

平成 25 年度

長岡技術科学大学 大学院
工学研究科 修士論文

擬似参照信号を利用した
アクティブフィードフォワード制御系の
補償器調整

A tuning method for active feedforward control system
using fictitious reference

指導教員 小林 泰秀 准教授

長岡技術科学大学 大学院 工学研究科 機械創造工学専攻

学籍番号 10306988

仲居 洋介

平成 26 年 2 月 7 日

目次

第 1 章	序論	5
1.1	研究の背景	5
1.2	本論文の構成	6
第 2 章	アクティブフィードフォワード制御系	7
2.1	Active Noise Control	7
第 3 章	Filtered-x LMS アルゴリズムによる補償器調整	10
3.1	LMS アルゴリズム	10
3.2	Filtered-x LMS アルゴリズム	12
3.3	フィードバックパスを含むシステムの LMS アルゴリズム	14
第 4 章	実験データに基づく補償器調整アルゴリズム	16
4.1	目標値追従制御系系の補償器調整	16
4.2	Iterative Feedback Tuning(IFT)	18
4.3	Fictitious Reference Iterative Tuning(FRIT)	22
第 5 章	アクティブフィードフォワード制御系への FRIT の拡張	26
5.1	アクティブフィードフォワード制御系の等価変換	26
5.2	規範モデルの扱い	28
5.3	評価関数の定義	30
5.4	擬似参照入力信号の導出	30
5.5	各種データ取得のための実験手順	31
第 6 章	シミュレーションによる実装	35

目次	2
6.1 シミュレーション条件の設定	35
6.2 パラメータ数が二つの場合	37
6.3 パラメータ数が三つの場合	39
第7章 結論	43
7.1 まとめ	43
7.2 課題	44
参考文献	46

目次

2.1	Active noise control system	8
2.2	Closed-loop system	8
2.3	Block diagram of active feedforward control system.	9
3.1	LMS algorithm	11
3.2	Active feedforward system(with Secondary path)	13
3.3	Filtered-x LMS algorithm	14
3.4	Active feedforward system(with Secondary path and Feedback path)	15
3.5	Compensate feedback path	15
4.1	Reference following feedback system	17
4.2	Reference following feedback system and desired model	18
4.3	An experiment to calculate $\frac{\partial e(\rho)}{\partial \rho}$	22
4.4	Error of FRIT	24
5.1	Block diagram of active feedforward control system(Fig.2.3).	26
5.2	Active noise control system($w \rightarrow y, z$)	27
5.3	Block diagram of active feedforward control system(equivalent to Fig.5.1).	27
5.4	Block diagram of active feedforward control system(simplified).	28
5.5	Active feedforward control system and desired model($T_d = 0$)	29
5.6	FRIT for active feedforward control system(Primary path: \bar{G}_{zw})	29
5.7	Experiment (a)	32
5.8	Experiment (b)	33
5.9	Virtual experiment (c)	34

6.1	Disturbance: w	36
6.2	Transition of controller's parameters: ρ (Number of parameters:2)	38
6.3	Error sensor output(Number of parameters:2)	39
6.4	Transition of controller's parameters: ρ (Number of parameters:3)	41
6.5	Error sensor output(Number of parameters:3)	42

第1章

序論

1.1 研究の背景

能動騒音抑制制御に代表されるアクティブフィードフォワード制御系は、制御用のアクチュエータによって外部よりエネルギーを与えることによって騒音や振動などの外乱を相殺する制御系である [1]。

アクティブフィードフォワード制御系の補償器調整の手法には、適応制御アルゴリズムの一種である Filtered-x LMS アルゴリズムなどが従来より用いられている [2]。Filtered-x LMS アルゴリズムは二次経路やフィードバックパスなどの数学的なモデルを予め同定する必要がある。この数学的モデルの同定精度は制御性能との直接的な相関を持たない。さらに、フィードバックパスのモデル化誤差は制御系の安定性に特に強い影響を及ぼすと考えられるため、フィードバックパスのモデルに依存しない手法が望まれる。

プラントモデルに依存しない手法として、実験データに基づくデータ駆動型の補償器調整手法がいくつか提唱されている。これらの手法は実機による実験によって得られた入出力データより制御性能を評価関数によって評価し、最適化問題として補償器のパラメータを調整する手法である。代表的なアルゴリズムの一つとして Iterative Feedback Tuning (IFT) が挙げられる [3]。IFT は補償器パラメータの更新ごとに実機で実験を行う手法である。シンプルなアルゴリズムであるため、アクティブフィードフォワード制御系への応用事例も報告されている [4][5]。しかし IFT は補償器パラメータの更新ごとに実機による実験を繰り返すため、補償器の調整完了まで時間がかかる。

IFT が持つ時間的コストの問題に対する方策の一つとして、Fictitious Reference Iterative Tuning (FRIT) [6] を利用することが挙げられる。FRIT は擬似参照信号と呼ばれる補償器の逆システムから逆算される擬似的な参照入力信号を利用し、補償器のパラメータ調整を実機での実験を繰り返すことなく行うことができるため、大幅な時間的コストの低減が見込める。

しかし、アクティブフィードフォワード制御系の補償器調整に対して IFT を応用した事例は報告されているが、FRIT を応用した事例はまだ報告されていない。これは、規範モデルの設定が目標値追従制御系の調整問題と比較して困難であるためと考えられる。そこで、本研究では FRIT による補償器調整手法をアクティブフィードフォワード制御系に導入することを試み、その有効性をシミュレーションによって実証する。

1.2 本論文の構成

第 2 章では、本研究が対象とするアクティブフィードフォワード制御系の概要について述べる。代表的なアクティブフィードフォワード制御系として、騒音抑制制御系を取り上げる。

第 3 章では、アクティブフィードフォワード制御系の補償器調整法として従来より用いられている LMS アルゴリズムの概要について述べる。LMS アルゴリズムは適応制御アルゴリズムの一種である。Filtered-x LMS アルゴリズムが二次経路とフィードバックパスのモデルを要することを示す。

第 4 章では、プラントモデルに依存しない補償器調整アルゴリズムである Iterative Feedback Tuning(IFT) および Fictitious Reference Iterative Tuning(FRIT) の理論を紹介する。どちらも実機の運転によって得られた入出力データに基づいてオフラインで調整を行う手法である。IFT は補償器パラメータの更新ごとに実機の運転が必要となるが、FRIT は初回に実機を運転するだけでよいので時間的・経済的コストの低減が見込める。

第 5 章では、FRIT をアクティブフィードフォワード制御系に適用するために拡張を行う。一次経路のモデルを規範モデルとすることでアクティブフィードフォワード制御系でも FRIT が利用できることを示す。

第 6 章では、MATLAB/Simulink を用いたシミュレーションを通じて、第 5 章で提案した調整手法が実際に利用できることを実証する。

第 7 章で、本研究をまとめる。

第2章

アクティブフィードフォワード制御系

アクティブフィードフォワード制御系は、第1章で述べたように外乱をアクチュエーションによって能動的に相殺する制御系である。

本章では騒音抑制制御系である Active Noise Control(ANC) を例に、アクティブフィードフォワード制御系の概要を述べる。

2.1 Active Noise Control

アクティブフィードフォワード制御系の応用例として、後述する騒音抑制制御系や構造物の振動抑制制御系 [7] が挙げられる。

以降、代表的なアクティブフィードフォワード制御系の例として、Active Noise Control によるダクト内の騒音抑制システムを考える。騒音抑制システムの概略図を Fig.2.1 に示す。

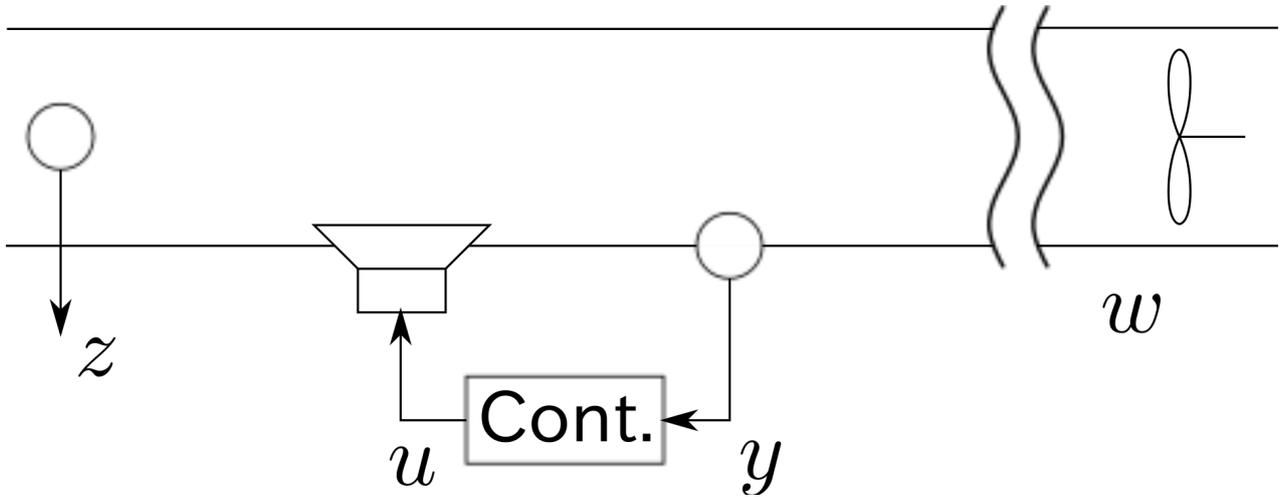


Fig. 2.1 Active noise control system

ここで、 w は外部から入力される未知の外乱を表し、具体的には騒音である。 y と z はそれぞれリファレンスセンサとエラーセンサの観測値、 u は制御用アクチュエータの指令値である。アクティブフィードフォワード制御系は、 z の観測値がなるべく小さくなるよう、 y の観測値に基づいて制御用アクチュエータ u を駆動することを目的とする。この目的を達成するために、 u を駆動する補償器を設計する。

Fig.2.1 のシステムをブロック線図で表現すると、Fig.2.2 となる。ここで、 G はプラント、 K は設計すべき補償器を表す。

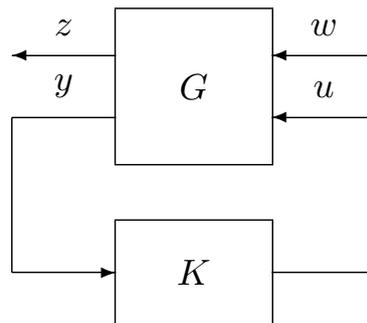


Fig. 2.2 Closed-loop system

Fig.2.2 のプラントを各入出力関係ごとに分解する。

$$\begin{bmatrix} z \\ y \end{bmatrix} = G \begin{bmatrix} w \\ u \end{bmatrix} \tag{2.1}$$

$$= \begin{bmatrix} G_{zw} & G_{zu} \\ G_{yw} & G_{yu} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{2.2}$$

すなわち、

$$G = \begin{bmatrix} G_{zw} & G_{zu} \\ G_{yw} & G_{yu} \end{bmatrix} \quad (2.3)$$

式 (2.3) に基づいてプラント G を各入出力関係ごとに分解した等価なブロック線図を Fig.2.3 に表す。

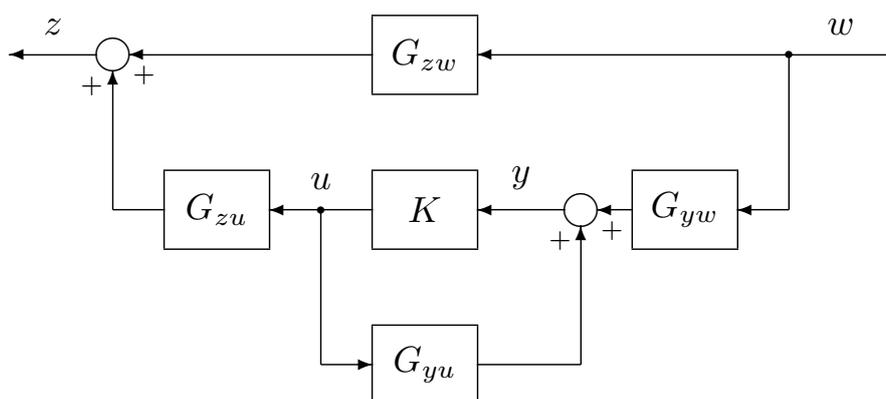


Fig. 2.3 Block diagram of active feedforward control system.

第3章

Filtered-x LMS アルゴリズムによる補償器調整

ANC のシステムを構成する際に用いられるアルゴリズムとして、適応制御アルゴリズムである LMS アルゴリズムが従来より用いられている [2][8][9]。

本章では、LMS アルゴリズムの基本的な理論について、

1. 一次経路のみを扱う LMS アルゴリズム
2. 一次経路、二次経路を考慮した Filtered-x LMS アルゴリズム
3. さらに、補償器まわりにフィードバックパスがある場合の Filtered-x LMS アルゴリズム

と順を追って説明する。

3.1 LMS アルゴリズム

Fig.3.1 に LMS アルゴリズムによる適応制御の例を示す。 P はプラントであり一次経路と呼ばれる。また、 K は補償器、 x はリファレンス信号、 e はエラー信号を示す。

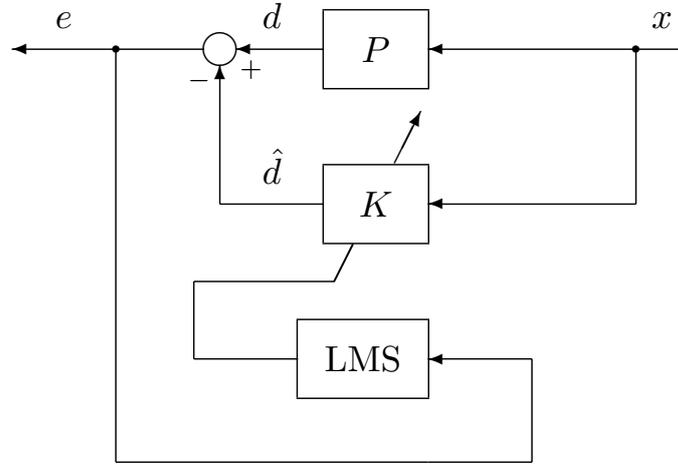


Fig. 3.1 LMS algorithm

Fig.3.1 より、

$$e = (P - K) x \tag{3.1}$$

$$= d - \hat{d} \tag{3.2}$$

である。LMS アルゴリズムでは補償器として FIR フィルタを用いる。タップ数を n で表したとき補償器は次式で表される。

$$K(z^{-1}, \rho) = \rho(0) + \rho(1)z^{-1} + \rho(2)z^{-2} + \dots + \rho(n)z^{-n} \tag{3.3}$$

なお、

$$\rho = \begin{bmatrix} \rho(0) \\ \rho(1) \\ \vdots \\ \rho(n) \end{bmatrix} \tag{3.4}$$

は補償器のパラメータを表す。よって、時刻 i における補償器 K の出力 \hat{d} は

$$\hat{d}(\rho)_i = \rho(0)x_i + \rho(1)x_{i-1} + \dots + \rho(n)x_{i-n} \tag{3.5}$$

と求められる。

LMS アルゴリズムは e を最小化することを目指し、次の式 (3.6) に示す評価関数を最小化するように補償器を調整する。なお $E\{\cdot\}$ は \cdot の期待値を表す。

$$J(\rho) = E\{e(\rho)^2\} \tag{3.6}$$

$x \neq 0$ として、式 (3.6) を最小化するということは $(P - K)$ を最小化することである。つまり、LMS アルゴリズムは補償器をプラントに近似させるアルゴリズムである。

LMS アルゴリズムは式 (3.6) を最小とするように補償器のパラメータを最急降下法によって更新する。最急降下法によるパラメータ ρ の更新式は、

$$\rho^{i+1}(l) = \rho^i(l) - c \text{grad}J(\rho^i) \quad (3.7)$$

である。ここで、 c はステップサイズを表す定数係数である。しかし、式 (3.6) で定義した通り $J(\rho)$ は期待値であるから、 $\text{grad}J(\rho)$ の計算には期待値計算が含まれる。期待値計算は多量の計算量を要し、特にマイコンなどの計算資源に乏しいシステムへ実装する場合には非現実的である。そこで、LMS アルゴリズムでは瞬時値をもって評価関数の勾配 $\text{grad}J(\rho^i)$ を近似する [9]。 $\text{grad}J(\rho^i)$ は、

$$\text{grad}J(\rho^i) \approx \frac{\partial J(\rho)_i}{\partial \rho} \quad (3.8)$$

$$= \frac{\partial}{\partial \rho} [e(\rho)_i^2] \quad (3.9)$$

$$= 2e(\rho)_i \frac{\partial e(\rho)_i}{\partial \rho} \quad (3.10)$$

式 (3.2) より、

$$= 2e(\rho)_i \frac{\partial \hat{d}(\rho)_i}{\partial \rho} \quad (3.11)$$

$$= 2e(\rho)_i \frac{\partial}{\partial \rho} [\rho(0)x_i + \rho(1)x_{i-1} + \cdots + \rho(n)x_{i-n}] \quad (3.12)$$

$$= -2e(\rho)_i \mathbf{x}_i \quad (3.13)$$

と近似される。ここで、時刻 i における評価値を $J(\rho)_i$ 、 $e(\rho)$ を $e(\rho)_i$ と表す。また、 \mathbf{x}_i は

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ x_{i-1} \\ \vdots \\ x_{i-n} \end{bmatrix} \quad (3.14)$$

なるベクトルである。

式 (3.13) を式 (3.7) に代入して、

$$\rho^{i+1}(l) = \rho^i(l) + \mu e_n x_{i-l} \quad (3.15)$$

ここで、 $\mu = 2c$ とまとめる。式 (3.15) は「時刻 i の情報を元に、 l 番目の補償器パラメータを更新する式」である。

3.2 Filtered-x LMS アルゴリズム

Filtered-x LMS アルゴリズムが想定するシステムを Fig.3.2 に示す。ここで S は制御音源から消音点までの特性を表すもので二次経路と呼ばれる。 $S = 1$ のとき、前節で取り上げた LMS アルゴリズムが応

用できる。しかし、二次経路の特性を無視できない場合、例えば制御音源からエラーマイクまでの距離が無視できないほど離れている場合には S の影響を陽に考慮しなければならない。そこで、二次経路の特性を適応則に組み込んだ Filtered-x LMS アルゴリズムが提案されている。

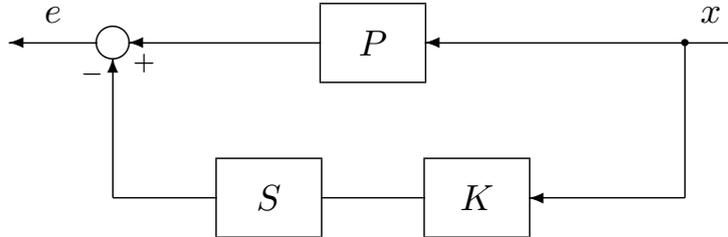


Fig. 3.2 Active feedforward system(with Secondary path)

Fig.3.2 のシステムにおいて、 e は

$$e = (P - SK)x \quad (3.16)$$

と表記される。

前節で示した LMS アルゴリズムと同様に、最急降下法によって補償器パラメータを更新する。このために必要な $\frac{\partial e(\rho)}{\partial \rho}$ は

$$\frac{\partial e(\rho)}{\partial \rho} = \frac{\partial}{\partial \rho} [(P - SK(\rho))x] \quad (3.17)$$

$$= \frac{\partial}{\partial \rho} [-SK(\rho)x] \quad (3.18)$$

と求められるが、この計算は S の情報に陽に依存する。そこであらかじめ同定を行うなどして作成した二次経路モデル \hat{S} を用いて、 x の値にフィルタリングを施して Sx を計算する。このことから、このアルゴリズムは “Filtered-x” と呼ばれる。 \hat{x} は

$$\hat{x} = \hat{S}x \quad (3.19)$$

である。 $\hat{S} = S$ のとき、式 (3.17) に代入して、

$$\frac{\partial e(\rho)}{\partial \rho} = \frac{\partial}{\partial \rho} [-\hat{S}K(\rho)x] \quad (3.20)$$

$$= \frac{\partial}{\partial \rho} [-K(\rho)\hat{x}] \quad (3.21)$$

$$= -\frac{\partial K(\rho)}{\partial \rho}\hat{x} \quad (3.22)$$

Filtered-x LMS アルゴリズムによる適応制御の例を Fig.3.3 に表す。

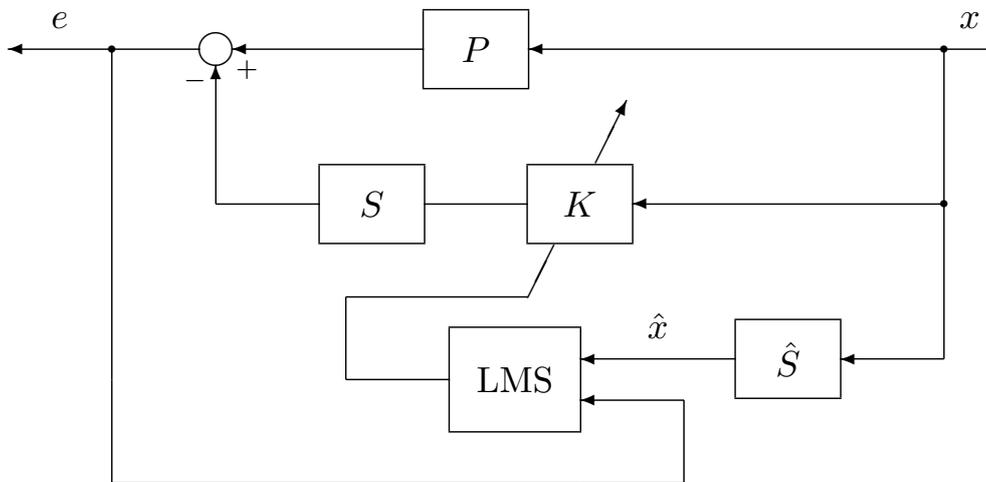


Fig. 3.3 Filtered-x LMS algorithm

grad $J(\rho)$ は LMS アルゴリズム

$$\text{grad}J(\rho^i) \approx \frac{\partial J(\rho)_i}{\partial \rho} \tag{3.23}$$

$$= 2e(\rho)_i \frac{\partial e(\rho)_i}{\partial \rho} \tag{3.24}$$

式 (3.16) より、

$$= 2e(\rho)_i \frac{\partial}{\partial \rho} [\rho(0)\hat{x}_i + \rho(1)\hat{x}_{i-1} + \dots + \rho(n)\hat{x}_{i-n}] \tag{3.25}$$

$$= -2e(\rho)_i \hat{x}_i \tag{3.26}$$

\hat{x} を用いて、補償器パラメータ ρ の更新式は、

$$\rho^{i+1}(l) = \rho^i(l) + \mu e_n \hat{x}_{i-l} \tag{3.27}$$

となる。

3.3 フィードバックパスを含むシステムの LMS アルゴリズム

Fig.3.2 のシステムに、さらにフィードバックパス F を考慮する。これを Fig.3.4 に表す。フィードバックパスは「制御用アクチュエータがリファレンスセンサに及ぼす影響」を表すものであり、システムの安定性に影響を及ぼすと考えられる。

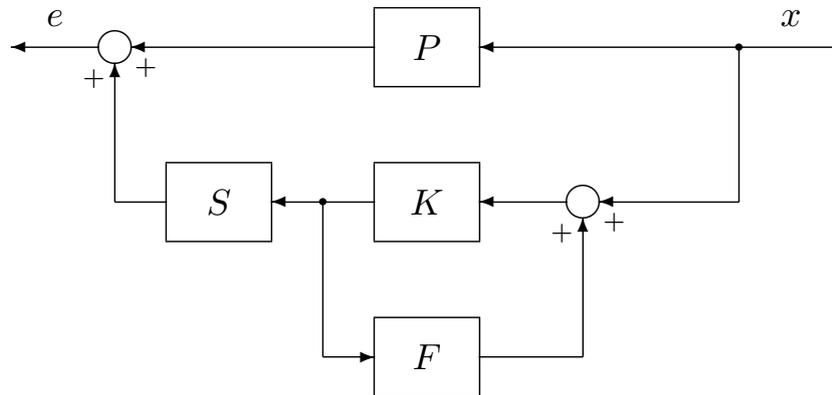


Fig. 3.4 Active feedforward system(with Secondary path and Feedback path)

文献 [2] では、Fig.3.4 の場合フィードバックパスのモデルを作成し、この影響を相殺するようにコントローラへさらにローカルな閉ループを作り、Filtered-x LMS アルゴリズムを使用している。Fig.3.5 にこの方策を示す。ここで、フィードバックパスのモデルを \hat{F} と表す。

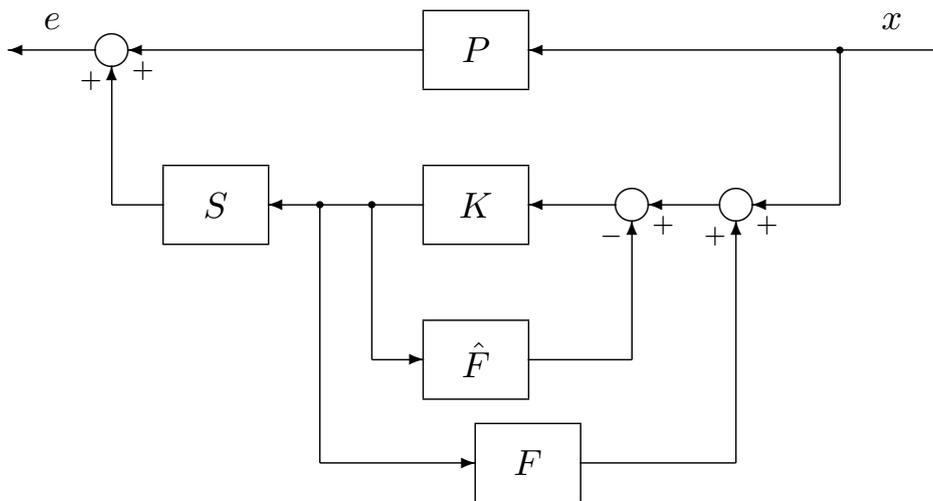


Fig. 3.5 Compensate feedback path

この方策をとる場合、フィードバックパスをモデルで補償するため、フィードバックパスのモデル化誤差がシステムの安定性に影響を及ぼす可能性が考えられる。文献 [2] では影響は小さいものとして無視しているものの、いつでも良好な同定精度が得られるとは限らない。

第4章

実験データに基づく補償器調整アルゴリズム

プラントモデルに基づく補償器設計法では、補償器設計に先立ってプラントのシステム同定を行うことがある。システム同定によってプラントモデルを取得する場合、同定を行うためのデータ取得に別途時間を要する。また、プラントモデルに基づく方法は第3章で言及したように、本質的にプラントのモデル化精度による影響も含み、そのシステムの安定性や性能に影響を及ぼす可能性が否定できない。

これらの手法に対して、プラントモデルに依存しない補償器の設計法が提案されている。これは、システムの入出力データに基づいた評価を行い、所望の特性を得るものである。

本章では目標値追従制御系の補償器調整アルゴリズムである Iterative Feedback Tuning[3] と、本研究の土台となる Fictitious Reference Iterative Tuning[6] の概要について解説する。これら二つのアルゴリズムは、実機による実験によって得られた入出力データを利用し、最適化問題として補償器のパラメータを調整する手法である。これらの手法はプラントモデルに依存しない。

4.1 目標値追従制御系の補償器調整

Iterative Feedback Tuning と Fictitious Reference Iterative Tuning はいずれも Fig.4.1 のような目標値追従制御系の補償器調整を想定している [3][6]。Fig.4.1 の $C(\rho)$ は補償器、 G はプラントを表し、い

ずれも SISO であるとする。 ρ は補償器の可調整パラメータであり、

$$\rho = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(n) \end{bmatrix} \quad (4.1)$$

と定義され、補償器の l 番目のパラメータを $\rho(l)$ と表わす。両アルゴリズムでは補償器の構造は決定されているものとし、補償器の可調整パラメータ ρ を調整する。例えば連続時間における PID 補償器は ρ を用いて

$$C(s, \rho) = \rho(1) + \rho(2) \frac{1}{s} + \rho(3) s \quad (4.2)$$

と表記される。また、 r は $y(\rho)$ の目標値である。Fig.4.1 より、 u, y はそれぞれ、

$$y(\rho) = Gu(\rho) \quad (4.3)$$

$$u(\rho) = C(\rho)(r - y(\rho)) \quad (4.4)$$

である。

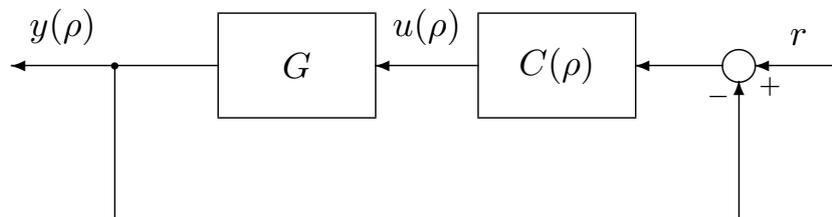


Fig. 4.1 Reference following feedback system

後述するアルゴリズムは Fig.4.1 に示した閉ループ系の挙動をユーザが定義した規範モデル T_d に近似させることを目標とする。理想的な応答 y_d を

$$y_d = T_d r \quad (4.5)$$

とにおいて、実システムと規範モデルの誤差を y_d と実際のシステムの出力 y との差分によって表現する。調整対象の閉ループ系を

$$T(\rho) = \frac{GC(\rho)}{1 + GC(\rho)} \quad (4.6)$$

としたとき、実システムと規範モデルの応答の差分 $e(\rho)$ は

$$e(\rho) = y(\rho) - y_d \quad (4.7)$$

$$= T(\rho)r - T_d r \quad (4.8)$$

$$= (T(\rho) - T_d)r \quad (4.9)$$

と表される。 $e(\rho)$ が十分小さいとき、 $T(\rho)$ は T_d に近似している。式 (4.9) をブロック線図で表すと、Fig.4.2 となる。

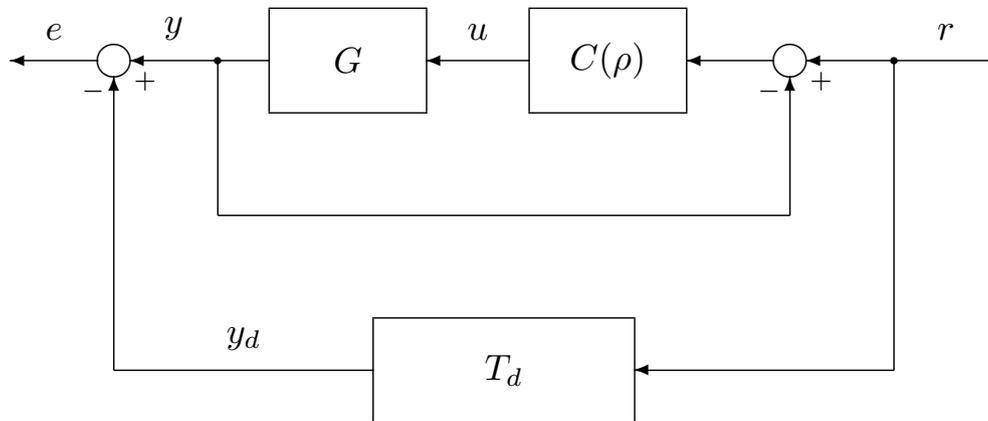


Fig. 4.2 Reference following feedback system and desired model

4.2 Iterative Feedback Tuning(IFT)

Iterative Feedback Tuning(IFT) は実機を運転して得られた入出力データに基づいてオフラインで補償器を調整するアルゴリズムである [10][3]。

シンプルなアルゴリズムであるため、アクティブフィードフォワード制御系への応用例も報告されている [5]。

4.2.1 問題設定

IFT は $e(\rho)$ を最小化する ρ を求めることを最終的な目的とする。文献 [3] では、評価関数を

$$J(\rho) = \|e(t, \rho)\|_2^2 \quad (4.10)$$

と設定する。式 (4.10) に示した評価関数の評価値を最小化するようなパラメータ ρ^*

$$\rho^* := \arg \min_{\rho} J(\rho) \quad (4.11)$$

を求める最適化問題を解くことで、補償器のパラメータチューニングを行う。

非線形最適化問題の最適化アルゴリズムは数多く存在するが [11]、文献 [3] では勾配法の一つである Gauss-Newton 法が用いられている。勾配を用いる最適化アルゴリズムには Gauss-Newton 法の他に、LMS アルゴリズムでも用いられている最急降下法が挙げられる。最急降下法は評価関数の形状によっ

ては収束に時間がかかることがある。また、勾配によらない最適化アルゴリズムの一例として Particle swarm optimization(PSO)[12] が挙げられる。PSO は勾配法と比較して local minimum へ解が落ち込む可能性は低いものの、Particle の更新に確率的な手法をとるため今回提案する調整手法の定性的な評価がし難い。そこで、本研究では Gauss-Newton 法を採用することとした。

勾配法は評価関数の勾配が 0 となる時、 $J(\rho)$ は極小となることから、

$$\frac{\partial J(\rho)}{\partial \rho} = 0 \quad (4.12)$$

となるような ρ を探索する。ここで、評価関数は単純な凸関数である保証は無いため、上式で求めた解が local minimum である可能性もあるが、ここでは得られた解を最適解として扱う。

いま、 i 回目の計算によって得られた補償器のパラメータを ρ^i と表現する。IFT で ρ^{i+1} に更新する計算式は、Gauss-Newton 法より次の式 (4.13) となる。

$$\rho^{i+1} = \rho^i - \gamma \left\{ \frac{\partial^2 J(\rho)}{\partial \rho \partial \rho^T} \Big|_{\rho=\rho^i} \right\}^{-1} \frac{\partial J(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} \quad (4.13)$$

γ はステップサイズを決定づけるパラメータであり、一回の更新ごとに ρ をどれくらい変化させるか調整するものである。また、 $\frac{\partial J(\rho)}{\partial \rho} \Big|_{\rho=\rho^i}$ は ρ^i における $J(\rho)$ の勾配である。 $\frac{\partial^2 J(\rho)}{\partial \rho \partial \rho^T} \Big|_{\rho=\rho^i}$ は Hessian であり、本来

$$\frac{\partial^2 J(\rho)}{\partial \rho \partial \rho^T} \Big|_{\rho=\rho^i} = \begin{bmatrix} \frac{\partial^2 J(\rho)}{\partial \rho(1)^2} \Big|_{\rho=\rho^i} & \cdots & \frac{\partial^2 J(\rho)}{\partial \rho(1) \partial \rho(n)} \Big|_{\rho=\rho^i} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J(\rho)}{\partial \rho(n) \partial \rho(1)} \Big|_{\rho=\rho^i} & \cdots & \frac{\partial^2 J(\rho)}{\partial \rho(n)^2} \Big|_{\rho=\rho^i} \end{bmatrix} \quad (4.14)$$

であるが、計算が複雑となるため、

$$\frac{\partial^2 J(\rho)}{\partial \rho \partial \rho^T} \Big|_{\rho=\rho^i} \approx \left(\frac{\partial J(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} \right)^T \left(\frac{\partial J(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} \right) \quad (4.15)$$

で近似的に求める。

ところで、多くの場合ではデジタルなコンピュータを用いて実験を行うため、補償器の実装や測定される入出力データは離散時間で考える必要がある。したがって、実験によって得られるデータ r や u , y はベクトルで表現される。たとえば、データ点数を N 点とすると、 r は、

$$r_{[1,N]} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \quad (4.16)$$

と定義する。 $e(\rho)$ については、

$$e_{[1,N]}(\rho) = T(q, \rho)r_{[1,N]} - T_d(q)r_{[1,N]} \quad (4.17)$$

と表される (q はすすみオペレータ)。実際に計算機上でパラメータチューニングを行う際には、式 (4.17) を用いて評価関数を次のように扱う。

$$\begin{aligned} J(\rho) &= \|e_{[1,N]}(\rho)\|_2^2 \\ &= \sum_{t=1}^N [e(t, \rho)]^2 \end{aligned} \quad (4.18)$$

4.2.2 チューニングの手順

式 (4.13) で ρ を更新する際に必要な情報は、

- 更新する一回前のパラメータ: ρ^i
- 評価関数の勾配: $\frac{\partial J(\rho)}{\partial \rho}$

である。

式 (4.18) より評価関数の勾配は、

$$\left. \frac{\partial J(\rho)}{\partial \rho} \right|_{\rho=\rho^i} = \sum_{t=1}^N \left. \frac{\partial}{\partial \rho} [(e(t, \rho))^2] \right|_{\rho=\rho^i} \quad (4.19)$$

$$= 2 \sum_{t=1}^N \left[e(t, \rho^i) \left. \frac{\partial e(t, \rho)}{\partial \rho} \right|_{\rho=\rho^i} \right] \quad (4.20)$$

となる。これを求めるためには、

- $e(\rho^i)$
- $\left. \frac{\partial e(\rho)}{\partial \rho} \right|_{\rho=\rho^i}$

を求めればよい。

$\frac{\partial e(\rho)}{\partial \rho}$ は、

$$\frac{\partial e(\rho)}{\partial \rho} = \frac{\partial}{\partial \rho} (y(\rho) - y_d) \quad (4.21)$$

$$= \frac{\partial y(\rho)}{\partial \rho} \quad (4.22)$$

より、 $\frac{\partial y(\rho)}{\partial \rho}$ を求めることで得られる。

$\frac{\partial y(\rho)}{\partial \rho}$ は

$$\frac{\partial y(\rho)}{\partial \rho} = \frac{\partial}{\partial \rho} [Gu(\rho)] \quad (4.23)$$

$$= G \frac{\partial u(\rho)}{\partial \rho} \quad (4.24)$$

と求められる。 $\frac{\partial u(\rho)}{\partial \rho}$ については

$$\frac{\partial u(\rho)}{\partial \rho} = \frac{\partial}{\partial \rho} [C(\rho)(r - y(\rho))] \quad (4.25)$$

$$= \frac{\partial C(\rho)}{\partial \rho} (r - y(\rho)) - C(\rho) \frac{\partial y(\rho)}{\partial \rho} \quad (4.26)$$

と求められる。式 (4.24) と式 (4.26) より、

$$\frac{\partial y(\rho)}{\partial \rho} = G \frac{\partial u(\rho)}{\partial \rho} \quad (4.27)$$

$$\frac{\partial y(\rho)}{\partial \rho} = G \left[\frac{\partial C}{\partial \rho} (r - y(\rho)) - C(\rho) \frac{\partial y(\rho)}{\partial \rho} \right] \quad (4.28)$$

$$(1 + GC(\rho)) \frac{\partial y(\rho)}{\partial \rho} = G \frac{\partial C}{\partial \rho} (r - y(\rho)) \quad (4.29)$$

$$\frac{\partial y(\rho)}{\partial \rho} = \frac{G}{1 + GC(\rho)} \frac{\partial C}{\partial \rho} (r - y(\rho)) \quad (4.30)$$

と求めることができる。

これらの値を取得・算出するために、パラメータ更新の度に実験を行う。補償器パラメータ更新一回につき、実験は二種類行われる。

4.2.2.1 通常の実験 (一つ目)

まず、Fig.4.1 に実機での使用が想定される参照信号 $r_1 = r$ を入力し、 y をデータとして取得する。更新プロセスの一つ目の実験であることから、 y を y_1 と呼ぶことにする。

$$S(\rho^i) = \frac{1}{1 + GC(\rho^i)} \quad (4.31)$$

として、 $y_1(\rho^i)$ は

$$y_1(\rho^i) = GC(\rho^i)S(\rho^i)r \quad (4.32)$$

である。

4.2.2.2 勾配導出のための実験 (二つ目)

次に、参照信号として $r_2(\rho^i) = r_1 - y_1(\rho^i)$ を与え、 $y_2(\rho^i)$ なるデータを取得する。すなわち、

$$y_2(\rho^i) = GC(\rho^i)S(\rho^i)r_2(\rho^i) \tag{4.33}$$

なるデータが得られる。式 (4.33) の $y_2(\rho^i)$ についてさらに、 $\frac{\partial C(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} C(\rho^i)^{-1}$ を施して、 $\frac{\partial y(\rho)}{\partial \rho} \Big|_{\rho=\rho^i}$ を導出する。

$$\begin{aligned} \frac{\partial C(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} C(\rho)^{-1} y_2(\rho) &= \frac{\partial C(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} C(\rho^i)^{-1} [GC(\rho^i)S(\rho^i)r_2(\rho^i)] \\ &= \frac{\partial C(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} GS(\rho^i)r_2(\rho^i) \end{aligned} \tag{4.34}$$

式 (4.34) は式 (4.30) と等しいデータを二回の実験で求められることを示している。すなわち、

$$\frac{\partial e(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} = \frac{\partial C(\rho)}{\partial \rho} \Big|_{\rho=\rho^i} C(\rho^i)^{-1} y_2(\rho^i) \tag{4.35}$$

この導出過程には、プラント G の情報は陽に含まれない。

二回目の実験と計算をブロック線図にまとめると、Fig.4.3 になる。

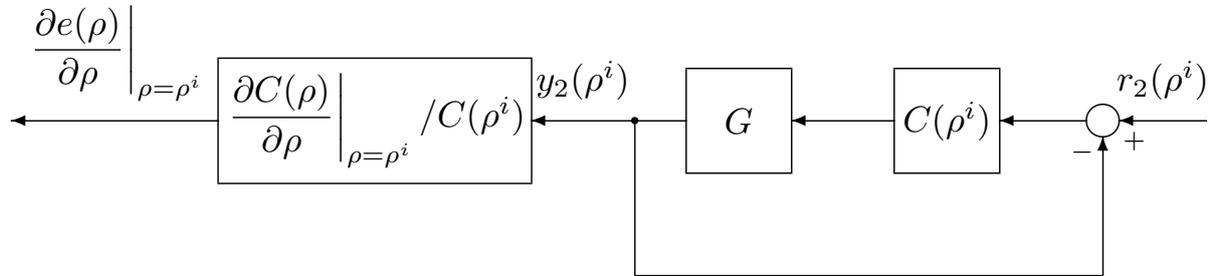


Fig. 4.3 An experiment to calculate $\frac{\partial e(\rho)}{\partial \rho}$

4.3 Fictitious Reference Iterative Tuning(FRIT)

IFT では補償器のパラメータ更新ごとに実機を運転し、入出力データを取得する必要がある。このため、多大な時間的・経済的コストを要する。これに対する方策として、Fictitious Reference Iterative Tuning(FRIT)[6] が提案されており、現用のシステムに応用した事例がいくつか報告されている [13]。

FRIT は「擬似参照信号 (Fictitious Reference Signal)」を利用することで、パラメータ更新ごとの実機の運転を必要としない。これにより、大幅なコストの削減が見込める。

IFT と同様、Fig.4.1 に示すシステムを想定する。

4.3.1 擬似参照信号

FRIT は補償器パラメータの変更ごとに擬似参照信号を更新し、これに基づいて評価関数の勾配を算出する。擬似参照信号は初回の実験で得られた入出力データを再現する、 ρ に依存した擬似的な参照信号である。

初回の実験で得られた制御入力を u_0 、閉ループ系の応答を y_0 と表す。これらはそれぞれ、

$$y_0 = Gu_0 \quad (4.36)$$

$$u_0 = C(\rho^0)(r - y_0) \quad (4.37)$$

と与えられる。

式 (4.4) を r について解いて、 u_0 を再現するような擬似的な参照信号 \tilde{r} を考える。擬似参照信号は次の式 (4.38) となる。

$$\tilde{r}(\rho) = C(\rho)^{-1}u_0 + y_0 \quad (4.38)$$

以下に擬似参照信号が初回の実験で取得した入出力データ u_0 と y_0 をそれぞれ再現することを示す [6]。

証明 4.3.1.

$$y(\rho) = T(\rho)\tilde{r}(\rho) \quad (4.39)$$

式 (4.38) を代入して

$$= \frac{GC(\rho)}{1 + GC(\rho)} (C(\rho)^{-1}u_0 + y_0) \quad (4.40)$$

$$= \frac{Gu_0 + GC(\rho)y_0}{1 + GC(\rho)} \quad (4.41)$$

$y_0 = Gu_0$ より

$$= \frac{y_0 + GC(\rho)y_0}{1 + GC(\rho)} \quad (4.42)$$

$$= \frac{1 + GC(\rho)}{1 + GC(\rho)} y_0 \quad (4.43)$$

$$= y_0 \quad (4.44)$$

証明 4.3.2.

$$u(\rho) = C(\rho) (\tilde{r} - y_0) \tag{4.45}$$

式 (4.38) を代入して

$$= C(\rho) (C(\rho)^{-1}u_0 + y_0 - y_0) \tag{4.46}$$

$$= C(\rho)C(\rho)^{-1}u_0 \tag{4.47}$$

$$= u_0 \tag{4.48}$$

4.3.2 パラメータの更新

\tilde{r} を利用した実システムと規範モデルの応答の差分をとる。

$$\tilde{e}(\rho) = T(\rho)\tilde{r}(\rho) - T_d\tilde{r}(\rho) \tag{4.49}$$

$$= y_0 - T_d\tilde{r}(\rho) \tag{4.50}$$

擬似参照信号 \tilde{r} の算出と式 (4.50) をブロック線図にまとめると、Fig.4.4 となる。これは計算機上で完結するため、実機の運転を必要としない。

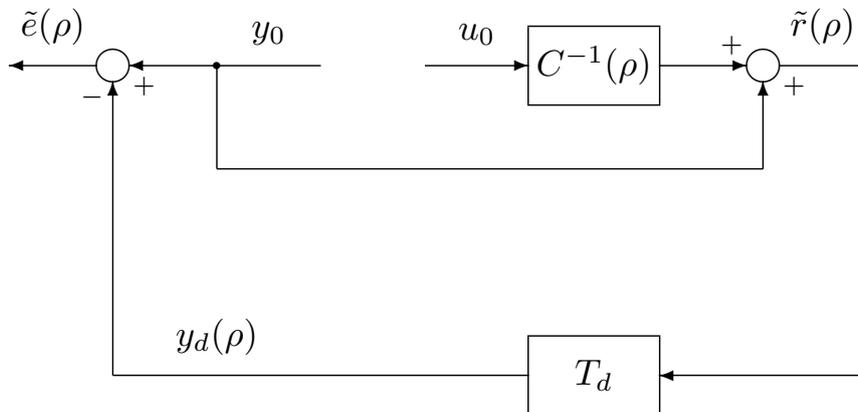


Fig. 4.4 Error of FRIT

評価関数は次の式 (4.51) の通りとする。

$$J(\rho) = \|\tilde{e}(\rho)\|_2^2 \tag{4.51}$$

IFT と同様に Gauss-Newton 法によってパラメータの更新を行う。評価関数の勾配は、

$$\frac{\partial J(\rho)}{\partial \rho} = \sum_{t=1}^N \frac{\partial}{\partial \rho} [(\tilde{e}(t, \rho))^2] \Big|_{\rho=\rho^i} \quad (4.52)$$

$$= 2 \sum_{t=1}^N \left[\tilde{e}(t, \rho^i) \frac{\partial \tilde{e}(t, \rho)}{\partial \rho} \Big|_{\rho=\rho^i} \right] \quad (4.53)$$

評価関数の勾配 $\frac{\partial J(\rho)}{\partial \rho}$ を求めるために $\frac{\partial \tilde{e}(\rho)}{\partial \rho}$ を求める。

$$\frac{\partial \tilde{e}(\rho)}{\partial \rho} = \frac{\partial}{\partial \rho} [y_0 - T_d \tilde{r}(\rho)] \quad (4.54)$$

$$= \frac{\partial}{\partial \rho} [y_0 - T_d (C(\rho)^{-1} u_0 + y_0)] \quad (4.55)$$

$$= \frac{\partial}{\partial \rho} [-T_d C(\rho)^{-1} u_0] \quad (4.56)$$

$$= T_d \frac{1}{C(\rho)^2} \frac{\partial C(\rho)}{\partial \rho} u_0 \quad (4.57)$$

あとは IFT と同様に最適化問題として ρ を更新し、チューニングを行う。

第 5 章

アクティブフィードフォワード制御系への FRIT の拡張

第 4 章で述べた FRIT は Fig.4.1 のような目標値追従制御系を想定しており、Fig.2.3 のアクティブフィードフォワード制御系へは直接適用できない。このため、FRIT をアクティブフィードフォワード制御系へ応用する場合、いくつか条件を変更する必要がある。

本章では FRIT をアクティブフィードフォワード制御系へ応用するための拡張を行う。

5.1 アクティブフィードフォワード制御系の等価変換

第 2 章で、アクティブフィードフォワード制御系が Fig.5.1 で表されるシステムであることを示した。

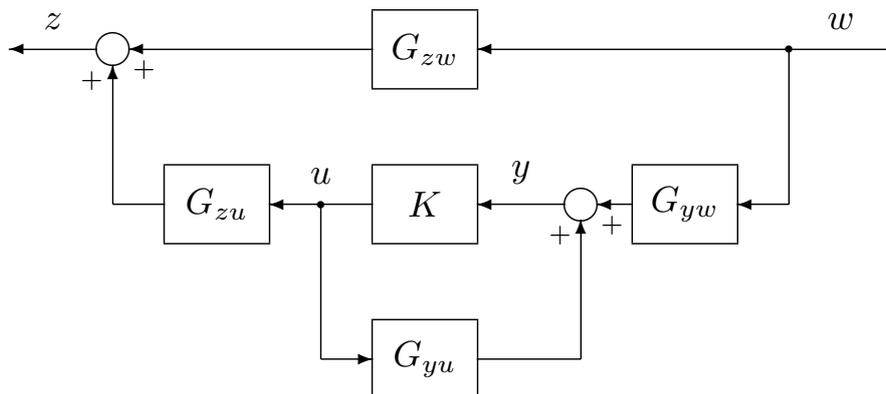


Fig. 5.1 Block diagram of active feedforward control system(Fig.2.3).

Fig.5.1 のシステムに適用するに当たって、Fig.5.2 の通り外乱源からリファレンスセンサまでの特性 G_{yw} についてを考慮しなければならない。ここで、外乱の影響についてリファレンスセンサより下流の

みを考慮することとする。Fig.5.2の通り、 G_{yw} は G_{zw} にも物理的に含まれるため、同時にまとめることができる。

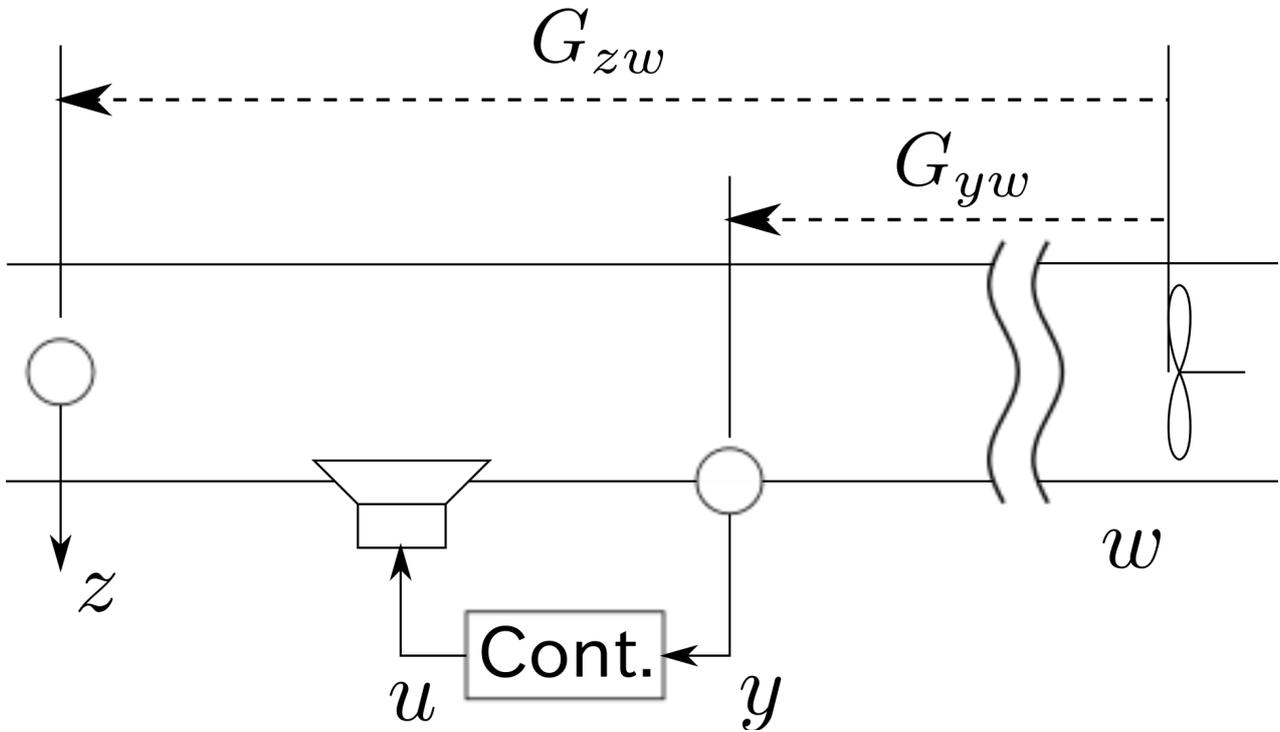


Fig. 5.2 Active noise control system($w \rightarrow y, z$)

以上より、Fig.5.1について以下の等価変換を行うことができる。

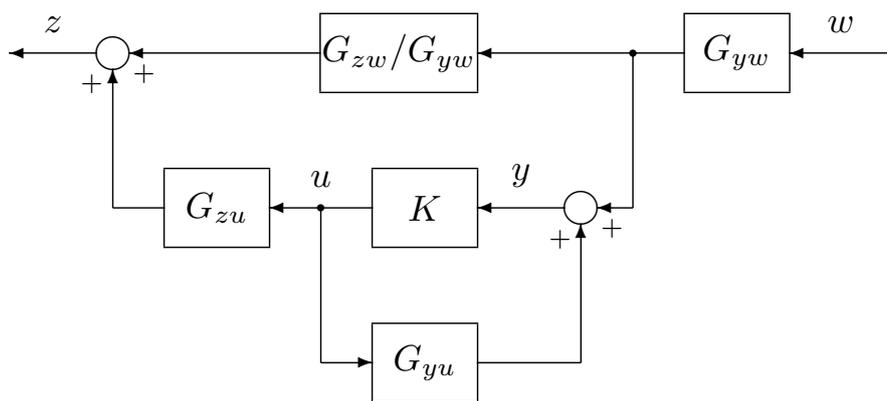


Fig. 5.3 Block diagram of active feedforward control system(equivalent to Fig.5.1).

さらに、外乱 w は未知であり、実際にFRITでの補償器調整や運転を行う際にもその情報を陽に利用できないことから、Fig.5.3の $G_{yw}w$ を改めて w と見なす。同時に、Fig.5.3上側のパス G_{zw}/G_{yw} を改めて、 G_{zw} と置きなおす。これら等価変換をFig.5.4のようにまとめる。

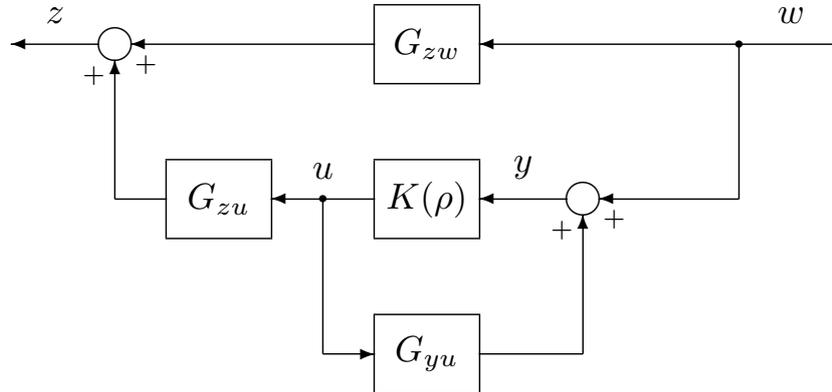


Fig. 5.4 Block diagram of active feedforward control system(simplified).

Fig.5.4 の G_{zw}, G_{zu}, G_{yu} が LMS アルゴリズムにおける一次経路、二次経路、フィードバックパスと対応する。以降でもこのように呼称することとする。

外乱入力 w により消音点 z での観測値は Fig.5.4 より、

$$z = \left(G_{zw} + G_{zu} \frac{K(\rho)}{1 + K(\rho)G_{yu}} \right) w \quad (5.1)$$

となる。アクティブフィードフォワード制御系の目的は $z \approx 0$ にすることであるから、

$$G_{zw} + G_{zu} \frac{K(\rho)}{1 + K(\rho)G_{yu}} \approx 0 \quad (5.2)$$

となるような補償器 $K(\rho)$ を設計すればよい。

5.2 規範モデルの扱い

アクティブフィードフォワード制御系は「 z の観測値がなるべく小さくなるよう、 y の観測値に基づいて制御用アクチュエータ u を駆動する」ことが目的であるから、理想的には $w \rightarrow z$ の規範モデル T_d は

$$T_d = 0 \quad (5.3)$$

としたい。しかし、擬似参照信号を $T_d = 0$ に入力しても応答は常に 0 となり、チューニングが行えない。

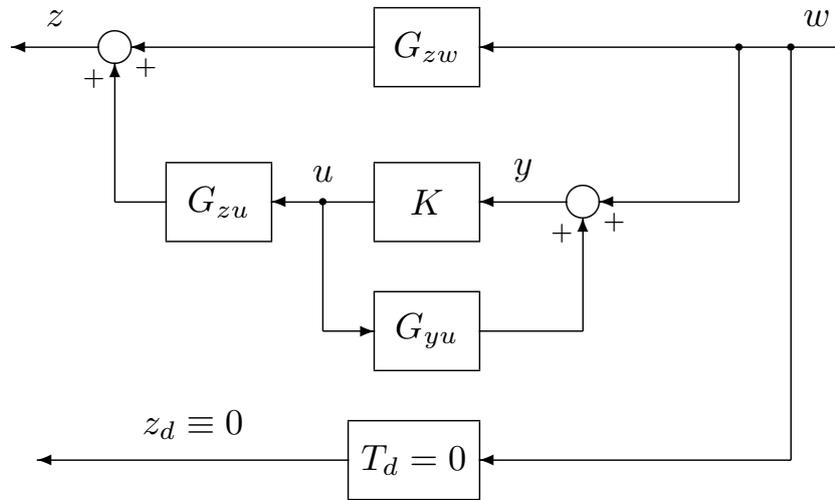


Fig. 5.5 Active feedforward control system and desired model($T_d = 0$)

そこで、本研究では補償器まわりの入出力データに着目し、第 3 章で示した LMS アルゴリズムと同様に Fig.5.4 の二次経路側のパスを一次経路側のパスに近似させることを目指す。一次経路の数学的なモデルを取得してこれを規範モデルとして取り扱い、Fig.5.4 の二次経路側のパスをこれに近似させる。以下、一次経路モデルを \bar{G}_{zw} と表し、これを Fig.5.4 に適用したブロック線図を 5.6 に示す。第 3 章で述べたようにフィードバックパスのモデル化誤差はシステムの安定性に影響を及ぼすが、フィードフォワード項となる一次経路のモデル化誤差が及ぼす影響は性能劣化程度であり、フィードバックパスのモデル化と比較し許容できるものとする。

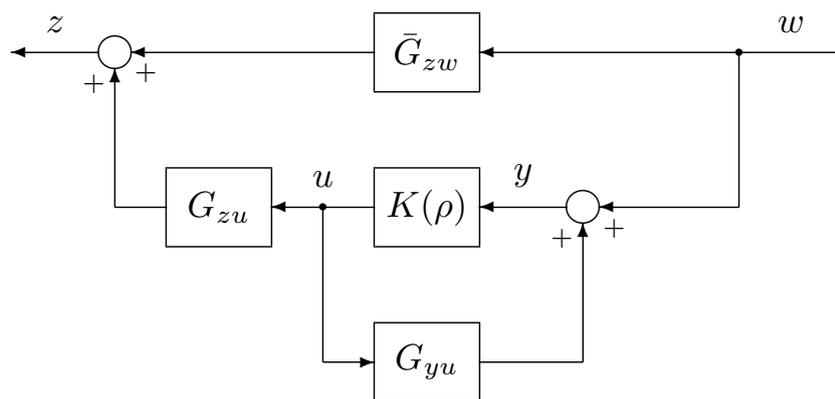


Fig. 5.6 FRIT for active feedforward control system(Primary path: \bar{G}_{zw})

5.3 評価関数の定義

一次経路モデル \bar{G}_{zw} を規範モデルとして用いて、システムを評価する差分は

$$z(\rho) = \left(\bar{G}_{zw} + G_{zu} \frac{K(\rho)}{1 - K(\rho)G_{yu}} \right) w \quad (5.4)$$

となる。式 (5.4) を最小化したいから、評価関数を式 (5.5) のように設定する。

$$\begin{aligned} J(\rho) &= \|z(\rho)\|_2^2 \\ &= \sum_{t=1}^N z(t, \rho)^2 \end{aligned} \quad (5.5)$$

5.4 擬似参照入力信号の導出

第4章で示したFRITと同様に、制御音源 u を初回の実験で得られたデータ u_0 に固定する。このために Fig.5.4 の $K(\rho)$ 直前の加算点に着目して、任意の補償器パラメータ ρ が適用されている制御系における u_0 を生成する擬似参照入力 $\tilde{w}(\rho)$ を式 (5.6) で求める。

$$\tilde{w}(\rho) = (K(\rho)^{-1} - G_{yu}) u_0 \quad (5.6)$$

$\tilde{w}(\rho)$ を利用して $z(\rho)$ を求めると、次の式 (5.7) となる。

$$z(\rho) = \bar{G}_{zw} \tilde{w}(\rho) + G_{zu} u_0 \quad (5.7)$$

右辺第一項の $\bar{G}_{zw} \tilde{w}(\rho)$ は \bar{G}_{zw} のモデルに基づいて計算機上で算出する。右辺第二項の $G_{zu} u_0$ については、実験方法に工夫をすることでデータとして取得する。

5.4.1 最適化問題

式 (5.5) で設定した評価関数 $J(\rho)$ を最小化する補償器パラメータ ρ を求めることで、最適な補償器を設計する。すなわち、

$$\rho^* := \arg \min_{\rho} J(\rho) \quad (5.8)$$

式 (5.5) の評価関数に基づき勾配法的一种である Gauss-Newton 法によって非線形最適化を行い、 ρ を更新する。以下に Gauss-Newton 法によるパラメータの更新式を再掲する。

$$\rho^{i+1} = \rho^i - \gamma \left\{ \frac{\partial^2 J(\rho)}{\partial \rho \partial \rho^T} \Big|_{\rho=\rho^i} \right\}^{-1} \frac{\partial J(\rho)}{\partial \rho} \Big|_{\rho=\rho^i}$$

ここで、評価関数の勾配は式 (5.5) より、式 (5.9) で求められる。

$$\frac{\partial J(\rho)}{\partial \rho} = 2 \sum_{t=1}^N \left[z(t, \rho) \frac{\partial z(t, \rho)}{\partial \rho} \right] \quad (5.9)$$

式 (5.5), (5.9) より、 $\frac{\partial J(\rho)}{\partial \rho}$ は $z(\rho)$ と $\frac{\partial z(\rho)}{\partial \rho}$ から求めることができる。 $z(\rho)$ は式 (5.7) の通りに求められる。 $\frac{\partial z(\rho)}{\partial \rho}$ については、

$$\begin{aligned} \frac{\partial z(\rho)}{\partial \rho} &= \frac{\partial}{\partial \rho} \{ \bar{G}_{zw} \tilde{w}(\rho) + G_{zu} u_0 \} \\ &= \bar{G}_{zw} \frac{\partial}{\partial \rho} \{ \tilde{w}(\rho) \} \\ &= \bar{G}_{zw} \frac{\partial}{\partial \rho} \{ (K(\rho)^{-1} - G_{yu}) u_0 \} \\ &= \bar{G}_{zw} \frac{\partial}{\partial \rho} \{ K(\rho)^{-1} \} u_0 \\ &= \bar{G}_{zw} \frac{\partial}{\partial \rho} \left\{ \frac{1}{K(\rho)} \right\} u_0 \\ &= -\bar{G}_{zw} \frac{1}{(K(\rho))^2} \frac{\partial K(\rho)}{\partial \rho} u_0 \end{aligned} \quad (5.10)$$

と求められる。

5.5 各種データ取得のための実験手順

ρ の最適化には式 (5.7) と (5.10) で $z(\rho)$ と $\frac{\partial z(\rho)}{\partial \rho}$ が必要となる。これらは実験データを利用して求める。

補償器調整のプロセスは実機での実験 2 回と計算機上での仮想的な実験に基づく最適化からなる。実機での各実験をそれぞれ実験 (a) と実験 (b)、計算機上での仮想的な実験を仮想的実験 (c) と呼ぶこととする。なお、外乱源として換気ファンなどを想定し、外乱 w の ON/OFF については可能であるとする。

5.5.1 実験 (a)

まず外乱源を ON とし、適当な補償器 $K(\rho^0)$ を実装した状態で運転を行い、センサ y, z の観測値、及びコントローラの出力 u を記録する。

記録した y, z, u を y_a, z_a, u_a とする。この u_a が式 (5.6) の u_0 に相当する。

$$u_a = K(\rho^0) y_a \quad (5.11)$$

$$z_a = \left(G_{zw} + G_{zu} \frac{K}{1 - KG_{yu}} \right) w \quad (5.12)$$

実験 (a) の手順をまとめたブロック線図を Fig.5.7 に表す。Fig.5.7 の赤色の線で示した u_a をこの実験で得られたデータで固定する。

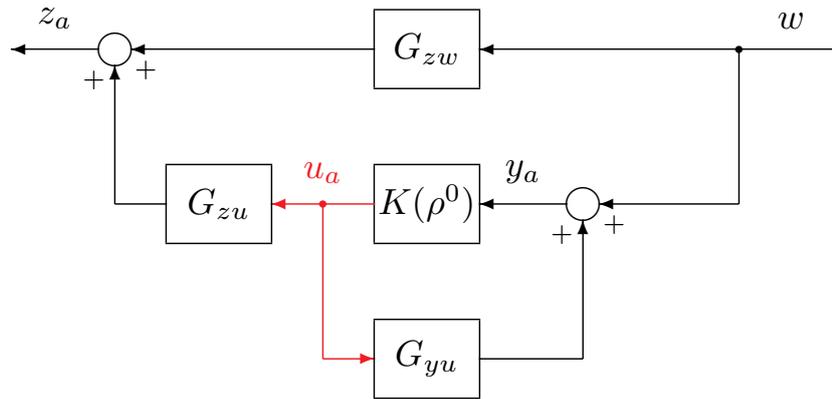


Fig. 5.7 Experiment (a)

5.5.2 実験 (b)

次に外乱源を OFF、すなわち $w \equiv 0$ として運転を行う。制御音源を実験 (a) で取得した u_a で駆動し、リファレンスセンサの観測値 y_b とエラーセンサの観測値 z_b を記録する。この実験を行うことで、実験 (a) で発生させた制御音による二次経路とフィードバックパスへの影響を分離する。

この実験より得られる y_b, z_b は

$$y_b = G_{yu}u_a \quad (5.13)$$

$$z_b = G_{zu}u_a \quad (5.14)$$

となる。

実験 (b) の手順をブロック線図で表すと、Fig.5.8 となる。

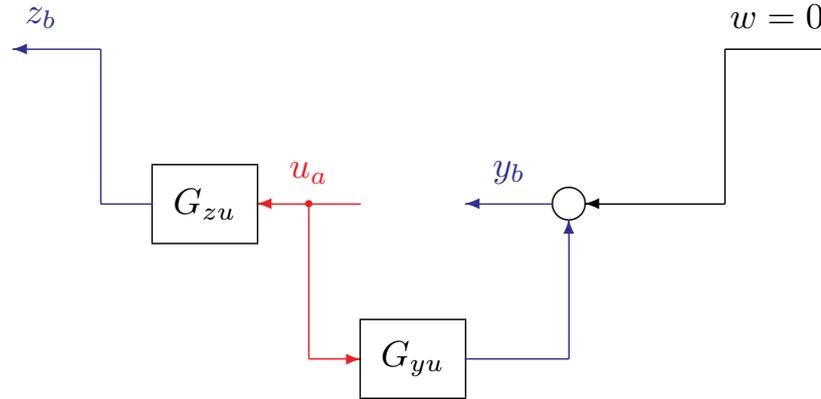


Fig. 5.8 Experiment (b)

5.5.3 仮想的実験 (c)

実験 (a) 及び (b) より取得した実験データを用いて、式 (5.10) を算出するよう計算機上で計算を行い、パラメータ更新のための評価関数の勾配を求める。計算機上で完結するため、「仮想的」実験と呼ぶこととする。

まず、式 (5.6) で擬似参照入力 $\tilde{w}(\rho)$ を求める。

$$\begin{aligned} \tilde{w}(\rho) &= (K(\rho)^{-1} - G_{yu}) u_0 \\ &= K(\rho)^{-1} u_0 - G_{yu} u_0 \\ &= K(\rho)^{-1} u_a - y_b \end{aligned} \tag{5.15}$$

算出した $\tilde{w}(\rho)$ を利用して、式 (5.7) に相当する演算を行う。

$$\begin{aligned} z(\rho) &= \bar{G}_{zw} \tilde{w}(\rho) + G_{zu} u_0 \\ &= \bar{G}_{zw} \tilde{w}(\rho) + z_b \end{aligned} \tag{5.16}$$

式 (5.15),(5.16) に相当するブロック線図を Fig.5.9 に示す。

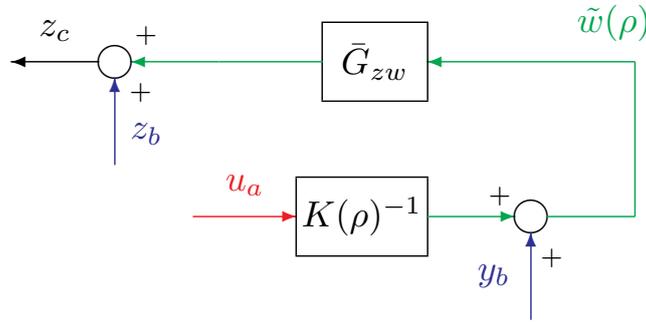


Fig. 5.9 Virtual experiment (c)

また、 $\frac{\partial z(\rho)}{\partial \rho}$ を次のように求める。

$$\begin{aligned} \frac{\partial z(\rho)}{\partial \rho} &= -\bar{G}_{zw} \frac{1}{(K(\rho))^2} \frac{\partial K(\rho)}{\partial \rho} u_0 \\ &= -\bar{G}_{zw} \frac{1}{(K(\rho))^2} \frac{\partial K(\rho)}{\partial \rho} u_a \end{aligned} \tag{5.17}$$

式 (5.16), (5.17) で算出した $z(\rho)$ と $\frac{\partial z(\rho)}{\partial \rho}$ を利用して、 $J(\rho)$ と $\frac{\partial J(\rho)}{\partial \rho}$ を算出し、式 (4.13) に基づいてパラメータ更新を行う。

第 6 章

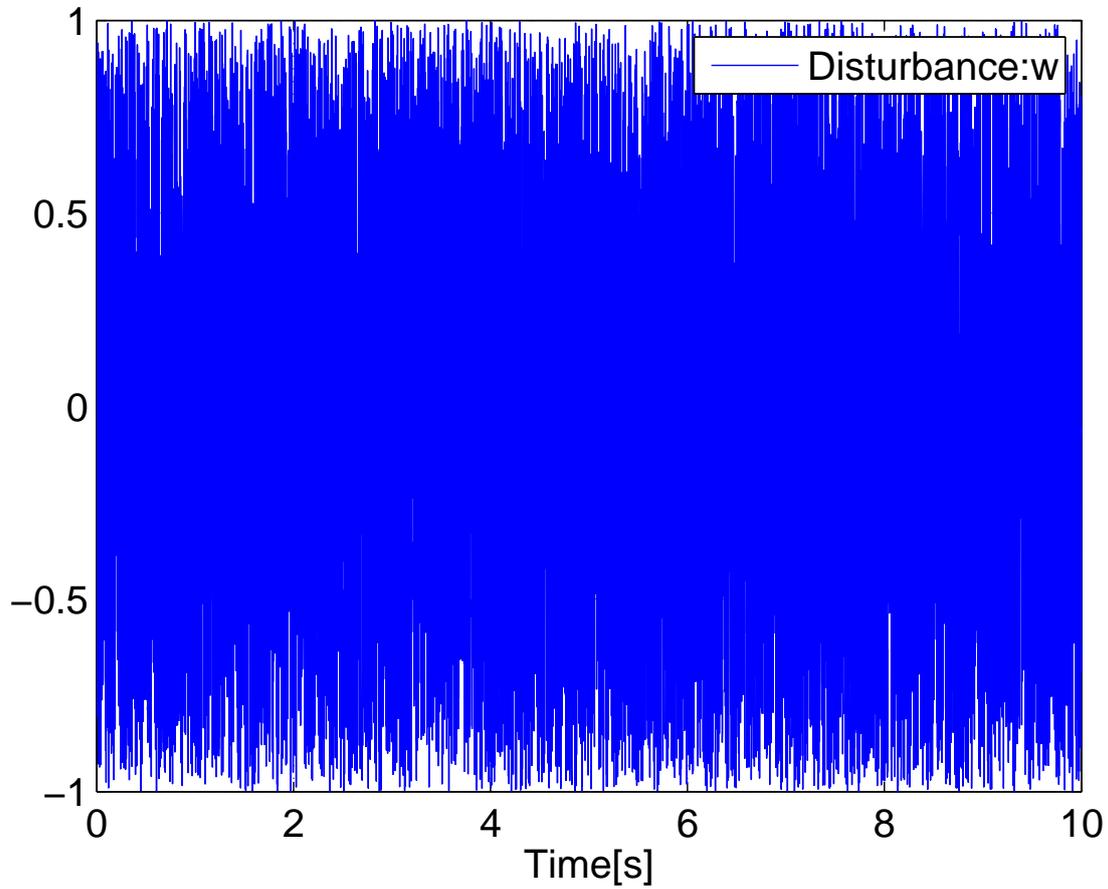
シミュレーションによる実装

第 5 章で述べた提案法が実際に使用できることを MATLAB/Simulink を用いたシミュレーションを通して実証する。

6.1 シミュレーション条件の設定

プラントと補償器をいずれも時不変な線形離散時間システムとして実装する。サンプリング時間は $T_s = 10^{-3}[\text{s}] = 1[\text{ms}]$ 、シミュレーション時間は $T = 10[\text{s}]$ と設定した。想定する制御系の構造は Fig.5.4 と同様に一次経路: G_{zw} 、二次経路: G_{zu} 、フィードバックパス: G_{yu} を有するシステムとする。

システムへ入力する外乱 w は、MATLAB の rand 関数を利用して生成した振幅 1 の一様分布の白色雑音を利用する。Fig.6.1 に入力した外乱のプロットを示す。

Fig. 6.1 Disturbance: w

6.1.1 プラント

能動騒音抑制制御系より、音波の伝播を想定してプラントを単純なむだ時間と見なし、各プラント要素を式 (6.1) のように定義する。

$$\begin{cases} G_{zw} = z^{-49} \\ G_{zu} = z^{-30} \\ G_{yu} = z^{-19} \end{cases} \quad (6.1)$$

なお一次経路モデル \bar{G}_{zw} については簡単のため、 $\bar{G}_{zw} = G_{zw}$ とする。

6.1.2 理想的な補償器

FRIT では、既存の補償器のパラメータのファインチューニングを目指す。今回のシミュレーションでは補償器の構造はプラントの伝達関数より逆算して決定することとする。

理想的な補償器 K_{ideal} は

$$G_{zw} + G_{zu} \frac{K(\rho)}{1 + K(\rho)G_{yu}} = 0 \quad (6.2)$$

を $K(\rho)$ について代数的に解くことで、

$$\begin{aligned} K_{\text{ideal}}(z^{-1}) &= -\frac{G_{zw}}{G_{zu} - G_{yu}G_{zw}} \\ &= -\frac{z^{-49}}{z^{-30} - z^{-19} \cdot z^{-49}} \\ &= -\frac{z^{-19}}{1 - z^{-38}} \end{aligned} \quad (6.3)$$

と得られる。

6.2 パラメータ数が二つの場合

6.2.1 補償器構造の設定

調整する補償器の構造を K_{ideal} と同様のものとして、式 (6.4) のように設定する。

$$K(\rho, z^{-1}) = \frac{\rho(1)z^{-19}}{1 + \rho(2)z^{-38}} \quad (6.4)$$

補償器の初期パラメータ ρ^0 を次のように設定する。

$$\rho^0 = [\rho^0(1), \rho^0(2)] = [-0.5, -0.5] \quad (6.5)$$

つまり、 $\rho^0 = 0.5 \cdot \rho_{\text{ideal}}$ である。すなわち、調整開始前の補償器 K_0 は式 (6.6) となる。

$$K_0(z^{-1}) = \frac{-0.5z^{-19}}{1 - 0.5z^{-38}} \quad (6.6)$$

式 (6.3) と式 (6.4) を比較して、理想的な補償器パラメータ ρ_{ideal} は

$$\rho_{\text{ideal}} = [-1, -1] \quad (6.7)$$

である。

6.2.2 シミュレーション結果

最終的に得られた最適な補償器パラメータ ρ^* は 25 回の更新で、

$$\rho^* = [\rho^*(1), \rho^*(2)] = [-1.000, -1.000] \quad (6.8)$$

が得られた。すなわち、調整完了後の補償器 K^* は、

$$K^*(z^{-1}) = \frac{-1.000z^{-19}}{1 - 1.000z^{-38}} \quad (6.9)$$

となった。理想的な補償器パラメータ ρ_{ideal} に近い値が得られたことから、所望の補償器パラメータが得られたと考えられる。Fig.6.2 に補償器パラメータ更新の推移を示す。青色の実線が ρ_1 、緑色の実線が ρ_2 の推移である。式 (6.13) の理想的な補償器パラメータへ収束している様子が判る。

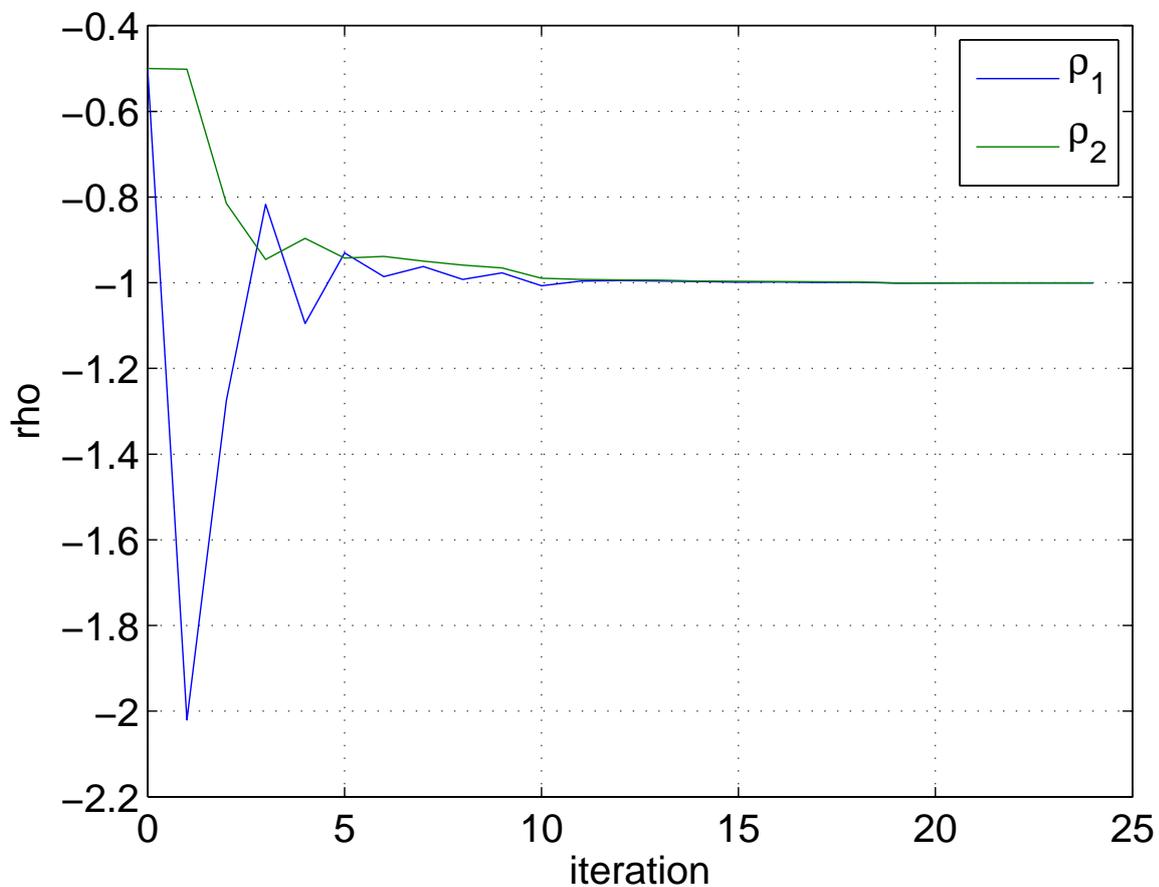


Fig. 6.2 Transition of controller's parameters: ρ (Number of parameters:2)

Fig.6.3 は外乱を入力して制御系を運転する際のエラーセンサの観測値をプロットしたものである。青色の実線が補償器なし ($K = 0$)、緑色の実線が調整前の補償器 (K^0) を実装した結果 (z^0)、赤色の実線が調整完了後の補償器 (K^*) を実装した結果 (z^*) である。 z^0 と z^* の 2-ノルムを比較すると、

$$\begin{aligned} \|z^0\| &= 28.7 \\ \|z^*\| &= 20.6 \times 10^{-2} \end{aligned} \quad (6.10)$$

となっている。Fig.6.3 及び式 (6.10) より、補償器の調整によって外乱 w の影響をなるべく低減させる、という目的を達成していることが判る。

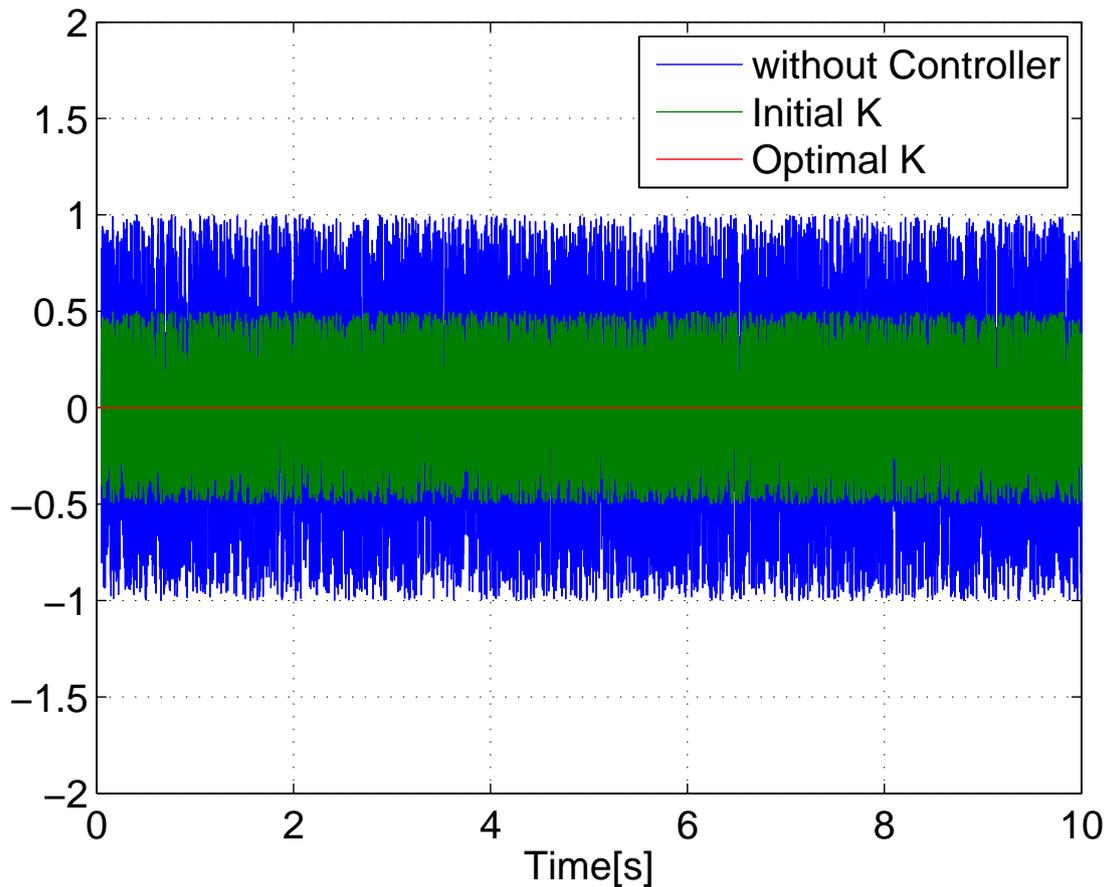


Fig. 6.3 Error sensor output(Number of parameters:2)

6.3 パラメータ数が三つの場合

次に、調整パラメータ数が増加しても提案法が適用できることを示すため、補償器の調整パラメータを三つに増やして補償器の調整を行う。

6.3.1 補償器構造の設定

補償器の構造を

$$K(z^{-1}, \rho) = \frac{\rho(1)z^{-19} + \rho(2)z^{-20}}{1 + \rho(3)z^{-38}} \quad (6.11)$$

と設定し、補償器の初期パラメータは

$$\rho^0 = [\rho^0(1), \rho^0(2), \rho^0(3)] = [-0.5, -0.5, -0.5] \quad (6.12)$$

とする。

式 (6.3) と式 (6.11) を比較して、理想的な補償器パラメータ ρ_{ideal} は

$$\rho_{\text{ideal}} = [-1, 0, -1] \quad (6.13)$$

である。

6.3.2 シミュレーション結果

最終的に得られた最適な補償器パラメータ ρ^* は 136 回の更新で、

$$\rho^* = [\rho^*(1), \rho^*(2), \rho^*(3)] = [-0.9990, -0.0004, -0.9986] \quad (6.14)$$

が得られた。すなわち、調整完了後の補償器 K^* は、

$$K^*(z^{-1}) = \frac{-0.9990z^{-19} - 0.0004z^{-20}}{1 - 0.9986z^{-38}} \quad (6.15)$$

となった。各パラメータとも理想的な補償器パラメータ ρ_{ideal} の $\pm 2 \times 10^{-3}$ 以内の値が得られたことから、所望の補償器パラメータに十分近い値に調整できたとと言える。Fig.6.4 に補償器パラメータ更新の推移を示す。

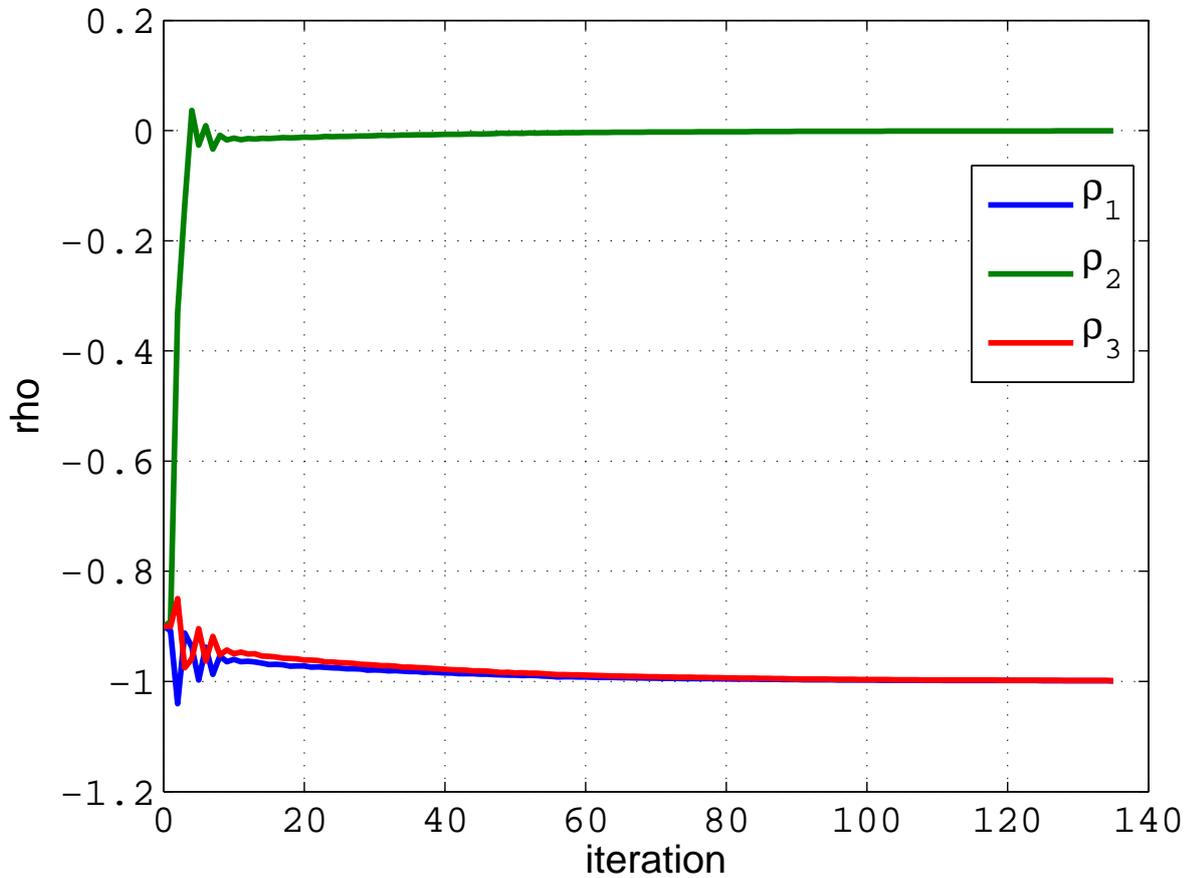


Fig. 6.4 Transition of controller's parameters: ρ (Number of parameters:3)

Fig.6.5 は外乱を入力して制御系を運転する際のエラーセンサの観測値をプロットしたものである。緑色の実線が調整前の補償器 (K^0) を実装した結果 (z^0)、赤色の実線が調整完了後の補償器 (K^*) を実装した結果 (z^*) である。 z^0 と z^* の 2-ノルムを比較すると、

$$\begin{aligned} \|z^0\| &= 162 \\ \|z^*\| &= 7.14 \times 10^{-4} \end{aligned} \quad (6.16)$$

となっている。Fig.6.5 及び式 (6.16) より、補償器パラメータの数を増やしてもパラメータ調整が可能であることが判る。

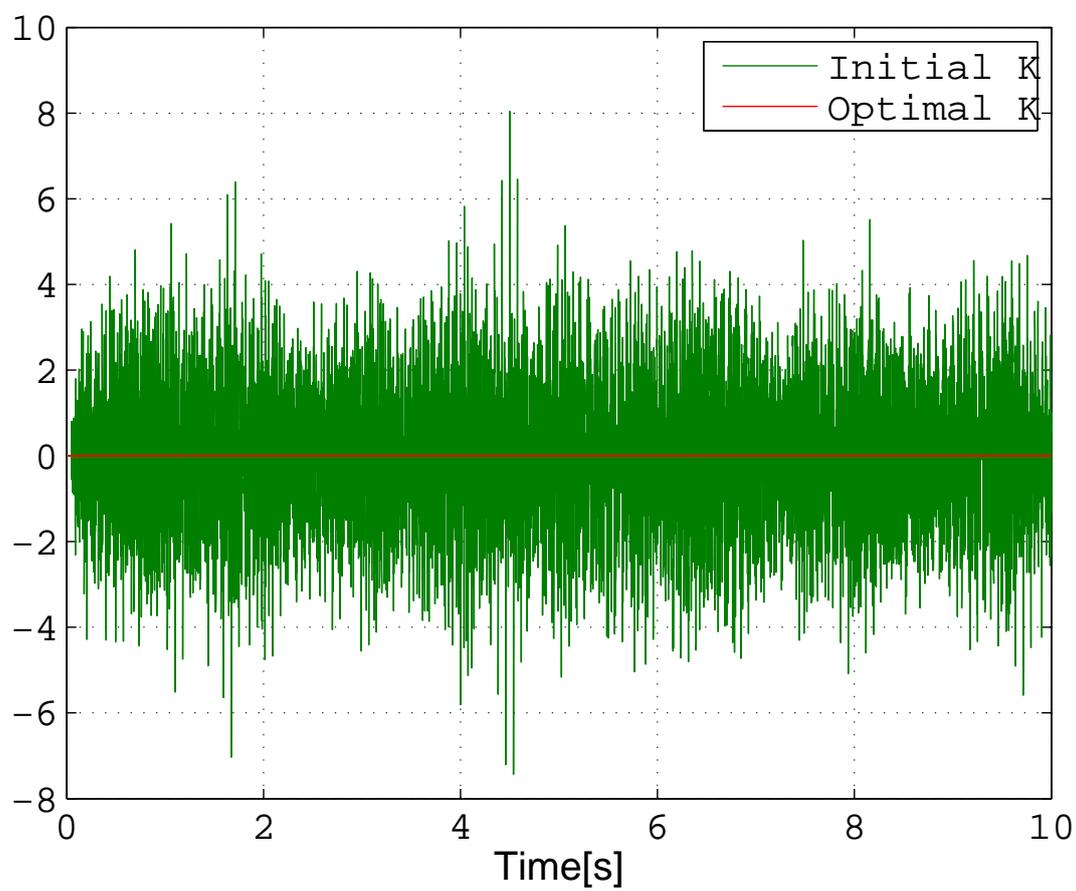


Fig. 6.5 Error sensor output(Number of parameters:3)

第7章

結論

7.1 まとめ

第2章では、本研究の対象とする制御系である、アクティブフィードフォワード制御系の概要について述べた。

第3章では、従来よりアクティブフィードフォワード制御系の補償器調整に利用されている LMS アルゴリズムについて述べた。フィードバックパスを有するシステムに対して LMS アルゴリズムを適用する場合、フィードバックパスのモデルによってこれを補償するが、モデル化誤差に起因する安定性の問題が考えられる。

第4章では、実機の運転データに基づいた目標値追従制御系の補償器を調整するアルゴリズムを紹介した。これらのアルゴリズムは実機の応答を規範モデルに追従させることを目的とする。実験データを利用して調整を行うため、プラントモデルを必要しない。IFT はパラメータ更新ごとに毎回実機の運転を行わなければならないが、FRIT は実機の運転は初回のみでよい。

第5章では、FRIT をアクティブフィードフォワード制御系へ導入するための拡張を行った。アクティブフィードフォワード制御系は理想的には外乱感度が0であることが望ましいが、規範モデルを0としてしまうとデータに基づくシステムの比較ができない。そこで、一次経路のモデルを規範モデルとして扱うことでFRIT が利用できるようになった。

第6章では、第5章で提案した手法によって実際に補償器を調整できることを MATLAB/Simulink を用いたシミュレーションによって実証した。

7.2 課題

今回シミュレーションで用いた補償器の構造はプラントの伝達関数から逆算して設定しているが、実際にはプラントの数学的モデルは未知であるため不可能である。今後の課題として、IIR フィルタなどのより一般化した構造の補償器調整について考察を行いたい。

また、実在するシステムに対して今回提案した手法を適用し、補償器調整が行えることも実証したい。

謝辞

小林泰秀先生には、研究を進めるにあたり、丁寧なご指導と多大なるご支援を賜りました。深く感謝致します。

平田研二先生には学部時代、制御工学の基礎をご指導いただきました。心よりお礼申し上げます。

研究室で研究を共に続けてきた先輩・後輩の皆さん、そして同期の角島悠太くん、西村光博くんにはたくさんの助言を頂きました。ありがとうございました。

最後に、修士2年まで学ぶ機会を与えてくれた両親に深く感謝致します。

平成 26 年 2 月

参考文献

- [1] 田中信雄. 振動制御. 養賢堂, 9 2008.
- [2] Stephen Elliott. *Signal processing for active control*. Academic press, 2001.
- [3] H Hjalmarsson. Iterative feedback tuning - an overview. *International Journal of Adaptive Control of Signal Processing*, Vol. 13, No. 5, pp. 553–572, 1999.
- [4] T Meurers and S M Veres. Iterative design for vibration attenuation. *International Journal of Acoustics and Vibration*, Vol. 4, No. 2, 1999.
- [5] T Meurers, S M Veres, and A C H Tan. Model-free frequency domain iterative active sound and vibration control. *Control Engineering Practice*, Vol. 11, , 2002.
- [6] 相馬将太郎, 金子修, 藤井隆雄. 一回の実験データに基づく制御器パラメータチューニングの新しいアプローチ-Fictitious Reference Iterative Tuning の提案. システム制御情報学会論文誌, Vol. 17, No. 12, pp. 528–536, 2004.
- [7] 西久保智昭. 柔軟片持ち梁のフィードフォワード外乱抑制制御系における指向性アクチュエータの効果. 修士論文, 長岡技術科学大学 大学院, 平成 23 年度.
- [8] 福嶋廣. トランジスタ技術 2002 年 3 月号, DSP スタータ・キット TMS320C5x で作る電子消音システムの製作. CQ 出版社, 2002.
- [9] 西山清. 最適フィルタリング, システム制御シリーズ, 第 6 巻. 培風館, 2001.
- [10] H Hjalmarsson, M Gevers, S Gunnarsson, and O Lequin. Iterative feedback tuning: theory and applications. *IEEE Control Systems*, Vol. 18, No. 4, pp. 26–41, 1998.
- [11] Laurence Dixon. 非線形最適化計算法. 培風館, 1974.
- [12] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm Intelligence*, Vol. 1, No. 1, pp. 33–57, 2007.
- [13] 計測自動制御学会. 特集 データ駆動制御-新機軸と新地平. 計測と制御. 計測自動制御学会, 2013.