

—修士学位論文—

$L_1$  正則化を考慮した周波数応答に基づく  
Gain-Scheduled 制御器設計に関する研究

A Study on Frequency-Response-Based  
Gain-Scheduled Controller Design in Consideration  
of  $L_1$  regularization

令和5年度

三重大学大学院 工学研究科  
博士前期課程 電気電子工学専攻  
水谷 力也



# 目次

第 1 章	緒言	1
1.1	研究背景と目的 . . . . .	1
1.2	論文の構成 . . . . .	3
第 2 章	$L_1$ 正則化	4
2.1	回帰分析 . . . . .	4
2.2	$L_1$ 正則化 . . . . .	8
2.3	数値シミュレーション . . . . .	10
第 3 章	Gain-Scheduled 制御	17
3.1	Gain-Scheduled 制御の概要 . . . . .	17
3.2	設計問題 . . . . .	18
第 4 章	ベイズ最適化による正則化パラメータの探索	21
4.1	ベイズ最適化 (Bayesian Optimization) . . . . .	21
4.2	ガウス過程 (Gaussian Process) . . . . .	22
4.3	ベイズ最適化のアルゴリズム . . . . .	23
第 5 章	数値シミュレーション	26
5.1	シミュレーション条件 . . . . .	26
5.2	結果 . . . . .	31
第 6 章	結言	35
6.1	まとめ . . . . .	35
6.2	今後の課題 . . . . .	35
	参考文献	37

謝辞	38
論文目録	39

# 第1章 緒言

## 1.1 研究背景と目的

近年、産業用ロボットや協働ロボットなど高速かつ高精度な位置決め制御を求められている制御対象が増加している。このような制御対象に対する制御器は高次数化または複雑化する傾向にあり、計算コスト・実装コストの増加は避けることが非常に困難な問題である。そのため、高い制御性能を保証しながら、低次数な制御器設計が求められる。本研究では、 $L_1$  正則化を考慮することにより、制御性能と労働コストの両方を考慮した制御器設計を目指す。

一般的に、産業分野で使用されている制御器設計法としてモデルベース制御器設計法が挙げられる。モデルベース制御器設計法は、制御対象のモデル同定が正確であれば、高い制御性能を持つ制御器の設計が可能となる。しかし、実際の制御システムには入出力外乱、制御対象の不確かさなどが存在し、完璧なモデル同定が困難である。そのため、望みの仕様を満たす制御器を設計した場合でも、モデル化誤差の影響から制御性能が悪化する恐れがある。また、高精度なモデル同定を行う際に、制御対象のモデルが高次数化する可能性があるため、それに伴い、設計した制御器も高次数化・複雑化する恐れがある。高次数または複雑な制御器は高単価・高い実装コストが必要となるため、制御器の実装において好まれない。一方、周波数応答データに基づいた制御器設計法は、制御対象のモデル同定を必要としない制御器設計法である。そのため、モデルベース制御器設計法の問題点であるモデル化誤差による影響を考慮する必要がなく、望みの性能を満たす制御器設計が容易である点が挙げられる。また、周波数応答データを用いた最適化計算によって自動的に制御器設計することが可能であり、かつ、制御器構造は設計者が任意に決定することが可能であるため、低次数な制御器設計が可能となる。周波数応答データに基づいた制御器設計法と類似して、時間応答に基づいたデータ駆動型制御器設計法が挙げられる。この手法では、周波数応答による推定を必要としないため、より簡易に制御器設計が可能となる。しかし、データ駆動型制御器設計では、安定余裕を考慮した設計が困難である。一方、周波数応答に基づく制御器設計法では、ナイキスト軌跡に基づいた安定性やロバスト性の

確保が容易にできるため、設計者が望む安定余裕を考慮した制御器設計が可能となる。

上記に示したように、周波数応答に基づく制御器設計法で、安定かつ低次数な制御器を最適化計算で簡易に設計することが可能である。しかし、この手法では制御器構造は設計者が任意に決定するため、制御器構造の良し悪しが設計者の熟練度に依存する恐れがある。例えば、低次数な制御器を設計できたが、良い制御性能が得られない場合や、望みの制御性能を満たすが、性能に対して冗長なパラメータを有する制御器を設計する場合がある。つまり、望みの制御性能を満たし、安定性とロバスト性を確保した上で、可能な限り低次数な制御器を決定するのは困難である。そこで、本手法では、 $L_1$  正則化を用いて、制御性能と制御器の低次数化を同時に考慮した制御器設計法を提案する。

$L_1$  正則化は、評価関数にパラメータの絶対値和を付与することで、評価関数の低減に貢献しないパラメータの削除を行う手法である。周波数応答に基づく制御器設計をする際に、最適化問題で扱われる評価関数に  $L_1$  正則化を適用することで、評価値を小さくしつつ、不要な制御器パラメータを削除することが可能となる。本研究では、制御性能に関する評価関数を考えることで、制御性能と制御器の低次数化を同時に考慮した制御器設計を提案する。

$L_1$  正則化では、性能と複雑さのバランスを表すために正則化パラメータの決定が重要となる。正則化パラメータはハイパーパラメータであり、一般的には設計者による任意の値として決定される。ハイパーパラメータは多くの場合、実験的に決定され、実験結果や設計者の熟練度に依存する可能性がある。そこで、ベイズ最適化によるハイパーパラメータの探索法を提案する。

ベイズ最適化とは、ベイズ推定を応用したブラックボックス関数推定と最適解を導出する最適化法である。推定したい関数がガウス過程に属していることを前提に、獲得関数が最大となる点を次の探索点として繰り返し最適化を行い、目的関数の推定と大域的最適解を導出する手法である。この手法を用いて、正則化パラメータを探索するデータとして扱い、望みのモデルに対して適合度が高く、スパース性も高いモデルを推定する。ここで、正則化パラメータが異なれば、評価関数自体も異なるため、評価値による比較が意味をなさないと考えられる。そのため、ベイズ最適化を適用する際に、評価値から次の探索点を決定することは困難な問題であると言える。そこで、赤池情報量規準 AIC を用いた統一的な評価指標を考慮して、正則化パラメータを探索に適用することを考える。AIC はモデル尤度とモデルの複雑さを評価する指標であり、 $L_1$  正則化によるモデルの適合度とスパース性を同時に考慮する評価指標として適していると言える。本手法では、正則化パラメータの探索にベイズ最適化を適用し、AIC が最小となる正則化パラメータを探索

させることで、制御器設計において、制御性能と制御器の低次数化を考慮した制御器パラメータの導出が可能であると考えている。

最後に、重回帰モデルのような  $1 \times q$  個のパラメータを持つモデル推定問題のみならず、多変量回帰モデルのような  $p \times q$  個の多変量なパラメータを持つモデルの推定に対しても適用したいと考えている。 $L_1$  正則化を考慮する際に、罰則項を複数にして、パラメータ行列の各行に対して罰則項を付与する。それにより、スパースなモデルでありながら、1つの罰則項を用いたときよりも評価値もしくは AIC の値が小さくなるようなモデル推定が可能であると考えられる。この手法を用いて、Gain-Scheduled 制御器の各ゲインに対して罰則項による重み付けを考慮して、各ゲインのスパース性を考慮した制御器設計を行う。罰則項が複数となる場合、正則化パラメータも複数となるが、多変量 AIC と多変数モデルに対するベイズ最適化を用いて適用が可能である。以上より、 $L_1$  正則化を適用して、制御性能と制御器の低次数化を同時に考慮した制御器設計を最適化計算によって可能とする手法を提案する。

## 1.2 論文の構成

本稿では、 $L_1$  正則化を考慮した周波数応答に基づく Gain-Scheduled 制御器設計について提案している。なお、本論文は全 6 章で構成されており、各章の構成は以下のとおりである。

**第 2 章**  $L_1$  正則化の概要とそれを用いた制御器設計の例を示す。

**第 3 章** Gain-Scheduled 制御器設計する際に、 $L_1$  正則化を考慮した凸最適化問題へと定式化する手法を示す。

**第 4 章** AIC を利用したベイズ最適化による正則化パラメータの探索法を示す。

**第 5 章** スケジューリングされた非線形要素を有する制御対象を用いた数値シミュレーションを示す。

**第 6 章** 本研究のまとめと今後の課題を述べる。

## 第2章 $L_1$ 正則化

本章では、 $L_1$  正則化 [1] を説明するために、機械学習分野における基礎知識を述べ、 $L_1$  正則化の概要について述べる。また、HDD ベンチマーク問題 [2] に対する設計に  $L_1$  正則化を適用した数値シミュレーションを示す。

### 2.1 回帰分析

一般的に、機械学習分野では回帰分析と呼ばれるモデル推定法が提案されている。回帰分析とは、取得したデータから法則性を導出し、任意の変数を変化させたときの予測値を算出する手法である。例えば、観測値  $Y_{data}$  を取得しており、予測値  $y'$  を決定するための変数を  $X$  とする。機械学習分野では、予測したい変数  $Y$  は目的変数、変数  $X$  は説明変数と呼ばれている。回帰分析では、観測値  $Y_{data}$  のデータから目的関数  $Y$  と説明変数  $X$  の関係性を数式モデルとして導出し、予測値  $y'$  を算出するようなモデルの推定を行っている。

数式を用いて説明する。説明変数  $\mathbf{X}$  のデータ  $\mathbf{X}_{data}$  に対して、目的関数  $Y$  について、 $n$  個の観測値  $\mathbf{Y}_{data}$  を取得していると仮定する。

$$\mathbf{X}_{data} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p-1} \\ 1 & x_{21} & \ddots & x_{2p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np-1} \end{pmatrix} \quad (2.1)$$
$$\mathbf{Y}_{data} = (y_0 \ y_1 \ \cdots \ y_{n-1})^T$$

説明変数は  $p$  個存在しており、この  $p$  個の要素によって  $Y$  の一つの要素が決定すると仮定する。この時、説明変数行列はヴァンデルモンド行列である。説明変数  $\mathbf{X}$  と目的関数  $Y$  に関する補間多項式を定義するため、 $\mathbf{X}$  と  $\mathbf{Y}$  がパラメータ  $\boldsymbol{\beta} = (\beta_0 \ \beta_1 \ \cdots \ \beta_{p-1})^T$



で表現可能な多項式であると仮定すると、補間多項式は以下のように表現できる。

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} \\ &= \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p-1} \\ 1 & x_{21} & \ddots & x_{2p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \cdots \\ \beta_{p-1} \end{pmatrix} \end{aligned} \quad (2.2)$$

式 (2.2) に、式 (2.1) のデータ  $\mathbf{X}_{data}$  と  $\mathbf{Y}_{data}$  を当てはめて、パラメータ  $\boldsymbol{\beta}$  を推定する。

$$\begin{aligned} J(\boldsymbol{\beta}) &= \sum_{i=1}^n (y_i - \mathbf{X}_i \boldsymbol{\beta})^T (y_i - \mathbf{X}_i \boldsymbol{\beta}) \\ \hat{\boldsymbol{\beta}} &= \arg \min \sum_{i=1}^n (y_i - \mathbf{X}_i \boldsymbol{\beta})^T (y_i - \mathbf{X}_i \boldsymbol{\beta}) \\ &= \arg \min \sum_{i=1}^n \{y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{ip-1})\}^2 \end{aligned} \quad (2.3)$$

$J(\boldsymbol{\beta})$  は損失関数と呼ばれ、モデルの当てはまり度を表している。 $\mathbf{X}_i$  は  $\mathbf{X}$  の第  $i$  行ベクトルである。パラメータ  $\boldsymbol{\beta}$  を推定することで、任意の  $x$  における予測値  $y$  を予測することができ、多項式モデルをデータから推定することを回帰分析と呼ぶ。特に、複数の説明変数で一つの目的変数が決定することを重回帰分析と呼ぶ。この重回帰分析は、周波数応答に基づく制御器設計における最適化計算と類似している。

周波数応答データから、開ループ伝達関数モデルと望みの開ループ伝達関数モデルに関するモデルマッチング誤差を最小とする問題を考える。制御対象  $P(s)$  に対して、制御器  $C(s)$  を設計すると仮定して、この時、制御器  $C(s)$  が固定次数制御器であり、制御器は制御器構造  $\Phi(s)$  で構成され、制御器パラメータ  $\rho$  に対して線形であると定義すると、制御器は  $C(s) = \rho^T \Phi(s)$  と表記される。その時、開ループ伝達関数モデル  $L(s) = C(s)P(s)$  の周波数応答を以下のように表現することができる。

$$\begin{aligned} L(\rho, j\omega_k) &= C(\rho, j\omega_k)P(j\omega_k) \\ &= \rho^T \Phi(j\omega_k)P(j\omega_k) \\ &= \rho^T R(\omega_k) + j\rho^T I(\omega_k) \end{aligned} \quad (2.4)$$

この時、 $\omega_k$  は任意の周波数点、 $R(\omega_k)$ 、 $I(\omega_k)$  はそれぞれ  $\Phi(j\omega_k)P(j\omega_k)$  の実部、虚部を表す基底を示している。制御器構造  $\Phi(s)$  は、 $p$  次元の伝達関数の行列  $\Phi(s) = (\phi_0(s) \ \phi_1(s) \ \cdots \ \phi_{p-1}(s))$  とし、それに対応するパラメータ  $\rho^T$  も  $p$  次元のパラメータ行列  $\rho^T = (\rho_0 \ \rho_1 \ \cdots \ \rho_{p-1})$  とする。周波数応答データを  $n$  点取得すると、式 (2.4) の

$R(j\omega)$  と  $I(j\omega)$  は以下の式で表される。

$$\Phi(j\omega)P(j\omega) = \begin{pmatrix} \phi_0(j\omega_1)P(j\omega_1) & \cdots & \phi_0(j\omega_n)P(j\omega_n) \\ \vdots & \ddots & \vdots \\ \phi_{p-1}(j\omega_1)P(j\omega_1) & \cdots & \phi_{p-1}(j\omega_n)P(j\omega_n) \end{pmatrix} \quad (2.5)$$

このとき、 $\Phi(j\omega)P(j\omega)$  は  $p \times n$  の行列であり、 $\Phi(j\omega)$  と  $P(j\omega)$  の行列要素の積となることに注意する。式 (2.5) を用いて、開ループ伝達関数の周波数応答は式 (2.4) を用いて、以下のように表現できる。

$$\begin{aligned} L(\rho, j\omega) &= \rho^T \Phi(j\omega)P(j\omega) \\ &= (\rho_0 \quad \rho_1 \quad \cdots \quad \rho_p) \times \begin{pmatrix} \phi_0(j\omega_1)P(j\omega_1) & \cdots & \phi_0(j\omega_n)P(j\omega_n) \\ \vdots & \ddots & \vdots \\ \phi_{p-1}(j\omega_1)P(j\omega_1) & \cdots & \phi_{p-1}(j\omega_n)P(j\omega_n) \end{pmatrix} \\ &= \begin{pmatrix} \phi_0(j\omega_1)P(j\omega_1) & \cdots & \phi_{p-1}(j\omega_1)P(j\omega_1) \\ \vdots & \ddots & \vdots \\ \phi_0(j\omega_n)P(j\omega_n) & \cdots & \phi_{p-1}(j\omega_n)P(j\omega_n) \end{pmatrix} \times \begin{pmatrix} \rho_0 \\ \rho_1 \\ \vdots \\ \rho_p \end{pmatrix} \end{aligned} \quad (2.6)$$

式 (2.2) と式 (2.6) は同様に表現することが可能となり、開ループ伝達関数の周波数応答によるモデルを重回帰モデルと扱うことができる。ただし、 $\Phi(s)P(s)$  はヴァンデルモンド行列ではないことに注意する。式 (2.6) から、開ループ伝達関数  $L(s)$  と望みの開ループ伝達関数  $L_d(s)$  のモデルマッチング誤差を最小化する問題を考える時、それぞれの周波数応答から以下のような最小化問題を定義する。

$$\min_{\rho} \sum_{k=1}^n |L_d(j\omega_k) - L(\rho, j\omega_k)|^2 \quad (2.7)$$

この最小化問題を解くことで、制御器パラメータ  $\rho$  は以下のアルゴリズムで導出される。

$$\begin{aligned} \hat{\rho} &= \arg \min_{\rho} \sum_{k=1}^n |L_d(j\omega_k) - L(\rho, j\omega_k)|^2 \\ &= \arg \min_{\rho} \sum_{k=1}^n |L_d(j\omega_k) - \rho^T \Phi(j\omega_k)P(j\omega_k)|^2 \\ &= \arg \min_{\rho} \sum_{k=1}^n \{L_d(j\omega_k) - (\rho_0 \phi_0(j\omega_k)P(j\omega_k) + \cdots + \rho_{p-1} \phi_{p-1}(j\omega_k)P(j\omega_k))\}^2 \end{aligned} \quad (2.8)$$

式 (2.8) は、式 (2.3) と同様に、望みのモデルに対して予測モデルを近づけるようなパラメータを最適化計算で算出するアルゴリズムとなっており、複素数データを用いた重回帰分析と相似している。以上より、回帰分析による回帰モデルの推定アルゴリズムが、周波数応答に基づく制御器設計における最適化計算に非常に類似していることがわかる。

これらを踏まえて、回帰分析と周波数応答に基づく制御器設計法には共通する問題点がある。式 (2.3)、式 (2.8) はパラメータを推定する際に、パラメータの数  $p$  は設計者が設定することが一般的である。回帰分析の場合、望みの回帰モデルを補間多項式で表現する際に、多項式の次数を設計者が任意に決定する。例えば、 $n$  個のデータに対してすべてのデータを通る  $n-1$  次の補間多項式を与えた場合、誤差やノイズなどの影響から、複雑でデータに強く依存した補間多項式の曲線となる可能性がある。例として、図 2.1 は、15 個のデータ点 (青い丸点) を取得し、式 (2.3) を解いて、算出したパラメータから作成した学習モデルを 14 次補間多項式 (橙色の曲線) で示した図である。

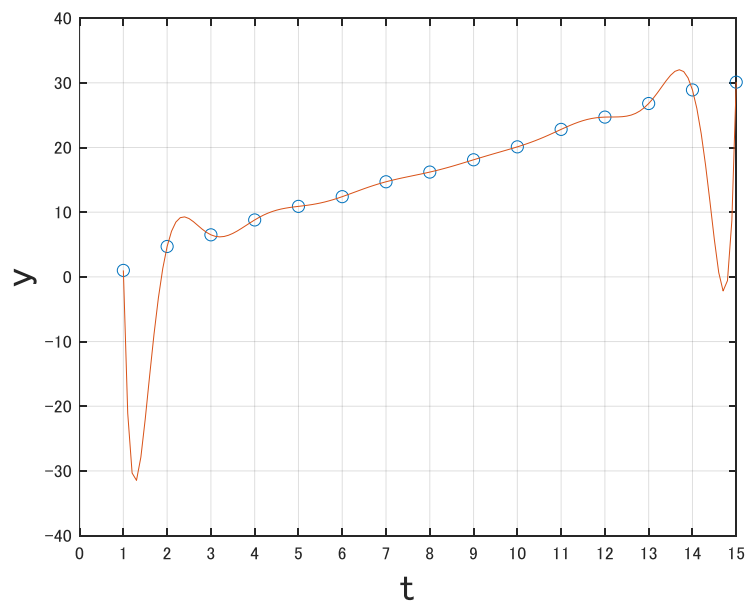


図 2.1: ノイズを含んだデータときの 14 次補間多項式モデル

実際のデータを用いて回帰分析を行う場合、データの誤差やノイズを考慮する必要がある。それらの影響を考慮せずに回帰分析を行うと、全てのデータ点を通るように働くため、図 2.1 のように、歪んだ曲線を形成してしまう。このような現象を過学習と呼び、補間多項式の次数がデータ数に比べて非常に大きい場合に発生する。図 2.1 では、直観的にデータ間は直線関係にあると推察できるが、設計者が与えた 14 次の補間多項式でモデリ

ングを行っているため、過学習が発生している。過学習抑制のために、設計者はデータ数に対して低次数な補間多項式を与えればよいが、どの程度小さい次数にすればよいかが不明であるという問題点もある。

制御器設計では、このような過学習が発生することはないが、制御性能に対して冗長的な制御器を設計する可能性がある。緒言で述べたように、周波数応答に基づく制御器設計では、設計者が任意の制御器構造と制御器の次数を与える。高次数な制御器ほど、高い制御性能を持つ傾向にあるが、性能に対して冗長な制御器パラメータを有する可能性があり、逆に、低次数な制御器で良い制御性能を得られない可能性がある。そのため、比較的に低次数な制御器であり、かつ、望みの制御性能を保証するような制御器設計が求められる。熟練の設計者であれば、これらの要求を満足するような制御器設計が可能であるが、若手の設計者にとっては困難な問題となりえる。このように、過学習と同様に、設計者が与えるパラメータの次数によって、性能・実装コストなどの面に影響をもたらす可能性がある。そこで、機械学習分野では、 $L_1$  正則化という手法が提案されており、周波数応答に基づく制御器設計での諸問題に対する解決策として提案する。

## 2.2 $L_1$ 正則化

$L_1$  正則化とは、式 (2.3) の損失関数  $J(\beta)$  に対して、罰則項 (または正則化項) を設けることで、過学習を抑制する正則化学習法の 1 つである。罰則項は、複雑さを表す項を示しており、 $L_1$  正則化では、パラメータの絶対値和を罰則項として扱っている。式 (2.3) に  $L_1$  正則化を適用した式を式 (2.9) に示す。

$$J_{L_1}(\beta) = \sum_{i=0}^{n-1} (y_i - \mathbf{X}_i \beta)^T (y - \mathbf{X}_i \beta) + \lambda \sum_{j=0}^{p-1} |\beta_j| \quad (2.9)$$

式 (2.9) の第 2 項が罰則項であり、パラメータ  $\beta$  内のパラメータの絶対値和を表している。 $\lambda$  は罰則項に対する重みであり、正則化パラメータと呼ばれる。

$L_1$  正則化は損失関数内でパラメータの大きさを評価することで、損失関数の低減に貢献しないパラメータを削除することが可能である。この手法により、設計者の試行錯誤を必要とせず、冗長なパラメータが削除された低次数な補間多項式の導出が可能となり、過学習による複雑な学習モデルを避けることができる。 $L_1$  正則化は、機械学習分野において、スパース (Sparse) モデリングで非常によく使用される手法である。図 2.1 で使用したデータを用いて、 $L_1$  正則化を考慮した損失関数式 (2.9) で損失関数が最小となるパラメータ推定アルゴリズムを用いたときの補間多項式モデルを図 2.2 に示す。

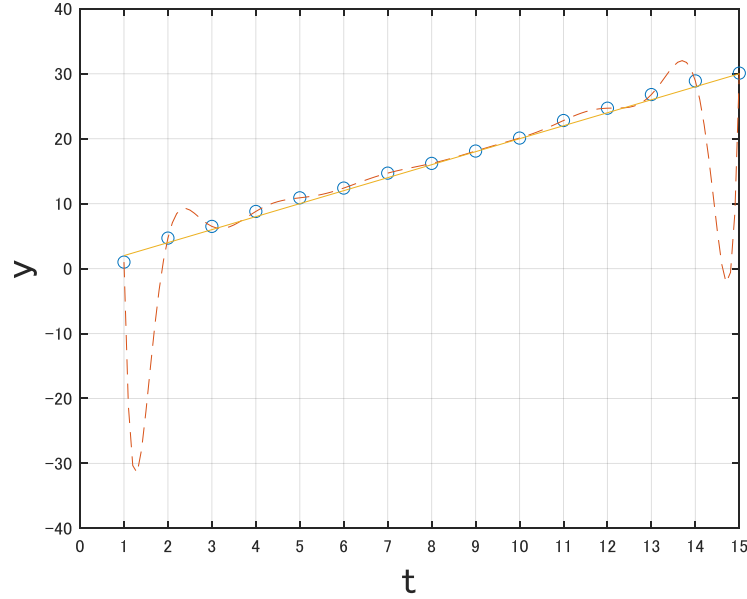


図 2.2:  $L_1$  正則化を考慮した損失関数を用いた場合の多項式モデル (直線) とノイズを含んだデータときの 14 次補間多項式モデル (破線)

図 2.2 より,  $L_1$  正則化を用いることによって, 過学習によるモデルの複雑化を回避することが可能であり, 周波数応答に基づく制御器設計にこの手法を適用することを考えた。

2.1 節の最後に述べたように, 周波数応答に基づく制御器設計では, 制御器構造と制御器パラメータは設計者が決定するため, それらの良し悪しは設計者の熟練度に依存し, 最適な制御器の次数は不明であるという問題点がある。そこで,  $L_1$  正則化を考慮した評価関数による最適化問題を解くことにより, 制御性能と制御器の低次数化を同時に考慮した制御器設計が実現を目指す。 $L_1$  正則化を考慮した場合, 式 (2.7) は以下のように変形することができる。

$$\min_{\rho} \sum_{k=1}^n |L_d(j\omega_k) - L(\rho, j\omega_k)|^2 + \lambda \sum_{j=0}^{p-1} |\rho_j| \quad (2.10)$$

正則化パラメータ  $\lambda$  を定数として与えることで, 式 (2.10) を凸な最適化問題として扱うことができる。式 (2.10) の第一項と第二項は制御器パラメータ  $\rho$  に対して共に凸な関数であり, 評価関数が凸 + 凸の形となるため, 凸最適化問題で解くことが可能である。

## 2.3 数値シミュレーション

本研究では、 $L_1$  正則化を考慮した周波数応答に基づく制御器設計として、HDD を制御対象とした設計を行った。HDD ベンチマーク問題 [2] に記載されている HDD を制御対象として、PID 制御器と FIR フィルタを設計する際に、FIR フィルタのパラメータに対して  $L_1$  正則化を適用する。フォロイング制御の性能に対して、冗長となる FIR フィルタのパラメータを削除することで、スパースな FIR フィルタの設計を行った。以下に設計問題とシミュレーション条件を記載する。

### HDD ベンチマーク問題

HDD ベンチマーク問題に記載されている HDD モデルを制御対象とする。制御対象である HDD の特性はブラックボックスと化しており、HDD を実測することは非常に困難である。そのため、シミュレーションにて統一な性能評価を行うために、次世代サーボ技術調査専門委員会は HDD に関するベンチマーク問題を作成した。図 2.3 に HDD ベンチマーク問題における制御対象の 10 種の変動モデルを示した。

HDD 装置のフォロイング制御では、制御対象の変動モデル全てに対して観測された非同期出力  $y(t)$  の  $3\sigma$  値をできる限り小さくすることで、目標追従性能が高いと判断できる。非同期出力の  $3\sigma$  値は制御器の性能を測る指標であり、10 種類の変動モデルに対してループゲイン変動 0%,  $\pm 10\%$  のすべてを組み合わせたときの値を算出する。また出力  $y(t)$  は記録再生ヘッドがディスクのトラック中心に追従したときの誤差を示している。このフォロイング制御に対する制御器設計のシミュレーションを行った。

### 制御対象の選定

制御対象を HDD ベンチマーク問題で設定してある 10 種類の変動モデルとした。またこの 10 種類の変動モデルからノミナルモデル  $P_n$  を得るとする。今回は制御対象が乗法的な不確かさ  $W_2$  を有していると仮定するので、ノミナルモデル  $P_n$  の設定法を定める必要がある。

乗法的な不確かさを有したフィードバック制御系を考える。変動モデルを  $P_{fi}(j\omega)$  ( $i = 1, \dots, 10$ ) とし、乗法的変動のゲインを  $|W_2(j\omega)| = \gamma$  とすると、各周波数におけるゲインの大きさは以下の (2.11) 式で表すことができる。

$$\left| \frac{P_{fi}(j\omega) - P_n(j\omega)}{P_n(j\omega)} \right| < \gamma \quad (2.11)$$

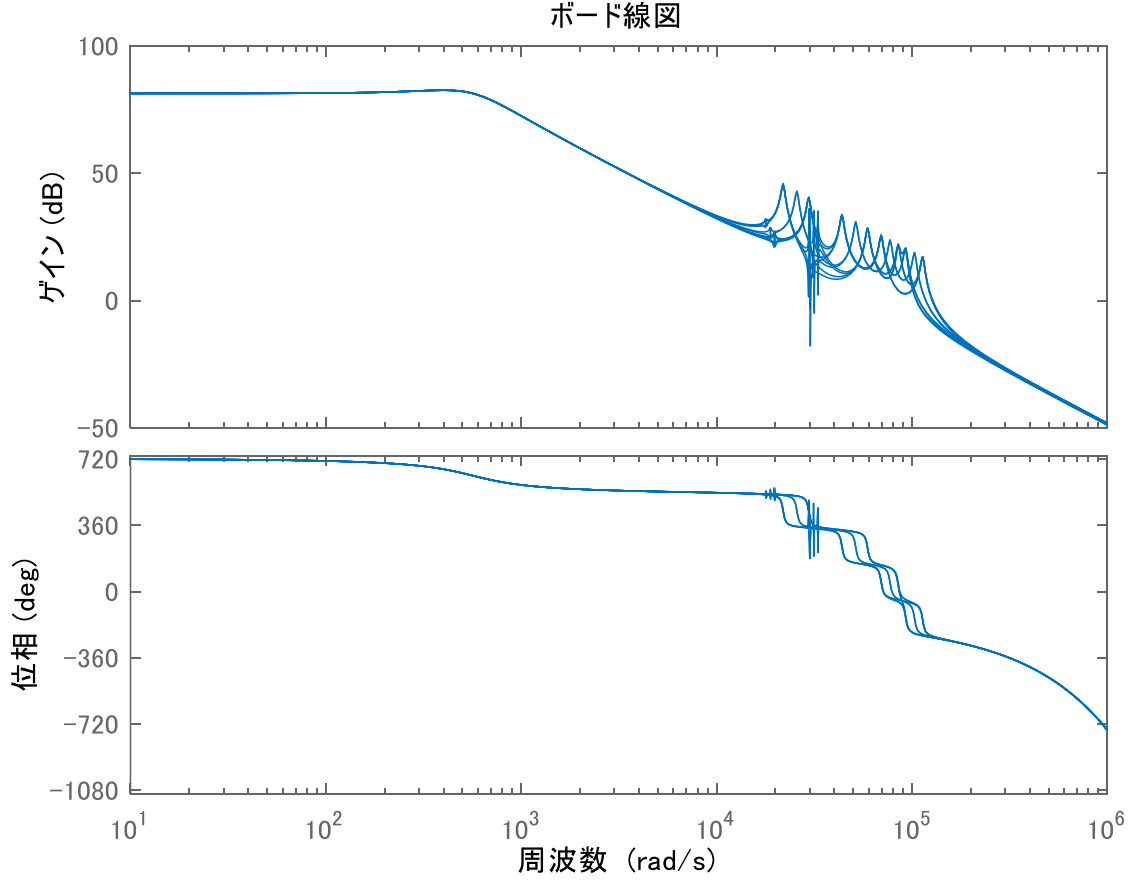


図 2.3: 制御対象の 10 種類の変動モデル

この式から,  $\gamma$  を最小化するときの  $P_n(j\omega)$  を求める。

これらを求めるにあたり, モデル  $P_n$  及び  $P_{fi}$  を複素数平面上にあると考え, (2.12) 式で表す。

$$\mathbf{a}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, z = \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.12)$$

そして, (2.11) の両辺を 2 乗することで (2.13) 式を得る。

$$\begin{aligned} (\mathbf{a}_i - z)^T (\mathbf{a}_i - z) &< \gamma^2 z^T z \\ \Leftrightarrow \mathbf{a}_i z + z^T \mathbf{a}_i - \mathbf{a}_i^T \mathbf{a}_i - (1 - \gamma^2) z^T z &> 0 \quad \forall i \end{aligned} \quad (2.13)$$

ここで  $\delta \equiv 1/\sqrt{1-\gamma^2}$  と置き, シューア補行列を (2.13) 式に適用して LMI(Linear Matrix Inequality) を構築する。

$$\begin{bmatrix} \delta^2 \mathbf{I} & z \\ z^T & \mathbf{a}_i z + z^T \mathbf{a}_i - \mathbf{a}_i^T \mathbf{a}_i \end{bmatrix} > 0 \quad \forall i \quad (2.14)$$

この LMI から、式 (2.14) を制約式とし、 $\delta^2$  が最小となる最適化問題をとくことで、制御対象のノミナルモデルを構築する。

$$\begin{aligned} \min \quad & \delta^2 \\ \text{subject to} \quad & (2.14) \end{aligned} \quad (2.15)$$

制御システムは乗法的不確かさを持つ制御対象に対して、PID 制御器とノッチフィルタを組み合わせた制御器を用いた単一フィードバックシステムとした。

### 制御器構造

制御器構造は PID 制御器と FIR フィルタを組み合わせたものとした。(2.16) 式に制御器  $C(z)$  の式を示した。

$$C(z) = \left[ 1, \frac{3.788 \times 10^{-5}}{z-1}, \frac{10000z-10000}{z-0.6847}, \Phi_{FIR} \right] \begin{bmatrix} K_p \\ K_i \\ K_d \\ \rho_{fir} \end{bmatrix} \quad (2.16)$$

ただし、

$$\Phi_{FIR} = [z^{-1}, z^{-2}, \dots, z^{-p}] \quad (2.17)$$

$$\rho_{fir} = [\rho_{fir1}, \rho_{fir2}, \dots, \rho_{firp}] \quad (2.18)$$

とした。サンプリングタイムを  $3.788 \times 10^{-5} s$  で離散化したものを用いる。本シミュレーションでは、低次数な FIR フィルタの設計を行うために、罰則項を FIR フィルタのパラメータに対してのみ付与しており、FIR フィルタの次数は  $L_1$  正則化の有効性を測るために 100 次で設計を行った。先行研究 [3] では、FIR フィルタの次数を試行錯誤で、14 次と決定しており、これを比較対象とした。

### 最適化問題

この時の望みのモデル  $M(s)$  と望みの開ループ伝達関数  $L_d(s)$  を以下のように設定した。

$$M(s) = \frac{6300}{s+6300} \quad (2.19)$$

$$L_d(s) = \frac{6300}{s} \quad (2.20)$$

最適化問題は、式 (2.10) を用いて、FIR フィルタのパラメータに対する罰則項とした。

$$\min_{\rho} \sum_{k=1}^n |L_d(j\omega_k) - L(\rho, j\omega_k)|^2 + \lambda \sum_{j=1}^p |\rho_{firj}| \quad (2.21)$$



### シミュレーション結果

10 種類の変動モデルに対してノミナルループゲイン (変動が 0% 時) における  $3\sigma$  値の平均値を評価する。加えて, FIR フィルタのパラメータの絶対値の最大値  $\rho_{max}$  に対して 1% 以下の大きさのパラメータが連続して発生した場合, そのときの次数を  $n_{0.01}$  と置く。そして, 次数  $n_{0.01}$  以下のパラメータは不要であると考え削除する。まず, 罰則項を用いない場合 (正則化パラメータ  $\lambda = 0$  の場合) の  $3\sigma$  値の平均値は 8.62 となり, 10 種類の変動モデルに対するそれぞれの  $3\sigma$  のグラフを図 2.4 に示す。

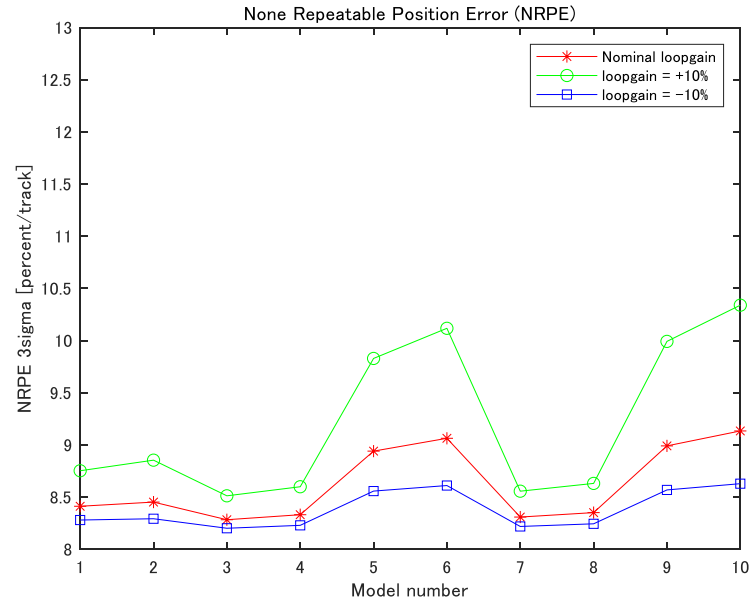


図 2.4: 正則化パラメータ  $\lambda$  を 0 とした時の  $3\sigma$  値

この時のパラメータを (2.22) 式に示し, FIR フィルタのパラメータの分布図を図 2.5 に示す。

$$\rho = \begin{bmatrix} K_p \\ K_i \\ K_d \\ \rho_{fir} \end{bmatrix} = \begin{bmatrix} -0.954 \\ 0.538 \\ -0.937 \\ -0.305 \\ -0.198 \\ \dots \\ 0.00051 \\ -0.00043 \end{bmatrix} \quad (2.22)$$

このとき FIR フィルタのパラメータの最大値  $\rho_{max} = -0.305$  となり, 最大値の 1% 以

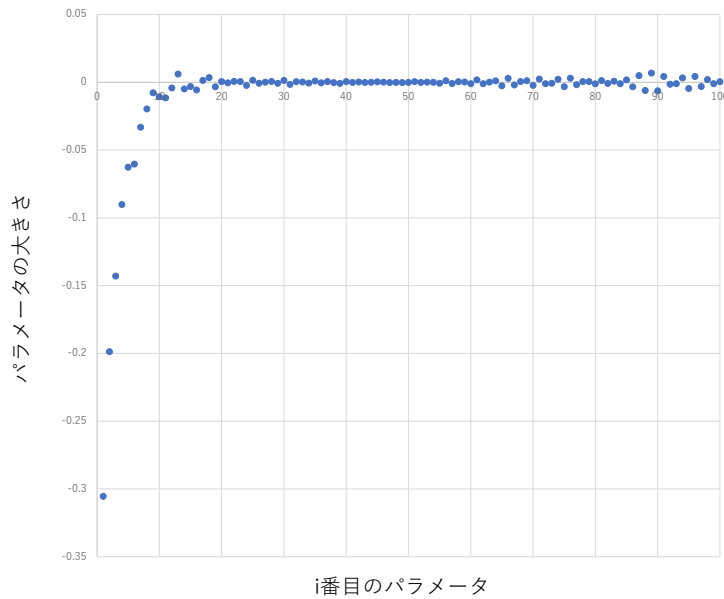


図 2.5: 正則化パラメータ  $\lambda$  を 0 とした時のパラメータの分布図

下が連続して発生するときの次数は 20 次であった。

次に正則化パラメータ  $\lambda$  を 1, 5, 10 とした時の  $3\sigma$  値を表 2.1 示し、同様に 10 種類の変動モデルに対するそれぞれの  $3\sigma$  値のグラフを図 2.6, 2.7, 2.8 に示す。

表 2.1: 正則化パラメータを変化させた時の  $3\sigma$  値

	$\lambda = 0$	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$
$3\sigma$ 値	8.62	8.75	8.81	8.83
次数 $n_{0.01}$	20	18	15	15

罰則項を付与したときに FIR フィルタの次数が低減していることが分かる。また、正則化パラメータが大きいほど、FIR フィルタのパラメータの最大値  $\rho_{max}$  の 1% 以下が連続して発生する次数  $n_{0.01}$  が小さくなることがわかる。このことから、正則化パラメータ  $\lambda$  が大きくなるほど性能としては下がるものの、次数を低減した制御器を設計することが可能となった。

また先行研究 [3] では、FIR フィルタの次数を試行錯誤して 14 次で設計を行った。本シミュレーション結果と比較すると正則化パラメータを変化させることで FIR フィルタの次数を近づけることができた。しかし、次数の低減率が正則化パラメータ  $\lambda$  に依存し

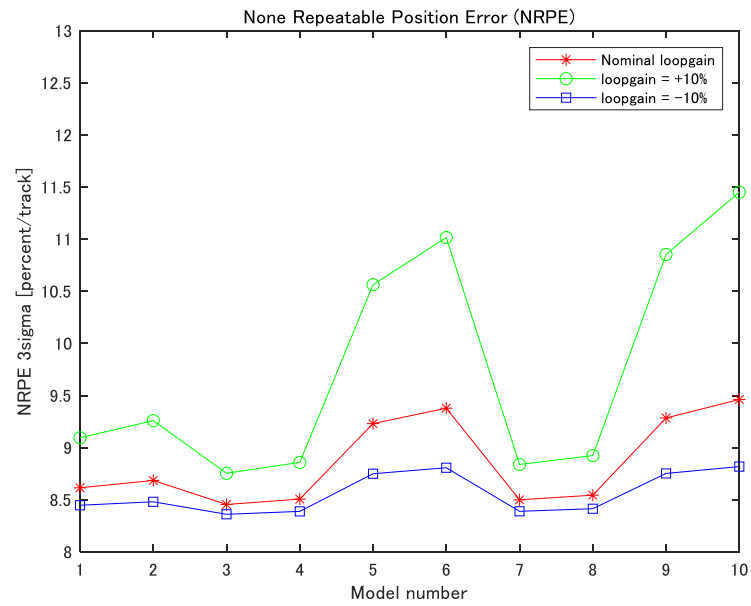


図 2.6: 正則化パラメータ  $\lambda$  を 1 とした時の  $3\sigma$  値

ているため、その決定法は試行錯誤的になると予測される。そこで、正則化パラメータの決定法については、4 章のベイズ最適化を用いた探索法を提案する。

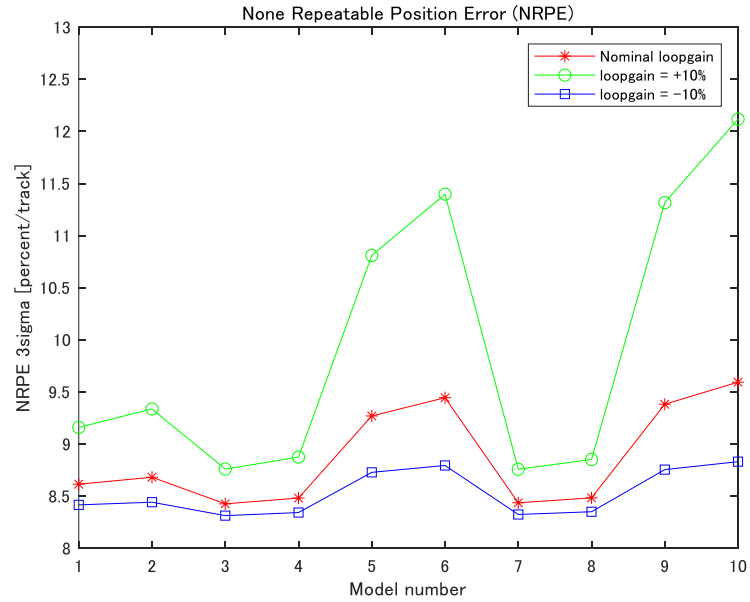


図 2.7: 正則化パラメータ  $\lambda$  を 5 とした時の  $3\sigma$  値

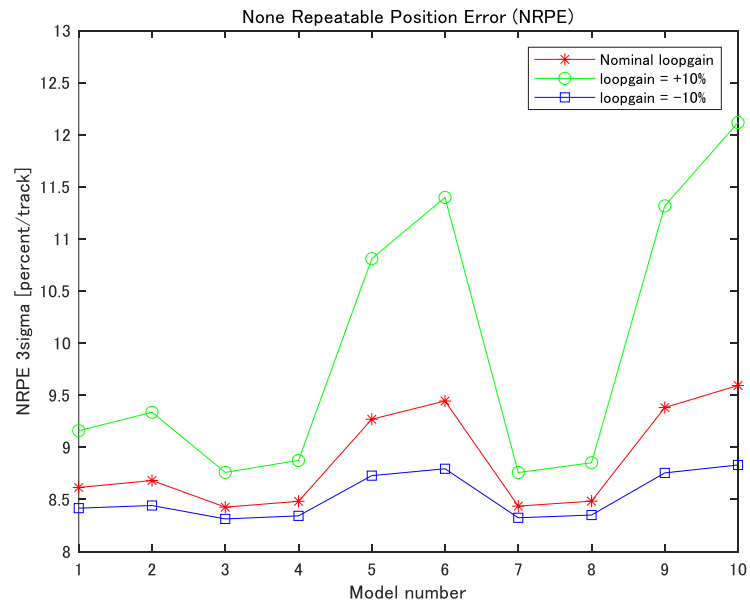


図 2.8: 正則化パラメータ  $\lambda$  を 10 とした時の  $3\sigma$  値

## 第3章 Gain-Scheduled 制御

2.3 節では、HDD に対して FIR フィルタを設計する際に、評価関数を FIR フィルタの重回帰モデルとして扱い、 $L_1$  正則化を適用することで冗長な FIR フィルタのパラメータを削減する数値シミュレーションを行った。本章では、Gain-Scheduled 制御器による開ループ伝達関数モデルを多変量回帰モデルとして扱い、 $L_1$  正則化を適用して各ゲインに対する Scheduling 変数の次数低減について考える。

### 3.1 Gain-Scheduled 制御の概要

線形時不変システムは名前の通りに入出力関係が線形かつ時間変化が発生しないシステムであり、そのような制御対象の伝達関数モデルは変化せず、周波数応答データや時系列データも変化しないと仮定されている。しかし、実際の制御対象は静的なシステムではなく、時間変化で特性が変化するものや非線形なシステムなど動特性を有するものが一般的である。そのため、動特性を有する制御対象に対する制御器設計は、線形時不変な制御対象に対する制御器設計と比較して困難である。非線形な制御対象に対する制御器設計は様々なものが提案されており、本研究では、Gain-Scheduled 制御<sup>[4]</sup>(以下、GS と略す。)に着目した。

Gain-Scheduled 制御とは、制御対象の変動をスケジューリングすることで、その変動に対応した制御器で制御を行うことである。制御対象が Scheduling 変数というパラメータに依存していると仮定し、制御器自体もその Scheduling 変数に依存して変化させることで、非線形な制御対象を線形な制御対象の群とみなして、それに対応させた線形な制御器の群を設計することができる。例えば、制御対象  $P(s)$  が Scheduling 変数  $\theta$  に依存して変化する場合、制御対象は  $P(s, \theta)$  と表される。GS 制御器  $C(s)$  は Scheduling 変数に依存して、 $C(s, \theta)$  と表記される。ここで、Scheduling 変数  $\theta$  が  $-1, 0, 1$  と変化するとき、制御対象  $P(s, -1)$  には制御器  $C(s, -1)$  が設計され、制御対象  $P(s, 0)$  には制御器  $C(s, 0)$  が設計され、制御対象  $P(s, 1)$  には制御器  $C(s, 1)$  が設計される、といったように、Scheduling 変数の変化に対応した制御器設計を行うことが Gain-Scheduled 制御器設計となる。文献 [4] では、周波数応答に基づく GS 制御器設計が提案されており、これ

を参考にし、設計問題を記述する。

## 3.2 設計問題

設計問題として、制御対象  $P(s)$  が Scheduling 変数  $\theta$  でスケールされていると仮定し、それに対応した制御器  $C(s, \theta)$  を設計し、開ループ伝達関数  $L(s, \theta)$  と望みの開ループ伝達関数  $L_d(s)$  の周波数応答のモデルマッチング誤差の最大値を最小化する問題を考える。

### Gain-Scheduled 制御

まず、Gain-Scheduled 制御をするために、Scheduling 変数  $\theta$  は  $n_\theta$  個の任意の実数と定義する。

$$\theta = \{\theta_l | l = 1, 2, \dots, n_\theta\} \quad (3.1)$$

このとき、制御対象と制御器は  $n_\theta$  個のモデルが存在している。モデルの数  $n_\theta$  は Scheduling 変数  $\theta$  に関して、十分に細かいグリッドでも利用可能であると仮定されている。

次に、制御対象を Scheduling 変数  $\theta$  でスケールする。制御対象  $P(s, \theta)$  は、SISO LPV システムであり、制御対象のすべてのモデルは安定であると仮定する。すべての制御対象  $P(s, \theta)$  は式 (3.2) で表記される。

$$P(s, \theta) = \{P(s, \theta_l) | l = 1, 2, \dots, n_\theta\} \quad (3.2)$$

それに対応して、 $n_c$  次の制御器  $C(s, \theta)$  は式 (3.3) で表記される。

$$C(s, \theta) = \rho^T(\theta)\phi(s) \quad (3.3)$$

ここで、制御器パラメータ  $\rho(\theta)$  と制御器構造  $\phi(s)$  は以下のように表される。

$$\begin{aligned} \rho^T(\theta) &= [\rho_1(\theta), \rho_2(\theta), \dots, \rho_{n_c}(\theta)] \\ \phi^T(s) &= [\phi_1(s), \phi_2(s), \dots, \phi_{n_c}(s)] \end{aligned} \quad (3.4)$$

式 (3.4) の  $\rho_i(\theta) (i = 1, 2, \dots, n_c)$  は、式 (3.1) を用いて、以下のように表現する。

$$\rho_i(\theta) = (\rho_{i,p_c})^T \theta^{p_c} + (\rho_{i,p_c-1})^T \theta^{p_c-1} + \dots + (\rho_{i,1})^T \theta + (\rho_{i,0})^T \quad (3.5)$$

このとき、 $p_c$  は Scheduling 変数の次数である。式 (3.5) より、Scheduling 変数に対するパラメータは  $n_c \times (p_c + 1)$  個導出される。式 (3.4) に式 (3.5) を代入し、パラメータ  $\rho_{i,j}$  と Scheduling 変数  $\theta$  を分けることで、式 (3.6) が得られる。

$$\begin{aligned}
 \boldsymbol{\rho}^T(\boldsymbol{\theta}) &= \begin{pmatrix} (\rho_{1,p_c})^T \theta^{p_c} + (\rho_{1,p_c-1})^T \theta^{p_c-1} + \cdots + (\rho_{1,1})^T \theta + (\rho_{1,0})^T \\ (\rho_{2,p_c})^T \theta^{p_c} + (\rho_{2,p_c-1})^T \theta^{p_c-1} + \cdots + (\rho_{2,1})^T \theta + (\rho_{2,0})^T \\ \vdots \\ (\rho_{n_c,p_c})^T \theta^{p_c} + (\rho_{n_c,p_c-1})^T \theta^{p_c-1} + \cdots + (\rho_{n_c,1})^T \theta + (\rho_{n_c,0})^T \end{pmatrix} \\
 &= \begin{pmatrix} (\rho_{1,p_c})^T & \cdots & (\rho_{1,1})^T & (\rho_{1,0})^T \\ \vdots & \ddots & \ddots & \vdots \\ (\rho_{n_c,p_c})^T & \cdots & (\rho_{n_c,1})^T & (\rho_{n_c,0})^T \end{pmatrix} \begin{pmatrix} \theta^{p_c} \\ \vdots \\ \theta \\ 1 \end{pmatrix}
 \end{aligned} \tag{3.6}$$

式 (3.6) により、評価関数は凸関数として扱うことが可能であり、パラメータの導出が容易に可能である。Scheduling 変数の次数  $p_c$  は設計者が任意に与えるものであり、制御対象のスケージングに合わせて調整することが求められる。本研究では、初期に適当な次数  $p_c$  を与え、 $L_1$  正則化により不要なパラメータ  $\rho_{i,j}$  を削除することで、設計者が適切な次数を与えることなく、最適化計算で GS 制御器の次数を決定することが可能となる。

#### AIC による統一的な評価<sup>[5]</sup>

設計した制御器による制御性能の評価は様々な方法が存在し、評価関数の評価値を比較する場合がある。評価値での比較は非常に簡単に比較が可能であるが、 $L_1$  正則化を用いた場合では、正則化パラメータが異なると評価関数自体が異なるものとなるため、評価値による性能の比較ができない。そのため、開ループ伝達関数モデルに関する統一的な評価のために、本研究では、赤池情報量規準 AIC(Akaike Information Criterion) を評価指標として用いる。AIC とは、赤池弘次氏により提案された統計学の理論であり、モデルの尤度と複雑さを同時に考慮した評価指標である。2 での説明変数  $\mathbf{X}$  と目的関数  $\mathbf{Y}$  とパラメータ  $\boldsymbol{\beta}$  を用いて、重回帰モデルに関する AIC を示した式を式 (3.7) に示す。

$$AIC = -2 \log \left( \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \mathbf{X}_i \boldsymbol{\beta})^T (y_i - \mathbf{X}_i \boldsymbol{\beta}) \right) + 2p \tag{3.7}$$

式 (3.7) の第一項は重回帰モデルのモデル尤度を示しており、第二項はモデルの複雑さを表現している。つまりは、第一項が小さいほど、モデルの当てはまりがよく、第二項が小さいほど、少ないパラメータでモデルが設計されていることを示している。設計問題として、望みの開ループ伝達関数モデル  $L_d(s)$  に対する開ループ伝達関数モデル  $L(s)$  の AIC を考える。式 (2.6) を用いて、周波数応答データに関する AIC を導出すると式 (3.8) のように表される。

$$AIC = -2 \log \left( \frac{1}{n} \sum_{k=1}^n (L_d(j\omega_k) - L(\rho, j\omega_k))^* (L_d(j\omega_k) - L(\rho, j\omega_k)) \right) + 2(p+1) \quad (3.8)$$

周波数応答は複素数であるため、複素転置 (ここでは行列  $A$  の複素転置行列を  $A^*$  と表現) を用いて AIC を算出する。多変量回帰モデルである GS 制御器による開ループ伝達関数モデルは、全てのパラメータを同列に考えた場合の AIC は式 (3.9) のように表現される。

$$\mathbf{AIC} = \{AIC_l | l = 1, 2, \dots, n_\theta\}$$

$$AIC_l = -2 \log \left( \frac{1}{n} \sum_{k=1}^n (L_d(j\omega_k) - L_l(\boldsymbol{\rho}(\theta_l), j\omega_k))^* (L_d(j\omega_k) - L_l(\boldsymbol{\rho}(\theta_l), j\omega_k)) \right) + 2n'_c(p_c + 1) \quad (3.9)$$

GS 制御の場合では、 $n_\theta$  個の Scheduling 変数  $\theta_l$  によって開ループ伝達関数の数が変化的ることから、望みの開ループ伝達関数に対する AIC も  $n_\theta$  個算出される。また、モデルの複雑さを表現する第二項は Scheduling 変数に関するパラメータ  $\rho_{i,j}$  の数を考慮している。後に示す最適化問題では  $L_1$  正則化を考慮した評価関数を適用するため、初期に与える Scheduling 変数の次数  $n_c$  が変化する可能性がある。そのため、AIC の第二項は設計後の Scheduling 変数の次数  $n'_c$  により算出され、モデルの当てはまりと複雑さを AIC で比較することが可能となる。以上より、GS 制御器で設計した開ループ伝達関数モデルに関する AIC を導出することで、設計した制御器の比較検討を行うことができる。



## 第4章 ベイズ最適化による正則化パラメータの探索

$L_1$  正則化を考慮した評価関数では、罰則項に正則化パラメータと呼ばれるハイパーパラメータが重みとして与えられている。正則化パラメータは評価関数の第一項である性能を評価する項と第二項であるモデルの複雑さを評価する項のバランスをとる際に重要となるパラメータである。一般的な機械学習において、正則化パラメータの決定は交差検証法などのように実験的に導出する場合があり、正則化パラメータの決定に設計者の試行錯誤的な作業が必要とされる。本研究では、制御器設計において、設計者による制御器次数の決定の煩わしさを排除する手法を提案しているため、 $L_1$  正則化を適用することによる設計者への煩わしい作業を減らしたい。そこで、ベイズ最適化による正則化パラメータの探索を行うことで、設計者が関与することなく、評価基準となる値が最適となる正則化パラメータの導出を提案する。本研究では、評価基準となる値を AIC として、ベイズ最適化により、AIC が最小となる正則化パラメータを探索を行う。

### 4.1 ベイズ最適化 (Bayesian Optimization)

ハイパーパラメータとは、機械学習分野などで学習などを行う前に、設計者が任意の与えるパラメータのことを指す。事前に与える必要があるパラメータであるため、学習によって一意に導出できるものではない。しかし、ハイパーパラメータや重みの導出に際し、様々な最適化アルゴリズムが提案されており、重みの探索に関する研究もなされている。本研究では、正則化パラメータの探索法として、ベイズ最適化を用いた探索法を提案する。ベイズ最適化とは、ベイズ推定を応用したブラックボックス関数の推定と最適解を導出する最適化法の 1 つである。ベイズ最適化の特徴は観測データを用いた推定法であり、推定したい関数がガウス過程に従うことが前提となっている。ベイズ最適化は、推定したい関数が連続系でも離散系であっても使用可能であり、GridSearch などの網羅的に探索する手法よりも少ない評価回数で推定が可能となる手法と言われている。また、目的関数が多変数な関数でも使用可能であり、複数の変数やパラメータを有する関数の

推定に適用可能である。文献 [6] では制御器調整でベイズ最適化が使用されており、パラメータの推定に役立つ手法である。

## 4.2 ガウス過程 (Gaussian Process)

ガウス過程とは、ガウス分布を用いた関数推定のための確率過程である。ガウス過程では、取得データを用いて事前確率分布を作成し、推定したい関数が存在するであろう不確かさを平均と標準偏差で表現することが可能となる。不確かさが大きい部分では、関数推定において情報が少ない部分を示しており、局所解や大域最適解が含まれている可能性がある。ベイズ最適化では、獲得関数を用いて、その不確かさが大きい領域で探索点を決定するアルゴリズムとなっている。獲得関数によって決定した探索点を含めた取得データから近似関数を更新することで、不確かさが大きい領域を狭めていき、効率的に探索点の決定をすることができる。ベイズ最適化を説明するためにはガウス過程についての知識が必要となるため、いくつかの数式を用いて、ガウス過程について説明する。まず、変数  $x$  に対して、ブラックボックスな関数  $f(x)$  を推定したいと考える。このとき、 $x$  と  $f(x)$  に関する  $n$  個の観測データ  $D_{1:n}$  が得られているとする。

$$D_{1:n} = \{(x_1, x_2, \dots, x_n), (f(x_1), f(x_2), \dots, f(x_n))\} \quad (4.1)$$

このデータ  $D_{1:n}$  から平均関数を用いた平均と共分散行列を算出する。平均関数  $\mu(x)$  と共分散行列  $\Sigma_0$  内で用いられるカーネル関数  $k(x_i, x_j)$  は様々な種類が存在する。本研究では、平均関数は単なる平均  $\mu_0$  として扱い、カーネル関数  $k(x_i, x_j)$  はガウスカーネルを使用する。式 (4.2) にガウスカーネルを用いた共分散行列を示す。

$$\Sigma_0(x_{1:n}) = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix} \quad (4.2)$$

$$k(x_i, x_j) = a_0 \exp(-\|x_i - x_j\|^2) \quad (4.3)$$

ここで、 $f(x)$  は平均と共分散行列を用いた正規分布に従う時、 $f(x)$  はガウス過程に属しているといえる。次に未知の  $x_{next}$  (次に探索する  $x$ ) に対する  $f(x_{next})$  は事前のデータ  $D$  と平均  $\mu_0$  と共分散行列  $\Sigma_0$  を用いた事後平均  $\mu(x_{next}|D_{1:n})$  と事後分散  $\sigma(x_{next}|D_{1:n})$  によって特徴づけられる。このときの事後平均  $\mu(x_{next}|D_{1:n})$  と事後分散  $\sigma^2(x_{next}|D_{1:n})$  を式 (4.4), (4.5) に示す。

$$\begin{aligned} \mu(x_{next}|D_{1:n}) = \\ \mu_0 + k_{mat}(x_{next})^T \Sigma_0(x_{1:n})^{-1} (f(x_{1:n}) - \mu_0(x_{1:n})) \end{aligned} \quad (4.4)$$

$$k_{mat}(x_{next}) = [k(x_{next}, x_1), \dots, k(x_{next}, x_n)]$$

$$f(x_{1:n}) = [f(x_1), f(x_2), \dots, f(x_n)]^T$$

$$\sigma^2(x_{next}|D_{1:n}) =$$

$$k(x_{next}, x_{next}) - k_{mat}(x_{next})\Sigma_0(x_{1:n})^{-1}k_{mat}^T(x_{next}) \quad (4.5)$$

このとき、 $f(x_{next})$  は事後平均と事後分散を用いた正規分布に従うといえる。

$$f(x_{next}) \sim N(\mu(x_{next}|D_{1:n}), \sigma^2(x_{next}|D_{1:n})) \quad (4.6)$$

これはデータ  $D$  を事前を取得している状態で、次の探索点における  $f(x_{next})$  が事後平均  $\mu(x_{next}|D_{1:n})$  と事後分散  $\sigma(x_{next}|D_{1:n})$  を用いた正規分布に従うことを意味している。

### 4.3 ベイズ最適化のアルゴリズム

次にベイズ最適化のアルゴリズムについて説明する。ベイズ最適化を使用する際は、推定したい関数がガウス過程に属していることを前提に繰り返し最適化で目的関数の推定と大域的最適解の導出を行う。

まず、ガウス過程で示したように、式 (4.1) を取得していると仮定する。このとき、 $f(x)$  を評価関数として、得られたデータから事後平均と事後分散を算出する (式 (4.4), (4.5))。この平均と分散を用いた獲得関数を定義し、獲得関数が最大となる点を次の探索点  $x_{next}$  とする。獲得関数には様々な種類があり、本研究では一般的に使用される信頼性下限関数 (LCB: Lower Confidence Bound) を使用する。 (式 (4.7))

$$LCB(x) = \mu(x) - k\sigma(x) \quad (4.7)$$

この獲得関数を用いて、次の探索点  $x_{next}$  を決定するため、更新式は以下のように定義される。

$$\max_{x_{next} = x} LCB(x) \quad (4.8)$$

次に、この  $x_{next}$  に対して  $f(x_{next})$  を観測し、その観測データを追加したデータ  $D_{next}$  を作成する。

$$D_{next} = \{(x_1, x_2, \dots, x_n, x_{next}),$$

$$(f(x_1), f(x_2), \dots, f(x_n), f(x_{next}))\} \quad (4.9)$$

その後、更新したデータ  $D_{next}$  から、事後平均と事後分散を更新し、新たな獲得関数からさらに次の探索点を得る。これを繰り返す行うことで、探索点が大域的最適解に収束するアルゴリズムとなっている。本研究で正則化パラメータの探索のためにこのベイズ最適化を用いる。まず、任意の正則化パラメータ  $\lambda$  を 2 つ以上決定し、そのときの AIC を観測する。 $\lambda$  に対する AIC の値をデータ  $D_n$  に格納し、これをガウス過程における関数推定に適用する。

$$D = \{(\lambda_{min}, \dots, \lambda_{max}), (AIC_{\lambda_{min}}, \dots, AIC_{\lambda_{max}})\} \quad (4.10)$$

式 (4.10) のデータから、 $\lambda$  に対して平均と共分散行列を求め、事後平均と事後分散を算出する。この事後平均と事後分散を用いた獲得関数を定義し、これが最大となる次の探索する正則化パラメータ  $\lambda_{next}$  を決定する。このとき、使用する獲得関数は信頼性下限関数 (LCB: Lower Confidence Bound) (式 (4.7)) と文献 [7] より近似関数精度を用いた A-LCB を使用する。

#### 近似関数精度に基づいた信頼性下限関数 A-LCB

A-LCB とは、近似関数精度 AE を定義し、近似関数と目的関数の誤差を計算して、その変化率からハイパーパラメータ  $k$  を変化させることが可能な LCB である。A-LCB ではハイパーパラメータ  $k$  を動的なパラメータとして扱い、探索の序盤から終盤にかけてハイパーパラメータ  $k$  の大きさを変化させることで、探索区域を未知の区域から既存点付近にシフトさせ、探索効率の向上を図っている。数式を用いて A-LCB を説明する。まず、未知の目的関数  $f(x)$  を、評価回数  $n$  回で  $m$  個のサンプル点  $x$  を得ているとし、この時の平均  $\mu_x$  を計算する。ここで、近似関数誤差 AE を以下のように定義する。

$$AE = \frac{\sum_{t=1}^m (f(x_t) - f(\mu_x))}{m} \quad (4.11)$$

これは  $m$  個のデータ点に対する誤差の平均値である。近似関数誤差の変化率を求めるために、初期の評価 ( $n = 1$  の時) と評価後の近似関数誤差の相対平均誤差 RAE を算出する。その相対平均誤差から変化率  $s$  を算出し、それに基づいてハイパーパラメータ  $k$  を調整する。

$$RAE_n = \frac{AE_n}{AE_{n=1}} \quad (4.12)$$

$$s_n = RAE_n - RAE_{n-1} \quad (4.13)$$

このとき,  $RAE_{n=1} = 1$  である。ハイパーパラメータ  $k$  を調整するための式を式 (4.14) に示す。

$$\beta = e^{-\alpha s_n} \quad (4.14)$$

$$k_{n+1} = k_n \beta, s_n > 0 \quad (4.15)$$

$$k_{n+1} = k_n, s_n < 0 \quad (4.16)$$

$\alpha$  は  $k$  の変化率をコントロールするための値であり,  $\alpha > 0$  と設定する。上式から,  $s_n > 0$  の場合,  $-\alpha s_n < 0$ ,  $0 < \beta < 1$  となり,  $k_n$  が徐々に減少する。一方,  $s_n < 0$  の場合,  $k_n$  は変化しない。

探索序盤では, 近似関数が目的関数に対して大きく異なっていると推測されるため, 変化率  $s_n$  が比較的大きくなる。そのため, 式 (4.14) の  $\beta$  は大きな値となり, 式 (4.15) の更新後のハイパーパラメータ  $k_{n+1}$  は比較的小さくなるため, 未知の区域を探索しようとする。一方, 探索終盤では, ある程度近似関数が目的関数に近づいていると推測されるため, 変化率  $s_n$  が比較的大きくなる。そのため, 式 (4.14) の  $\beta$  は小さな値となり, 式 (4.15) の更新後のハイパーパラメータ  $k_{n+1}$  は大きく変化しないため, 既存の探索点で最も良い探索点付近を探索する。この手法を用いることにより, ベイズ最適化で必要となるハイパーパラメータの決定に設計者が関与することなく, データに対応したハイパーパラメータ  $k$  を得ることが可能となる。

本研究では,  $x$  が正則化パラメータ  $\lambda$  となり,  $f(\lambda_t)$  が正則化パラメータを  $\lambda_t$  としたときの AIC の値となる。これらの値はベイズ最適化で使用する AIC のデータを用いて行われる。

## 第5章 数値シミュレーション

本章では、スケジューリングされた非線形要素を含む制御対象に対して GS 制御器を設計する際に  $L_1$  正則化を適用した数値シミュレーションを行う。 $L_1$  正則化を用いることにより、制御性能に対して不必要となる Scheduling 変数に対するパラメータの次数低減を目指し、 $L_1$  正則化を用いない方法と比較し、AIC による評価を行う。それにより、 $L_1$  正則化を用いた手法が制御性能と制御器の複雑さの両方を同時に考慮した設計であるかを確認する。また、ベイズ最適化を用いることにより、 $L_1$  正則化で使用される正則化パラメータの決定を自動的に行う。シミュレーション条件は文献 [4] を参考にしている。

### 5.1 シミュレーション条件

#### 制御対象

制御対象は、非線形ばねの一つである SAT(Stiffness Adjustable Tendon) を含むマspbaneダンパシステムとした。SAT はバネが伸展時と収縮時のバネの位置でばね定数が一定ではなく、シミュレーションではばね定数がばねの位置で変化する特性として扱う。バネの位置を  $x$ 、バネ定数を  $k(x)$  としたときの SAT の特性近似式を式 (5.1) に示し、特性近似式から本稿で扱う SAT の特性を図 5.1 に示す。

$$f_a(x) \triangleq a_1 e^{a_2 x} + a_3 \quad (5.1)$$

式 (5.1) はヒステリシス特性に対して最小二乗法を適用して指数関数として近似した特性式である。このとき、式 (5.1) をテイラー展開することでバネ特性  $k(x)$  も近似式で表現することが可能である。式 (5.1) をテイラー展開した特性式を式 (5.2) に示す。

$$f_a(x) \simeq a_1 \left( 1 + a_2 x + \frac{a_2^2}{2!} x^2 + \frac{a_2^3}{3!} x^3 + \cdots \right) + a_3 \quad (5.2)$$

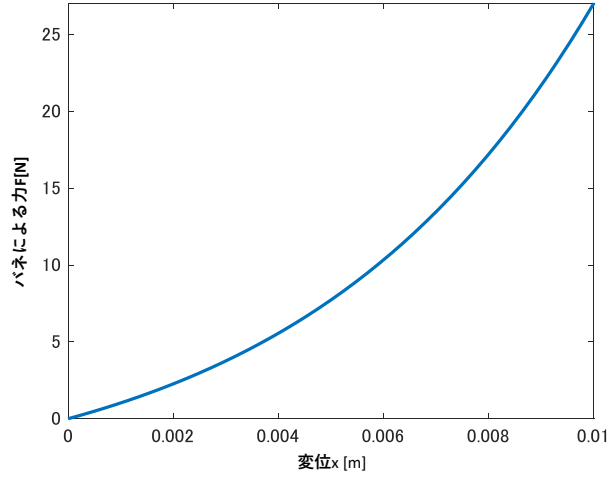


図 5.1: SAT の近似特性

ここで、バネが伸展収縮していない場合 ( $x = 0$ )、バネに加わる力は 0 とするために  $a_3 = -a_1$  とする。そのため、式 (5.2) を式 (5.3) に変換する。

$$\begin{aligned} f_a(x) &\simeq a_1 \left( a_2 x + \frac{a_2^2}{2!} x^2 + \frac{a_2^3}{3!} x^3 + \dots \right) \\ k(x) &= a_1 \left( a_2 + \frac{a_2^2}{2!} x + \frac{a_2^3}{3!} x^2 + \dots \right) \end{aligned} \quad (5.3)$$

このように、位置  $x$  によってバネ定数  $k(x)$  が変化すると考え、制御対象の特性が位置  $x$  によって変化するものとしている。SAT を含むマスバネダンパシステムでは、一般的なマスバネダンパシステムと同様に、質量  $m$  を持つ物体にバネとダンパが壁に接続しているシステムであるが、バネを SAT として考えている。ダンパを  $c$ 、加える力を  $f(t)$  としたときの制御対象の運動方程式をラプラス変換した式を式 (5.4) に示す。

$$\begin{aligned} ms^2 X(s) + csX(s) + k(x)X(s) &= F(s) \\ P(s, x) = \frac{X(s)}{F(s)} &= \frac{1}{ms^2 + cs + k(x)} \end{aligned} \quad (5.4)$$

このとき、 $X(s)$  は  $x(t)$  をラプラス変換したもの、 $F(s)$  は  $f(t)$  をラプラス変換したもの、 $P(s, x)$  は制御対象の伝達関数である。また、バネ定数  $k(x)$  は 3 次の項までテイラー展開したものとし、バネの変位  $x(t)$  は 0mm から 10mm まで変位しているものと仮定している。本稿では、この変位  $x$  を Scheduling 変数としており、制御対象は 0mm から 10mm までの範囲内を 10 点取得した線形なシステムの群として扱う。非線形バネのパラメータは参考文献 [8] を参考にしている。制御対象のパラメータを表 5.1 に示す。制御対象の Bode 線図を図 5.2 に示す。



表 5.1: 制御対象のパラメータ

制御対象	パラメータ
質量 $m$	1[kg]
減衰係数 $c$	0.1[Ns/mm]
$a_1$	5.0
$a_2$	187
$a_3$	-5.0

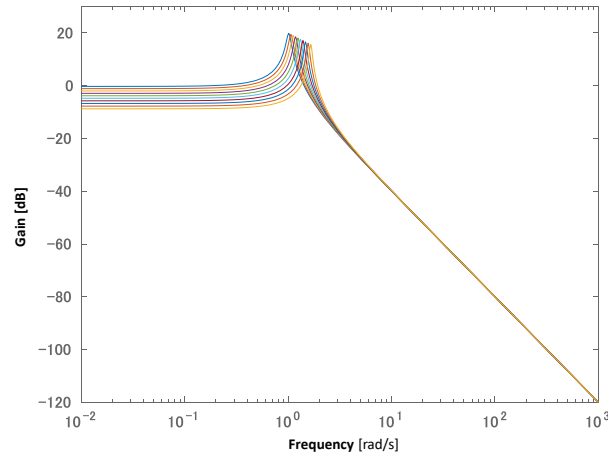


図 5.2: 制御対象のボード線図

### 制御器

制御器構造は  $N_c$  次の Scheduling 変数を持つ PID 制御器を用いている。式 (5.5) に制御器  $C(s, x)$  を示す。

$$C(s, x) = \boldsymbol{\rho}^T(x)\phi(s) \quad (5.5)$$

ただし、制御器パラメータ  $\boldsymbol{\rho}^T(x)$  と制御器構造  $\phi(s)$  は以下のように示される。

$$\boldsymbol{\rho}^T(x) = [\mathbf{K}_p(x), \mathbf{K}_i(x), \mathbf{K}_d(x)] \quad (5.6)$$

$$\phi(s) = [1, \frac{1}{s}, \frac{s}{1 + 0.01s}]^T \quad (5.7)$$

GS 制御器の次数を  $N_c$  次とすると、各制御器パラメータは以下のように示される。

$$\begin{aligned} \mathbf{K}_p(x) &= K_{pN_c}x^{N_c} + \cdots + K_{p1}x + K_{p0} \\ \mathbf{K}_i(x) &= K_{iN_c}x^{N_c} + \cdots + K_{i1}x + K_{i0} \\ \mathbf{K}_d(x) &= K_{dN_c}x^{N_c} + \cdots + K_{d1}x + K_{d0} \end{aligned} \quad (5.8)$$



本研究では, Scheduling 変数  $x$  の次数低減のために  $L_1$  正則化を適用することを考えている。

### ゲイン余裕と位相余裕

制御対象の変動による影響を考慮する場合ではゲイン余裕と位相余裕の確保が必要となる。ゲイン余裕と位相余裕を満たす制約を考えるために, 図 5.3 に開ループ伝達関数の Nyquist 線図を示す。Nyquist 線図上において, 所望の制御帯域の周波数点  $\omega_x$  に対し, 任意の周波数点  $\omega_k$  における開ループ伝達関数が臨界点  $(-1, j0)$  から遠ざかるように制約式を定義する。

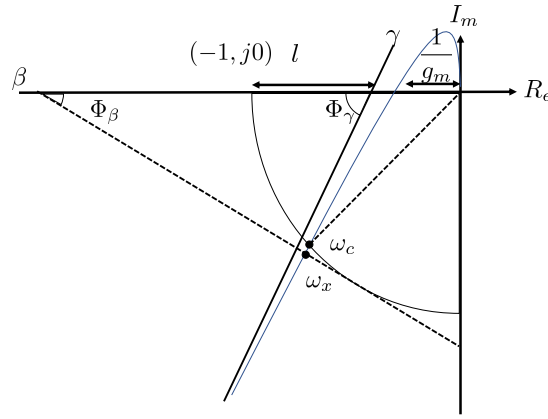


図 5.3: Nyquist 線図上におけるゲイン余裕と位相余裕

まず, 直線  $\alpha$ ,  $\beta$  を以下に定義する。

$$\beta : y = -\tan \Phi_\beta x - \frac{1}{\cos \Phi_\beta} \quad (5.9)$$

$$\gamma : y = \tan \Phi_\gamma x - (l - 1) \tan \Phi_\gamma \quad (5.10)$$

直線  $\beta$  は実軸となす角  $\phi_\beta$  であり, 単位円との接線であると定義し, 直線  $\gamma$  は実軸となす角  $\phi_\gamma$  であり, 実軸上において臨界点  $(-1, j0)$  から距離  $l$  だけ離れた点で交わる直線であると定義した。このとき, 所望のゲイン余裕と位相余裕を得るために, 直線  $\beta$  及び  $\gamma$  を用いて, 任意の周波数  $\omega_k$  における制約式を以下のように定義した。

制約式①  $\omega_k > \omega_x$  のとき, 開ループ伝達関数  $L(s)$  が直線  $\beta$  よりも下に存在する。

制約式②  $\omega_k < \omega_x$  のとき, 開ループ伝達関数  $L(s)$  が直線  $\beta$  よりも上に存在する。

制約式③  $\omega_k < \omega_x$  のとき, 開ループ伝達関数  $L(s)$  が直線  $\gamma$  よりも下に存在する。

上記の制約式を満たすことで、所望のゲイン余裕と位相余裕を満たす制約式を設定できる。

次に、開ループ伝達関数  $L(s)$  を式 (5.11) に定義する。

$$\begin{aligned}
 L(j\omega_k, x_i) &= C(j\omega_k, x_i)P(j\omega_k, x_i) \\
 &= \rho^T(x_i)\phi(j\omega_k)P(j\omega_k, x_i) \\
 &= \rho^T R_e(\omega_k, x_i) + j\rho^T I_m(\omega_k, x_i) \\
 &= (\mathbf{M}\bar{\mathbf{x}}_i)^T R_e(\omega_k, x_i) + j(\mathbf{M}\bar{\mathbf{x}}_i)^T I_m(\omega_k, x_i) \\
 \mathbf{M} &= \begin{bmatrix} K_{pN_c} & \cdots & K_{p1} & K_{p0} \\ K_{iN_c} & \cdots & K_{i1} & K_{i0} \\ K_{dN_c} & \cdots & K_{d1} & K_{d0} \end{bmatrix} \\
 \bar{\mathbf{x}}_i &= [x_i^{N_c} \cdots x_i 1]^T \text{ for } i = 1, \dots, 10
 \end{aligned} \tag{5.11}$$

所望のゲイン余裕と位相余裕を満たすための 3 つの制約式を式 (5.12), (5.13), (5.14) に示す。

$$\begin{aligned}
 (\mathbf{M}\bar{\mathbf{x}}_i)^T [\cos \Phi_\beta I_m(\omega_k) + \sin \Phi_\beta R_e(\omega_k)] &> -1 \\
 \text{for } \omega_k > \omega_x, i &= 1, \dots, 10
 \end{aligned} \tag{5.12}$$

$$\begin{aligned}
 (\mathbf{M}\bar{\mathbf{x}}_i)^T [\cos \Phi_\beta I_m(\omega_k) + \sin \Phi_\beta R_e(\omega_k)] &\leq -1 \\
 \text{for } \omega_k < \omega_x, i &= 1, \dots, 10
 \end{aligned} \tag{5.13}$$

$$\begin{aligned}
 (\mathbf{M}\bar{\mathbf{x}}_i)^T \left[ \frac{I_m(\omega_k)}{\tan \Phi_\gamma} - R_e(\omega_k) \right] + l &\leq 1 \\
 \text{for } \omega_k > \omega_x, i &= 1, \dots, 10
 \end{aligned} \tag{5.14}$$

シミュレーションでは、ゲイン余裕を 5 dB, 位相余裕を 25 deg と定義し、それを満たすようななす角  $\phi_\beta$  と  $\phi_\gamma$ , 距離  $l$  を決定した。

### 最適化問題

最適化問題として開ループ伝達関数と望みの開ループ伝達関数の差の最大値を最小化するミニマックス問題を式 (5.15) で示す。

$$\begin{aligned}
 \min_{\omega} \max |L(j\omega_k, x_i) - L_d(j\omega_k)| \\
 + \lambda \left| \sum_{j=0}^{N_c} |K_{pj}| + \sum_{j=0}^{N_c} |K_{ij}| + \sum_{j=0}^{N_c} |K_{dj}| \right| \\
 \text{subject to (5.12), (5.13), (5.14)}
 \end{aligned} \tag{5.15}$$

ここで, Scheduling 変数  $\mathbf{x}$  のグリッドが 10 個であるため, 開ループ伝達関数  $L(j\omega_k, x_i)$  は 10 個存在する。この最適化問題は, 10 個の開ループ伝達関数  $L(j\omega_k, x_i)$  それぞれに対して望みの開ループ伝達関数  $L_d(j\omega_k)$  に対するミニマックス問題を解き, 評価値の総合値を最小化する最適化問題としている。また, シミュレーションにおいて, 望みの開ループ伝達関数  $L_d(j\omega_k)$  は以下のように決定した。

$$L_d(s) = \frac{30}{s(0.01s + 1)} \quad (5.16)$$

### 正則化パラメータの探索

AIC が正則化パラメータ  $\lambda$  の関数である仮定し, その関数推定と最適解の探索にベイズ最適化を適用する。初期データとして, 探索する正則化パラメータ  $\lambda$  の最小値と最大値を 0.01 と 10 に設定し, その間を対数的に等間隔に 9 点取得したものを初期データとした。ベイズ最適化内で使用される獲得関数は A-LCB[7] を用いて, 獲得関数で使用されるハイパーパラメータ ( $k_{ac}$  とする) の決定に設計者の試行錯誤な作業を必要としないアルゴリズムとなっている。その時,  $k_{ac}$  の初期値は 0.01, 0.1, 1, 10 としている。また, 探索の終了条件として, 事後平均と事後分散の両方が前回の値と有効数字 6 桁で同等の値となった場合に終了する。

シミュレーションの比較として, 制御器設計は  $L_1$  正則化を用いない方法と  $L_1$  正則化を用いた方法で比較し, 評価値と AIC を比較する。正則化パラメータの探索として, 網羅的に探索する GridSearch 法とベイズ最適化を比較して, AIC が最小となる正則化パラメータの探索と計算時間, 評価回数を比較する。ベイズ最適化で使用される獲得関数について, 一般的に使用される LCB と, 近似関数精度を用いた獲得関数 A-LCB を用いた場合で比較を行う。

## 5.2 結果

まず,  $L_1$  正則化を用いずに初期の Scheduling 次数  $N_c$  を変化させた手法と  $L_1$  正則化を用いた手法で評価値と AIC を比較する。 $L_1$  正則化を用いない場合で初期のスケジューリング変数の次数  $N_c$  を変化させたときの評価値と AIC を表 5.2 に,  $L_1$  正則化を用いて, 正則化パラメータを網羅的に探索した結果を表 5.3 に示す。また, 正則化パラメータの探索を網羅的に探索した場合とベイズ最適化を適用した場合の評価回数と計算時間を表 5.4 に示す。

表 5.2 では初期に与える次数  $N_c$  が 2 以上であれば, 評価値が大きく変化しないが,

$N_c = 2$  の時に AIC が最小値となった。 $L_1$  正則化を使用した場合の表 5.3 及び表 5.4 では、 $\lambda$  の最適解が 0.274 となり、AIC の値は  $-21.887$  となった。 $L_1$  正則化を考慮した場合には、考慮していない場合と比較して、評価値は大きく変化せず、AIC は全体的に低い値となった。これは、 $L_1$  正則化を適用することで、冗長なパラメータを削除した GS-PID 制御器を設計していることが読み取れる。図 5.4 と図 5.5 を比較したとき、開ループ伝達関数のボード線図からも  $L_1$  正則化を使用した場合としていない場合の手法による評価値の違いが大きいことが見て取れる。加えて、 $L_1$  正則化を使用しない手法では、初期に与える次数  $N_c$  を変化させるのみ必要となるため、評価回数や計算時間が  $L_1$  正則化を使用した場合と比較して、大きくなることはない。しかし、各 P, I, D 制御器それぞれのスケジューリング変数の次数  $N_c$  は同一にする必要があるため、設計時の自由度が低くなるといった欠点がある。一方、 $L_1$  正則化を使用した場合には、設計後のスケジューリング変数の次数  $N_c(P,I,D)$  が同一である必要がないため、自由度が高い設計が可能となる。

表 5.4 より、網羅的に探索する手法では、探索範囲と刻み幅にもよるが、膨大な評価回数と時間が発生する恐れがある。一方、ベイズ最適化を用いた場合には、少ない評価回数と短い計算時間で探索することが読み取れる。このことから、 $L_1$  正則化を考慮することで、最適化計算でスケジューリング変数の次数を低減した制御器設計が可能であり、正則化パラメータの探索にベイズ最適化を適用することで、設計者の試行錯誤な作業を必要としない設計が可能となる。

表 5.2:  $L_1$  正則化を用いない場合の評価値と AIC

$N_c$	0	1	2	3	...	10
評価値	593.9	310.3	288.5	287.4	...	281.7
AIC	-6.769	-12.542	-15.935	-3.788	...	18.099

表 5.3:  $L_1$  正則化を用いた場合の評価値と AIC( $\lambda$  を網羅的に探索)

$\lambda$	0.01	0.270	0.274	0.280	10
評価値	286.4	292.5	293.3	293.7	603.7
AIC	-14.869	-21.634	-21.887	-21.263	-3.823
$N_c(P,I,D)$	(2,2,2)	(2,1,0)	(2,0,0)	(2,0,0)	(0,0,0)

次に、獲得関数が LCB と A-LCB の場合について比較する。2 つの場合を比較した時の最適解と AIC を表 5.5 に示す。表 5.5 より、A-LCB を用いた場合には、 $k = 0.1$  と

表 5.4:  $\lambda$  の探索における最適解と評価回数, 計算時間の比較

	最適解 $\lambda_{opt}$	評価回数	計算時間 [sec]
Grid Search	0.274	1,000	約 13,000
ベイズ最適化	0.274	210	約 1,150

表 5.5: 獲得関数 LCB, A-LCB の比較

	最適解	AIC	評価回数	計算時間 [sec]
LCB( $k = 0.01$ )	0.274	-21.887	457	約 2,500
LCB( $k = 0.1$ )	0.274	-21.887	396	約 2,200
LCB( $k = 1$ )	0.274	-21.887	221	約 1,200
LCB( $k = 10$ )	0.274	-21.887	288	約 1,570
A-LCB( $k = 0.01$ )	0.274	-21.887	457	約 2,500
A-LCB( $k = 0.1$ )	0.274	-21.887	396	約 2,200
A-LCB( $k = 1$ )	0.274	-21.887	259	約 1,400
A-LCB( $k = 10$ )	0.274	-21.887	210	約 1,150

$k = 0.01$  では初期値が最適化序盤と終盤でほとんど変化せず, 評価回数が増えて探索に時間が費やされた。これは最適化序盤にて未知の区域を探索をする能力があまり発揮できていないと考えられる。また, LCB を用いたとき,  $k = 10$  の場合では未知の区域を探索することが重要視されているため, メモリの限界までベイズ最適化を繰り返しており, A-LCB の場合と比較して評価回数が大きくなった。そのため, A-LCB ではハイパーパラメータ  $k$  の値を比較的大きくすることで, 未知の探索区域から既存データ付近へと探索区域のシフトが上手く言っていると考えられる。また, A-LCB では, ハイパーパラメータ  $k$  の初期値を任意に決定することで, 設計者によるハイパーパラメータの決定を省いたベイズ最適化が可能となることから, ハイパーパラメータを決める煩わしい作業をなくすことが可能である。

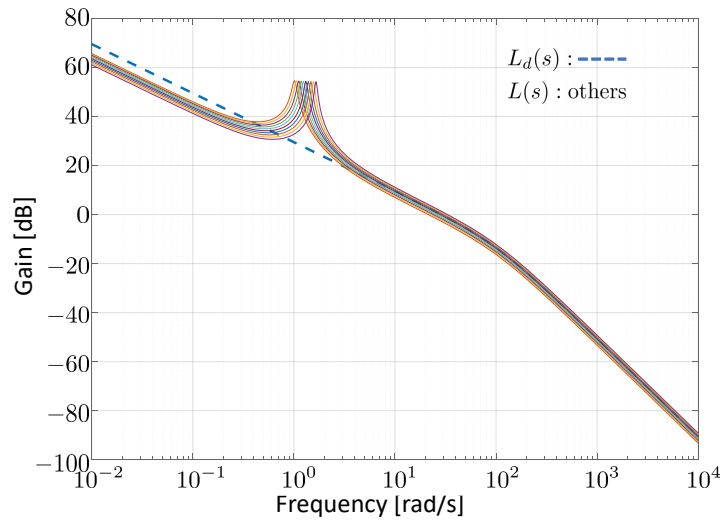


図 5.4: 開ループ伝達関数のボード線図 ( $L_1$  正則化なし,  $N_c = 2$ )

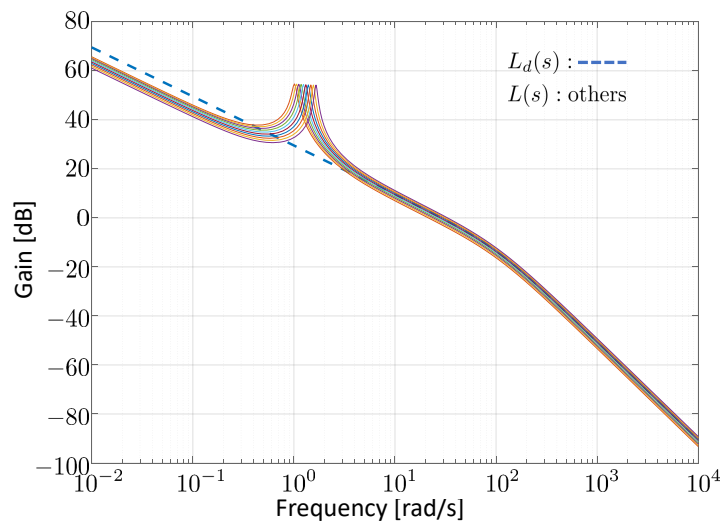


図 5.5: 開ループ伝達関数のボード線図 ( $L_1$  正則化あり,  $\lambda = 0.274$ )

## 第6章 結言

### 6.1 まとめ

本稿では、 $L_1$  正則化を考慮した周波数応答に基づく GS-PID 制御器設計の提案をした。周波数応答に基づく制御器設計では、制御器構造の良し悪しが設計者の熟練度に依存する恐れがあることから、設計した制御器が制御性能に対して冗長的な制御器となってしまう可能性がある。本研究では、 $L_1$  正則化を考慮した評価関数の低減に貢献しないパラメータの自動的な削除を提案しており、制御性能と制御器の低次数化を同時に考慮した制御器設計法を実現できると考えた。また、罰則項の重みである正則化パラメータは設計者が任意に与えるハイパーパラメータであり、この決定に設計者の作業が増える可能性がある。この点は、ベイズ最適化を用いることで、設計者が初期値を与えるだけで自動的に正則化パラメータの探索を行う手法を提案した。ベイズ最適化内で使用される獲得関数内部のハイパーパラメータに関しても、近似関数精度を用いた獲得関数 A-LCB を適用した手法を提案した。これらの手法を用いることで、設計者の試行錯誤的な作業を完全に排除した制御器設計法が可能であると考えた。本稿では、数値シミュレーションから、 $L_1$  正則化を考慮することで、制御性能に貢献しない不要な Scheduling 変数に対応するパラメータの削除を行う結果となった。これにより、設計者にとって煩わしい作業を必要としない手法であり、初期に与える次数やパラメータに依存しない結果を導出することが可能となる。

### 6.2 今後の課題

今後の課題として、複数の罰則項を用いた  $L_1$  正則化による GS-PID 制御器の設計を提案したい。6 章では、GS-PID 制御器の各ゲインを一律に罰則項として扱い、1 つの罰則項を付与した評価関数を用いた最適化を行った。しかし、各 P, I, D ゲインをそれぞれ別々の正則化パラメータを付与し、3 つの罰則項を考えることで、1 つの罰則項を用いた場合よりも、AIC が最小となる正則化パラメータの探索と制御器の設計が可能となると考えている。1 つの罰則項を用いた  $L_1$  正則化を適用した評価関数である式 (5.15) よ

り，複数の罰則項を用いた場合の最適化問題は式 (6.1) のように記述される。

$$\begin{aligned}
 & \min_{\omega} \max |L(j\omega_k, x_i) - L_d(j\omega_k)| \\
 & + \lambda_p \sum_{j=0}^{N_c} |K_{pj}| + \lambda_i \sum_{j=0}^{N_c} |K_{ij}| + \lambda_d \sum_{j=0}^{N_c} |K_{dj}| \\
 & \text{subject to (5.12), (5.13), (5.14)}
 \end{aligned} \tag{6.1}$$

ベイズ最適化は，多変数なモデルに対しても有効な手法であり，複数の正則化パラメータの探索にも適用可能である。

その他の課題として，5 のシミュレーション条件の見直しが必要となる。シミュレーション条件にて，ばねの位置  $x$  を Scheduling 変数と仮定し，それをスケールリングしている。ばねの位置は時間領域で変化するが，本シミュレーションでは，位置  $x$  で変形する伝達関数として扱っているため，再現性が低いシミュレーションとなっている。それに付随して，シミュレーション結果で必要となる GS-PID 制御器の次数が  $N_c = 2$  と低次数だったことも関係している可能性がある。



# 参考文献

- [1] 永原 正章:「スパースモデリング 基礎から動的システムの応用」, コロナ社, (2017)
- [2] 山口高司, 平田光男, 藤本博志:「ナノスケールサーボ制御 高速・高精度に位置を決める技術」, コロナ社, (2007)
- [3] 西口佳孝, 弓場井一裕, 平井淳之 (三重大大学):「Nyquist 線図を用いた望みの制御仕様を保証する固定構造ゲインスケジューリング制御器設計に関する研究」(2012)
- [4] Kunze Marc, Alireza Karimi and Roland Longchamp: “Gain-scheduled controller design by linear programming”, 2007 European Control Conference, pp.5432-5438 , (2007)
- [5] Akaike Hirotugu: “Information theory and an extension of the maximum likelihood principle”, Selected papers of hirotugu akaike, pp.199-213, (1998)
- [6] 中野雄一郎, 藤本悠介, 杉江俊治:「ベイズ最適化を用いた制御器チューニングー提案と実験検証ー」, 計測自動制御学会論文, Vol.55, No.4, pp.269-274, (2019)
- [7] Tianyi Zhang, Shinya Watanabe, Jiayi Han: “ベイズ最適帰化における獲得関数の動的パラメータ調整に関する検討”, 2023 年電気学会電子・情報・システム部門大会, pp.167-172, (2023)
- [8] 灰屋和勇, 駒田諭, 平井淳之:「腱駆動機構における非線形バネの特性式誤差補償による張力制御」, 電気学会論文誌 D (産業応用部門誌), Vol.130, No.6, pp.816-823, (2010)

# 謝辞

本研究の遂行ならびに卒業論文作成にあたって、熱心なご指導とご鞭撻を賜りました、三重大学大学院工学研究科教授 弓場井 一裕 先生に心から感謝の意を表します。また、日頃から公私ともにお世話になった技術職員 中村 勝 氏に心から感謝します。

また、電機システム研究室と制御システム研究室とエネルギーシステム研究室の合同で実施する全体検討会において貴重なご意見を頂きました、同大学教授 駒田 諭 先生，同大学教授 山村 直紀 先生，同大学助教 矢代 大祐 先生，同大学助教 小山 昌人 先生に深く感謝いたします。

本研究を遂行するにあたり，制御理論分野の先輩として，親身にご指導いただきました，伊東 航平 氏，田中 虹輝 氏，中山 尚貴 氏に心から感謝します。また，同じ研究室の同期として協力し，切磋琢磨し合いながら研究を遂行してきた遠藤 健史 君，高井 寛人 君，田中 龍之介 君，渡邊 晃康 君に深く感謝します。また，何かと至らない自分を受け入れ，親身になって交流して頂いた電機システム研究室の皆さまに心から感謝します。

最後に，遠く離れていても学生生活を支援してくださった両親に心から感謝します。

# 論文目録

## 学会発表

- (1) 水谷力也，弓場井一裕，矢代大祐，駒田諭：「L1 正則化を考慮した周波数応答に基づくゲインスケジュールド制御器の設計」，電気学会研究会資料，CT/制御研究会，71-75，(2023.4)
- (2) 水谷力也，弓場井一裕，矢代大祐，駒田諭：「L1 正則化を考慮した周波数応答に基づくゲインスケジュールド制御器設計と AIC による比較検討」，C 部門大会講演論文集，PS2-5，(2024. 7 掲載予定)
- (3) 水谷力也，弓場井一裕，矢代大祐，駒田諭：「L1 正則化を考慮した周波数応答に基づくゲインスケジュールド制御器設計とベイズ最適化による正則化パラメータの探索」，令和 6 年度電気学会全国大会，No.K403-A4，(2024.3)