

修士論文の和文要旨

研究科・専攻	大学院情報システム学研究科 情報システム基盤学専攻 博士前期課程		
氏名	吉川 貴博	学籍番号	1153027
論文題目	注入同期における引き込み能力を最適化する外部周期信号の設計		
要旨	<p>近年、同期現象の1つである注入同期を無線通信端末における発振器など工学分野へ応用する研究が注目されている。注入同期とは、振動子に外部より周期信号を与えることで、周期信号の周波数に振動子の周波数が引き込まれ、同調する現象である。</p> <p>注入同期において、実環境を想定するとノイズ等の周辺環境の影響により、振動子の周波数は揺らぎ、同期が外れてしまう事がある。したがって、振動子の引き込み能力を最適化する周期信号を設計する事は重要な課題の1つであるといえる。また、この周期信号を設計するために、振動子の外部入力への応答の影響を示す位相応答曲線を正確に推定する事も重要な課題である。従来、田中によってノイズのない理想的な環境下、ノイズ環境下での最適外部信号を設計するアルゴリズムが構築されている。また、PRC推定によっても菊地らがノイズ耐性の強い位相応答曲線の推定手法を提案している。しかし、菊地らは振動子への数値シミュレーションにおけるノイズ環境下での確率微分方程式の解き方に誤りがあった。そのため、菊地らの手法のノイズ耐性を再検証する必要がある。</p> <p>そこで本研究ではまず、代表的な振動子系を想定して、正しい設定で菊地らによる位相応答曲線の推定手法のノイズ耐性の検証を行う。次に、推定した位相応答曲線を用いて、理論的に最適周期信号を設計する。また、最適周期信号の設計の別のアプローチとして遺伝的アルゴリズムを設計し、遺伝的アルゴリズムによる探索解でも上記の信号を得る。その結果、従来手法より精度よく位相応答曲線を推定できるノイズの大きさの範囲を得た。そして、遺伝的アルゴリズムの結果と比較する事により、理論解の妥当性を示した。</p>		

平成24年度 修士論文

注入同期における引き込み能力を最適化する
外部周期信号の設計

学籍番号

1153027

氏名

吉川 貴博

情報システム学研究科

情報システム基盤学専攻

主指導教員

田中 久陽 准教授

指導教員

大森 匡 教授

指導教員

近藤 正章 准教授

提出日

平成25年1月24日

概要

近年、同期現象の1つである注入同期を無線通信端末における発振器など工学分野へ応用する研究が注目されている。注入同期とは、振動子に外部より周期信号を与えることで、周期信号の周波数に振動子の周波数が引き込まれ、同調する現象である。注入同期において、実環境を想定するとノイズ等の周辺環境の影響により、振動子の周波数は揺らぎ、同期が外れてしまう事がある。したがって、振動子の引き込み能力を最適化する周期信号を設計する事は重要な課題の1つであるといえる。また、この周期信号を設計するために、振動子の外部入力への応答の影響を示す位相応答曲線を正確に推定する事も重要である。従来、田中によってノイズのない理想的な環境下、ノイズ環境下での最適外部信号を設計するアルゴリズムが構築されている。また、PRC推定によっても菊地らがノイズ耐性の強い位相応答曲線の推定手法を提案している。しかし、菊地らは振動子への数値シミュレーションにおけるノイズ環境下での確率微分方程式の解き方に誤りがあった。そのため、菊地らの手法のノイズ耐性を再検証する必要がある。

そこで本研究ではまず、代表的な振動子系を想定して、菊地らによる位相応答曲線の推定手法の再検証を行う。次に、推定した位相応答曲線を用いて、理論的に最適周期信号を設計する。また、別のアプローチとして遺伝的アルゴリズムを設計し、遺伝的アルゴリズムによる探索解でも上記の信号を得る。

その結果、従来手法より位相応答曲線を推定できるノイズの大きさを得た。そして、最適周期信号の設計において、理論の適用内である振動子については、遺伝的アルゴリズムの結果と比較する事により、理論解の妥当性を示した。

目次

第 1 章	序論	1
1.1	同期現象と実例	1
1.2	対象とするモデル	1
1.2.1	Hodgkin-Huxley 方程式	1
1.2.2	Rössler 方程式	5
1.3	位相応答曲線	6
1.4	振動子の位相記述法	6
1.4.1	弱い摂動を受けた時の振動子	6
1.4.2	弱い周期外力を受けた時の振動子	7
1.4.3	ランダム外力を受けた時の振動子	8
1.5	本研究の目的	9
第 2 章	理想的な環境下での位相応答曲線の推定及び最適な外部周期信号の設計	10
2.1	理想的な環境下での位相応答曲線の推定	10
2.2	ヘルダーの不等式による最適周期信号の理論的導出	11
2.2.1	ヘルダーの不等式による最適化アルゴリズム	11
2.2.2	Hodgkin-Huxley 振動子の最適周期信号の理論的導出	23
2.3	Genetic Algorithm を用いた最適周期信号の探索	36
2.3.1	Genetic Algorithm の設計	36
2.3.2	理論解と Genetic Algorithm による探索解の比較	37
2.4	設計した最適外部周期信号の引き込み能力のシミュレーション検証	39
2.5	過倍動作時の最適外部周期信号の引き込み能力のシミュレーション検証	43
	パワー一定の場合	43
	振幅一定の場合	44
	面積一定の場合	47
第 3 章	ノイズ環境下での位相応答曲線の推定及び最適な外部周期信号の設計	50
3.1	数値シミュレーションにおけるノイズの与え方	50
3.2	ノイズ環境下での位相応答曲線の推定	51
3.2.1	菊地らによる位相応答曲線の推定手法の原理	52
	$n \geq 1$ の場合	52
	$n = 0$ の場合	53
	注入正弦波の角周波数が振動子の自然角周波数の整数倍でない場合	53
3.2.2	位相応答曲線の推定手順	54
3.2.3	位相応答曲線の推定のノイズ耐性の検証	57
	Hodgkin-Huxley 方程式の位相応答曲線の推定	57

	Rössler 方程式の位相応答曲線の推定	61
3.3	ノイズ環境下での最適波形の導出	62
3.3.1	確立分布のピーク値を最大化する周期信号の理論的導出	62
3.3.2	確立分布のピーク値を対象とした時の理論解と Genetic Algorithm の探索解の 比較	70
	Hodgkin-Huxley 振動子を対象とした時の理論解と GA による探索解の比較 . .	70
	Rössler 方程式を対象とした時の理論解と GA による探索解の比較	74
第 4 章	結論	81
	謝辞	82
	参考文献	83
	付録	84

第1章 序論

1.1 同期現象と実例

自然界には固有のリズムを持つ振動子が数多く存在する。このような振動子はリミットサイクル振動子と呼ばれ、複数の振動子が相互に影響し合うと、互いの周波数が一致する。この現象を同期現象といい、古くから実例が発見されている。例えば、ホテルの集団同期明滅、メトロノームの集団同期などが知られており、これらは理論的な解析も行われている。工学分野においては、無線端末の発振器の位相雑音を低減させるために、注入同期と呼ばれる現象が利用されている。注入同期とは、振動子に外部より周期信号を与えることで、周期信号の周波数に振動子の周波数が引き込まれ、同調する現象である。

本論文では、振動子の数理モデルを対象に、以上の同期現象の解析を行う。

1.2 対象とするモデル

本研究で対象とする振動子のモデルとして、Hodgkin-Huxley 方程式、Rössler 方程式を採用する。本節では、Hodgkin-Huxley 方程式 [1]、Rössler 方程式 [2] の概要を説明する。

1.2.1 Hodgkin-Huxley 方程式

Hodgkin-Huxley 方程式は神経細胞に見られる活動電位について現象論的に記述した式で、時間変数 V , m , h , n の4つの方程式からなる連立方程式であらわされる。

$$C \cdot \frac{dV}{dt} = G_{Na}(E_{Na} - V) + G_K(E_K - V) + G_L(E_L - V) \quad (1.1)$$

ここで、 G_{Na} , G_K はそれぞれナトリウムイオンとカリウムイオンの濃度で決まるコンダクタンス、 G_L はリークチャネルを表す抵抗のコンダクタンスである。これらのコンダクタン

スは、さらに各コンダクタンスのピーク値を表す定数 g_{Na} , g_K , g_L , 時間変数 m , n , h を導入して次のように表せる.

$$G_{Na} = g_{Na}m^3h \quad (1.2)$$

$$G_K = g_Kn^4 \quad (1.3)$$

$$G_L = g_L \quad (1.4)$$

ただし、文献 [3] に倣い、コンデンサの容量を表す定数 C , 平衡電位を表す定数 E_{Na} , E_K , E_L , 各コンダクタンスのピーク値を表す定数 g_{Na} , g_K , g_L , 及び時間変数 m , n , h の方程式は次のように設定した.

$$C = 1[\mu\text{F}/\text{cm}^2] \quad (1.5)$$

$$E_{Na} = 50[\text{mV}] \quad (1.6)$$

$$E_K = -77[\text{mV}] \quad (1.7)$$

$$E_L = -54.4[\text{mV}] \quad (1.8)$$

$$g_{Na} = 120[\text{mS}/\text{cm}^2] \quad (1.9)$$

$$g_K = 36[\text{mS}/\text{cm}^2] \quad (1.10)$$

$$g_L = 0.3[\text{mS}/\text{cm}^2] \quad (1.11)$$

$$\frac{dm}{dt} = 0.1 \cdot \frac{V + 40}{1 - \exp\left(-\frac{V+40}{10}\right)} \cdot (1 - m) - 4 \cdot \exp\left(-\frac{V + 65}{18}\right) \cdot m \quad (1.12)$$

$$\frac{dh}{dt} = 0.07 \cdot \exp\left(-\frac{V + 65}{20}\right) \cdot (1 - h) - \frac{1}{1 + \exp\left(-\frac{V+35}{10}\right)} \cdot h \quad (1.13)$$

$$\frac{dn}{dt} = 0.01 \cdot \frac{V + 55}{1 - \exp\left(-\frac{V+55}{10}\right)} \cdot (1 - n) - 0.125 \cdot \exp\left(-\frac{V + 65}{80}\right) \cdot n \quad (1.14)$$

V , m , h , n の時間変化と V , m からなるリミットサイクルは以下のようなになる. 以下の図より, Hodgkin-Hoxkley 振動子は周期的に運動している振動子であることがわかる.

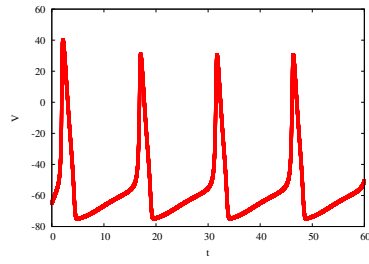


图 1.1: V

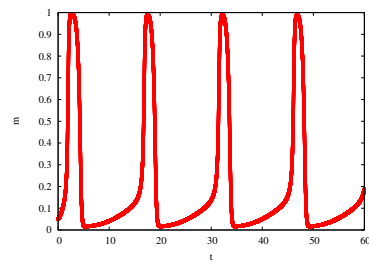


图 1.2: m

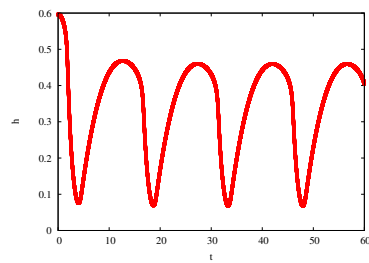


图 1.3: h

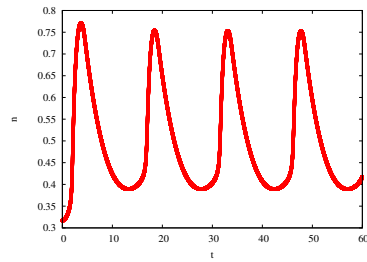


図 1.4: n

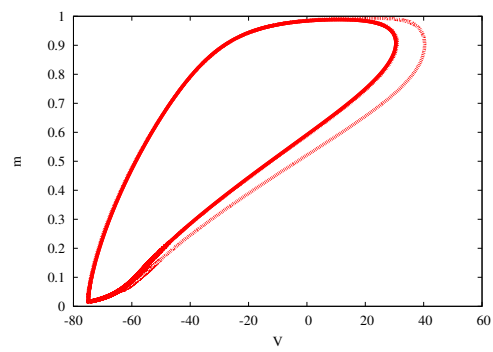


図 1.5: リミットサイクル

1.2.2 Rössler 方程式

Rössler 方程式とは、対流のモデルとして知られているローレンツ方程式を単純化したもので、以下の3式で表される。

$$\frac{dx}{dt} = -\alpha y - z \quad (1.15)$$

$$\frac{dy}{dt} = \alpha x + 0.15y - Cy \quad (1.16)$$

$$\frac{dz}{dt} = 0.2 + z(x - 2) \quad (1.17)$$

x , y , z の時間変化と x , y からなるリミットサイクルは以下ようになる。以下の図より、Rössler 振動子は周期的な定常解に落ち着く振動子であることがわかる。

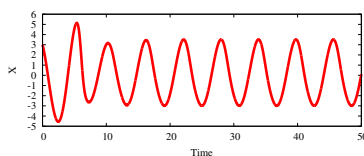


図 1.6: x

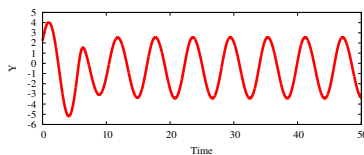


図 1.7: y

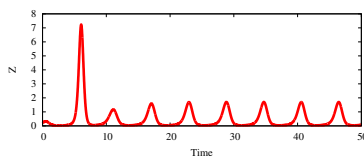


図 1.8: z

この方程式の非線形性は、式 (1.17) のみである。このような単純な非線形性を有する方程式でありながら、天体物理学、化学など広範囲で適用される。

以上の2つの方程式を対象に本研究でシミュレーション検証を行なった。

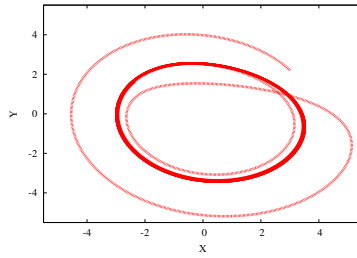


図 1.9: リミットサイクル

1.3 位相応答曲線

注入同期において，振動子の周波数は外部周期信号の影響を受けて変化している．これは，振動子の周期的運動における周期中の位置を位相と定義すると，振動子は外部周期信号の影響により位相を変化させているともいえる．振動子の位相の変化は外部周期信号を受ける位相によって異なる．この位相変化の位相依存性を表したものを位相応答曲線と呼ぶ．位相応答曲線は各位相に対し，その位相で受けた振動子の位相変化量を表す周期関数である．したがって，位相応答曲線によって，外部周期信号を受けた振動子の振る舞いを知ることができるので振動子の位相応答曲線を正確に推定する事は重要である．

1.4 振動子の位相記述法

振動子の振る舞いを解析する際，位相記述法がよく用いられる [4]．本節では，様々な外力を受けた場合における振動子の振る舞いを示す．

1.4.1 弱い摂動を受けた時の振動子

リミットサイクル振動子は以下の n 次元力学系で表される．

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}) \quad (1.18)$$

ここで， $\mathbf{X} = (X_1, X_2, \dots, X_n)$ は n 次元ユークリッド空間のベクトルであり， t は時間， \mathbf{F} は \mathbf{X} の非線形なベクトル関数である．また，リミットサイクル振動子の周期を T とすると，任意の点 \mathbf{X} の位相 $\phi(\mathbf{X})$ と 1 周期後の位相 $\phi(\mathbf{X} + T)$ は $\phi(\mathbf{X}) = \phi(\mathbf{X} + T)$ の関係となる．つまり，振動子の角周波数を ω とすると，位相 ϕ はリミットサイクル起動 C 上のある位置を表

し、 C 上の周回運動と共に ϕ が $\frac{d\phi}{dt} = \omega$ に従うものと定義する。 ϕ は \mathbf{X} を通じてのみ時間変化するため、 $\frac{d\phi}{dt} = \text{grad}_{\mathbf{X}}\phi \mathbf{F}(\mathbf{X})$ となり、上記の ϕ の定義より、以下が成り立つ。

$$\frac{d\phi}{dt} = \text{grad}_{\mathbf{X}}\phi \mathbf{F}(\mathbf{X}) = \omega \quad (1.19)$$

ここで、外部から弱い摂動 $\mathbf{p}(t)$ を受けた振動子の時間変化は以下で表される。

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}) + \varepsilon \mathbf{p}(t) \quad (1.20)$$

ただし、 ε は摂動の弱さを表すための微小パラメータである。式 (1.20) のとき、式 (1.19) を用いると位相の時間変化は以下で表される。

$$\frac{d\phi}{dt} = \text{grad}_{\mathbf{X}}\phi \cdot [\mathbf{F}(\mathbf{X}) + \mathbf{p}(t)] = \omega + \text{grad}_{\mathbf{X}}\phi \cdot \mathbf{p}(t) \quad (1.21)$$

$\mathbf{p}(t)$ は弱い摂動のため、 \mathbf{X} はリミットサイクル起動から外れないと考える事が出来る。よって、式 (1.21) の位相変化量 $\text{grad}_{\mathbf{X}}\phi$ は同じ位相 ϕ をもつ周期関数 $Z(\phi)$ で近似できる。この $Z(\phi)$ が 1.3 節で述べた位相応答曲線である。したがって、位相応答曲線を用いると (1.21) は以下の式で表される。

$$\frac{d\phi}{dt} = \omega + \mathbf{Z}(\phi) \cdot \mathbf{p}(t) \quad (1.22)$$

式 (1.22) より、弱い摂動を受けた振動子の振る舞いを記述できた。

1.4.2 弱い周期外力を受けた時の振動子

前説では弱い外力を受けた振動子の振る舞いを示した。本節では、外力が平均角周波数 Ω の周期的な信号 $\varepsilon f(\Omega t)$ である場合を考える。振動子の自然角周波数を ω_0 としたとき、振動子と周期外力の周波数差を $\varepsilon \Delta\omega = \omega - \Omega$ とおき、これを周波数離調という。 ε は微小なパラメータのため、離調周波数は十分小さいものとする。このとき、式 (1.22) より移送方程式は以下のようなになる。

$$\frac{d\phi}{dt} = \omega + Z(\phi) \cdot \varepsilon f(\Omega t) \quad (1.23)$$

ここで、 $\phi = \Omega t + \psi$ として新しい位相変数 ψ を導入すると、式 (1.23) を以下のように表せる。

$$\frac{d\psi}{dt} = \omega_0 - \Omega + Z(\psi + \Omega t) \cdot \varepsilon f(\Omega t) \quad (1.24)$$

$$= \varepsilon \Delta\omega + \varepsilon Z(\psi + \Omega t) f(\Omega t) = \varepsilon [\Delta + Z(\psi + \Omega t) f(\Omega t)] \quad (1.25)$$

ψ は振動子の周期外力の位相差を表しており、位相差が時間的に一定なら、あるいは ψ の変動が有限の範囲に収まるとき、 $\frac{d\psi}{dt} = 0$ となる。このとき、振動子は周期外力に同期したといえる。逆に、 ψ がドリフトし続けてしまうとき、非同期という。式 (1.25) より、 ψ の時間的変化量は、 ϕ に比べて非常にゆっくりとした変動になる。

1.4.3 ランダム外力を受けた時の振動子

本節では、振動子に与える外力をランダム外力、つまりノイズ $\xi(t)$ である場合を考える。ノイズ $\xi(t)$ は定常確率分布であるとし、以下を満たすホワイトガウスノイズとする。

$$\langle \xi(t) \rangle = 0 \quad (1.26)$$

$$\langle \xi(t_0)\xi(t_0 + t) \rangle = 2D\delta(t) \quad (1.27)$$

ここで、 $\langle \dots \rangle$ は平均を表しており、式 (1.27)、(1.27) は $\xi(t)$ の確率分布が 0 を中心として分散値 D のガウス分布となることを示している。このとき、振動子の位相の時間発展は式 (1.22) より以下のように表せる。

$$\frac{d\phi}{dt} = \omega_0 + Z(\phi) \cdot \xi(t) \quad (1.28)$$

確率的に揺らぐ量 $\xi(t)$ を含む式 (1.28) のような確率微分方程式を Langevin 方程式という。以上より、式 (1.28) よりランダム外力を受けた時の振動子の振る舞いを示した。前節と同様に $\phi = \omega_0 t + \psi$ とおき、ゆっくり変化する変数 ψ を導入すると、Langevin 方程式は以下のような Fokker-Planck 方程式に変換される。

$$\frac{\partial P(\psi, t)}{\partial t} = -M_1 \frac{\partial P(\psi)}{\partial \psi} + \frac{M_2}{2} \frac{\partial^2 P(\psi, t)}{\partial \psi^2} \quad (1.29)$$

ここで、 $P(\psi, t)$ は ψ の確率分布であり、時刻 t において位相が ψ から $\psi + \Delta\psi$ に飛躍する確率を表している。また、 M_1 及び M_2 は確率繊維の 1 次、2 次モーメントを表しており、式 (1.29) の右辺第一項はドリフト項、第二項は拡散項である。ランダム外力の影響がないか、極めて弱い場合に振動子と周期外力が同期すればそのとき両者の位相差は一定となるため $P(\psi, t)$ は $\psi = 0$ に鋭いピークを持つ。一方、ランダム外力の影響が強い場合、分布の粗野は広がり、ピークは $\psi = 0$ で固定されない。よって、位相差の確率分布のピークが高いほど、その周期外力は振動子の位相をよく固定できる信号と言える。

1.5 本研究の目的

本研究では、位相応答曲線の推定と最適周期信号の設計の2つの問題に注目した。位相応答曲線を用いる事でその振動子の挙動を解析しやすくなるため、位相応答曲線を精度よく推定する事は本研究で注目する外部周期信号の最適化など工学分野において必要不可欠な問題であることがわかる。また、最適周期信号を設計する事で、高速無線通信端末の位相雑音の低減を可能にするなど、これも重要な問題となっている。上記2つの問題は、個々では広く研究されている。しかし、上記2つの問題を同時に注目した研究、つまり、現実的な環境下で位相応答曲線の推定を行い、そこから最適周期信号の設計を行った例はない。そこで本研究では、振動子系を対象にノイズがない理想的な環境とノイズのある現実的な環境の環境下両方を想定して、位相応答曲線の推定及び最適周期信号の設計をシミュレーションで行なった。

第2章 理想的な環境下での位相応答曲線の推定及び最適な外部周期信号の設計

2.1 理想的な環境下での位相応答曲線の推定

理想的な環境下を想定した位相応答曲線の推定手法として、インパルス応答法がある [5]. インパルス応答法とは、発振器に微小インパルスを注入し、その結果生じる応答を計測することにより位相応答曲線を求めるという単純な手法である. 例えば、位相シフト量が0であった場合、その時間での位相応答曲線の取る値も0であり、位相シフトが発生すれば、その時間での位相応答曲線の取る値は位相シフト量である. そして、横軸をインパルスを注入した時間、縦軸を位相シフト量とすれば、位相応答曲線を推定する事ができる.

実際にシミュレーションにより得た Hodgkin-Huxley 振動子の位相応答曲線を以下に示す. 注入したインパルスはパルス幅を Hodgkin-Huxley 振動子の約 1/100 である 0.14[ms], パルスの高さを振動子の約 1/30 である 0.2 と十分微小なインパルスを用いた.

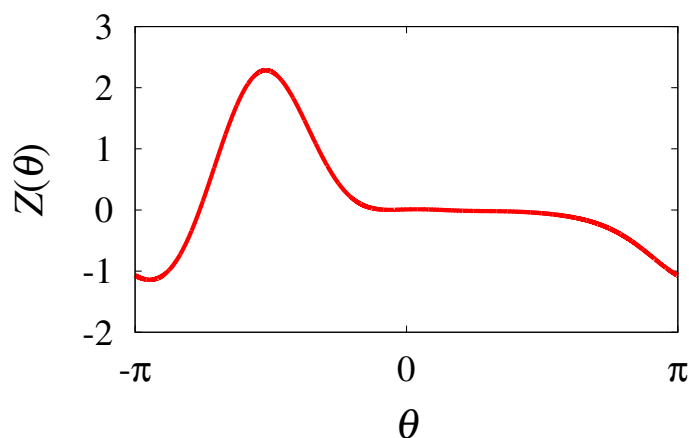


図 2.1: Hodgkin-Huxley 振動子の位相応答曲線

図 2.1 の PRC を用いて、本章では最適周期信号の設計を行う.

2.2 ヘルダーの不等式による最適周期信号の理論的導出

2.2.1 ヘルダーの不等式による最適化アルゴリズム

弱い外部周期信号を受けた振動子の発振位相 ψ の挙動は、一般的に位相縮約法を用いて表される。この時、外部周期信号の影響を受けた振動子の振る舞いは位相応答関数 $Z(\psi)$ で表される。ここで、外部周期信号 $f(\theta)$ の発振位相を $\theta = \Omega t$ とすると、 ψ の時間変化は式 (2.1) で表される。

$$\frac{d\psi}{dt} = \omega + Z(\psi)f(\theta) \quad (2.1)$$

このモデルは Winfree モデルという。式 (2.1) に対し、振動子と外部周期信号の位相差を表す変数 $\phi = \psi - \theta$ を導入する。一般的に、 $m : n$ 同期は $\frac{\omega}{m} - \frac{\Omega}{n} \sim O(\epsilon)$ が正の相対的な整数 m と n で満たされるときに起こる。この場合、振動子と外部周期信号の周波数差 $\Delta\omega$ が比較的小さいとすると、式 (2.1) を外部周期信号の 1 周期で平均化することにより、 ϕ の時間変化は式 (2.2) で表される。

$$\frac{d\phi}{dt} = \Delta\omega + \Gamma_{m/n}(\phi) \quad (2.2)$$

ここで、 ϕ と $\Delta\omega$ はそれぞれ $m\phi = \psi - \frac{m}{n}\Omega t$ と $\Delta\omega = \frac{\omega}{m} - \frac{\Omega}{n}$ として定義され、相互作用関数 $\Gamma_{m/n}(m\phi)$ は (2.3) 式のように f と Z によって与えられる。

$$\begin{aligned} \Gamma_{m/n}(m\phi) &= \frac{1}{T} \int_0^T \frac{1}{m} \left(\frac{m}{n}\Omega t + m\phi \right) f(\theta) dt \\ &= \frac{1}{2\pi m} \int_{-\pi}^{\pi} Z(m\theta + m\phi) f(n\theta) d\theta \equiv \langle Z(m\theta + m\phi) f(n\theta) \rangle \end{aligned} \quad (2.3)$$

ここで、外部周期信号の周期 T は $T = \frac{2\pi n}{\Omega}$ で、 $\theta \in [-\pi, \pi]$ は $\frac{\Omega t}{n}$ を表す。本研究では主に $m = n = 1$ のケースの 1:1 同期を考え、簡単のために $\Gamma_{m/n}$ を Γ として表わすこととする。また、 $\frac{d\phi}{dt} = 0$ となると、振動子と外部周期信号の位相差は一定となり同期している。 $\Delta\omega$ は定数なので、その値が $\Gamma(\phi)$ の最大値と最小値の範囲内であり、そこで $\Gamma' < 0$ ならば、振動子と外部周期信号は安定に同期可能である。そのため、 $\Gamma(\phi)$ の最大値を $\Gamma(\phi_+)$ 、最小値を $\Gamma(\phi_-)$ とおくと、ロッキングレンジ R は式 (2.4) で表される。

$$R[f] = \Gamma(\phi_+) - \Gamma(\phi_-) = \langle Z(\theta + \phi_+)f(\theta) - Z(\theta + \phi_-)f(\theta) \rangle \quad (2.4)$$

本研究では外部周期信号に対して 2 つの制約を課す。一つは

$$\langle f(\theta) \rangle = 0 \quad (2.5)$$

である。すなわち、外部周期信号は直流成分のない周期信号である。もう一つは、

$$\|f\|_p \equiv \langle |f(\theta)|^p \rangle^{\frac{1}{p}} = M \quad (2.6)$$

を満たすことであり、これを p ノルムという。ここで、 $p \geq 1$ は必ず正の実数で、 M は正の定数であると想定する。特に、 $p = 2$ のケースで、制約の式 (2.6) は $\langle f^2 \rangle = M^2$ となり、 f のパワーが M^2 で固定されるということである。一方、 $p = \infty$ のケースで、制約式 (2.6) の $\|f\|_\infty = M$ は $|f|$ の最大値（絶対値）が M という制約を与える。また、 $p = 1$ のケースで、制約式 (2.6) の $\|f\|_1 = M$ は f の 1 周期面積が M であるという制約を与える。このように p の値を変えることにより、異なる制約条件下での外部周期信号を設計できる。

制約条件を満たす外部周期信号 $f(\theta)$ の最適波形を得るために、式 (2.7) に示す汎関数 $J[f]$ を定義する。

$$J[f] = R[f] - \lambda_1 \langle f(\theta) \rangle - \lambda_2 [\|f(\theta)\|_p - M] \quad (2.7)$$

λ_1, λ_2 はラグランジュ未定乗数である。ところが一般に、式 (2.8) に示すヘルダーの不等式が成立し、

$$\|fg\|_1 \leq \|f\|_p \|g\|_q \quad (2.8)$$

式 (2.8) において等号が成立する時に左辺が最大化される。ここで、 p, q は $1 \leq p, q \leq \infty$ と $p^{-1} + q^{-1} = 1$ を満たす実数である。中でも、 $1 < p, q < \infty$ で式 (2.8) での等式 $\|fg\|_1 = \|f\|_p \|g\|_q$ は、0 でない定数 α と β を用いて $\alpha |f(\theta)|^p = \beta |g(\theta)|^q$ ($\theta \in S$) であるときに限り成り立つ。

本研究で想定している問題において、式 (2.8) での $f(\theta)$ は式 (2.1) 式の外部周期信号 $f(\theta)$ としてみなせる。従って、式 (2.8) は式 (2.6) から次のようになる。

$$\langle fg \rangle \leq \langle |fg| \rangle = \|fg\|_1 \leq \|f\|_p \|g\|_q = M \|g\|_q \quad (2.9)$$

この式 (2.9) より、等式条件 $\alpha |f(\theta)|^p = \beta |g(\theta)|^q$ を満たす f を探すことは、今得たい最適周期信号を設計することと等価である。この時、 $q = \frac{p}{p-1}$ を用いて $\alpha |f(\theta)|^p = \beta |g(\theta)|^q$ より、式 (2.10) を得る。

$$|f(\theta)| = \left(\frac{\beta}{\alpha} \right)^{\frac{1}{p}} |g(\theta)|^{\frac{1}{p-1}} > 0 \text{ または, } f(\theta) = \pm \left(\frac{\beta}{\alpha} \right)^{\frac{1}{p}} |g(\theta)|^{\frac{1}{p-1}} \quad (2.10)$$

ここで、 $(\beta/\alpha)^{1/p}$ は $\alpha\|f\|_p^p = \beta\|g\|_q^q$ なので、式 (2.11) が得られる。

$$\left(\frac{\beta}{\alpha}\right)^{\frac{1}{p}} = \frac{\|f\|_p}{\|g\|_q^{\frac{q}{p}}} = \frac{M}{\|g\|_q^{\frac{1}{p-1}}} \quad (2.11)$$

式 (2.10) と式 (2.11) より、式 (2.12) 最適周期信号 $f_{\text{opt},p}(\theta)$ の成分を形成する。

$$f(\theta) = \pm \left(\frac{\beta}{\alpha}\right)^{\frac{1}{p}} |g(\theta)|^{\frac{1}{p+1}} = \pm M \left(\frac{|g(\theta)|}{\|g\|_q}\right)^{\frac{1}{p-1}} \quad (2.12)$$

また、ヘルダーの不等式の式 (2.8) と式 (2.7) を対応づけるために、式 (2.7) の第 1,2 項を式 (2.13) のように書き換える。

$$R[f] - \lambda_1 \langle f(\theta) \rangle = \langle f(\theta)(\bar{Z}(\theta) + \lambda_1) \rangle = \langle fg \rangle \quad (2.13)$$

ここで $\bar{Z}(\theta) = Z(\theta + \Delta\phi) - Z(\theta)$, $\Delta\phi = \phi_+ - \phi_-$ である。 $J[f]$ の最適化は (2.6) 式の制約下での $R[f] + \lambda_1 \langle f(\theta) \rangle = \langle fg \rangle$ の最適化となることは自明である。そして、このケースで (2.13) 式の $g(\theta)$ は (2.14) 式のように与えられる。

$$g(\theta) = \bar{Z}(\theta) + \lambda_1 \quad (2.14)$$

このとき、 $\|g\|_q$ は f の選択から独立である。つまり、この $\|g\|_q$ は $Z(\theta)$ と $\Delta\phi = \phi_+ - \phi_-$ と λ_1 にのみによって決定される。さらに、 $\|g\|_q$ と関係のあるパラメータ $\Delta\phi$ と λ_1 は $Z(\theta)$ のみによって決まる。 $\Delta\phi$ と λ_1 がどのように得られるかについては次節で説明する。

この節で、次の 3 つのケース (i) $1 < p < \infty$, (ii) $p = \infty$, (iii) $p = 1$ をそれぞれ分けて考える。

(i) $1 < p < \infty$ の場合

この場合、最適な周期信号 $f_{\text{opt},p}$ の候補は式 (2.15) のように構築できる。

$$f_{*,p}(\theta) = \left\{ \begin{array}{ll} M \left(\frac{|g(\theta)|}{\|g\|_q}\right)^{\frac{1}{p'}} & (g(\theta) \geq 0) \\ -M \left(\frac{|g(\theta)|}{\|g\|_q}\right)^{\frac{1}{p'}} & (g(\theta) \leq 0) \end{array} \right\} = M \text{sig}[g(\theta)] \left(\frac{|g(\theta)|}{\|g\|_q}\right)^{\frac{1}{p'}} \quad (2.15)$$

明らかに、この式 (2.15) は式 (2.12) を満たし、このときヘルダーの不等式の等式状態を満たす。もちろん、すでに式 (2.9) で $\|f\|_p = M$ を想定しているので、 $\|f_{*,p}\|_p = M$ は満たされる。しかし、もし式 (2.10) と式 (2.12) の負の部分よりもう一つ候補となり得る解を構築すると、つまり、もし $f_{*,p}$ を $-M \text{sig}[g(\theta)](g(\theta)/\|g\|_q)^{\frac{1}{p'}}$ とおくと、 $f_{*,p}(\theta)g(\theta)$ は常に負で式 (2.9) の最初の等号を満たさない。したがって、この候補は不要であり、式 (2.15) での $f_{*,p}$ は最適周期信号のための唯一つの候補となる。今、もしこの式 (2.15) での $f_{*,p}$ が存在し、式 (2.8) での $\|g\|_q$ が固定されるならば (つまり、 $\|g\|_q$ は f と独立)、この時 $f_{*,p}$ は関数 $\langle fg \rangle$ の唯一

の最大を与えることとなる．なぜなら， $\langle fg \rangle$ の上限は式 (2.15) での $f = f_{*,p}$ の時のみに達成される $\|f\|_p \|g\|_q = M \|g\|_q$ だからである．

一般に， f と g の周期は必ずしも一緒ではない．それらの比は $m:n$ になり，これは $m:n$ 同期のケースである．この時，この一般的な $m:n$ 同期での最適周期信号は式 (2.15) によって得られるわけではない．明らかに，この $m:n$ のケースは当たり前の $g(\theta) = 0$ のケースを除いて， $\alpha|f(\theta)|^p = \beta|g(\theta)|^q$ が $\theta \in S$ を反するので，成り立たない．したがって，式 (2.8) の最適周期信号は $m = n$ のケースである $1:1$ 同期だけであり， $m \neq n$ のケースでこの理想的な最適周期信号である式 (2.15) は得られない．

に， $p = \infty$ と $p = 1$ の場合への見識を与える，式 (2.15) での (a) $p' \rightarrow \infty$ の場合と (b) $p' \rightarrow +0$ の場合の2つの制限を考える． q が Z に関連するとき，以下は g で自然に想定される．

$$0 < \|g\|_q < \infty, \text{ と } 0 \leq |g(\theta)| < \infty \quad (\forall \theta \text{ in } S). \quad (2.16)$$

式 (2.16) より，(a) $p' \rightarrow \infty$ の場合で，式 (2.15) での $0 \leq |g(\theta)|/\|g\|_q$ と式 (2.17) を得る．

$$\left(\frac{|g(\theta)|}{\|g\|_q} \right)^{\frac{1}{p'}} \rightarrow \begin{cases} 1 & (\text{if } |g(\theta)| > 0) \\ 0 & (\text{if } |g(\theta)| = 0). \end{cases} \quad (p' \rightarrow \infty) \quad (2.17)$$

この式 (2.17) を踏まえると，式 (2.15) より式 (2.18) が得られる．

$$f_{*,p}(\theta) \rightarrow M \text{sig}[g(\theta)], \text{ for } \forall \theta \in S \quad (p \rightarrow \infty) \quad (2.18)$$

次に (b) の $p' \rightarrow +0$ の場合について考える．ここで $|g(\theta)|$ が最大化される S 内の点を θ_* と示すことにする．まず $\theta = \theta_*$ のケースと $\theta \neq \theta_*$ のケースを分けて考える．今，式 (2.15) での $|g(\theta)|/\|g(\theta)\|_q$ を $\bar{g}(\theta) = g(\theta)/|g(\theta_*)|$ と置くことによって変更する．

$$\left(\frac{|g(\theta)|}{\|g\|_q} \right)^{\frac{1}{p'}} = \left(\frac{|C\bar{g}(\theta)|}{\|C\bar{g}\|_q} \right)^{\frac{1}{p'}} = \frac{|\bar{g}(\theta)|^{\frac{1}{p'}}}{\langle |\bar{g}(\theta)|^{1+\frac{1}{p'}} \rangle^{\frac{1}{1+p'}}} = \left(\frac{|\bar{g}(\theta)|}{\|\bar{g}\|_q} \right)^{\frac{1}{p'}} \quad (2.19)$$

ここで， C は $|g(\theta_*)| (< \infty)$ を示す．明らかに，任意の p' で $|\bar{g}(\theta_*)|^{\frac{1}{p'}} = 1$ となる．一方， S の範囲で $|\bar{g}(\theta)| < 1$ であり， $\frac{1}{1+p'} \rightarrow 1 (p' \rightarrow +0)$ だから，この時

$\langle |\bar{g}(\theta)|^{1+\frac{1}{p'}} \rangle^{\frac{1}{1+p'}} \rightarrow +0 (p' \rightarrow +0)$ である．したがって，式 (2.19) から我々は

$\left(\frac{|g(\theta_*)|}{\|g\|_q} \right)^{\frac{1}{p'}} \rightarrow +\infty (p' \rightarrow +0)$ を得る．

一方，式 (2.19) での $\frac{|g(\theta)|}{\|g\|_q} = \frac{|\bar{g}(\theta)|}{\|\bar{g}\|_q} < 1$ は p' が 0 となる時に保たれるから， $\theta \neq \theta_*$ である．なぜなら， $\|\bar{g}\|_q \rightarrow \|\bar{g}\|_\infty = |\bar{g}(\theta_*)| (p' \rightarrow +0)$ で， $|\bar{g}(\theta)| < \|\bar{g}\|_\infty$ だからである．したがって，式

(2.15) より式 (2.20) を得る.

$$f_{*,p}(\theta) \rightarrow \begin{cases} +0 & (\text{if } \theta \neq \theta_*) \\ +\infty & (\text{if } \theta = \theta_*) \end{cases} \quad (p \rightarrow 1) \quad (2.20)$$

ここで, 式 (2.14) のように $g(\theta) = \bar{Z}(\theta) + \lambda_1$ として式 (2.15) の $f_{\text{opt},p}$ のように f を選んだら, ヘルダーの不等式が成立している前提なので式 (2.13) は最大化される.

$$f_{\text{opt},p}(\theta) = M \text{sig}[\bar{Z}(\theta) + \lambda_1] \left(\frac{|\bar{Z}(\theta) + \lambda_1|}{\|\bar{Z}(\theta) + \lambda_1\|_q} \right)^{\frac{1}{p'}} \quad (2.21)$$

ここで, p' は $p' = p - 1$ を示す.

そして, $\langle fg \rangle$ の上限は次のように与えられる.

$$\begin{aligned} \langle f_{\text{opt},p} g \rangle &\leq \langle |f_{\text{opt},p} g| \rangle = \|f_{\text{opt},p} g\| = \|f_{\text{opt},p}\|_p \|g\|_q \\ &= M \langle |\bar{Z}(\theta) + \lambda_1|^q \rangle^{\frac{1}{q}} = M \langle |\bar{Z}(\theta) + \lambda_1|^{\frac{p'+1}{p'}} \rangle^{\frac{p'}{p'+1}} \end{aligned} \quad (2.22)$$

このとき, 2つのパラメータ $\Delta\phi$ と λ_1 を調整することで得られる式 (2.22) の最大化は $\langle |\bar{Z}(\theta) + \lambda_1|^{\frac{p'+1}{p'}} \rangle$ を最大化することと等価である. なぜなら, M と p' は固定の正の定数と見なせるからである. このとき, 次の関数を定義する.

$$F(\Delta\phi, \lambda_1) \equiv \langle |\bar{Z}(\theta) + \lambda_1|^{\frac{p'+1}{p'}} \rangle = \langle |Z(\theta + \Delta\phi) - Z(\theta) + \lambda_1|^{\frac{p'+1}{p'}} \rangle \quad (2.23)$$

ここで, $\bar{Z}(\theta) = Z(\theta + \phi_+) - Z(\theta + \phi_-)$ で, $\Delta\phi = \phi_+ - \phi_-$ である. つまりロッキングレンジ $R[f]$ である $\langle fg \rangle$ を最大化することは $F(\Delta\phi, \lambda_1)$ を最大化することと同じである. また, 式 (2.21) から $\|\bar{Z}(\theta) + \lambda_1\|_q$ は Z と λ_1 によって与えられる定数なので, このとき $f_{\text{opt},p} \sim \text{sig}[\bar{Z}(\theta) + \lambda_1] [\bar{Z}(\theta) + \lambda_1]^{\frac{1}{p'}}$ だから, 式 (2.5) の制約は次のようになる.

$$\left\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^{\frac{1}{p'}} \right\rangle = 0 \quad (2.24)$$

ここで, 次のように外部周期信号 f の一周平均という意味をもつ関数 $G(\Delta\phi, \lambda_1)$ を定義する.

$$G(\Delta\phi, \lambda_1) \equiv \left\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^{\frac{1}{p'}} \right\rangle \quad (2.25)$$

本研究では $G(\Delta\phi, \lambda_1) = 0$ の制約下で $F(\Delta\phi, \lambda_1)$ を最大化したい. これを達成するために, ラグランジュの未定乗数法より次のような関数 $H(\Delta\phi, \lambda_1)$ を導く.

$$H(\Delta\phi, \lambda_1) \equiv F(\Delta\phi, \lambda_1) + \lambda G(\Delta\phi, \lambda_1) \quad (2.26)$$

λ はラグランジュの未定定数である．ここで，独立した最適解のみを持つ H を想定し，その導関数 $\frac{\partial^2 H}{\partial \Delta \phi^2}, \frac{\partial^2 H}{\partial \Delta \phi \partial \lambda_1}, \frac{\partial^2 H}{\partial \lambda_1 \partial \Delta \phi}, \frac{\partial^2 H}{\partial \lambda_1^2}$ は少なくとも最適解の周りで連続になる．そして， $\frac{p'+1}{p'}$ と $\frac{1}{p'}$ をそれぞれ次のように α と $\beta (= \alpha - 1)$ と省略することにする．

$$\alpha \equiv \frac{p'+1}{p'}, \quad \beta \equiv \frac{1}{p'} \quad (2.27)$$

$\left(\frac{\partial G}{\partial \Delta \phi}, \frac{\partial G}{\partial \lambda_1} \right) \neq \mathbf{0}$ だから，ラグランジュの未定乗数法のルールから λ が存在し，式 (2.26) の最適解 $(\Delta \phi_*, \lambda_{1,*})$ が存在するならば，この最適解は式 (2.28) を満たす．

$$\left(\frac{\partial H}{\partial \Delta \phi}, \frac{\partial H}{\partial \lambda_1} \right) = \mathbf{0} \quad (2.28)$$

つまり， $\Delta \phi, \lambda_1, \lambda$ のために式 (2.28) と $G(\Delta \phi, \lambda_1) = 0$ を解くことによって式 (2.26) の最適解の候補を決定することが出来る．これは次のように実行される．

式 (2.23) から， $F(\Delta \phi, \lambda_1)$ は次のように微分される．

$$\begin{aligned} \frac{\partial F}{\partial \Delta \phi} &= \frac{\partial}{\partial \Delta \phi} \langle |\bar{Z} + \lambda_1|^\alpha \rangle = \left\langle \frac{\partial}{\partial \Delta \phi} |\bar{Z} + \lambda_1|^\alpha \right\rangle \\ &= \alpha \left\langle |\bar{Z} + \lambda_1|^\beta \frac{\partial}{\partial \Delta \phi} |\bar{Z} + \lambda_1| \right\rangle \equiv a(\Delta \phi, \lambda_1) \end{aligned} \quad (2.29)$$

$$\begin{aligned} \frac{\partial F}{\partial \lambda_1} &= \frac{\partial}{\partial \lambda_1} \langle |\bar{Z} + \lambda_1|^\alpha \rangle = \left\langle \frac{\partial}{\partial \lambda_1} |\bar{Z} + \lambda_1|^\alpha \right\rangle \\ &= \alpha \langle |\bar{Z} + \lambda_1|^\beta \frac{\partial}{\partial \Delta \phi} |\bar{Z} + \lambda_1| \rangle = \alpha \langle \text{sig}(\bar{Z} + \lambda_1) |\bar{Z} + \lambda_1|^\beta \rangle \equiv b(\Delta \phi, \lambda_1) \end{aligned} \quad (2.30)$$

面白いことに，この $b = \alpha \langle \text{sig}(\bar{Z} + \lambda_1) |\bar{Z} + \lambda_1|^\beta \rangle = 0$ は $G(\Delta \phi, \lambda_1) = 0$ にすぎない．また， $G(\Delta \phi, \lambda_1)$ は次のように微分可能である．

$$\begin{aligned} \frac{\partial G}{\partial \Delta \phi} &= \frac{\partial}{\partial \Delta \phi} \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\alpha \rangle \\ &= \beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} Z'(\theta + \Delta \phi) \rangle \end{aligned} \quad (2.31)$$

$$\begin{aligned} \frac{\partial G}{\partial \lambda_1} &= \frac{\partial}{\partial \lambda_1} \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\alpha \rangle \\ &= \beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} \rangle > 0 \end{aligned} \quad (2.32)$$

今，式 (2.28) は $\frac{\partial F}{\partial \lambda_1} = \alpha G(\Delta \phi, \lambda_1) = 0$ と $\frac{\partial G}{\partial \lambda_1} > 0$ において $\frac{\partial H}{\partial \lambda_1} = \frac{\partial F}{\partial \lambda_1} + \lambda \frac{\partial G}{\partial \lambda_1} = 0$ を得る．このとき， λ は $\lambda = 0$ として唯一に決まる．これは λG の項が式 (2.26) で消えたとき，少し矛盾があるように見える．しかし，この $\lambda = 0$ で式 (2.28) は $a(\Delta \phi, \lambda_1) = b(\Delta \phi, \lambda_1) = 0$ となり，そして $a = 0$ と $b = 0$ の解はラグランジュ乗数のルールを否定しなければ，式 (2.5) の制約を自動的に満たすしたがって，最適解候補は $a = b = 0$ より簡単に決定される¹．

¹もし， $\frac{\partial F}{\partial \lambda_1} = 0$ は制約の $G = 0$ にすぎないという性質を利用すれば，問題はこの特定の設定で F の最適化に縮約される．ここで，一般的に F の代わりに H の最適化を行う

次に、様々な $(\Delta\phi, \lambda_1)$ のペアの最適ではない解から最適解を見分けるために、 H の境界付きのヘッセ行列を考える。

$$\mathcal{H}(H) = \begin{bmatrix} 0 & \mathcal{H}_{12} & \mathcal{H}_{13} \\ \mathcal{H}_{21} & \mathcal{H}_{22} & \mathcal{H}_{23} \\ \mathcal{H}_{31} & \mathcal{H}_{32} & \mathcal{H}_{33} \end{bmatrix} \quad (2.33)$$

このヘッセ行列の各要素は次のように与えられる。

$$\mathcal{H}_{12} = \mathcal{H}_{21} = \frac{\partial G}{\partial \Delta\phi} = \beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} Z'(\theta + \Delta\phi) \rangle \quad (2.34a)$$

$$\mathcal{H}_{13} = \mathcal{H}_{31} = \frac{\partial G}{\partial \lambda_1} = \beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} \rangle > 0 \quad (2.34b)$$

$$\mathcal{H}_{22} = \frac{\partial^2 F}{\partial \Delta\phi^2} = \alpha\beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} Z'(\theta + \Delta\phi)^2 \rangle + \alpha \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta Z''(\theta + \Delta\phi) \rangle \quad (2.34c)$$

$$\mathcal{H}_{23} = \frac{\partial^2 F}{\partial \Delta\phi \partial \lambda_1} = \alpha\beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} Z'(\theta + \Delta\phi) \rangle = \alpha\mathcal{H}_{12} \quad (2.34d)$$

$$\begin{aligned} \mathcal{H}_{32} &= \frac{\partial^2 F}{\partial \lambda_1 \partial \Delta\phi} \\ &= \alpha \frac{\partial}{\partial \Delta\phi} \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta \rangle = \alpha\beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} Z'(\theta + \Delta\phi) \rangle = \alpha\mathcal{H}_{12} \end{aligned} \quad (2.34e)$$

$$\mathcal{H}_{33} = \frac{\partial^2 F}{\partial \lambda_1^2} = \alpha \frac{\partial}{\partial \lambda_1} \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta \rangle = \alpha\beta \langle |\bar{Z}(\theta) + \lambda_1|^{\beta-1} \rangle = \alpha\mathcal{H}_{13} > 0 \quad (2.34f)$$

このとき、ヘッシアン $|\mathcal{H}(H)|$ は式 (2.35) のように得られ、

$$|\mathcal{H}(H)| = \mathcal{H}_{13}(\alpha\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22}) \quad (2.35)$$

$|\mathcal{H}(H)| > 0$ を満たすとき $a = b = 0$ となる解が最大となり、もし $|\mathcal{H}(H)| < 0$ ならば最小となる

以上のことをまとめると、式 (2.23) での最適解 $(\Delta\phi, \lambda_1)$ は次の条件を満たすときに存在する。

$$\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta Z'(\theta + \Delta\phi) \rangle = 0 \quad (2.36)$$

$$\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta \rangle = 0 \quad (2.37)$$

$$|\mathcal{H}(H)| = \mathcal{H}_{13}(\alpha\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22}) > 0 \quad (2.38)$$

ここで、式 (2.36) と式 (2.37) の積分内の $\bar{Z}(\theta) = Z(\theta + \phi_+) - Z(\theta + \phi_-)$ は $Z(\theta + \Delta\phi) - Z(\theta)$ と等価である。今、この最適解において次の式 (2.39) の不等式を満たさなければいけないことに注意する。

$$\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta Z''(\theta + \Delta\phi) \rangle < 0 \quad (2.39)$$

ここで、 Γ を特徴づけることが重要である。

つまり、式 (2.32) の $|\bar{Z}(\theta) + \lambda_1|^{\beta-1}$ に関する積分は $2 < p < \infty$ 、言い換えると

$0 < \beta = \frac{1}{p-1} < 1$ である時に特異になることに注意する。なぜなら、 $\bar{Z}(\theta) + \lambda_1$ は θ_* で 0 になり、 $|\bar{Z}(\theta) + \lambda_1|^{\beta-1}$ は $\theta = \theta_*$ で無限大となるからである。しかし、もし $\bar{Z}(\theta)$ が有限調波を用いたフーリエ級数によって表現され、例として $\bar{Z}'(\theta) \neq 0$ であると想定すると、このような積分は全て有限な値を持つ。

続いて、式 (2.36),(2.37),(2.39) の解の対称性を考える。まず、式 (2.23) から

$F(\Delta\phi, \lambda_1) = F(-\Delta\phi, -\lambda_1)$ と $G(\Delta\phi, \lambda_1) = G(-\Delta\phi, -\lambda_1) = 0$ は任意の $\Delta\phi$ と λ_1 で保持されることを注意する。このとき、もし $(\Delta\phi_*, \lambda_{1,*})$ は式 (2.36) と式 (2.37) の解であるならば、

$(-\Delta\phi_*, -\lambda_{1,*})$ もまた式 (2.36) と式 (2.37) の解である。そして、両方の解

$(\Delta\phi, \lambda_1) = (\pm\Delta\phi_*, \pm\lambda_{1,*})$ において $g(\theta), f_{\text{opt},p}(\theta)$ はそれぞれ次のように与えられる。

$$g(\theta) = \pm[\bar{Z}(\theta) + \lambda_{1,*}] \quad (2.40a)$$

$$f_{\text{opt},p}(\theta) = \pm M \text{sig}[\bar{Z}(\theta) + \lambda_1] \left(\frac{|\bar{Z}(\theta) + \lambda_{1,*}|}{\|\bar{Z}(\theta) + \lambda_{1,*}\|_q} \right)^{\frac{1}{p}} \quad (2.40b)$$

以上より、 $1 < p < \infty$ の場合のロックレンジを最大化する最適周期信号 $f_{\text{opt},p}(\theta)$ を理論的に導出することができる。

(ii) $p = \infty$ の場合

$p = \infty$ の場合を考える。まず、 $\|f\|_\infty = M$ となるので、次の式が得られる。

$$\|fg\|_1 \leq \|f\|_\infty \|g\|_1 = M \|g\|_1 \quad (2.41)$$

この時、定数 $\|g\|_1$ で $\|fg\|_1$ を最大化するために、 $f_{*,\infty}$ が何かを定める。 $f_{*,\infty}$ の候補は簡単に以下のように見つかる。

$$f_{*,\infty}(\theta) = M \text{sig}[g(\theta)] \quad (2.42)$$

これは、式 (2.18) にすぎず、 $p \rightarrow \infty$ の場合で得られる。実際、この特定の $f_{*,p}$ で $\|fg\|_1$ は最大化される。なぜなら、式 (2.43) のようになるからである。

$$\|f_{*,\infty}g\|_1 = \langle M \text{sig}[g(\theta)] g(\theta) \rangle = M \langle \text{sig}[g(\theta)] g(\theta) \rangle = M \langle |g(\theta)| \rangle = M \|g\|_1 \quad (2.43)$$

また、この候補 $f_{*,\infty}$ の一意性は次のように証明される。 $\|f_{*,\infty}g\|_1 = M \|g\|_1$ を最大化するもう一つの候補 $\bar{f}_{*,\infty}$ が存在すると想定する。この時、式 (2.44) は任意の g で満たされる。

$$\|f_{*,\infty}g\|_1 - \|\bar{f}_{*,\infty}g\|_1 = 0 \quad (2.44)$$

そして、式 (2.44) は式 (2.45) と等価である.

$$\left\langle \left(1 - \frac{|\bar{f}_{*,\infty}|}{M}\right) f_{*,\infty} g \right\rangle = 0 \quad (2.45)$$

なぜなら,

$$\begin{aligned} |f_{*,\infty}(\theta)g(\theta)| - |\bar{f}_{*,\infty}(\theta)g(\theta)| &= f_{*,\infty}(\theta)g(\theta) - |\bar{f}_{*,\infty}(\theta)||g(\theta)| \quad ((\dagger) \text{ の式より}) \\ &= f_{*,\infty}(\theta)g(\theta) - \frac{|\bar{f}_{*,\infty}(\theta)|}{M} f_{*,\infty}(\theta)g(\theta) \quad ((\ddagger) \text{ の式(2.46)}) \end{aligned}$$

だからである. ここで、この式 (2.46) では以下の関係式を使った.

$$f_{*,\infty}(\theta)g(\theta) = M \text{sig}[g(\theta)] \cdot g(\theta) \geq 0 \quad (\dagger)$$

$$|g(\theta)| = f_{*,\infty}(\theta)g(\theta)/M \quad (\ddagger)$$

したがって、(2.45) から式 (2.47) を得る.

$$|\bar{f}_{*,\infty}(\theta)| = M \quad \text{a.e. in } S \quad (2.47)$$

しかし、 M か $-M$ の値のみを持つこのような関数で、 $f = M \text{sig}[g(\theta)] (= f_{*,\infty})$ のみが $\langle fg \rangle$ を最大化する. したがって、 $\bar{f}_{*,\infty}$ は存在できず、最適な $f_{*,\infty}$ の一意性は証明された.

$1 < p < \infty$ の場合と同様に、この一意的な最適周期信号の式 (2.42) は 1 : 1 同期の場合のみ有効である.

これより $p = \infty$ の場合、式 (2.13) は $g(\theta) = \bar{Z}(\theta) + \lambda_1$ を用いて式 (2.48) によって最大化される.

$$f_{\text{opt},\infty}(\theta) = M \text{sig}[\bar{Z}(\theta) + \lambda_1] \quad (2.48)$$

$\langle fg \rangle$ の上限は次のように与えられる.

$$\langle f_{\text{opt},\infty} g \rangle = M \langle |\bar{Z}(\theta) + \lambda_1| \rangle \quad (2.49)$$

このとき、 $F(\Delta\phi, \lambda_1)$ と $G(\Delta\phi, \lambda_1)$ を次のように定義する.

$$F(\Delta\phi, \lambda_1) \equiv \langle |\bar{Z}(\theta) + \lambda_1| \rangle \quad (2.50)$$

$$G(\Delta\phi, \lambda_1) = \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] \rangle \quad (2.51)$$

$G(\Delta\phi, \lambda_1) = 0$ は式 (2.6) と式 (2.17) から得られる制約である. このとき、 $H(\Delta\phi, \lambda_1)$ を式 (2.26) のように導入し、 H に同じ想定をおく. $(\frac{\partial G}{\partial \Delta\phi}, \frac{\partial G}{\partial \lambda_1}) \neq \mathbf{0}$ だから、 λ は存在し、

$\left(\frac{\partial H}{\partial \Delta\phi}, \frac{\partial H}{\partial \lambda_1}\right) = \mathbf{0}$ は最適解 $(\Delta\phi_*, \lambda_{1,*})$ で満たされる。これより、 $1 < p < \infty$ の場合と同じ手順で次のように $\lambda_{1,*}$, $\Delta\phi_*$, λ を決定する。

まず、 $F(\Delta\phi, \lambda_1)$ は次のように微分される。

$$\frac{\partial F}{\partial \Delta\phi} = \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z'(\theta + \Delta\phi) \rangle \quad (2.52)$$

$$\frac{\partial F}{\partial \lambda_1} = \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] \rangle = G(\Delta\phi, \lambda_1) \quad (2.53)$$

そして、 $G(\Delta\phi, \lambda_1)$ は次のように微分される。

$$\frac{\partial G}{\partial \Delta\phi} = \frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)}{|\bar{Z}'(\theta_i)|} \quad (2.54)$$

$$\frac{\partial G}{\partial \lambda_1} = \frac{1}{\pi} \sum_{i=1}^n \frac{1}{|\bar{Z}'(\theta_i)|} > 0 \quad (2.55)$$

ここで、 θ_i は $\bar{Z}(\theta) + \lambda_1 = 0$ の i 番目の根を表し、 n は $\bar{Z}(\theta)$ が周期関数であることから偶数となる根の数である。このとき、 λ は $\lambda = 0$ と決定される。そして、 $\Delta\phi$ と λ_1 は $\langle \text{sig}(\bar{Z} + \lambda_1) Z'(\theta + \Delta\phi) \rangle = 0$ と $\langle \text{sig}(\bar{Z} + \lambda_1) \rangle = 0$ によって決定される。境界付きヘッセ行列は次のように得られる。

$$\mathcal{H}_{12} = \mathcal{H}_{21} = \frac{\partial G}{\partial \Delta\phi} = \frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)}{|\bar{Z}'(\theta_i)|} \quad (2.56a)$$

$$\mathcal{H}_{13} = \mathcal{H}_{31} = \frac{\partial G}{\partial \lambda_1} = \frac{1}{\pi} \sum_{i=1}^n \frac{1}{|\bar{Z}'(\theta_i)|} > 0 \quad (2.56b)$$

$$\mathcal{H}_{22} = \frac{\partial^2 F}{\partial \Delta\phi^2} = \frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)^2}{|\bar{Z}'(\theta_i)|} + \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z''(\theta + \Delta\phi) \rangle \quad (2.56c)$$

$$\mathcal{H}_{23} = \frac{\partial^2 F}{\partial \Delta\phi \partial \lambda_1} = \frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)}{|\bar{Z}'(\theta_i)|} = \mathcal{H}_{12} \quad (2.56d)$$

$$\mathcal{H}_{32} = \frac{\partial^2 F}{\partial \lambda_1 \partial \Delta\phi} = \frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)}{|\bar{Z}'(\theta_i)|} = \mathcal{H}_{12} \quad (2.56e)$$

$$\mathcal{H}_{33} = \frac{\partial^2 F}{\partial \lambda_1^2} = \frac{1}{\pi} \sum_{i=1}^n \frac{1}{|\bar{Z}'(\theta_i)|} = \mathcal{H}_{13} > 0 \quad (2.56f)$$

したがって、(2.50) 式 of 最適解 $(\Delta\phi, \lambda_1)$ は次の条件を満たすときに存在する。

$$\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z'(\theta + \Delta\phi) \rangle = 0 \quad (2.57a)$$

$$\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] \rangle = 0 \quad (2.57b)$$

$$|\mathcal{H}(H)| = \mathcal{H}_{13}(\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22}) > 0 \quad (2.57c)$$

今、我々はもし (2.57c) が満たされるならば、次の式が成り立つことを示す。

$$\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z''(\theta + \Delta\phi) \rangle < 0 \quad (2.58)$$

$\mathcal{H}_{13} > 0$ だから、 $\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22}$ が正となる条件を考える。まず、次の2つの式を得る。

$$\mathcal{H}_{12}^2 = \left[\frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)}{|\bar{Z}'(\theta_i)|} \right]^2 \quad (2.59)$$

$$\mathcal{H}_{13}\mathcal{H}_{22} = \left[\frac{1}{\pi} \sum_{i=1}^n \frac{1}{|\bar{Z}'(\theta_i)|} \right] \left[\frac{1}{\pi} \sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)^2}{|\bar{Z}'(\theta_i)|} + \langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z''(\theta + \Delta\phi) \rangle \right] \quad (2.60)$$

そして、コーシー・シュワルツの不等式から、次の式が成り立つ。

$$\left[\sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)}{|\bar{Z}'(\theta_i)|} \right]^2 - \left[\sum_{i=1}^n \frac{1}{|\bar{Z}'(\theta_i)|} \right] \left[\sum_{i=1}^n \frac{Z'(\theta_i + \Delta\phi)^2}{|\bar{Z}'(\theta_i)|} \right] = (X \cdot Y)^2 - |X|^2 |Y|^2 < 0$$

ここで、 $X = (1/\sqrt{|\bar{Z}'(\theta_1)|}, \dots, 1/\sqrt{|\bar{Z}'(\theta_n)|})$ で、

$Y = (Z'(\theta_1 + \Delta\phi)/\sqrt{|\bar{Z}'(\theta_1)|}, \dots, Z'(\theta_n)/\sqrt{|\bar{Z}'(\theta_n)|})$ である。したがって、

(2.57c), (2.59), (2.60) 式から、 $\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z''(\theta + \Delta\phi) \rangle < 0$ は満たされなければならない。

今、 $(\Delta\phi, \lambda_1) \leftrightarrow (-\Delta\phi, -\lambda_1)$ 下で F と G はそれぞれ対称であることは明らかである。このとき、 Γ は (2.57a), (2.57a), (2.57c) 式を満たす最適解のために定義される。

以上より、 $p = \infty$ の場合の最適周期信号 $f_{\text{opt}, \infty}$ を理論的に導出することができる。

(iii) $p = 1$ の場合

式 (2.41) と同様に、式 (2.5) から f は $\|f\|_1 = M$ を満たすので、式 (2.61) を得る。

$$\|fg\|_1 \leq \|f\|_1 \|g\|_\infty = M \|g\|_\infty \quad (2.61)$$

この $p = 1$ の場合で、我々は f が前出の $p > 1$ のケースでは許されない任意のパルスのような鋭い関数になることに注意する。もし、この $p = 1$ の場合で任意の理想の候補 $f_{\text{ideal}, 1}$ が存在するならば、 $f_{\text{ideal}, 1}$ は次のように満たされるべきである。

$$\langle f_{\text{ideal}, 1} g \rangle \leq \langle |f_{\text{ideal}, 1} g| \rangle = \|f_{\text{ideal}, 1} g\|_1 = \|f_{\text{ideal}, 1}\|_1 \|g\|_\infty = M \|g\|_\infty \quad (2.62)$$

ここで、 $\langle f_{*, 1} f \rangle = \|f_{*, 1} g\| \rightarrow M \|g\|_\infty$ を満たす式 (2.63) のような $f_{*, 1}$ を構築する。

$$f_{*, 1}(\theta) = M \sum_{i=1}^n \text{sig}[g(\theta_{*, i})] \Delta(\theta - \theta_{*, i}), \quad (2.63)$$

ここで、 $\theta_{*,i}$ は $|g(\theta)|$ を最大化する i 番目の点を示す。(言い換えると、 $1 \leq i, j \leq n$ で $|g(\theta_{*,i})| = |g(\theta_{*,j})| = |g(\theta_*)|$ となる。) そして、 $\Delta(\theta - \theta_{*,i})$ は次のように定義される鋭いパルスである。

$$\Delta(\theta - \theta_{*,i}) = \begin{cases} \frac{1}{2n\epsilon} & \text{if } |\theta - \theta_{*,i}| \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (2.64)$$

ここで、 n は $\theta_{*,i} \in S$ の番号を示す。この $f_{*,1}$ は制約式 (2.5) より $\|f_{*,1}\| = M$ を満たし、
 $\langle f_{*,1} | g \rangle = M \left\langle \sum_{i=1}^n \text{sig}[g(\theta_{*,i})] \Delta(\theta - \theta_{*,i}) g(\theta) \right\rangle \rightarrow M \sum_{i=1}^n |g(\theta_*)|$ だから、
 $\langle \Delta(\theta - \theta_{*,i}) g(\theta) \rangle = g(\theta_{*,i})$ より $\langle f_{*,1} | g \rangle \rightarrow M \|g\|_\infty$ ($\epsilon \rightarrow +0$) である。

$1 < p < \infty$ の場合と $p = \infty$ の場合では導出した最適周期信号は 1:1 同期のみで実現される。対照的に、この $p = 1$ の場合では、式 (2.63) の任意の“良い”周期信号の種類は $m \neq n$ のケースでさえも実現される。この特性は実用の応用にとって特に重要である。

これより、 $\epsilon \rightarrow 0$ の制限での式 (2.63) から我々は次の式を得る。

$$\begin{aligned} \Gamma(\phi) &= \langle Z(\theta + \phi) f_{*,1}(\theta) \rangle = \langle Z(\theta + \phi) M \sum_{i=1}^n \text{sig}[g(\theta_{*,i})] \Delta(\theta - \theta_{*,i}) \rangle \\ &\rightarrow M \sum_{i=1}^n \text{sig}[g(\theta_{*,i})] Z(\theta_{*,i} + \phi) \quad (\text{均一に } \phi \in S) \end{aligned} \quad (2.65)$$

このとき、 $\epsilon \rightarrow 0$ の制限での $R[f]$ は式 (2.65) の $M \sum_{i=1}^n \text{sig}[g(\theta_{*,i})] Z(\theta_{*,i} + \phi)$ の最大値 - 最小値によって与えられる。ここで、式 (2.65) での均一の収束は $g(\theta)$ が連続的に微分可能であるという事実と式 (2.64) から簡単に証明されることに注意する。

もし制約 $\langle f_{*,1} | \rangle = 0$ が課されれば、 $g(\theta) = \bar{Z}(\theta) + \lambda_1$ は式 (2.14) からわかる。ここで、一般性の欠落を除いて、 $\bar{Z}(\theta)$ は $\theta = \theta_{\max}, \theta_{\min}$ でそれぞれ最大値と最小値をもつと想定する。このとき、この特定の $\bar{Z}(\theta)$ で、 $\theta = \theta_{\max}, \theta_{\min}$ でそれぞれ $g(\theta)$ の最大値と最小値のペアとなるような θ において、 $\lambda_{1,*} = -\frac{1}{2}[\bar{Z}(\theta_{\max}) - \bar{Z}(\theta_{\min})]$ と置けることは明らかである。ここで、 $g(\theta_{\min}) < 0 < g(\theta_{\max})$ を満たし、 $|g(\theta_{\min})| = |g(\theta_{\max})|$ である。

このとき式 (2.64) から、この $\lambda_{1,*}$ は次の $f_{*,1}(\theta)$ を与える。

$$f_{*,1}(\theta) = M[\Delta(\theta - \theta_{\max}) - \Delta(\theta - \theta_{\min})], \quad (2.66)$$

そして、この特定の $f_{*,1}$ で次の式が得られる。

$$\Gamma(\phi) \rightarrow M[Z(\theta_{\max} + \phi) - Z(\theta_{\min} + \phi)] \equiv \Gamma_0(\phi) \quad (\text{均一に } \epsilon \rightarrow 0) \quad (2.67)$$

今、式 (2.67) の周期信号で引き込まれる範囲の最大化は、 $R = (\Gamma_0 \text{ の最大値}) - (\Gamma_0 \text{ の最小値})$ であるロッキングレンジを最大化するための $\Delta\theta = \theta_{\max} - \theta_{\min}$ を見つけることとなる。これは $\Delta\theta = \theta_{\max} - \theta_{\min}$ と $\bar{\phi} = \phi + \theta_{\min}$ で $\Gamma_0(\bar{\phi}) = M[Z(\bar{\phi} + \Delta\theta) - Z(\bar{\phi})]$ と置くことと、様々な $\Delta\theta \in [-\pi, \pi]$ の値での $\Gamma_0(\bar{\phi})$ のグラフを描くことによって任意に与えた $Z(\theta)$ で数値的に分かる。 $\Delta\theta$ に関連する R のグラフの結果は R を最大化する $\Delta\theta$ を見つける。

つまり、次のように $\Delta\theta$ の項で $\Delta\phi$ を特徴づけることはできる。まず、その定義によって我々は $\Delta\phi$ は $\Gamma_0(\phi) \sim Z(\theta_{\max} + \phi) - Z(\theta_{\min} + \phi)$ の (最大点; ϕ_+) - (最小点; ϕ_-) であると思い出す。そして、最大のロッキングレンジ R_{\max} は

$R_{\max} \sim Z(\theta_{\max} + \phi_+) - Z(\theta_{\min} + \phi_+) - Z(\theta_{\max} + \phi_-) + Z(\theta_{\min} + \phi_-)$ のとき得られる。

一方、すでに $\bar{Z}(\theta) = Z(\theta + \phi_+) - Z(\theta + \phi_-)$ は $\theta = \theta_{\max}$ と $\theta = \theta_{\min}$ でそれぞれ最大点と最小点をもつことを見ている。このとき、 $(\bar{Z} \text{ の最大値}) - (\bar{Z} \text{ の最小値}) \equiv \bar{R}$ は

$\bar{R} = Z(\theta_{\max} + \phi_+) - Z(\theta_{\max} + \phi_-) - Z(\theta_{\min} + \phi_+) + Z(\theta_{\min} + \phi_-)$ によって与えられる。これは、上記で得られた R_{\max} と等価である。そして、もし $\Delta\phi = \phi_+ - \phi_-$ と $\bar{\theta} = \theta + \phi_-$ をおくと、 $\bar{Z}(\theta)$ は $\bar{Z}(\bar{\theta}) = Z(\bar{\theta} + \Delta\phi) - Z(\bar{\theta})$ と等価である。もし $\bar{\phi}$ が $\bar{\theta}$ で置き換えられたら、 $\bar{Z}(\bar{\theta})$ は $\Gamma_0(\bar{\phi})$ にすぎないことに注意する。加えて、一般的な Z で $R_{\max}(\Delta\phi)$ が最大化される唯一の $\Delta\phi$ があると想定される。したがって、上記の事実から、 $\Gamma_0(\bar{\phi})$ での $\Delta\theta$ と $\bar{Z}(\bar{\theta})$ での $\Delta\phi$ は一致すべきで、その関連した $\bar{\phi}_{\pm}$ と $\bar{\theta}_{\max, \min}$ もまたそれぞれ一致する。

以上より、 $p = 1$ の場合の最適周期信号 $f_{*,1}(\theta)$ を理論的に導出することができる。

2.2.2 Hodgkin-Huxley 振動子の最適周期信号の理論的導出

実際に、2.2.1 節で説明したヘルダーの不等式による最適化アルゴリズムを用いて、ロッキングレンジを最大化する最適周期信号を理論的に導出する。対象とする振動子は

Hodgkin-Huxley 振動子とする。Hodgkin-Huxley 振動子の位相応答関数 $Z(\theta)$ はインパルス応答法より求めた以下の式 (2.68) を用いる。

$$Z(\theta) = \frac{a_0}{2} + \sum_{n=1}^4 a_n \cos(n\theta) + b_n \sin(n\theta) \quad (2.68)$$

n	a_n	b_n
0	0.352231	–
1	0.371736	-0.740283
2	-0.819478	0.00225226
3	0.181875	0.403816
4	0.111464	-0.0892503

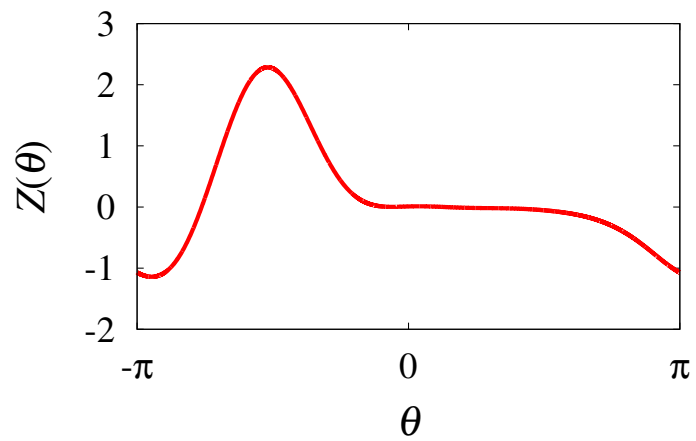


図 2.2: Hodgkin-Huxley 振動子の位相応答曲線

1 < p < ∞ の場合

1 < p < ∞ の場合について考える．最適波形 $f_{\text{opt},p}$ を得るには，まず (2.21) 式内の 2 つのパラメータ $(\Delta\phi, \lambda_1)$ を決定する必要がある．そのために，(2.36) 式と (2.37) 式の連立方程式を解き，2 つのパラメータ $(\Delta\phi, \lambda_1)$ の解候補 $(\Delta\phi_*, \lambda_{1,*})$ を求める．このとき，解候補 $(\Delta\phi_*, \lambda_{1,*})$ が (2.39) 式を満たすかを確認する．(2.36),(2.37) 式の解の全体像を知るために， $(\Delta\phi, \lambda_1)$ をパラメータをある範囲で変動させ，(2.36) 式と (2.37) 式のそれぞれについて解曲線を描く．そして，(2.36) 式と (2.37) 式より得られた 2 つの解曲線の交点を全て記録する．この記録した交点が (2.36) 式と (2.37) 式の連立方程式の解であり，今得たい解候補 $(\Delta\phi_*, \lambda_{1,*})$ となる．Hodgkin-Huxley 振動子を対象とした際の (2.36) 式と (2.37) 式の解曲線を図 2.3, 2.4, 2.5 に示す．p の値は $p = 1.01, 2, 5$ の 3 通りとした．ここで，(2.36) 式と (2.37) 式において， $\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta Z'(\theta + \Delta\phi) \rangle \equiv S_p(\Delta\phi, \lambda_1)$ ， $\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] |\bar{Z}(\theta) + \lambda_1|^\beta \rangle \equiv T_p(\Delta\phi, \lambda_1)$ と置くことにする．図 2.3, 2.4, 2.5 を見て分かるように， $T_p(\Delta\phi, \lambda_1)$ は $\Delta\phi$ の連続関数であり， $S_p(\Delta\phi, \lambda_1)$ は λ_1 の連続関数である．また本研究では， $f_{\text{opt},p}$ は周期 2π の周期関数であると仮定しているので， $\Delta\phi \in [-\pi, \pi]$ として，解曲線を描いた．

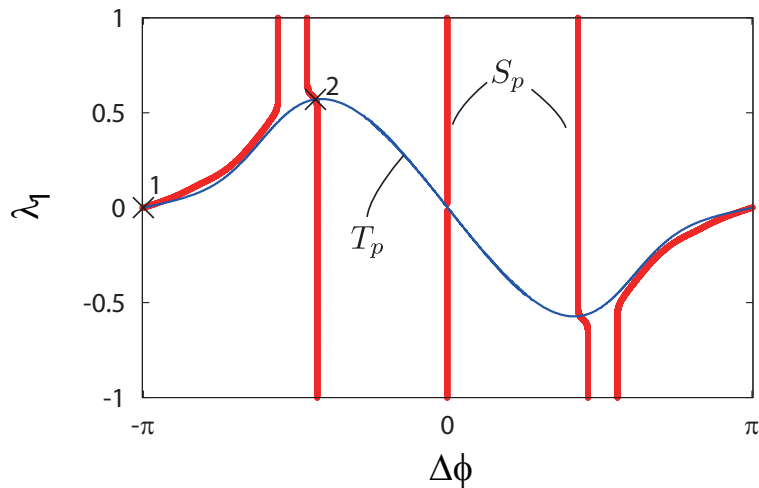


図 2.3: (2.37) 式の T_p と (2.36) 式の S_p の解曲線 ($p = 1.01$)

(2.37) 式の T_p と (2.36) 式の S_p の解曲線の交点より得られる解候補 $(\Delta\phi_*, \lambda_{1,*})$ をより正確なものとするために，ニュートン法を用いた．ニュートン法で使用する (2.36) 式と (2.37) 式に対するヤコビ行列はそれぞれ (2.34) 式である．(2.34) 式を用いてニュートン法を解く際に， $(\Delta\phi, \lambda_1)$ の初期値が必要となるが，この初期値は (2.37) 式の T_p と (2.36) 式の S_p の解曲線の

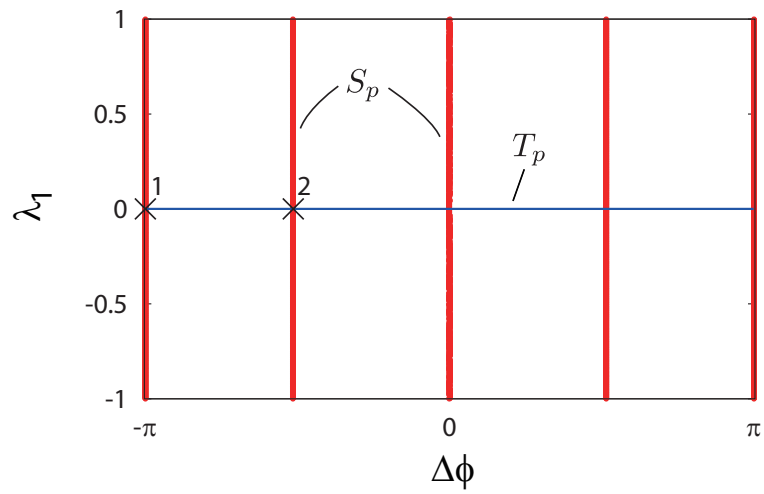


図 2.4: (2.37) 式の T_p と (2.36) 式の S_p の解曲線 ($p = 2$)

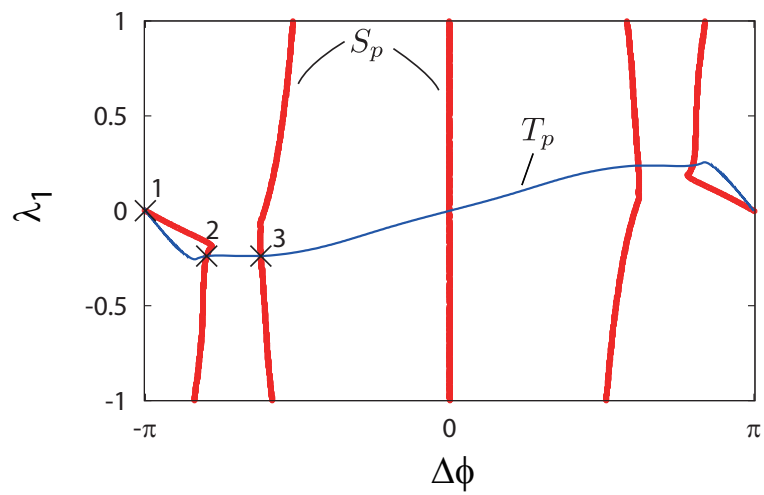


図 2.5: (2.37) 式の T_p と (2.36) 式の S_p の解曲線 ($p = 5$)

交点である $(\Delta\phi, \lambda_1)$ とした. ニュートン法より得られた解候補 $(\Delta\phi_*, \lambda_{1,*})$ のリストを表 2.1, 2.2, 2.3 に示す. ここで, 2.2.1 節で述べたように $(\Delta\phi, \lambda_1) \leftrightarrow (-\Delta\phi, -\lambda_1)$ と対称性を持つ.

次にニュートン法より得られた全ての解候補 $(\Delta\phi_*, \lambda_{1,*})$ について, $M = 1$ とし (2.22) 式よりロッキングレンジ R と (2.38) 式より $|\mathcal{H}(H)|$ を求める. ロッキングレンジ R と $|\mathcal{H}(H)|$ を求めた結果を解候補 $(\Delta\phi_*, \lambda_{1,*})$ と並べて表 2.1, 2.2, 2.3 に示す. また, $\Delta\phi \in [-\pi, \pi]$ に対する T_p の全ての解について, $M = 1$ とし (2.22) 式よりロッキングレンジ $R(\Delta\phi)$ を求めた結果を図 2.6, 2.7, 2.8 に示す.

表 2.1: (2.36) 式と (2.37) 式を満たす解 $(\Delta\phi, \lambda_1)$ とロッキングレンジ $R(p = 1.01)$

	$(\Delta\phi, \lambda_1)$	R	$ \mathcal{H}(H) = \mathcal{H}_{13}(\alpha\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22})$
solution 1	$(-3.1415927, 0.0)$	2.2585700	$-6.4861331 \times 110^{110} (< 0)$
solution 2	$(-1.3649363, 0.57040837)$	2.7613265	$5.7290424 \times 10^{137} (> 0)$

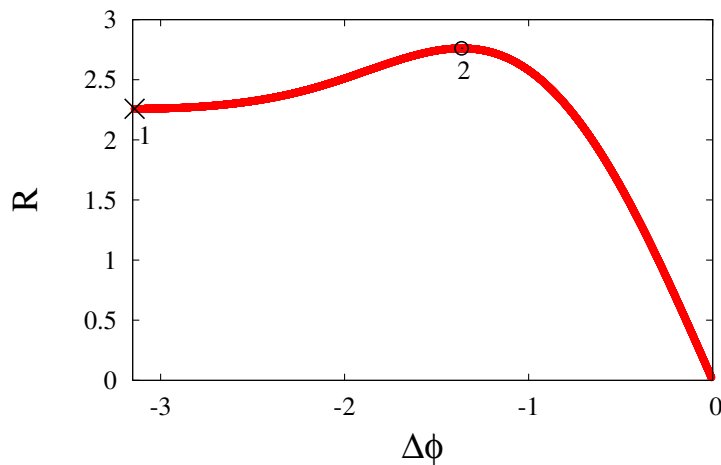


図 2.6: T_p における $R(\Delta\phi)$ を求めた結果 ($p = 1.01$)

$p = 1.01$ の場合, solution2 の $(\Delta\phi, \lambda_1) \sim (-1.364936336, 0.5704083690)$ は $|\mathcal{H}(H)| > 0$ であるため, $F(\Delta\phi, \lambda_1)$ の極大値, つまり $R[f]$ の極大値となる. 他に $R[f]$ の極大値となる解候補はないため, 最適解は solution2 となる. また, solution1 の $(\Delta\phi, \lambda_1) \sim (-3.1415927, 0.0)$ は $|\mathcal{H}(H)| < 0$ であるため, $F(\Delta\phi, \lambda_1)$ の極小値, つまり $R[f]$ の極小値となり, 今得たい最適解ではない. これらの関係は図 2.6 を見ても明らかである.

$p = 2$ の場合, solution2 の $(\Delta\phi, \lambda_1) \sim (-1615065266, 0.0)$ は $|\mathcal{H}(H)| > 0$ であるため,

表 2.2: (2.36) 式と (2.37) 式を満たす解 $(\Delta\phi, \lambda_1)$ とロッキングレンジ $R(p=2)$

	$(\Delta\phi, \lambda_1)$	R	$ \mathcal{H}(H) = \mathcal{H}_{13}(\alpha\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22})$
solution 1	$(-3.1415927, 0.0)$	1.3287487	$-0.54999893 (< 0)$
solution 2	$(-1.6150653, 0.0)$	1.4926698	$2.153595723 (> 0)$

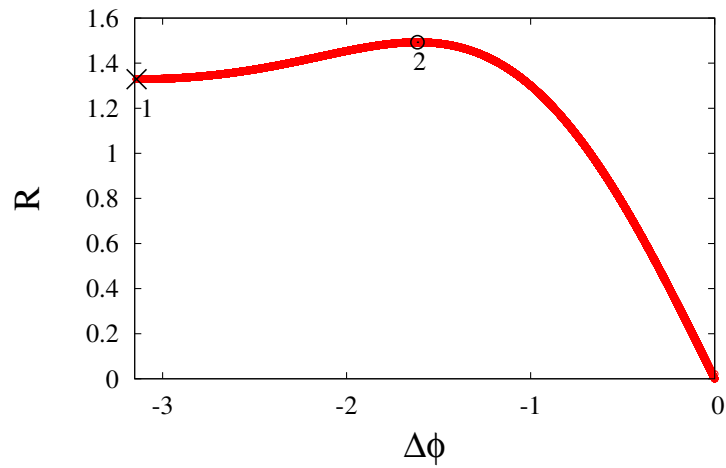


図 2.7: T_p における $R(\Delta\phi)$ を求めた結果 ($p=2$)

$F(\Delta\phi, \lambda_1)$ の極大値, つまり $R[f]$ の極大値となる. 他に $R[f]$ の極大値となる解候補はないため, 最適解は solution2 となる. また, solution1 の $(\Delta\phi, \lambda_1) \sim (-3.1415927, 0.0)$ は $|\mathcal{H}(H)| < 0$ であるため, $F(\Delta\phi, \lambda_1)$ の極小値, つまり $R[f]$ の極小値となり, 今得たい最適解ではない. これらの関係は図 2.7 を見ても明らかである.

表 2.3: (2.36) 式と (2.37) 式を満たす解 $(\Delta\phi, \lambda_1)$ とロッキングレンジ $R(p = 5)$

	$(\Delta\phi, \lambda_1)$	R	$ \mathcal{H}(H) = \mathcal{H}_{13}(\alpha\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22})$
solution 1	$(-3.1415927, 0.0)$	1.21853096	0.017074170 (> 0)
solution 2	$(-2.5085617, -0.23864305)$	1.209864499	$-0.080256775 (< 0)$
solution 3	$(-1.948572843, -0.23837645)$	1.2161630	0.18880119 (> 0)

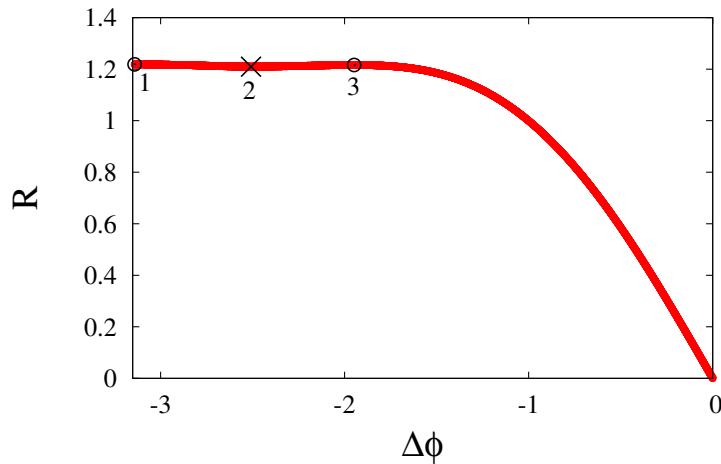


図 2.8: T_p における $R(\Delta\phi)$ を求めた結果 ($p = 5$)

$p = 5$ の場合, solution1,3 の $(\Delta\phi, \lambda_1) \sim (-3.1415927, 0.0), (-1.948572843, -0.23837645)$ は $|\mathcal{H}(H)| > 0$ であるため, $F(\Delta\phi, \lambda_1)$ の極大値, つまり $R[f]$ の極大値となる. solution1,3 の R の値を比較すると solution1 のほうが大きいため, 最適解は solution1 となり, solution3 は 2 番目の最適解となる. また, solution2 の $(\Delta\phi, \lambda_1) \sim (-2.5085617, -0.23864305)$ は $|\mathcal{H}(H)| < 0$ であるため, $F(\Delta\phi, \lambda_1)$ の極小値, つまり $R[f]$ の極小値となり, 今得たい最適解ではない. これらの関係は図 2.8 を見ても明らかである.

以上の結果より得られた最適解 $(\Delta\phi, \lambda_1)$ を (2.21) 式に代入し, 理論的に導出した

Hodgkin-Huxley 振動子の最適周期信号 $f_{\text{opt},p}$ を示す. $p = 1.01$ の場合における最適周期信号は図 2.12, $p = 2$ の場合における最適周期信号は図 2.13, $p = 5$ の場合における最適周期信号

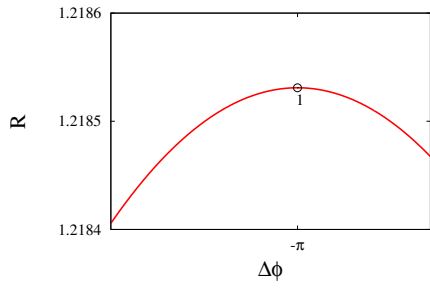


図 2.9: solution1 の拡大図

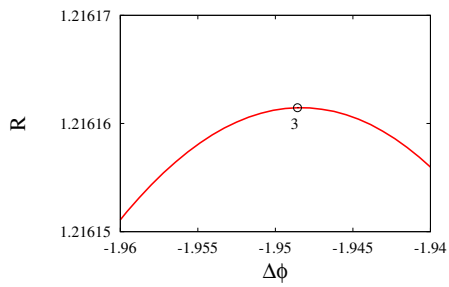


図 2.11: solution3 の拡大図

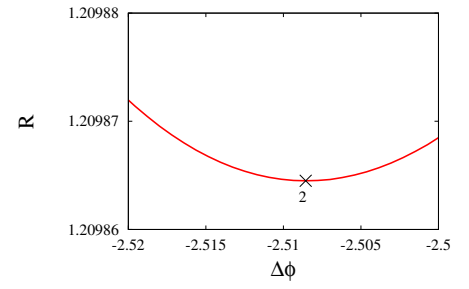


図 2.10: solution2 の拡大図

は図 2.14, 図 2.15 である.

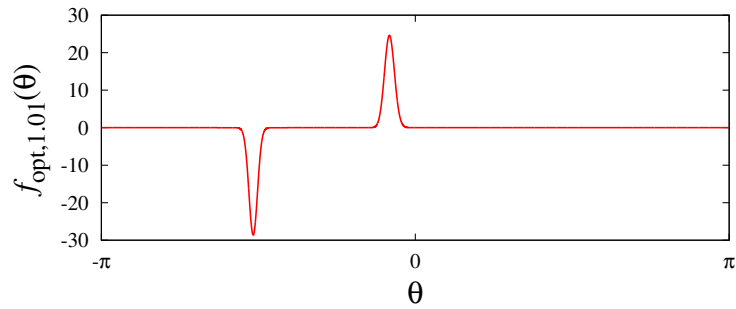


図 2.12: $p = 1.01$ の場合の最適周期信号 $f_{\text{opt},1.01}$ ($(\Delta\phi, \lambda_1) = (-1.364936336, 0.5704083690)$)

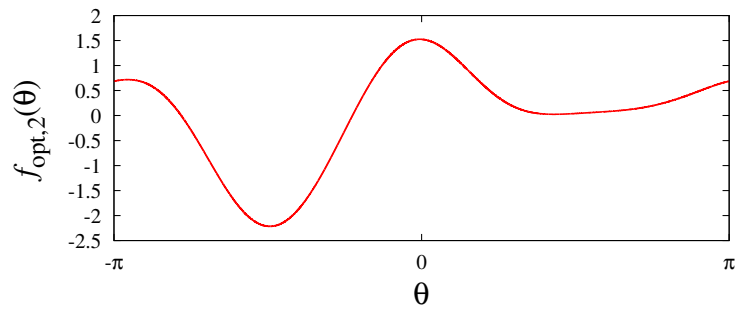


図 2.13: $p = 2$ の場合の最適周期信号 $f_{\text{opt},2}$ ($(\Delta\phi, \lambda_1) = (-1615065266, 0.0)$)

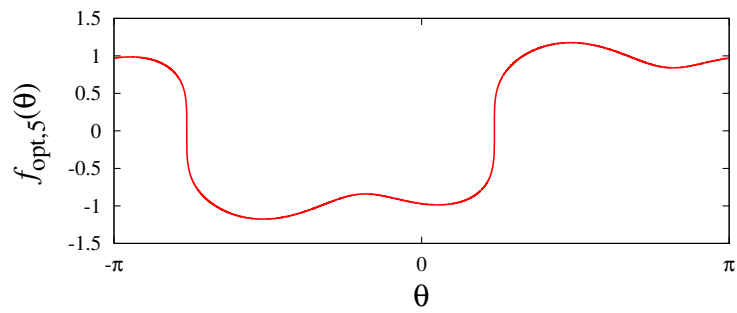


図 2.14: $p = 5$ の場合の最適周期信号 $f_{\text{opt},5}$ ($(\Delta\phi, \lambda_1) = (-3.1415927, 0.0)$)

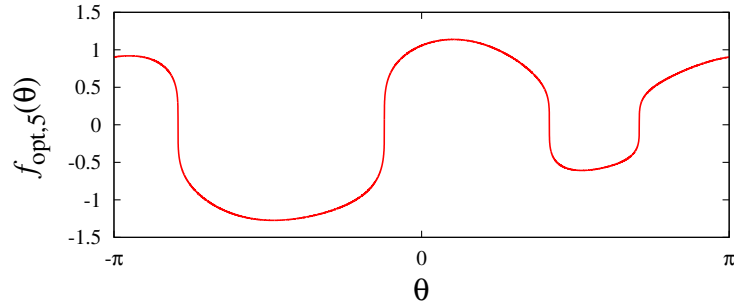


図 2.15: $p = 5$ の場合の 2 番目の最適周期信号 $f_{\text{subopt},5} ((\Delta\phi, \lambda_1) = (-1.948572843, -0.23837645))$

$p = \infty$ の場合

$1 < p < \infty$ の場合と同様に, $p = \infty$ の場合の最適周期信号を導出する. まず, (2.48) 式 of 最適波形 $f_{\text{opt},\infty}$ 内の 2 つのパラメータ $(\Delta\phi, \lambda_1)$ を決定するために, (2.57a) 式と (2.57b) 式の連立方程式を解き, 2 つのパラメータ $(\Delta\phi, \lambda_1)$ の解候補 $(\Delta\phi_*, \lambda_{1,*})$ を求める. (2.57a),(2.57b) 式の解の全体像を知るために, $(\Delta\phi, \lambda_1)$ をパラメータをある範囲で変動させ, (2.57a) 式と (2.57b) 式のそれぞれについて解曲線を描く. (2.57a) 式と (2.57b) 式より得られた 2 つの解曲線の交点を全て記録する. この記録した交点が (2.57a) 式と (2.57b) 式の連立方程式の解であり, 今得たい解候補 $(\Delta\phi_*, \lambda_{1,*})$ となる. 求めた (2.57a) 式と (2.57b) 式の解曲線を図 2.16 に示す. ここで, (2.57a) 式と (2.57b) 式において, $\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] Z'(\theta + \Delta\phi) \rangle \equiv S_\infty(\Delta\phi, \lambda_1)$, $\langle \text{sig}[\bar{Z}(\theta) + \lambda_1] \rangle \equiv T_\infty(\Delta\phi, \lambda_1)$ と置くことにする. $1 < p < \infty$ の場合と同様に, 図 2.16 より $T_\infty(\Delta\phi, \lambda_1)$ は $\Delta\phi$ の連続関数であり, $S_\infty(\Delta\phi, \lambda_1)$ は λ_1 の連続関数である. また本研究では, $f_{\text{opt},\infty}$ は周期 2π の周期関数であると仮定しているので, $\Delta\phi \in [-\pi, \pi]$ として, 解曲線を描いた.

(2.57b) 式の T_∞ と (2.57a) 式の S_∞ の解曲線の交点より得られる解候補 $(\Delta\phi_*, \lambda_{1,*})$ をより正確なものとするために, ニュートン法を用いた. ニュートン法で使用する (2.57a) 式と (2.57b) 式に対するヤコビ行列はそれぞれ (2.56) 式である. (2.56) 式を用いてニュートン法を解く際に, $(\Delta\phi, \lambda_1)$ の初期値が必要となるが, この初期値は (2.57b) 式の T_p と (2.57a) 式の S_p の解曲線の交点である $(\Delta\phi, \lambda_1)$ とした. ニュートン法より得られた解候補 $(\Delta\phi_*, \lambda_{1,*})$ のリストを表 2.4 に示す. ここで, 2.2.1 節で述べたように $(\Delta\phi, \lambda_1) \leftrightarrow (-\Delta\phi, -\lambda_1)$ と, 対称性を持つ.

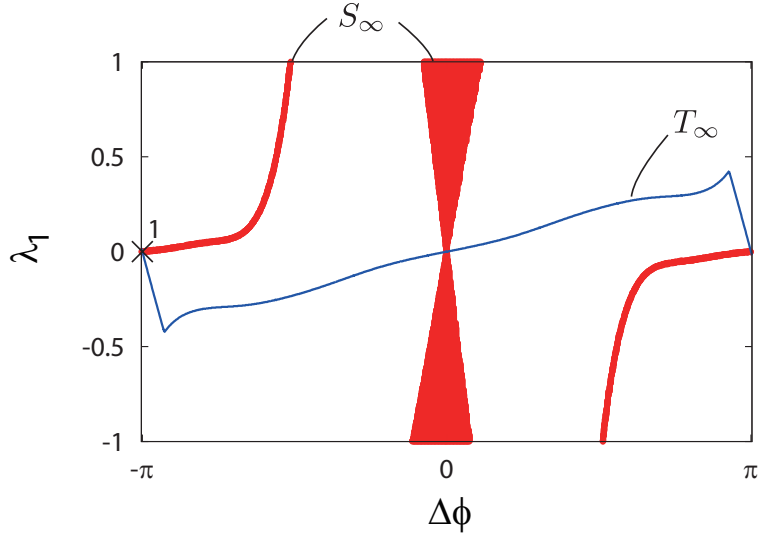


図 2.16: (2.57b) 式の T_∞ と (2.57a) 式の S_∞ の解曲線 ($p = \infty$)

次にニュートン法より得られた全ての解候補 $(\Delta\phi_*, \lambda_{1,*})$ について, $M = 1$ とし (2.49) 式よりロッキングレンジ R と (2.57c) 式より $|\mathcal{H}(H)|$ を求める. ロッキングレンジ R と $|\mathcal{H}(H)|$ を求めた結果を解候補 $(\Delta\phi_*, \lambda_{1,*})$ と並べて表 2.4 に示す. また, $\Delta\phi \in [-\pi, \pi]$ に対する T_∞ の全ての解について, $M = 1$ とし (2.49) 式よりロッキングレンジ $R(\Delta\phi)$ を求めた結果を図 2.17 に示す.

表 2.4: (2.57a) 式と (2.57b) 式を満たす解 $(\Delta\phi, \lambda_1)$ とロッキングレンジ $R(p = \infty)$

	$(\Delta\phi, \lambda_1)$	R	$ \mathcal{H}(H) = \mathcal{H}_{13}(\mathcal{H}_{12}^2 - \mathcal{H}_{13}\mathcal{H}_{22})$
solution 1	$(-3.1415927, 0.0)$	1.1784578	0.012231488 (> 0)

$p = \infty$ の場合, solution1 の $(\Delta\phi, \lambda_1) \sim (-3.1415927, 0.0)$ は $|\mathcal{H}(H)| > 0$ であるため, $F(\Delta\phi, \lambda_1)$ の極大値, つまり $R[f]$ の極大値となる. 他に $R[f]$ の極大値となる解候補はないため, 最適解は solution1 となる. この関係は図 2.17 を見ても明らかである.

以上の結果より得られた最適解 $(\Delta\phi, \lambda_1)$ を (2.48) 式に代入し, 理論的に導出した Hodgkin-Huxley 振動子の最適周期信号 $f_{\text{opt},\infty}$ を図 2.18 に示す.

$p = 1$ の場合

$p = 1$ の場合では, 2.2.1 節で述べたように 2 つのパラメータ $(\Delta\phi, \lambda_1)$ は既にわかっている, $\Delta\phi = \Delta\theta$ と $\lambda_1 = -\frac{1}{2}[\bar{Z}(\theta_{\max}) - \bar{Z}(\theta_{\min})]$ である. この $\Delta\theta$ は次のように計算的に求めること

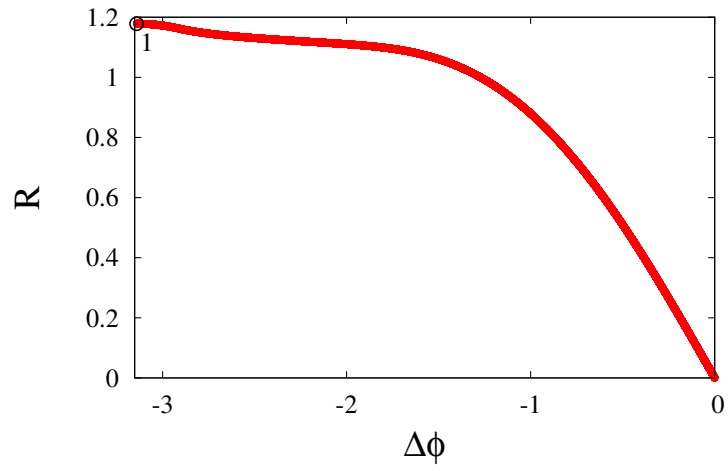


図 2.17: T_p における $R(\Delta\phi)$ を求めた結果 ($p = \infty$)

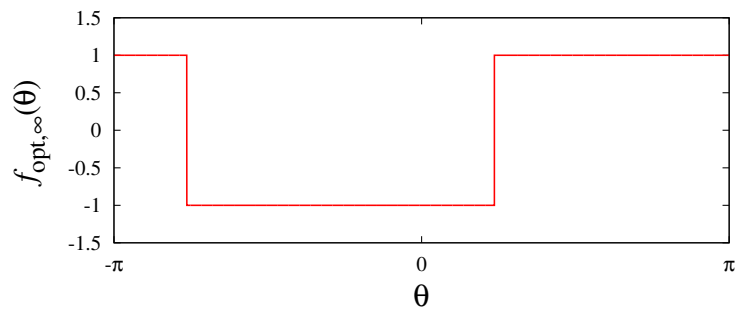


図 2.18: $p = \infty$ の場合の最適周期信号 $f_{\text{opt},\infty}$ ($(\Delta\phi, \lambda_1) = (-\pi, 0.0)$)

ができる。まず、 $\Delta\theta \in [-\pi, 0]$ として (2.69) 式の $\Gamma_0(\bar{\phi})$ を求める。

$$\Gamma_0(\bar{\phi}) = M[Z\bar{\phi} + \Delta\theta] - Z(\bar{\phi}) \quad (2.69)$$

ここで、ロッキングレンジ R は (2.70) 式より求めることができる。

$$R(\Delta\theta) = (\text{the maximum of } \Gamma_0) - (\text{the minimum of } \Gamma_0) \quad (2.70)$$

この (2.70) 式を $\Delta\theta \in [-\pi, 0]$ をパラメータとして求めた結果を図 2.19 に示す。

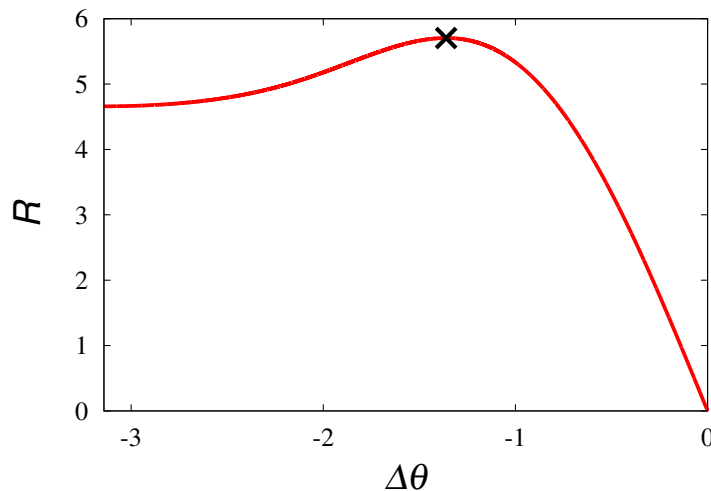


図 2.19: 理論的に求めた $\Delta\theta \in [-\pi, 0]$ に対する R

図 2.19 より、 $\Delta\theta$ の最適解は $\Delta\theta \sim -1.36094$ なる。また、図 2.19 は既に求めた $p = 1.01$ における図 2.6 の $R - \Delta\phi$ のグラフとほぼ一致していることが分かる。

次に、この理論的に求めた $\Delta\theta \in [-\pi, 0]$ に対する R を検証するために、以下の 2 つの行う。

1. (2.65) 式の $\Gamma_0(\phi)$ を用いて、位相方程式 (2.2) から直接ロッキングレンジ R を求める。
2. (2.66) 式の弱いインパルス状のパルス波形を用いて、Winfree モデル式 (2.1) からロッキングレンジ R を求める。このとき、パルス幅は 0.07 とし、パルスの高さは 1.14 とする。

1., 2. の方法で R を求めた結果を図 2.20 に示す。図 2.20 中の黒実線は (2.70) 式より求めた結果、+ 点は 1. の方法で R を求めた結果、 \times 点は 2. の方法で R を求めた結果である。

図 2.20 より、どの方法で求めた R の結果も全て精度よく一致していることが分かる。したがって、 $p = 1$ の場合の最適周期信号はパルス間の位相差 $\Delta\theta$ を $\Delta\theta = -1.36094$ とした正負一対のパルスとなる。

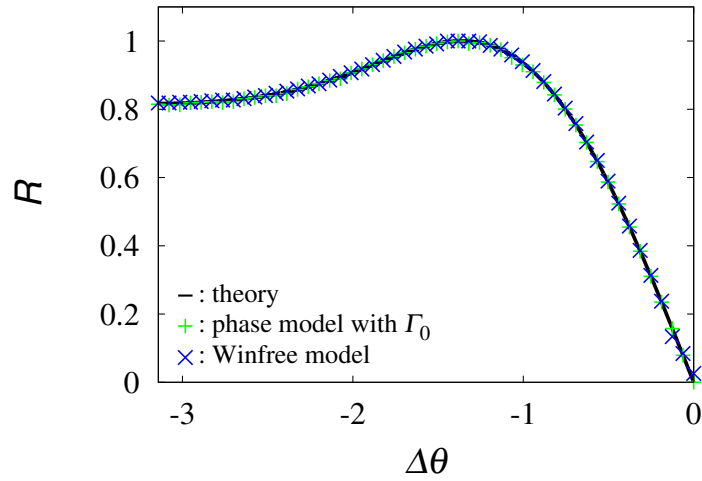


図 2.20: 数値計算によって求めた $R - \Delta\theta$ の結果

2.3 Genetic Algorithm を用いた最適周期信号の探索

2.3.1 Genetic Algorithm の設計

遺伝的アルゴリズム (Genetic Algorithm, 以下 GA) は汎用性の高さから、最適化問題を解くアプローチとして広く知られている [?]. この GA では解の候補を遺伝子で表現した「個体」を複数個用意し、各個体に付けられる評価値の高い個体を優先的に選択して、遺伝、増殖、突然変異といった操作を繰り返しながら最適な解を探索する。つまり、評価値の高い個体ほど次の世代にその遺伝子が引き継がれやすく、生き残っていくのである。この操作は、自然界の生物の進化過程をヒントとしたものである。

GA による最適周期信号の探索手順を図 2.21 に示す。

本研究で使用する GA において探索する解は、ロッキングレンジを最大化する最適周期信号である。従って、この最適周期信号を探索パラメータとして設定する。設定する際、探索する周期信号 $f_{GA}(\theta)$ を $\theta \in [-\pi, \pi]$ の範囲で N 点のベクトルを等間隔に配置し、ベクトル点によって $f_{GA}(\theta)$ を表現する。この N 個のベクトル点全てが探索パラメータとなる。

このとき、各パラメータは遺伝子情報として 20 ビットの 2 進数で与えられる。したがって、各個体が持つ遺伝子情報は $20 \times N$ ビット数となる。

ここで、2.2 節と同様に周期信号 $f_{GA}(\theta)$ には $\|f_{GA}(\theta)\|_p = M$ と $\langle f_{GA}(\theta) \rangle = 0$ という制約を課す。そのため、各探索パラメータはこの制約を満たすように決定されるようにした。

GA において、一番重要な書く個体における評価値の設定についてだが、本研究では、ロッ

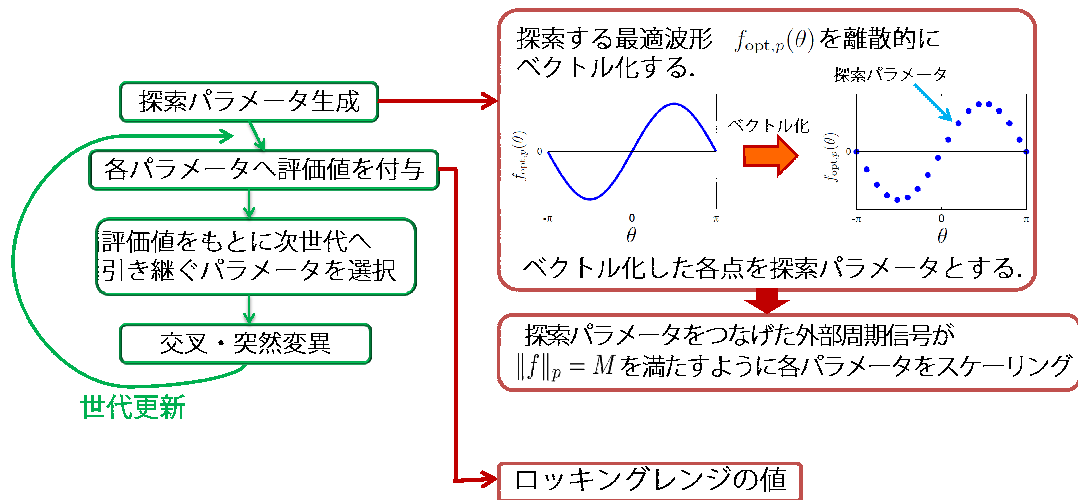


図 2.21: 本研究での GA による解探索の手順

キングレンジを最大化する最適周期信号を探索するため、評価値としてロッキングレンジの値を採用する。今回、周期信号のパワーはそれほど大きく設定しないため、ロッキングレンジは式 (2.1) の相互作用関数より導出することとする。

以上のように設計した GA よりロッキングレンジを最大化する最適周期信号を探索していく。しかし、次節を見れば明らかだが、GA は局所的な解に収束するため、初期探索パラメータによって異なる解が出てきてしまうことがある。そのため、出てきた複数の解に対してロッキングレンジを求め、大域的な最適解を見極める必要がある。

GA による探索解と理論解の結果が一致すれば、理論解の妥当性を示すことになる。ただ、GA では最適解を出すために時間がかかるため、GA はあくまで理論解の妥当性を示すための検証ツールの1つである。

2.3.2 理論解と Genetic Algorithm による探索解の比較

Hodgkin-Huxley 振動子を対象とし、2.2.2 節で理論的に導出した最適周期信号（以下、理論解）と GA より求めた探索解を比較する。GA による最適解の探索において、 p ノルムの制約条件を $M = 1$ 、探索世代数を 10 万世代、個体数 1000 個体とした。

まず、 $p = 1.01$ の場合の比較結果を図 2.22 に示す。

図 2.22 より、 $p = 1.01$ の場合の理論解と探索解は高い精度で一致している。また理論解のロッキングレンジは 2.7613265、探索解のロッキングレンジは 2.7609771 であり、誤差率は

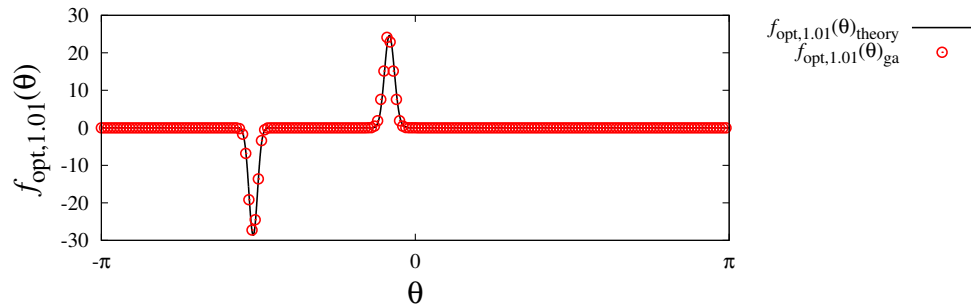


図 2.22: 理論解と探索解の比較結果 ($p = 1.01$)

0.01265 となった。この結果、全く異なるアプローチにより得た 2 つの解が一致することを確認できた。

次に、 $p = 2$ の場合の比較結果を図 2.23 に示す。

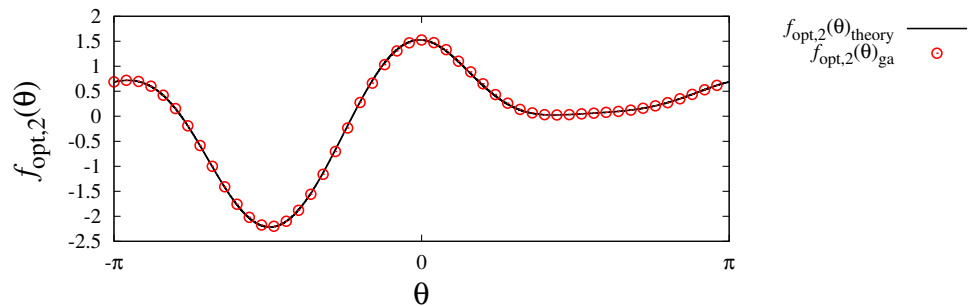


図 2.23: 理論解と探索解の比較結果 ($p = 2$)

図 2.23 より、 $p = 2$ の場合の理論解と探索解は高い精度で一致している。また理論解のロッキングレンジは 1.4926698、探索解のロッキングレンジは 1.4925093 であり、誤差率は 0.01075 となった。

$p = 5$ の場合の比較結果を図 2.24, 2.25 に示す。

図 2.24, 2.25 より、 $p = 5$ の場合の理論解と探索解は高い精度で一致している。また図 2.24 の solution1 に対する理論解のロッキングレンジは 1.21853096、探索解のロッキングレンジは 1.2177683 であり、誤差率は 0.06263 となった。図 2.25 の solution3 に対する理論解のロッキングレンジは 1.2161630、探索解のロッキングレンジは 1.2162259 であり、誤差率は 0.005171737 となった。

最後に $p = \infty$ の場合の比較結果を図 2.26 に示す。

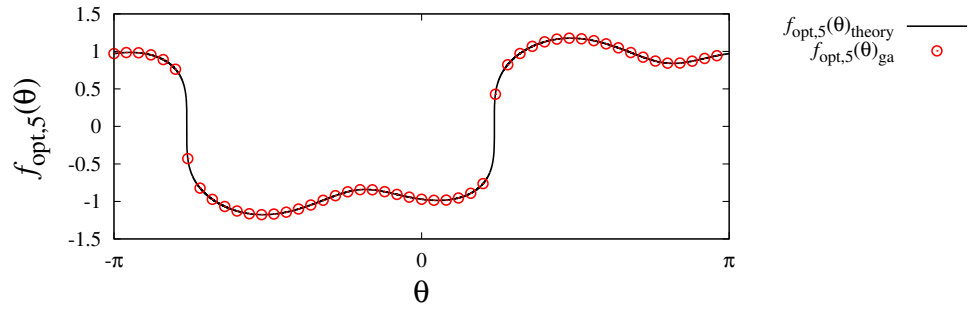


図 2.24: 理論解と探索解の比較結果 ($p = 5$, solution1)

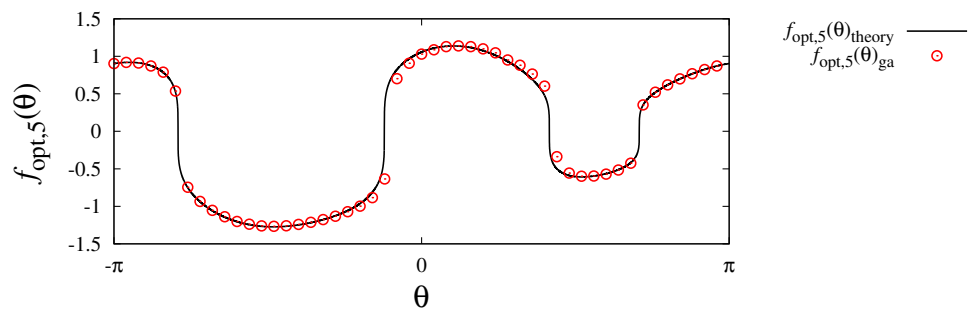


図 2.25: 理論解と探索解の比較結果 ($p = 5$, solution3)

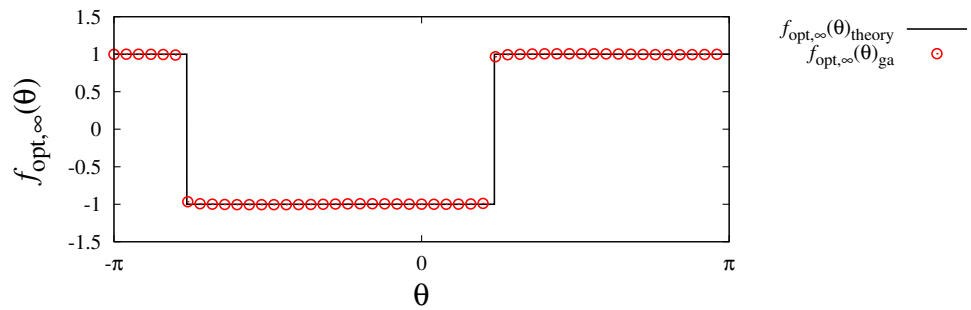


図 2.26: 理論解と探索解の比較結果 ($p = \infty$)

図 2.26 より, $p = \infty$ の場合の理論解と探索解は高い精度で一致している. また理論解のロッキングレンジは 1.1784578, 探索解のロッキングレンジは 1.1783933 であり, 誤差率は 0.005473555 となった.

2.4 設計した最適外部周期信号の引き込み能力のシミュレーション検証

本節では, 前節で設計した外部周期信号の引き込み能力をシミュレーション検証した. シミュレーション検証方法としては位相応答方程式 (1.25) を用いる. 式 (1.25) において, 位相差が一定となる時, 振動子は外部信号に引き込まれたといえる. 逆に, 位相差が一定とならなかった場合, 振動子は外部信号に引き込まれなかったといえる. よって, 振動子の位相応答関数 $Z(\psi)$ に信号を注入し, 十分時間がたった後の位相差の時間変化をみることで, 注入した信号の引き込み能力を検証できる.

比較するために用いた信号は以下の 7 つである.

- (a) 正弦波 (図 2.27)
- (b) 位相差を π としたときの正負一對のパルス (パルス幅 0.2π) (図 2.28)
- (c) 位相差を $\frac{\pi}{2}$ としたときの正負一對のパルス (パルス幅 0.2π) (図 2.29)
- (d) $\frac{\pi}{4}$ の区間正となり, 残りの $\frac{3\pi}{4}$ の区間負となる矩形波 (図 2.30)
- (e) $p = 1$ の時の最適波形 (パルス幅 0.07) (図 2.31)
- (f) $p = 2$ の時の最適波形 (図 2.32)
- (g) $p = \infty$ の時の最適波形 (図 2.33)

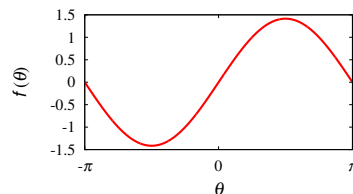


図 2.27: a

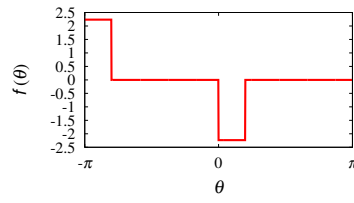


图 2.28: a

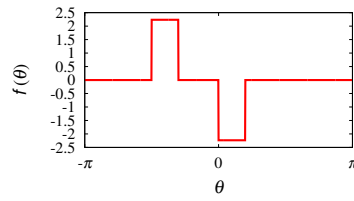


图 2.29: b

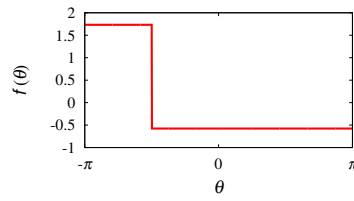


图 2.30: c

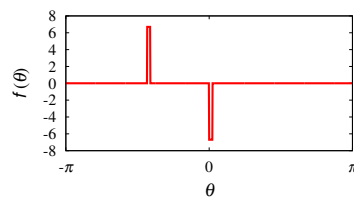


图 2.31: d

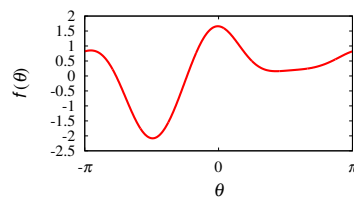


图 2.32: e

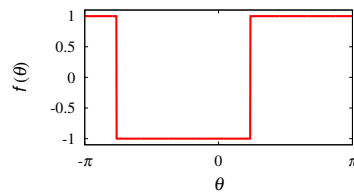


図 2.33: f

上記の注入信号を用いてロッキングレンジを比較した．その際，横軸を注入信号の周波数，縦軸を注入信号の大きさを表す指標とすることで，引き込み周波数帯を求める事ができ，これをアーノルドタングと呼ぶ．

今回，注入信号の大きさを表す指標として，パワー，振幅，面積を用いた．本研究で設計した外部信号が最適であれば，パワーを指標とした時は $p = 2$ の時の最適外部信号，振幅を指標とした時は $p = \infty$ の時の最適外部信号，面積を指標とした時は $p = 1$ の時の最適外部信号が最もロッキングレンジが大きくなる，つまり引き込み能力が最適な外部信号といえる．以上のシミュレーション設定により得たアーノルドタングを以下に示す．

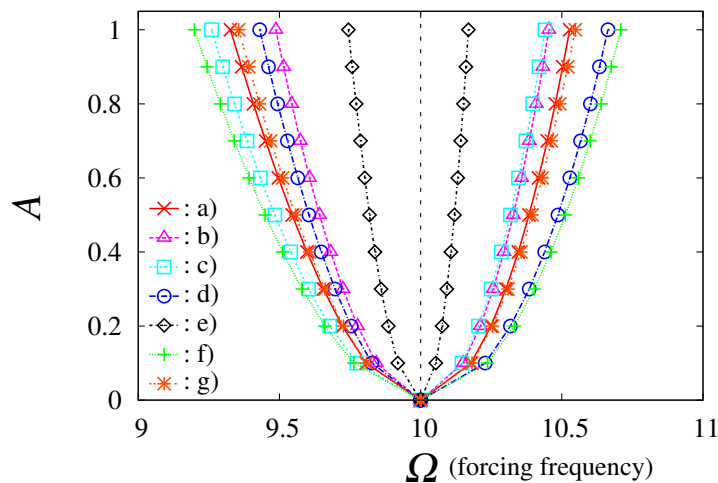


図 2.34: アーノルドタング (パワー一定)

図 2.34, 2.35, 2.36 より，本研究で得られた最適外部信号が引き込み能力を最適とすることを示すことができた．

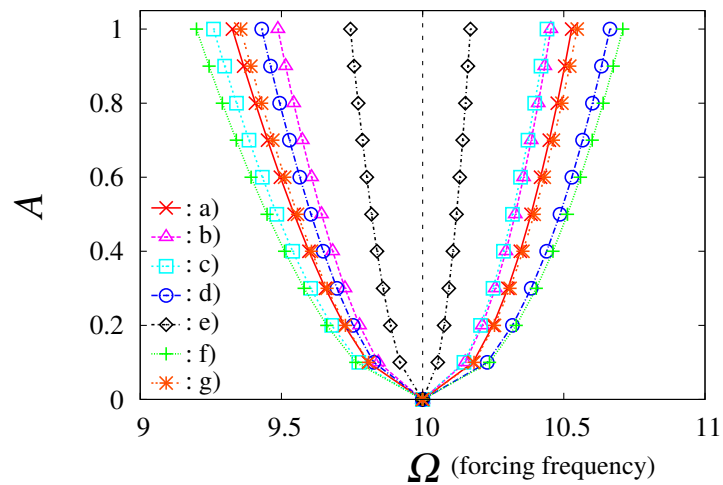


図 2.35: アーノルドタンク (振幅一定)

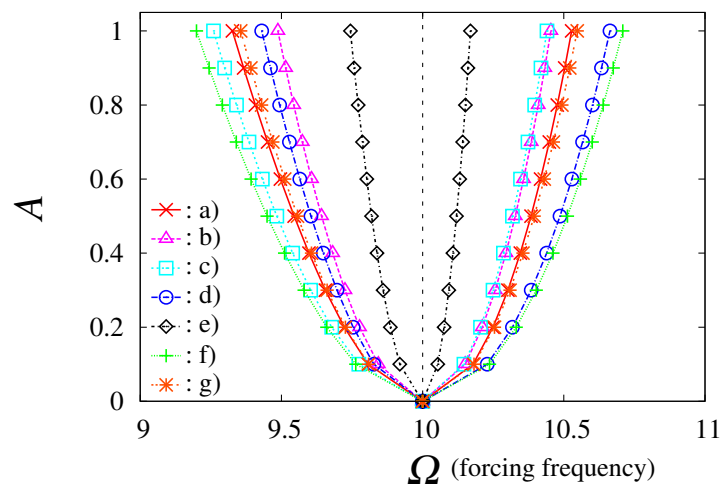


図 2.36: アーノルドタンク (面積一定)

2.5 通倍動作時の最適外部周期信号の引き込み能力のシミュレーション 検証

前説までは、振動子の周波数:外部信号の周波数が 1:1 同期となるときについて考えてきた。本節では、 $m:1$ 同期 (m は整数) について議論する。 $m:1$ 同期を達成できれば、小さい周波数の外部信号を用いて、大きい周波数の発振器を制御できれば、低コストが期待できる。簡単な $m:1$ 同期である 2:1 同期のロッキングレンジのシミュレーション検証の結果を示す。

パワー一定の場合

パワー一定の場合、つまり $p = 2$ の時の最適周期信号 $f_{\text{opt},2}(\theta)$ を用いる。また、比較する注入信号として以下の 4 つを考える。

(P0) 1:1 同期を想定した時の信号 (図 2.37)

(P1) 外部信号の周期を振動子の周期の半分にし、残りの半分为 0 とした信号 (図 2.38)

(P2) 外部信号の周波数を 2 分の 1 とした信号 (図 2.39)

(P3) P0 に微小摂動を加えた時の信号 (図 2.40)

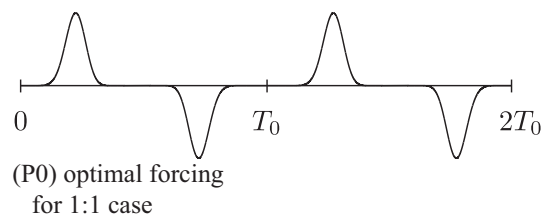


図 2.37: P0

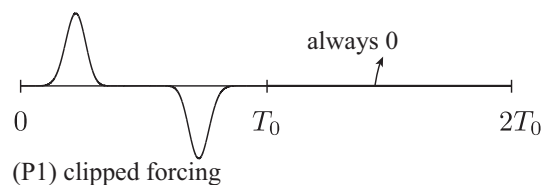


図 2.38: P1

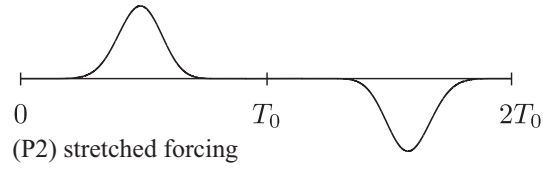


図 2.39: P2

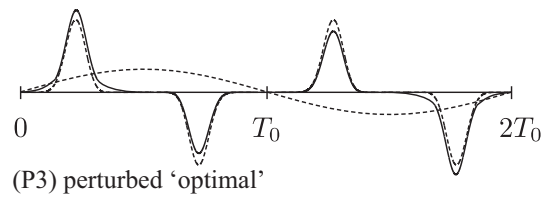


図 2.40: P3

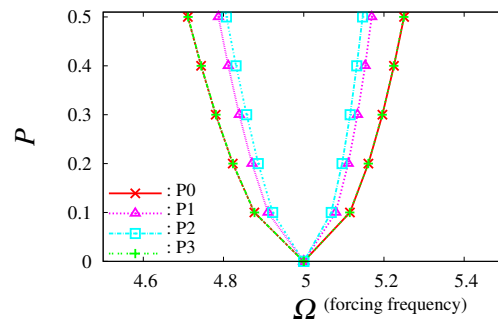


図 2.41: 2:1 同期の時のアーノルドタンク (パワー一定)

以上の注入信号を用いて求めたアーノルドタングを図 2.41 に示す.

図 2.41 において, パワー $P = 0.1, 0.5$ とした時のロックレンジを棒グラフにして比較したものを図 2.42 に示す

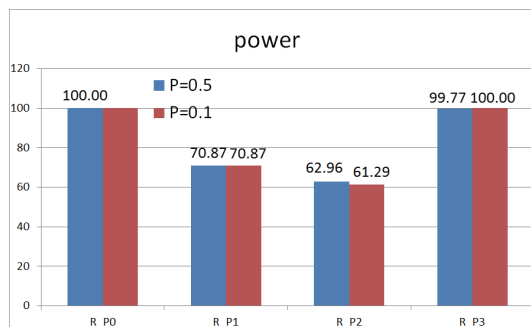


図 2.42: ロックレンジの比較 (パワー一定)

図 2.41 より, 各信号のロックレンジを $R_{p0}, R_{p1}, R_{p2}, R_{p3}$ とすると,
 $R_{p0} = \sqrt{2}R_{p1} > R_{p3} \gg R_{p2}$ となっていることがわかる.

振幅一定の場合

振幅一定の場合, つまり $p = \infty$ としたときの注入信号を用いる. 比較する信号として, パワー一定の場合と同様に図 2.43 のように設定した.

以上の注入信号を用いて求めたアーノルドタングを図 2.44 に示す.

図 2.44 において, $M = 0.5, 1$ とした時のロックレンジを棒グラフにして比較した結果を図 2.45 に示す.

図 2.45 より, 各信号のロックレンジを $R_{M0}, R_{M1}, R_{M2}, R_{M3}$ とすると,
 $R_{M0} = 2R_{M1} > R_{M3} \gg R_{M2}$ となっていることがわかる.

面積一定の場合

面積一定の場合, つまり $p = 1$ としたときの注入信号を用いる. 比較する信号として, パワー一定, 振幅一定の場合と同様に図 2.46 のように設定した. 以上の注入信号を用いて求めたアーノルドタングを図 2.47 に示す.

図 2.47 において, $A = 0.5, 1$ とした時のロックレンジを棒グラフにして比較した結果を図 2.48 に示す.

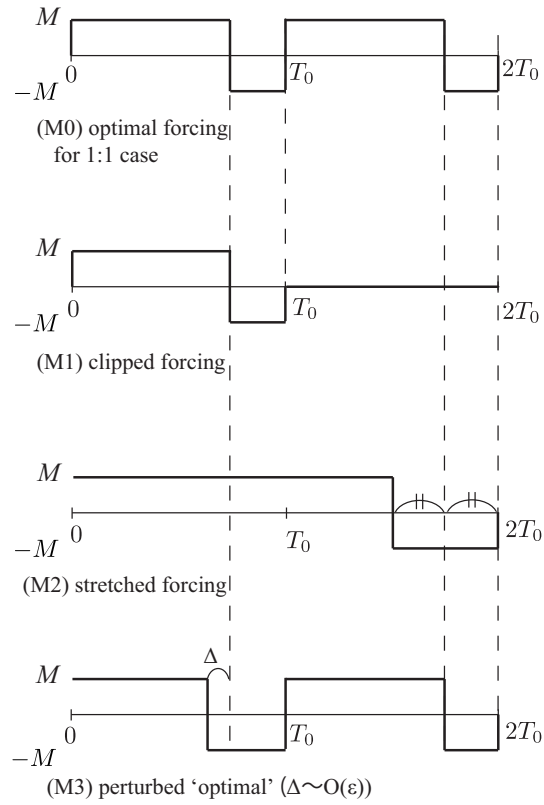


図 2.43: 注入信号 M0~M3 (振幅一定)

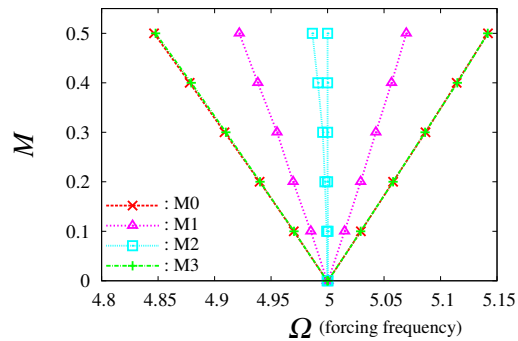


図 2.44: アーノルドタンク (振幅一定)

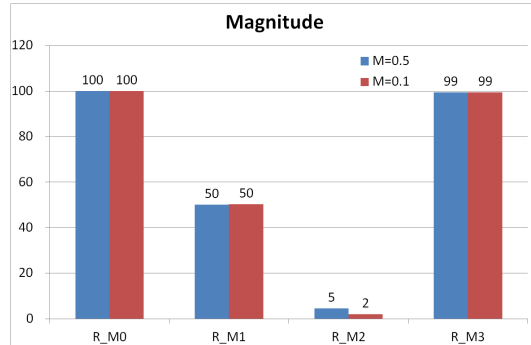


図 2.45: ロッキングレンジの比較 (振幅一定)

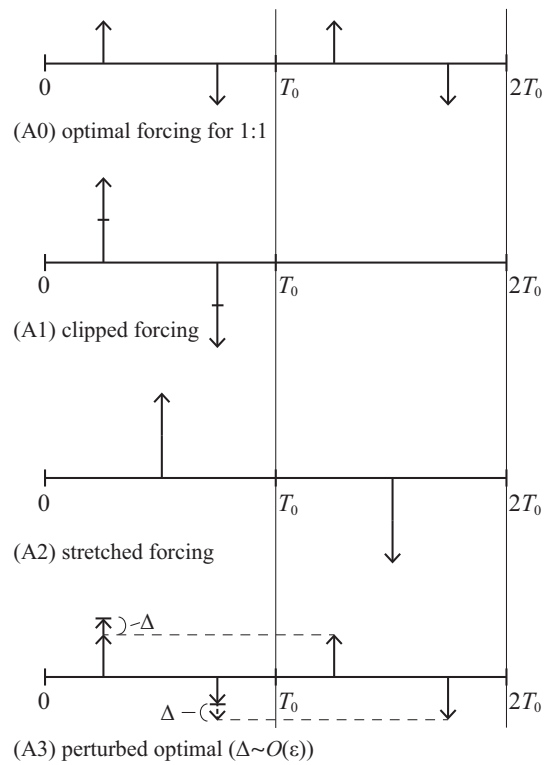


図 2.46: 注入信号 A0~A3 (面積一定)

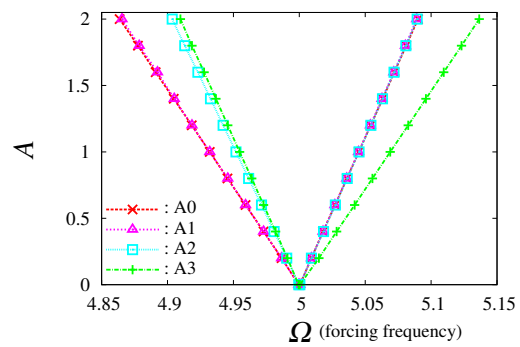


図 2.47: アーノルドタンク (面積一定)

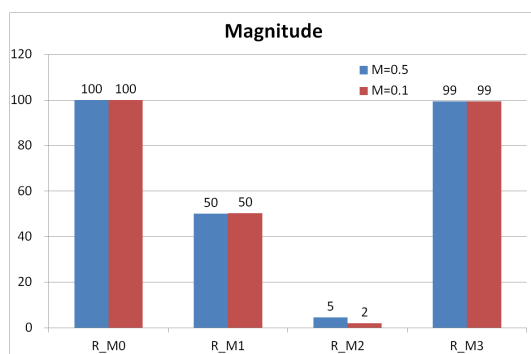


図 2.48: ロッキングレンジの比較 (面積一定)

図 2.48 より，各信号のロックレンジを R_{A0} , R_{A1} , R_{A2} , R_{A3} とすると，
 $R_{A0} = R_{A1} > R_{A3} \gg R_{A2}$ となっていることがわかる．

第3章 ノイズ環境下での位相応答曲線の推定及び 最適な外部周期信号の設計

より現実的な環境を想定する場合，シミュレーションによるノイズの与え方を適切に設定しなくてはならない．しかし，菊地らはシミュレーションにおいて，間違ったノイズの与え方を行っていた．そのため，ノイズの与え方を正しくしてシミュレーションをやり直し，菊地らの手法のノイズ耐性を再検証する事には手法の有用性を示すため，また，理論へ拡張した時の裏付けのために価値があると考えられる．

3.1 数値シミュレーションにおけるノイズの与え方

Hodgkin-Huxley 振動子を例に，確率微分方程式を解く方法を以下に記載する [7, 8]．ノイズを含む振動子の時間発展を表す式として，1.4.3 節で述べたランジュバン方程式を用いる．

$$\frac{dV}{dt} = f(V, m, h, n) + \sigma \cdot \xi(t) \quad (3.1)$$

ただし， V ， m ， h ， n は HH 振動子のパラメータ， ξ はホワイトノイズ， σ はノイズの大きさである標準偏差を表している．白色ノイズ $\xi(t)$ は，次式のように Wiener 過程 $W(t)$ の導関数で表される．

$$\xi(t) = \frac{dW(t)}{dt} \quad (3.2)$$

以上より，式 (3.1) の Langevin 方程式は以下の確率微分方程式で表される．

$$dV = f(V, m, h, n)dt + \sigma \cdot dW(t) \quad (3.3)$$

ここで，Wiener 過程 $W(t)$ とはガウス過程（正規過程）の事であり，以下の性質を持つ．

$$P(W(0) = 0) = 1 \quad (3.4)$$

$$\mathbf{E}(W(t)) = 0 \quad \forall t \in [0, \infty) \quad (3.5)$$

$$\mathbf{E}(W(t)W(s)) = \min(t, s) \quad (3.6)$$

ここで、 \mathbf{P} は確率測度、 \mathbf{E} は平均である。また、Wiener 過程は次の性質を持つことも知られている。

$$\mathbf{E}(W(t) - W(s)) = 0, \quad \mathbf{E}((W(t) - W(s))^2) = t - s, \quad 0 \leq s \leq t \quad (3.7)$$

$$\mathbf{E}(\{W(t_2) - W(t_1)\}\{W(t_4) - W(t_3)\}) = 0, \quad t_1 \leq t_2 \leq t_3 \leq t_4 \quad (3.8)$$

以上の Wiener 過程の性質 (3.7) より、 $W(t) - W(s) (0 \leq s \leq t)$ は平均 0、分散 $t - s$ の正規分布 $N(0; t - s)$ に従う。すなわち、独立変数 t によるステップ点を $t_k (k = 0, 1, 2, \dots, N)$ 、ステップ幅を $h = t_{k+1} - t_k$ 、ウィナー過程の増分を $dW_k = W(t_{k+1}) - W(t_k)$ とすると、ウィナー過程 $W(t_n)$ は

$$W(t_n) = \sum_{k=0}^{n-1} \Delta W_k, \quad n = 1, 2, \dots, N \quad (3.9)$$

とウィナー過程の増分 ΔW の単純な和として表される。 ΔW_k は平均 0、分散 h の正規分布 $N(0, h)$ に従う確率変数であるから、この確率変数 $W(t)$ は

$$\Delta W_k = \xi_k \sqrt{h} \quad (3.10)$$

となる。 ξ_k は平均 0、分散 1 の標準正規分布 $N(0, 1)$ に従う標準正規乱数である。したがって、確率微分方程式 (3.3) は

$$dV(t) = f(V, m, h, n)dt + \xi_k \sqrt{h} \quad (3.11)$$

となり、 $h = t_{k+1} - t_k = dt$ より

$$dV(t) = f(x)dt + \xi_k \sqrt{dt} \quad (3.12)$$

となる。したがって、 $V(t)$ の差分方程式は以下のようなになる。

$$V(t_{k+1}) = V(t_k) + f(V, m, h, n)dt + \xi_k \sqrt{dt} \quad (3.13)$$

3.2 ノイズ環境下での位相応答曲線の推定

2.1 節で行った、インパルス応答法は、高周波数の発振器を対象としているため、ノイズのない理想的な環境を想定していた。この手法をノイズのある環境下を想定した場合、微小インパルスがノイズに埋もれてしまい、位相応答曲線の精度が極端に悪くなる。そこで本節では、ノイズ耐性のある菊地らによる手法で位相応答曲線を推定した [9]。

3.2.1 菊地らによる位相応答曲線の推定手法の原理

菊地らは位相応答曲線を推定するために、周波数引き込みの現象を利用している。この周波数引き込みにより得られる振動子の特性より、位相応答曲線を逆算する事ができる。以下では、そのメカニズムを説明する。

位相応答曲線 $Z(\phi)$ は周期関数のため以下のようにフーリエ級数展開できる。

$$Z(\phi) = \sum_n Z_n(\phi) \quad (3.14)$$

また、 Z_n は次のように未知の振幅 a_n と b_n をもつ2つの正弦波の和として表せる。

以下ではまず、簡単のために振動子が n 倍周波数正弦波 $p_n(\tau)$ に引き込まれた場合を考える。

$$Z_n(\phi) = Z\left(\frac{\omega'}{n}\tau + \psi\right) \quad (3.15)$$

$$Z_n(\phi) = a_n \cos(n \cdot \omega \cdot \tau + n \cdot \psi_n) + b_n \cdot \sin(n \cdot \omega \cdot \tau + n \cdot \psi_n) \quad (3.16)$$

$$Z_0(\phi) = a_0 \quad (3.17)$$

ただし、 ϕ_n は Z_n が持つ固有の位相で $[0, 2\pi]$ をとる。よって、

$$Z(\phi) = Z_0(\phi) + \sum_{n=1} [a_n \cdot \cos(n \cdot \omega \cdot \tau + n \cdot \psi_n) + b_n \cdot \sin(n \cdot \omega \cdot \tau + n \cdot \psi_n)]. \quad (3.18)$$

また、注入する正弦波 $p_n(\tau)$ は次のようにする。

$$p_n(\tau) = A \cdot \sin(n \cdot \omega \cdot \tau + n \cdot \psi') \quad (3.19)$$

$$p_0(\tau) = A \quad (3.20)$$

$n \geq 1$ の場合

$p_n(n \geq 1)$ を注入した時、離調 $\Delta\omega = \frac{\omega'}{n} - \omega$ と位相差 $\Delta\psi = \psi' - \psi$ の関係は以下のようなになる。ここで、 ω' は引き込まれた後の振動子の周波数、 ψ' は注入正弦波の遅い時間変数である。

$$\begin{aligned} \frac{n \cdot \omega}{n} - \omega &= A \cdot a_0 \cdot \sin(n \cdot \omega \cdot \tau + n \cdot \psi') \\ &+ A \cdot \frac{1}{2} \cdot \sum_{k=1} [a_k \cdot \sin((k+n) \cdot \omega \cdot \tau + k \cdot \psi_k + n \cdot \psi')] \\ &- A \cdot \frac{1}{2} \cdot \sum_{k=1} [a_k \cdot \sin((k-n) \cdot \omega \cdot \tau + k \cdot \psi_k - n \cdot \psi')] \\ &+ A \cdot \frac{1}{2} \cdot \sum_{k=1} [b_k \cdot \sin((k-n) \cdot \omega \cdot \tau + k \cdot \psi_k - n \cdot \psi')] \\ &- A \cdot \frac{1}{2} \cdot \sum_{k=1} [b_k \cdot \sin((k+n) \cdot \omega \cdot \tau + k \cdot \psi_k + n \cdot \psi')] \end{aligned} \quad (3.21)$$

ここで、 ψ の変化は ω と比べて十分遅い事に着目する。 ψ は注入信号の一周期 $0 \rightarrow \frac{2\pi}{n}$ の間にほとんど変化がないといえるので、周期 $\frac{2\pi}{n}$ で平均した値を用いても問題ない。そのため、(3.21) を以下のように $\frac{2\pi}{n}$ で平均化する事ができる。

$$\begin{aligned} \frac{n \cdot \omega}{n} - \omega = & \frac{A}{2} \cdot \frac{n}{2\pi} \sum_{k=1} \int_0^{\frac{2\pi}{n}} [a_n \cdot \sin((k+n) \cdot \omega \cdot \tau + k \cdot \psi_k + n \cdot \psi')] d(\omega\tau) \\ & - \frac{A}{2} \cdot \frac{n}{2\pi} \sum_{k=1} \int_0^{\frac{2\pi}{n}} [a_n \cdot \sin((k-n) \cdot \omega \cdot \tau + k \cdot \psi_k - n \cdot \psi')] d(\omega\tau) \\ & + \frac{A}{2} \cdot \frac{n}{2\pi} \sum_{k=1} \int_0^{\frac{2\pi}{n}} [b_n \cdot \cos((k-n) \cdot \omega \cdot \tau + k \cdot \psi_k - n \cdot \psi')] d(\omega\tau) \\ & - \frac{A}{2} \cdot \frac{n}{2\pi} \sum_{k=1} \int_0^{\frac{2\pi}{n}} [b_n \cdot \cos((k+n) \cdot \omega \cdot \tau + k \cdot \psi_k + n \cdot \psi')] d(\omega\tau) \quad (3.22) \end{aligned}$$

積分を実行すると、次のような簡単な形にすることができる。

$$\frac{n \cdot \omega}{n} - \omega = A \cdot \frac{1}{2} [a_n \cdot \sin(n \cdot (\psi' - \psi_n)) + b_n \cdot \cos(n \cdot (\psi' - \psi_n))] \quad (3.23)$$

$n = 0$ の場合

$n \geq 1$ の場合と同様に平均化を用いると、

$$\omega' - \omega = a_0 A \quad (3.24)$$

となる。ただし、 ω' は振動子が注入を受けて安定したときの角周波数である。

注入正弦波の角周波数が振動子の自然角周波数の整数倍でない場合

注入正弦波の角周波数が振動子の自然角周波数の整数倍から少しずれたときも、周波数引き込みは起こり、位相差と角周波数差の関係式に安定解が存在する。よって、注入正弦波の角周波数が振動子の自然角周波数の n 倍調波から少しずれた $\omega' (\simeq n\omega)$ の場合も、式(3.23) 同様以下の方程式が成り立つ。

$$\frac{\omega'}{n} - \omega = A \cdot \frac{1}{2} [a_n \cdot \sin(n \cdot (\psi' - \psi_n)) + b_n \cdot \cos(n \cdot (\psi' - \psi_n))] \quad (3.25)$$

なぜなら、振動子の自然角周波数の n 倍から少しずれた注入正弦波に引き込まれて、振動子は注入正弦波の $1/n$ 倍の角周波数で振動するため、位相応答曲線の角周波数も引き込まれた後のものになるためである。したがって、得られた引き込み特性（位相差-角周波数差）の

データ点を2つ用いると、式(3.25)に位相差と角周波数差の値を代入した方程式が2式できることになり、これらを連立させて位相応答曲線のフーリエ係数 a_n, b_n を推定する事が可能となる。このフーリエ係数を各調波でそれぞれ a_n, b_n を求める事で、位相応答曲線を推定する事が可能となる。

ここで、菊地らの手法は位相方程式を用いているため正弦波の大きさは可能な限り小さい方が望ましい。そこで、菊地らはある程度大きさの強い正弦波に対する引き込み範囲を複数算出し、これらの引き込み範囲から大きさの弱い正弦波に対する引き込み範囲を推定することによって、位相応答曲線を推定している。

3.2.2 位相応答曲線の推定手順

具体的な位相応答曲線の推定手順を以下に示す。位相応答曲線は以下の引き込み特性より位相応答曲線のフーリエ係数 a_n, b_n を各調波毎に求める事によって推定することができる。

$$\Delta\omega = \frac{A}{2}(a_n \sin(n\Delta\psi) - b_n \cos(n\Delta\psi)) \quad (3.26)$$

A は注入正弦波の振幅のため、定数である。したがって、振動子と注入正弦波の周波数離調 $\Delta\omega$ 、位相差 $n\Delta\psi$ を求める事によって、式(3.26)からフーリエ係数を求める事が可能となる。式(3.26)より変数は a_n, b_n の2つであるため $(\Delta\omega, \Delta\psi)$ のペアは少なくとも2つ以上必要である。

まず、 $\Delta\omega$ を求める方法を以下に記載する。振幅 A の正弦波を注入した際の引き込み可能範囲を求め、引き込み可能な注入正弦波の周波数と振動子の周波数差である $\Delta\omega$ の最大値、最小値を得る。ここで、位相方程式が成り立つためには注入正弦波の振幅が小さい事が望ましい。しかし、振幅が小さい場合、注入正弦波がノイズに埋もれてしまい、引き込み範囲を求める事が困難となってしまう。一般的に位相方程式が成り立つ場合、注入正弦波の振幅が高くなるほど引き込み範囲は広くなるため、ノイズの影響を受けにくいといえる。そこで、振幅が高い場合の引き込み範囲を測定し、振幅が小さい場合の引き込み範囲を推定する。まず、注入正弦波の振幅が高い場合の周波数特性を取得する。ここで、「高い」振幅の「高い」の定義は、ノイズなしで取得した引き込み可能周波数帯（アーノルドタング）の線形的になっている部分の振幅の最大値とする。例えば、図3.1はHodgkin-Huxley振動子を対象とした時の1調波のノイズがない環境下におけるアーノルドタングであるが、振幅 $A = 0$ のときの注入正弦波の周波数 Ω 、すなわち振動子の自然発振周波数を通るようにアーノルドタン

グの左端に直線を引くと、振幅約 0.7 まで線形的になっている事がわかる。よって、図 3.1 の場合の注入正弦波の「高い」振幅の最大値は 0.7 となる。

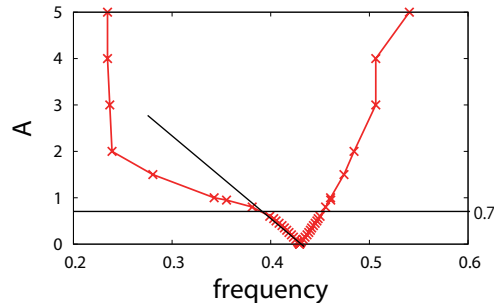


図 3.1: アーノルドタンク (1 調波)

以上の注入正弦波の振幅の制限内の振幅で周波数特性を取得し、図 3.2 のように中心の同期している領域と両端の同期していない領域でフィッティングを行う。フィッティングに用いる式は、左端の同期していない領域を

$$a/\Delta\omega + b, \tag{3.27}$$

中心の同期している領域を

$$c\Delta\omega + d, \tag{3.28}$$

右端の同期していない領域を

$$e/\Delta\omega + f, \tag{3.29}$$

とした。

また、低い振幅の正弦波の引き込み範囲を推定するためには、複数の振幅の引き込み範囲が必要となる。そのため、高い振幅から小さくしていき、その都度引き込み範囲を取得する。アーノルドタンクを推定する際の誤差を小さくするために、引き込み範囲を測定する振幅の測定点は多く、測定幅は大きくすることが望ましい。このように測定した複数の振幅での引き込み範囲からアーノルドタンクを推定することができる。位相方程式が成り立つとき、アーノルドタンクは図 3.3 のように線形的であるため、アーノルドタンクの $\Delta\omega = 0$ を中心にして、(3.28) でフィッティングでき、振幅が小さい時の引き込み可能な注入正弦波の周波数 Ω の最大値、最小値を推定できる。推定した Ω の最大値、最小値から振動子の自然発振周波数 ω_0 の調波数倍である $n\omega_0$ を引くことで振幅が小さい時の $\Delta\omega$ を求める事ができる。

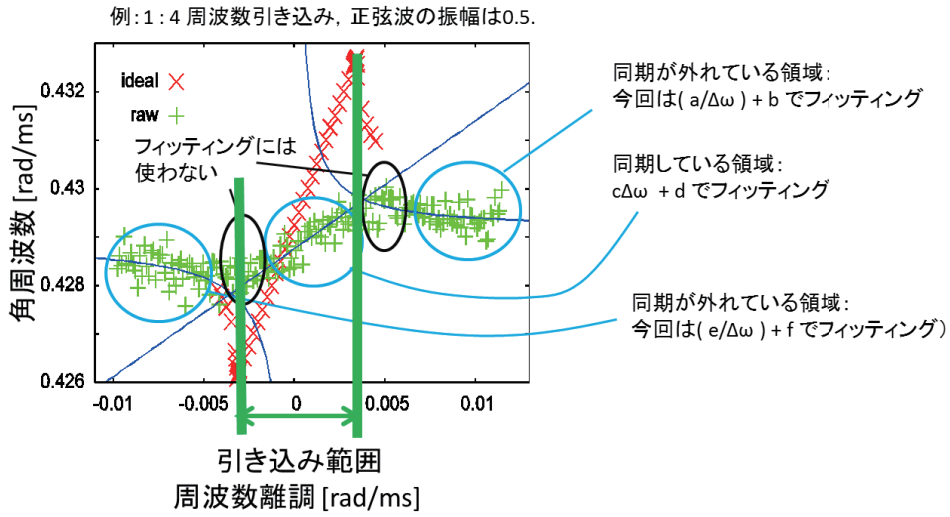


図 3.2: 周波数特性のフィッティング

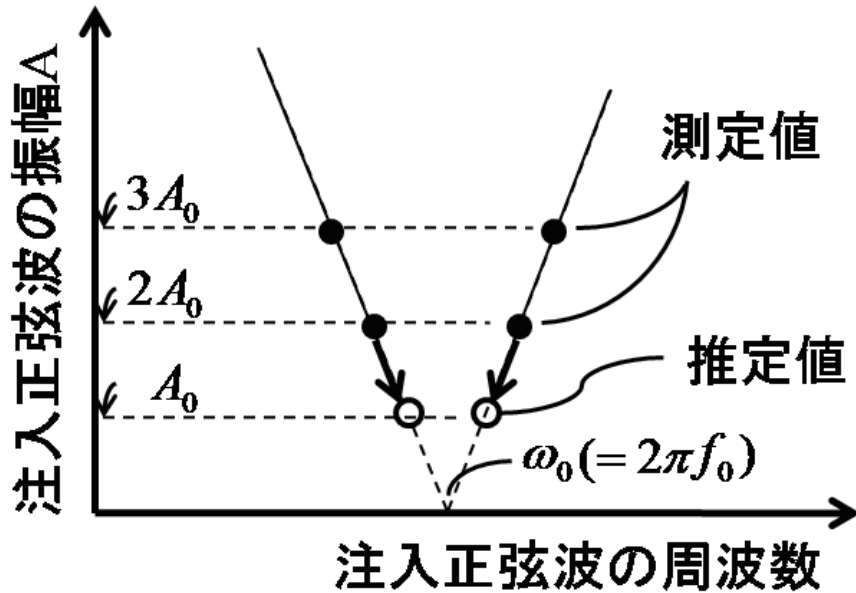


図 3.3: アーノルドタングの模式図

次に、位相差 $n\Delta\psi$ を求める方法を以下に示す。まず、引き込み特性の式 (3.26) より、 $\Delta\omega$ が最大となる時、つまり引き込み可能範囲の右端のときの位相差は $n\Delta\psi = \alpha - \frac{\pi}{2}$ となる。同様に、 $\Delta\omega$ が最小となる時、つまり引き込み可能範囲の左端のときの位相差は $n\Delta\psi = \alpha + \frac{\pi}{2}$ となる。ここで、 α とは注入正弦波の周波数を振動子の自然発振周波数とした時の位相差 ($\Delta\omega = 0$ の時 $n\Delta\psi = \alpha$) である。ただし、振動子の自然発振周波数は、ノイズ環境下を想定する場合、そのノイズを加えた時の自然発振周波数を用いる。つまり、ノイズの大きさによって振動子の自然発振周波数 ω_0 は異なる。振幅が大きければ大きいほど、位相差は安定するが、振幅が大きすぎると位相方程式の適用外となってしまう。そのため、前述したアーノルドタンクが線形的になっている部分の最大振幅とした正弦波を用いて位相差 α を測定した。位相差 α の測定方法は、位相差のヒストグラムのピーク値とする。 α が求められたので、 $n\Delta\psi = \alpha \pm \frac{\pi}{2}$ により $\Delta\omega$ の最小値、最大値の位相差 $\Delta\psi$ も求める事ができる。

以上より、振幅が小さい時の引き込み範囲の右端、左端の $(\Delta\omega, n\Delta\psi)$ と注入正弦波の周波数を振動子の自然発振周波数の調波数倍とした時の $(\Delta\omega, n\Delta\psi) = (0, \alpha)$ の計3点を推定する事ができた。この3点を用いて式 (3.26) よりフィッティングを行い、フーリエ係数 a_n, b_n を推定する。図 3.4 は1調波の時のフィッティング例である。以上の方法で各調波のフーリエ係数を求める事により、位相応答曲線を推定できる。

3.2.3 位相応答曲線の推定のノイズ耐性の検証

前述したとおり、菊地らの推定手法におけるノイズ耐性を再検証する必要がある。したがって、2つの振動子に対してノイズがある環境下を想定し、位相応答曲線の推定を行った。

Hodgkin-Huxley 方程式の位相応答曲線の推定

ノイズの標準偏差 0.5 の場合を考える。図 3.5, 3.6, 3.7, 3.8 としたときのアーノルドタンクを示す。また、ノイズのない環境下でのアーノルドタンクも同時に示す。今回、微小といえる正弦波の振幅を 0.35 とした。これはインパルス応答法の微小インパルスのパワーと揃えたものである。ここで、微小インパルスの「微小」とは測定した位相応答曲線の波形がくずれないときのインパルスである。図 3.5, 3.6, 3.7, 3.8 より、どの調波でもノイズのない環境下でのときのアーノルドタンクとほぼ一致していることがわかる。1調波～3調波までは微小といえる注入正弦波の振幅 0.35 以下で引き込み特性を得る事が出来た。しかし、4調波では、周波数

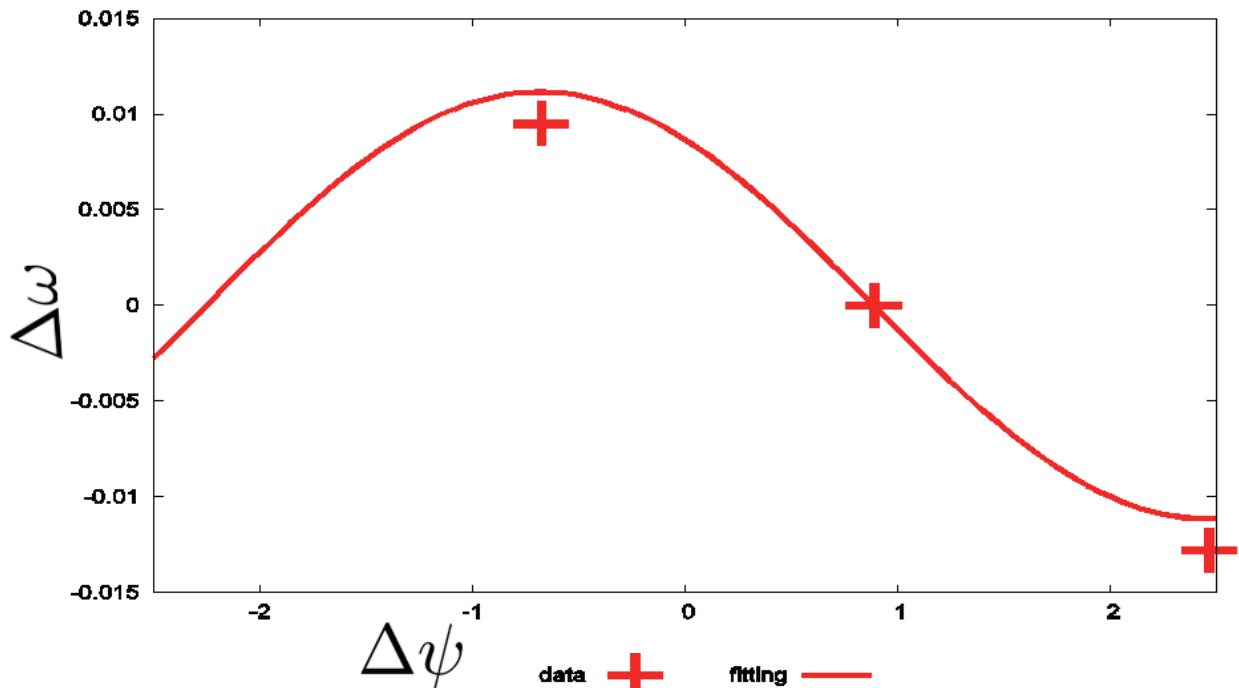


図 3.4: 引き込み特性のフィッティング

特性がノイズの影響によりバラついてしまい、振幅が微小といえるときの引き込み特性を得る事が出来なかった。そのため、菊地らの手法のポイントである振幅の大きい場合の引き込み特性から振幅の小さい場合の引き込み特性を推定した。

以上より推定した位相応答曲線は図??のようになる。推定精度の比較対象として [3] の手法、ノイズのない環境下でインパルス応答法により得た位相応答曲線を同時に示す。インパルス応答法により得た位相応答曲線が理想的な位相応答曲線のため、これに近いほど精度が良いといえる。[3] の手法を緑の鎖線、菊地らの手法を赤の実線、インパルス応答法を黒の破線で示す。

図 3.9 より、[3] の手法と比べて、菊地らの手法の方が精度が良いことがわかる。よって、従来手法よりノイズ耐性が強いといえる。

しかし、ノイズの標準偏差を 1.0 以上とした場合、周波数特性をフィッティングすることができない。図 3.10 はノイズの標準偏差 1.0, 注入正弦波の振幅 1.0, 4 調波のときの周波数特性で、ほぼ平らになっているためフィッティングできないことがわかる。

そのため、菊地らの手法では、ノイズの標準偏差 0.5 のとき精度よく位相応答曲線を推定で

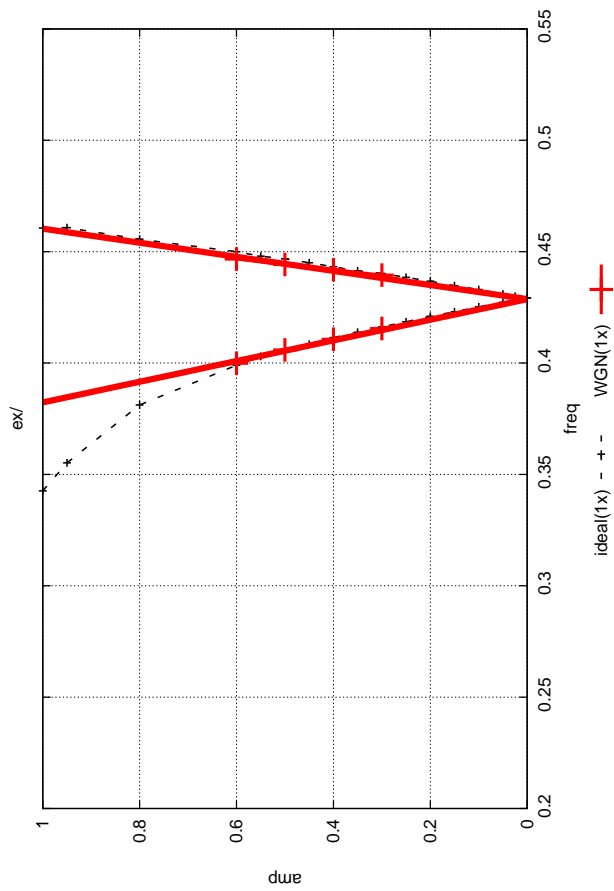


図 3.5: 1 調波のアーノルドタンク (Hodgkin-Huxley 方程式)

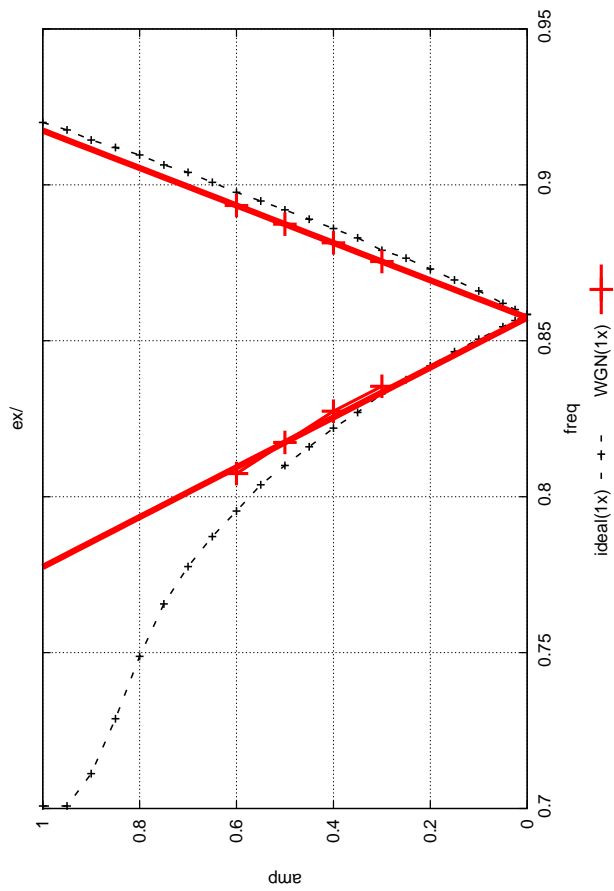


図 3.6: 2 調波のアーノルドタンク (Hodgkin-Huxley 方程式)

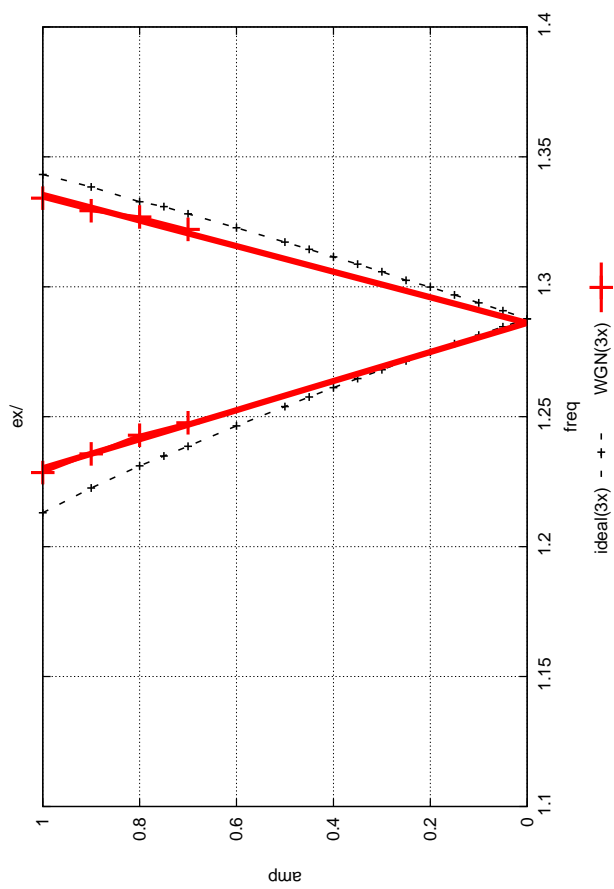


図 3.7: 3 調波のアーノルドタング (Hodgkin-Huxley 方程式)

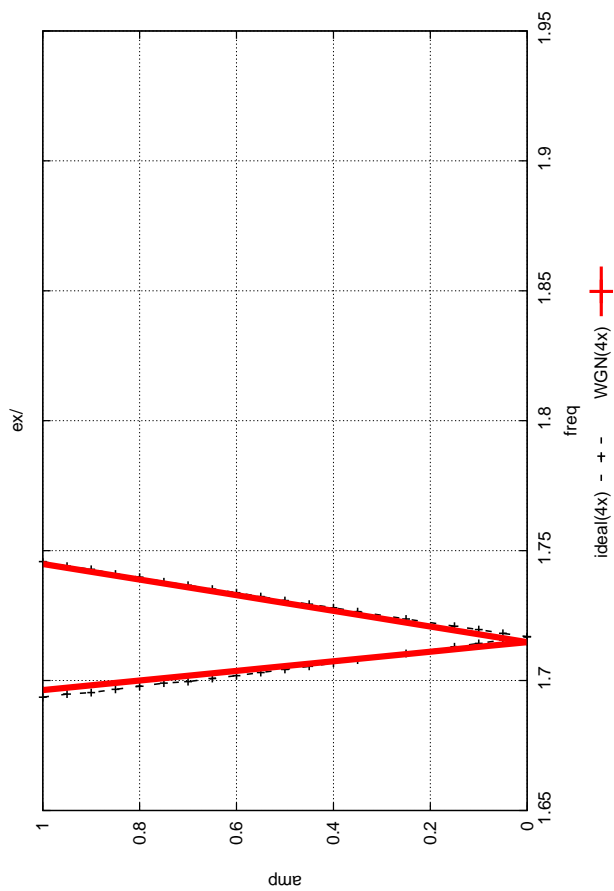


図 3.8: 4 調波のアーノルドタンク (Hodgkin-Huxley 方程式)

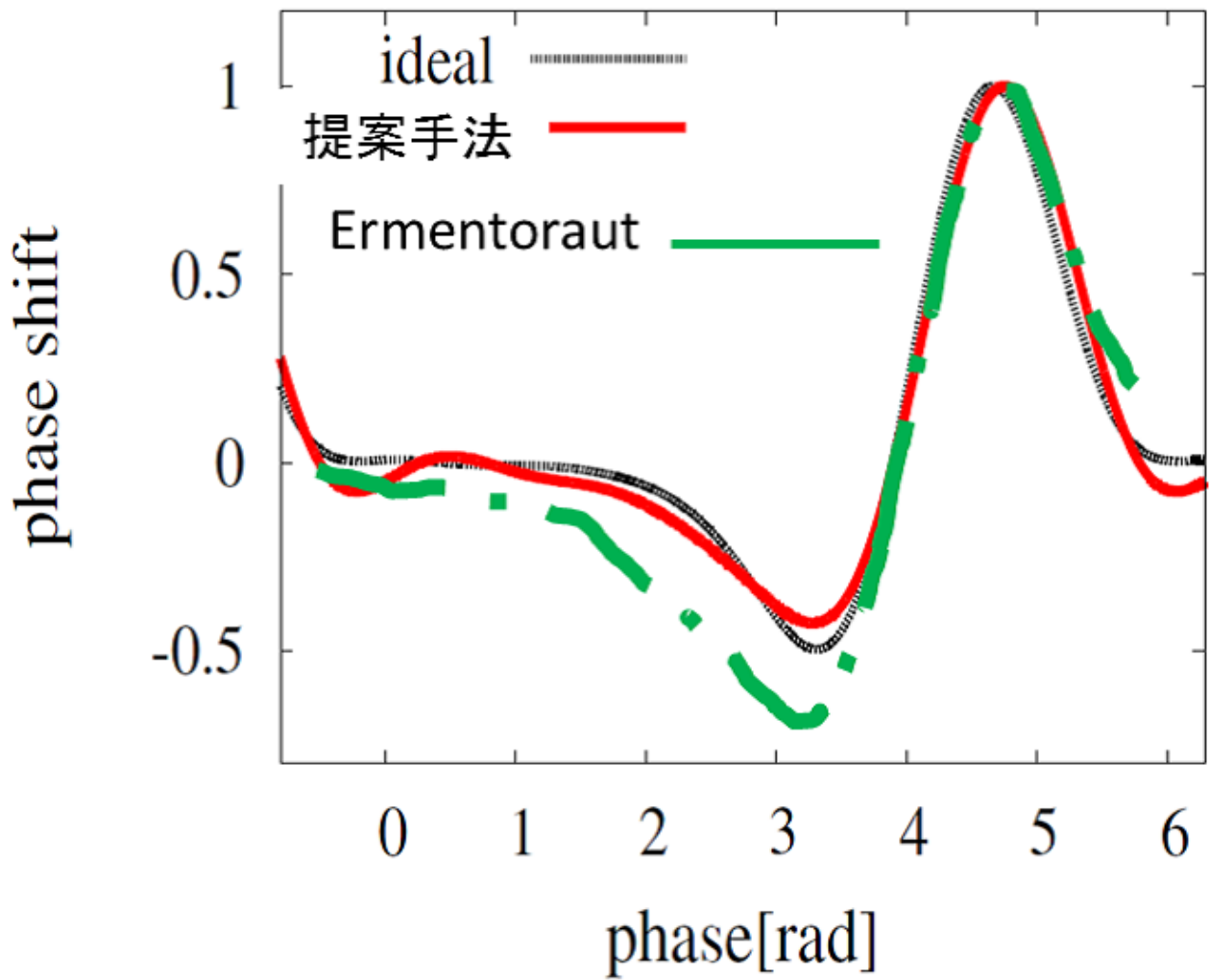


図 3.9: 推定した位相応答曲線 (Hodgkin-Huxley 方程式, ノイズの標準偏差 0.5)

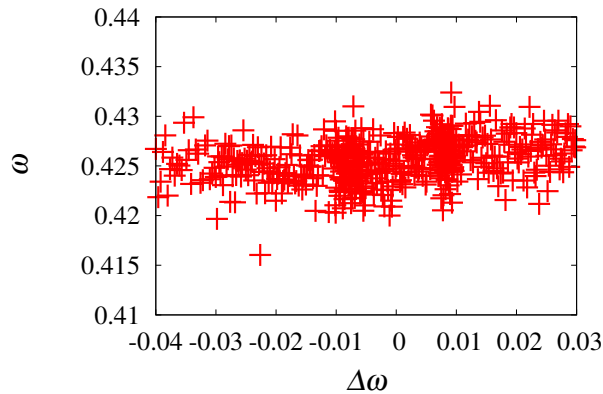


図 3.10: 周波数特性 (Hodgkin-Huxley 方程式, ノイズの標準偏差 1.0, 4 調波)

き, 標準偏差を 1.0 以上とした時, 位相応答曲線を推定できないことがわかった.

Rössler 方程式の位相応答曲線の推定

同様に, ノイズの標準偏差 0.5 の場合を考える. 1 調波の時のアーノルドタンクを図??に示す. ここで, Rössler 方程式は非線形性が強くないため 1 調波のみで位相応答曲線を推定できる.

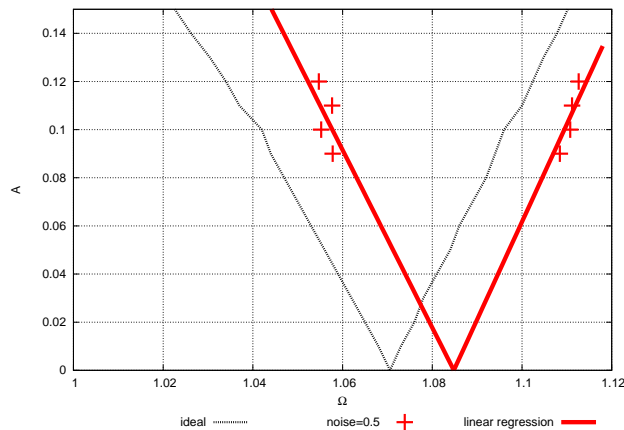


図 3.11: 1 調波のアーノルドタンク (Rössler 方程式)

また, Hodgkin-Huxley 方程式の時と同様に Rössler 方程式に対する微小正弦波の振幅を求めると, 微小といえる正弦波の振幅は 0.13 である. 図 3.11 より, 振幅 0.13 いかでも引き込み特性を測定できている. したがって, より小さい振幅を推定する必要はなく, 位相応答曲線

を推定できる. ノイズの標準偏差が 0.5 のときの位相応答曲線を図??に示す.

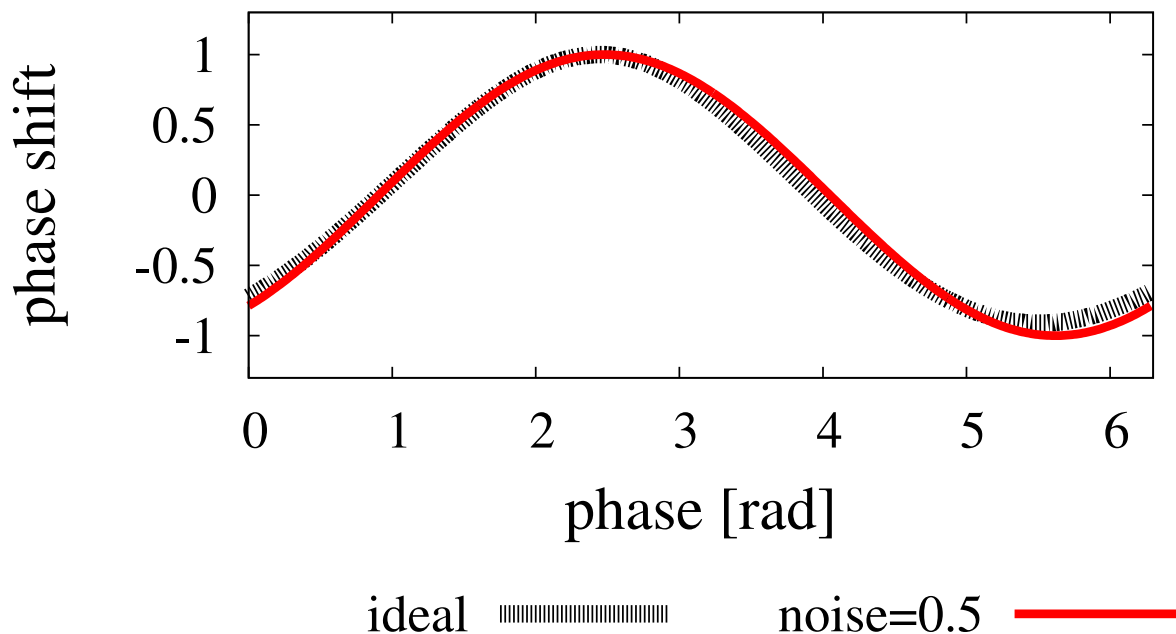


図 3.12: 推定した位相応答曲線 (Rössler 方程式, ノイズの標準偏差 0.5)

図??より, 精度よく位相応答曲線を推定できたことがわかった. ここで, Rössler 方程式は 1 調波のみの単純な波形ということ考えると, よりノイズを大きくしても精度よく位相応答曲線を推定できると期待されるかもしれない. しかし, ノイズの標準偏差を 1.2 とした場合の変数 x の時間変化を図 3.13 を以下に示すが, 波形がドリフトしてしまっていることが見て取れる. そのため, 引き込みを利用する菊地らの手法は適用できなくなってしまう事がわかる. したがって, ノイズの標準偏差 0.5 の時は精度よく位相応答曲線を推定でき, 1.2 以上では推定できないことがわかった.

3.3 ノイズ環境下での最適波形の導出

3.3.1 確立分布のピーク値を最大化する周期信号の理論的導出

本節では FP 方程式を用いて確率分布のピーク値を最大にする周期外力を理論的に導出する. まずノイズに比べて周期外力が非常に弱い場合, つまり確率分布がフラットになる場合につ

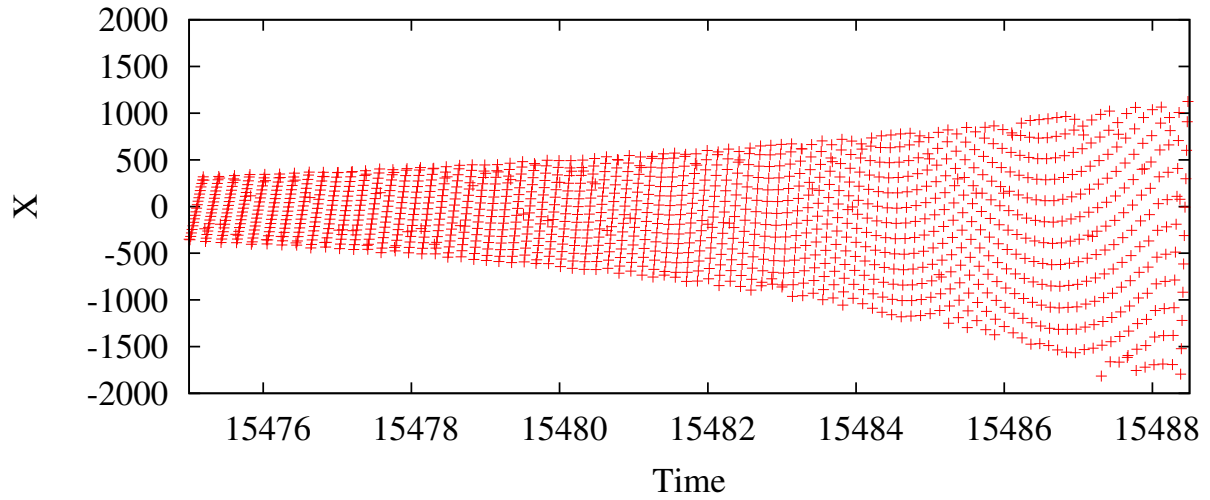


図 3.13: x の時間変化 (Rössler 方程式, ノイズの標準偏差 1.2)

いて導出し. 次に周期外力が強い場合, つまり確率分布が鋭いピークをもつ場合について導出する.

周期外力が弱い場合

FP 方程式を示す.

$$\begin{aligned} \frac{\partial}{\partial t} P(\psi, t) = & -\epsilon^2 \frac{\partial}{\partial \psi} [(\Delta + \Gamma(\psi))P(\psi, t)] \\ & + \epsilon^2 \tilde{D} \frac{\partial^2}{\partial \psi^2} P(\psi, t) + O(\epsilon^4) \end{aligned}$$

右辺の第 3 項は ϵ^4 オーダーと極めて小さいので無視して (3.30) とする.

$$\begin{aligned} \frac{\partial}{\partial t} P(\psi, t) &= -\epsilon^2 \frac{\partial}{\partial \psi} [(\Delta + \Gamma(\psi))P(\psi, t)] + \epsilon^2 \tilde{D} \frac{\partial^2}{\partial \psi^2} P(\psi, t) \\ &= -\epsilon^2 \frac{\partial}{\partial \psi} [(\Delta + \Gamma(\psi))P(\psi, t) + \frac{\partial}{\partial \psi} \tilde{D} P(\psi, t)] \\ &= -\epsilon^2 \frac{\partial}{\partial \psi} J(\psi, t) \end{aligned} \tag{3.30}$$

$P(\psi, t)$ が定常解 $P_s(\psi)$ となったとき, 時間変化量は 0 となるので $\frac{\partial}{\partial t} P_s(\psi) = 0$ となる.

$\frac{\partial}{\partial \psi} P_s(\psi) = 0$ が成立するためには右辺 $J(\psi, t)$ が定数値をとる必要がある. この J_0 を定数値

とすると定常解のとき，以下を満たさなければならない．

$$\begin{aligned} J_0 &= (\Delta + \Gamma(\psi))P_s(\psi) - \tilde{D}\frac{\partial}{\partial\psi}P_s(\psi) \\ -\frac{J_0}{D} &= \frac{\partial}{\partial\psi}P_s(\psi) - \tilde{D}^{-1}(\Delta + \Gamma(\psi))P_s(\psi) \end{aligned} \quad (3.31)$$

次に $\frac{\partial}{\partial\psi}P_s = \tilde{D}^{-1}(\Delta + \Gamma(\psi))P_s$ を解くと

$$P_s = \exp\left[\int_0^\psi \tilde{D}^{-1}(\Delta + \Gamma(\psi'))d\psi' + C\right] = c \exp B(\psi) \quad (3.32)$$

となる．ただし $B(\psi) = \int_0^\psi \tilde{D}^{-1}(\Delta + \Gamma(\psi'))d\psi'$ と置いた．ここで (3.32) を (3.31) に代入すると

$$\frac{\partial}{\partial\psi}[c \exp B(\psi)] - \tilde{D}(\Delta + \Gamma(\psi))c \exp B(\psi) = -\frac{J_0}{D} \quad (3.33)$$

となる．(3.33) の両辺を積分すると

$$C = \int_0^\psi -\frac{J_0}{D} \exp(-B(\psi'))d\psi' + c \quad (3.34)$$

を得る．これを (3.32) に代入すると

$$P(\psi) = c \exp B(\psi) - \frac{J_0}{D} \int_0^\psi \exp[B(\psi) - B(\psi')] \quad (3.35)$$

となる．ここで周期境界条件 ($P(0) = P(2\pi)$) を用いると

$$P(0) = c = P(2\pi) = c \exp[B(2\pi)] - \frac{J_0}{\tilde{D}} \int_0^{2\pi} \exp[B(2\pi) - B(\psi')]d\psi' \quad (3.36)$$

となる．これを变形すると

$$\frac{J_0}{D} = \frac{c[\exp B(2\pi) - 1]}{\oint \exp[B(2\pi) - B(\psi')]d\psi'} \quad (3.37)$$

ここで $P_s(\psi)$ が偶関数である場合， $P_s(\psi) = P_s(-\psi)$ ， $\frac{\partial}{\partial\psi}P_s(\psi) = -\frac{\partial}{\partial\psi}P_s(\psi)$ が成り立つ．

よって，(3.31) より

$$\begin{aligned} \tilde{D}\frac{\partial}{\partial\psi}P_s(-\psi) &= (\Delta + \Gamma(-\psi))P_s(-\psi) - J_0 \\ &= -\tilde{D}\frac{\partial}{\partial\psi}P_s(\psi) = -(\Delta + \Gamma(\psi))P_s(\psi) + J_0 \end{aligned} \quad (3.38)$$

となる．(3.38) が成立するためには $\Delta + \Gamma(-\psi) = -(\Delta + \Gamma(\psi))$ ， $J_0 = 0$ でなければならない．

$\Delta + \Gamma(\psi) = -(\Delta + \Gamma(-\psi))$ である場合， $B(\psi) = \int_0^\psi \tilde{D}^{-1}(\Delta + \Gamma(\psi'))d\psi'$ より $B(\psi) = B(-\psi)$ となる．さらに (3.36) より $B(2\pi) = 0$ ， $J_0 = 0$ を得る．よって P_ψ が偶関数である時，

$$P_s(\psi) = c \exp B(\psi) = P_s(-\psi) = c \exp B(-\psi) \quad (3.39)$$

$$J_0 = 0 \quad (3.40)$$

となる。このとき

$$\Delta + \Gamma(\psi) \quad (3.41)$$

$$\Delta + \Gamma(0) = 0 \quad (3.42)$$

であり、 $\Delta + \Gamma(\psi)$ は奇関数で表される。よって $\Delta + \Gamma(\psi)$ が奇関数であることと $P_s(\psi)$ が偶関数であることは等価であるといえる。以下では $\Delta + \Gamma(\psi)$ が奇関数である場合について $P_s(\psi)$ のピークを求める。まず、(3.31) に $J_0 = 0$ を代入し、 $\psi = 0$ とすると

$$(\Delta + \Gamma(0))P_s(0) - \tilde{D} \frac{\partial}{\partial \psi} P_s(0) = 0$$

(3.42) より

$$\frac{\partial}{\partial \psi} P_s(0) = 0 \quad (3.43)$$

となり、周期関数 P_s は $\psi = 0$ で極値解を得る事がわかる。また $P_s(\psi) = 0$ の定義 (3.30) より

$$-\epsilon^2 \frac{\partial}{\partial \psi} [(\Delta + \Gamma(\psi))P_s(\psi)] + \epsilon^2 \tilde{D} \frac{\partial^2}{\partial \psi^2} P_s(\psi) = 0 \quad (3.44)$$

であり、これを解くと

$$\begin{aligned} \tilde{D} \frac{\partial^2}{\partial \psi^2} P_s(\psi) &= \frac{\partial}{\partial \psi} [(\Delta + \Gamma(\psi))P_s(\psi)] \\ &= \frac{\partial}{\partial \psi} \Gamma(\psi)P_s(\psi) + (\Delta + \Gamma(\psi)) \frac{\partial}{\partial \psi} P_s(\psi) \end{aligned} \quad (3.45)$$

となる。 $\psi = 0$ とすると、

$$\begin{aligned} \frac{\partial^2}{\partial \psi^2} P_s(0) &= \tilde{D}^{-1} \frac{\partial}{\partial \psi} \Gamma(\psi)P_s(0) + \tilde{D}^{-1} (\Delta + \Gamma(\psi)) \frac{\partial}{\partial \psi} P_s(\psi) \\ &= \tilde{D}^{-1} \frac{\partial}{\partial \psi} \Gamma(\psi)P_s(0) \\ &= c \tilde{D}^{-1} \frac{\partial}{\partial \psi} \Gamma(\psi) \end{aligned} \quad (3.46)$$

となる。 $c > 0$, $\tilde{D} > 0$ であるので、 $\frac{\partial}{\partial \psi} \Gamma(\psi) < 0$ のとき、 $P_s(\psi)$ は $\psi = 0$ で極大となる。次に $P_s(0)$ を極大化するような入力信号 $q(\theta)$ を考える。 $q(\theta)$ 及び振動子と入力信号の離調周波数 Δ が極めて小さいものであることを表すため、 $|\alpha| (\ll 1)$ を導入し、上の2つを $\alpha q(\theta)$, $\alpha \Delta$ と書き改める。さらに、入力信号は以下の制約が課せられているとする。

$$Q = \oint (\alpha q(\theta))^2 d\theta = \alpha^2 \oint q^2(\theta) d\theta = \alpha^2 \bar{Q} \quad (3.47)$$

Q はパワーを表しており，2章における P と同義である．本章では FP 方程式の分布関数 P と混在してしまうのをさけるため，パワーを P とおいた．

ここで， $\alpha\Delta + \Gamma(\psi)$ は奇関数であり，

$$P_s(\psi) = c \exp B(\psi) \quad (3.48)$$

$$\oint P_s(\psi) d\psi = c \oint \exp B(\psi) d\psi = 1 \quad (3.49)$$

$P_s(0) = c$ であるので，確率分布のピークを高くすることは c を極大化することである．

(3.49) より $c = \frac{1}{\oint \exp B(\psi) d\psi}$ である． c を極大化することは $\oint \exp B(\psi)$ を極小化することと同義である．以上より (3.47) の制約条件により， $\oint \exp B(\psi) d\psi$ を極小化することは以下の変分問題を解くことと等価である．

$$\begin{aligned} T[q] &= \oint \exp B(\psi) d\psi - \lambda \left[\oint q^2(\theta) d\theta - \bar{Q} \right] \\ &= \oint \exp \left[\int_0^\psi \left(\frac{\alpha\Delta}{\bar{D}} + \frac{\alpha}{2\pi} \int_0^{2\pi} Z(\theta + \psi) q(\theta) d\theta \right) d\psi' \right] d\psi \\ &\quad - \lambda \left[\oint q^2(\theta) d\theta - \bar{Q} \right] \end{aligned} \quad (3.50)$$

ここで λ はラグランジュの未定乗数である．以下より，(3.50) を解き， $P_s = c$ を極大化する入力信号 $q_*(\theta)$ を導出する．まず， $B(\psi)$ をマクローリン展開すると以下のようなになる．

$$\exp B(\psi) = 1 + B(\psi) + O(\alpha^2) \quad (3.51)$$

さらに

$$\oint \exp B(\psi) = 2\pi + \oint B(\psi) d\psi + O(\alpha^2) \quad (3.52)$$

とし， $O(\alpha^2)$ は極めて小さいパラメータであるため無視する．すると， $\exp B(\psi) d\psi$ を極小化することは $\oint B(\psi) d\psi$ を極小化することと等価である．よって，(3.50) は

$$\begin{aligned} T[q] &= \oint B(\psi) d\psi - \lambda \left[\oint q^2(\theta) d\theta - \bar{Q} \right] \\ &= \oint \int_0^\psi \frac{\alpha}{\bar{D}} \left[\Delta + \frac{1}{2\pi} \oint Z(\theta + \psi') q(\theta) d\theta \right] d\psi' d\psi \\ &\quad - \lambda \left[\oint q^2(\theta) d\theta - \bar{Q} \right] \end{aligned} \quad (3.53)$$

と書き直せる．第一変分 $\delta T[q]$ を求めると，

$$\begin{aligned}\delta T[q] &= \oint \int_0^\psi \frac{\alpha}{\tilde{D}} \left[0 + \frac{1}{2\pi} \oint Z(\theta + \psi') q(\theta) d\theta \right] d\psi' d\psi \\ &\quad - \lambda [2 \oint q(\theta) \delta(\theta) d\theta - 0] \\ &= \oint [\delta(\theta) \oint \int_0^\psi \frac{\alpha}{2\pi\tilde{D}} Z(\theta + \psi') d\psi' d\psi - 2\lambda q(\theta)] d\theta\end{aligned}\tag{3.54}$$

となる．

(3.54) が $\delta T[q_*] = 0$ を満たす q_* は

$$\begin{aligned}2\lambda q_*(\theta) &= \oint \int_0^\psi \frac{\psi}{0} \frac{\alpha}{2\pi\tilde{D}} Z(\theta + \psi') d\psi' d\psi \\ q_*(\theta) &= \frac{\alpha}{4\lambda\pi\tilde{D}} \oint \int_0^\psi Z(\theta + \psi) d\psi' d\psi\end{aligned}\tag{3.55}$$

と与えられる．ここで $Z(\psi) = \sum_{n=-\infty}^{\infty} Z_n e^{in\psi}$ とすると

$$\begin{aligned}\int_0^\psi Z(\psi) d\psi' &= \int_0^\psi [Z_0 + \sum' Z_n e^{in\psi}] d\psi' \\ &= Z_0\psi + \sum' \frac{Z_n}{in} e^{in\psi} - \sum' \frac{Z_n}{in}\end{aligned}\tag{3.56}$$

となる．ここで \sum' は $\sum_{n=-\infty}^{\infty}$ の内， $n=0$ を含まない総和である．(3.56) を用いると，

$$\begin{aligned}\int_0^\psi Z(\theta + \psi') d\psi' &= z_0\psi + \int_0^{\theta+\psi} \sum' z_n e^{in\psi'} d\psi' - \int_0^\theta \sum' z_n e^{in\psi'} d\psi' \\ &= z_0\psi + \sum' \frac{z_n}{in} e^{in\theta} - \sum' \frac{z_n}{in}\end{aligned}\tag{3.57}$$

となり，これを (3.55) に代入すると

$$\begin{aligned}2\lambda q_*(\theta) &= \frac{\alpha}{2\pi\tilde{D}} \oint [z_0\psi + \sum' \frac{z_n}{in} e^{in(\theta+\psi)} - \sum' \frac{z_n}{in} e^{in\theta}] d\psi' \\ &= \frac{\alpha}{2\pi\tilde{D}} \left(\frac{4\pi^2 z_0}{2} - \sum' \frac{z_n}{in} e^{in\theta} 2\pi \right) \\ &= \frac{\alpha}{\tilde{D}} (z_0\pi - \sum' \frac{z_n}{in} e^{in\theta}) \\ &= \frac{\alpha}{\tilde{D}} (M - X(\theta))\end{aligned}\tag{3.58}$$

となる．ここで $z_0\pi = M$ ， $\sum' \frac{z_n}{in} e^{in\theta} = X(\theta)$ とおいた．よって， $q_*(\theta)$ は

$$q_*(\theta) = \frac{\alpha}{2\lambda\tilde{D}} (M - X(\theta))\tag{3.59}$$

入力信号のパワー一定という制約条件 (3.47) より,

$$\oint q_*^2 d\theta = \oint \frac{\alpha^2}{4\lambda^2 \tilde{D}^2} (M - X(\theta))^2 d\theta = Q \quad (3.60)$$

$\oint (M - X(\theta))^2 d\theta = S$ とおくと,

$$\begin{aligned} \lambda^2 &= \frac{\alpha^2}{4Q\tilde{D}} S \\ \lambda &= \pm \frac{\alpha}{2\tilde{D}} \sqrt{\frac{S}{Q}} \end{aligned} \quad (3.61)$$

また第2変分 $\delta^2 q$ は

$$\delta^2 T[q] = -2\lambda \oint \delta^2(\theta) d\theta \quad (3.62)$$

よって $\lambda < 0$ のとき第2変分 $\delta^2[q] > 0$ であるので q_* は $T[q]$ の極小解を与える.

つまり, $P_s(\psi) = c$ を極大化する最適入力信号 q_{opt} は

$$\begin{aligned} q_{opt}(\theta) &= \frac{1}{2\lambda_-} (M - X(\theta)) \\ &= \frac{\tilde{D}}{\alpha} \sqrt{\frac{Q}{S}} \frac{\alpha}{\tilde{D}} (X(\theta) - M) \\ &= \sqrt{\frac{Q}{S}} (X(\theta) - M) \end{aligned} \quad (3.63)$$

(3.63) もしくは (3.60) より, $\alpha \ll 1$ のとき, つまり周期外力がノイズに比べて十分小さいとき, 最適解は PRC の積分によって得られる.

この q_{opt} から相互作用関数 Γ を求めると,

$$\begin{aligned} \Gamma_{opt}(\psi) &= \frac{1}{2\pi} \oint Z(\theta + \psi) \alpha q_{opt}(\theta) d\theta \\ &= \frac{\alpha}{2\pi} \int_{n=-\infty}^{\infty} Z_n e^{in(\theta+\psi)} \times \sqrt{\frac{Q}{S}} (\sum' \frac{z_n}{in} e^{in\theta} - z_0\pi) d\theta \\ &= \frac{\alpha}{2\pi} \sqrt{QS} \oint (z_0 + \sum' z_n e^{in(\theta+\psi)}) (\sum' \frac{z_n}{in} e^{in\theta} - z_0\pi) \end{aligned} \quad (3.64)$$

ここで

$$\sum' z_n e^{in(\theta+\psi)} = \sum_{n=1}^{\infty} e^{in(\theta+\psi)} + \sum_{n=1}^{\infty} Z_{-n} e^{-in(\theta+\psi)} \quad (3.65)$$

$$\sum' \frac{z_n}{in} e^{in\theta} = \sum_{n=1}^{\infty} \frac{z_n}{in} e^{in\theta} + \sum_{n=1}^{\infty} \frac{Z_{-n}}{-in} e^{-in\theta} \quad (3.66)$$

のように分解すると

$$\begin{aligned}
\Gamma_{opt}(\psi) &= \alpha \sqrt{\frac{\bar{Q}}{S}} \left(-z_0^2 \pi + \sum_{n=1}^{\infty} z_{-n} \frac{z_n}{in} e^{in\psi} + \sum_{n=1}^{\infty} z_n \frac{z_{-n}}{-in} e^{-in\psi} \right) \\
&= \alpha \sqrt{\frac{\text{bar}Q}{S}} \left(-z_0^2 \pi - 2 \sum_{n \geq 1} \frac{|z_n|^2}{n} \sin n\psi \right)
\end{aligned} \tag{3.67}$$

また $\Delta + \Gamma(\psi)$ は奇関数であるので

$$\begin{aligned}
\alpha \Delta + \Gamma_{opt}(0) &= 0 \\
\alpha \Delta &= -\Gamma_{opt}(0) \\
\alpha \Delta &= -\alpha \sqrt{\frac{\bar{Q}}{S}} \left(-z_0^2 \pi - 2 \sum_{n=1}^{\infty} \frac{|z_n|^2}{n} \sin n\psi \right) \\
\Delta &= z_0^2 \pi \sqrt{\frac{\bar{Q}}{S}}
\end{aligned} \tag{3.68}$$

以上より周期外力が弱い場合における $P(\psi)$ を最大化する周期外力が PRC の積分によって得られる。

周期外力が強い場合

今回、信号がパワー一定という制約条件 (3.47) であるとし、 $P(\psi)$ の傾きの変化量 $|\frac{\partial^2}{\partial \psi^2} P(\psi)|$ が最大となる入力 $q(\theta)$ を考える。この $q(\theta)$ がピークが最も鋭くなる入力信号である。

$\frac{\partial^2}{\partial \psi^2} \Gamma(\psi) < 0$ のとき、 ψ_* で $P_s(\psi)$ は極大となるので、以下では $\frac{\partial^2}{\partial \psi^2} P(\psi)$ を極小化する。

(3.44) より $\frac{\partial}{\partial \psi} P(\psi_*)|_{\psi=\psi_*}$ として

$$\tilde{D} \frac{\partial^2}{\partial \psi^2} P_s(\psi_*)|_{\psi=\psi_*} P_s(\psi_*) \tag{3.69}$$

よって $\Gamma'(\psi_*)$ を極小化すればよい。ここで簡単のため $\psi_* = 0$ とすると、ラグランジュの未定乗数法より

$$\begin{aligned}
K[q] &= \frac{\partial}{\partial \psi} \Gamma(\psi)|_{\psi=0} - \lambda \left[\frac{1}{2\pi} \oint q^2(\theta) d\theta - Q \right] \\
&= \frac{1}{2\pi} \oint Z'(\theta) q(\theta) d\theta - \lambda \left[\frac{1}{2\pi} \oint q^2(\theta) d\theta - Q \right]
\end{aligned} \tag{3.70}$$

を極小化する信号を得ればよい。まず第一変分 $\delta K[q]$ を求めると

$$\delta K[q] = \frac{1}{2\pi} \oint Z'(\theta) \delta(\theta) d\theta + \frac{2\lambda}{2\pi} \oint q(\theta) \delta(\theta) d\theta \tag{3.71}$$

となる．よって $K[q]$ を極小化する信号 q_* は以下で与えられる．

$$\begin{aligned} Z'(\theta) + 2\lambda q_*(\theta) &= 0 \\ q_*(\theta) &= -\frac{1}{2\lambda} Z'(\theta) \end{aligned} \quad (3.72)$$

外部信号の制約条件を満たすには

$$8\pi\lambda^2 Q = \oint (Z'(\theta))^2 d\theta \quad (3.73)$$

$$\lambda^2 = (8\pi Q) \oint (Z'(\theta))^2 d\theta \quad (3.74)$$

とならなければならない，このとき第2変分 $\delta^2 K[q]$ は

$$\delta^2 K[q] - 2\lambda \oint \delta^2(\theta) d\theta = 16\pi Q [\oint (Z'(\theta))^2]^{-1} \oint \delta^2 d\theta > 0 \quad (3.75)$$

となる．よって q_* は $K[q]$ を極小化する．この q_* を用いたとき $\frac{\partial}{\partial \psi} P_s(\psi)|_{\psi=0} = 0$ ， $\frac{\partial^2}{\partial \psi^2} P_s(\psi)|_{\psi=0} > 0$ であり，同時に $P_s(\psi)|_{\psi=0}$ を極小化する．よって注入信号が強い場合，ピークを最も鋭くする信号は振動子の PRC を微分したものであることがわかった．

3.3.2 確立分布のピーク値を対象とした時の理論解と Genetic Algorithm の探索解の比較

本節では，2.3節と同様に理論解と GA との比較結果を示す．2.3節と異なる点は，評価関数を確率分布のピーク値としている点である．図の凡例は f_{dif} を位相応答関数の微分， f_{dif} を位相応答関数の積分， f_{ga} を GA による探索解とする．

Hodgkin-Huxley 振動子を対象とした時の理論解と GA による探索解の比較

まず周期外力が弱い場合として $Q = 0.0001$ としたときの理論解と探索解の比較結果を図 3.14 に示す．また，このときの分布関数の比較結果を図 3.15 に示す．

図 3.14, 3.15 より，周期外力が弱い場合，位相応答関数の積分 f_{int} と等しくなっており，このとき分布関数のピークも最も高くなっていることがわかる．

次に，周期外力が強い場合として $Q = 10$ としたときの理論解と探索解の比較結果を図 3.16 に示す．また，このときの分布関数の比較結果を図 3.17 に示す．図 3.14, 3.15 より，周期外力が強い場合，位相応答関数の微分 f_{dif} と等しくなっており，このとき分布関数のピークも最も高くなっていることがわかる．

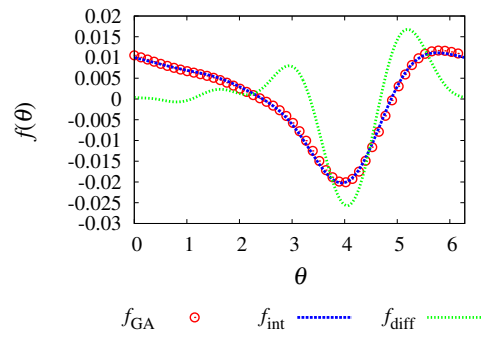


図 3.14: 理論解と探索解の最適波形の比較 ($Q=0.0001$, Hodgkin-Huxley 方程式)

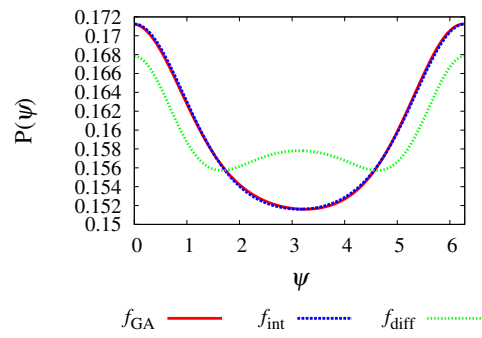


図 3.15: 理論解と探索解の分布関数の比較 ($Q=0.0001$, Hodgkin-Huxley 方程式)

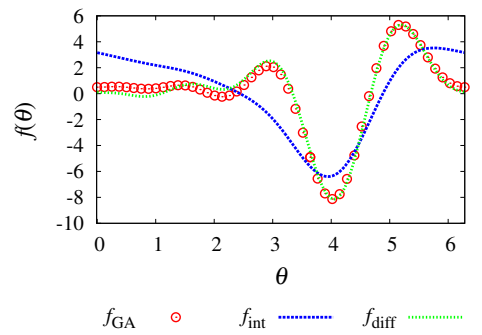


図 3.16: 理論解と探索解の最適波形の比較 ($Q=10$, Hodgkin-Huxley 方程式)

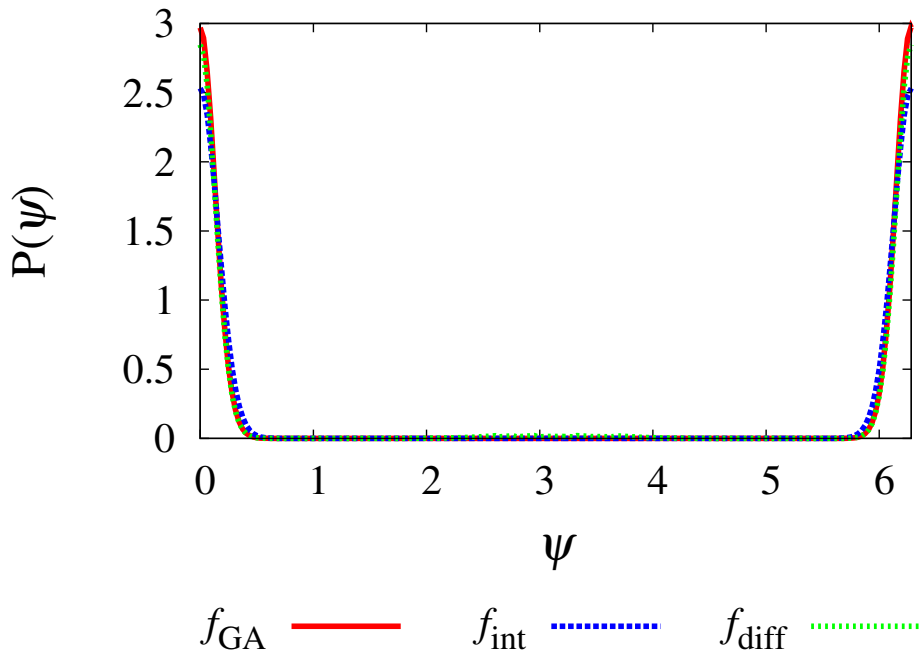


図 3.17: 理論解と探索解の分布関数の比較 ($Q=10$, Hodgkin-Huxley 方程式)

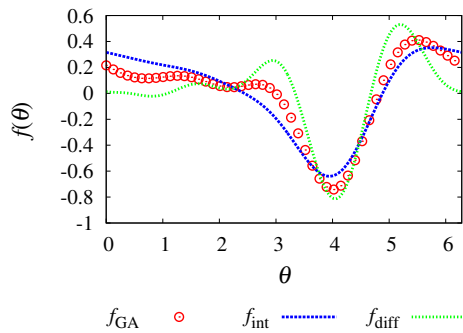


図 3.18: 理論解と探索解の最適波形の比較 ($Q=0.1$, Hodgkin-Huxley 方程式)

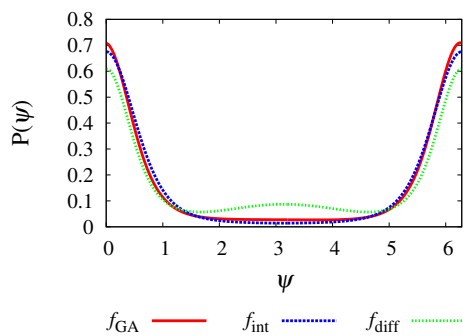


図 3.19: 理論解と探索解の分布関数の比較 ($Q=0.1$, Hodgkin-Huxley 方程式)

最後に、理論解と一致しなかった $Q = 0.1$ とした場合の比較結果を図 3.18, 3.19 に示す。図 3.14, 3.15 より、 $Q = 0.1$ の場合、GA による探索解 f_{GA} が位相応答関数の微分 f_{diff} と積分 f_{int} どちらとも一致していないことがわかる。そこで、探索解が最も最適であることを示すために、ランダム関数との分布関数と比較した。図 3.20, 3.20 より、GA による探索解が最

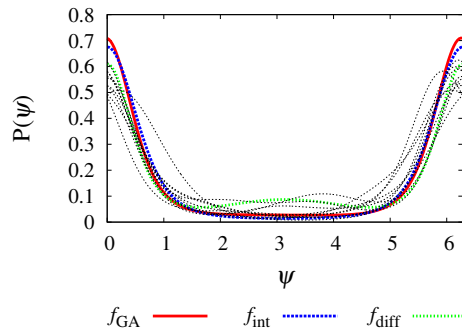


図 3.20: ランダム関数との分布関数の比較 ($Q=0.1$, Hodgkin-Huxley 方程式)

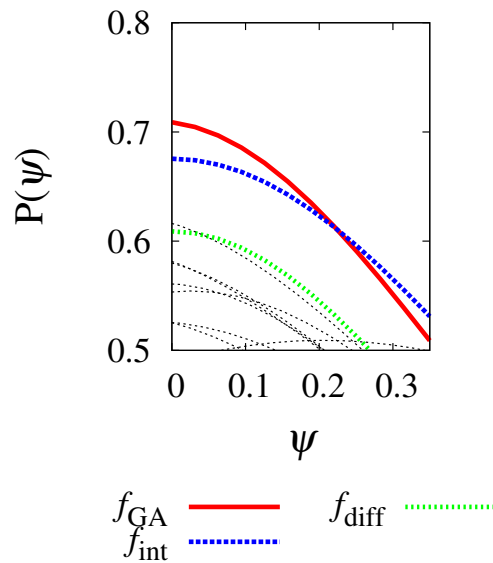


図 3.21: ランダム関数との分布関数の比較 ($Q=0.1$, Hodgkin-Huxley 方程式, ピーク付近の拡大)

も分布関数のピーク値が高くなっていることがわかる。この結果より、設計した GA では最も最適な波形を得る事ができると数値的に証明された。

Rössler 方程式を対象とした時の理論解と GA による探索解の比較

まず周期外力が弱い場合として $Q = 0.001$ としたときの理論解と探索解の比較結果を図 3.22 に示す。また、このときの分布関数の比較結果を図 3.24 に示す。

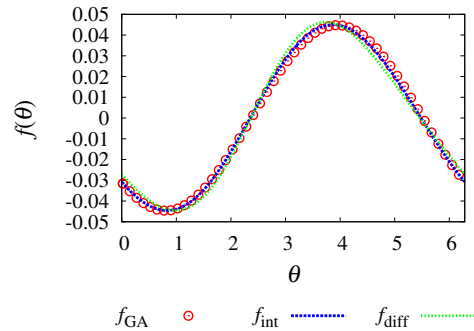


図 3.22: 理論解と探索解の最適波形の比較 ($Q=0.001$, Rössler 方程式)

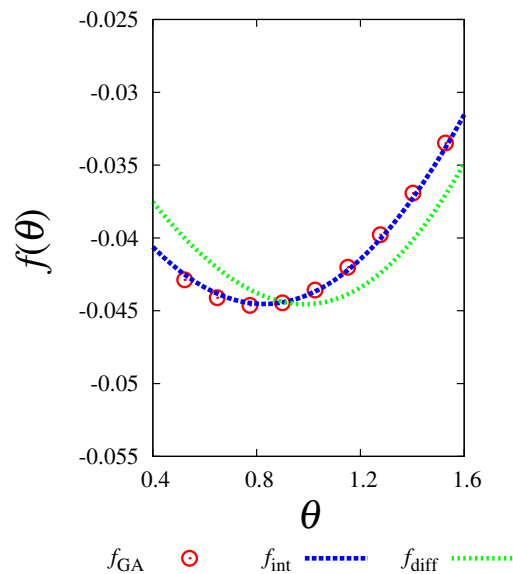


図 3.23: 理論解と探索解の最適波形の比較 ($Q=0.001$, Rössler 方程式, 拡大)

図 3.22, 3.24 より、周期外力が弱い場合、位相応答関数の積分 f_{int} と等しくなっており、このとき分布関数のピークも最も高くなっていることがわかる。

次に、周期外力が強い場合として $Q = 0.1$ としたときの理論解と探索解の比較結果を図 3.26 に示す。また、このときの分布関数の比較結果を図 3.28 に示す。

図 3.26, 3.26 より、周期外力が強い場合、位相応答関数の微分 f_{diff} と等しくなっており、

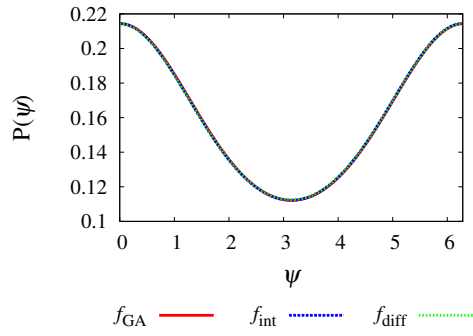


図 3.24: 理論解と探索解の分布関数の比較 ($Q=0.001$, Rössler 方程式)

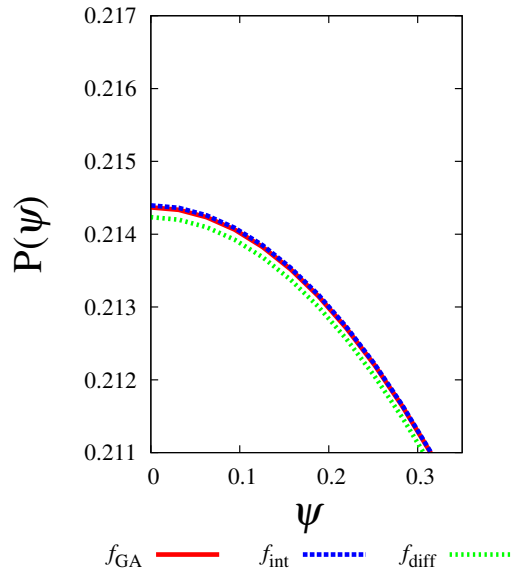


図 3.25: 理論解と探索解の分布関数の比較 ($Q=0.001$, Rössler 方程式, ピーク付近の拡大)

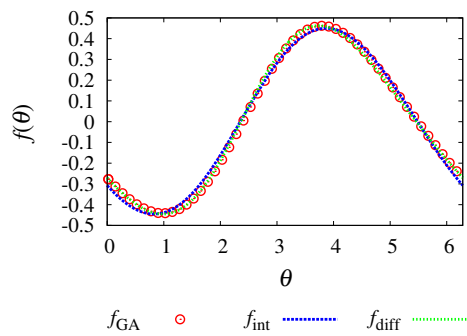


図 3.26: 理論解と探索解の最適波形の比較 ($Q=0.1$, Rössler 方程式)

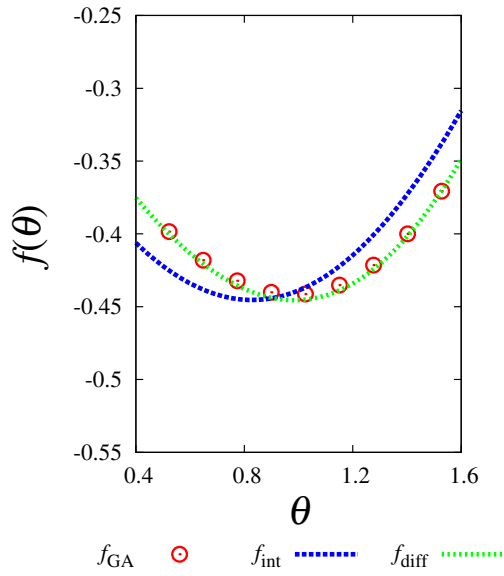


図 3.27: 理論解と探索解の最適波形の比較 ($Q=0.1$, Rössler 方程式, 拡大)

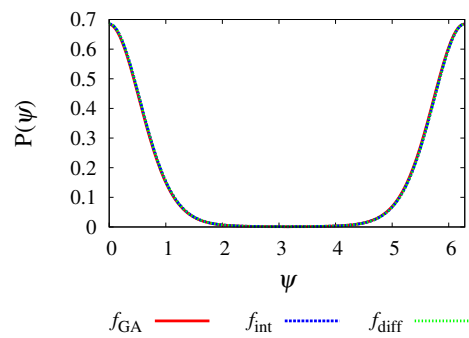


図 3.28: 理論解と探索解の分布関数の比較 ($Q=0.1$, Rössler 方程式)

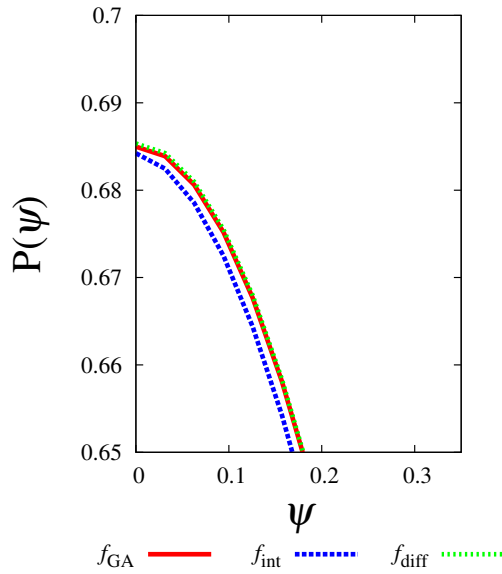


図 3.29: 理論解と探索解の分布関数の比較 ($Q=0.1$, Rössler 方程式, 拡大)

このとき分布関数のピークも最も高くなっていることがわかる.

最後に, 理論解と一致しなかった $Q = 0.01$ とした場合の比較結果を図 3.30, 3.32 に示す.

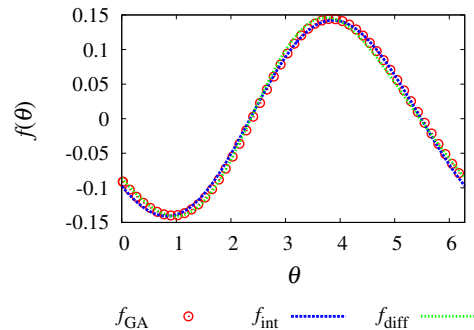


図 3.30: 理論解と探索解の最適波形の比較 ($Q=0.01$, Rössler 方程式)

図 3.30, 3.32 より, $Q = 0.01$ の場合, GA による探索解 f_{GA} が位相応答関数の微分 f_{diff} と積分 f_{int} と一致しているように見える. しかし, 図 3.31, 3.33 のように拡大するとどちらとも一致していないことがわかる. そこで, 探索解が最も最適であることを示すために, ランダム関数との分布関数と比較した. 図 3.34, 3.34 より, GA による探索解が最も分布関数のピーク値が高くなっていることがわかる. この結果より, 設計した GA では最も最適な波形を得る事ができると数値的に証明された.

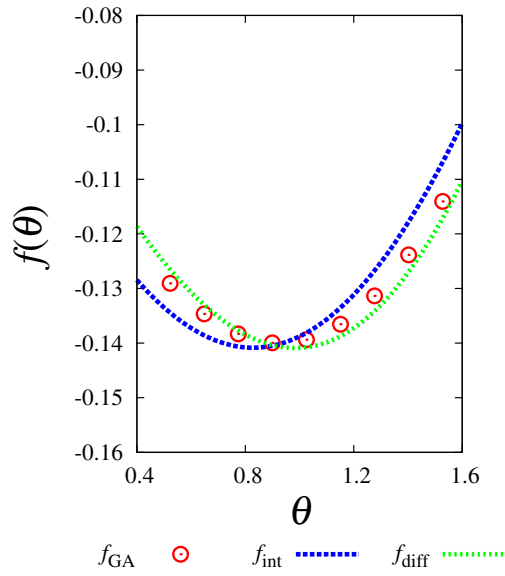


図 3.31: 理論解と探索解の最適波形の比較 (Q=0.01,Rössler 方程式, 拡大)

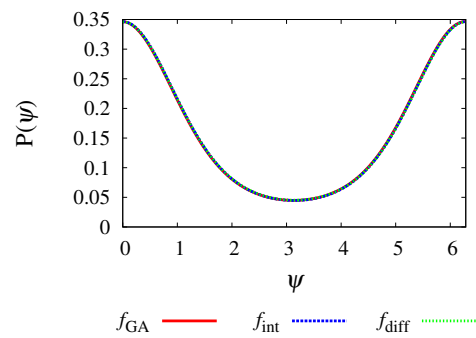


図 3.32: 理論解と探索解の分布関数の比較 (Q=0.01,Rössler 方程式)

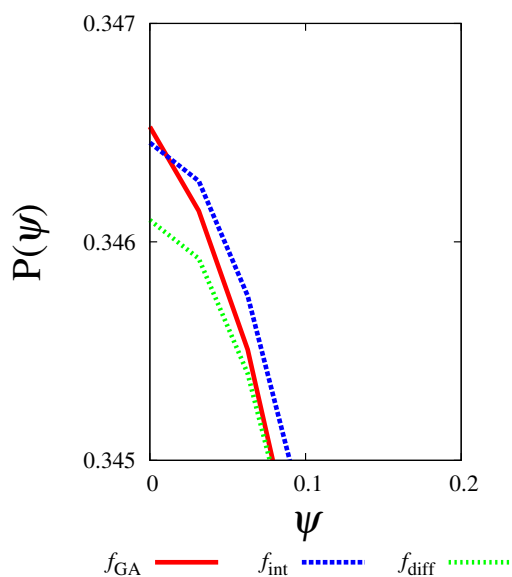


図 3.33: 理論解と探索解の分布関数の比較 ($Q=0.01$, Rössler 方程式, ピーク付近の拡大)

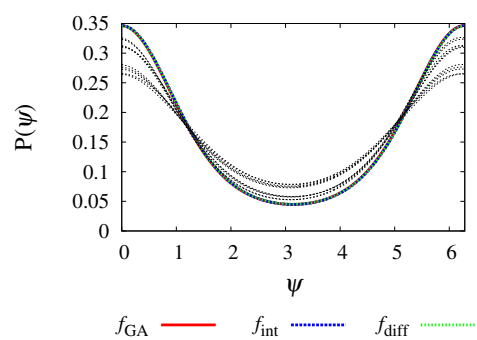


図 3.34: ランダム関数との分布関数の比較 ($Q=0.01$, Rössler 方程式)

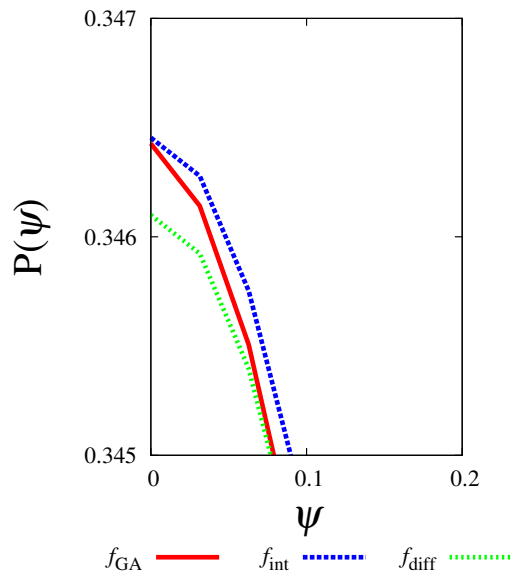


図 3.35: ランダム関数の分布関数の比較 ($Q=0.01$, Rössler 方程式, ピーク付近の拡大)

第4章 結論

本論文では、まずノイズ環境下で菊地らによる位相応答曲線の推定手法を正しい方法でノイズ耐性を検証し直した。結果、ある程度のノイズの大きさまでであれば、従来手法より精度よく推定できることを数値的に導いた。また、本研究で設計されたノイズのない環境下、ノイズ環境下それぞれにおける最適周期信号が確かに引き込み能力を最適化する事を別のアプローチによる Genetic Algorithm により示した。さらに、理論の適用できない範囲でも最適波形を求める事ができた。

今後の課題としては、本研究で用いた位相差の分布関数の Fokker-Planck 方程式を用いて、理論的に位相応答曲線を求める事があげられる。そのための足掛かりとして、今回の正しい設定での数値実験による位相応答曲線の推定は理論の裏付けとなり価値があるといえる。また、注入同期において、同期のロバスト性も重要な課題といえる。これは2章で示した理論を拡張する事で設計可能である。

謝辞

本研究を行うにあたり多くのご指導を頂いた田中久陽先生に心から感謝致します。また、研究室配属より研究をはじめとして多くの助言を頂いた M2 西山英里さん，研究を進めるうえで様々な助言を頂いた M1 飯倉秀策さん，砂田晃良さん，矢部洋司さん，ご卒業された先輩方に深く感謝します。そして，多くの励ましを頂いた B4 倉本淳さん，出口慎太郎さん，小田淳さん，荻原祐介さんに感謝いたします。最後に，心の支えとなって頂いた家族に心より感謝いたします。

関連図書

- [1] A. L. Hodgkin and A. F. Huxley “A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve,” *Bulletin of Mathematical Biology*, Vol. 52, No 1/2pp. 25 – 71, 1990.
- [2] O. E. Rössler “An Equation for Continuous Chaous,” *Physics Letrers*, July 1976.
- [3] G. B. Ermentraout, R. F. Galan, and N. N. Urban, “Relating neural dynamics to neural coding,” *Phys. Rev. Lett.* 99,248103, 2007.
- [4] 蔵本由樹, 「同期現象の数理」, 培風館, 2010.
- [5] A. Hajimiri and T. H. Lee, “A general theory of phase noise in electrical oscillators,” *IEEE J. Solid-State Circuits*, Vol. 33, pp. 179–194, Feb. 1998.
- [6] 三宮信夫, 玉置久, 喜多一, 岩本貴志, 「遺伝アルゴリズムと最適化」, 朝倉書店, 1998.
- [7] C. W. Gardiner, *Handbook of Stochastic Methods*, Springer, 2003.
- [8] 三井斌友, 小藤俊幸, 齋藤善弘, 「微分方程式による計算科学入門」, 共立出版株式会社, 2004.
- [9] 菊地淳弘, 宮崎紀子, 田中久陽, “引き込みのロバスト性による位相応答曲線の推定手法の提案,” *信学技法*, NLP2008-16, pp.43–48, June 2008.

付録

本研究で使ったプログラムの内、代表的なものを記す。

Genetic Algorithm を行うプログラム

- Ga.java

```
1  import java.util.*;
2  import java.io.*;
3
4  class main{//メイン関数
5      public static void main(String [] args){
6          new main();//実行
7
8
9          /*開始時刻取得（実行時間取得のため）*/
10         long starttime = System.currentTimeMillis();
11         Calendar call = Calendar.getInstance();
12         int hour = call.get(Calendar.HOUR_OF_DAY);
13         int minute = call.get(Calendar.MINUTE);
14         int second = call.get(Calendar.SECOND);
15         System.out.println("Start time is "+hour+":"+minute+":"+second);
16
17
18
19         double [] phi=new double[Def.SAMPLE+1];//位相
20         double [][] z=new double[Def.SAMPLE+1][Def.SAMPLE+1];//位相応答関数
21         /*関数定義*/
22         Evolution Evo=new Evolution();//GA における関数
23         Equation Eq=new Equation();//方程式を定義するクラス
24         Makefile Mf=new Makefile();//gnuplot 用ファイルを作成するクラス
25         /*個体の構造体定義*/
26         Ind[] CUR=new Ind[Def.NoI];//現代の個体
27         Ind[] NXT=new Ind[Def.NoI];//次世代の個体
28         for(int i=0;i<Def.NoI;i++){//構造体初期化
29             CUR[i]=new Ind();
30             NXT[i]=new Ind();
31         }
32         /*位相と位相応答関数のサンプリング*/
33         for(int smp=0;smp<=Def.SAMPLE;smp++){
34             phi[smp]=(2*Def.PI/Def.SAMPLE)*smp-Def.PI;
35         for(int smp=0;smp<=Def.SAMPLE;smp++){
36             for(int smp2=0;smp2<=Def.SAMPLE;smp2++){
37                 z[smp][smp2]=Eq.prc(phi[smp]+phi[smp2]*Def.Ntime);
38             }
39         }
40         /******
41         /*GENETIC ALGORITHM*****
42         /******
43         /******GENETIC ALGORITHM*/
44         int[] fopt;
45         if(Def.N_INIT>0)fopt = Evo.initDec2bin();
46         /*初期世代の発生*/
47         int r;
48         for(int i=0;i<Def.NoI;i++){//初期世代の染色体を生成
49             for(int n=0;n<Def.LoC;n++){
50                 r=(int)(Math.random()*2);
51                 if(Def.N_INIT>0&&i<Def.N_INIT) r=fopt[n];
52                 //int r=(int)(0.99*2.0);
53                 CUR[i].setChrom(r,n);
54             }
55         }
```

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133

```

double para_sum = 0.0;//積分値
double para_sum2 = 0.0;//点移動後の積分値
double para_M = 0.0;//制約値
int para_posi = 0;//正の点の個数
int para_neg = 0;//負の点の個数
double move = 0.0;//移動量
int para_posi_Z = 0;
int para_neg_Z = 0;

int Gen =1;

/*世代の更新*/
for(Gen=1;Gen<=Def.NoG;Gen++){
//System.out.println(Gen+" ");
Evo.act(CUR,NXT,phi,z,Gen);
    if(Gen%1000==0){
        /*1 周期平均を 0 にする操作 (2)*/
        para_posi=0;
        para_posi_Z=0;
        para_neg=0;
        para_neg_Z=0;
        para_sum =0.0;
        para_sum2 = 0.0;
        para_M = 0.0;
        for(int i=0; i<Def.N_PARAM; i++){
            para_sum += CUR[0].Para[i];//すべての点の値を足す (積分)
            if(CUR[0].Para[i]>=0)para_posi+= 1;
            if(CUR[0].Para[i]<0)para_neg+= 1;
            para_M += Math.pow(Math.abs(CUR[0].Para[i]), Def.Pnol);
        }
        para_M = para_M / (Def.N_PARAM);
        //シンプソン
        //double ss1 =0.0;
        //for(int i=2; i<=Def.N_PARAM-2; i+=2){
        //    ss1 += Math.pow(Math.abs(CUR[0].Para[i]),Def.Pnol);
        //}
        //double ss2 =0.0;
        //for(int i=1; i<=Def.N_PARAM-1; i+=2){
        //    ss2 += Math.pow(Math.abs(CUR[0].Para[i]),Def.Pnol);
        //}
        //para_M=2*Def.PI/(3.0*Def.N_PARAM)*(Math.pow(Math.abs(CUR[0].Para[0]),Def.Pnol)+Math.pow(Math.abs(CUR[0].Para[Def.N_PARAM]),Def.Pnol));
        //para_M/=2*Def.PI;//平均化

        para_M = Math.pow(para_M, (1.0/Def.Pnol));
        System.out.println("条件 : p="+Def.Pnol+", M="+Def.M+"で探索中");
        System.out.println("探索点の P ノルム="+para_M+" 正の点"+para_posi+" 負の点"+para_neg);

        /*if(para_sum != 0){/1 周期積分が 0 になるように移動を行う
            //move = para_sum/ Def.N_PARAM;//全ての点の移動量を均等に設定
            move=para_sum / (para_posi + para_neg);
            for(int i=0; i<Def.N_PARAM; i++){
                if(CUR[0].Para[i]>=Def.ZERO||CUR[0].Para[i]<=-Def.ZERO)CUR[0].Para[i] = CUR[0].Para[i] - move;
                if(i==0)CUR[0].Para[Def.N_PARAM] = CUR[0].Para[i];//theta=0 と theta=2pi の点は等しい
                para_sum2 += CUR[0].Para[i];//移動後のすべての点の値を足す (移動後の積分値)
            }
        }*/
        System.out.println("1 周期積分="+para_sum);

        para_M = 0.0;//初期化
        para_sum = 0.0;//初期化
    }
    /*結果表示*/
    System.out.println(Gen+" "+ CUR[0].R+" "+//BEST4 を表示する
        CUR[1].R+" "+CUR[2].R+" "+CUR[3].R);
    System.out.println("-----");
}
/*指定した世代毎に探索波形 data を出力*/
if(Gen%Def.ANSprint==0){
    Mf.data(phi, CUR[0].Para, Gen, CUR[0].R, CUR[0].shift);
    Mf.gnu(Gen, CUR[0].R);
}
}

/*終了時刻取得, 実行時間表示*/
long stoptime = System.currentTimeMillis();
System.out.println("Elapsed time is about "+ (stoptime - starttime)/60000+" min." );
System.out.println((stoptime - starttime)+" msec/10g.");

/*探索解の表示*/

```

```

134         double Lockrange=CUR[0].R;
135         System.out.printf("The Lockrange is %4.6f.\n",Lockrange);
136         System.out.printf("---->gnuplot_*****.txt\n");
137     }
138 }

```

- Ga_func.java

```

1  class Ga_func{//GA における関数
2      /*関数定義*/
3      Equation Eq=new Equation();//方程式を定義するクラス
4      /*個体をソートする関数*/
5      void quickSort(Ind[] IND){
6          /*予備の個体の構造体定義*/
7          Ind[] TMP=new Ind[1];//予備個体生成
8          TMP[0]=new Ind();
9          /*並べ替え*/
10         for(int i=Def.NoI-1;i>=0;i--){
11             for(int j=0;j<i;j++){
12                 if(IND[j].Evaluation<=IND[j+1].Evaluation){
13                     TMP[0]=IND[j];
14                     IND[j]=IND[j+1];
15                     IND[j+1]=TMP[0];
16                 }
17             }
18         }
19     }
20     /*評価値を計算する関数*/
21     void evaluate(Ind[] IND,double[] phi,double[][] z, int Gen){
22         bin2dec(IND);//遺伝子情報を実数パラメタに変換
23         /*各個体の評価値を導出 (ロックレンジの導出) */
24         double [] qq=new double[Def.N_PARAM+1];//周期外力 (正規化前)
25         double [] q=new double[Def.N_PARAM+1];//周期外力 (正規化後)
26         double INT=0;//積分の解
27         double min=0,max=0,lr=0;//ロックレンジの右端、左端、ロックレンジ
28         int min_number=0;
29         for(int p=0;p<=Def.N_PARAM;p++){
30             q[p]=0;
31             qq[p]=0;
32         }
33         int posi = 0;
34         double posi_a = 0;
35         int nega = 0;
36         double nega_a = 0;
37         double move = 0;
38         double move_a = 0;
39         double kakumove = 0;
40         int set1 = 0;//両方一緒 1 プラスの個体が多い 2 マイナスの個体が多い
41         double taishod = 0;
42         int taisho = 0;
43         int aa = 0;
44         double para_a = 0;
45         double kaku_temp = 0;
46         int dispset = 0;//動作確認用, 1 なら表示, 0 ならなし
47
48         double zyoge = 0.0;
49
50         double tmp0606 = 0.0;
51         int tmp0606i = 0;
52
53         for(int i=0;i<Def.NoI;i++){//畳み込み積分
54             for(int p=0;p<=Def.N_PARAM;p++){
55                 qq[p]=(IND[i].Para[p]);
56                 if(Def.F_ZERO&&p>=Def.N_PARAM/Def.Ntime&&p<Def.N_PARAM) qq[p]=0.0;
57             }
58         }
59
60         /*1 周期平均を 0 にする操作 (1)*/
61         if(Gen>=100){//100 世代終了後に点移動操作を開始
62             posi = 0;
63             nega = 0;
64             posi_a = 0;
65             nega_a = 0;
66             move_a = 0;
67             nega_a = 0;
68             para_a = 0;//動かすパラメタの合計値
69             kakumove = 0;
70             kaku_temp = 0;
71

```

```

72
73
74
75
76
77
78
79
80
81
82 //
83 //
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101 //
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
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
for(int r=0;r<Def.N_PARAM;r++){
    if(qq[r]>=0){
        posi += 1;//プラス点のカウンント
        posi_a += qq[r];//プラス点の値加算
    }
    if(qq[r]<0){
        nega += 1;//マイナス点のカウンント
        nega_a += Math.abs(qq[r]);//マイナス点の値加算
    }
}
if(posi>nega)set1 = 1;//プラス点の方が多い場合
if(posi<nega)set1 = 2;//マイナス点の方が多い場合
if(posi==nega)set1 = 0;//両方同じ数の場合
move = Math.abs(posi - nega)/2;//動かすべきパラメタの個数 (両方の差の半分)
if(dispsset == 1)System.out.println("-----"+Gen);
if(dispsset == 1)System.out.println("p:"+posi+",n:"+nega+",set="+set1+", move="+move);
if(dispsset == 1)System.out.println("p_a:"+posi_a+",n_a:"+nega_a);
double[] num=new double[(int)move];
//move_a = Math.abs((posi_a - Math.abs(nega_a)))/2;//動かす量 (全パラメタに対して)
//kakumove = move_a / move;//個々のパラメタに対して動かす量 (量/動かすべきパラメタ数
//プラスが多かった場合
if(set1==1){
    //if(posi_a - Math.abs(nega_a) < 0)break;
    ka:for(int c=0; c<move; c++){
        taishod = 0;//動かす対象の配列番号
        //taishod = 3;
        taishod =Math.random()*Def.N_PARAM;//動かす対象をランダムに決定
        taisho = (int)taishod;
        if(qq[taisho]>=0){//決定した動かす対象がプラスの値だったら動かす
            if(qq[taisho]>=Def.ZERO){
                if(c>0){
                    for(int e=0; e<c; e++){
                        if(num[e]==taisho){//さっき選んだやつと重複していないか?
                            if(dispsset == 1)System.out.println("s");
                            c=c-1;
                            continue ka;
                        }
                    }
                }
                num[c]=taisho;
                if(dispsset == 1)System.out.println("動かす番号"+num[c]);
            }else{
                c=c-1;//動かさなかったらカウントを戻す
                if(dispsset == 1)System.out.println("x");
                continue ka;
            }
        }
    }
    for(int c=0; c<move; c++){
        posi_a -= qq[(int)num[c]];
        nega_a += qq[(int)num[c]];
        qq[(int)num[c]] = -qq[(int)num[c]];
        if(dispsset == 1)System.out.println("正負移動のみ");
    }
}
//マイナスが多かった場合
if(set1==2){
    //if(Math.abs(nega_a) - posi_a < 0)break;
    if(dispsset == 1)System.out.println("move="+move);
    kb:for(int c=0; c<move; c++){
        taishod = 0;
        taishod = Math.random()*Def.N_PARAM;
        taisho = (int)taishod;
        if(qq[taisho]<0){
            if(qq[taisho]<=-Def.ZERO){
                if(c>0){
                    for(int e=0; e<c; e++){
                        if(num[e]==taisho){
                            c=c-1;
                            continue kb;
                        }
                    }
                }
                num[c]=taisho;
                if(dispsset == 1)System.out.println("動かす番号"+num[c]);
            }else{
                c=c-1;
                if(dispsset == 1)System.out.println("x");
                continue kb;
            }
        }
    }
}
for(int d=0; d<move; d++){

```

```

153         nega_a -= Math.abs(qq[(int)num[d]]);
154         posi_a += Math.abs(qq[(int)num[d]]);
155         qq[(int)num[d]] = -qq[(int)num[d]];
156         if(dispset == 1)System.out.println("正負移動のみ");
157     }
158 }
159 //////////////////////////////////////////////////////////////////20120128 追加
160 if(Math.abs(posi_a - nega_a) != 0){//上と下の差が 0.001 より大きい場合
161     if(posi_a > nega_a){//上の合計のほうが大きい場合
162         move = (posi_a - nega_a);
163         if(dispset == 1)System.out.println("全ての点を下へ"+move+"移動します");
164         for(int v=0; v<Def.N_PARAM; v++){
165             qq[v] = qq[v] - (Math.abs(qq[v]) * move/ (posi_a+nega_a));
166             if(v==0)qq[Def.N_PARAM] = qq[v];
167         }
168     }
169     if(posi_a < nega_a){//下の合計のほうが大きい場合
170         move = (nega_a - posi_a);
171         if(dispset == 1)System.out.println("全ての点を上へ"+move+"移動します");
172         for(int v=0; v<Def.N_PARAM; v++){
173             qq[v] = qq[v] + (Math.abs(qq[v]) * move/(posi_a+nega_a));
174             if(v==0)qq[Def.N_PARAM] = qq[v];
175         }
176     }
177 }
178 //if(Gen%1000==0&&(i==0||i==999||i==30)){
179 //    double S=0.0;
180 //    for(int v=0;v<Def.N_PARAM;v++) S+=qq[v];
181 //    System.out.println(i+":S="+S);
182 //}
183 }
184 //////////////////////////////////////////////////////////////////
185 q=Eq.func(q,qq);
186 //q=qq;
187 for(int p=0;p<=Def.N_PARAM;p++)IND[i].Para[p]=q[p];
188 min_number = 0;//相互作用関数が最小となる時のパラメタ番号
189 //位相シフトせずに理論解と同位相にするためのもの
190 min=max=0;
191 for(int t=0;t<=Def.N_PARAM;t++){//theta と psi を離散化して畳み込み積分
192 //    INT=Eq.simp(z,q,t)/(2*Def.PI);//積分 (シンプソン法)
193     INT=Eq.square(z,q,t)/(2*Def.PI);
194     if(INT>max) max=INT;//最大値の判定
195     if(INT<min){
196         min=INT;//最小値の判定
197         min_number = t;//この時のパラメタ番号も記録
198     }
199 }
200
201 lr=Math.abs(max-min);//ロックレンジの導出
202 IND[i].setEva(lr);//個体 i に評価値を代入
203 IND[i].setR(lr);//個体 i にロックレンジを代入
204 IND[i].setshift(min_number);//個体 i に理論解とフィッティングするための位相シフト量を記録
205 if(lr>tmp0606){
206     tmp0606 = lr;
207     tmp0606i = i;
208 }
209 lr=0;
210 }
211 if(Gen%1000==0){
212     double S=0.0;
213     for(int v=0;v<Def.N_PARAM;v++) S+=IND[tmp0606i].Para[v];
214     System.out.println(tmp0606i+":S(lrmax)="+S);
215 }
216 }
217 /*遺伝子情報をパラメタ値に変換する関数*/
218 void bin2dec(Ind[] IND){
219     int s=0,e=0;//各パラメタの遺伝子情報が記述された染色体始点、終点
220     int p=0;//パラメタ番号 (1.C1、2.P2)
221     int bit=Def.P_BIT-1;//指数
222     int sum=0;//2 進数から 10 進数に変換する際の和
223     double tmp=0.0;
224     double[][] dn=new double[Def.NoI][Def.N_PARAM+1];
225
226     for(int i=0;i<Def.NoI;i++){//dn 初期化
227         for(p=0;p<=Def.N_PARAM;p++)dn[i][p]=0;
228     }
229     for(int i=0;i<Def.NoI;i++){//各個体の dn を求める
230         s=0;
231         e=Def.P_BIT;
232         for(p=0;p<=Def.N_PARAM;p++){
233             for(int n=s;n<e;n++){

```

```

234         sum+=IND[i].Chromosome[n]*Math.pow(2,bit);
235         bit--;
236     }
237     bit=Def.P_BIT-1;
238     s+=Def.P_BIT;
239     e=s+Def.P_BIT;
240     dn[i][p]=(double)(sum/(Math.pow(2,Def.P_BIT)-1));
241     sum=0;
242 }
243 }
244 for(int i=0;i<Def.NoI;i++){//各個体のパラメタに代入
245     IND[i].setPara(0.0,0);
246     for(p=0;p<=Def.N_PARAM;p++){
247         tmp=(Def.UPPER-Def.LOWER)*dn[i][p]+Def.LOWER;
248         IND[i].setPara(tmp,p);
249     }
250     //IND[i].setPara(0.0,Def.N_PARAM);
251 }
252 }
253 /*GA用のルーレット作成*/
254 void makeRoulette(Ind[] IND){
255     double eva_sum=0;//評価値の和
256
257     for(int i=0;i<Def.NoI;i++) eva_sum+=IND[i].Evaluation;
258
259     IND[0].Roulette=IND[0].Evaluation;
260     for(int i=1;i<Def.NoI;i++)IND[i].Roulette=IND[i-1].Roulette+IND[i].Evaluation;
261     for(int i=0;i<Def.NoI;i++)IND[i].setRou(IND[i].Roulette/eva_sum);
262     IND[Def.NoI-1].setRou(1.0);
263 }
264 }

```

- Equation.java

```

1 class Equation{//方程式を記述するクラス
2     /*方程式*/
3     /***** prc メソッド *****/
4     double prc(double x){
5         Def def = new Def();//Defクラスの宣言
6         return def.a0/2.0+def.b1*Math.sin(x)+def.b2*Math.sin(2.0*x)+def.b3*Math.sin(3.0*x)+def.b4*Math.sin(4.0*x)+def.b5*Math.sin(5.0*x)
7             +def.a1*Math.cos(x)+def.a2*Math.cos(2.0*x)+def.a3*Math.cos(3.0*x)+def.a4*Math.cos(4.0*x)+def.a5*Math.cos(5.0*x);//
8     }
9
10    double[] func(double q[],double[] qq){//探索周期外力
11        double PP=0;//規格化前の周期外力のパワー (p=2 以外ではパワーとは呼ばない)
12        /*規格化前のパワーを計算 (シンプソン法)*/
13        double ss1 =0.0;
14        /*
15        for(int i=2; i<=Def.N_PARAM-2; i+=2){
16            ss1 += Math.pow(Math.abs(qq[i]),Def.Pnol);
17        }
18        double ss2 =0.0;
19        for(int i=1; i<=Def.N_PARAM-1; i+=2){
20            ss2 += Math.pow(Math.abs(qq[i]),Def.Pnol);
21        }
22        PP=2*Def.PI/(3.0*Def.N_PARAM)*(Math.pow(Math.abs(qq[0]),Def.Pnol)+Math.pow(Math.abs(qq[Def.N_PARAM]),Def.Pnol))+2.0*ss1+4.0*ss2);
23        PP/=2*Def.PI;//平均化
24        */
25        for(int i=0;i<Def.N_PARAM;i++){
26            PP+=Math.pow(Math.abs(qq[i]), Def.Pnol);
27        }
28        PP/=Def.N_PARAM;
29        double sahen = Math.pow(PP, 1/Def.Pnol);
30
31        for(int i=0; i<=Def.N_PARAM; i++){
32            q[i] = 0;
33        }
34
35        /*規格化*/
36        double normalize = 0.0;//規格化のために各周期外力 qq に対してかける係数
37        normalize = (Def.M/sahen);
38        for(int i=0; i<=Def.N_PARAM; i++){//探索点すべてを規格化
39            q[i] = normalize * qq[i];
40        }
41
42        return q;
43    }
44
45    double simp(double[][] z,double[] q,int t){//積分:シンプソン法
46        double ss1 =0.0;
47        for(int i=2; i<=Def.SAMPLE-2; i+=2){
48            ss1 += z[t][i]*q[i];
49        }

```

```

48         double ss2 =0.0;
49         for(int i=1; i<=Def.SAMPLE-1; i+=2){
50             ss2 += z[t][i]*q[i];
51         }
52         return 2*Def.PI/(3.0*Def.SAMPLE)*
53         (z[t][0]*q[0]+z[t][Def.SAMPLE]*q[Def.SAMPLE]+2.0*ss1+4.0*ss2);
54     }
55
56     double square(double[][] z, double[] q, int t){
57         double s=0.0;
58         for(int i=0; i<Def.SAMPLE;i++){
59             s+=z[t][i] * q[i];
60         }
61         return s * 2*Def.PI / Def.SAMPLE;
62     }
63 }

```

– Evolution.java

```

1  import java.io.*;
2  class Evolution{
3      double dice=0;//親選択のサイコロ
4      int PARENT1=0,PARENT2=0;//親 1、親 2 の個体
5      /*関数定義*/
6      Ga_func Gf=new Ga_func();//GA における関数
7
8      void act(Ind[] CUR,Ind[] NXT,double[] phi,double[][] z, int Gen){
9          Gf.evaluate(CUR,phi,z, Gen);//個体の評価
10         Gf.quickSort(CUR);//クイックソート
11
12         for(int i=0;i<Def.NoE;i++){//エリート戦略
13             for(int n=0;n<Def.LoC;n++){
14                 NXT[i].setChrom(CUR[i].Chromosome[n],n);
15             }
16         }
17
18         Gf.makeRoulette(CUR);//GA 用ルーレット作成
19
20         for(int i=Def.NoE;i<Def.NoI;i+=2){//エリート以外の次世代作成
21             dice=Math.random();//GA 用ルーレットを回す
22             for(int j=0;j<Def.NoI;j++){//親 1 選択
23                 if(dice>CUR[j].Roulette)PARENT1=j;
24                 else break;
25             }
26             dice=Math.random();//GA 用ルーレットを回す
27             for(int j=0;j<Def.NoI;j++){//親 2 選択
28                 if(dice>CUR[j].Roulette)PARENT2=j;
29                 else break;
30             }
31             if(Math.random()<Def.PROB_CROSSOVER){//交叉 (一様交叉) を行う
32                 for(int n=0;n<Def.LoC;n++){//交叉処理
33                     if(Math.random()<0.5){//遺伝子交換 : パターン 1
34                         NXT[i].setChrom(CUR[PARENT1].Chromosome[n],n);
35                         NXT[i+1].setChrom(CUR[PARENT2].Chromosome[n],n);
36                     }else{//遺伝子交換 : パターン 2
37                         NXT[i].setChrom(CUR[PARENT2].Chromosome[n],n);
38                         NXT[i+1].setChrom(CUR[PARENT1].Chromosome[n],n);
39                     }
40                 }
41             }else{//交叉しない
42                 for(int n=0;n<Def.LoC;n++){//染色体ごと交換
43                     NXT[i].setChrom(CUR[PARENT1].Chromosome[n],n);
44                     NXT[i+1].setChrom(CUR[PARENT2].Chromosome[n],n);
45                 }
46             }
47             if(Math.random()<Def.PROB_MUTATION){//突然変異 (遺伝子を 1 つだけ反転)
48                 int po=(int)(Math.random()*2.0);//突然変異する個体番号
49                 int n_op=(int)(Math.random()*Def.LoC);//反転する遺伝子番号
50                 NXT[i+po].Chromosome[n_op]=1-NXT[i+po].Chromosome[n_op];
51             }
52         }
53
54         for(int i=0;i<Def.NoI;i++){//次世代の染色体を現世代へ代入
55             for(int n=0;n<Def.LoC;n++){
56                 CUR[i].setChrom(NXT[i].Chromosome[n],n);
57             }
58         }
59     }
60     /*理論解を 2 進数に変換, 出力する関数*/
61     int[] initDec2bin() {
62         double[] foft = new double[Def.SAMPLE+1];//サンプリング数を揃えた理論解
63         int[] foftbin = new int[Def.LoC];//理論解を 2 進数に変換した染色体を全て格納

```

```

64     try {
65         BufferedReader br0 = new BufferedReader(new FileReader(Def.ffopt));
66         String s;
67         int line = 0;
68         while((s = br0.readLine())!=null){
69             line++;
70         }
71         br0.close();
72         int div = (line-1)/Def.SAMPLE;//データ点を取得する間隔
73         if((line-1)%Def.SAMPLE!=0) System.out.println("Caution: the number of the file lines is indivisible.");
74         br0 = new BufferedReader(new FileReader(Def.ffopt));
75         line = 0;
76         String[] tmp;
77         while((s = br0.readLine())!=null){
78             if((line%div)==0){//理論解を間隔毎に取得
79                 tmp = s.split(" ");
80                 fopt[line/div]=Double.parseDouble(tmp[1]);
81             }
82             line++;
83         }
84         br0.close();
85     } catch (IOException e) {//ファイルがない、もしくはなにかしらのエラーとなった場合は乱数を出力
86         for(int i=0;i<Def.SAMPLE;i++) fopt[i]=Math.random();
87         fopt[Def.SAMPLE]=fopt[0];
88         System.out.println(Def.ffopt+" is not found.");
89     }
90     double max=1.0;//理論解の絶対値の最大値 (1 以上なら 1 になるように規格化)
91     for(int i=0;i<Def.SAMPLE;i++){
92         if(Math.abs(fopt[i]) > max) max = Math.abs(fopt[i]);
93     }
94     double fstd;
95     int sum;
96     String bin;
97     for(int i=0;i<=Def.SAMPLE;i++){
98         fstd = (fopt[i]/max - Def.LOWER) / (Def.UPPER - Def.LOWER);//0-1 に規格化
99         sum = (int)(fstd * (Math.pow(2,Def.P_BIT)-1));//0-2(P_BIT-1) に規格化
100         bin = String.format("%"+Def.P_BIT+"s", Integer.toBinaryString(sum)).replaceAll(" ", "0");//2 進数に変換(桁
101         が P_BIT になるよう 0 埋め)
102         for(int j=0;j<Def.P_BIT;j++) foptbin[i*Def.P_BIT+j] = Integer.parseInt("" + bin.charAt(j));//染色体を 1 文
103         字ずつ格納
104     }
105     return foptbin;
106 }

```

オイラー法で微分方程式を解き周波数特性を取得するプログラム

— freq.java

```

1 //HH 方程式の位相の時間変化 (ノイズあり)
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.text.*;
6 import java.io.*;
7 import java.util.*;
8
9 class main{
10     public static void main(String[] args){
11
12         for(int k=0;k<=0;k++){
13             Def.amp=(double)k*(Def.amp_max-Def.amp_min)/1.0+Def.amp_min;
14             System.out.println("amp"+Def.amp);
15             new main();
16             long starttime = System.currentTimeMillis();
17             Date date = new Date();
18             DateFormat df = new SimpleDateFormat("yy-MM-dd-HH-mm-ss");
19             Equation Eq=new Equation();
20
21             PrintWriter pw_time = null;//gnuplot(位相差)
22             String filename_time;
23             filename_time = "time-amp"+Def.amp+".txt";
24             try {
25                 pw_time = new PrintWriter(filename_time);
26             } catch (Exception e) {
27             }

```



```

28
29
30
31     PrintWriter pw = null;//sin の周波数ごとの位相差
32         String filename;
33     PrintWriter pw1 = null;//sin の周波数ごとの周波数差
34         String filename1;
35     PrintWriter pw2 = null;//周波数特性
36         String filename2;
37         filename2 = "HH"+Def.tei;
38         filename2 = filename2+"delta_omega";
39         filename2 = filename2+"-amp"+Def.amp;
40         filename2 = filename2 + "-date_time"+df.format(date);
41         filename2 = filename2 + ".txt";
42     try {
43         pw2 = new PrintWriter(filename2);
44     } catch (Exception e) {
45     }
46     PrintWriter pw3 = null;//gnuplot(周波数特性)
47         String filename3;
48         filename3 = "amp"+Def.amp;
49         filename3 = /* filename3+*/"gnuplot_omega.txt";
50     try {
51         pw3 = new PrintWriter(filename3);
52     } catch (Exception e) {
53     }
54     PrintWriter pw4 = null;//gnuplot(位相差)
55         String filename4;
56         filename4 = "amp"+Def.amp;
57         filename4 = /*filename4+*/"gnuplot_phase_difference.txt";
58     try {
59         pw4 = new PrintWriter(filename4);
60     } catch (Exception e) {
61     }
62
63     double xi;//ノイズ
64
65     double PI=Def.PI;
66     double Time=0.0;//グローバル時間
67     double T;//振動子の周期
68     double h_Time=Def.h_Time;
69     double sh_Time=Def.sh_Time;
70
71     double sine_omega;//注入正弦波の角周波数
72     double b_max_sine_omega=Def.b_max_sine_omega;//角周波数の最大値
73     double b_min_sine_omega=Def.b_min_sine_omega;//角周波数の最小値
74     double b_h_sine_omega=Def.b_h_sine_omega;//角周波数の刻み
75
76     double V,next_V,m,next_m,h,next_h,n,next_n;//HHのパラメータ
77     double[] para= new double[4];//next_*を1つの配列に(複数returnしたいから)
78     double[] para_ex=new double[4];//HHのパラメータを継承
79
80     double noise;//ノイズ
81
82     double psi;//振動子のmがピークをとる時間
83     double psi_ext;//sineの正方向へ0クロスした時間
84     double phase_difference=0.0;//振動子とsineの位相差
85     double sine;
86     double past_sine;
87     double amp=Def.amp;
88
89     double OMEGA_HH=Def.OMEGA_HH;
90     double past_diff_m=0.0;//mの差(old)
91     double diff_m=0.0;//mの差(new)
92     double aveTime_0=0.0;//取り始める時間
93     double aveTime_1=0.0;//取り終える時間
94     double ave_period=0.0;//平均する周期数
95     double past_peak_m=0.0,peak_m=0.0;//mがピークをとる時間
96     double period=0.0,frequency=0.0;//振動子の周期,角周波数
97     double sum_p=0.0,number_p=0.0;//周期の和,周期数
98     double ave_frequency=0.0;//振動子の周波数の平均
99     double delta_omega=0.0;//周波数差
100     double phase_diff_sum=0.0;//位相差の和
101     double ave_phase_diff=0.0;//位相差の周期平均
102     double tei=Def.tei;//連倍数
103 // System.out.println(123/1.);
104
105 //sin の周波数=HH 振動子の周波数にして回す
106

```

```

107     sine_omega=Def.OMEGA_HH;
108     filename = "HH"+tei;
109     filename = filename+"amp"+Def.amp;
110     filename = filename+"phase_difference";
111     filename = filename+"sin_omega"+sine_omega;
112     filename = filename+"-amp"+Def.amp;
113     filename = filename+"-date_time"+df.format(date)+".txt";
114     try {
115         pw = new PrintWriter(filename);
116     } catch (Exception e) {
117     }
118     filename1 = "HH"+tei;
119     filename1 = filename1+"amp"+Def.amp;
120     filename1 = filename1+"delta_omega";
121     filename1 = filename1 + "sin_omega"+sine_omega;
122     filename1 = filename1+"-amp"+Def.amp;
123     filename1 = filename1 + "-date_time"+df.format(date)+".txt";
124     try {
125         pw1 = new PrintWriter(filename1);
126     } catch (Exception e) {
127     }
128     pw.println("#Time phase_difference");
129     pw1.println("#Time period frequency sine_omega delta_omega");
130     pw2.println("delta_omega frequency ave_period ave_frequency phase_difference ave_phase_diff");
131     System.out.print("sin_omega="+sine_omega);
132 // while(sine_omega<=b_max_sine_omega){ //sine の角周波数を b_min_sine_omega から b_max_sine_omega まで変化させる
133     //HHを初期化
134     V=Def.V0;
135     m=Def.m0;
136     h=Def.h0;
137     n=Def.n0;
138     for(int i=0;i<=3;i++){
139         para[i]=0.0;
140     }
141     psi=0.0;
142     psi_ext=0.0;
143     sine=0.0;
144     past_sine=0.0;
145
146
147     Time=Def.T0;//シミュレーションの開始時間を設定
148
149     while(Time<=Def.SINE_IN){
150         xi=Eq.noise();//ノイズ生成
151
152
153
154     // オイラー法
155     next_V=V+Eq.func_V(V,m,h,n)*h_Time+xi*sh_Time;
156     next_m=m+Eq.func_m(V,m)*h_Time;
157     next_h=h+Eq.func_h(V,h)*h_Time;
158     next_n=n+Eq.func_n(V,n)*h_Time;
159
160     //パラメータを更新
161     V=next_V;
162     m=next_m;
163     h=next_h;
164     n=next_n;
165
166     Time=Time+h_Time;//時間を更新
167     }
168
169
170
171     //注入正弦波の基準の位相まで待つ
172
173     while(psi_ext==0.0){
174
175         xi=Eq.noise();
176         past_sine=sine;//old
177         sine=Eq.sine(sine_omega,Time,amp);//new
178
179         if(past_sine<=0.0&&sine*past_sine<0.0){
180
181             psi_ext=Time;
182
183         }
184
185
186     //オイラー法
187     next_V=V+Eq.func_V(V,m,h,n)*h_Time+xi*sh_Time;

```

```

188         next_m=m+Eq.func_m(V,m)*h_Time;
189         next_h=h+Eq.func_h(V,h)*h_Time;
190         next_n=n+Eq.func_n(V,n)*h_Time;
191
192         //パラメータを更新
193         V=next_V;
194         m=next_m;
195         h=next_h;
196         n=next_n;
197
198
199
200         Time=Time+h_Time;//時間を更新
201
202     }
203
204     //m が極大となる点を位相 0 とし数周期を平均
205
206     //念のため色々初期化
207     past_diff_m=0.0;//m の差 (old)
208     diff_m=0.0;//m の差 (new)
209     aveTime_0=0.0;//取り始める時間
210     aveTime_1=0.0;//取り終える時間
211     ave_period=0.0;//平均する周期数
212     past_peak_m=0.0;
213     peak_m=0.0;//m がピークをとる時間
214     period=0.0;
215     frequency=0.0;//振動子の周期, 角周波数
216     sum_p=0.0;
217     number_p=0.0;//周期の和, 周期数
218     ave_frequency=0.0;
219     delta_omega=0.0;
220     phase_diff_sum=0.0;
221     ave_phase_diff=0.0;
222     //sin 波が入った後
223     while(Time<=Def.REC_T){
224
225         xi=Eq.noise(); //ノイズ生成
226         sine=Eq.sine(sine_omega,Time,amp);//new
227         if(past_sine<=0.0&&sine*past_sine<0.0){
228             psi_ext=Time;
229         }
230         past_sine=sine;
231         past_diff_m=diff_m;
232         //オイラー (注入あり)
233         next_V=V+Eq.func_Vs(V,m,h,n,sine)*h_Time+xi*sh_Time;
234         next_m=m+Eq.func_m(V,m)*h_Time;
235         next_h=h+Eq.func_h(V,h)*h_Time;
236         next_n=n+Eq.func_n(V,n)*h_Time;
237
238         diff_m=next_m-m;//m の差を更新
239
240         if(m>0.9){
241             if(diff_m<0.0 && past_diff_m>=0.0){
242                 psi=Time;
243                 phase_difference=psi-psi_ext;
244                 pw.println(Time+" "+phase_difference);
245                 //m がピークをとる時間を更新
246                 past_peak_m=peak_m;
247                 peak_m=Time;
248             }
249         }
250     }
251
252     //パラメータを更新
253     V=next_V;
254     m=next_m;
255     h=next_h;
256     n=next_n;
257
258     System.out.println(V+" "+m+" "+h+" "+n);
259
260     Time=Time+h_Time;
261 }
262 //
263
264
265
266
267
268 //最後平均するまで

```

```

269         while(Time<=Def.END_T){
270
271             xi=Eq.noise(); //ノイズ生成
272             past_sine=sine;
273             sine=Eq.sine(sine_omega,Time,amp);//new
274
275             if(past_sine<=0.0&&amine*past_sine<0.0){
276
277                 psi_ext=Time;
278
279             }
280
281             past_diff_m=diff_m;
282             //オイラー (注入あり)
283             next_V=V+Eq.func_Vs(V,m,h,n,sine)*h_Time+xi*sh_Time;
284             next_m=m+Eq.func_m(V,m)*h_Time;
285             next_h=h+Eq.func_h(V,h)*h_Time;
286             next_n=n+Eq.func_n(V,n)*h_Time;
287
288             diff_m=next_m-m;
289
290             if(m>0.9){
291
292                 if(diff_m<0.0 && past_diff_m>=0.0){ //ピークを検出
293
294                     psi=Time;
295                     phase_difference=psi-psi_ext;//位相差
296                     phase_diff_sum=phase_diff_sum+phase_difference;//位相差の和
297
298                     pw.println(Time+" "+phase_difference);
299                     //m がピークをとる時間を検出
300                     past_peak_m=peak_m;
301                     peak_m=Time;
302
303                     period=peak_m-past_peak_m;//周期
304                     frequency=2.0*PI/period;//周波数
305                     delta_omega=(sine_omega/tei-OMEGA_HH);
306
307                     pw1.println(Time+" "+period+" "+frequency+" "+sine_omega+" "+delta_omega);
308
309                     sum_p=sum_p+period;//振動子の周期の和
310                     number_p=number_p+1.0;//振動子の周期数
311
312                 }
313
314             }
315
316             //パラメータを更新
317             V=next_V;
318             m=next_m;
319             h=next_h;
320             n=next_n;
321
322             Time=Time+h_Time;
323
324         }
325
326         // pw.close();
327         // pw1.close();
328         ave_phase_diff=(phase_diff_sum/number_p);
329         ave_period=(sum_p/number_p);
330         ave_frequency=2.0*PI/ave_period;
331
332         System.out.println(":done");
333         pw.close();
334         pw1.close();
335         pw2.println(delta_omega+" "+sine_omega+" "+ave_period+" "+ave_frequency+" "+phase_difference+" "+ave_phase_diff);
336         // sine_omega=sine_omega+b_h_sine_omega;//sine 波の周波数を更新
337         // }
338
339         //////////////////////////////////////
340
341         sine_omega=b_min_sine_omega;
342         while(sine_omega<=b_max_sine_omega){ //sine の角周波数を b_min_sine_omega から b_max_sine_omega まで変化させる
343
344             filename = "HH"+tei;
345             filename = filename+"amp"+Def.amp;
346             filename = filename+"phase_difference";

```

```

351 filename = filename+"sin_omega"+sine_omega;
352 filename = filename+"-amp"+Def.amp;
353 filename = filename+"-date_time"+df.format(date)+".txt";
354 try {
355     pw = new PrintWriter(filename);
356 } catch (Exception e) {
357 }
358 filename1 = "HH"+tei;
359 filename1 = filename1+"amp"+Def.amp;
360 filename1 = filename1+"delta_omega";
361 filename1 = filename1 + "sine_omega"+sine_omega;
362 filename1 = filename1+"-amp"+Def.amp;
363 filename1 = filename1 + "-date_time"+df.format(date)+".txt";
364 try {
365     pw1 = new PrintWriter(filename1);
366 } catch (Exception e) {
367 }
368 pw.println("#Time phase_difference");
369 pw1.println("#Time period frequency sine_omega delta_omega");
370 System.out.print("sin_omega="+sine_omega);
371 //HH を初期化
372 V=Def.V0;
373 m=Def.m0;
374 h=Def.h0;
375 n=Def.n0;
376 for(int i=0;i<=3;i++){
377     para[i]=0.0;
378 }
379 psi=0.0;
380 psi_ext=0.0;
381 sine=0.0;
382 past_sine=0.0;
383
384 Time=Def.T0;//シミュレーションの開始時間を設定
385
386 while(Time<=Def.SINE_IN){
387     xi=Eq.noise();//ノイズ生成
388
389     //オイラー法
390     next_V=V+Eq.func_V(V,m,h,n)*h_Time+xi*sh_Time;
391     next_m=m+Eq.func_m(V,m)*h_Time;
392     next_h=h+Eq.func_h(V,h)*h_Time;
393     next_n=n+Eq.func_n(V,n)*h_Time;
394
395     //パラメータを更新
396     V=next_V;
397     m=next_m;
398     h=next_h;
399     n=next_n;
400
401     Time=Time+h_Time;//時間を更新
402 }
403
404 //注入正弦波の基準の位相まで待つ
405 while(psi_ext==0.0){
406
407     xi=Eq.noise();
408     past_sine=sine;//old
409     sine=Eq.sine(sine_omega,Time,amp);//new
410
411     if(past_sine<=0.0&&sine*past_sine<0.0){
412
413         psi_ext=Time;
414
415     }
416
417     //オイラー法
418     next_V=V+Eq.func_V(V,m,h,n)*h_Time+xi*sh_Time;
419     next_m=m+Eq.func_m(V,m)*h_Time;
420     next_h=h+Eq.func_h(V,h)*h_Time;
421     next_n=n+Eq.func_n(V,n)*h_Time;
422
423     //パラメータを更新
424     V=next_V;
425     m=next_m;
426     h=next_h;
427     n=next_n;
428
429
430

```

```

431         Time=Time+h_Time;//時間を更新
432
433     }
434
435     //m が極大となる点を位相 0 とし数周期を平均
436
437     //念のため色々初期化
438     past_diff_m=0.0;//m の差 (old)
439     diff_m=0.0;//m の差 (new)
440     aveTime_0=0.0;//取り始める時間
441     aveTime_1=0.0;//取り終える時間
442     ave_period=0.0;//平均する周期数
443     past_peak_m=0.0;//m がピークをとる時間 (old)
444     peak_m=0.0;//m がピークをとる時間 (new)
445     period=0.0;//周期 (peak_m マイナス past_peak_m)
446     frequency=0.0;//振動子の周期、角周波数
447     sum_p=0.0; //周期の和
448     number_p=0.0;//周期数
449     ave_frequency=0.0;//平均周波数
450     delta_omega=0.0;//(delta omega=omega/n-omega0)
451     phase_diff_sum=0.0;//位相差の和
452     ave_phase_diff=0.0;//位相差の平均
453     //sin 波が入った後
454     while(Time<=Def.REC_T){
455         //
456         while(Time<=(Def.REC_T-Def.SINE_IN)){
457             xi=Eq.noise();//ノイズ生成
458             past_sine=sine;
459             sine=Eq.sine(sine_omega,Time,amp);//new
460
461             if(past_sine<=0.0&& sine*past_sine<0.0){
462                 psi_ext=Time;
463             }
464
465             past_diff_m=diff_m;
466             //オイラー (注入あり)
467             next_V=V+Eq.func_Vs(V,m,h,n,sine)*h_Time+xi*sh_Time;
468             next_m=m+Eq.func_m(V,m)*h_Time;
469             next_h=h+Eq.func_h(V,h)*h_Time;
470             next_n=n+Eq.func_n(V,n)*h_Time;
471             diff_m=next_m-m;//m の差を更新
472
473             if(m>0.9){
474                 if(diff_m<0.0 && past_diff_m>=0.0){
475                     psi=Time;
476                     phase_difference=psi-psi_ext;//位相差
477                     pw.println(Time+" "+phase_difference);
478                     //m がピークをとる時間を更新
479                     past_peak_m=peak_m;
480                     peak_m=Time;
481                 }
482             }
483
484             //パラメータを更新
485             V=next_V;
486             m=next_m;
487             h=next_h;
488             n=next_n;
489
490             Time=Time+h_Time;
491         }
492     }
493     //最後平均するまで
494     while(Time<=(Def.END_T)){
495         //
496         while(Time<=(Def.END_T-Def.REC_T)){
497             xi=Eq.noise();
498

```

```

514         past_sine=sine;
515         sine=Eq.sine(sine_omega,Time,amp);//new
516
517
518
519         if(past_sine<=0.0&&sine*past_sine<0.0){
520
521             psi_ext=Time;
522
523         }
524
525
526
527
528         past_diff_m=diff_m;
529         //オーバー (注入あり)
530         next_V=V+Eq.func_Vs(V,m,h,n,sine)*h_Time+xi*sh_Time;
531         next_m=Eq.func_m(V,m)*h_Time;
532         next_h=Eq.func_h(V,h)*h_Time;
533         next_n=Eq.func_n(V,n)*h_Time;
534         diff_m=next_m-m;
535
536
537
538         if(m>0.9){
539             if(diff_m<0.0 && past_diff_m>=0.0){
540
541                 psi=Time;
542                 phase_difference=psi-psi_ext;//位相差
543                 phase_diff_sum=phase_diff_sum+phase_difference;//位相差の和
544                 pw.println(Time+" "+phase_difference);
545                 //m がピークをとる時間を検出
546                 past_peak_m=peak_m;
547                 peak_m=Time;
548
549                 period=peak_m-past_peak_m;//周期
550                 frequency=2.0*PI/period;//周波数
551                 delta_omega=(sine_omega/tei-OMEGA_HH);
552                 System.out.println(delta_omega);
553                 pw1.println(Time+" "+period+" "+frequency+" "+sine_omega+" "+delta_omega);
554
555                 sum_p=sum_p+period;//振動子の周期の和
556                 number_p=number_p+1.0;//振動子の周期数
557
558             }
559         }
560
561     }
562
563     //パラメータを更新
564     V=next_V;
565     m=next_m;
566     h=next_h;
567     n=next_n;
568
569
570
571
572     Time=Time+h_Time;
573
574
575
576     }
577     ave_phase_diff=(phase_diff_sum/number_p);
578     ave_period=(sum_p/number_p);
579     ave_frequency=2.0*PI/ave_period;
580
581     System.out.println(":done");
582
583     pw2.println(delta_omega+" "+sine_omega+" "+ave_period+" "+ave_frequency+" "+phase_difference+" "+ave_phase_diff);
584
585     sine_omega=sine_omega+b_h_sine_omega;//sine 波の周波数を更新
586 }
587 pw.close();
588 pw1.close();
589 pw2.close();
590
591 //周波数特性のグラフ
592 pw3.println("set term postscript eps enhanced font 'Times,20'");//書き出しファイル形式を eps に指定
593 pw3.println("set output 'delta_omega-sine_omega'+sine_omega+'.eps'");//書き出しファイル名を指定
594 pw3.println("set nokey");//凡例を非表示に
595 pw3.println("#set xrange[:]");//x 軸のプロット範囲
596 pw3.println("set xlabel '{delta-omega}'");//x 軸のラベル

```

```

597     pw3.println("set ylabel '{frequency}'"); //y 軸のラベル
598     pw3.println("set size 1.0,0.6"); //作成するグラフの縦横比 (多分)
599     pw3.println("plot '+' '+'filename2+' '+' using 1:4 w p lc 1"); //別ファイルにあるデータをプロット
600
601     pw3.flush();
602     pw3.close();
603
604     //位相差のグラフ ()
605     pw4.println("set term postscript eps enhanced font 'Times,20'"); //書き出しファイル形式を eps に指定
606     pw4.println("set output 'phase_deference.eps'"); //書き出しファイル名を指定
607     pw4.println("set nokey"); //凡例を非表示に
608     pw4.println("#set xrange[:]"); //x 軸のプロット範囲
609     pw4.println("set xlabel '{Time}'"); //x 軸のラベル
610     pw4.println("set ylabel '{phase-difference}'"); //y 軸のラベル
611     pw4.println("set size 1.0,0.6"); //作成するグラフの縦横比 (多分)
612     pw4.println("plot '+' .txt+' '+' using 1:2 w p lc 1"); //別ファイルにあるデータをプロット
613
614     pw4.flush();
615     pw4.close();
616
617
618     // System.out.println(PI);
619
620
621     long stoptime = System.currentTimeMillis();
622     System.out.println();
623     System.out.println("Elapsed time is about "+ (stoptime - starttime)/60000+" min." );
624     pw_time.println("Elapsed time is about "+ (stoptime - starttime)/60000+" min.");
625     pw_time.close();
626     }
627     }
628 }

```

– Equation.java

```

1  import java.util.*;
2
3
4  class Equation{
5
6      Random Rd=new Random();
7      double h_Time=Def.h_Time;
8      double sh_Time=Def.sh_Time;
9      double C=Def.C;
10     double I=Def.I;
11
12     double noise(){// ランダムノイズ (ボックスミュラー法)
13         double dev=Def.noise_range;
14         double v1,v2, fac,rsq;
15         double ran_1,ran_2;
16         long idum=Def.idum;
17         do{
18
19             v1=2.0*ran1()-1.0;
20
21             v2=2.0*ran1()-1.0;
22             rsq=v1*v1+v2*v2;
23             }while(rsq>=1.0||rsq==0);
24
25             fac=Math.sqrt(-2.0*Math.log(rsq)/rsq);
26
27             if(Def.i==0){
28                 Def.i=1;
29
30                 return dev*v1*Math.sqrt(-2.0*Math.log(rsq)/rsq);
31             }else{
32                 Def.i=0;
33
34                 return dev*v2*Math.sqrt(-2.0*Math.log(rsq)/rsq);
35             }
36     }
37     double ran1(){
38         long IA=16807;
39         long IM=2147483647;
40         double AM=1.0/IM;
41         long IQ=127773;
42         long IR=2836;
43         int NTAB=32;
44         double NDIIV=67108864;
45         double ESP=1.200000e-07;
46         double RNMIX=1.0-ESP;
47         int j;

```



```

47     long k;
48     long idum=Def.idum;
49     long[] iv=Def.iv;
50
51     double temp;
52
53
54     if(idum<=0||Def.iy==0){
55
56         if((-idum)<1){
57             idum=1;
58         }else{
59             idum=(-idum);
60         }
61         for(j=NTAB+7;j>=0;j--){
62             k=idum/IQ;
63             idum=IA*(idum-k*IQ)-IR*k;
64             if(idum<0)idum=idum+IM;
65             if(j<NTAB){
66                 iv[j]=idum;
67
68             }
69         }
70         Def.iy=iv[0];
71     }
72     k=idum/IQ;
73     idum=IA*(idum-k*IQ)-IR*k;
74     if(idum<0)idum=idum+IM;
75     j=(int)Def.iy/(int)NDIV;
76     Def.iy=iv[j];
77     iv[j]=idum;
78     Def.idum=idum;
79     Def.iv=iv;
80     temp=AM*Def.iy;
81     if((temp)>RNMx)return RNMx;
82     else return temp;
83 }
84
85 double sine(double sine_omega,double Time,double amp){//注入正弦波
86
87     return amp*Math.sin(Time*sine_omega);
88
89 }
90
91
92
93
94
95
96 /* H-H 方程式 */
97 // 変数 V の微分方程式 (正弦波の注入前)
98 double func_V( double V, double m, double h, double n){
99
100     return ( I - 120.*h*m*m*m*( V-50.0 ) - 36.*n*n*n*n*( V+77.0 ) - 0.3*( V+54.4 ) ) /C;
101 }
102 // 変数 V の微分方程式 (正弦波の注入後)
103 double func_Vs( double V, double m, double h, double n, double sine ){
104
105     return ( I + sine - 120.0*h*m*m*m*( V-50.0 ) - 36.0*n*n*n*n*( V+77.0 ) - 0.3*( V+54.4 ) ) /C;
106 }
107
108 }
109
110 // 変数 m の微分方程式
111 double func_m( double V, double m ){
112
113     return ( 0.1*( V+40.0 ) / ( 1.-Math.exp( -( V+40.0 ) /10.0 ) ) )*( 1.-m ) - ( 4.0*Math.exp( -( V+65.0 ) /18.0 ) )*m;
114 }
115
116 }
117
118 // 変数 h の微分方程式
119 double func_h( double V, double h ){
120
121     return ( 0.07*Math.exp( -( V+65.0 ) /20.0 ) )*( 1.0-h ) - ( 1.0 / ( 1.0+Math.exp( -( V+35.0 ) /10.0 ) ) ) *h;
122 }
123
124 }
125
126 // 変数 n の微分方程式
127 double func_n( double V, double n ){
128
129     return ( 0.01*( V+55.0 ) / ( 1.-Math.exp( -( V+55.0 ) /10.0 ) ) )*( 1.0-n ) - ( 0.125*Math.exp( -( V+65.0 ) /80.0 ) ) *n;
130 }

```

```

131     }
132
133     }
134
135 }

```

位相差の分布関数を取得する Fokker-Planck 方程式を解くプログラム

– Pdf_fpe.java

```

1  import java.util.*;
2  class main{
3      public static void main(String[] args){
4          new main();// 実行
5          /* インスタンス化 */
6          Equation Eq = new Equation();// 方程式を定義するクラス
7          Makefile Mf=new Makefile();// ファイルを作成するクラス
8          /* 変数定義 */
9          int Smp=Def.SAMPLE;// FPE の空間刻み数
10         double [] phi=new double[Smp+1];// 位相
11         double [][] Z=new double[Smp+1][Smp+1];// PRC
12         double [] f=new double[Smp+1];// 注入信号
13         double [] g=new double[Smp+1];// Gamma
14         double [][] pdf = new double[2][Smp+1];// 確率分布関数
15         /* サンプリング */
16         for(int i=0;i<=Smp;i++){
17             phi[i]=(2*Def.PI/Smp)*i;
18             f[i]=Eq.func(phi[i]);
19         }
20         for(int i=0;i<=Smp;i++){
21             for(int j=0;j<=Smp;j++)Z[i][j]=Eq.prc(phi[i]+phi[j]);
22         }
23         for(int i=0;i<=Smp;i++)g[i]=Eq.simp(Z,f,i)/(2.0*Def.PI);
24
25         /******
26         *****
27         PROBABILITY DISTRIBUTION*****
28
29         /* 初期分布の生成 (初期分布は 1/2pi で均一) */
30         for(int i=0;i<=Def.SAMPLE;i++) {
31             pdf[0][i]=1/(2*Def.PI);
32             pdf[1][i]=0.0;
33         }
34         /* 時間発展 */
35         for(int t=0;t<=(int)(Def.END/Def.DT);t++){
36             /* FP 方程式の計算 */
37             for(int i=0;i<=Def.SAMPLE;i++)pdf[1][i] = Eq.fpe(pdf[i],g);
38             /* 確率密度の更新 */
39             for(int i=0;i<=Def.SAMPLE;i++)pdf[0][i] = pdf[1][i];
40             /* 時刻の表示 */
41             if((t*Def.DT%1)==0)System.out.print("\r@time:"+int)(t*Def.DT));
42         }
43         System.out.println(" ");
44         /* ファイル作成 */
45         Mf.data(phi,pdf);// 位相確率分布データファイルの作成
46         Mf.gnu();// gnuplot 用ファイルの作成
47     }
48 }

```

– Equation.java

```

1  class Equation{// 方程式を記述するクラス
2      /* 変数定義 */
3      int Smp=Def.SAMPLE;// FPE の空間刻み数
4      double Eps=Def.EPS;// 微小パラメタ
5      double Del=Def.DELTA;// 離調周波数
6      double Cour=Def.COUR;// 定数
7      double Diff=Def.DIFF;// 定数
8
9      /*GA の*/
10     double c1=3.252462512726967;
11     double c2=1.2451801922501846;
12     double c3=2.8055327287582257;
13     double p2=1.5723318070752434;
14     double p3=3.144663614150487;

```

```

15      /* 関数定義 */
16      double fpe(double pdf[][], int i, double g[]){// FP 方程式
17          int p1=0;
18          int p2=0;
19          double [] v = new double[4];
20
21          p1=(i-1+Smp)%Smp;
22          p2=(i+1)%Smp;
23
24          v[0] = (Del/(Eps*Eps)+g[i])*(pdf[0][p2]-pdf[0][p1]);
25          v[1] = pdf[0][i]*(g[p2]-g[p1]);
26          v[2] = Cour*(v[0] + v[1]);
27          v[3] = Diff*(pdf[0][p2]-2*pdf[0][i]+pdf[0][p1])*(1.0*1.0);
28
29          return pdf[0][i] + v[2] + v[3];
30      }
31      double prc(double x){// PRC
32          return 2.0*(Math.sin(x)+0.2*Math.sin(2*x)+0.3*Math.sin(3*x));
33      }
34      double func(double x){// 注入信号
35          double shift=1.602212253307948;
36          // return c1*Math.sin(x-shift)+c2*Math.sin(2*(x-shift)+p2)+c3*Math.sin(3*(x-shift)+p3);
37          return -3.186264940*Math.cos(x)-1.274505976*Math.cos(2*x)-2.867638446*Math.cos(3*x);
38          // return -4.428074428*Math.cos(x)-0.4428074428*Math.cos(2.0*x)-0.4428074428*Math.cos(3.0*x);
39      }
40      double simp(double[] z,double[] f,int t){//Gamma: シンプソン法
41          double[] S=new double[4];
42          for(int i=2; i<=Smp-2; i+=2){
43              S[0]+=z[t][i]*f[i];
44          }
45          for(int i=1; i<=Smp-1; i+=2){
46              S[1]+=z[t][i]*f[i];
47          }
48          S[2]=z[t][0]*f[0];
49          S[3]=z[t][Smp]*f[Smp];
50          return 2*Def.PI/(3.0*Smp)*(S[2]+S[3]+2.0*S[0]+4.0*S[1]);
51      }
52  }

```