で利用する場合は適応システムに非線形な要素が含まれる場合がある。

本論文では、LMSアルゴリズムを用いた適応信号処理システムにおいて、未知システム出力と適応フィルタ出力の両方が飽和型非線形性を有する場合の動的・静的ふるまいを統計力学的手法を用いて理論的に明らかにする。すなわち、二個の巨視的変数を導入し、それらに関する連立微分方程式を導出し数値的に解くことで、二乗平均誤差(MSE)および二乗平均偏差(MSD)を導出する。その結果、理論的に得られたMSEおよびMSDの動的ふるまいが計算機実験とよく一致した。この一致はステップサイズの大きさや飽和値の大きさを変化させても同様であった。解析的に求めることができないサンプル平均について二重積分から一重積分に変換を行うことで、数値計算の精度、速度を大きく改善することに成功した。未知システム側の飽和値 S_1 と適応フィルタ側の飽和値 S_2 が等しい場合、MSE、MSDは0に漸近した。ステップサイズが大きくなるとMSE、MSDが0に収束するのが速くなる一方、数値計算が不安定になりやすいことが分かった。さらに、MSE、MSDの定常値についても調べた。ステップサイズ $\mu=0.1$ において数値計算が安定する時刻300でのMSE、MSDをそれぞれ準定常MSE、準定常MSDとして調べた。準定常MSE、準定常MSDは、どちらも $S_1 = S_2$ の場合に最小値をとるが、 S_1, S_2 の値によっては時刻300でも収束せず準定常MSE、準定常MSDが最小となる値は $S_1 = S_2$ から少しずれる結果となった。またヒートマップにより二つの飽和値が準定常MSE、準定常MSDに与える影響を明らかにした。 $S_1 = S_2$ に

において準定常 MSE, 準定常 MSD は最小値をとること, S_1 が大きく, S_2 が小さい場合に準定常 MSE, 準定常 MSD が大きくなることが分かった.

目 次

第 1 章 序論	1
第 2 章 適応信号処理と LMS アルゴリズム	3
2.1 適応信号処理	3
2.2 LMS アルゴリズム	4
第 3 章 未知システム出力と適応フィルタ出力に非線形性を有するシステムの解析	5
3.1 モデル	5
3.2 理論	6
3.2.1 二乗平均誤差	6
3.2.2 二乗平均偏差	9
3.2.3 巨視的変数 r の微分方程式	9
3.2.4 巨視的変数 Q の微分方程式	10
第 4 章 未知システム出力と適応フィルタ出力に飽和型非線形性を有するシステムの解析	12
4.1 飽和型非線形性	12
4.2 理論	13
4.2.1 二乗平均誤差	13
4.2.2 二重積分から一重積分への変換	13
4.2.3 巨視的変数 r の微分方程式	18
4.2.4 巨視的変数 Q の微分方程式	19
第 5 章 結果と議論	20
5.1 二乗平均誤差	20
5.1.1 二重積分と一重積分の違い	20
5.1.2 学習曲線	21
5.1.3 二つの飽和値と準定常二乗平均誤差の関係	22
5.2 二乗平均偏差	27
5.2.1 学習曲線	27

5.2.2 二つの飽和値と準定常正規化二乗平均偏差の関係	29
第6章 結論	36
謝辞	38
参考文献	39
付録A 計算の詳細	41
A.1 $\langle f_G^2(d) \rangle$	41
A.2 $\langle f_W^2(y) \rangle$	42
A.3 $\langle df_G(d) \rangle$	43
A.4 $\langle df_W(y) \rangle$	43
A.5 $\langle yf_G(d) \rangle$	47
A.6 $\langle yf_W(y) \rangle$	50
付録B プログラムリスト	51
付録C 関連発表	91

第1章 序論

ディジタル信号処理技術は、携帯電話、ドローン、ワイヤレスマイクなどをはじめとした情報信号分野を中心に利用されている。ディジタル信号処理は様々な分野において応用されており、具体的には信号から必要な情報を抽出することや情報をできる限り正確に伝達するための信号の加工などがある [1]。ディジタル信号処理技術の一つに適応信号処理がある。適応信号処理は周囲の環境や対象となる信号の性質、それらの時間的変動に応じて処理方法を変化させることにより、目的の出力に近づくようになる技術である。これは適応フィルタを用意し、その出力が目的とする信号に近づくように係数を繰り返し更新していくことで実現される [2], [3]。適応信号処理は、通信や音響システムなどの分野で広く利用されており、アクティブノイズコントロール(ANC) [4], [5], [6], [7]、音響エコーチャンセラ [8]、システム同定など、さまざまな応用において重要な役割を果たしている。

適応フィルタのインパルス応答は最適化アルゴリズムを用いて誤差信号が最小化されるように更新される。代表的な最適化アルゴリズムとしてLMSアルゴリズムがある。LMSアルゴリズムは確率的勾配降下法と呼ばれる手法の一種でコスト関数である二乗平均誤差の期待値が小さくなるようインパルス応答を更新するものである。

一方、物理学の中に統計力学と呼ばれる分野がある。これを情報分野に応用したものは情報統計力学と呼ばれている [9], [10]。情報統計力学はシステムサイズが大きい極限を考えることにより問題の個別詳細によらない普遍的性質を巨視的に議論することをその精神としている [11]。

これまで情報統計力学の枠組みでは線形ディジタルフィルタの解析が進められてきた。しかし、実環境で利用する場合はフィルタが非線形なふるまいをすることが多々発生する。実際に、適応信号処理システムではスピーカー、マイク、アンプなどのハードウェアによる、無視できない本質的な非線形性が存在する場合が多い。これらの非線形性がシステムの全体的な性能に与える影響を調べることは、非常に重要である。従来の研究では、適応フィルタ側の非線形性、すなわちアクティブノイズコントロールのモデルを中心に解析が行われていた [12], [13], [14]。また、未知システム側の非線形性、すなわち音響エコーチャンセラのモデルに関する解析も行われている [15]。しかし、未知システム出力と適応フィルタ出力の両方に非線形性が存在するモデルについては未だに解析が行われていない。

本論文では、オンライン学習に対する統計力学的解析の手法を適用することにより、未知システム出力と適応フィルタ出力の両方が飽和型非線形性を有する適応信号処理システムにおける、MSE, MSD の動的・静的ふるまいを理論的に明らかにする。[16], [17], [18] すなわち、未知システムの係数ベクトルと適応フィルタの係数ベクトルの相互関、適応フィルタの係数ベクトルの自己相関を巨視的変数としてそれらの動的ふるまいを記述する連立微分方程式をフィルタのタップ長が十分に長いという条件の下で数値的に解く。解析的に解くことが困難なサンプル平均について、二重積分を一重積分に変換することで数値計算の精度、速度が改善されることを確認する。また時刻が大きい場合の MSE を準定常 MSE および準定常正規化 MSD として、二つの飽和値がそれぞれに与える影響を調べる。

第2章 適応信号処理とLMSアルゴリズム

2.1 適応信号処理

適応信号処理のブロック図を図2.1に示す。適応フィルタの係数ベクトルは $w(n)$ 、適応フィルタのタップ入力ベクトルは $u(n) = [u(n), u(n-1), \dots, u(n-N+1)]^\top$ 、誤差信号は

$$e(n) = d(n) - y(n) + \xi(n) \quad (2.1)$$

とする。ここで \top は転置を表す。適応信号処理は適応フィルタを最適化アルゴリズムによって更新し誤差信号を最小化することによって達成される。

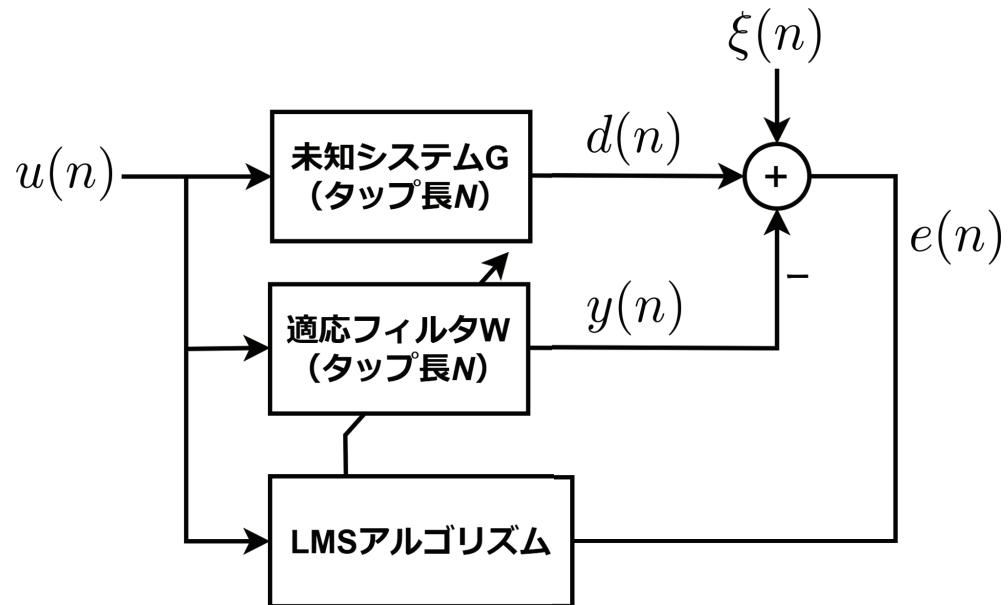


図 2.1: 適応信号処理のブロック図

2.2 LMS アルゴリズム

更新則の一つに勾配法があり、その更新式はコスト関数を J として

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \left(\frac{\partial J(n)}{\partial \mathbf{w}} \right) \quad (2.2)$$

と表される。 μ はステップサイズを表している。誤差信号の二乗平均誤差 $J(n) = \langle e^2(n) \rangle$ を代入すると

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \left(\frac{\partial \langle e^2(n) \rangle}{\partial \mathbf{w}} \right) \quad (2.3)$$

$$= \mathbf{w}(n) + \mu \langle e(n) \mathbf{u}(n) \rangle \quad (2.4)$$

となる。ここで $\langle \cdot \rangle$ は期待値を表す。この更新式はコスト関数として二乗平均誤差を使用しているため Least Mean Square(LMS) アルゴリズムと呼ばれている。しかし、一般的には期待値を計算することはできないので更新則は

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{u}(n) \quad (2.5)$$

として用いる。

第3章 未知システム出力と適応フィルタ出力に非線形性を有するシステムの解析

3.1 モデル

未知システムの出力と適応フィルタの出力に非線形性を有するような適応信号処理のブロック図を図 3.1 に示す。

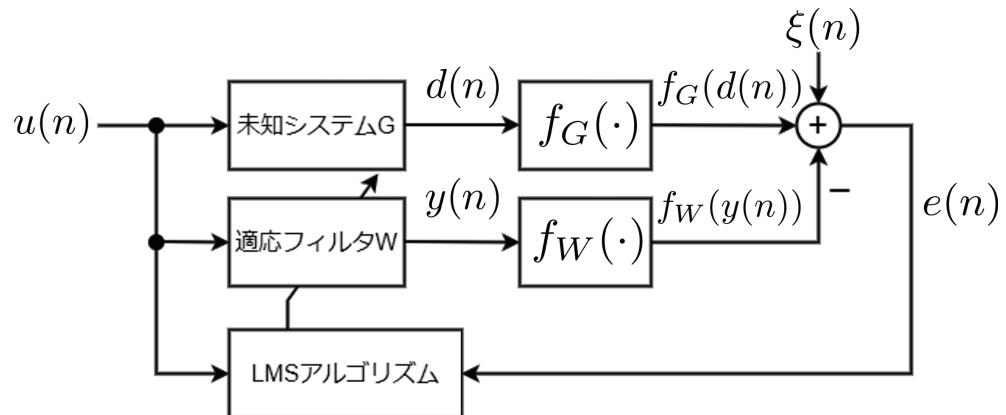


図 3.1: 未知システムの出力と適応フィルタの出力に非線形性を有するような適応信号処理のブロック図

未知システム G はタップ長 N の FIR フィルタでモデル化される。その係数ベクトル

$$\mathbf{g} = [g_1, g_2, \dots, g_N]^\top \quad (3.1)$$

は時不变である。ここで、

$$\sigma_g^2 \equiv \frac{1}{N} \| \mathbf{g} \|_2^2 = \frac{1}{N} \sum_{i=1}^N g_i^2 \quad (3.2)$$

と定義する。適応フィルタ W はタップ長 N で構成されるものとすると係数ベクトルは

$$\mathbf{w}(n) = [w_1(n), w_2(n), \dots, w_N(n)]^\top \quad (3.3)$$

である。ここで n は時刻ステップを表す。それぞれの要素の初期値 $w_i(0)$ はゼロである。入力信号 $u(n)$ は平均と分散がそれぞれ

$$\langle u(n) \rangle = 0, \quad \langle u(n)^2 \rangle = \sigma^2 \quad (3.4)$$

である分散から独立に生成される。時刻ステップ n におけるタップ入力ベクトルは

$$\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-N+1)]^\top \quad (3.5)$$

である。未知システム G と適応フィルタ W の出力をそれぞれ $d(n)$, $y(n)$ とすると

$$d(n) = \mathbf{g}^\top \mathbf{u}(n) = \sum_{i=1}^N g_i u(n-i+1) \quad (3.6)$$

$$y(n) = \mathbf{w}(n)^\top \mathbf{u}(n) = \sum_{i=1}^N w_i(n) u(n-i+1) \quad (3.7)$$

である。

誤差信号 $e(n)$ は $d(n)$ と $f(y(n))$ の差に、平均 0, 分散 σ_ξ^2 で時刻ステップごとに独立に生成された背景雑音 $\xi(n)$ を加えたもので、

$$e(n) = f_G(d(n)) - f_W(y(n)) + \xi(n) \quad (3.8)$$

である。ここで関数 $f_G(\cdot)$, $f_W(\cdot)$ はそれぞれ未知システム G と適応フィルタ W の出力の非線形性を表す。 $\mathbf{w}(n)$ は LMS アルゴリズムによって更新されるのでその更新式は

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{u}(n) \quad (3.9)$$

である。

3.2 理論

3.2.1 二乗平均誤差

式 (3.8) から二乗平均誤差は

$$\begin{aligned} \langle e^2(n) \rangle &= \langle (f_G(d(n)) - f_W(y(n)) + \xi(n))^2 \rangle \\ &= \langle f_G^2(d(n)) \rangle + \langle f_W^2(y(n)) \rangle - 2\langle f_G(d(n)) f_W(y(n)) \rangle + \sigma_\xi^2 \end{aligned} \quad (3.10)$$

である。また、以降では $\rho^2 \equiv N\sigma^2$ を一定に保ったままで $N \rightarrow \infty$ を考えることにする。式(3.10)の右辺第一項、第二項、第三項を計算するにあたり、 d と y がどのような確率分布に従うかを考える。 $d(n)$ と $d(n)$ の積の平均は

$$\begin{aligned}
\langle d(n)d(n) \rangle &= \left\langle \left(\sum_{i=1}^N g_i u(n-i+1) \right)^2 \right\rangle \\
&= \left\langle \sum_{i=1}^N \sum_{j=1}^N g_i g_j u(n-i+1) u(n-j+1) \right\rangle \\
&= \sum_{i=1}^N g_i^2 \langle u(n-i+1)^2 \rangle \\
&= \sigma^2 \sum_{i=1}^N g_i^2 \xrightarrow{N \rightarrow \infty} \rho^2 \sigma_g^2
\end{aligned} \tag{3.11}$$

である。また、 $d(n)$ と $y(n)$ の積の平均は

$$\begin{aligned}
\langle d(n)y(n) \rangle &= \left\langle \sum_{i=1}^N g_i u(n-i+1) \sum_{j=1}^N w_j(n) u(n-j+1) \right\rangle \\
&= \sum_{i=1}^N \sum_{j=1}^N \langle g_i w_j(n) \rangle \langle u(n-i+1) u(n-j+1) \rangle \\
&= \sum_{i=1}^N \sum_{j=1}^N g_i w_j(n) \langle u(n-i+1) u(n-j+1) \rangle \\
&= \sigma^2 \sum_{i=1}^N g_i w_i(n) \xrightarrow{N \rightarrow \infty} \rho^2 r
\end{aligned} \tag{3.12}$$

である。 $y(n)$ と $y(n)$ の積の平均は

$$\begin{aligned}
\langle y(n)y(n) \rangle &= \left\langle \sum_{i=1}^N w_i(n) u(n-i+1) \sum_{j=1}^N w_j(n) u(n-j+1) \right\rangle \\
&= \sum_{i=1}^N \sum_{j=1}^N \langle w_i(n) w_j(n) \rangle \langle u(n-i+1) u(n-j+1) \rangle \\
&= \sigma^2 \sum_{i=1}^N w_i^2(n) \xrightarrow{N \rightarrow \infty} \rho^2 Q
\end{aligned} \tag{3.13}$$

である。ここで、 r , Q はそれぞれ

$$r(n) \equiv \frac{1}{N} \mathbf{g}^\top \mathbf{w}(n) \quad (3.14)$$

$$Q(n) \equiv \frac{1}{N} \mathbf{w}(n)^\top \mathbf{w}(n) \quad (3.15)$$

で定義される巨視的変数である。式 (3.11), (3.12), (3.13) より $d(n)$ と $y(n)$ は平均が 0, 分散共分散行列が

$$\Sigma = \rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \quad (3.16)$$

である二次元ガウス分布にしたがう。次に、式 (3.10) の右辺第一項、第二項、第三項を計算する。式 (3.16) より、 $f_G(d(n))$ と $f_G(d(n))$ の積の平均は

$$\begin{aligned} \langle f_G(d(n)) f_G(d(n)) \rangle &= \int_{-\infty}^{\infty} f_G^2(d(n)) P(d(n)) dd \\ &= \int_{-\infty}^{\infty} f_G^2(d(n)) \frac{\exp(-\frac{d^2}{2\rho^2\sigma_g^2})}{\sqrt{2\pi\rho^2\sigma_g^2}} dd \end{aligned} \quad (3.17)$$

$f_W(y(n))$ と $f_W(y(n))$ の積の平均は

$$\begin{aligned} \langle f_W(y(n)) f_W(y(n)) \rangle &= \int_{-\infty}^{\infty} f_W^2(y(n)) P(y(n)) dy \\ &= \int_{-\infty}^{\infty} f_W^2(y(n)) \frac{\exp(-\frac{y^2}{2\rho^2Q})}{\sqrt{2\pi\rho^2Q}} dy \end{aligned} \quad (3.18)$$

である。

$f_G(d(n))$ と $f_W(y(n))$ の積の平均は

$$\begin{aligned} \langle f_G(d(n)) f_W(y(n)) \rangle &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_G(d(n)) f_W(y(n)) P(d(n), y(n)) dd dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_G(d(n)) f_W(y(n)) \frac{\exp\left(-\frac{1}{2}\begin{pmatrix} d(n) \\ y(n) \end{pmatrix}^\top \Sigma^{-1} \begin{pmatrix} d(n) \\ y(n) \end{pmatrix}\right)}{\sqrt{(2\pi)^2 |\Sigma|}} dd dy \end{aligned} \quad (3.19)$$

である。

3.2.2 二乗平均偏差

式(3.14),(3.15)から二乗平均偏差は

$$\begin{aligned} \text{MSD} &= \| \mathbf{g} - \mathbf{w}(n) \|_2^2 \\ &= \| \mathbf{g} \|_2^2 - 2\mathbf{g}^\top \mathbf{w}(n) + \| \mathbf{w}(n) \|_2^2 \\ &= N(\sigma_g^2 - 2r + Q) \end{aligned} \quad (3.20)$$

のように巨視的変数 r, Q の関数である。式(3.20)より MSD はタップ長 N に比例する。そこで、本論文では MSD を N で正規化した値を正規化 MSD とする。

3.2.3 巨視的変数 r の微分方程式

r の微分方程式を導出する。式(3.3), (3.5), (3.9)よりフィルタ係数の更新式は

$$w_i(n+1) = w_i(n) + \mu e(n) u(n-i+1) \quad (3.21)$$

である。式(3.21)の両辺に未知システム G の要素 g_i をかけ、 i について和をとると

$$\sum_{i=1}^N g_i w_i(n+1) = \sum_{i=1}^N g_i w_i(n) + \mu e(n) \sum_{i=1}^N g_i u(n-i+1) \quad (3.22)$$

である。式(3.22)を式(3.6), (3.14)を用いて整理すると

$$Nr(n+1) = Nr(n) + \mu e(n) d(n) \quad (3.23)$$

と求まる。 $t \triangleq n/N$ を時間尺度とし、微小時間 dt の間に Ndt 回更新を行うと以下のように Ndt 個の式が書ける。

$$Nr(n+1) = Nr(n) + \mu e(n) d(n) \quad (3.24)$$

$$Nr(n+2) = Nr(n+1) + \mu e(n+1) d(n+1) \quad (3.25)$$

$$\vdots \quad \vdots \quad \vdots$$

$$Nr(n+Ndt) = Nr(n+Ndt-1) + \mu e(n+Ndt-1) d(n+Ndt-1) \quad (3.26)$$

式(3.24)–(3.26)の総和をとると

$$Nr(n+Ndt) = Nr(n) + \sum_{i=1}^{Ndt} \mu e(n+i-1) d(n+i-1) \quad (3.27)$$

である。式(3.27)の右辺第二項は Ndt 個の確率変数の総和であるため自己平均性を用いると

$$Nr(n + Ndt) = Nr(n) + Ndt\mu\langle e(m)d(m) \rangle \quad (3.28)$$

となる。 r が dr だけ変化し、 Ndt 回の更新が行われているので式(3.28)は

$$Ndr(t) = Ndt\mu\langle e(m)d(m) \rangle \quad (3.29)$$

となる。式(3.8), (3.29)より、 r の微分方程式は

$$\begin{aligned} \frac{dr(t)}{dt} &= \mu\langle e(m)d(m) \rangle \\ &= \mu\langle (f_G(d(m)) - f_W(y(m)) + \xi(m))d(m) \rangle \\ &= \mu\langle d(m)f_G(d(m)) - d(m)f_W(y(m)) \rangle \\ &= \mu(\langle d(m)f_G(d(m)) \rangle - \langle d(m)f_W(y(m)) \rangle) \end{aligned} \quad (3.30)$$

となる。

3.2.4 巨視的変数 Q の微分方程式

Q の微分方程式を導出する。式(3.21)の両辺を二乗して i について和をとると

$$\begin{aligned} \sum_{i=1}^N w_i^2(n+1) &= \sum_{i=1}^N w_i^2(n) \\ &\quad + 2\mu e(n) \sum_{i=1}^N w_i(n)u(n-i+1) \\ &\quad + \mu^2 e^2(n) \sum_{i=1}^N u^2(n-i+1) \end{aligned} \quad (3.31)$$

となる。式(3.31)を式(3.7), 式(3.15)を用いて整理すると

$$\begin{aligned} NQ(n+1) &= NQ(n) + 2\mu e(n)y(n) \\ &\quad + \mu^2 e^2(n) \sum_{i=1}^N u^2(n-i+1) \end{aligned} \quad (3.32)$$

である。 r と同様に微小時間 dt の間に Ndt 回更新を行ったとすると Ndt 個の式が書ける。それらを総和をとり自己平均性を用いると

$$NQ(n + Ndt) = NQ(n) + 2Ndt\mu\langle e(m)y(m) \rangle + Ndt\mu^2\rho^2\langle e^2(m) \rangle \quad (3.33)$$

となる. Q が dQ 変化し, Ndt 回の更新が行われているので式 (3.33) は

$$NdQ(t) = 2Ndt\mu \langle e(m)y(m) \rangle + Ndt\mu^2\rho^2 \langle e^2(m) \rangle \quad (3.34)$$

となる. 式を整理すると Q の微分方程式は

$$\begin{aligned} \frac{dQ(t)}{dt} &= 2\mu \langle e(m)y(m) \rangle + \mu^2\rho^2 \langle e^2(m) \rangle \\ &= 2\mu \langle (f_G(d(m)) - f_W(y(m)) + \xi(m))y(m) \rangle \\ &\quad + \mu^2\rho^2 \langle (f_G(d(m)) - f_W(y(m)) + \xi(m))^2 \rangle \\ &= \mu^2\rho^2 (\langle f_G^2(d(m)) \rangle - 2\langle f_G(d(m))f_W(y(m)) \rangle + \langle f_W^2(y(m)) \rangle + \sigma_\xi^2) \\ &\quad + 2\mu(\langle y(m)f_G(d(m)) \rangle - \langle y(m)f_W(y(m)) \rangle) \end{aligned} \quad (3.35)$$

である.

第4章 未知システム出力と適応フィルタ出力に飽和型非線形性を有するシステムの解析

4.1 飽和型非線形性

本論文で扱う飽和型非線形性の関数 $f_G(\cdot)$, $f_W(\cdot)$ はそれぞれ

$$f_G(x) = \begin{cases} S_1 & (x > S_1) \\ -S_1 & (x < -S_1) \\ 0 & (\text{otherwise}) \end{cases} \quad (4.1)$$

$$f_W(x) = \begin{cases} S_2 & (x > S_2) \\ -S_2 & (x < -S_2) \\ 0 & (\text{otherwise}) \end{cases} \quad (4.2)$$

である。ここで S_1 , S_2 は非負の実数である。関数 $f_G(\cdot)$, $f_W(\cdot)$ をそれぞれ図 4.1, 4.2 に示す。

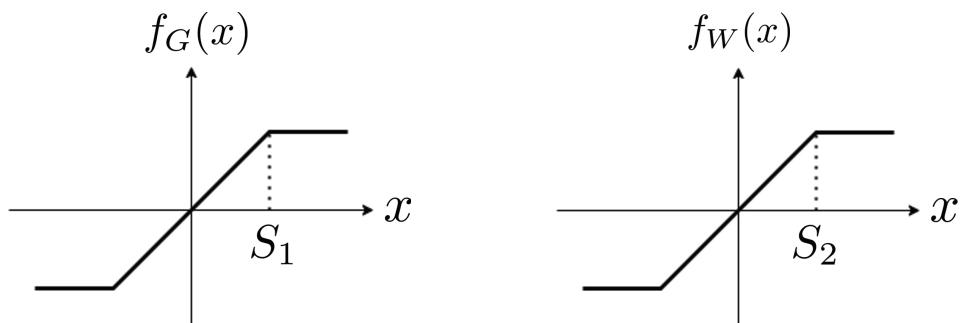


図 4.1: 未知システムの非線形関数 $f_G(\cdot)$

図 4.2: 適応フィルタの非線形関数 $f_W(\cdot)$

4.2 理論

4.2.1 二乗平均誤差

式(3.10)の第一項、第二項について考える。式(3.17)より $\langle f_G^2(d(n)) \rangle$ は

$$\langle f_G^2(d(n)) \rangle = S_1^2 - S_1 \sqrt{\frac{2\rho^2\sigma_g^2}{\pi}} \exp\left(-\frac{S_1^2}{2\rho^2\sigma_g^2}\right) + (\rho^2\sigma_g^2 - S_1^2) \operatorname{erf}\left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}\right) \quad (4.3)$$

となる。計算の詳細は付録Aに記述する。ここで、 $\operatorname{erf}(\cdot)$ は

$$\operatorname{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt \quad (4.4)$$

で定義される誤差関数である。式(3.18)より $\langle f_W^2(y(n)) \rangle$ は

$$\langle f_W^2(y(n)) \rangle = S_2^2 - S_2 \sqrt{\frac{2\rho^2Q}{\pi}} \exp\left(-\frac{S_2^2}{2\rho^2Q}\right) + (\rho^2Q - S_2^2) \operatorname{erf}\left(\frac{S_2}{\sqrt{2\rho^2Q}}\right) \quad (4.5)$$

となる。計算の詳細は付録Aに記述する。これらの値より二乗平均誤差は

$$\begin{aligned} \langle e^2(n) \rangle &= S_1^2 - S_1 \sqrt{\frac{2\rho^2\sigma_g^2}{\pi}} \exp\left(-\frac{S_1^2}{2\rho^2\sigma_g^2}\right) + (\rho^2\sigma_g^2 - S_1^2) \operatorname{erf}\left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}\right) \\ &\quad + S_2^2 - S_2 \sqrt{\frac{2\rho^2Q}{\pi}} \exp\left(-\frac{S_2^2}{2\rho^2Q}\right) + (\rho^2Q - S_2^2) \operatorname{erf}\left(\frac{S_2}{\sqrt{2\rho^2Q}}\right) \\ &\quad - 2 \langle f_G(d(n)) f_W(y(n)) \rangle + \sigma_\xi^2 \end{aligned} \quad (4.6)$$

となる。

4.2.2 二重積分から一重積分への変換

式(3.10)の第三項について考える。第三項のサンプル平均は式(3.19)に示しているが、この二重積分は解析的に実行することが困難である。よって二重積分を数値的に実行する必要があるが、この場合、式(3.16)の共分散行列が特異に近くなり数値計算が不安定になってしまう。そのため積分範囲を単純化し、数値計算の安定化と高速化を図るため、二重積分から一重積分への変換を行う。 d と y の二次元ガウス分布のイメージを図4.3に示す。

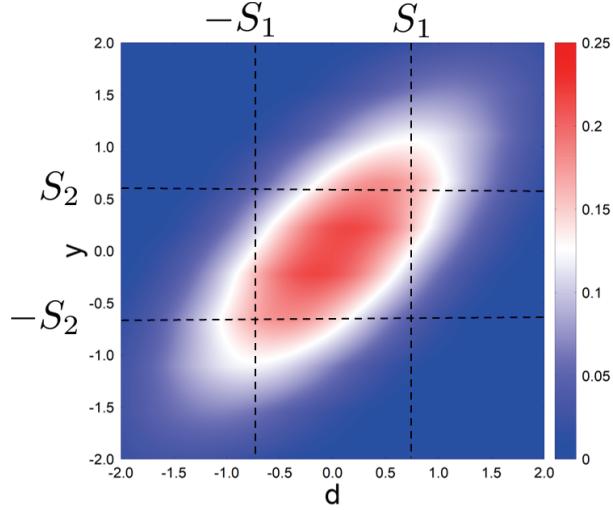


図 4.3: 二次元ガウス分布

図 4.3 に示すように、式 (3.19) の二重積分は九つの積分区間に分割できる。すなわち、

$$\begin{aligned}
 \langle f_G(d(n))f_W(y(n)) \rangle &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_G(d(n))f_W(y(n))P(d(n), y(n)) dd dy \\
 &= \int_{-\infty}^{-S_1} \int_{-\infty}^{-S_2} (-S_1)(-S_2)P(d(n), y(n)) dd dy \\
 &\quad + \int_{-\infty}^{-S_1} \int_{-S_2}^{S_2} (-S_1)yP(d(n), y(n)) dd dy \\
 &\quad + \int_{-\infty}^{-S_1} \int_{S_2}^{\infty} (-S_1)(S_2)P(d(n), y(n)) dd dy \\
 &\quad + \int_{-S_1}^{S_1} \int_{-\infty}^{-S_2} d(-S_2)P(d(n), y(n)) dd dy \\
 &\quad + \int_{-S_1}^{S_1} \int_{-S_2}^{S_2} dyP(d(n), y(n)) dd dy \\
 &\quad + \int_{-S_1}^{S_1} \int_{S_2}^{\infty} dS_2P(d(n), y(n)) dd dy \\
 &\quad + \int_{S_1}^{\infty} \int_{-\infty}^{-S_2} S_1(-S_2)P(d(n), y(n)) dd dy \\
 &\quad + \int_{S_1}^{\infty} \int_{-S_2}^{S_2} S_1yP(d(n), y(n)) dd dy \\
 &\quad + \int_{S_1}^{\infty} \int_{S_2}^{\infty} S_1S_2P(d(n), y(n)) dd dy
 \end{aligned} \tag{4.7}$$

と書ける。ここで $P(d(n), y(n))$ は

$$P(d(n), y(n)) = \frac{\exp\left(-\frac{1}{2}\begin{pmatrix} d(n) \\ y(n) \end{pmatrix}^T \Sigma^{-1} \begin{pmatrix} d(n) \\ y(n) \end{pmatrix}\right)}{\sqrt{(2\pi)^2 |\Sigma|}} \quad (4.8)$$

であり、 Σ は式 (3.16) に示す分散共分散行列である。

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} = \Sigma^{-1} = \left(\rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \right)^{-1} \quad (4.9)$$

として、式 (4.8) の二次形式の部分を展開すると

$$\begin{aligned} \begin{pmatrix} d & y \end{pmatrix} \Sigma^{-1} \begin{pmatrix} d \\ y \end{pmatrix} &= ad^2 + 2bdy + cy^2 \\ &= a \left(d + \frac{by}{a} \right)^2 + \left(c - \frac{b^2}{a} \right) y^2 \end{aligned} \quad (4.10)$$

となる。 $d' = d + \frac{by}{a}$ とすると式 (4.7) の第一項は

$$\begin{aligned} &\int_{-\infty}^{-S_2} \int_{-\infty}^{-S_1} \frac{(-S_1)(-S_2)}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\left(a\left(d + \frac{by}{a}\right)^2 + \left(c - \frac{b^2}{a}\right)y^2\right)\right) dd dy \\ &= \int_{-\infty}^{-S_2} \frac{S_1 S_2}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{\left(c - \frac{b^2}{a}\right)y^2}{2}\right) \int_{-\infty}^{-S_1 + \frac{by}{a}} \exp\left(-\frac{ad'^2}{2}\right) dd' dy \end{aligned} \quad (4.11)$$

となる。式 (4.11) の d' に関する積分は誤差関数を用いて表すことができ、第一項は

$$\begin{aligned} &\int_{-\infty}^{-S_1} \int_{-\infty}^{-S_2} (-S_1)(-S_2) P(d(n), y(n)) dd dy \\ &= \int_{-\infty}^{-S_2} \frac{S_1 S_2}{2\pi\sqrt{\Sigma}} \sqrt{\frac{\pi}{2a}} \exp\left(-\frac{\left(c - \frac{b^2}{a}\right)y^2}{2}\right) \left(1 + \operatorname{erf}\left(\frac{\sqrt{a}(-S_1 + \frac{by}{a})}{\sqrt{2}}\right)\right) dy \end{aligned} \quad (4.12)$$

となり、一重積分の形で記述できる。第二項、第三項及び第七項から第九項までは同様に一重積分の形で記述することができる。次に式 (4.7) の第四項について考える。 $d' = d + \frac{by}{a}$ とすると第四項は

$$\begin{aligned}
& \int_{-\infty}^{-S_2} \int_{-S_1}^{S_1} d(-S_2) \frac{1}{2\pi\sqrt{|\sum|}} \exp\left(-\frac{1}{2}\left(a\left(d+\frac{by}{a}\right)^2 + \left(c-\frac{b^2}{a}\right)y^2\right)\right) dd dy \\
&= \int_{-\infty}^{-S_2} \frac{S_2}{2\pi\sqrt{|\sum|}} \exp\left(-\frac{(c-\frac{b^2}{a})y^2}{2}\right) \int_{-S_1+\frac{by}{a}}^{S_1+\frac{by}{a}} \left(d' - \frac{by}{a}\right) \exp\left(-\frac{ad'^2}{2}\right) dd' dy
\end{aligned} \tag{4.13}$$

となる。ここで式(4.13)において d' に関する積分を考えると

$$\begin{aligned}
& \int_{-S_1+\frac{by}{a}}^{S_1+\frac{by}{a}} \left(d' - \frac{by}{a}\right) \exp\left(-\frac{ad'^2}{2}\right) dd' \\
&= \int_{-S_1+\frac{by}{a}}^{S_1+\frac{by}{a}} d' \exp\left(-\frac{ad'^2}{2}\right) dd' - \frac{by}{a} \int_{-S_1+\frac{by}{a}}^{S_1+\frac{by}{a}} \exp\left(-\frac{ad'^2}{2}\right) dd'
\end{aligned} \tag{4.14}$$

式(4.14)の第一項は、 $d'' = d'\sqrt{a}$ とすると

$$\begin{aligned}
\int_{-S_1+\frac{by}{a}}^{S_1+\frac{by}{a}} d' \exp\left(-\frac{ad'^2}{2}\right) dd' &= \int_{(-S_1+\frac{by}{a})\sqrt{a}}^{(S_1+\frac{by}{a})\sqrt{a}} \left(\frac{d''}{\sqrt{a}}\right) \exp\left(-\frac{d''^2}{2}\right) \left(\frac{1}{\sqrt{a}}\right) dd'' \\
&= \frac{1}{a} \int_{(-S_1+\frac{by}{a})\sqrt{a}}^{(S_1+\frac{by}{a})\sqrt{a}} d'' \exp\left(-\frac{d''^2}{2}\right) dd'' \\
&= -\frac{1}{a} \left(\exp\left(-\frac{d''^2}{2}\right)\right)_{(-S_1+\frac{by}{a})\sqrt{a}}^{(S_1+\frac{by}{a})\sqrt{a}} \\
&= -\frac{1}{a} \left(\exp\left(-\frac{a(S_1+\frac{by}{a})^2}{2}\right) - \exp\left(-\frac{a(-S_1+\frac{by}{a})^2}{2}\right)\right)
\end{aligned} \tag{4.15}$$

となる。式(4.14)の第二項については、式(4.12)と同様に誤差関数を用いて表すことができ、

$$\begin{aligned}
& \frac{by}{a} \int_{-S_1+\frac{by}{a}}^{S_1+\frac{by}{a}} \exp\left(-\frac{ad'^2}{2}\right) dd' \\
&= \frac{by}{a} \sqrt{\frac{\pi}{2a}} \left(\operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(S_1+\frac{by}{a}\right)\right) - \operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(-S_1+\frac{by}{a}\right)\right)\right)
\end{aligned} \tag{4.16}$$

となる。式(4.14)に式(4.15), (4.16)を代入したものを式(4.13)に代入すると、式(4.7)の第四項は

$$\begin{aligned}
& \int_{-S_1}^{S_1} \int_{-\infty}^{-S_2} d(-S_2) P(d(n), y(n)) dd dy \\
&= \int_{-\infty}^{-S_2} \frac{-S_2}{2\pi\rho^2 \sqrt{(\sigma_g^2 Q - r^2)}} \exp\left(-\frac{\left(c - \frac{b^2}{a}\right)y^2}{2}\right) \\
&\quad \left(-\frac{1}{a} \left(\exp\left(-\frac{a(S_1 + \frac{by}{a})^2}{2}\right) - \exp\left(-\frac{a(-S_1 + \frac{by}{a})^2}{2}\right) \right) \right. \\
&\quad \left. - \frac{by}{a} \sqrt{\frac{\pi}{2a}} \left(\operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(S_1 + \frac{by}{a}\right)\right) - \operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(-S_1 + \frac{by}{a}\right)\right) \right) \right) dy \quad (4.17)
\end{aligned}$$

となる. 式 (4.12), (4.17) より式 (4.7) は

$$\begin{aligned}
& \langle f_G(d(n)) f_W(y(n)) \rangle \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_G(d(n)) f_W(y(n)) P(d(n), y(n)) dd dy \\
&= \int_{-\infty}^{-S_2} \frac{S_1 S_2}{2\pi\rho^2 \sqrt{\sigma_g^2 Q - r^2}} \sqrt{\frac{\pi}{2a}} \exp\left(-\frac{(c - \frac{b^2}{a})y^2}{2}\right) \left(1 + \operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(-S_1 + \frac{by}{a}\right)\right) \right) dy \\
&+ \int_{-S_2}^{S_2} \frac{-S_1 y}{2\pi\rho^2 \sqrt{\sigma_g^2 Q - r^2}} \sqrt{\frac{\pi}{2a}} \exp\left(-\frac{(c - \frac{b^2}{a})y^2}{2}\right) \left(1 + \operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(x_1 + \frac{by}{a}\right)\right) \right) dy \\
&+ \int_{S_2}^{\infty} \frac{-S_1 S_2}{2\pi\rho^2 \sqrt{\sigma_g^2 Q - r^2}} \sqrt{\frac{\pi}{2a}} \exp\left(-\frac{(c - \frac{b^2}{a})y^2}{2}\right) \left(1 + \operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(x_1 + \frac{by}{a}\right)\right) \right) dy \\
&+ \int_{-\infty}^{-S_2} \frac{-S_2}{2\pi\rho^2 \sqrt{(\sigma_g^2 Q - r^2)}} \exp\left(-\frac{\left(c - \frac{b^2}{a}\right)y^2}{2}\right) \\
&\cdot \left(-\frac{1}{a} \left(\exp\left(-\frac{a(S_1 + \frac{by}{a})^2}{2}\right) - \exp\left(-\frac{a(-S_1 + \frac{by}{a})^2}{2}\right) \right) \right. \\
&\left. - \frac{by}{a} \sqrt{\frac{\pi}{2a}} \cdot \left(\operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(S_1 + \frac{by}{a}\right)\right) - \operatorname{erf}\left(\sqrt{\frac{a}{2}}\left(-S_1 + \frac{by}{a}\right)\right) \right) \right) dy \\
&+ \int_{-S_2}^{S_2} \frac{y}{2\pi\rho^2 \sqrt{(\sigma_g^2 Q - r^2)}} \exp\left(-\frac{\left(c - \frac{b^2}{a}\right)y^2}{2}\right) \\
&\cdot \left(-\frac{1}{a} \left(\exp\left(-\frac{a(S_1 + \frac{by}{a})^2}{2}\right) - \exp\left(-\frac{a(-S_1 + \frac{by}{a})^2}{2}\right) \right) \right)
\end{aligned}$$

$$\begin{aligned}
& -\frac{by}{a}\sqrt{\frac{\pi}{2a}} \cdot \left(\operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(S_1 + \frac{by}{a} \right) \right) - \operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(-S_1 + \frac{by}{a} \right) \right) \right) dy \\
& + \int_{S_2}^{\infty} \frac{S_2}{2\pi\rho^2\sqrt{(\sigma_g^2 Q - r^2)}} \exp \left(-\frac{\left(c - \frac{b^2}{a} \right) y^2}{2} \right) \\
& \cdot \left(-\frac{1}{a} \left(\exp \left(-\frac{a(S_1 + \frac{by}{a})^2}{2} \right) - \exp \left(-\frac{a(-S_1 + \frac{by}{a})^2}{2} \right) \right) \right. \\
& \left. - \frac{by}{a}\sqrt{\frac{\pi}{2a}} \cdot \left(\operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(S_1 + \frac{by}{a} \right) \right) - \operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(-S_1 + \frac{by}{a} \right) \right) \right) \right) dy \\
& + \int_{-\infty}^{-S_2} \frac{-S_1 S_2}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \sqrt{\frac{\pi}{2a}} \exp \left(-\frac{(c - \frac{b^2}{a})y^2}{2} \right) \cdot \left(1 - \operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(x_2 + \frac{by}{a} \right) \right) \right) dy \\
& + \int_{-S_2}^{S_2} \frac{S_1 y}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \sqrt{\frac{\pi}{2a}} \exp \left(-\frac{(c - \frac{b^2}{a})y^2}{2} \right) \cdot \left(1 - \operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(x_2 + \frac{by}{a} \right) \right) \right) dy \\
& + \int_{S_2}^{\infty} \frac{S_1 S_2}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \sqrt{\frac{\pi}{2a}} \exp \left(-\frac{(c - \frac{b^2}{a})y^2}{2} \right) \cdot \left(1 - \operatorname{erf} \left(\sqrt{\frac{a}{2}} \left(x_2 + \frac{by}{a} \right) \right) \right) dy
\end{aligned} \tag{4.18}$$

となり、一重積分の形で記述することができる。

4.2.3 巨視的変数 r の微分方程式

巨視的変数 r の微分方程式を導出する。式(3.30)の第一項について考える。式(3.16)より、 $d(n)$ と $f_G(d(n))$ の積の平均は

$$\begin{aligned}
\langle d(n)f_G(d(n)) \rangle &= \int_{-\infty}^{\infty} d(n)f_G(d(n))P(d(n)) dd \\
&= \int_{-\infty}^{\infty} d(n)f_G(d(n)) \frac{\exp(-\frac{d^2}{2\rho^2\sigma_g^2})}{\sqrt{2\pi\rho^2\sigma_g^2}} dd \\
&= \rho^2\sigma_g^2 \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}} \right)
\end{aligned} \tag{4.19}$$

$d(n)$ と $f_W(y(n))$ の積の平均は

$$\begin{aligned}\langle(d(n))f_W(y(n))\rangle &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d(n)f_W(y(n))P(d(n),y(n)) \, dddy \\ &= \rho^2 \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right)\end{aligned}\quad (4.20)$$

である。計算の詳細は付録 A に記述する。式 (3.30) に式 (4.19), (4.20) を代入すると巨視的変数 r の微分方程式は

$$\frac{dr}{dt} = \mu\rho^2 \left(\sigma_g^2 \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}} \right) - r \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) \right) \quad (4.21)$$

となる。

4.2.4 巨視的変数 Q の微分方程式

巨視的変数 Q の微分方程式を導出する。式 (3.35) のそれぞれの項について考える。
 $\langle y(n)f_G(d(n)) \rangle$ は

$$\langle y(n)f_G(d(n)) \rangle = \rho^2 \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}} \right) \quad (4.22)$$

$\langle y(n)f_W(y(n)) \rangle$ は

$$\langle y(n)f_w(y(n)) \rangle = \rho^2 Q \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) \quad (4.23)$$

となる。計算の詳細は付録 A に記述する。式 (3.35) に式 (3.17), (3.18), (3.19), (4.22), (4.23) を代入すると巨視的変数 Q の微分方程式は

$$\begin{aligned}\frac{dQ}{dt} &= \mu\rho^2 \left((2r + \mu\rho^2 \sigma_g^2 - 2Q) \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}} \right) \right. \\ &\quad \left. + (\mu\rho^2 Q - \mu S_2^2 - 2Q) \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) \right) \\ &\quad - \sqrt{\frac{2}{\pi}} \mu^2 \rho^3 \left(S_1 \sigma_g \exp \left(-\frac{S_1^2}{2\rho^2 \sigma_g^2} \right) + S_2 \sqrt{Q} \exp \left(-\frac{S_2^2}{2\rho^2 Q} \right) \right) \\ &\quad + \mu^2 \rho^2 (S_1^2 + S_2^2 + \sigma_g^2 - 2 \langle f_G(d(n))f_W(y(n)) \rangle)\end{aligned}\quad (4.24)$$

となる。

第5章 結果と議論

5.1 二乗平均誤差

5.1.1 二重積分と一重積分の違い

時間と MSE の関係を図 5.1, 5.2 に示す。これは学習曲線と呼ばれる。図 5.1 における MSE は式 (4.7) の二重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求め、図 5.2 における MSE は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。飽和値は $S_1=1$, $S_2=1$, 背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2=1$ で、ステップサイズは $\mu=0.1$ とした。図 5.1 から、二重積分を数値的に解いた場合は時刻 100 度程までしか計算することができず、計算が不安定になり理論に基づかれていたりしている部分が確認できる。それに対して、図 5.2 の一重積分を数値的に解いた場合では時刻 300 度程まで理論に基づかなく、図 5.1 と比較して数値計算が安定化していることが分かる。また、図 5.1 の二重積分を数値的に解いた場合の計算時間は約 120 秒であったのに対し、図 5.2 の一重積分を数値的に解いた場合の計算時間は約 3 秒であり、計算時間の観点においても大きく改善することができた。

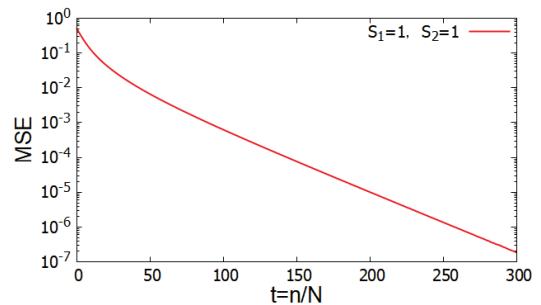
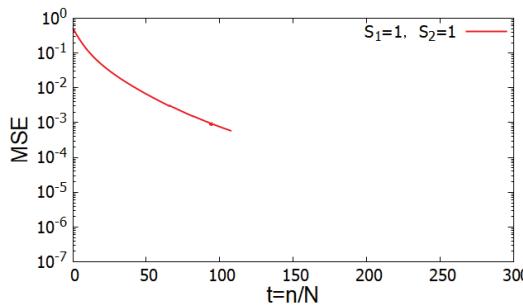


図 5.1: 二重積分を数値的に実行した場合 図 5.2: 一重積分を数値的に実行した場合

5.1.2 学習曲線

MSE の学習曲線を図 5.3(a)–5.3(d) に示す。点は計算機実験 ($N=200$)、線は理論値 ($N \rightarrow \infty$) を示している。計算機実験の各シンボルは 1000 回の試行の平均値であり、理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。飽和値は $S_1=1$ で固定し S_2 を 0.3, 0.5, 1, 2 に変化させた場合について調べた。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2 = 1$ であり、4通りのステップサイズについて計算した。計算機実験と理論はよく一致しており、理論が二乗平均誤差のふるまいをよく説明していることがわかる。図 5.3(a)–5.3(d) より、 $S_1=S_2$ の場合 MSE は 0 に漸近していることが分かる。ステップサイズ μ が大きくなるほど、MSE が 0 に漸近するのが速くなる一方で、図 5.3(c), 5.3(d) では理論に小さなぶれが見られた。このことから、ステップサイズが大きくなり MSE が 0 に収束するのが速いほど、数値的な計算が不安定になることが分かる。

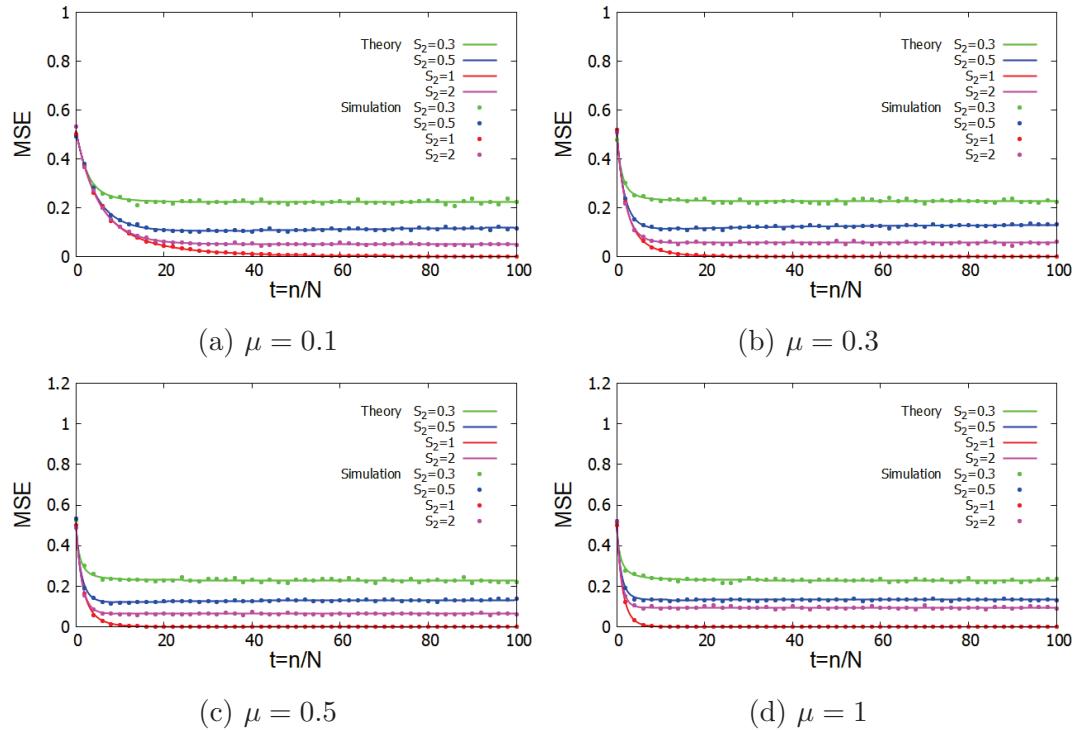


図 5.3: MSE の学習曲線 ($S_1=1$)

MSE の学習曲線を図 5.4(a)–5.4(d) に示す。点は計算機実験 ($N=200$)、線は理論値 ($N \rightarrow \infty$) を示している。計算機実験の各シンボルは 1000 回の試行の平均値であり、理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。飽和値は $S_2=1$ で固定し S_1 を 0.3, 0.5, 1, 2 に変化させた場合について調べた。背

景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2=1$ であり, 4通りのステップサイズについて計算した. 計算機実験と理論はよく一致しており, 理論が二乗平均誤差のふるまいをよく説明していることがわかる. 図 5.4(a)–5.4(d) より, $S_1=S_2$ の場合 MSE は 0 に漸近していることが分かる. ステップサイズ μ が大きくなるほど, MSE が 0 に漸近するのが速くなる一方で, 図 5.4(c), 5.4(d) では理論に小さなぶれが見られた. このことから, ステップサイズが大きくなり MSE が 0 に収束するのが速いほど, 数値的な計算が不安定になることが分かる.

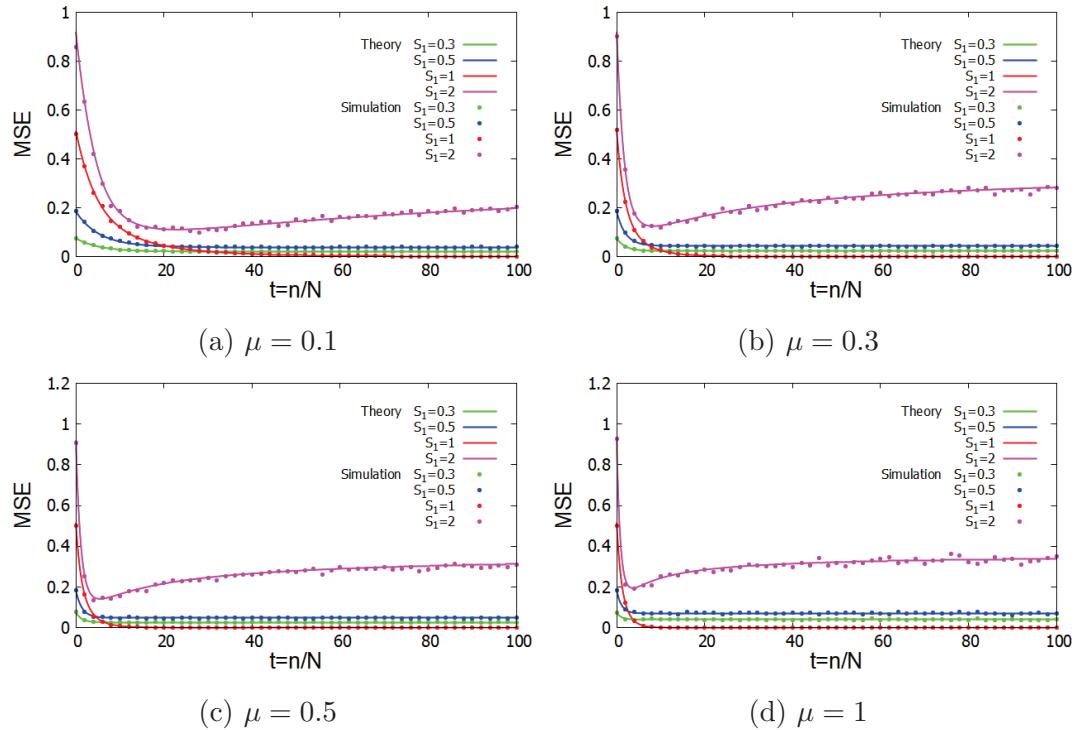


図 5.4: MSE の学習曲線 ($S_2=1$)

5.1.3 二つの飽和値と準定常二乗平均誤差の関係

二つの飽和値と準定常 MSE の関係を図 5.5 に示す. 計算機実験は $N=200$, 時刻 5000 で, 100 回の試行の平均値である. 理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた. ここでは準定常 MSE は時刻 300 での MSE を示し, S_1 を 0.3, 0.5, 1, 2 に固定した場合の結果である. 背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2=1$ の場合の結果であり, ステップサイズ $\mu=0.1$ について計算した. 図 5.5 から $S_1=S_2$ 付近で準定常 MSE が 0 に漸近し, 最小値をとっていることが分かる. しかし, $S_1=0.3, 0.5$ では MSE が十分に収束していないことから, 準定常 MSE が最小となる S_2/S_1 の値は

1から少しずれる結果となった。また、 S_1 と S_2 の差が大きくなるほど準定常MSEが大きくなつた。 $S_1=2$ では、理論とシミュレーションに誤差が見られることから、数値積分の精度が課題であり、パラメータの組み合わせによらない計算精度が課題である。

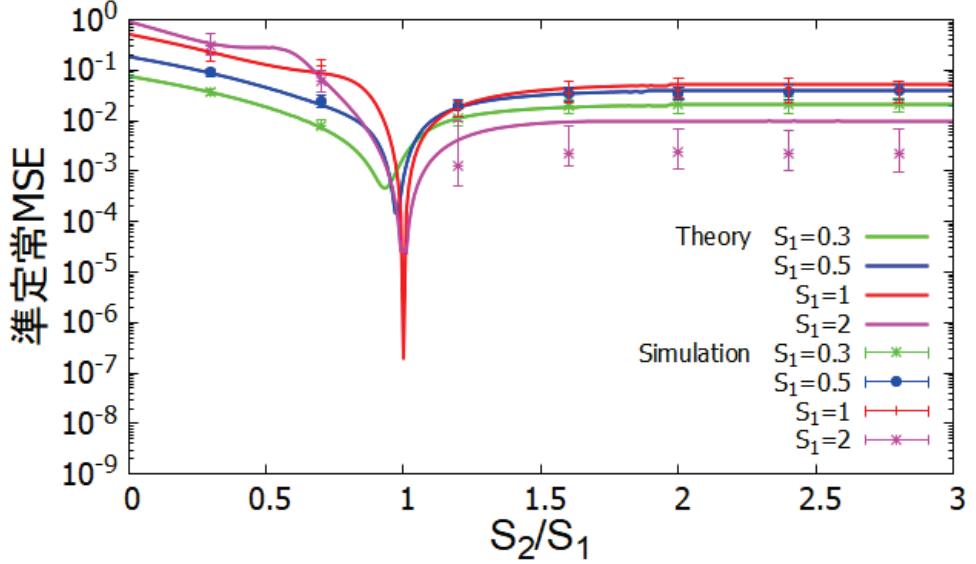


図 5.5: 二つの飽和値と準定常二乗平均誤差の関係 ($\sigma_\xi^2=0$)

二つの飽和値と準定常MSEの関係を図5.6, 5.7に示す。図5.6は背景雑音の分散が $\sigma_\xi^2 = 0.01$ 、図5.7は背景雑音の分散が $\sigma_\xi^2 = 0.1$ の場合の結果である。計算機実験は $N=200$ 、時刻5000で、100回の試行の平均値である。理論は式(4.18)の一重積分を用いて式(4.6),(4.21),(4.24)を数値的に解くことにより求めた。 $\rho^2=1$ 、ステップサイズ $\mu=0.1$ として計算した。ここでは準定常MSEは時刻300でのMSEを示し、 S_1 を0.3, 0.5, 1, 2に固定した場合の結果である。背景雑音の分散が $\sigma_\xi^2 = 0.01$ の場合、準定常MSEの最小値は約0.01であり、 $\sigma_\xi^2 = 0.1$ の場合、準定常MSEの最小値は約0.1であることが分かった。このことから、準定常MSEの最小値は背景雑音の大きさに依存していると考えられる。背景雑音の値を大きくしていくほど、 $S_2 > S_1$ において S_1 の値による準定常MSEの差が小さくなつた。また、 $S_1=2$ における理論とシミュレーションの誤差も小さくなることが分かった。

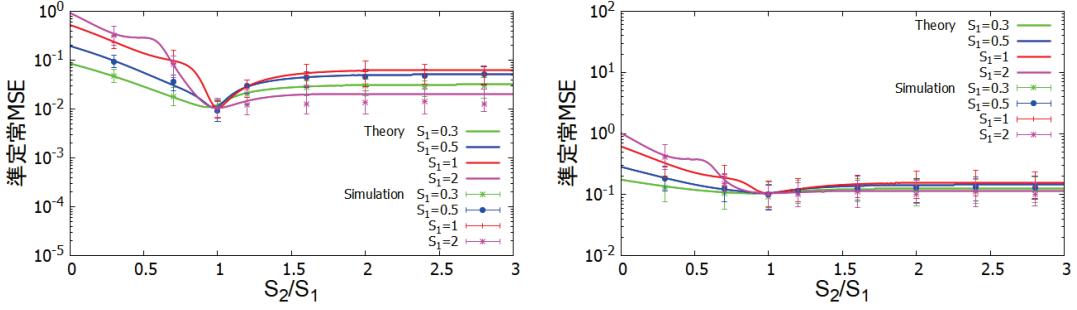


図 5.6: 二つの飽和値と準定常二乗平均誤差の関係 ($\sigma_\xi^2=0.01$)

図 5.7: 二つの飽和値と準定常二乗平均誤差の関係 ($\sigma_\xi^2=0.1$)

二つの飽和値と準定常 MSE の関係を図 5.8 に示す。計算機実験は $N=200$, 時刻 5000 で, 100 回の試行の平均値である。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。ここでは準定常 MSE は時刻 300 での MSE を示し, S_2 を 0.3, 0.5, 1, 2 に固定した場合の結果である。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2 = 1$ の場合の結果であり, ステップサイズ $\mu = 0.1$ について計算した。図 5.8 から, $S_1 = 0$, $S_1 = S_2$ で準定常 MSE が 0 に漸近し, 最小値をとっていることが分かる。しかし, $S_2 = 0.3, 0.5$ では MSE が十分に収束していないことから, 準定常 MSE が最小となる S_2/S_1 の値は 1 から少しずれる結果となった。また, S_1 と S_2 の差が大きくなるほど準定常 MSE が大きくなかった。 $S_2 = 2$ では, 理論とシミュレーションに誤差が見られることから, 数値積分の精度が課題であり, パラメータの組み合わせによらない計算精度が課題である。

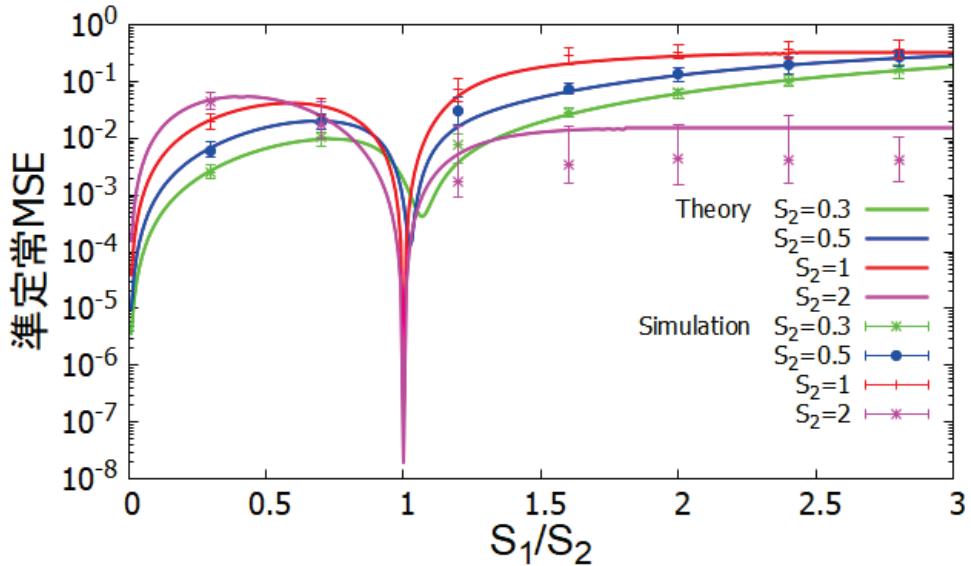


図 5.8: 二つの飽和値と準定常二乗平均誤差の関係 ($\sigma_\xi^2=0$)

二つの飽和値と準定常 MSE の関係を図 5.9, 5.10 に示す。図 5.9 は背景雑音の分散が $\sigma_\xi^2 = 0.01$, 図 5.10 は背景雑音の分散が $\sigma_\xi^2 = 0.1$ の場合の結果である。計算機実験は $N=200$, 時刻 5000 で, 100 回の試行の平均値である。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。 $\rho^2=1$, ステップサイズ $\mu=0.1$ として計算した。ここでは準定常 MSE は時刻 300 での MSE を示し, S_2 を 0.3, 0.5, 1, 2 に固定した場合の結果である。図 5.9, 5.10 から, $S_1 = 0$, $S_1=S_2$ で準定常 MSE が 0 に漸近し, 最小値をとっていることが分かる。背景雑音の分散が $\sigma_\xi^2 = 0.01$ の場合, 準定常 MSE の最小値は約 0.01 であり, $\sigma_\xi^2 = 0.1$ の場合, 準定常 MSE の最小値は約 0.1 であることが分かった。このことから, 準定常 MSE の最小値は背景雑音の大きさに依存していると考えられる。背景雑音の値を大きくしていくほど, $S_1 > S_2$ において S_2 の値による準定常 MSE の差が小さくなつた。また, $S_2=2$ における理論とシミュレーションの誤差も小さくなることが分かった。

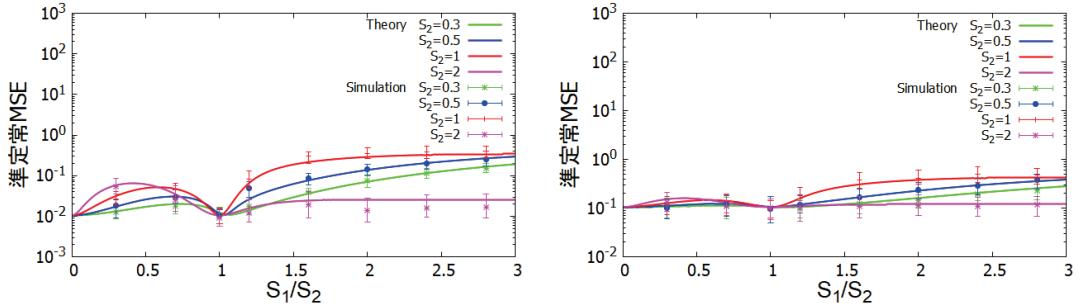


図 5.9: 二つの飽和値と準定常二乗平均誤差の関係 ($\sigma_\xi^2=0.01$) 図 5.10: 二つの飽和値と準定常二乗平均誤差の関係 ($\sigma_\xi^2=0.1$)

二つの飽和値と準定常 MSE の関係を表したヒートマップを図 5.11, 5.12 に示す。計算機実験は $N=200$, 時刻 5000 で, 100 回の試行の平均値であり, 理論の準定常 MSE は時刻 300 での MSE を示す。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2=1$ の場合の結果であり, ステップサイズ $\mu=0.1$ について計算した。図 5.11, 5.12 から, 計算機実験と理論はよく合っており, $S_1=S_2$ の部分で準定常 MSE が小さくなっていることが分かる。また S_1 の値が小さい部分でも準定常 MSE が小さくなつた。 S_1 が大きく S_2 が小さい部分では準定常 MSE が大きくなっていることがわかる。これは未知システムの出力に対して適応フィルタの出力が大きく制限されるためだと考えられる。

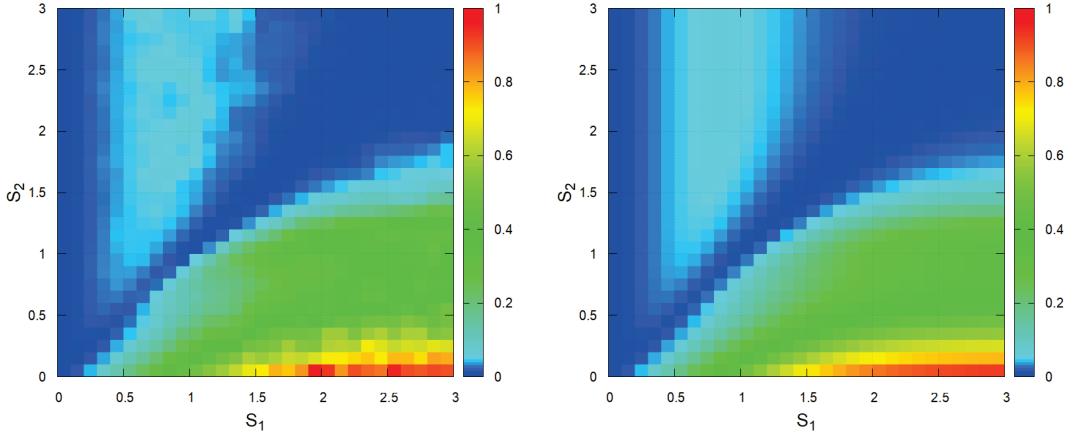


図 5.11: 計算機実験 ($\sigma_\xi^2=0$)

図 5.12: 理論 ($\sigma_\xi^2=0$)

二つの飽和値と準定常 MSE の関係を表したヒートマップを図 5.13, 5.14 に示す. 計算機実験は $N=200$, 時刻 5000 で, 100 回の試行の平均値であり, 理論の準定常 MSE は時刻 300 での MSE を示す. 理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた. 背景雑音の分散は $\sigma_\xi^2 = 0.1$, $\rho^2=1$ の場合の結果であり, ステップサイズ $\mu=0.1$ について計算した. 図 5.13, 5.14 から, 計算機実験と理論はよく合っている. 図 5.11, 5.12 と比較して青色の領域がなくなっていることから, 準定常 MSE の最小値が大きくなっていることがわかる. S_1 が大きく S_2 が小さい部分では準定常 MSE が大きくなっている. これは未知システムの出力に対して適応フィルタの出力が大きく制限されるためだと考えられる.

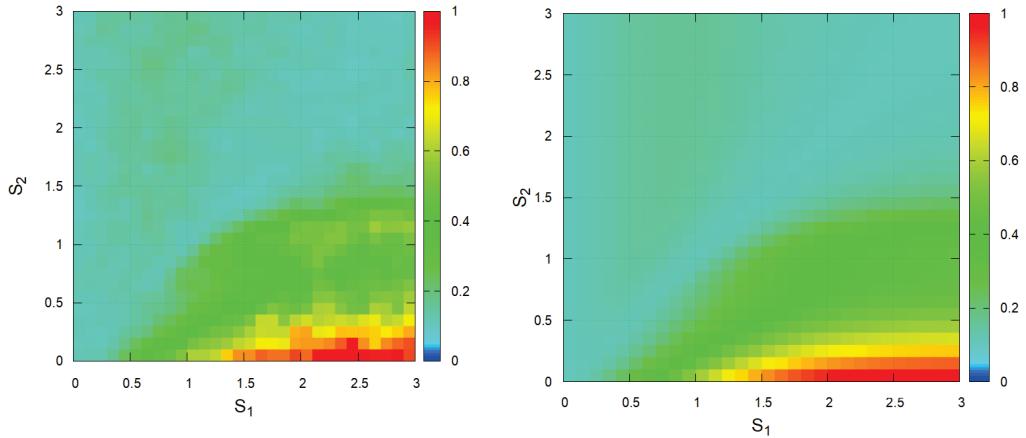


図 5.13: 計算機実験 ($\sigma_\xi^2=0.1$)

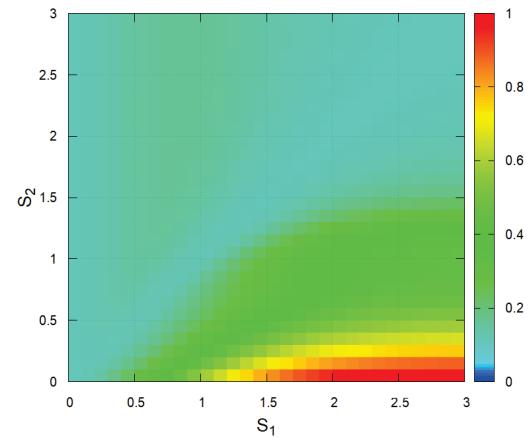


図 5.14: 理論 ($\sigma_\xi^2=0.1$)

5.2 二乗平均偏差

5.2.1 学習曲線

時間と MSD の関係を図 5.15(a)–5.15(d) に示す。これは学習曲線と呼ばれる。点は計算機実験 ($N=200$), 線は理論値 ($N \rightarrow \infty$) を示している。計算機実験の各シンボルは 1000 回の試行の平均値であり、理論は式 (4.18) の一重積分を用いて式 (3.20), (4.21), (4.24) を数値的に解くことにより求めた。飽和値の大きさは $S_1=1$ で固定し S_2 を 0.3, 0.5, 1, 2 に変化させた場合について調べた。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2=1$ であり、4 通りのステップサイズについて計算した。計算機実験と理論はよく一致しており、理論が二乗平均偏差のふるまいをよく説明していることがわかる。 $S_1=1$ で固定した場合、 $S_1 = S_2$ で MSD は 0 に漸近し、 $S_1 \leq S_2$ では MSD が収束すること、 $S_1 > S_2$ では MSD が単調増加することが分かった。ステップサイズ μ が大きくなるほど、MSD が 0 に漸近するのが速くなる一方で、図 5.15(c), 5.15(d) では MSD が 0 に限りなく近づくと数値計算が不安定になり、理論にぶれが生じた。

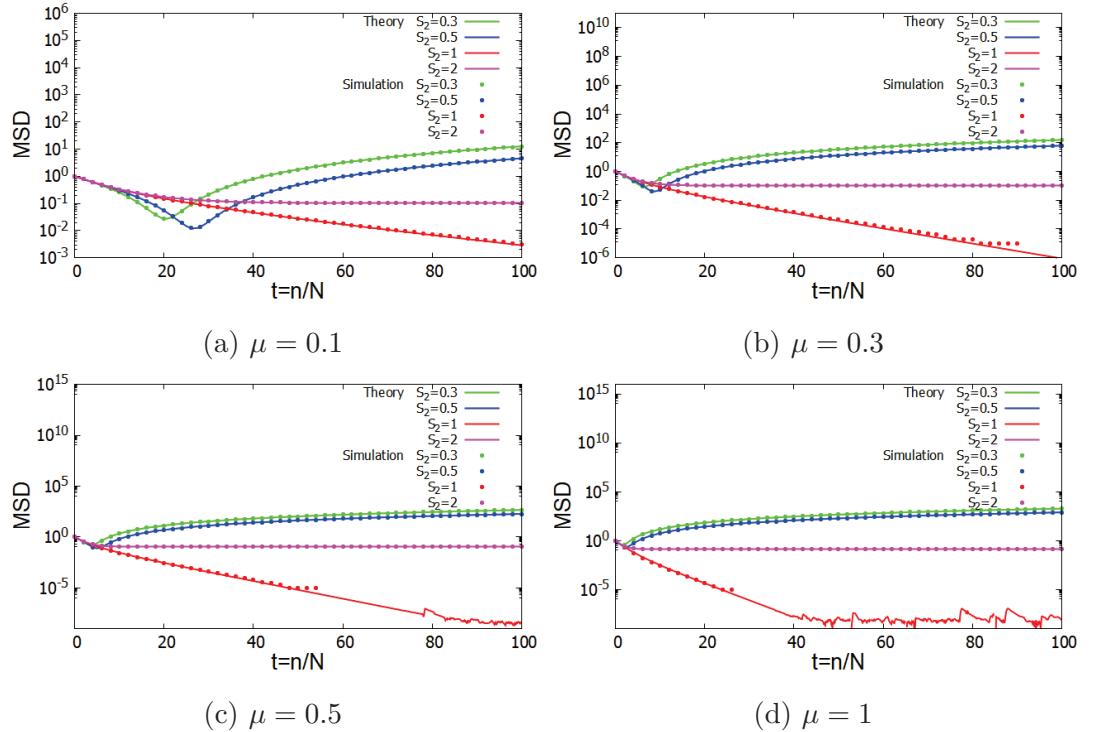


図 5.15: MSD の学習曲線

時間と MSD の関係を図 5.16(a)–5.16(d) に示す。これは学習曲線と呼ばれる。点は計算機実験 ($N=200$)、線は理論値 ($N \rightarrow \infty$) を示している。計算機実験の各シンボルは 1000 回の試行の平均値であり、理論は式 (4.18) の一重積分を用いて式 (3.20), (4.21), (4.24) を数値的に解くことにより求めた。飽和値の大きさは $S_1=1$ で固定し S_2 を 0.3, 0.5, 1, 2 に変化させた場合について調べた。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2 = 1$ であり、4通りのステップサイズについて計算した。計算機実験と理論はよく一致しており、理論が二乗平均偏差のふるまいをよく説明していることがわかる。 $S_2=1$ で固定した場合、 $S_1=1$ で固定した場合と同様に $S_1 = S_2$ で MSD は 0 に漸近し、 $S_1 \leq S_2$ では MSD が収束すること、 $S_1 > S_2$ では MSD が単調増加することが分かった。ステップサイズ μ が大きくなるほど、MSD が 0 に漸近するのが速くなる一方で、図 5.16(c), 5.16(d) では MSD が 0 に限りなく近づくと数値計算が不安定になり、理論にぶれが生じた。

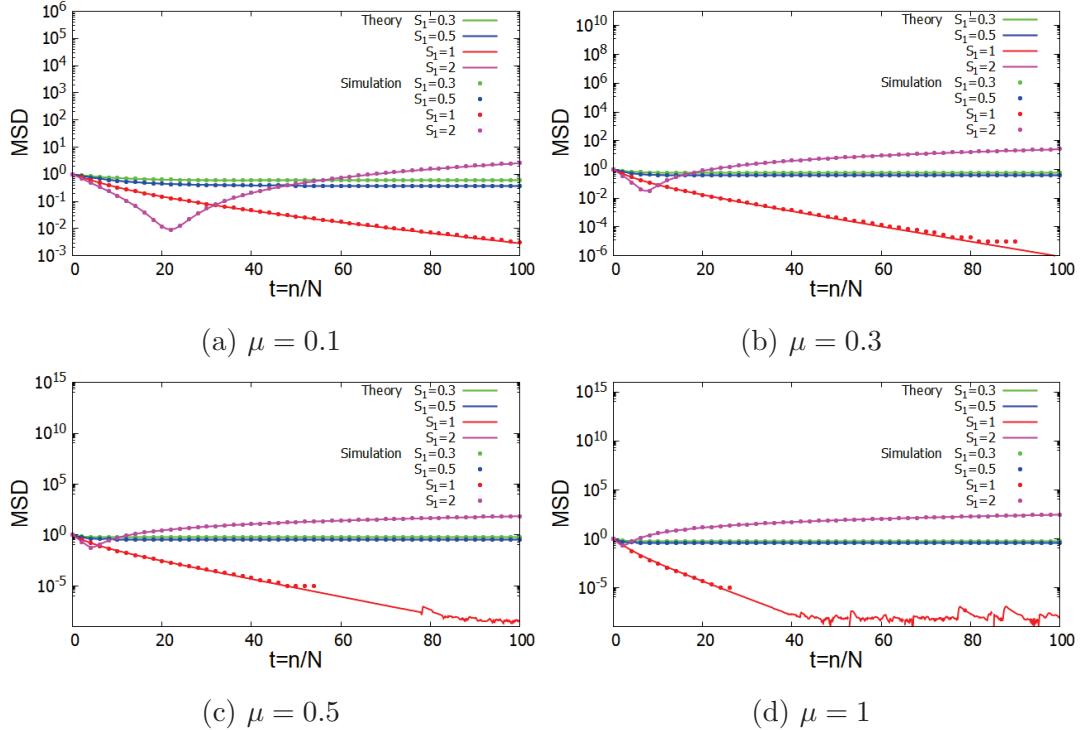


図 5.16: MSD の学習曲線

5.2.2 二つの飽和値と準定常正規化二乗平均偏差の関係

二つの飽和値と準定常 MSD の関係を図 5.17 に示す。計算機実験は $N=200$, 時刻 300 で, 100 回の試行の平均値である。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。ここでは準定常 MSD は時刻 300 での MSD を示し, S_1 を 0.3, 0.5, 1, 2 に固定した場合の結果である。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2 = 1$ の場合の結果であり, ステップサイズ $\mu=0.1$ について計算した。図 5.17 から, 図 5.5 と同様に $S_1=S_2$ 付近で準定常 MSD が 0 に漸近し, 最小値をとっていることが分かる。しかし, $S_1=0.3, 0.5$ では MSE が十分に収束していないことから, 準定常 MSD が最小となる S_2/S_1 の値は 1 から少しづれる結果となった。また, S_1 と S_2 の差が大きくなるほど準定常 MSD が大きくなつた。 S_2 の値が大きいほど準定常 MSD の初期値は大きいが, $S_1 \leq S_2$ では S_1 の値が大きいほど準定常 MSD は小さくなる結果となつた。図 5.5 では, $S_1=2$ において理論とシミュレーションの誤差があったが, 準定常 MSD では誤差は見られなかった。これは MSE の場合とは異なり, 式 (3.20) の MSD の計算式に一重積分の項が含まれないためだと思われる。

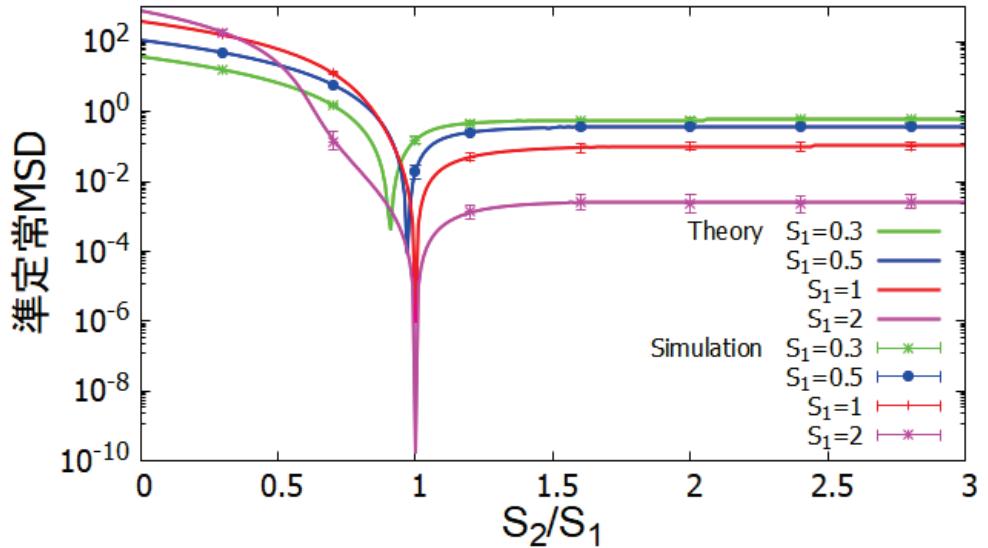


図 5.17: 二つの飽和値と準定常二乗平均偏差の関係 ($\sigma_\xi^2=0$)

二つの飽和値と準定常 MSD の関係を図 5.18 に示す。計算機実験は $N=200$, 時刻 300 で, 100 回の試行の平均値である。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。ここでは準定常 MSD は時刻 300 での MSD を示し, S_1 を 0.3, 0.5, 1, 2 に固定した場合の結果である。背景雑音の分散は $\sigma_\xi^2 = 1$, $\rho^2 = 1$ の場合の結果であり, ステップサイズ $\mu = 0.1$ について計算した。時刻 300 では MSD が十分に収束していないことから, 準定常 MSD が最小となる S_2/S_1 の値は 1 から少しずれる結果となった。背景雑音がある場合では準定常 MSD は 0 に漸近しない結果となった。また, 図 5.17 と同様に S_1 の値が大きいほど準定常 MSD の初期値は大きいが, $S_1 \leq S_2$ では S_1 の値が大きいほど準定常 MSD は小さくなる結果となった。

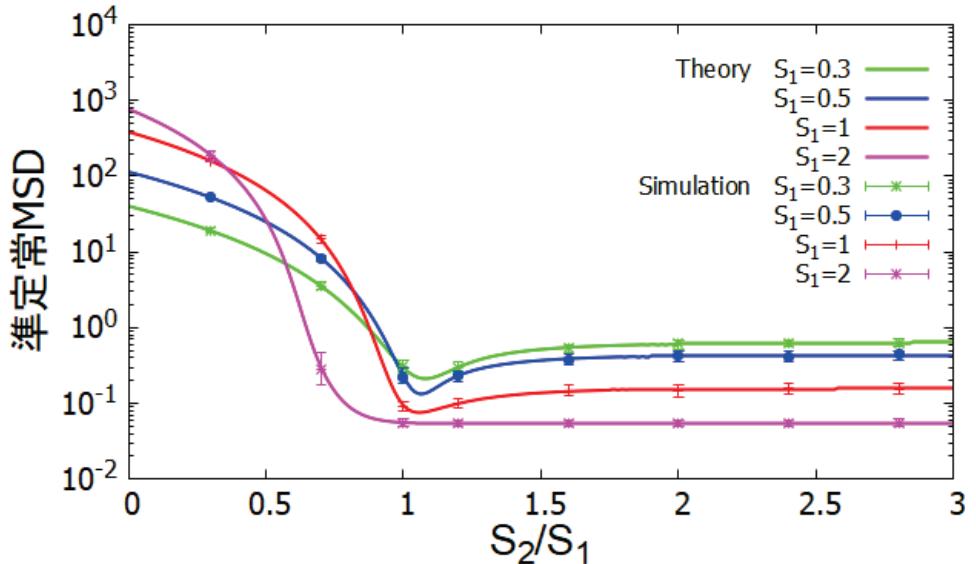


図 5.18: 二つの飽和値と準定常二乗平均偏差の関係 ($\sigma_\xi^2=1$)

二つの飽和値と準定常 MSD の関係を図 5.19 に示す。計算機実験は $N=200$, 時刻 300 で, 100 回の試行の平均値である。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。ここでは準定常 MSD は時刻 300 での MSD を示し, S_2 を 0.3, 0.5, 1, 2 に固定した場合の結果である。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2 = 1$ の場合の結果であり, ステップサイズ $\mu = 0.1$ について計算した。図 5.19 から, 図 5.8 と同様に $S_1=S_2$ 付近で準定常 MSD が 0 に漸近し, 最小値をとっていることが分かる。しかし, $S_2=0.3, 0.5$ では MSE が十分に収束していないことから, 準定常 MSD が最小となる S_2/S_1 の値は 1 から少しづれる結果となった。また, S_1 と S_2 の差が大きくなるほど準定常 MSD が大きくなった。図 5.8 では, $S_1=2$ において理論とシミュレーションの誤差があったが, 準定常 MSD では誤差は見られなかった。これは MSE の場合とは異なり, 式 (3.20) の MSD の計算式に一重積分の項が含まれないためだと考えられる。

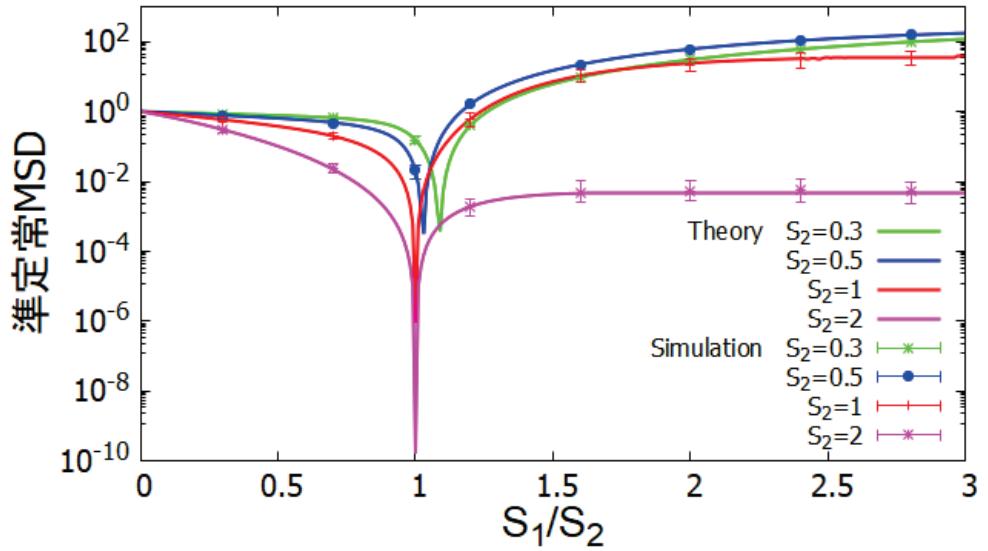


図 5.19: 二つの飽和値と準定常二乗平均偏差の関係 ($\sigma_\xi^2=0$)

二つの飽和値と準定常 MSD の関係を図 5.20 に示す。計算機実験は $N=200$, 時刻 300 で, 100 回の試行の平均値である。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。ここでは準定常 MSD は時刻 300 での MSD を示し, S_2 を 0.3, 0.5, 1, 2 に固定した場合の結果である。背景雑音の分散は $\sigma_\xi^2 = 1$, $\rho^2 = 1$ の場合の結果であり, ステップサイズ $\mu = 0.1$ について計算した。時刻 300 では MSD が十分に収束していないことから, 準定常 MSD が最小となる S_2/S_1 の値は 1 から少しずれる結果となった。また背景雑音がある場合には準定常 MSD は 0 に漸近しない結果となった。

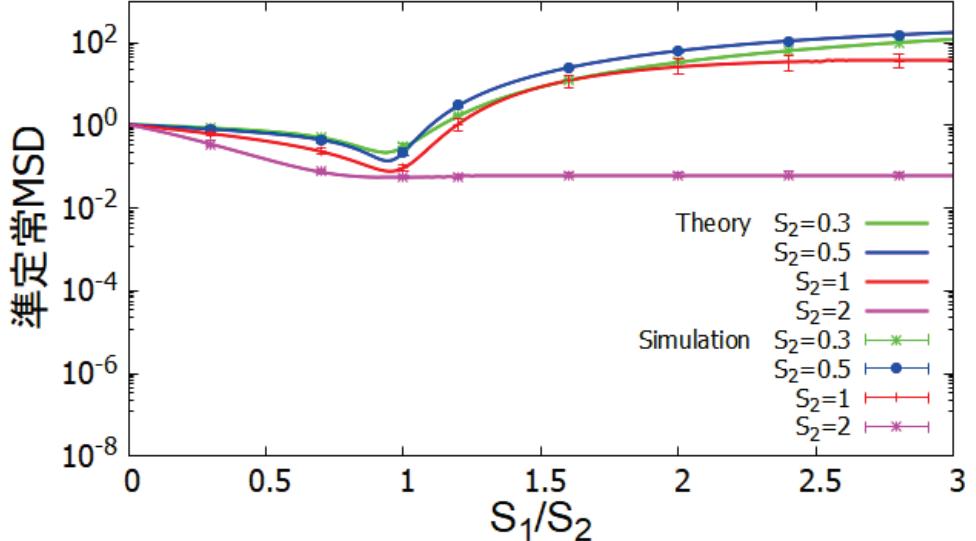


図 5.20: 二つの飽和値と準定常二乗平均偏差の関係 ($\sigma_\xi^2=1$)

二つの飽和値と準定常 MSD の関係を表したヒートマップを図 5.21, 5.22 に示す。計算機実験は $N=200$, 時刻 300 で, 100 回の試行の平均値であり, 理論の準定常 MSE は時刻 300 での MSE を示す。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2=1$ の場合の結果であり, ステップサイズ $\mu=0.1$ について計算した。図 5.21, 5.22 から, 計算機実験と理論はよく合っており, $S_1=S_2$ の部分で準定常 MSD が小さくなっていることが分かる。 S_1 が大きく S_2 が小さい部分では準定常 MSD の値が大きくなっていることがわかる。これは未知システムの出力に対して適応フィルタの出力が大きく制限され誤差が小さくならないのにかかわらず, 適応フィルタは誤差を減らそうと更新されるため, 適応フィルタが発散し, 結果的に MSD が発散していると考えられる。

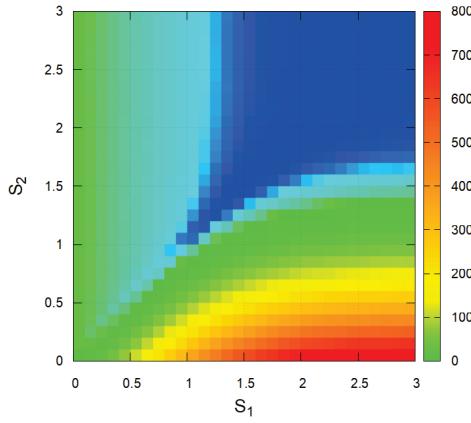


図 5.21: 計算機実験 ($\sigma_\xi^2=0$)

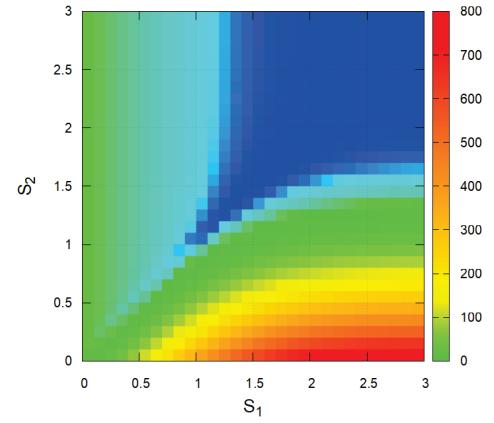


図 5.22: 理論 ($\sigma_\xi^2=0$)

二つの飽和値と準定常 MSD の関係を表したヒートマップを図 5.23, 5.24 に示す。計算機実験は $N=200$, 時刻 300 で, 100 回の試行の平均値であり, 理論の準定常 MSE は時刻 300 での MSE を示す。理論は式 (4.18) の一重積分を用いて式 (4.6), (4.21), (4.24) を数値的に解くことにより求めた。背景雑音の分散は $\sigma_\xi^2 = 0$, $\rho^2 = 1$ の場合の結果であり, ステップサイズ $\mu = 0.1$ について計算した。図 5.23, 5.24 から, 計算機実験と理論はよく合っており, $S_1=S_2$ の部分で準定常 MSD が小さくなっていることが分かる。また図 5.21, 5.22 と比較して準定常 MSD の最小値が大きくなっていることがわかる。 S_1 が大きく S_2 が小さい部分では準定常 MSD の値が大きくなっている。これは未知システムの出力に対して適応フィルタの出力が大きく制限され誤差が小さくならないのにかかわらず, 適応フィルタは誤差を減らそうと更新されるため, 適応フィルタが発散し, 結果的に MSD が発散していると考えられる。

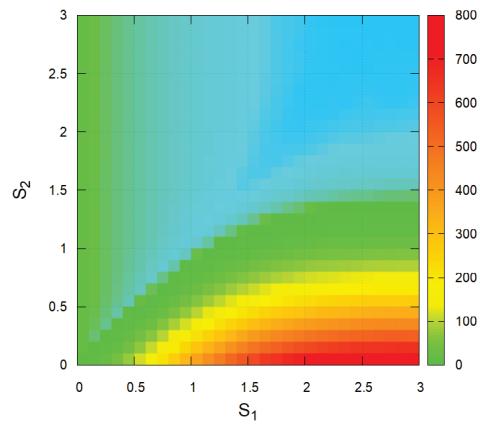


図 5.23: 計算機実験 ($\sigma_\xi^2=1$)

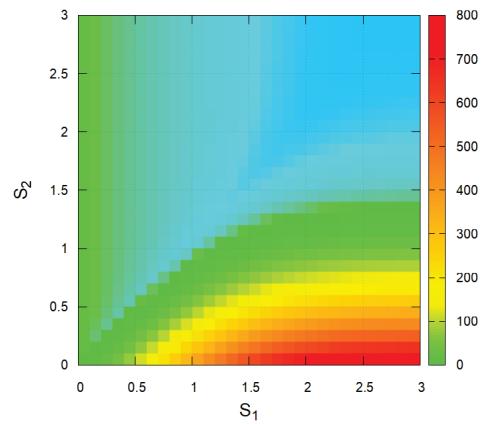


図 5.24: 理論 ($\sigma_\xi^2=1$)

第6章 結論

本論文では、LMS アルゴリズムを用いた適応信号処理システムにおいて、未知システム出力と適応フィルタ出力の両方が飽和型非線形性を有する場合の動的・静的ふるまいを統計力学的手法を用いて理論的に明らかにした。二乗平均誤差および二乗平均偏差を導出し、未知システムの係数ベクトルと適応フィルタの係数ベクトルの内積 r 、適応フィルタの係数ベクトルの二乗ノルム Q を巨視的変数とし、これらの動的ふるまいを記述する連立微分方程式をフィルタのタップ長が十分に長いという条件の下で数値的に解いた。

第3章では未知システムと適応フィルタの出力が非線形性を有する場合についての統計力学的解析を行った。誤差 e の両辺を二乗することで二乗平均誤差の理論式を導出し、未知システムと適応フィルタの係数ベクトルの差の L2 ノルムを二乗することで二乗平均偏差の理論式を導出した。また、巨視的変数 r と Q の微分方程式を導入し、二乗平均誤差に関するサンプル平均 $\langle f_G^2(d) \rangle$, $\langle f_G(d) f_W(y) \rangle$, $\langle f_W^2(y) \rangle$, $\langle d^2 \rangle$, $\langle dy \rangle$, $\langle y^2 \rangle$ を導出した。さらに導出した MSD は N に比例することがわかるので N で正規化し正規化 MSD とした。

第4章では、未知システム出力と適応フィルタ出力の両方が飽和型非線形性を有する適応信号処理システムについて解析を行った。二乗平均誤差および巨視的変数 Q の微分方程式に含まれるサンプル平均 $\langle f_G(d) f_W(y) \rangle$ について、九つの二重積分を d に関して解析的に積分することで一重積分の形に変換を行った。また MSE, MSD は巨視的変数 r , Q の関数であるため、 r と Q の微分方程式をそれぞれ導出した。

第5章では未知システム出力と適応フィルタ出力の両方が飽和型非線形性を有する適応信号処理システムの解析に関する考察を行った。二つの飽和値が学習曲線および定常値に与える影響について調べた。すなわち S_1 を固定して S_2 を変化させた場合、 S_2 を固定して S_1 を変化させた場合の二乗平均誤差および二乗平均偏差の動的・静的ふるまいを統計力学的手法を用いて理論的に求めた。二乗平均誤差、二乗平均偏差の動的・静的ふるまいが計算機実験とよく一致した。二乗平均誤差および巨視的変数 Q の微分方程式に含まれるサンプル平均 $\langle f_G(d) f_W(y) \rangle$ について、二重積分を数値的に解いた場合 $d(n)$ と $y(n)$ の共分散行列が特異に近くなり数値的な積分が不安定になった。そこで d に関する積分を解析的に実行し y に関する一重積分に変換することで、数値計算の精度と速度を改善することに成功した。MSE, MSD の学習曲線から $S_1 = S_2$ の時

に MSE, MSD は 0 に漸近することが分かった. ステップサイズを大きくすると MSE, MSD が 0 に漸近するのが速くなる一方, 理論に基づくものが生じやすくなつた. またステップサイズ $\mu=0.1$ において数値計算が安定する時刻 300 での MSE, MSD を準定常 MSE, 準定常 MSD として, ヒートマップにより二つの飽和値が準定常 MSE, 準定常 MSD に与える影響を明らかにした. $S_1 = S_2$ において準定常 MSE, 準定常 MSD は最小値をとり, S_1 が大きく, S_2 が小さい場合には準定常 MSE, 準定常 MSD が大きくなることが分かった. サンプル平均 $\langle f_G(d) f_W(y) \rangle$ を一重積分に変換することで数値計算の精度と速度の改善に成功したが, 正確な学習曲線, 定常値を求めるることはできていないため, 数値計算精度のさらなる向上が今後の課題である.

謝辞

本研究を進めるにあたり終始お世話になった関西大学システム理工学部教授の三好先生、准教授の本仲先生に深く御礼申し上げます。三好教授には本研究を進めるための基礎知識、計算機実験や理論解析などの方法について丁寧で熱心なご指導をしていただきました。本仲先生には研究室生活を送る上でのサポートや、研究内容に関する相談などをさせていただきました。これらの様々な研究のこと、研究外のこと問わず、助けていただいたことに心から感謝の意を表します。

最後に大学に進学させていただき、大学での生活を金銭面だけでなく精神面等で支えてくれた家族には心から感謝の意を表します。

参考文献

- [1] 貴家 仁志, ディジタル信号処理のエッセンス, オーム社.
- [2] 中西 功, 尾知 博, 堀田 英輔, “適応信号処理”, 電子情報通信学会知識ベース「知識の森」, 1群9編, <http://www.ieice-hbkb.org/>, 参照 Feb.2009.
- [3] 梶川 嘉延, “適応信号処理の基礎と応用 [VI・完] : 非線形適応信号処理”, 電子情報通信学会誌, Vol.87, No2, pp.123–128, Feb.2004.
- [4] P. A. Nelson and S. J. Elliott, *Active Control of Sound*, San Diego, CA, USA: Academic, 1992.
- [5] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems—Algorithms and DSP Implementations*, New York, NY, USA: Wiley, 1996.
- [6] S. M. Kuo and D. R. Morgan, “Active noise control: A tutorial review,” *Proc. IEEE*, vol. 87, no. 6, pp. 943–973, Jun. 1999.
- [7] Y. Kajikawa, W.-S. Gan, and S. M. Kuo, “Recent advances on active noise control: Open issues and innovative applications,” *IEEE Transactions on Industrial Electronics*, vol. 50, no. 3, pp. 561–569, Jun. 2003.
- [8] M. M. Sondhi, “The history of echo cancellation,” *IEEE Signal Process. Mag.*, vol. 23, no. 5, pp. 95–102, Sep. 2006.
- [9] 西森 秀稔, スピングラス理論と情報統計力学, 岩波書店, 東京, 1999.
- [10] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*, Oxford University Press, New York, 2001.
- [11] S. Miyoshi and Y. Kajikawa, “Statistical-Mechanics Approach to Theoretical Analysis of the FXLMS Algorithm”, *IEICE Trans. Fundam. Commun. Comput. Sci.*, Vol.E101-A, No.12, pp. 2419-2433, Dec. 2018.

- [12] M. H. Costa, L. R. Ximenes, and J. C. Bermudez, “Statistical analysis of the LMS adaptive algorithm subjected to a symmetric dead-zone nonlinearity at the adaptive filter output,” *Signal Processing*, vol. 88, no. 6, pp. 1485–1495, 2008.
- [13] S. Miyoshi, “Statistical-mechanical analysis of adaptive filter with clipping saturation-type nonlinearity,” *IEEE Trans. Signal Process.*, vol. 70, pp. 4867–4882, 2022.
- [14] M. H. Costa, J. C. M. Bermudez, and N. J. Bershad, “Statistical analysis of the LMS adaptive algorithm with a saturation nonlinearity following the adaptive filter output”, *IEEE Trans. Signal Process.*, vol. 49, no.7, pp. 1370-1387, Oct. 2001.
- [15] S. Miyoshi and Y. Yamaguchi, “Exact equality of the MSEs for two types of nonlinear adaptive systems: Saturation and dead-zone types,” *IEEE Access*, 10.1109/ACCESS.2024.3510422, 2024.
- [16] 三好誠司, “オンライン学習の統計力学的解析”, システム／制御／情報, Vol.51, No.5, pp.216–223, May 2007.
- [17] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, Hoboken, NJ, 2003.
- [18] S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2002.

付 錄A 計算の詳細

A.1 $\langle f_G^2(d) \rangle$

$$\langle f_G^2(d) \rangle = \int_{-\infty}^{\infty} f_G(d)^2 P(d) dd \quad (\text{A.1})$$

$$= \int_{-\infty}^{-S_1} (-S_1)^2 \frac{1}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd + \int_{-S_1}^{S_1} d^2 \frac{1}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.2})$$

$$+ \int_{S_1}^{\infty} S_1^2 \frac{1}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.3})$$

$$= 2 \left(\frac{\int_{S_1}^{\infty} S_1^2 \frac{1}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd}{A1} + \frac{\int_0^{S_1} d^2 \frac{1}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd}{A2} \right) \quad (\text{A.4})$$

$$A1 = \frac{S_1^2}{\sqrt{2\rho^2\pi\sigma_g^2}} \int_{S_1}^{\infty} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.5})$$

$$= \frac{S_1^2}{\sqrt{\pi}} \int_{\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}}^{\infty} \exp(-d'^2) dd', \text{ where } d' = \frac{d}{\sqrt{2\rho^2\sigma_g^2}} \quad (\text{A.6})$$

$$= \frac{S_1^2}{2} \left(1 - \operatorname{erf}\left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}\right) \right) \quad (\text{A.7})$$

$$A2 = \frac{1}{\sqrt{2\rho^2\pi\sigma_g^2}} \int_0^{S_1} d^2 \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.8})$$

$$= \frac{2\rho^2\sigma_g^2}{\sqrt{\pi}} \int_0^{\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}} d'^2 \exp(-d'^2) dd', \text{ where } d' = \frac{d}{\sqrt{2\rho^2\sigma_g^2}} \quad (\text{A.9})$$

$$= \frac{\rho^2\sigma_g^2}{2} \operatorname{erf}\left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}\right) - \sqrt{\frac{\rho^2 S_1^2 \sigma_g^2}{2\pi}} \exp\left(-\frac{S_1^2}{2\rho^2\sigma_g^2}\right) \quad (\text{A.10})$$

$$\langle f_G^2(d) \rangle = 2(A1 + A2) \quad (\text{A.11})$$

$$= S_1^2 + (\rho^2 \sigma_g^2 - S_1^2) \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}} \right) - \sqrt{\frac{2\rho^2 S_1^2 \sigma_g^2}{\pi}} \exp \left(-\frac{S_1^2}{2\rho^2 \sigma_g^2} \right) \quad (\text{A.12})$$

A.2 $\langle f_W^2(y) \rangle$

$$\langle f_W^2(y) \rangle = \int_{-\infty}^{\infty} f_W^2(y) P(y) dy \quad (\text{A.13})$$

$$= \int_{-\infty}^{-S_2} (-S_2)^2 \frac{1}{\sqrt{2\rho^2 \pi Q}} \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy + \int_{-S_2}^{S_2} y^2 \frac{1}{\sqrt{2\rho^2 \pi Q}} \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.14})$$

$$+ \int_{S_2}^{\infty} S_2^2 \frac{1}{\sqrt{2\rho^2 \pi Q}} \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.15})$$

$$= 2 \left(\underbrace{\int_{S_2}^{\infty} S_2^2 \frac{1}{\sqrt{2\rho^2 \pi Q}} \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy}_{B1} + \underbrace{\int_0^{S_2} y^2 \frac{1}{\sqrt{2\rho^2 \pi Q}} \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy}_{B2} \right) \quad (\text{A.16})$$

$$B1 = \frac{S_2^2}{\sqrt{2\rho^2 \pi Q}} \int_{S_2}^{\infty} \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.17})$$

$$= \frac{S_2^2}{\sqrt{\pi}} \int_{\frac{S_2}{\sqrt{2\rho^2 Q}}}^{\infty} \exp(-y'^2) dy', \text{ where } y' = -\frac{y}{\sqrt{2\rho^2 Q}} \quad (\text{A.18})$$

$$= \frac{S_2^2}{2} \left(1 - \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) \right) \quad (\text{A.19})$$

$$B2 = \frac{1}{\sqrt{2\rho^2 \pi Q}} \int_0^{S_2} y^2 \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.20})$$

$$= \frac{2\rho^2 Q}{\sqrt{\pi}} \int_0^{\frac{S_2}{\sqrt{2\rho^2 Q}}} y'^2 \exp(-y'^2) dy', \text{ where } y' = -\frac{y}{\sqrt{2\rho^2 Q}} \quad (\text{A.21})$$

$$= \frac{\rho^2 Q}{2} \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) - \sqrt{\frac{2\rho^2 S_2^2 Q}{\pi}} \exp \left(-\frac{S_2^2}{2\rho^2 Q} \right) \quad (\text{A.22})$$

$$\langle f_W^2(y) \rangle = B1 + B2 \quad (\text{A.23})$$

$$= S_2^2 + (\rho^2 Q - S_2^2) \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) - \sqrt{\frac{2\rho^2 S_2^2 Q}{\pi}} \exp \left(-\frac{S_2^2}{2\rho^2 Q} \right) \quad (\text{A.24})$$

A.3 $\langle df_G(d) \rangle$

$$\langle df_G(d) \rangle = \int_{-\infty}^{\infty} df_G(d) P(d) dd \quad (\text{A.25})$$

$$= \frac{\int_{-\infty}^{-S_1} (-S_1) \frac{d}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd + \int_{-S_1}^{S_1} \frac{d^2}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd}{C1} \quad (\text{A.26})$$

$$+ \frac{\int_{S_1}^{\infty} S_1 \frac{d}{\sqrt{2\rho^2\pi\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd}{C3} \quad (\text{A.27})$$

$$C1 = C3 = \frac{S_1}{\sqrt{2\rho^2\pi\sigma_g^2}} \int_{S_1}^{\infty} d \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.28})$$

$$= \frac{S_1}{\sqrt{2\rho^2\pi\sigma_g^2}} \left[-\rho^2\sigma_g^2 \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) \right]_{S_1}^{\infty} \quad (\text{A.29})$$

$$= S_1 \sqrt{\frac{\rho^2\sigma_g^2}{2\pi}} \exp\left(-\frac{S_1^2}{2\rho^2\sigma_g^2}\right) \quad (\text{A.30})$$

$$C2 = 2 \int_0^{S_1} \frac{d^2}{\sqrt{2\pi\rho^2\sigma_g^2}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.31})$$

$$= 2 \left(\left[-d \sqrt{\frac{\rho^2\sigma_g^2}{2\pi}} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) \right]_0^{S_1} + \sqrt{\frac{\rho^2\sigma_g^2}{2\pi}} \int_0^{S_1} \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \right) \quad (\text{A.32})$$

$$= -S_1 \sqrt{\frac{2\rho^2\sigma_g^2}{\pi}} \exp\left(-\frac{S_1^2}{2\rho^2\sigma_g^2}\right) + \rho^2\sigma_g^2 \operatorname{erf}\left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}\right) \quad (\text{A.33})$$

$$\langle df_G(d) \rangle = C1 + C2 + C3 \quad (\text{A.34})$$

$$= \rho^2\sigma_g^2 \operatorname{erf}\left(\frac{S_1}{\sqrt{2\rho^2\sigma_g^2}}\right) \quad (\text{A.35})$$

A.4 $\langle df_W(y) \rangle$

$$\langle df_W(y) \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} df_W(y) P(d, y) dd dy \quad (\text{A.36})$$

$$= \frac{\int_{-\infty}^{\infty} d dd \int_{-\infty}^{-S_2} (-S_2) P(d, y) dy}{D1} + \frac{\int_{-\infty}^{\infty} d dd \int_{-S_2}^{S_2} y P(d, y) dy}{D2} + \frac{\int_{-\infty}^{\infty} d dd \int_{S_2}^{\infty} S_2 P(d, y) dy}{D3} \quad (\text{A.37})$$

$$D1 = \int_{-\infty}^{\infty} d dd \int_{-\infty}^{-S_2} (-S_2) \frac{1}{2\pi \sqrt{\left| \rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \right|}} \exp \left(-\frac{\left(\begin{pmatrix} d \\ y \end{pmatrix}^T \begin{pmatrix} \rho^2 \sigma_g^2 & \rho^2 r \\ \rho^2 r & \rho^2 Q \end{pmatrix}^{-1} \begin{pmatrix} d \\ y \end{pmatrix} \right)^T}{2} \right) dy \quad (\text{A.38})$$

$$= \int_{-\infty}^{\infty} d dd \int_{-\infty}^{-S_2} (-S_2) \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(-\frac{Qd^2 - 2rdy + \sigma_g^2 y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.39})$$

$$= \int_{-\infty}^{\infty} d dd \int_{-\infty}^{-S_2} (-S_2) \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(-\frac{Q(d - \frac{r}{Q}y)^2 + (\sigma_g^2 - \frac{r^2}{Q})y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.40})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \int_{-\infty}^{-S_2} (-S_2) \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \int_{-\infty}^{\infty} d \exp \left(-\frac{(d - \frac{r}{Q}y)^2}{2\rho^2 (\sigma_g^2 - \frac{r^2}{Q})} \right) dd \quad (\text{A.41})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \frac{\rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}}}{Q} r \sqrt{2\pi} \int_{-\infty}^{-S_2} y (-S_2) \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.42})$$

$$= \frac{-rS_2}{\sqrt{2\pi \rho^2 Q^3}} \int_{-\infty}^{-S_2} y \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.43})$$

$$= \frac{\rho S_2 r \sqrt{2}}{\sqrt{\pi Q}} \int_{\frac{S_2}{\rho \sqrt{2Q}}}^{\infty} y' \exp(-y'^2) dy', \text{ where } y' = -\frac{y}{\sqrt{2\rho^2 Q}} \quad (\text{A.44})$$

$$= \frac{\rho S_2 r}{\sqrt{2\pi Q}} \exp \left(-\frac{S_2^2}{2\rho^2 Q} \right) \quad (\text{A.45})$$

$$D2 = \int_{-\infty}^{\infty} d dd \int_{-S_2}^{S_2} y \frac{1}{2\pi \sqrt{\left| \rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \right|}} \exp \left(- \frac{\left(\begin{pmatrix} d \\ y \end{pmatrix}^T \begin{pmatrix} \rho^2 \sigma_g^2 & \rho^2 r \\ \rho^2 r & \rho^2 Q \end{pmatrix}^{-1} \begin{pmatrix} d \\ y \end{pmatrix} \right)^T}{2} \right) dy \quad (\text{A.46})$$

$$= \int_{-\infty}^{\infty} d dd \int_{-S_2}^{S_2} y \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(- \frac{Qd^2 - 2rdy + \sigma_g^2 y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.47})$$

$$= \int_{-\infty}^{\infty} d dd \int_{-S_2}^{S_2} y \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(- \frac{Q(d - \frac{r}{Q}y)^2 + (\sigma_g^2 - \frac{r^2}{Q})y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.48})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \int_{-S_2}^{S_2} y \exp \left(- \frac{y^2}{2\rho^2 Q} \right) dy \underbrace{\int_{-\infty}^{\infty} d \exp \left(- \frac{(d - \frac{r}{Q}y)^2}{2\rho^2 (\sigma_g^2 - \frac{r^2}{Q})} \right) dd}_{A4} \quad (\text{A.49})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \frac{\rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}}}{Q} r \sqrt{2\pi} \int_{-S_2}^{S_2} y^2 \exp \left(- \frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.50})$$

$$= \frac{r}{\sqrt{2\pi \rho^2 Q^3}} \int_{-S_2}^{S_2} y^2 \exp \left(- \frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.51})$$

$$= \frac{2\rho^2 r}{\sqrt{\pi}} \int_{\frac{-S_2}{\rho \sqrt{2Q}}}^{\frac{S_2}{\rho \sqrt{2Q}}} y'^2 \exp(-y'^2) dy', \text{ where } y' = -\frac{y}{\sqrt{2\rho^2 Q}} \quad (\text{A.52})$$

$$= \frac{-2\rho S_2 r}{\sqrt{2\pi Q}} \exp \left(- \frac{S_2^2}{2\rho^2 Q} \right) + \rho^2 r \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) \quad (\text{A.53})$$

$$D3 = \int_{-\infty}^{\infty} d dd \int_{S_2}^{\infty} S_2 \frac{1}{2\pi \sqrt{\left| \rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \right|}} \exp \left(-\frac{\left(\begin{pmatrix} d \\ y \end{pmatrix}^T \begin{pmatrix} \rho^2 \sigma_g^2 & \rho^2 r \\ \rho^2 r & \rho^2 Q \end{pmatrix}^{-1} \begin{pmatrix} d \\ y \end{pmatrix} \right)^T}{2} \right) dy \quad (\text{A.54})$$

$$= \int_{-\infty}^{\infty} d dd \int_{S_2}^{\infty} S_2 \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(-\frac{Qd^2 - 2rdy + \sigma_g^2 y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.55})$$

$$= \int_{-\infty}^{\infty} d dd \int_{S_2}^{\infty} S_2 \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(-\frac{Q(d - \frac{r}{Q}y)^2 + (\sigma_g^2 - \frac{r^2}{Q})y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.56})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \int_{S_2}^{\infty} S_2 \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \underbrace{\int_{-\infty}^{\infty} d dd \exp \left(-\frac{(d - \frac{r}{Q}y)^2}{2\rho^2 \left(\sigma_g^2 - \frac{r^2}{Q} \right)} \right)}_{A4} \quad (\text{A.57})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \frac{\rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}}}{Q} r \sqrt{2\pi} \int_{S_2}^{\infty} y S_2 \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.58})$$

$$= \frac{r S_2}{\sqrt{2\pi \rho^2 Q^3}} \int_{S_2}^{\infty} y \exp \left(-\frac{y^2}{2\rho^2 Q} \right) dy \quad (\text{A.59})$$

$$= \frac{\rho S_2 r \sqrt{2}}{\sqrt{\pi Q}} \int_{\frac{S_2}{\rho \sqrt{2Q}}}^{\infty} y' \exp(-y'^2) dy', \text{ where } y' = \frac{y}{\sqrt{2\rho^2 Q}} \quad (\text{A.60})$$

$$= \frac{\rho S_2 r}{\sqrt{2\pi Q}} \exp \left(-\frac{S_2^2}{2\rho^2 Q} \right) \quad (\text{A.61})$$

$$\langle df(y) \rangle = D1 + D2 + D3 = \rho^2 r \operatorname{erf} \left(\frac{S_2}{\sqrt{2\rho^2 Q}} \right) \quad (\text{A.62})$$

$$D4 = \int_{-\infty}^{\infty} \rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}} dd' \left(d' \rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}} + \frac{yr}{Q} \right) \exp \left(-\frac{d'^2}{2} \right), \text{ where } d' = \frac{d - \frac{r}{Q}y}{\rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}}} \quad (\text{A.63})$$

$$= \rho^2 \left(\sigma_g^2 - \frac{r^2}{Q} \right) \int_{-\infty}^{\infty} dd' d' \exp \left(-\frac{d'^2}{2} \right) + \frac{\rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}}}{Q} yr \int_{-\infty}^{\infty} dd' \exp \left(-\frac{d'^2}{2} \right) \quad (\text{A.64})$$

$$= \frac{\rho \sqrt{\sigma_g^2 - \frac{r^2}{Q}}}{Q} yr \sqrt{2\pi} \quad (\text{A.65})$$

A.5 $\langle y f_G(d) \rangle$

$$\langle y f_G(d) \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f_G(d) P(d, y) dd dy \quad (\text{A.66})$$

$$= \frac{\int_{-\infty}^{\infty} y dy \int_{-\infty}^{-S_1} (-S_1) P(d, y) dd}{E1} + \frac{\int_{-\infty}^{\infty} y dy \int_{-S_1}^{S_1} dP(d, y) dd}{E2} + \frac{\int_{-\infty}^{\infty} y dy \int_{S_1}^{\infty} S_1 P(d, y) dd}{E3} \quad (\text{A.67})$$

$$E1 = \int_{-\infty}^{\infty} y dy \int_{-\infty}^{-S_1} (-S_1) \frac{1}{2\pi\sqrt{\left|\rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix}\right|}} \exp\left(-\frac{\left(\begin{pmatrix} d \\ y \end{pmatrix}^T \begin{pmatrix} \rho^2 \sigma_g^2 & \rho^2 r \\ \rho^2 r & \rho^2 Q \end{pmatrix}^{-1} \begin{pmatrix} d \\ y \end{pmatrix}\right)^2}{2}\right) dd \quad (\text{A.68})$$

$$= \int_{-\infty}^{\infty} y dy \int_{-\infty}^{-S_1} (-S_1) \frac{1}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \exp\left(-\frac{Qd^2 - 2rdy + \sigma_g^2 y^2}{2\rho^2(\sigma_g^2 Q - r^2)}\right) dd \quad (\text{A.69})$$

$$= \int_{-\infty}^{\infty} y dy \int_{-\infty}^{-S_1} (-S_1) \frac{1}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \exp\left(-\frac{\sigma_g^2(y - \frac{r}{\sigma_g^2}d)^2 + (Q - \frac{r^2}{\sigma_g^2})d^2}{2\rho^2(\sigma_g^2 Q - r^2)}\right) dd \quad (\text{A.70})$$

$$= \frac{1}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \int_{-\infty}^{-S_1} (-S_1) \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \int_{-\infty}^{\infty} y \exp\left(-\frac{(y - \frac{r}{\sigma_g^2}d)^2}{2\rho^2(Q - \frac{r^2}{\sigma_g^2})}\right) dy \quad (\text{A.71})$$

$$= \frac{1}{2\pi\rho^2\sqrt{\sigma_g^2 Q - r^2}} \frac{\rho\sqrt{Q - \frac{r^2}{\sigma_g^2}}}{\sigma_g^2} r\sqrt{2\pi} \int_{-\infty}^{-S_1} d(-S_1) \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.72})$$

$$= \frac{-rS_1}{\sqrt{2\pi\rho^2\sigma_g^6}} \int_{-\infty}^{-S_1} d \exp\left(-\frac{d^2}{2\rho^2\sigma_g^2}\right) dd \quad (\text{A.73})$$

$$= \frac{\rho S_1 r \sqrt{2}}{\sqrt{\pi\sigma_g^2}} \int_{\frac{S_1}{\rho\sqrt{2Q}}}^{\infty} d' \exp(-d'^2) dd', \text{ where } d' = -\frac{d}{\sqrt{2\rho^2\sigma_g^2}} \quad (\text{A.74})$$

$$= \frac{\rho S_1 r}{\sqrt{2\pi\sigma_g^2}} \exp\left(-\frac{S_1^2}{2\rho^2\sigma_g^2}\right) \quad (\text{A.75})$$

$$E2 = \int_{-\infty}^{\infty} y dy \int_{-S_1}^{S_1} d \frac{1}{2\pi \sqrt{\left| \rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \right|}} \exp \left(- \frac{\left(\begin{pmatrix} d \\ y \end{pmatrix}^T \begin{pmatrix} \rho^2 \sigma_g^2 & \rho^2 r \\ \rho^2 r & \rho^2 Q \end{pmatrix}^{-1} \begin{pmatrix} d \\ y \end{pmatrix} \right)^T}{2} \right) dd \quad (\text{A.76})$$

$$= \int_{-\infty}^{\infty} y dy \int_{-S_1}^{S_1} d \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(- \frac{Qd^2 - 2rdy + \sigma_g^2 y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dy \quad (\text{A.77})$$

$$= \int_{-\infty}^{\infty} y dy \int_{-S_1}^{S_1} d \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(- \frac{\sigma_g^2 (y - \frac{r}{\sigma_g^2} d)^2 + (Q - \frac{r^2}{\sigma_g^2}) d^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dd \quad (\text{A.78})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \int_{-S_1}^{S_1} d \exp \left(- \frac{d^2}{2\rho^2 \sigma_g^2} \right) dd \underbrace{\int_{-\infty}^{\infty} y \exp \left(- \frac{(y - \frac{r}{\sigma_g^2} d)^2}{2\rho^2 (Q - \frac{r^2}{\sigma_g^2})} \right) dy}_{A4} \quad (\text{A.79})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \frac{\rho \sqrt{Q - \frac{r^2}{\sigma_g^2}}}{\sigma_g^2} r \sqrt{2\pi} \int_{-S_1}^{S_1} d^2 \exp \left(- \frac{d^2}{2\rho^2 \sigma_g^2} \right) dd \quad (\text{A.80})$$

$$= \frac{r}{\sqrt{2\pi \rho^2 \sigma_g^6}} \int_{-S_1}^{S_1} d^2 \exp \left(- \frac{d^2}{2\rho^2 \sigma_g^2} \right) dd \quad (\text{A.81})$$

$$= \frac{2\rho^2 r}{\sqrt{\pi}} \int_{\frac{-S_1}{\sqrt{2\rho^2 \sigma_g^2}}}^{\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}}} d'^2 \exp(-d'^2) dd', \text{ where } d' = -\frac{d}{\sqrt{2\rho^2 \sigma_g^2}} \quad (\text{A.82})$$

$$= \frac{-2\rho S_1 r}{\sqrt{2\pi \sigma_g^2}} \exp \left(- \frac{S_1^2}{2\rho^2 \sigma_g^2} \right) + \rho^2 r \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}} \right) \quad (\text{A.83})$$

$$E3 = \int_{-\infty}^{\infty} y dy \int_{S_1}^{\infty} S_1 \frac{1}{2\pi \sqrt{\left| \rho^2 \begin{pmatrix} \sigma_g^2 & r \\ r & Q \end{pmatrix} \right|}} \exp \left(-\frac{\left(\begin{pmatrix} d \\ y \end{pmatrix}^T \begin{pmatrix} \rho^2 \sigma_g^2 & \rho^2 r \\ \rho^2 r & \rho^2 Q \end{pmatrix}^{-1} \begin{pmatrix} d \\ y \end{pmatrix} \right)^T}{2} \right) dd \quad (\text{A.84})$$

$$= \int_{-\infty}^{\infty} y dy \int_{S_1}^{\infty} S_1 \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(-\frac{Qd^2 - 2rdy + \sigma_g^2 y^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dd \quad (\text{A.85})$$

$$= \int_{-\infty}^{\infty} y dy \int_{S_1}^{\infty} S_1 \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \exp \left(-\frac{\sigma_g^2 (y - \frac{r}{\sigma_g^2} d)^2 + (Q - \frac{r^2}{\sigma_g^2}) d^2}{2\rho^2 (\sigma_g^2 Q - r^2)} \right) dd \quad (\text{A.86})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \int_{S_1}^{\infty} S_1 \exp \left(-\frac{d^2}{2\rho^2 \sigma_g^2} \right) dd \underbrace{\int_{-\infty}^{\infty} y \exp \left(-\frac{(y - \frac{r}{\sigma_g^2} d)^2}{2\rho^2 (Q - \frac{r^2}{\sigma_g^2})} \right) dy}_{E4} \quad (\text{A.87})$$

$$= \frac{1}{2\pi \rho^2 \sqrt{\sigma_g^2 Q - r^2}} \frac{\rho \sqrt{Q - \frac{r^2}{\sigma_g^2}}}{\sigma_g^2} r \sqrt{2\pi} \int_{S_1}^{\infty} d S_1 \exp \left(-\frac{d^2}{2\rho^2 \sigma_g^2} \right) dd \quad (\text{A.88})$$

$$= \frac{r S_1}{\sqrt{2\pi \rho^2 \sigma_g^6}} \int_{S_1}^{\infty} d \exp \left(-\frac{d^2}{2\rho^2 \sigma_g^2} \right) dd \quad (\text{A.89})$$

$$= \frac{\rho S_1 r \sqrt{2}}{\sqrt{\pi \sigma_g^2}} \int_{\frac{S_1}{\rho \sqrt{2} Q}}^{\infty} d' \exp(-d'^2) dd', \text{ where } d' = \frac{d}{\sqrt{2\rho^2 \sigma_g^2}} \quad (\text{A.90})$$

$$= \frac{\rho S_1 r}{\sqrt{2\pi \sigma_g^2}} \exp \left(-\frac{S_1^2}{2\rho^2 \sigma_g^2} \right) \quad (\text{A.91})$$

$$\langle y f_G(d) \rangle = E1 + E2 + E3 = \rho^2 r \operatorname{erf} \left(\frac{S_1}{\sqrt{2\rho^2 \sigma_g^2}} \right) \quad (\text{A.92})$$

$$E4 = \int_{-\infty}^{\infty} \rho \sqrt{Q - \frac{r^2}{\sigma_g^2}} \left(y' \rho \sqrt{Q - \frac{r^2}{\sigma_g^2} + \frac{yd}{\sigma_g^2}} \right) \exp \left(-\frac{y'^2}{2} \right) dy', \text{ where } y' = \frac{y - \frac{r}{\sigma_g^2} d}{\rho \sqrt{Q - \frac{r^2}{\sigma_g^2}}} \quad (\text{A.93})$$

$$= \rho^2 \left(Q - \frac{r^2}{\sigma_g^2} \right) \int_{-\infty}^{\infty} y' \exp \left(-\frac{y'^2}{2} \right) dy' + \frac{\rho \sqrt{Q - \frac{r^2}{\sigma_g^2}}}{\sigma_g^2} dr \int_{-\infty}^{\infty} \exp \left(-\frac{y'^2}{2} \right) dy' \quad (\text{A.94})$$

$$= \frac{\rho \sqrt{Q - \frac{r^2}{\sigma_g^2}}}{\sigma_g^2} dr \sqrt{2\pi} \quad (\text{A.95})$$

A.6 $\langle y f_W(y) \rangle$

$$\langle y f_W(y) \rangle = \int_{-\infty}^{\infty} y f_W(y) P(y) dy \quad (\text{A.96})$$

$$= \underbrace{\int_{-\infty}^{-S_2} (-S_2) \frac{y}{\sqrt{2\rho^2\pi Q}} \exp\left(-\frac{y^2}{2\rho^2Q}\right) dy}_{F1} + \underbrace{\int_{-S_2}^{S_2} \frac{y^2}{\sqrt{2\rho^2\pi Q}} \exp\left(-\frac{y^2}{2\rho^2Q}\right) dy}_{F2} \quad (\text{A.97})$$

$$+ \underbrace{\int_{S_2}^{\infty} S_2 \frac{y}{\sqrt{2\rho^2\pi Q}} \exp\left(-\frac{y^2}{2\rho^2Q}\right) dy}_{F3} \quad (\text{A.98})$$

$$F1 = F3 = \frac{S_2}{\sqrt{2\rho^2\pi Q}} \int_{S_2}^{\infty} y \exp\left(-\frac{y^2}{2\rho^2Q}\right) dy \quad (\text{A.99})$$

$$= \frac{S_2}{\sqrt{2\rho^2\pi Q}} \left[-\rho^2 Q \exp\left(-\frac{y^2}{2\rho^2Q}\right) \right]_{S_2}^{\infty} \quad (\text{A.100})$$

$$= S_2 \sqrt{\frac{\rho^2 Q}{2\pi}} \exp\left(-\frac{S_2^2}{2\rho^2Q}\right) \quad (\text{A.101})$$

$$F2 = 2 \int_0^{S_2} \frac{y^2}{\sqrt{2\pi\rho^2Q}} \exp\left(-\frac{y^2}{2\rho^2Q}\right) dy \quad (\text{A.102})$$

$$= 2 \left(\left[-y \sqrt{\frac{\rho^2 Q}{2\pi}} \exp\left(-\frac{y^2}{2\rho^2Q}\right) \right]_0^{S_2} + \sqrt{\frac{\rho^2 Q}{2\pi}} \int_0^{S_2} \exp\left(-\frac{y^2}{2\rho^2Q}\right) dy \right) \quad (\text{A.103})$$

$$= -S_2 \sqrt{\frac{2\rho^2 Q}{\pi}} \exp\left(-\frac{S_2^2}{2\rho^2Q}\right) + \rho^2 Q \operatorname{erf}\left(\frac{S_2}{\sqrt{2\rho^2Q}}\right) \quad (\text{A.104})$$

$$\langle y f_W(y) \rangle = F1 + F2 + F3 \quad (\text{A.105})$$

$$= \rho^2 Q \operatorname{erf}\left(\frac{S_2}{\sqrt{2\rho^2Q}}\right) \quad (\text{A.106})$$

付 錄B プログラムリスト

Listing B.1: dyhouwa2int_gakusyu.m

```
1 clc;
2 clear all;
3
4 tEnd = 100;
5
6 tic; % ストップウォッチタイマー開始
7
8 global fd2 dfd PRS RNG mu S1 S2 xi rho sgm_g
9
10 rho = 1;
11 S1 = 1;
12 S2 = 1;
13 xi = 0;
14 mu = 0.1;
15 sgm_g = sqrt(1);
16
17
18 PRS = 1e-10; % 数値積分の精度 RelTol, AbsTol
19 RNG = 7; % 数値積分の範囲を決めるパラメータ
20
21
22 fname = char(['dyhouwa_tMSE', ',tEnd=', num2str(tEnd), ',mu=', num2str(mu), ','
23 ,S1=', num2str(S1), ',S2=', num2str(S2), '.txt']);
24 Fid = fopen(fname, 'w');
25 Header = '#time Q r MSE';
26 fprintf(Fid, '%s\n', Header);
27
28 % 定数で表されるサンプル平均
29 fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) * sgm_g^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
30 dfd = rho^2 * sgm_g^2 * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
31
32 options = odeset('RelTol', 1e-10, 'AbsTol', 1e-10); % 精度の調整
33 [t, y] = ode45(@(t, y) odefun(t, y), [0 tEnd], [1e-9, 1e-9], options);
34
35 Q = y(:, 1);
36 r = y(:, 2);
37
38 [sizet, ~] = size(t);
39 MSE = zeros(sizet, 1);
40
41 % 進行状況の表示のための変数
42 totalSteps = sizet; % ループの総ステップ数
43
44 for k = 1:sizet
45
46     % 現在の進捗率を計算
47     progress = (k / totalSteps) * 100;
48     % 進行状況を表示
49     fprintf('Progress: %.2f%%\n', progress);
50
51     dycov = rho^2 * [sgm_g^2 r(k); r(k) Q(k)];
```

```

51     invcov = inv(dycov);
52
53     time = t(k);
54
55     fun1=@(d,y) (-S1).*(-S2)./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
56         invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
57     fun2=@(d,y) (-S1).*y./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
58         invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
59     fun3=@(d,y) (-S1).*S2./ (2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
60         invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
61     fun4=@(d,y) d.*(-S2)./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
62         invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
63     fun5=@(d,y) d.*y./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
64         (2,1).*d.*y+invcov(2,2).*y.^2)/2);
65     fun6=@(d,y) d.*S2./ (2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
66         (2,1).*d.*y+invcov(2,2).*y.^2)/2);
67     fun7=@(d,y) S1.*(-S2)./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
68         invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
69     fun8=@(d,y) S1.*y./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
70         (2,1).*d.*y+invcov(2,2).*y.^2)/2);
71     fun9=@(d,y) S1.*S2./ (2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
72         invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
73
74
75     fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q(k) / pi) * exp(-S2^2 / (2 * rho^2 * Q(k))) + (
76         rho^2 * Q(k) - S2^2) * erf(S2 / sqrt(2 * rho^2 * Q(k)));
77
78     % 積分範囲に上限を設ける
79     dr = RNG * sqrt(rho^2 * sgm_g^2);
80     yr = RNG * sqrt(rho^2) * max(sqrt(Q(k)), 1);
81
82     fdfy = integral2(fun1, -dr, -S1, -yr, -S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
83         integral2(fun2, -dr, -S1, -S2, S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
84         integral2(fun3, -dr, -S1, S2, +yr, 'RelTol', PRS, 'AbsTol', PRS) + ...
85         integral2(fun4, -S1, S1, -yr, -S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
86         integral2(fun5, -S1, S1, -S2, S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
87         integral2(fun6, -S1, S1, S2, +yr, 'RelTol', PRS, 'AbsTol', PRS) + ...
88         integral2(fun7, S1, +dr, -yr, -S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
89         integral2(fun8, S1, +dr, -S2, S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
90         integral2(fun9, S1, +dr, S2, +yr, 'RelTol', PRS, 'AbsTol', PRS);
91
92     MSE(k) = fd2 + fy2 - 2 * fdfy + xi;
93
94     fprintf(Fid, '%g %g %g %g\n', time, Q(k), r(k), MSE(k));
95 end
96
97 fclose(Fid); % 記録用ファイルのクローズ
98 toc;
99
100 % 微分方程式の関数
101 function dydt = odefun(t, y)
102     global fd2 dfd PRS RNG mu S1 S2 xi rho sgm_g
103     dydt = zeros(2, 1); % 行列ベクトルとして連立方程式を表示 21
104     Q = y(1);
105     r = y(2);
106
107
108     fy2 = S2^2-S2*sqrt(2*rho^2*Q/pi)*exp(-S2^2/2/rho^2/Q)+(rho^2*Q-S2^2)*erf(S2/sqrt
109         (2*rho^2*Q));
110     dfy = rho^2 * r * erf(S2/sqrt(2*rho^2*Q));
111     yfd = rho^2 * r * erf(S1/sqrt(2*rho^2*sgm_g^2));
112     yfy = rho^2 * Q * erf(S2/sqrt(2*rho^2*Q));
113
114     dycov = rho^2 * [sgm_g^2 r; r Q];
115
116     invcov = inv(dycov);

```

```

107
108 fun1=@(d,y) (-S1).*(-S2)./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
109      invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
110 fun2=@(d,y) (-S1).*y./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
111      invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
112 fun3=@(d,y) (-S1).*S2./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
113      invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
114 fun4=@(d,y) d.*(-S2)./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
115      invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
116 fun5=@(d,y) d.*y./((2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
117      (2,1).*d.*y+invcov(2,2).*y.^2)/2));
118 fun6=@(d,y) d.*S2./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
119      (2,1).*d.*y+invcov(2,2).*y.^2)/2);
120 fun7=@(d,y) S1.*(-S2)./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*
121      invcov(2,1).*d.*y+invcov(2,2).*y.^2)/2);
122 fun8=@(d,y) S1.*y./((2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
123      (2,1).*d.*y+invcov(2,2).*y.^2)/2);
124 fun9=@(d,y) S1.*S2./(2.*pi.*sqrt(det(dycov))).*exp(-(invcov(1,1).*d.^2+2*invcov
125      (2,1).*d.*y+invcov(2,2).*y.^2)/2);
126
127 dr = RNG * sqrt(rho^2 * sgm_g^2);
128 yr = RNG * sqrt(rho^2) * max(sqrt(Q), 1);
129
130 fdfy = integral2(fun1, -dr, -S1, -yr, -S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
131     integral2(fun2, -dr, -S1, -S2, S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
132     integral2(fun3, -dr, -S1, S2, +yr, 'RelTol', PRS, 'AbsTol', PRS) + ...
133     integral2(fun4, -S1, S1, -yr, -S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
134     integral2(fun5, -S1, S1, -S2, S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
135     integral2(fun6, -S1, S1, S2, +yr, 'RelTol', PRS, 'AbsTol', PRS) + ...
136     integral2(fun7, S1, +dr, -yr, -S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
137     integral2(fun8, S1, +dr, -S2, S2, 'RelTol', PRS, 'AbsTol', PRS) + ...
138     integral2(fun9, S1, +dr, S2, +yr, 'RelTol', PRS, 'AbsTol', PRS);
139
140 dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
141 dydt(2) = mu * (dfd - dfy);
142
143 end

```

Listing B.2: dyhouwalint_gakusyu.m

```

1 clc;
2 clear all;
3
4 tEnd = 100;
5
6 tic; % ストップウォッチタイマー開始
7
8 global fd2 dfd PRS RNG mu S1 S2 xi rho sgm_g
9
10 rho = 1;
11 S1 = 1;
12 S2 = 1;
13 xi = 0;
14 mu = 1;
15 sgm_g = sqrt(1);
16
17 PRS = 1e-10; % 数値積分の精度 RelTol, AbsTol
18 RNG = 8; % 数値積分の範囲を決めるパラメータ
19
20 fname = char(['dyhouwalint', ',tEnd=', num2str(tEnd), ',mu=', num2str(mu), ',',
21 S1=', num2str(S1), ',S2=', num2str(S2), '.txt']);
22 Fid = fopen(fname, 'w');
23 Header = '#time Q r MSE';
24 fprintf(Fid, '%s\n', Header);
25
26 % 定数で表されるサンプル平均
27 fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2 * sgm_g
28 ^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(S1 / sqrt(2 * rho^2 * sgm_g^2)));
29 dfd = rho^2 * sgm_g^2 * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
30
31 options = odeset('RelTol', 1e-6, 'AbsTol', 1e-6, 'InitialStep', 1e-3, 'MaxStep',
32 , 1);
33 [t, y] = ode45(@(t, y) odefun(t, y), [0 tEnd], [1e-9, 1e-9], options);
34
35 Q = y(:, 1);
36 r = y(:, 2);
37
38 [sizet, ~] = size(t);
39 MSE = zeros(sizet, 1);
40
41 for k = 1:sizet
42     fprintf('Progress: %.2f%%\n', (k / sizet) * 100);
43     dycov = rho.^2.* [sgm_g^2 r(k); r(k) Q(k)];
44     invcov = inv(dycov);
45
46     % 積分計算
47     MSE(k) = calculateMSE(Q(k), r(k), dycov, invcov);
48
49     fprintf(Fid, '%g %g %g %g\n', t(k), Q(k), r(k), MSE(k));
50 end
51
52 fclose(Fid);
53 toc;
54
55 function dydt = odefun(t, y)
56     global fd2 dfd mu S1 S2 xi rho sgm_g
57     dydt = zeros(2, 1);
58     Q = max(y(1), 1e-12); % が負値にならないように制限 Q
59     r = max(y(2), 1e-12); % が負値にならないように制限 r
60
61     erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
62     erf_arg2 = S1 / sqrt(2 * rho^2 * sgm_g^2);
63

```

```

64 % が実数で有限な値かをチェック erf_arg
65 if isreal(erf_arg1) && isfinite(erf_arg1)
66   fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
67   Q - S2^2) * erf(erf_arg1);
68 else
69   % が無効な場合 erf_arg
70   fy2 = 0; % の代わりに erf0
71 end
72 if isreal(erf_arg1) && isfinite(erf_arg1)
73   dfy = rho^2 * r * erf(erf_arg1);
74 else
75   dfy = 0; % の代わりに erf0
76 end
77 if isreal(erf_arg2) && isfinite(erf_arg2)
78   yfd = rho^2 * r * erf(erf_arg2);
79 else
80   yfd = 0; % の代わりに erf0
81 end
82 if isreal(erf_arg1) && isfinite(erf_arg1)
83   yfy = rho^2 * Q * erf(erf_arg1);
84 else
85   yfy = 0; % の代わりに erf0
86 end
87 dycov = rho.^2 .* [sgm_g.^2 r; r Q];
88
89 invcov = inv(dycov);
90
91 fdfy = calculateIntegral(Q, dycov, invcov);
92 dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
93 dydt(2) = mu * (dfd - dfy);
94 end
95
96 function result = calculateMSE(Q, r, dycov, invcov)
97   global S2 rho xi fd2
98
99   erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
100  if isreal(erf_arg1) && isfinite(erf_arg1)
101    fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
102    Q - S2^2) * erf(erf_arg1);
103  else
104    % が無効な場合 erf_arg
105    fy2 = 0; % の代わりに erf0
106  end
107  fdfy = calculateIntegral(Q, dycov, invcov);
108  result = fd2 + fy2 - 2 * fdfy + xi;
109 end
110
111 function result = calculateIntegral(Q, dycov, invcov)
112   global S1 S2 PRS RNG rho
113
114   % y に関する積分範囲
115   yr = RNG * sqrt(rho^2 * max(Q, 1e-12));
116
117   % に関して積分済みの関数を定義 (x-Inf to -S1 の範囲)
118   result1 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -yr, -S2, ,
119     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
120   result2 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -S2, S2, ,
121     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
122   result3 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), S2, yr, ,
123     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);

```

```

127     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
128 % -S1 to S1 の範囲
129 result4 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -yr, -
130     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
131 result5 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -S2,
132     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
133 result6 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), S2,
134     yr, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
135 % S1 to Inf の範囲
136 result7 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -yr, -S2, ,
137     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
138 result8 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -S2, S2, ,
139     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
140 result9 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), S2, yr, ,
141     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
142 result = result1 + result2 + result3 + result4 + result5 + result6 +
143     result7 + result8 + result9;
144 end
145 function result = integratedXLower(y, x1, dycov, invcov)
146 % -Inf からx1 までのx に関する積分結果
147 global S2 S1
148 % 次形式の係数 2
149 a = invcov(1,1);
150 b = invcov(1,2);
151 c = invcov(2,2);
152 detCov = det(dycov);
153 % の値に応じて係数を決定 y
154 if y <= -S2
155     coef = -S2;
156 elseif y >= S2
157     coef = S2;
158 else
159     coef = y;
160 end
161 % -S1 に対する積分結果
162 erf_arg = sqrt(a) * (x1 + b * y / a) / sqrt(2);
163 if isreal(erf_arg) && all(isfinite(erf_arg)) % 実数かつ有限な値であることを確認
164     result = (-S1) .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
165         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
166         (1 + erf(erf_arg));
167 else
168     % の引数が無効な場合 erf
169     result = 0;
170 end
171 end
172 end
173 function result = integratedXMiddle(y, x1, x2, dycov, invcov)
174 % x1 からx2 までのx に関する積分結果
175 global S2
176 % 次形式の係数 2
177 a = invcov(1,1);
178 b = invcov(1,2);
179 c = invcov(2,2);
180 detCov = det(dycov);

```

```

185      % の値に応じて係数を決定 y
186      if y <= -S2
187          coef = -S2;
188      elseif y >= S2
189          coef = S2;
190      else
191          coef = y;
192      end
193
194      erf_arg1 = sqrt(a./2) * (x2 + b * y ./ a);
195      erf_arg2 = sqrt(a./2) * (x1 + b * y ./ a);
196
197      % の引数が実数かつ有限な値であることを確認 erf
198      if isreal(erf_arg1) && all(isfinite(erf_arg1)) && isreal(erf_arg2) && all(
199          isfinite(erf_arg2))
200          result = coef ./ (2.*pi.*sqrt(detCov)) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
201              ((-1./a) .* (exp(-a*(x2 + b.*y./a).^2./2) - exp(-a*(x1 + b.*y./a).^2./2)) - ...
202                  (b.*y./a) .* sqrt(pi./2./a) .* (erf(erf_arg1) - erf(erf_arg2)));
203      else
204          result = 0; % の引数が無効な場合 erf
205      end
206  end
207
208  function result = integratedXUpper(y, x2, dycov, invcov)
209      % x2 から Inf までのx に関する積分結果
210      global S2 S1
211
212      % 次形式の係数 2
213      a = invcov(1,1);
214      b = invcov(1,2);
215      c = invcov(2,2);
216
217      detCov = det(dycov);
218
219      % の値に応じて係数を決定 y
220      if y <= -S2
221          coef = -S2;
222      elseif y >= S2
223          coef = S2;
224      else
225          coef = y;
226      end
227
228      % S1 に対する積分結果
229      erf_arg = sqrt(a) * (x2 + b * y / a) / sqrt(2);
230
231      % の引数が実数かつ有限な値であることを確認 erf
232      if isreal(erf_arg) && all(isfinite(erf_arg))
233          result = S1 .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
234              sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
235                  (1 - erf(erf_arg));
236      else
237          result = 0; % の引数が無効な場合 erf
238      end
239  end

```

Listing B.3: dyhouwasimu.m

```

1 clc ;
2 clear all ;
3
4 t = 5000; %input(' 時刻を入力してください標準値 ( 30 ) ') ;
5 N = 200; %input(' 次元を入力してください標準値 ( 200 ) ') ;
6 ens = 1000; %input(' 試行回数を入力してください ') ;
7 a = 2; %input(' データを記録する間隔を調整するパラメータを入力してください ') ;
8
9
10 tic; %ストップウォッチタイマー開始
11
12 mu = 0.1 ; % ステップサイズ
13 xi = 0 ; % 背景雑音
14 n = N*t+1; % 更新回数
15 S1 = 1; % 未知システム飽和値
16 S2 = 1; % 適応フィルタ飽和値
17 MSE = zeros(a*t+1 , 1); % MSE 初期化
18
19 for i = 1:ens
20     g = randn(N, 1); % 未知システム
21     w = zeros(N, 1); % 適応フィルタの初期化
22     x = randn(N, 1) / sqrt(N); % 行列の入力ベクトル N1
23     for j = 1:n
24         x = circshift(x, 1); % 入力ベクトルをすべてずらす
25         x(1) = randn / sqrt(N); % 値を上書きする
26         d = g.' * x; % 未知システムの出力
27         y = w.' * x; % 適応フィルタの出力
28
29         if d < -S1
30             d = -S1;
31         elseif d > S1
32             d = S1;
33         else
34             d = d;
35         end
36
37         if y < -S2
38             y = -S2;
39         elseif y > S2
40             y = S2;
41         else
42             y = y;
43         end
44         e = d - y + sqrt(xi) * randn; % 誤差の生成
45
46         if(rem(j,N/a)==1) % の計算 MSE
47             MSE((j-1)/(N/a)+1) = MSE((j-1)/(N/a)+1) + e^2/ens;
48         end
49
50         w = w + mu * e * x; % 適応フィルタの更新
51
52     end
53
54     disp([num2str(i*100/ens), '%']) % 進行状況の確認
55
56 end
57
58 data = [(0:1/a:t);MSE'];
59 header = 'dyhouwaMSE_Simulation';
60 fname = char(header, '(N=', num2str(N), ', T=', num2str(t), ', S1=', num2str(S1), ',',
61                 S2=', num2str(S2), ', mu=', num2str(mu), ', ens=', num2str(ens), ', xi=', num2str(
62                 xi), ').txt');
63 Fid = fopen(fname, 'w');

```

```
62 F = '#time\u30d5\u30a1\u30d5\u30a4MSE'; % 記録ファイルの一行目
63
64
65 fprintf(Fid,'%s\n',F);
66 fprintf(Fid,'%f\u30d5\u30a1\u30d5\u30a4%1.5f\n',data);
67 fclose(Fid);
68 toc; % タイマー終了
```

Listing B.4: dyhouwaSTVMSE_S1fix.m

```

1 clc;
2 clear all;
3
4 tEnd = 300;
5
6 tic; % ストップウォッチタイマー開始
7
8 global fd2 dfd PRS RNG mu S1 S2 xi rho sgm_g
9
10 rho = 1;
11 S1 = 1;
12 S2loop = 0:S1/100:S1*3; %初期値, 刻み幅, 終了値
13 xi = 0.1;
14 mu = 0.1;
15 sgm_g = sqrt(1);
16
17 PRS = 1e-6; % 数値積分の精度 RelTol, AbsTol
18 RNG = 7; % 数値積分の範囲を決めるパラメータ
19
20 fname = char(['dyhouwa1intS2loop', ', tEnd=', num2str(tEnd), ', mu=', num2str(
    mu), ', S1=', num2str(S1), ', S2=', num2str(S2), ', xi=', num2str(xi), '.txt']);
21 Fid = fopen(fname, 'w');
22 Header = '#S2/S1 STVMSE';
23 fprintf(Fid, '%s\n', Header);
24
25 % 定数で表されるサンプル平均
26 fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) * sgm_g^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
27 dfd = rho^2 * sgm_g^2 * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
28
29 for i = 1:length(S2loop)
30     S2 = S2loop(i);
31     fprintf('Progress for S2 = %.2f : %.2f%%\n', S2, (i / length(S2loop)) *
    100);
32
33 % ソルバーで時間発展を計算 ODE
34 options = odeset('RelTol', 1e-6, 'AbsTol', 1e-6); % 精度の調整
35 [t, y] = ode45(@(t, y) odefun(t, y), [0 tEnd], [1e-9, 1e-9], options);
36
37 % 最終時刻の Q, を取得 r
38 Q_final = y(end, 1);
39 r_final = y(end, 2);
40
41 % 最終時刻での共分散行列とその逆行列を計算
42 dycov_final = rho^2 * [sgm_g^2, r_final; r_final, Q_final];
43 invcov_final = inv(dycov_final);
44
45 % 最終時刻でのを計算 MSE
46 MSE_final = calculateMSE(Q_final, r_final, dycov_final, invcov_final);
47
48 % 結果をファイルに書き込み
49 fprintf(Fid, '%g %g\n', S2/S1, MSE_final);
50 end
51
52 fclose(Fid);
53 toc;
54
55 function dydt = odefun(t, y)
56     global fd2 dfd mu S1 S2 xi rho sgm_g
57     dydt = zeros(2, 1);
58     Q = max(y(1), 1e-12); % が負値にならないように制限 Q
59     r = max(y(2), 1e-12); % が負値にならないように制限 r
60
61     erf_arg1 = S2 / sqrt(2 * rho^2 * Q);

```

```

62  erf_arg2 = S1 / sqrt(2 * rho^2 * sgm_g^2);
63
64  % が実数で有限な値かをチェック erf_arg
65  if isreal(erf_arg1) && isnan(erf_arg1)
66      fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
67      Q - S2^2) * erf(erf_arg1);
68  else
69      % が無効な場合は、代わりに erf_arg0
70      fy2 = 0; % の代わりに erf0
71  end
72
73  if isreal(erf_arg1) && isnan(erf_arg1)
74      dfy = rho^2 * r * erf(erf_arg1);
75  else
76      dfy = 0; % の代わりに erf0
77  end
78
79  if isreal(erf_arg2) && isnan(erf_arg2)
80      yfd = rho^2 * r * erf(erf_arg2);
81  else
82      yfd = 0; % の代わりに erf0
83  end
84
85  if isreal(erf_arg1) && isnan(erf_arg1)
86      yfy = rho^2 * Q * erf(erf_arg1);
87  else
88      yfy = 0; % の代わりに erf0
89  end
90
91  dycov = rho.^2 .* [sgm_g.^2 r; r Q];
92
93
94  invcov = inv(dycov);
95
96
97  fdfy = calculateIntegral(Q, dycov, invcov);
98
99  dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
100 dydt(2) = mu * (dfd - dfy);
101 end
102
103 function result = calculateMSE(Q, r, dycov, invcov)
104     global S2 rho xi fd2
105
106     erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
107     if isreal(erf_arg1) && isnan(erf_arg1)
108         fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
109         Q - S2^2) * erf(erf_arg1);
110     else
111         % が無効な場合は、代わりに erf_arg0
112         fy2 = 0; % の代わりに erf0
113     end
114     fdfy = calculateIntegral(Q, dycov, invcov);
115
116     result = fd2 + fy2 - 2 * fdfy + xi;
117 end
118
119 function result = calculateIntegral(Q, dycov, invcov)
120     global S1 S2 PRS RNG rho
121
122     % y に関する積分範囲
123     yr = RNG * sqrt(rho^2) * max(sqrt(Q), 1e-9);
124
125     % に関して積分済みの関数を定義 (x-Inf to -S1 の範囲)
126     result1 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -yr, -S2, '
127     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);

```

```

126 result2 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -S2, S2, '
127     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
128 result3 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), S2, yr, '
129     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
130 % -S1 to S1 の範囲
131 result4 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -yr, -
132     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
133 result5 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -S2,
134     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
135 result6 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), S2,
136     yr, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
137 % S1 to Inf の範囲
138 result7 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -yr, -S2, '
139     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
140 result8 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -S2, S2, '
141     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
142 result9 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), S2, yr, '
143     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
144 result = result1 + result2 + result3 + result4 + result5 + result6 +
145     result7 + result8 + result9;
146 end
147
148 function result = integratedXLower(y, x1, dycov, invcov)
149 % -Inf から x1 までの x に関する積分結果
150 global S2 S1
151 % 次形式の係数 2
152 a = invcov(1,1);
153 b = invcov(1,2);
154 c = invcov(2,2);
155 detCov = det(dycov);
156 % の値に応じて係数を決定 y
157 if y <= -S2
158     coef = -S2;
159 elseif y >= S2
160     coef = S2;
161 else
162     coef = y;
163 end
164 % -S1 に対する積分結果
165 erf_arg = sqrt(a) * (x1 + b * y / a) / sqrt(2);
166 if isreal(erf_arg) && all(isfinite(erf_arg)) % 実数かつ有限な値であることを確認
167     result = (-S1) .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
168         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
169         (1 + erf(erf_arg));
170 else
171     % の引数が無効な場合 erf
172     result = 0;
173 end
174
175 function result = integratedXMiddle(y, x1, x2, dycov, invcov)
176 % x1 から x2 までの x に関する積分結果
177 global S2
178 % 次形式の係数 2
179 a = invcov(1,1);
180 b = invcov(1,2);
181 c = invcov(2,2);

```

```

183 detCov = det(dycov);
185
186 % の値に応じて係数を決定 y
187 if y <= -S2
188   coef = -S2;
189 elseif y >= S2
190   coef = S2;
191 else
192   coef = y;
193 end
194
195 erf_arg1 = sqrt(a./2) * (x2 + b * y ./ a);
196 erf_arg2 = sqrt(a./2) * (x1 + b * y ./ a);
197
198 % の引数が実数かつ有限な値であることを確認 erf
199 if isreal(erf_arg1) && all(isfinite(erf_arg1)) && isreal(erf_arg2) && all(
200   isfinite(erf_arg2))
201   result = coef ./ (2.*pi.*sqrt(detCov)) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
202     ((-1./a) .* (exp(-a*(x2 + b.*y./a).^2./2) - exp(-a*(x1 + b.*y./a).^2./2)) - ...
203     (b.*y./a) .* sqrt(pi./2./a) .* (erf(erf_arg1) - erf(erf_arg2)));
204 else
205   result = 0; % の引数が無効な場合、を返す erf0
206 end
207
208 end
209
210 function result = integratedXUpper(y, x2, dycov, invcov)
211   % x2 から Inf までの x に関する積分結果
212   global S2 S1
213
214   % 次形式の係数 2
215   a = invcov(1,1);
216   b = invcov(1,2);
217   c = invcov(2,2);
218
219   detCov = det(dycov);
220
221   % の値に応じて係数を決定 y
222   if y <= -S2
223     coef = -S2;
224   elseif y >= S2
225     coef = S2;
226   else
227     coef = y;
228   end
229
230   % S1 に対する積分結果
231   erf_arg = sqrt(a) * (x2 + b * y / a) / sqrt(2);
232
233   % の引数が実数かつ有限な値であることを確認 erf
234   if isreal(erf_arg) && all(isfinite(erf_arg))
235     result = S1 .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
236       sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
237       (1 - erf(erf_arg));
238   else
239     result = 0; % の引数が無効な場合、を返す erf0
240   end

```

Listing B.5: dyhouwaerrorbar_S1fix.m

```

1 clc;
2 clear all;
3
4 t = 300;
5 N = 200;
6 ens = 100;
7 mu = 0.1; % ステップサイズ
8 xi = 0; % 背景雑音
9 S1 = 1;
10 S2_values = [S1*0.3, S1*0.7, S1, S1*1.2, S1*1.6, S1*2, S1*2.4, S1*2.8]; % 任意の値を指定
    % 0.3 , 0.7 , 1.0 , 1.2 , 1.6 , 2.0 , 2.4 , 2.8
11 num_S2_values = numel(S2_values);
12 MSE_values = zeros(num_S2_values, 1); % の値を保存するベクトル MSE
13 MSE_std_p = zeros(num_S2_values, 1); % の標準偏差を保存するベクトル MSE
14 MSE_std_n = zeros(num_S2_values, 1);
15
16 tic; % ストップウォッチタイマー開始
17 for k = 1:num_S2_values
18     S2 = S2_values(k);
19     n = N * t + 1; % 更新回数
20     MSE = zeros(ens, 1); % MSE 初期化
21     for i = 1:ens
22         g = randn(N, 1); % 未知システム
23         w = zeros(N, 1); % 適応フィルタの初期化
24         x = randn(N, 1) / sqrt(N); % 行列の入力ベクトル N1
25         for j = 1:n
26             x = circshift(x, 1); % 入力ベクトルをすべてずらす
27             x(1) = randn / sqrt(N); % 値を上書きする
28             d = g.' * x; % 未知システムの出力
29             y = w.' * x; % 適応フィルタの出力
30             if d < -S1
31                 d = -S1;
32             elseif d > S1
33                 d = S1;
34             else
35                 d = d;
36             end
37
38             if y < -S2
39                 y = -S2;
40             elseif y > S2
41                 y = S2;
42             else
43                 y = y;
44             end
45             e = d - y + sqrt(xi) * randn; % 誤差の生成
46
47             if j >= n - 9
48                 MSE(i) = MSE(i) + e^2; % 最後の回分の誤差の二乗を足す 10
49             end
50
51             w = w + mu * e * x; % 適応フィルタの更新
52         end
53         MSE(i) = MSE(i) / 10; % 最後の回分の平均を計算 10
54     end
55     MSE_values(k) = median(MSE); % の中央値 MSE
56     MSE_descend = sort(MSE, 'descend'); % を降順にソート MSE
57     MSE_std_p(k) = MSE_descend(round(0.1587 * ens)); % 標準偏差の近似上位% (16)
58     MSE_std_n(k) = MSE_descend(round((1-0.1587) * ens)); % 標準偏差の近似下位% (16)
59     disp(['num2str(k * 100 / num_S2_values), '%' ]); % 進行状況の確認
60 end
61
62 % 結果の保存

```

```
63 data = [S2_values/S1; MSE_values'; MSE_std_p'; MSE_std_n'];
64 header = 'dyhouwaerrorbar';
65 fname = char([header, '(N=', num2str(N), ',S1=', num2str(S1), ',T=', num2str(t),
66 ), ',mu=', num2str(mu), ',ens=', num2str(ens), ',xi=', num2str(xi), ',
67 _errorbars).txt']);
68 Fid = fopen(fname, 'w');
69 fprintf(Fid, 'S2/S1\tMSE\tMSE_std_p\tMSE_std_n\n');
70 fprintf(Fid, '%.2f\t%.5f\t%.5f\t%.5f\n', data);
71 fclose(Fid);
72 toc; % タイマー終了
```

Listing B.6: dyhouwaMSDgakusyu.m

```

1 clc;
2 clear all;
3
4 tEnd = 100;
5
6 tic; % ストップウォッチタイマー開始
7
8 global fd2 dfd PRS RNG mu S1 S2 xi rho sgm_g
9
10 rho = 1;
11 S1 = 1;
12 S2 = 1;
13 xi = 0;
14 mu = 0.1;
15 sgm_g = sqrt(1);
16
17 PRS = 1e-7; % 数値積分の精度 RelTol,AbsTol
18 RNG = 7; % 数値積分の範囲を決めるパラメータ
19
20 fname = char(['dyhouwa1MSDg', ', ,tEnd=', num2str(tEnd), ', ,mu=', num2str(mu), ,
21 , ,S1=', num2str(S1), ', ,S2=', num2str(S2), '.txt']);
22 Fid = fopen(fname, 'w');
23 Header = '#time Q r MSD';
24 fprintf(Fid, '%s\n', Header);
25
26 % 定数で表されるサンプル平均
27 fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) * sgm_g
28 ^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
29 dfd = rho^2 * sgm_g^2 * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
30
31 options = odeset('RelTol', 1e-5, 'AbsTol', 1e-5);
32 [t, y] = ode45(@(t, y) odefun(t, y), [0 tEnd], [1e-9, 1e-9], options);
33
34 Q = y(:, 1);
35 r = y(:, 2);
36
37 [sizet, ~] = size(t);
38 MSD = zeros(sizet, 1);
39
40 for k = 1:sizet
41     fprintf('Progress: %.2f%%\n', (k / sizet) * 100);
42     dycov = rho.^2.* [sgm_g^2 r(k); r(k) Q(k)];
43     invcov = inv(dycov);
44
45     % 積分計算
46     MSD(k) = calculateMSD(Q(k), r(k));
47
48     fprintf(Fid, '%g %g %g %g\n', t(k), Q(k), r(k), MSD(k));
49 end
50
51 fclose(Fid);
52 toc;
53
54 function dydt = odefun(t, y)
55     global fd2 dfd mu S1 S2 xi rho sgm_g
56     dydt = zeros(2, 1);
57     Q = max(y(1), 1e-12); % が負値にならないように制限 Q
58     r = max(y(2), 1e-12); % が負値にならないように制限 r
59
60     erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
61     erf_arg2 = S1 / sqrt(2 * rho^2 * sgm_g^2);
62
63     % が実数で有限な値かをチェック erf_arg
64     if isreal(erf_arg1) && isfinite(erf_arg1)

```

```

65      fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
66      Q - S2^2) * erf(erf_arg1);
67  else
68      % が無効な場合は、代わりに erf_arg0
69      fy2 = 0; % の代わりに erf0
70  end
71 if isreal(erf_arg1) && isfinite(erf_arg1)
72     dfy = rho^2 * r * erf(erf_arg1);
73 else
74     dfy = 0; % の代わりに erf0
75 end
76
77 if isreal(erf_arg2) && isfinite(erf_arg2)
78     yfd = rho^2 * r * erf(erf_arg2);
79 else
80     yfd = 0; % の代わりに erf0
81 end
82
83 if isreal(erf_arg1) && isfinite(erf_arg1)
84     yfy = rho^2 * Q * erf(erf_arg1);
85 else
86     yfy = 0; % の代わりに erf0
87 end
88
89 dycov = rho.^2 .* [sgm_g.^2 r; r Q];
90
91 invcov = inv(dycov);
92
93 fdfy = calculateIntegral(Q, dycov, invcov);
94
95 dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
96 dydt(2) = mu * (dfd - dfy);
97 end
98
99 function result = calculateMSD(Q, r)
100    global sgm_g
101    result = sgm_g^2-2*r+Q;
102 end
103
104 function result = calculateIntegral(Q, dycov, invcov)
105    global S1 S2 PRS RNG rho
106
107    % y に関する積分範囲
108    yr = RNG * sqrt(rho^2) * max(sqrt(Q), 1e-9);
109
110    % y に関して積分済みの関数を定義 (x-Inf to -S1 の範囲)
111    result1 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -yr, -S2, '
112        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
113    result2 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -S2, S2, '
114        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
115    result3 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), S2, yr, '
116        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
117
118    % -S1 to S1 の範囲
119    result4 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -yr, -S2, '
120        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
121    result5 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -S2, S2, '
122        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
123    result6 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), S2, yr, '
124        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
125
126    % S1 to Inf の範囲
127    result7 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -yr, -S2, '
128        RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
129
130    result = result1 + result2 + result3 + result4 + result5 + result6 + result7;
131
132    % 結果を返す
133    result = result / (S1 - (-S1));
134
135 end

```

```

    RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
125 result8 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -S2, S2, ,
    RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
126 result9 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), S2, yr, ,
    RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
127 result = result1 + result2 + result3 + result4 + result5 + result6 +
    result7 + result8 + result9;
128
129 end
130
131 function result = integratedXLower(y, x1, dycov, invcov)
132 % -Inf から x1 までの x に関する積分結果
133 global S2 S1
134
135 % 次形式の係数 2
136 a = invcov(1,1);
137 b = invcov(1,2);
138 c = invcov(2,2);
139
140 detCov = det(dycov);
141
142 % の値に応じて係数を決定 y
143 if y <= -S2
144     coef = -S2;
145 elseif y >= S2
146     coef = S2;
147 else
148     coef = y;
149 end
150
151 % -S1 に対する積分結果
152 erf_arg = sqrt(a) * (x1 + b * y / a) / sqrt(2);
153 if isreal(erf_arg) && all(isfinite(erf_arg)) % 実数かつ有限な値であることを確認
154     result = (-S1) .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
155         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
156         (1 + erf(erf_arg));
157 else
158     % の引数が無効な場合 erf
159     result = 0;
160 end
161
162 end
163
164 function result = integratedXMiddle(y, x1, x2, dycov, invcov)
165 % x1 から x2 までの x に関する積分結果
166 global S2
167
168 % 次形式の係数 2
169 a = invcov(1,1);
170 b = invcov(1,2);
171 c = invcov(2,2);
172
173 detCov = det(dycov);
174
175 % の値に応じて係数を決定 y
176 if y <= -S2
177     coef = -S2;
178 elseif y >= S2
179     coef = S2;
180 else
181     coef = y;
182 end
183
184 erf_arg1 = sqrt(a./2) * (x2 + b * y ./ a);
185 erf_arg2 = sqrt(a./2) * (x1 + b * y ./ a);
186
% の引数が実数かつ有限な値であることを確認 erf

```

```

188 if isreal(erf_arg1) && all(isfinite(erf_arg1)) && isreal(erf_arg2) && all(
189     isfinite(erf_arg2))
190     result = coef ./ (2.*pi.*sqrt(detCov)) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
191         ((-1./a) .* (exp(-a*(x2 + b.*y./a).^2./2) - exp(-a*(x1 + b.*y./a).^2./2)) - ...
192             (b.*y./a) .* sqrt(pi./2./a) .* (erf(erf_arg1) - erf(erf_arg2)));
193 else
194     result = 0; % の引数が無効な場合、を返す erf0
195 end
196
197 function result = integratedXUpper(y, x2, dycov, invcov)
198     % x2 から Inf までの x に関する積分結果
199     global S2 S1
200
201     % 次形式の係数 2
202     a = invcov(1,1);
203     b = invcov(1,2);
204     c = invcov(2,2);
205
206     detCov = det(dycov);
207
208     % の値に応じて係数を決定 y
209     if y <= -S2
210         coef = -S2;
211     elseif y >= S2
212         coef = S2;
213     else
214         coef = y;
215     end
216
217     % S1 に対する積分結果
218     erf_arg = sqrt(a) * (x2 + b * y / a) / sqrt(2);
219
220     % の引数が実数かつ有限な値であることを確認 erf
221     if isreal(erf_arg) && all(isfinite(erf_arg))
222         result = S1 .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
223             sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
224                 (1 - erf(erf_arg));
225     else
226         result = 0; % の引数が無効な場合、を返す erf0
227     end
228 end
229
```

Listing B.7: dyhouwaMSDs imu.m

```

1 clc ;
2 clear all ;
3
4 t = 100 ;
5 N = 200 ;
6 ens = 1000 ;
7 a = 1 ;
8 xi = 0 ; % 背景雑音
9 n = N*t+1; % 更新回数
10 S1 = 1; % 未知システム飽和値
11 S2 = 1; % 適応フィルタ飽和値
12 mu = 1 ; % ステップサイズ
13 MSD = zeros(a*t+1 , 1); % MSD 初期化
14
15
16 tic; %ストップウォッチタイマー開始
17
18 for i = 1 : ens
19
20     g = randn(N,1); % 未知システム
21     w = zeros(N,1); % 適応フィルタの初期化
22     x = randn(N,1)/sqrt(N); % N 行 1 列の入力ベクトル
23
24     for j = 1 : n
25
26         x = circshift(x,1); % 入力ベクトルをすべてずらす
27         x(1) = randn/sqrt(N); % 値を上書きする
28
29         d = g.*x; % 未知システムの出力
30         y = w.*x; % 適応フィルタの出力
31
32         if d < -S1
33             d = -S1;
34         elseif d > S1
35             d = S1;
36         else
37             d = d;
38         end
39
40         if y < -S2
41             y = -S2;
42         elseif y > S2
43             y = S2;
44         else
45             y = y;
46         end
47
48
49         gw = g-w; % 偏差の生成
50         e = d - y + sqrt(xi) * randn;
51
52         if(rem(j,N/a)==1) % の計算 MSD
53             MSD((j-1)/(N/a)+1) = MSD((j-1)/(N/a)+1) + ((gw.*gw)/N)/ens;
54         end
55
56         w = w + mu*e*x; % 適応フィルタの更新
57
58     end
59
60     disp([num2str(i*100/ens),'%']) % 進行状況の確認
61
62 end
63
64 data = [(0:1/a:t);MSD'];
65 header = 'MSD_Simulation';

```

```
66 fname = char([header, '(N=',num2str(N), ',T=',num2str(t), ',S1=',num2str(S1), ',  
67 S2=',num2str(S2), ',mu=',num2str(mu), ',ens=',num2str(ens), ',xi=',num2str(  
68 xi),').txt']);  
69 Fid = fopen(fname,'w');  
70 S = '#time MSD'; % 記録ファイルの一行目  
71 fprintf(Fid,'%s\n',S);  
72 fprintf(Fid,'%f\n',data);  
73 fclose(Fid);  
74 toc; % タイマー終了
```

Listing B.8: dySTVMSD.m

```

1 clc;
2 clear all;
3
4 tEnd = 300;
5
6 tic; % ストップウォッチタイマー開始
7
8 global fd2 dfd PRS RNG mu S1 S2 xi rho sgm_g
9
10 rho = 1;
11 S1 = 1;
12 S2loop = 0:S1/100:S1*3; %初期値, 刻み幅, 終了値
13 xi = 1;
14 mu = 0.1;
15 sgm_g = sqrt(1);
16
17 PRS = 1e-7; % 数値積分の精度 RelTol, AbsTol
18 RNG = 7; % 数値積分の範囲を決めるパラメータ
19
20 fname = char(['dyhouwa1MSDSTV', ',tEnd=', num2str(tEnd), ',mu=', num2str(mu),
   ',S1=', num2str(S1), ',S2=', num2str(S2), ',xi=', num2str(xi), '.txt']);
21 Fid = fopen(fname, 'w');
22 Header = '#S2/S1' MSD';
23 fprintf(Fid, '%s\n', Header);
24
25 % 定数で表されるサンプル平均
26 fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) * sgm_g
   ^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
27 dfd = rho^2 * sgm_g^2 * erf(S1 / sqrt(2 * rho^2 * sgm_g^2));
28
29 for i = 1:length(S2loop)
30   S2 = S2loop(i);
31   fprintf('Progress for S2 = %.2f : %.2f%\n', S2, (i / length(S2loop)) *
   100);
32
33 % ソルバーで時間発展を計算 ODE
34 options = odeset('RelTol', 1e-6, 'AbsTol', 1e-6); % 精度の調整
35 [t, y] = ode45(@(t, y) odefun(t, y), [0 tEnd], [1e-9, 1e-9], options);
36
37 % 最終時刻の Q, を取得 r
38 Q_final = y(end, 1);
39 r_final = y(end, 2);
40
41 % 最終時刻での共分散行列とその逆行列を計算
42 dycov_final = rho^2 * [sgm_g^2, r_final; r_final, Q_final];
43 invcov_final = inv(dycov_final);
44
45 % 最終時刻でのを計算 MSE
46 MSD_final = calculateMSD(Q_final, r_final);
47
48 fprintf(Fid, '%g %g\n', S2/S1, MSD_final);
49 end
50
51 fclose(Fid);
52 toc;
53
54 function dydt = odefun(t, y)
55   global fd2 dfd mu S1 S2 xi rho sgm_g
56   dydt = zeros(2, 1);
57   Q = max(y(1), 1e-12); % が負値にならないように制限 Q
58   r = max(y(2), 1e-12); % が負値にならないように制限 r
59
60   erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
61   erf_arg2 = S1 / sqrt(2 * rho^2 * sgm_g^2);
62

```

```

63 % が実数で有限な値かをチェック erf_arg
64 if isreal(erf_arg1) && isfinite(erf_arg1)
65     fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
66 else
67     % が無効な場合は、代わりに erf_arg0
68     fy2 = 0; % の代わりに erf0
69 end
70
71 if isreal(erf_arg1) && isfinite(erf_arg1)
72     dfy = rho^2 * r * erf(erf_arg1);
73 else
74     dfy = 0; % の代わりに erf0
75 end
76
77 if isreal(erf_arg2) && isfinite(erf_arg2)
78     yfd = rho^2 * r * erf(erf_arg2);
79 else
80     yfd = 0; % の代わりに erf0
81 end
82
83 if isreal(erf_arg1) && isfinite(erf_arg1)
84     yfy = rho^2 * Q * erf(erf_arg1);
85 else
86     yfy = 0; % の代わりに erf0
87 end
88
89 dycov = rho.^2 .* [sgm_g.^2 r; r Q];
90
91
92 invcov = inv(dycov);
93
94
95 fdfy = calculateIntegral(Q, dycov, invcov);
96
97 dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
98 dydt(2) = mu * (dfd - dfy);
99
100 end
101
102 function result = calculateMSD(Q, r)
103     global sgm_g
104     result = sgm_g^2-2*r+Q;
105 end
106
107 function result = calculateIntegral(Q, dycov, invcov)
108     global S1 S2 PRS RNG rho
109
110     % y に関する積分範囲
111     yr = RNG * sqrt(rho^2) * max(sqrt(Q), 1e-9);
112
113     % 関して積分済みの関数を定義 (x-Inf to -S1 の範囲)
114     result1 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -yr, -S2, ,
115         'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
116     result2 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -S2, S2, ,
117         'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
118     result3 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), S2, yr, ,
119         'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
120
121     % -S1 to S1 の範囲
122     result4 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -yr, -
123         S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
124     result5 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -S2,
125         S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
126     result6 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), S2,
127         yr, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589

```

```

123    %  $S_1$  to  $Inf$  の範囲
124    result7 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -yr, -S2, ,
125        'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
126    result8 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -S2, S2, ,
127        'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
128    result9 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), S2, yr, ,
129        'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
130    result = result1 + result2 + result3 + result4 + result5 + result6 +
131        result7 + result8 + result9;
132 end
133
134 function result = integratedXLower(y, x1, dycov, invcov)
135 %  $-Inf$  から  $x_1$  までの  $x$  に関する積分結果
136 global S2 S1
137
138 % 次形式の係数  $\mathcal{Z}$ 
139 a = invcov(1,1);
140 b = invcov(1,2);
141 c = invcov(2,2);
142
143 detCov = det(dycov);
144
145 % の値に応じて係数を決定  $y$ 
146 if y <= -S2
147     coef = -S2;
148 elseif y >= S2
149     coef = S2;
150 else
151     coef = y;
152 end
153
154 %  $-S_1$  に対する積分結果
155 erf_arg = sqrt(a) * (x1 + b * y / a) / sqrt(2);
156 if isreal(erf_arg) && all(isfinite(erf_arg)) % 実数かつ有限な値であることを確認
157     result = (-S1) .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
158         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
159         (1 + erf(erf_arg));
160 else
161     % の引数が無効な場合 erf
162     result = 0;
163 end
164
165 function result = integratedXMiddle(y, x1, x2, dycov, invcov)
166 %  $x_1$  から  $x_2$  までの  $x$  に関する積分結果
167 global S2
168
169 % 次形式の係数  $\mathcal{Z}$ 
170 a = invcov(1,1);
171 b = invcov(1,2);
172 c = invcov(2,2);
173
174 detCov = det(dycov);
175
176 % の値に応じて係数を決定  $y$ 
177 if y <= -S2
178     coef = -S2;
179 elseif y >= S2
180     coef = S2;
181 else
182     coef = y;
183 end
184
185 erf_arg1 = sqrt(a./2) * (x2 + b * y ./ a);

```

```

185     erf_arg2 = sqrt(a./2) * (x1 + b * y ./ a);
186
187 % の引数が実数かつ有限な値であることを確認 erf
188 if isreal(erf_arg1) && all(isfinite(erf_arg1)) && isreal(erf_arg2) && all(
189     isfinite(erf_arg2))
190     result = coef ./ (2.*pi.*sqrt(detCov)) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
191         ((-1./a) .* (exp(-a*(x2 + b.*y./a).^2./2) - exp(-a.*(x1 + b.*y./a).^2./2)) - ...
192             (b.*y./a) .* sqrt(pi./2./a) .* (erf(erf_arg1) - erf(erf_arg2)));
193 else
194     result = 0; % の引数が無効な場合、を返す erf0
195 end
196
197 function result = integratedXUpper(y, x2, dycov, invcov)
198 % x2 から Inf までの x に関する積分結果
199 global S2 S1
200
201 % 次形式の係数 Z
202 a = invcov(1,1);
203 b = invcov(1,2);
204 c = invcov(2,2);
205
206 detCov = det(dycov);
207
208 % の値に応じて係数を決定 y
209 if y <= -S2
210     coef = -S2;
211 elseif y >= S2
212     coef = S2;
213 else
214     coef = y;
215 end
216
217 % S1 に対する積分結果
218 erf_arg = sqrt(a) * (x2 + b * y / a) / sqrt(2);
219
220 % の引数が実数かつ有限な値であることを確認 erf
221 if isreal(erf_arg) && all(isfinite(erf_arg))
222     result = S1 .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
223         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
224             (1 - erf(erf_arg));
225 else
226     result = 0; % の引数が無効な場合、を返す erf0
227 end
228
229 end

```

Listing B.9: MSDerrorbar_S1fix.m

```

1 clc;
2 clear all;
3
4 t = 300;
5 N = 200;
6 ens = 100;
7 mu = 0.1; % ステップサイズ
8 xi = 0; % 背景雑音
9 S1 = 1;
10 S2_values = [S1*0.3, S1*0.7, S1, S1*1.2, S1*1.6, S1*2, S1*2.4, S1*2.8]; % 任意の値を指定
    S0.3 , 0.7 , 1.0 , 1.2 , 1.6 , 2.0 , 2.4 , 2.8
11 num_S2_values = numel(S2_values);
12 MSD_values = zeros(num_S2_values, 1); % の値を保存するベクトル MSD
13 MSD_std_p = zeros(num_S2_values, 1); % の標準偏差を保存するベクトル MSD
14 MSD_std_n = zeros(num_S2_values, 1);
15
16 tic; % ストップウォッチタイマー開始
17 for k = 1:num_S2_values
18     S2 = S2_values(k);
19     n = N * t + 1; % 更新回数
20     MSD = zeros(ens, 1); % MSD 初期化
21     for i = 1:ens
22         g = randn(N, 1); % 未知システム
23         w = zeros(N, 1); % 適応フィルタの初期化
24         x = randn(N, 1) / sqrt(N); % 行列の入力ベクトル N1
25         for j = 1:n
26             x = circshift(x, 1); % 入力ベクトルをすべてずらす
27             x(1) = randn / sqrt(N); % 値を上書きする
28             d = g.' * x; % 未知システムの出力
29             y = w.' * x; % 適応フィルタの出力
30             if d < -S1
31                 d = -S1;
32             elseif d > S1
33                 d = S1;
34             else
35                 d = d;
36             end
37
38             if y < -S2
39                 y = -S2;
40             elseif y > S2
41                 y = S2;
42             else
43                 y = y;
44             end
45             gw = g-w; % 偏差の生成
46             e = d - y + sqrt(xi) * randn; % 誤差の生成
47
48             if j >= n - 9
49                 MSD(i) = MSD(i) + (gw.'*gw)/N; % 最後の回分の偏差 10
50             end
51
52             w = w + mu * e * x; % 適応フィルタの更新
53         end
54         MSD(i) = MSD(i) / 10; % 最後の回分の平均を計算 10
55     end
56     MSD_values(k) = median(MSD); % の中央値 MSD
57     MSD_descend = sort(MSD, 'descend'); % を降順にソート MSD
58     MSD_std_p(k) = MSD_descend(round(0.1587 * ens)); % 標準偏差の近似上位% (16)
59     MSD_std_n(k) = MSD_descend(round((1-0.1587) * ens)); % 標準偏差の近似下位% (16)
60     disp([num2str(k * 100 / num_S2_values), '%']); % 進行状況の確認
61 end

```

```
62 % 結果の保存
63 data = [S2_values/S1; MSD_values'; MSD_std_p'; MSD_std_n'];
64 header = 'dyMSDerrorbar';
65 fname = char([header, '(N=', num2str(N), ',S1=', num2str(S1), ',T=', num2str(t
    ), ',mu=', num2str(mu), ',ens=', num2str(ens), ',xi=', num2str(xi), '
    _errorbars).txt']);
66 Fid = fopen(fname, 'w');
67 fprintf(Fid, 'S2/S1\tMSD\tMSD_std_p\tMSD_std_n\n');
68 fprintf(Fid, '%.2f\t%.5f\t%.5f\t%.5f\n', data);
69 fclose(Fid);
70 toc; % タイマー終了
```

Listing B.10: STVMSEheatmap_simu.m

```

1 clc; clear; close all;
2
3 % パラメータ設定
4 t = 5000; % シミュレーション時間
5 N = 200; % 次元数
6 ens = 100; % 試行回数
7 mu = 0.1; % ステップサイズ
8 xi = 0.1; % 背景雑音
9 n = N * t + 1; % 更新回数
10
11 % 飽和パラメータの範囲
12 S1_vals = linspace(0, 3, 31); % 初期値, 終了値, 分割数
13 S2_vals = linspace(0, 3, 31); % 初期値, 終了値, 分割数
14 MSE_values = zeros(length(S1_vals), length(S2_vals)); % MSE を格納する行列
15
16 total_start_time = tic; % 全体の計算時間測定開始
17
18
19 for idx1 = 1:length(S1_vals)
20     S1 = S1_vals(idx1);
21     for idx2 = 1:length(S2_vals)
22         S2 = S2_vals(idx2);
23         MSE_total = 0; % MSE の合計値（後で平均を取る）
24
25         loop_start_time = tic; % ループごとの時間測定開始
26
27         for i = 1:ens
28             g = randn(N, 1); % 未知システム
29             w = zeros(N, 1); % 適応フィルタの初期化
30             x = randn(N, 1) / sqrt(N); % 入力ベクトル
31
32             for j = 1:n
33                 x = circshift(x, 1); % 入力ベクトルをすべてずらす
34                 x(1) = randn / sqrt(N); % 値を上書きする
35                 d = g.' * x; % 未知システムの出力
36                 y = w.' * x; % 適応フィルタの出力
37
38                 % 飽和非線形性の適用
39                 d = max(min(d, S1), -S1);
40                 y = max(min(y, S2), -S2);
41
42                 e = d - y + sqrt(xi) * randn; % 誤差の計算
43                 w = w + mu * e * x; % 適応フィルタの更新
44
45                 % 最終時刻のMSE
46                 if j == n
47                     MSE_total = MSE_total + e^2;
48                 end
49             end
50         end
51
52         % 平均MSE を計算して保存
53         MSE_values(idx1, idx2) = MSE_total / ens;
54
55         % 経過時間の取得
56         elapsed_time = toc(loop_start_time);
57         fprintf('S1=%f, S2=%f, MSE=%f\n, Time=%f sec\n', S1,
58               S2, MSE_values(idx1, idx2), elapsed_time);
59     end
60
61     % データの保存
62     filename = sprintf('mse_data_xi=%.2f.txt', xi);

```

```
63 fid = fopen(filename, 'w');
64 fprintf(fid, '#\u21d3S1\u21d3S2\u21d3MSE\n');
65
66 for idx1 = 1:length(S1_vals)
67     for idx2 = 1:length(S2_vals)
68         fprintf(fid, '%.2f\u21d3%.2f\u21d3%.5f\n', S1_vals(idx1), S2_vals(idx2),
69                  MSE_values(idx1, idx2));
70     end
71     fprintf(fid, "\n"); % S1 のブロックが終わるごとに空行を挿入
72 end
73 fclose(fid);
74
75 total_elapsed_time = toc(total_start_time);
76 fprintf('データ保存完了:\u21d3%s\n', filename);
77 fprintf('全体の計算時間:\u21d3%.2f\u21d3sec\n', total_elapsed_time);
```

Listing B.11: STVMSEheatmap_Theory.m

```

66 % が実数で有限な値かをチェック erf_arg
67 if isreal(erf_arg1) && isfinite(erf_arg1)
68   fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
69   Q - S2^2) * erf(erf_arg1);
70   dfy = rho^2 * r * erf(erf_arg1);
71   yfy = rho^2 * Q * erf(erf_arg1);
72 else
73   fy2 = 0; % の代わりに erf0
74   dfy = 0;
75   yfy = 0;
76 end
77
78 if isreal(erf_arg2) && isfinite(erf_arg2)
79   yfd = rho^2 * r * erf(erf_arg2);
80   fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) *
81   sgm_g^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(erf_arg2);
82   dfd = rho^2 * sgm_g^2 * erf(erf_arg2);
83 else
84   yfd = 0; % の代わりに erf0
85   fd2 = 0;
86   dfd = 0;
87 end
88 dycov = rho.^2 .* [sgm_g.^2 r; r Q];
89
90 invcov = inv(dycov);
91
92 fdfy = calculateIntegral(Q, dycov, invcov);
93
94 dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
95 dydt(2) = mu * (dfd - dfy);
96 end
97
98 function result = calculateMSE(Q, r, dycov, invcov)
99 global S2 rho xi S1 sgm_g
100
101 erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
102 erf_arg2 = S1 / sqrt(2 * rho^2 * sgm_g^2);
103
104 if isreal(erf_arg1) && isfinite(erf_arg1)
105   fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
106   Q - S2^2) * erf(erf_arg1);
107 else
108   % が無効な場合は、代わりに erf_arg0
109   fy2 = 0; % の代わりに erf0
110 end
111
112 if isreal(erf_arg2) && isfinite(erf_arg2)
113   fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) *
114   sgm_g^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(erf_arg2);
115 else
116   fd2 = 0;
117 end
118
119 fdfy = calculateIntegral(Q, dycov, invcov);
120
121 result = fd2 + fy2 - 2 * fdfy + xi;
122
123 if isnan(result) || result < 0
124   result = 0;
125 end
126 end
127
128 function result = calculateIntegral(Q, dycov, invcov)

```

```

129 global S1 S2 PRS RNG rho
130
131 %  $y$  に関する積分範囲
132 yr = RNG * sqrt(rho^2) * max(sqrt(Q), 1e-9);
133
134 % に関して積分済みの関数を定義 ( $x - Inf$  to  $-S1$  の範囲)
135 result1 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -yr, -S2, ,
136     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
137 result2 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -S2, S2, ,
138     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
139 result3 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), S2, yr, ,
140     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
141
142 %  $-S1$  to  $S1$  の範囲
143 result4 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -yr, -
144     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
145 result5 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -S2,
146     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
147 result6 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), S2,
148     yr, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
149
150 %  $S1$  to  $Inf$  の範囲
151 result7 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -yr, -S2, ,
152     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
153 result8 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -S2, S2, ,
154     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
155 result9 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), S2, yr, ,
156     'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
157
158 result = result1 + result2 + result3 + result4 + result5 + result6 +
159     result7 + result8 + result9;
160 end
161
162 function result = integratedXLower(y, x1, dycov, invcov)
163 %  $-Inf$  から  $x1$  までの  $x$  に関する積分結果
164 global S2 S1
165
166 % 次形式の係数  $\mathcal{Z}$ 
167 a = invcov(1,1);
168 b = invcov(1,2);
169 c = invcov(2,2);
170
171 detCov = det(dycov);
172
173 % の値に応じて係数を決定  $y$ 
174 if y <= -S2
175     coef = -S2;
176 elseif y >= S2
177     coef = S2;
178 else
179     coef = y;
180 end
181
182 %  $-S1$  に対する積分結果
183 erf_arg = sqrt(a) * (x1 + b * y / a) / sqrt(2);
184 if isreal(erf_arg) && all(isfinite(erf_arg)) % 実数かつ有限な値であることを確認
185     result = (-S1) .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
186         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
187         (1 + erf(erf_arg));
188 else
189     % の引数が無効な場合 erf
190     result = 0;
191 end
192
193 end
194
195 end

```

```

185 function result = integratedXMiddle(y, x1, x2, dycov, invcov)
186 %  $x_1$  から  $x_2$  までの  $x$  に関する積分結果
187 global S2
188
189 % 次形式の係数  $\mathcal{Z}$ 
190 a = invcov(1,1);
191 b = invcov(1,2);
192 c = invcov(2,2);
193
194 detCov = det(dycov);
195
196 % の値に応じて係数を決定  $y$ 
197 if y <= -S2
198   coef = -S2;
199 elseif y >= S2
200   coef = S2;
201 else
202   coef = y;
203 end
204
205 erf_arg1 = sqrt(a./2) * (x2 + b * y ./ a);
206 erf_arg2 = sqrt(a./2) * (x1 + b * y ./ a);
207
208 % の引数が実数かつ有限な値であることを確認  $erf$ 
209 if isreal(erf_arg1) && all(isfinite(erf_arg1)) && isreal(erf_arg2) && all(
210   isfinite(erf_arg2))
211   result = coef ./ (2.*pi.*sqrt(detCov)) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
212     ((-1./a) .* (exp(-a*(x2 + b.*y./a).^2./2) - exp(-a*(x1 + b.*y./a).^2./2)) - ...
213     (b.*y./a) .* sqrt(pi./2./a) .* (erf(erf_arg1) - erf(erf_arg2)));
214 else
215   result = 0; % の引数が無効な場合、を返す  $erf0$ 
216 end
217
218
219 function result = integratedXUpper(y, x2, dycov, invcov)
220 %  $x_2$  から  $Inf$  までの  $x$  に関する積分結果
221 global S2 S1
222
223 % 次形式の係数  $\mathcal{Z}$ 
224 a = invcov(1,1);
225 b = invcov(1,2);
226 c = invcov(2,2);
227
228 detCov = det(dycov);
229
230 % の値に応じて係数を決定  $y$ 
231 if y <= -S2
232   coef = -S2;
233 elseif y >= S2
234   coef = S2;
235 else
236   coef = y;
237 end
238
239 %  $S_1$  に対する積分結果
240 erf_arg = sqrt(a) * (x2 + b * y / a) / sqrt(2);
241
242 % の引数が実数かつ有限な値であることを確認  $erf$ 
243 if isreal(erf_arg) && all(isfinite(erf_arg))
244   result = S1 .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
245     sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
246     (1 - erf(erf_arg));
247 else
248   result = 0; % の引数が無効な場合、を返す  $erf0$ 
249 end

```

250 **end**

Listing B.12: STVMSDheatmap_simu.m

```

1 clc ;
2 clear all ;
3
4 t = 300 ;
5 N = 200 ;
6 ens = 100 ;
7 xi = 0 ; % 背景雑音
8 n = N*t+1; % 更新回数
9 mu = 0.1 ; % ステップサイズ
10 S1_vals = linspace(0, 3, 31); % 初期値, 終了値, 分割数
11 S2_vals = linspace(0, 3, 31); % 初期値, 終了値, 分割数
12 MSD_values = zeros(length(S1_vals), length(S2_vals)); % MSE を格納する行列
13
14 total_start_time = tic; % 全体の計算時間測定開始
15
16
17 for idx1 = 1:length(S1_vals)
18     S1 = S1_vals(idx1);
19     for idx2 = 1:length(S2_vals)
20         S2 = S2_vals(idx2);
21         MSD_total = 0; % MSE の合計値 (後で平均を取る)
22
23         loop_start_time = tic; % ループごとの時間測定開始
24
25         for i = 1:ens
26             g = randn(N, 1); % 未知システム
27             w = zeros(N, 1); % 適応フィルタの初期化
28             x = randn(N, 1) / sqrt(N); % 入力ベクトル
29
30             for j = 1:n
31                 x = circshift(x, 1); % 入力ベクトルをすべてずらす
32                 x(1) = randn / sqrt(N); % 値を上書きする
33                 d = g.' * x; % 未知システムの出力
34                 y = w.' * x; % 適応フィルタの出力
35
36                 % 飽和非線形性の適用
37                 d = max(min(d, S1), -S1);
38                 y = max(min(y, S2), -S2);
39
40                 gw = g-w; % 偏差の生成
41                 e = d - y + sqrt(xi) * randn; % 誤差の計算
42
43                 if j == n
44                     MSD_total = MSD_total + (gw.*gw)/N;
45                 end
46
47                 w = w + mu*e*x; % 適応フィルタの更新
48
49             end
50         end
51
52         % 平均MSD を計算して保存
53         MSD_values(idx1, idx2) = MSD_total / ens;
54
55         % 経過時間の取得
56         elapsed_time = toc(loop_start_time);
57         fprintf('S1=% .2f, S2=% .2f, MSD=% .5f\n, Time=% .2f sec\n', S1,
58               S2, MSD_values(idx1, idx2), elapsed_time);
59
60     end
61
62     % データの保存
63     filename = sprintf('msd_heatmapdatasimu_xi=% .2f, t=% .2f.txt', xi, t);

```

```
64 fid = fopen(filename, 'w');
65 fprintf(fid, '#\u21d3S1\u21d3S2\u21d3MSD\n');
66
67 for idx1 = 1:length(S1_vals)
68     for idx2 = 1:length(S2_vals)
69         fprintf(fid, '%.2f\u21d3%.2f\u21d3%.5f\n', S1_vals(idx1), S2_vals(idx2),
70                  MSD_values(idx1, idx2));
71     end
72     fprintf(fid, "\n"); % S1 のブロックが終わるごとに空行を挿入
73 end
74 fclose(fid);
75
76 total_elapsed_time = toc(total_start_time);
77 fprintf('データ保存完了:\u21d3%s\n', filename);
78 fprintf('全体の計算時間:\u21d3%.2f\u21d3sec\n', total_elapsed_time);
```

Listing B.13: STVMSDheatmap_Theory.m

```

1 clc;
2 clear all;
3
4 tEnd = 300;
5
6 tic; % ストップウォッチタイマー開始
7
8 global PRS RNG mu S1 S2 xi rho sgm_g
9
10 rho = 1;
11 S1loop = 0:0.1:3;
12 S2loop = 0:0.1:3;
13 xi = 1;
14 mu = 0.1;
15 sgm_g = sqrt(1);
16
17 PRS = 1e-7; % 数値積分の精度 RelTol, AbsTol
18 RNG = 7; % 数値積分の範囲を決めるパラメータ
19
20 fname = char(['dyhouwaSTVMSDheatmap', ',xi=', num2str(xi), '.txt']);
21 Fid = fopen(fname, 'w');
22 Header = '#S1' S1loop 'S2' S2loop 'MSD';
23 fprintf(Fid, '%s\n', Header);
24
25 for i = 1:length(S1loop)
26   S1 = S1loop(i);
27
28 for j = 1:length(S2loop)
29   S2 = S2loop(j);
30   fprintf('Progress: S1=% .2f, S2=% .2f\n', S1, S2);
31
32 % ソルバーで時間発展を計算 ODE
33 options = odeset('RelTol', 1e-6, 'AbsTol', 1e-6); % 精度の調整
34 [t, y] = ode45(@(t, y) odefun(t, y), [0 tEnd], [1e-9, 1e-9], options);
35
36 % 最終時刻の Q, を取得 r
37 Q_final = y(end, 1);
38 r_final = y(end, 2);
39
40 % 最終時刻での共分散行列とその逆行列を計算
41 dycov_final = rho^2 * [sgm_g^2, r_final; r_final, Q_final];
42 invcov_final = inv(dycov_final);
43
44 % 最終時刻でのを計算 MSE
45 MSD_final = calculateMSD(Q_final, r_final);
46
47 % 結果をファイルに書き込み
48 fprintf(Fid, '%g %g %g\n', S1, S2, MSD_final);
49 end
50 fprintf(Fid, '\n');
51 end
52
53 fclose(Fid);
54 toc;
55
56 function dydt = odefun(t, y)
57   global mu S1 S2 xi rho sgm_g
58   dydt = zeros(2, 1);
59   Q = max(y(1), 1e-12); % が負値にならないように制限 Q
60   r = max(y(2), 1e-12); % が負値にならないように制限 r
61
62 erf_arg1 = S2 / sqrt(2 * rho^2 * Q);
63 erf_arg2 = S1 / sqrt(2 * rho^2 * sgm_g^2);
64
65 % が実数で有限な値かをチェック erf_arg

```

```

66 if isreal(erf_arg1) && isfinite(erf_arg1)
67     fy2 = S2^2 - S2 * sqrt(2 * rho^2 * Q / pi) * exp(-S2^2 / (2 * rho^2 * Q)) + (rho^2 *
68         Q - S2^2) * erf(erf_arg1);
69     dfy = rho^2 * r * erf(erf_arg1);
70 else
71     fy2 = 0; % の代わりに erf0
72     dfy = 0;
73     yfy = 0;
74 end
75
76
77 if isreal(erf_arg2) && isfinite(erf_arg2)
78     yfd = rho^2 * r * erf(erf_arg2);
79     fd2 = S1^2 - S1 * sqrt(2 * rho^2 * sgm_g^2 / pi) * exp(-(S1^2) / (2 * (rho^2) *
80         sgm_g^2)) + (rho^2 * sgm_g^2 - S1^2) * erf(erf_arg2);
81     dfd = rho^2 * sgm_g^2 * erf(erf_arg2);
82 else
83     yfd = 0; % の代わりに erf0
84     fd2 = 0;
85     dfd = 0;
86 end
87 dycov = rho.^2 .* [sgm_g.^2 r; r Q];
88
89 invcov = inv(dycov);
90
91 fdfy = calculateIntegral(Q, dycov, invcov);
92
93 dydt(1) = mu^2 * rho^2 * (fd2+fy2-2*fdfy+xi) + 2*mu*(yfd-yfy);
94 dydt(2) = mu * (dfd - dfy);
95
96
97
98 function result = calculateMSD(Q, r)
99     global sgm_g
100    result = sgm_g^2-2*r+Q;
101
102
103 function result = calculateIntegral(Q, dycov, invcov)
104     global S1 S2 PRS RNG rho
105
106
107 % y に関する積分範囲
108 yr = RNG * sqrt(rho^2) * max(sqrt(Q), 1e-9);
109
110 % に関して積分済みの関数を定義 (x-Inf to -S1 の範囲)
111 result1 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -yr, -S2, ,
112     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
113 result2 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), -S2, S2, ,
114     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
115 result3 = integral(@(y) integratedXLower(y, -S1, dycov, invcov), S2, yr, ,
116     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
117
118 % -S1 to S1 の範囲
119 result4 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -yr, -
120     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
121 result5 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), -S2,
122     S2, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
123 result6 = integral(@(y) integratedXMiddle(y, -S1, S1, dycov, invcov), S2,
124     yr, 'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
125
126 % S1 to Inf の範囲
127 result7 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -yr, -S2, ,
128     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
129 result8 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), -S2, S2, ,
130     RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);

```

```

123     result9 = integral(@(y) integratedXUpper(y, S1, dycov, invcov), S2, yr, '
124         'RelTol', PRS, 'AbsTol', PRS, 'ArrayValued', true);
125     result = result1 + result2 + result3 + result4 + result5 + result6 +
126     result7 + result8 + result9;
127 end
128
129 function result = integratedXLower(y, x1, dycov, invcov)
130 % -Inf から  $x_1$  までの  $x$  に関する積分結果
131 global S2 S1
132
133 % 次形式の係数  $\mathcal{Z}$ 
134 a = invcov(1,1);
135 b = invcov(1,2);
136 c = invcov(2,2);
137
138 detCov = det(dycov);
139
140 % の値に応じて係数を決定  $y$ 
141 if y <= -S2
142     coef = -S2;
143 elseif y >= S2
144     coef = S2;
145 else
146     coef = y;
147 end
148
149 %  $-S_1$  に対する積分結果
150 erf_arg = sqrt(a) * (x1 + b * y / a) / sqrt(2);
151 if isreal(erf_arg) && all(isfinite(erf_arg)) % 実数かつ有限な値であることを確認
152     result = (-S1) .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
153         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
154         (1 + erf(erf_arg));
155 else
156     % の引数が無効な場合 erf
157     result = 0;
158 end
159
160
161 function result = integratedXMiddle(y, x1, x2, dycov, invcov)
162 %  $x_1$  から  $x_2$  までの  $x$  に関する積分結果
163 global S2
164
165 % 次形式の係数  $\mathcal{Z}$ 
166 a = invcov(1,1);
167 b = invcov(1,2);
168 c = invcov(2,2);
169
170 detCov = det(dycov);
171
172 % の値に応じて係数を決定  $y$ 
173 if y <= -S2
174     coef = -S2;
175 elseif y >= S2
176     coef = S2;
177 else
178     coef = y;
179 end
180
181 erf_arg1 = sqrt(a./2) * (x2 + b * y ./ a);
182 erf_arg2 = sqrt(a./2) * (x1 + b * y ./ a);
183
184 % の引数が実数かつ有限な値であることを確認 erf
185 if isreal(erf_arg1) && all(isfinite(erf_arg1)) && isreal(erf_arg2) && all(
186     isfinite(erf_arg2))
187     result = coef ./ (2.*pi.*sqrt(detCov)) .* exp(-(c - b.^2./a).*y.^2./2) .* ...

```

```

187      ((-1./a) .* (exp(-a*(x2 + b.*y./a).^2./2) - exp(-a.*x1 + b.*y./a).^2./2)) - ...
188      (b.*y./a) .* sqrt(pi./2./a) .* (erf(erf_arg1) - erf(erf_arg2)));
189  else
190      result = 0; % の引数が無効な場合、を返す erf0
191  end
192
193 end
194
195 function result = integratedXUpper(y, x2, dycov, invcov)
196 % x2 から Inf までの x に関する積分結果
197 global S2 S1
198
199 % 次形式の係数 2
200 a = invcov(1,1);
201 b = invcov(1,2);
202 c = invcov(2,2);
203
204 detCov = det(dycov);
205
206 % の値に応じて係数を決定 y
207 if y <= -S2
208     coef = -S2;
209 elseif y >= S2
210     coef = S2;
211 else
212     coef = y;
213 end
214
215 % S1 に対する積分結果
216 erf_arg = sqrt(a) * (x2 + b * y / a) / sqrt(2);
217
218 % の引数が実数かつ有限な値であることを確認 erf
219 if isreal(erf_arg) && all(isfinite(erf_arg))
220     result = S1 .* coef ./ (2.*pi.*sqrt(detCov)) .* ...
221         sqrt(pi./a./2) .* exp(-(c - b.^2./a).*y.^2./2) .* ...
222         (1 - erf(erf_arg));
223 else
224     result = 0; % の引数が無効な場合、を返す erf0
225 end
226 end

```

付 錄C 関連発表

学会口頭発表

- [1] 伊原 大地, 山本 直季, 本仲 君子, 三好 誠司, “二種類の非線形適応フィルタに関する統計力学的解析”, 電子情報通信学会 ソサイエティ大会（名古屋大学）, A-8-1, 2023.9.15
- [2] 伊原 大地, 山本 直季, 本仲 君子, 三好 誠司, “二種類の非線形適応フィルタの挙動に関する統計力学的解析”, 第 38 回信号処理シンポジウム（京都テルサ）, P1-18, 2023.11.7
- [3] 伊原 大地, 本仲 君子, 三好 誠司, “未知システムと適応フィルタが飽和特性を有する適応信号処理システムの解析”, 第 39 回信号処理シンポジウム（北海道大学 学術交流会館）, P2-16, 2024.12.17

二種類の非線形適応フィルタに関する統計力学的解析

STATISTICAL-MECHANICAL ANALYSIS OF TWO TYPES OF NONLINEAR ADAPTIVE FILTERS

伊原大地¹

Taichi Ihara

山本直季²

Naoki Yamamoto

本仲君子²

Kimiko Motonaka

三好誠司²

Seiji Miyoshi

関西大学大学院 理工学研究科¹

Graduate School of Science and Engineering,
Kansai University

関西大学 システム理工学部²

Faculty of Engineering Science,
Kansai University

1 まえがき

適応信号処理は広い分野で使われているが、多くの現実の適応システムではその内部に無視できない本質的な非線形性が存在する。本稿では適応フィルタの出力が非線形性を有するようなシステムのふるまいについて統計力学的手法を用いた解析を行う。非線形性としては飽和型と不感帯型の二種類を扱う。

2 解析モデル

本稿で扱うシステムのブロック図を図1に示す。

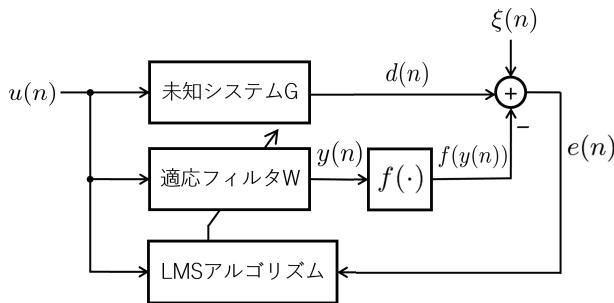


図1 システムのブロック図

未知システム G と適応フィルタ W はいずれもタップ長 N の FIR フィルタであり、その係数ベクトルはそれぞれ、 $\mathbf{g} = [g_1, g_2, \dots, g_N]^\top$, $\mathbf{w}(n) = [w_1(n), w_2(n), \dots, w_N(n)]^\top$ である。ここで n は時刻ステップである。二種類の非線形関数 $f(\cdot)$ を図2(a),(b) にそれぞれ示す。入力信号 $u(n)$ は平均 0, 分散 σ^2 の分布から独立に生成される。適応フィルタ W は LMS アルゴリズム

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)u(n) \quad (1)$$

で更新される。ここで μ はステップサイズである。

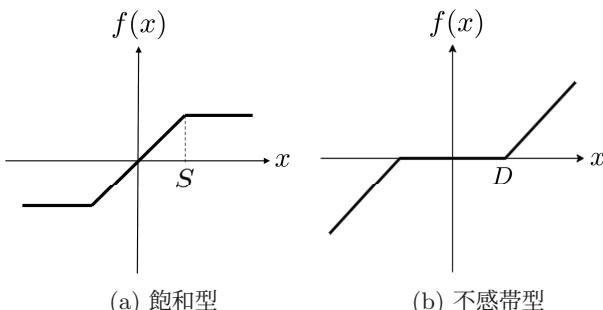


図2 非線形関数 $f(\cdot)$

3 理論

本稿では図1のシステムのふるまいについて統計力学的解析を行う[1]。その概略は以下の通りである。まず、二個の巨視的変数 $r(n) \equiv \frac{1}{N}\mathbf{g}^\top \mathbf{w}(n)$, $Q(n) \equiv \frac{1}{N}\mathbf{w}(n)^\top \mathbf{w}(n)$ を導入し、二乗平均誤差を r, Q の関数として表す。その際、 $N\sigma^2$ を一定に保ったまま $N \rightarrow \infty$ を仮定する。次に、 r, Q の動的ふるまいを記述する連立微分方程式を $N \rightarrow \infty$ において成り立つ自己平均性に基づいて決定論的に導出する。導出された連立微分方程式を数値的に解くことにより得られる r, Q を用いて二乗平均誤差の動的ふるまいと定常値を得る。

4 結果と考察

二種類の非線形性に対する定常 MSE をそれぞれ図3(a),(b) に示す。線は理論、シンボルは計算機実験を示す。計算機実験では $N=200$ とし、 $n=10^5$ まで更新を行った。各シンボルは 1000 回の試行の平均値である。不感帯型で $\mu=2.1$, $D<0.6$ の計算機実験においては MSE が発散したため、理論との不一致があるが、それ以外は飽和型、不感帯型のいずれにおいても理論と計算機実験がよく一致している。なお、いずれの図においても“Wiener”は最小二乗解（ウィーナー解[2]）を表す。飽和型の場合には飽和値 S に臨界値 S_C があり、 $S < S_C$ では定常 MSE がステップサイズ μ によらない[1]。また、 S が S_C に近い場合には LMS アルゴリズムの定常 MSE は最小二乗解と大きな差がある。一方、不感帯型の場合には、不感帯幅 D には臨界値は存在しないことがわかった。

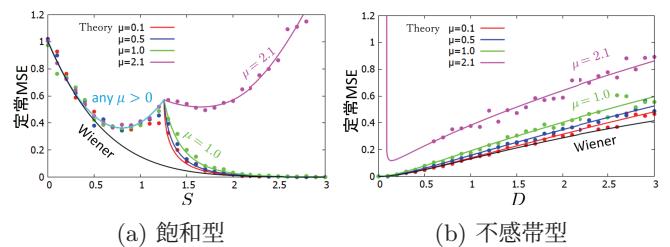


図3 定常 MSE

参考文献

- [1] S. Miyoshi, “Statistical-mechanical analysis of adaptive filter with clipping saturation-type nonlinearity,” IEEE Trans. Signal Process., vol.70, pp.4867-4882, 2022.
- [2] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, Hoboken, NJ, 2003.

二種類の非線形適応フィルタの挙動に関する統計力学的解析

Statistical-mechanical analysis of the behavior of two types of nonlinear adaptive filters

伊原 大地[†]

山本 直季^{‡,*}

本仲 君子[‡]

三好 誠司[‡]

† 関西大学大学院 理工学研究科

‡ 関西大学 システム理工学部

* 株式会社 UACJ

Taichi IHARA[†]

Naoki YAMAMOTO^{‡,*}

Kimiko MOTONAKA[‡]

Seiji MIYOSHI[‡]

† Graduate School of Science and Engineering, Kansai University

‡ Faculty of Engineering Science, Kansai University

* UACJ Corporation

アブストラクト 適応信号処理は広い分野で使われているが、多くの現実の適応システムではその内部に無視できない本質的な非線形性が存在する。本稿では適応フィルタの出力に非線形性を有するようなシステムのふるまいについて統計力学的手法を用いた解析を行う。非線形性としては飽和型と不感帯型の二種類について検討する。二個の巨視的変数を導入し、それらの連立微分方程式を導出し、解くことにより二乗平均誤差のふるまいを理論的に調べる。また、計算機実験との比較によりその妥当性を示す。さらに、最小二乗解との比較を行う。その結果、二種類の非線形性には質的な違いがあることが明らかになる。

1 はじめに

ディジタル信号処理技術は、情報通信分野を中心に様々な分野で利用されている。その一つに適応信号処理という技術がある[1],[2]。適応信号処理は周囲の環境や対象となる信号の性質に応じて適応フィルタのフィルタ係数を更新することで、適応フィルタの出力が目的の出力に近づくようとする。本稿では、非線形性を有しLMSアルゴリズムで更新される適応フィルタのふるまいを統計力学的手法を用いて理論的に明らかにする[3]。非線形性としては飽和型と不感帯型の二種類について検討する[4]。また、計算機実験との比較により理論の妥当性を示す。さらに、これら二種類の非線形性について最小二乗解を導出し、LMSアルゴリズムによる定常MSEと最小二乗解の比較を行う。

2 モデル

本稿で扱うシステムのブロック図を図1に示す。未知システムGと適応フィルタWはいずれもタップ長NのFIRフィルタであり、その係数ベクトルはそれぞれ、 $\mathbf{g} = [g_1, g_2, \dots, g_N]^\top$, $\mathbf{w}(n) = [w_1(n), w_2(n), \dots, w_N(n)]^\top$ で

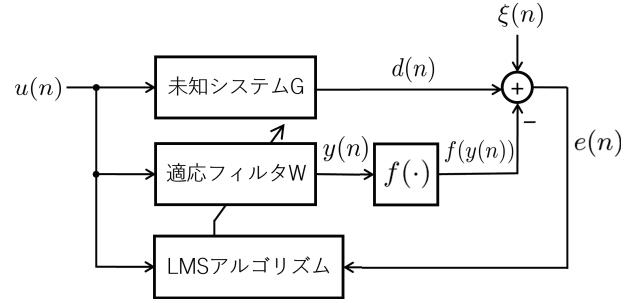


図1: システムのブロック図

ある。ここで $^\top$ は転置、 n は時刻ステップである。 $d(n)$ と $f(y(n))$ の差に、平均0、分散 σ_ξ^2 で時刻ステップごとに独立に生成された背景雑音 $\xi(n)$ が重畠されて誤差信号 $e(n)$ となる。すなわち

$$e(n) = d(n) - f(y(n)) + \xi(n) \quad (1)$$

である。 $f(\cdot)$ は適応フィルタ出力部の非線形関数であり、本稿では飽和型と不感帯型の二種類を扱う。それぞれを図2(a),(b)に示す。入力信号 $u(n)$ は平均0、分散 σ^2 の分布から独立に生成される。適応フィルタWはLMSアルゴリズム

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{u}(n) \quad (2)$$

で更新される。ここで、 $\mathbf{u}(n)$ はタップ入力ベクトル $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-N+1)]^\top$ 、 μ はステップサイズである。

3 理論

本稿では前節で述べたモデルの二乗平均誤差 $\langle e^2(n) \rangle$ のふるまいについて統計力学的解析を行う。その概略は以下の通りである。まず、二個の巨視的変数 $r(n) \equiv \frac{1}{N} \mathbf{g}^\top \mathbf{w}(n)$, $Q(n) \equiv \frac{1}{N} \mathbf{w}(n)^\top \mathbf{w}(n)$ を導入し、二乗平均誤差を r, Q の

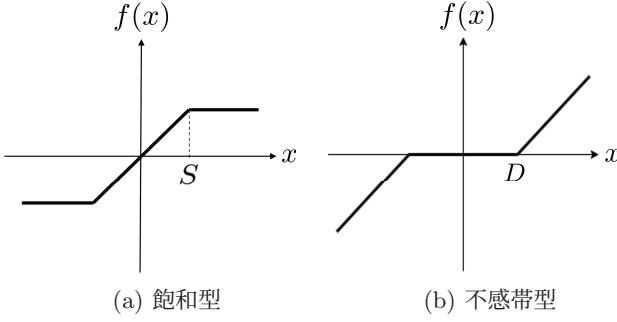


図 2: 非線形関数 $f(\cdot)$

関数として表す。その際、 $N\sigma^2$ を一定に保ったまま $N \rightarrow \infty$ を仮定する。次に、 r, Q の動的ふるまいを記述する連立微分方程式を式(2)と $N \rightarrow \infty$ において成り立つ自己平均性に基づいて決定論的に導出する。導出された連立微分方程式を数値的に解くことにより得られる r, Q を用いて二乗平均誤差の動的ふるまいと定常値を得る。また、微分を0と置くことで二乗平均誤差の定常値を得る。

一方、未知システムの係数ベクトル \mathbf{g} とウィーナーフィルタの係数ベクトル \mathbf{w}^* のなす角を θ とすると、 $\cos \theta = 1$ であることが証明できる。すなわち $\mathbf{w}^* = k\mathbf{g}$ (k は実数)である。

4 結果と考察

二種類の非線形性に対する定常 MSE をそれぞれ図 3, 4 に示す。線は理論、シンボルは計算機実験を示す。計算機実験では $N=200$ とし、 $n=10^5$ まで更新を行った。各シンボルは 1000 回の試行の平均値である。いずれの図においても “Wiener” は最小二乗解 [1], [2] を表す。飽和型では k の値は S の値によらず 1 であるが、不感帯では k の値は D の値が大きくなるにつれて単調に増加する。不感帯型で $\mu=2.1$, $D < 0.6$ の計算機実験においては MSE が発散したため、理論との不一致があるが、それ以外は飽和型、不感帯型のいずれにおいても理論と計算機実験がよく一致している。飽和型の場合には飽和値 S_C に臨界値 S_C があり、 $S < S_C$ では定常 MSE がステップサイズ μ によらない [4]。図 3 より S が S_C に近い場合には LMS アルゴリズムの定常 MSE は最小二乗解と大きな差があることがわかる。また、 $S > S_C$ では S が大きくなるほど定常 MSE と最小二乗解は減少することがわかる。一方、図 4 より不感帯型の場合には、不感帯幅 D に臨界値は存在せず、 D が大きくなるほど定常 MSE と最小二乗解は単調に増大することがわかる。なお、図 4 ではわかりにくいが、ステップサイズ μ を小さくしても定常 MSE は最小二乗解に漸近しない。

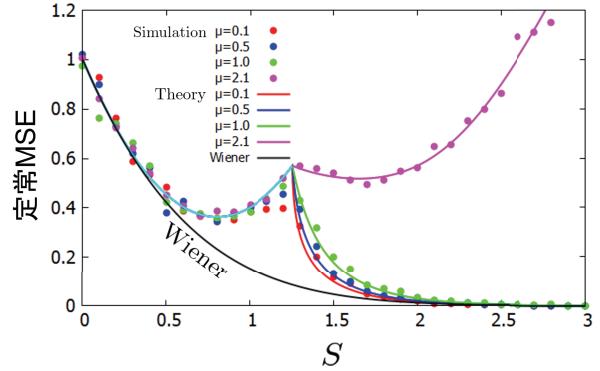


図 3: 定常 MSE (飽和型)

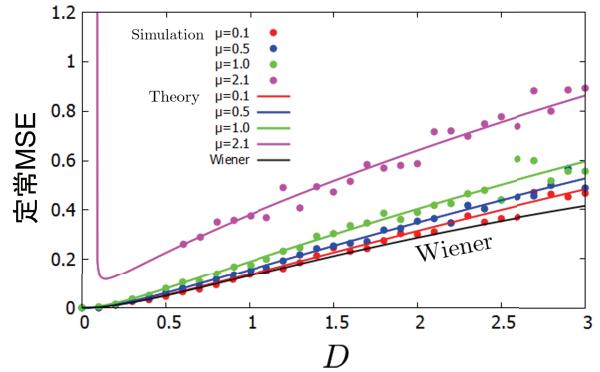


図 4: 定常 MSE (不感帯型)

5 結論

本稿では、適応フィルタの出力に非線形性を有するようなシステムのふるまいについて、統計力学的手法を用いて解析した。非線形性としては飽和型と不感帯型の二種類を扱った。その結果、飽和型では飽和値に臨界値が存在し、その飽和値付近で定常 MSE と最小二乗解に大きな差が存在するのに対し、不感帯型では不感帯幅に臨界値は存在せず、不感帯幅が大きくなると定常 MSE と最小二乗解は単調に増大するものの、ステップサイズを小さくしても定常 MSE が最小二乗解に漸近するわけではないことが明らかになった。

参考文献

- [1] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, Hoboken, NJ, 2003.
- [2] S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2002.
- [3] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*, Oxford University Press, New York, 2001.
- [4] S. Miyoshi, “Statistical-mechanical analysis of adaptive filter with clipping saturation-type nonlinearity,” *IEEE Trans. Signal Process.*, vol. 70, pp. 4867–4882, 2022.

未知システムと適応フィルタが飽和特性を有する 適応信号処理システムの解析

Analysis of adaptive signal processing systems with
saturation-type unknown system and adaptive filter

伊原 大地[†] 本仲 君子[‡] 三好 誠司[‡]

[†]関西大学大学院 理工学研究科 [‡]関西大学 システム理工学部

Taichi IHARA[†] Kimiko MOTONAKA[‡] Seiji MIYOSHI[‡]

[†]Graduate School of Science and Engineering, Kansai University

[‡]Faculty of Engineering Science, Kansai University

アブストラクト 適応信号処理は広い分野で用いられているが、多くの現実の適応信号処理システムにおいてはその内部に無視できない本質的な非線形性が存在する。本研究では未知システムと適応フィルタの両方がその出力に飽和型非線形性を有するような適応信号処理システムのふるまいについて統計力学的手法を用いた解析を行う。すなわち、二個の巨視的変数を導入し、それらの連立微分方程式を導出し、これを数値的に解くことにより二乗平均誤差のふるまいを理論的に調べる。また、計算機実験との比較によりその妥当性を示す。解析の結果、二つの飽和型非線形性の飽和値が学習曲線や定常 MSE に与える影響が明らかになる。

1 はじめに

デジタル信号処理技術は、情報通信分野を中心に様々な分野で利用されている。その一つに適応信号処理という技術がある [1][2]。適応信号処理は周囲の環境や対象となる信号の性質に応じて適応フィルタの係数を更新することで、適応フィルタの出力を目的の出力に近づける。本研究では、未知システムと適応フィルタの両方がその出力に飽和型非線形性を有し、適応フィルタが LMS アルゴリズムで更新される適応信号処理システムのふるまいを統計力学的手法を用いて理論的に明らかにする [3][4]。

2 モデル

本研究で扱うシステムのブロック図を図 1 に示す。未知システム G と適応フィルタ W はいずれもタップ長 N の FIR フィルタであり、その係数ベクトルはそれぞれ、 $\mathbf{g} = [g_1, g_2, \dots, g_N]^\top$, $\mathbf{w}(n) = [w_1(n), w_2(n), \dots, w_N(n)]^\top$ である。ここで $^\top$ は転置、 n は時刻ステップを表す。入力信号 $u(n)$ は平均 0, 分散 σ^2 の分布から独立に生成される。 $d(n) = \mathbf{g}^\top \mathbf{u}(n)$ は未知システム G の出力、

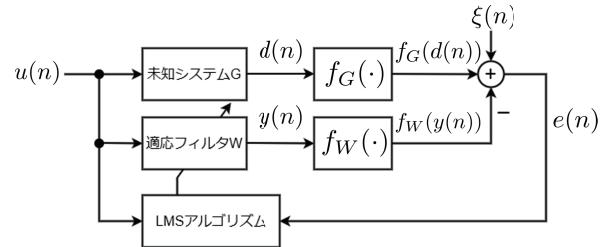


図 1: システムのブロック図

$y(n) = \mathbf{w}(n)^\top \mathbf{u}(n)$ は適応フィルタ W の出力である。ここで、 $\mathbf{u}(n)$ はタップ入力ベクトル $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-N+1)]^\top$ である。G, W の出力にはそれぞれ $f_G(\cdot)$, $f_W(\cdot)$ で表される非線形性が存在する。本研究では図 2(a), (b) に示す飽和型非線形関数を扱う。誤差信号 $e(n)$ は、

$$e(n) = f_G(d(n)) - f_W(y(n)) + \xi(n) \quad (1)$$

である。ここで背景雑音 $\xi(n)$ は平均 0, 分散 σ_ξ^2 の分布から独立に生成される。適応フィルタ W は、LMS アルゴリズム

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{u}(n) \quad (2)$$

で更新される。ここで、 μ はステップサイズである。

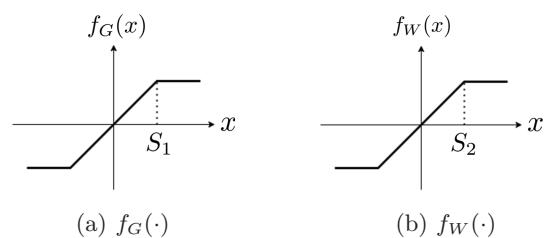


図 2: 2 つの飽和型非線形性

3 理論

本研究では前節で述べたモデルの二乗平均誤差 $\langle e^2(n) \rangle$ のふるまいについて統計力学的解析 [4] を行う。ここで $\langle \cdot \rangle$ は平均を表す。その概略は以下の通りである。二乗平均誤差は式(1)より

$$\begin{aligned} \langle e^2(n) \rangle &= \langle (f_G(d(n)) - f_W(y(n)) + \xi(n))^2 \rangle \\ &= \langle f_G^2(d(n)) \rangle + \langle f_W^2(y(n)) \rangle \\ &\quad - 2\langle f_G(d(n))f_W(y(n)) \rangle + \sigma_\xi^2 \end{aligned} \quad (3)$$

である。ここで、 $\rho^2 = N\sigma^2$ を一定に保ったまま $N \rightarrow \infty$ を考える。このとき、中心極限定理より $d(n)$ と $y(n)$ は、平均がいずれも 0 で共分散行列が $\rho^2(\sigma_g^2, r(n); r(n), Q(n))$ である二次元ガウス分布に従う。ここで、 $r(n), Q(n)$ はそれぞれ $r(n) = \frac{1}{N}\mathbf{g}^\top \mathbf{w}(n)$, $Q(n) = \frac{1}{N}\mathbf{w}(n)^\top \mathbf{w}(n)$ で定義される巨視的変数である。式(3)の右辺第一項、第二項は、ガウス積分を解析的に実行することで求めることができる。第三項を求める際には、解析的に実行することが困難な二重ガウス積分が必要となるため、数値的に求める。結果的に式(3)は $r(n)$ と $Q(n)$ の関数になる。 $r(n)$ と $Q(n)$ の動的ふるまいを記述する連立微分方程式は、 $N \rightarrow \infty$ における自己平均性 [3] に基づき決定論的な形で導出することができる。これを数値的に解くことにより二乗平均誤差の動的ふるまい、すなわち学習曲線を得ることができる。

4 結果と考察

S_1 を 1 に固定し、 S_2 の値を変化させた場合の学習曲線を図 3(a) に、 S_2 を 1 に固定し、 S_1 の値を変化させた場合の学習曲線を図 3(b) に示す。横軸 t は $t = n/N$ で定義される時間尺度である。線は理論、シンボルは計算機実験の結果を示しており、ステップサイズ μ は 0.1 である。計算機実験では $N=200$ とし、各シンボルは 10000 回の試行の平均値を示す。背景雑音の分散 σ_ξ^2 は 0 とした。いずれの図においても理論は計算機実験の結果をよく説明していることがわかる。また、 $S_1=S_2$ の場合、MSE は 0 に漸近することが分かる。一方、 S_1 と S_2 の差が大きくなると、MSE は大きくなる。

次に定常 MSE について調べる。ただし t が大きくなると $d(n)$ と $y(n)$ の共分散行列が特異に近くなり、式(3)右辺第三項を求めるための数値積分が不安定になるため、ここでは $t=80$ の MSE を準定常 MSE とする。 S_2/S_1 と準定常 MSE の関係を図 4 に示す。この結果から、 $S_1=S_2$ 附近で準定常 MSE が最小となっていることが確認できる。しかし、図 3(b) からもわかるように $t=80$ では MSE は十分に定常値に達していない場合もあるため、準定常 MSE が最小となる S_2/S_1 の値は 1 から少しずれていることが分かる。

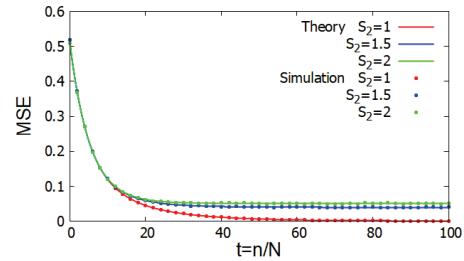
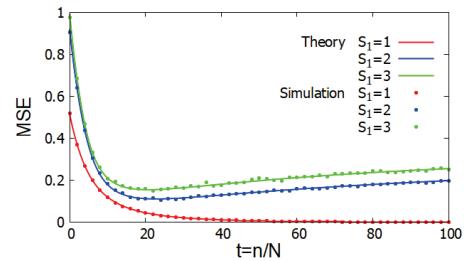
(a) $S_1=1$ に固定した場合(b) $S_2=1$ に固定した場合

図 3: 学習曲線

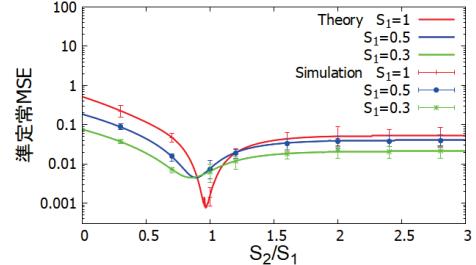


図 4: 準定常 MSE

5 結論

本研究では、未知システムと適応フィルタの両方の出力に飽和型非線形性を有するような適応信号処理システムのふるまいについて統計力学的手法を用いた解析を行った。その結果、 $S_1=S_2$ の場合、MSE は 0 に漸近すること、 S_1 と S_2 の差が大きくなると MSE は大きくなることが分かった。時間 t が大きい場合に MSE の計算が不安定となり定常 MSE を正しく求められない問題の解決は今後の課題である。

参考文献

- [1] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, Hoboken, NJ, 2003.
- [2] S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2002.
- [3] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*, Oxford University Press, New York, 2001.
- [4] S. Miyoshi, “Statistical-mechanical analysis of adaptive filter with clipping saturation-type nonlinearity,” *IEEE Trans. Signal Process.*, vol.70, pp.4867–4882, 2022.