

# 概要

近年、電力機器の力率の改善と、高調波のさらなる低減化が要求されてきている。一般的な電力機器には、非線形性を持つ整流器があり、入力側の力率の低下と高調波の発生を招く原因となっている。機器の力率が悪いと、送電の効率が悪く電力の浪費となる。また、電力系統の電流波形に高調波が存在すると、電力線やコンデンサの過熱焼損事故などを引き起こす原因となる。このため、欧州では早くから力率と高調波が規制され、日本でも規制が始まっている。

力率と高調波の特性を改善する方法として、一般に、アクティブ PFC(Power Factor Correction:力率改善)回路、特に電流導通モード昇圧型コンバータ回路が用いられる。昇圧型コンバータは、非線形システムであり、負荷や時比率・入力電圧に応じてその動特性が大きく変動する。PFC 昇圧型コンバータの多くの応用では、負荷や入力電圧は一意に決定できず、仕様の最小値から最大値まで急に変更される。さらに力率改善を行う際は、時比率も連続的に変化するため、制御が難しいとされる。

これまでに DC-DC コンバータをロバストに制御する近似的 2 自由度制御器が提案された。この制御器は制御対象の伝達関数を 1 次モデルに近似しつつ、目標値と出力間、外乱と出力間の伝達関数をそれぞれ独立に指定でき、ロバストなものとなっている。しかしながら、この制御器は、入力電流と出力電圧の 2 つを制御する PFC 昇圧コンバータに直接適用すると設計が困難である。

本研究では、電流導通モードの PFC 昇圧コンバータを制御対象とし、力率を改善し、高調波を十分低減しながら、負荷の変動に対して出力電圧の変動を十分小さくするロバストな近似的 2 自由度ディジタル積分形制御器の設計法を提案する。本手法により設計される制御器は、電流制御ループと電圧制御ループの 2 重ループを持ち、パラメータの設計条件から、内側の電流制御ループを一次モデルに近似することで、ロバスト性を保ちながら電圧制御ループの設計を簡易にする。

提案された制御器は、シミュレーションによってその特性を確認しながら、パラメータを決定する。設計されたディジタル制御器はマイクロコントローラ(SH7216)に実装された後、PFC 昇圧型コンバータに結合され、ディジタル制御器搭載 PFC 昇圧コンバタシステムを実現する。実験によって、新しく設計・製作した PFC 昇圧コンバタシステムが、シミュレーションとほぼ同様の動作をし、力率と高調波特性を改善しながら、各動作点において出力電圧のステップ応答特性が変わらず、負荷急変による出力電圧変動を小さく抑えることが可能なことを示している。以上から、提案した力率改善近似的 2 自由度ディジタル積分形制御器とその設計法は有効であり、実用的であることを確認している。

# 目次

概要 .....	1
目次 .....	2
第1章 序論.....	4
1.1 力率改善の必要性とその手法.....	4
1.2 PFC 昇圧型コンバータの制御.....	5
1.3 本研究の目的.....	7
第2章 力率改善回路の特性・数学モデルと離散化.....	9
2.1 力率改善回路の基本構成.....	9
2.2 制御対象のモデル化.....	11
2.2.1 状態平均化法による制御対象の数学モデル化.....	11
2.2.2 制御対象の静特性.....	12
2.2.3 制御対象の動作点近傍における動特性.....	13
2.3 状態方程式の離散化.....	15
2.4 遅れ時間の考慮と制御対象の伝達関数.....	16
第3章 実験基板とデジタルコントローラの製作.....	18
3.1 マイクロコントローラ .....	18
3.1.1 マイクロコントローラの概要.....	18
3.1.2 マイクロコントローラの動作.....	19
3.2 PFC 昇圧型コンバータ基板.....	21
3.2.1 PFC 昇圧型コンバータ基板の概要.....	21
3.2.2 PFC 昇圧型コンバータ基板とマイクロコントローラの接続.....	22
第4章 出力電圧のみを制御する近似的2自由度制御系の設計法 .....	23
4.1 近似的2自由度制御による目標 .....	23
4.1.1 等価外乱 .....	23
4.1.2 細分系制御対象 .....	24
4.1.3 昇圧型DC-DCコンバータの仕様と動作点の決定 .....	25
4.2 昇圧型DC-DCコンバータの近似的2自由度制御器の設計 .....	26
4.2.1 状態フィードバック則の適用 .....	26
4.2.2 モデルマッチングシステムの等価変換 .....	27
4.2.3 逆システムとフィルタを用いたロバスト制御器 .....	29
4.3 制御器パラメータの決定法 .....	32
4.4 制御器の設計 .....	33
4.5 シミュレーションとその結果 .....	34
4.6 実験結果 .....	40

第 5 章 力率改善を行う近似的 2 自由度制御系の設計法.....	44
5.1 力率改善手法の概要と目標.....	44
5.2 PFC 昇圧型コンバータ.....	45
5.2.1 離散系制御対象.....	45
5.2.2 PFC 昇圧型コンバータの仕様と動作点の決定.....	46
5.3 PFC 昇圧型コンバータの近似的 2 自由度制御器の設計.....	47
5.3.1 電流制御器の設計.....	47
5.3.2 電流制御器の近似と乗算器の接続.....	50
5.3.3 電圧制御器の設計.....	53
5.4 制御器パラメータの決定法.....	55
5.5 制御器の設計.....	58
5.6 シミュレーションとその結果.....	59
5.7 実験結果 .....	71
5.8 入力電圧の変動.....	84
5.8.1 入力電圧変動の影響.....	84
5.8.2 制御器への補正.....	85
5.8.3 入力電圧急変シミュレーション.....	86
第 6 章 結論.....	92
謝辞 .....	94
参考文献 .....	95
本研究で使用した実験装置.....	96
本研究で使用したソフトウェア .....	96
付録 A MATLAB プログラム .....	97
A.1 出力電圧のみ制御する昇圧型コンバータの MATLAB プログラム .....	97
A.2 力率改善制御する PFC 昇圧型コンバータの MATLAB プログラム .....	103
付録 B マイクロコントローラプログラム .....	110
付録 C 数式の導出 .....	117
C.1 $W_{ry}$ の導出.....	117
C.2 G の導出.....	118
C.3 $r \rightarrow y$ 伝達関数の導出.....	119
C.4 $Q \rightarrow y$ 伝達関数の導出 .....	120
付録 D 実験装置と回路図.....	121
D.1 マイクロコントローラ SH7216 と昇圧コンバータボードの接続回路 .....	121
D.2 実験装置の外観 .....	122
図目次 .....	123

# 第1章 序論

## 1.1 力率改善の必要性とその手法

近年、電力機器の力率の改善と、高調波のさらなる低減化が要求されてきている。一般的な直流電源を必要とする電力機器は、非線形性を持つ整流器が存在するため、その機器の入力側では、電流波形が非正弦波となり高調波成分が生じる。また、機器に含まれる、平滑のためのキャパシタや、インダクタによって、入力電流の位相がずれ、力率の悪化を招く。この時の入力電圧と入力電流波形の例を図 1.1 に示す。このようにして生じた高調波電流や、低い力率により、以下のような問題が起こる。

- (1)送電効率の悪化による、電力の浪費と電力会社のコスト増
- (2)電力系統に接続されている、変圧器、コンデンサ、モータ、照明用バラストなどや電力線自体の過熱、損傷
- (3)ブレーカ、ヒューズ、リレーなどの誤動作
- (4)電力系統から電力を得ている他の電子機器の損傷
- (5)隣接する電子機器への EMC ノイズ等による干渉

こういった問題を防ぐため、入力電流波形を、入力電圧と同じ波形・同位相にして、電力機器を見かけ上抵抗のような振る舞いにする必要がある。これを、力率改善(力率補正、PFC, Power Factor Correction)と呼ぶ。最近では直流電源を必要とする多くの機器が使用され、それらが電力系統に接続されることによって、上に述べたような問題から、電力系統の品質低下を招く。このため、高調波電流に対して、国際規格(IEC/EN61000-3-2)・国内規格(JIS C61000-3-2)によって厳しい制約が課されている。

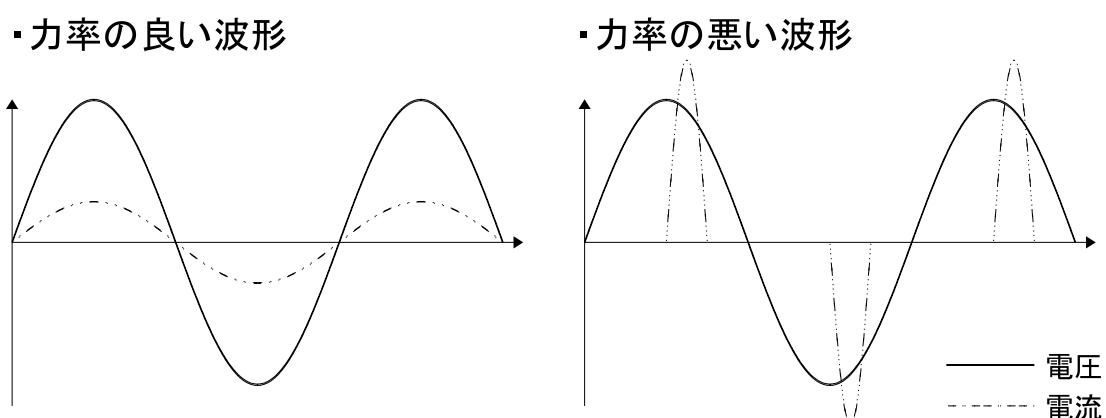


図 1.1 機器の入力電圧と入力電流波形の例

従来、力率の改善は、大電力を要する箇所に限られ、その手法は、進相キャパシタによる位相補正や、AC ラインフィルタによる高調波除去が用いられてきた。しかしながら、蛍光灯一本から高調波電流が規制される昨今では、そういう受動素子による改善では、高調波電流は十分除去できず、力率を 1 に近づけることは難しい。また、こういった手法では素子が大型になり、コスト増を招く。そのため、パワーエレクトロニクス技術を用いた、小型・低コストのアクティブフィルタなどによる、力率と高調波特性のさらなる改善が望まれる。

力率と高調波の特性を改善する方法として、交流直流変換を行う AC-DC コンバータに PFC 機能を備える手法が一般的に用いられる。図 1.2 は、PFC 機能付き AC-DC コンバータのブロック図である。ここで PFC コンバータ部には、スイッチング方式の降圧型、昇圧型、昇降圧型などの DC-DC コンバータが適用できる。小電力用途では、降圧型や昇降圧型が用いられることがあるが、これらの方では、原理的に電流波形が不連続になる点や、クロスオーバー歪みが生じてしまうといった理由から、高調波を減らしにくいため、一般に、昇圧型 DC-DC コンバータが用いられる。また、昇圧型 DC-DC コンバータにも不連続導通モード、臨界導通モード、連続導通モードといった動作モードがあるが、スイッチング損失を小さく抑えるため、中・大電力用途では、連続導通モードがよく用いられる。この、PFC コンバータ部に昇圧型 DC-DC コンバータを採用した AC-DC コンバータを、PFC 昇圧型コンバータと呼ぶ。

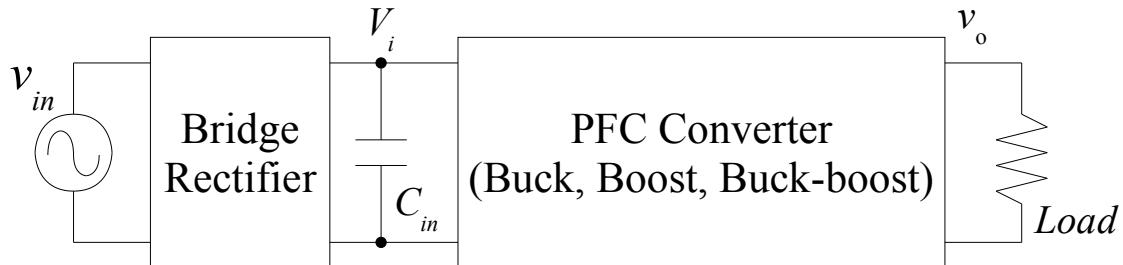


図 1.2 PFC 機能付き AC-DC コンバータのブロック図

## 1.2 PFC 昇圧型コンバータの制御

PFC 昇圧型コンバータの制御には、従来、アナログ IC によるシンプルな積分制御が用いられてきた。近年では、DSP やマイコンなどの小型省電力なプロセッサの急速な進歩により、高速化が進んでおり、PFC 昇圧型コンバータをデジタルで制御することが、現実的となっている。

制御部がアナログである場合とデジタルにした場合における比較を表 1.1 に示す。機能

が増すほど、デジタル制御を採用したほうが、部品点数を削減し小型化でき、また機能の仕様変更などに柔軟に対応できる。PFC 昇圧型コンバータには、効率を改善する、極力高調波を減らす、無負荷から定格負荷までの負荷変動に対応できる、立ち上がり時間を設定できる、使用地域による入力電圧とその周波数の変化にも対応できる、など、多くの機能が必要である点から、デジタル制御を採用すると有利となる。さらに、パラメータの精度等の問題でアナログでは実現できなかった、現代制御理論などの高度な制御が、デジタル制御を採用することで容易に実装できるといった利点がある。また、アナログ制御では、無負荷から定格負荷という広範囲の負荷変化に対応するため、乗算器が完全な線形でない、負荷が軽い場合は電流制御をしない、といった実装がなされていることが普通であるが、これらは入力電流の歪みを生じ、十分な高調波特性を得られない原因となる。デジタル制御では、このような実装をせずとも、広範囲な負荷変化に対応することが可能である。

以上のことから、本研究では、PFC 昇圧型コンバータをデジタルで制御する。

表 1.1 制御システム構成の比較

制御構成	アナログ制御	デジタル制御
制御方式	×: 一般に積分制御などの古典制御が限界	○: 現代制御理論等が容易に実装可能。
制御部高速化	○: スイッチング周波数がMHz帯の制御も可能	△: 現状では、~1MHzの電源制御が限界
機能拡張性	×: 部品や回路の変更が必要	○: シーケンス処理に加え、制御系の変更や追加が可能
消費電力	○: CMOS ICやオペアンプ等は数十mWと小さい	×: 数百mWと大きい
部品点数・コスト	△: 仕様により変わる。機能が増すほどデジタル制御が	

昇圧型コンバータは、2章で述べるように非線形システムであり、負荷や時比率、入力電圧に応じてその動特性が大きく変動する。PFC コンバータの多くの応用では、負荷や入力電圧は一意に決定できず、仕様の最小値から最大値まで急に変更される。さらに時比率も連続的に変化するため、制御が難しいとされる。一般に、コンバータの制御器の設計は、ある 1 つの動作点に対して、コンバータを線形近似し、それを制御対象として制御器を作る。しかし、非線形なシステムにおける動特性を改善したい場合、1 つの動作点に対するモデルのみを考慮するだけでは不十分である。動特性を改善する手法として、非線形モデルを線形モデルの集合として扱うゲインスケジュールド制御などがある。しかしながら、このような制御法では、制御器パラメータがそれぞれの動作点に対して必要となり、その実

装には膨大なテーブルや煩雑な制御ルーチンが要求される。また、ロバストに制御する手法として、 $H\infty$ 制御や最適制御などがあるが、これらの制御理論では制御器が高次になりやすく、演算処理に時間がかかる。PFC 昇圧型コンバータ等のスイッチング電源では、サンプリング周波数が 100kHz 以上と、非常に高速で動作しているため、こういった複雑・煩雑な制御は不向きである。また、一般にデジタル制御でよく用いられる、古典制御の代表である PID 制御などの技術では、十分な高調波特性を保ちながらロバスト性を確保するのは困難である。そこで、ひとつの制御器でこのような入力電圧の変化や負荷抵抗の変動に対応できる、構成が簡単でロバストな制御器と、その合理的な設計手法が求められている。

### 1.3 本研究の目的

これまで、一般的な DC-DC コンバータに求められる性能を簡単なアルゴリズムで実現する手法として、近似的 2 自由度制御器の設計法が提案されてきた。この制御器は、目標値から出力までの伝達特性と、外乱から出力までの伝達特性がそれぞれ近似的に独立であり、設計が容易で、かつ制御アルゴリズムが簡単で 1 次で実現できる、ロバスト性を持った制御器である。これまでに、この設計法を用いた、DC-DC と DC-AC コンバータへ適用する、近似的 2 自由度デジタル積分型制御器が提案されている。

しかしながら、この制御器は、これまで PFC 昇圧型コンバータのような、非線形の制御対象で、電流制御器と電圧制御器の 2 重ループを持つような制御系には適用されていない。そのまま適用したのでは、内側の電流制御器の次数が高くなるため、外側の電圧制御器の設計が複雑になり、さらに全体のシステムが大きくなつて、演算処理に時間がかかる。また、PFC 昇圧型コンバータでは、力率を改善するために時比率が常に変化し、それによつて動作点も変化するが、こういった非線形システムの動作点変化に対する考慮は十分になされていない。PFC 昇圧型コンバータに近似的 2 自由度制御を適用するには、動作点変化を考慮して、よりロバスト性を高めること、さらに電圧制御器の設計を簡易化することが必要である。

本研究では、連続導通モードの PFC 昇圧コンバータを制御対象とし、力率を改善・高調波を十分低減しながら、負荷変動などの外乱に対して出力電圧の変動を十分小さくするロバストな近似的 2 自由度制御器の設計手法について述べる。提案する制御器は入力電流・出力電圧とともに、多動作点における特性を考慮されてロバストであり、またパラメータの設計条件によって電流制御器を 1 次モデルに近似することで、電圧制御器の設計を簡素化し、簡単な制御アルゴリズムで実現できる。またこの制御器は、定められた仕様を満たし、十分なロバスト性を持つことを、シミュレーション及び実験から確認している。実験には、デジタル制御に必要な A/D コンバータや PWM ジェネレータを備えたマイクロコントローラ(SH-2A-FPU, SH7216 ルネサスエレクトロニクス社製)を使用して、PFC 昇圧型コンバ

タの実験機を設計・製作し、制御器の有用性を検証している。

最後に本論文の構成を示す。第2章では、制御対象であるPFCコンバータの特性と状態平均化法による数学モデル化、そのモデルの離散化及び遅れ時間の考慮について説明する。第3章では、使用したマイクロコントローラとその周辺装置・演算について示す。第4章では、出力電圧のみを制御することを目的とした、昇圧型コンバータの近似的2自由度ディジタル積分形制御器の設計法について説明する。第5章では、入力電流を入力電圧に追従させ、力率と高調波特性を改善することを目的とし、出力電圧とインダクタ電流両方を制御する、PFC昇圧型コンバータの近似的2自由度ディジタル積分形制御器の設計法について説明する。第6章では、結論を述べる。

# 第2章 力率改善回路の特性・数学モデルと離散化

## 2.1 力率改善回路の基本構成

図2.1にPWMジェネレータとA/Dコンバータを搭載したマイクロコントローラを用いたPFC昇圧型コンバータの基本構成を示す。

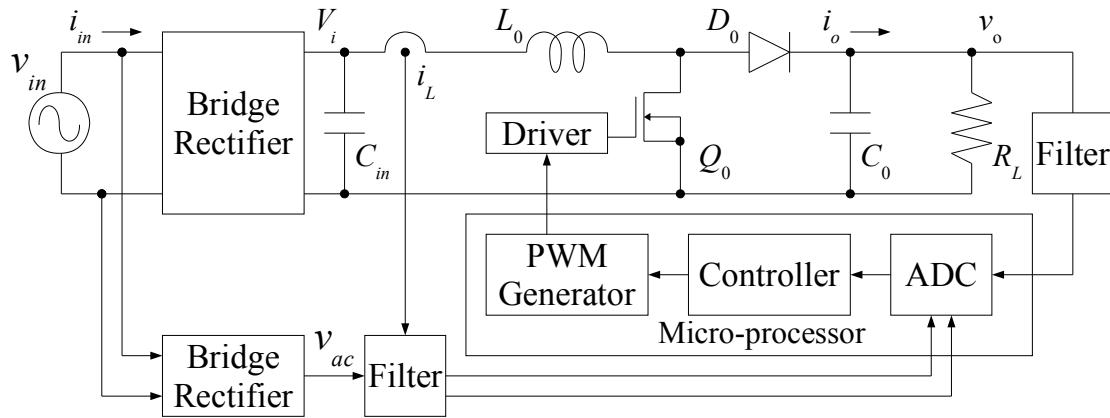


図2.1 PFC昇圧型DC-DCコンバータの構成

ここで、

$v_i$ が交流入力電圧,  $i_{ac}$ が交流入力電流,  $C_i$ は平滑コンデンサ,

$V_i$ は全波整流・平滑された直流の入力電圧,  $V_{ac}$ は全波整流された入力電圧,

$v_o$ は出力電圧,  $i_L$ はインダクタ電流,  $Q_0$ はメインスイッチ,

$\mu$ はデューティ比,  $L_0$ はインダクタ,  $D_0$ はダイオード,

$C_0$ は負荷キャパシタンス,  $R_L$ は負荷抵抗,  $i_o$ は出力電流,

である。本研究では、それぞれ、

$$\begin{array}{llll} v_i=100[\text{VAC}], & V_i=140[\text{VDC}], & C_i=1[\mu\text{F}], & v_o=385[\text{V}], \\ C_0=940[\mu\text{F}], & R_L=282[\Omega], & R_0=1.8[\Omega], & L_0=150[\mu\text{H}], \\ & & & D_0 : \text{SF20L60U}, \quad Q_0 : 2SK2730 \end{array}$$

を用いた。またこの時の定格負荷  $W_{max}$  は  $W_{max} = \frac{v_o^2}{R_L} = 525.620567375 \dots \approx 526 [\text{W}]$  である。

出力電圧・インダクタ電流・入力電圧の全波整流波形は、それぞれフィルタを介してマイクロコントローラ内の A/D コンバータへ入力され、デジタル値に変換される。そしてデジタルコントローラで制御演算され、PWM 波形が生成される。PWM 波がドライバ IC に入力され、昇圧型コンバータの MOS-FET をスイッチングする。時比率と入力電圧と負荷によって、出力電圧とインダクタ電流が決定される。

次に、サンプリング周波数と PWM スイッチング周波数について述べる。サンプリング周波数は、高ければ高いほど、細かなデータをとり制御ができるため、制御性能を上げることができるが、3 章で述べるように、操作量は PWM スイッチング周波数より早いタイミングで更新することができないため、サンプリング周波数=スイッチング周波数とすることが一般的である。PFC 昇圧型コンバータにおいては、大電力を扱うため、PWM 動作により放射される EMC ノイズが大きく、また電力線を介して他の機器に影響を与えるため、DC-DC コンバータ等に比べ、スイッチング周波数を低めにして、通信に使用される周波数における高調波電流・EMC ノイズを少なくする、といった対策が取られる。実際には、例えば、情報処理装置等電波障害自主規制 (VCCI) に定められている、情報技術装置の電源ポート伝導妨害波の許容値の下限周波数が 150[kHz] であるなど、150[kHz] を境に電波障害等の規制が存在するため、スイッチング周波数を、それ以下に設定する。一般のアナログ IC では、スイッチング周波数の 2 次高調波が 150[kHz] 以下になるように、スイッチング周波数が 65[kHz] 程度であるものが多い。

今回は、スイッチング周波数  $F_{sw}$  を  $F_{sw}=100[\text{kHz}]$  とし、サンプリング周波数  $F_s$  も同様に  $F_s=100[\text{kHz}]$  とした。また、このときのサンプリング周期は、 $T_s = \frac{1}{F_s} = 10 [\mu\text{s}]$  である。

## 2.2 制御対象のモデル化

### 2.2.1 状態平均化法による制御対象の数学モデル化

図 2.1 で示された制御対象の昇圧型 DC-DC コンバータ部を単純なモデル図で表すと、図 2.2 のようになる。

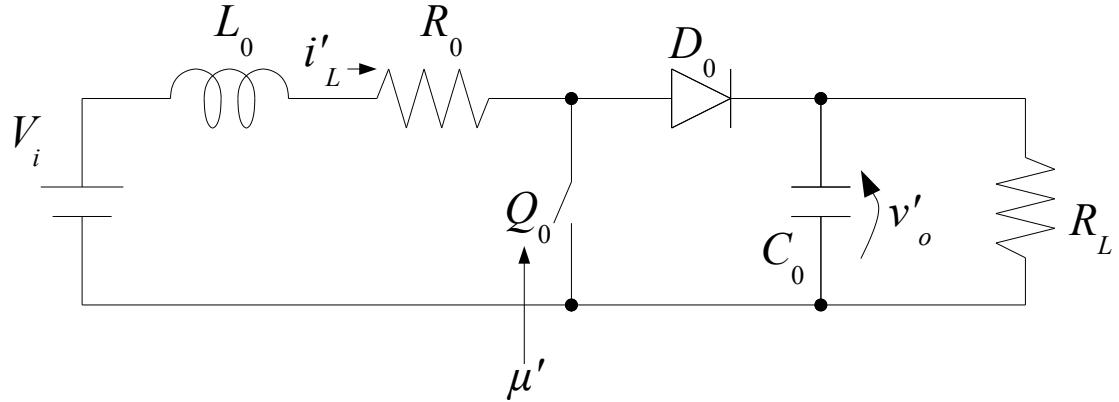


図 2.2 制御対象の回路モデル

ここで、 $R_0$  はインダクタ  $L_0$  とメインスイッチ  $Q_0$  の等価直列抵抗である。本研究では、 $R_0=1.8[\Omega]$ とした。

$\mu'$  はデューティ比で、 $\mu'$  は  $\mu'$  の平均値  $\mu_s$  と  $\mu'$  の微小変動  $\mu$  の和で表される

$$\mu' = \mu_s + \mu \quad (2.1)$$

$v'_o$  と  $i'_L$  についても同様に、平均値  $V_s$ 、 $I_s$  と、微小変動  $v_o$  と  $i_L$  の和である。

$$v'_o = V_s + v_o \quad (2.2)$$

$$i'_L = I_s + i_L \quad (2.3)$$

ここで、これらの変数と状態平均化法を用いて、図 2.2 の回路モデルから、状態平均化モデルを求めるとき、以下の式となる。

$$\frac{d}{dt} \begin{bmatrix} v'_o \\ i'_L \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_L C_0} & \frac{1}{C_0} \\ -\frac{1}{L_0} & -\frac{R_0}{L_0} \end{bmatrix} \begin{bmatrix} v'_o \\ i'_L \end{bmatrix} + \begin{bmatrix} \frac{V_i}{L_0} \\ \frac{1}{L_0} \end{bmatrix} + \begin{bmatrix} v'_o \\ i'_L \end{bmatrix} \left( -\frac{1}{C_0} \right) \mu' \quad (2.4)$$

### 2.2.2 制御対象の静特性

さて、この状態平均化モデルから、制御対象の平衡点、動作点を導出する。動作点方程式は、電圧・電流・デューティ比の時間変化がない、すなわち、 $v_o$ ,  $i_L$ ,  $\mu$  が零となるという条件から求められる。静特性を決める動作点方程式は、以下となる。

$$V_s = \frac{1}{1 + \frac{1}{(1 - \mu_s)^2} \frac{R_0}{R_L}} \frac{1}{1 - \mu_s} V_i \quad (2.5)$$

$$I_s = \frac{1}{R_L} \frac{V_s}{1 - \mu_s} \quad (2.6)$$

等価直列抵抗、負荷抵抗、入力電圧とデューティ比の平均値が、動作点を定めるパラメータとなる。図 2.3 に、本研究で用いた実際の PFC 昇圧型コンバータの出力電圧の静特性の実測値を示す。この図から、入力のデューティ比に対して、出力電圧は非線形であり、また、負荷抵抗の値によって、その特性は大きく変化することがわかる。これは、インダクタ電流についても同様である。

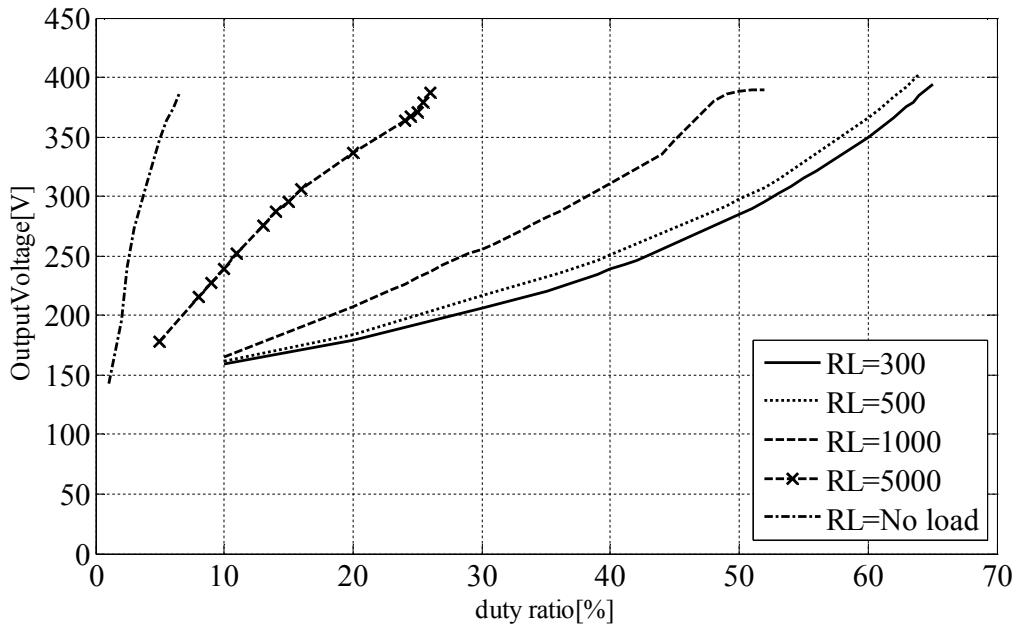


図 2.3 本研究で用いた PFC 昇圧型コンバータの静特性

### 2.2.3 制御対象の動作点近傍における動特性

続いて、動作点近傍における、微小変動に対する、線形近似された状態方程式を導出する。動作点近傍においては、平均値  $V_s$ ,  $I_s$ ,  $\mu_s$  はそれぞれ動作点を定める定数であり、微小変動  $v_o$ ,  $i_L$ ,  $\mu$  が変数となる。動特性である、動作点近傍の線形状態方程式は、以下となる。

$$\begin{aligned}\dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t)\end{aligned}\tag{2.7}$$

ただし、第1状態変数が  $v_o$  の時、

$$A_c = \begin{bmatrix} -\frac{1}{R_L C_0} & \frac{1-\mu_s}{C_0} \\ -\frac{1-\mu_s}{L_0} & -\frac{R_0}{L_0} \end{bmatrix}, \quad B_c = \begin{bmatrix} -\frac{I_s}{C_0} \\ \frac{V_s}{L_0} \end{bmatrix}$$

$$C_c = [1 \quad 0], \quad x(t) = \begin{bmatrix} v_o(t) \\ i_L(t) \end{bmatrix}, \quad u(t) = \mu(t)$$

である。

式(2.7)より、昇圧型コンバータの初期値や応答特性は、動作点を定めるパラメータに大きく依存する。また、負荷抵抗の変化や、温度変化、経年劣化等によるインダクタやキャパシタの値の変化は、 $A_c$  行列と  $B_c$  行列の変化であり、昇圧型コンバータの応答特性を変化させる原因となる。図 2.4・図 2.5 に、実際の PFC 昇圧型コンバータにおける動特性として、4 章で決定する、各動作点における、ステップ応答特性の実測値とボード線図のシミュレーション結果を示す。これらの結果から、負荷抵抗や出力電圧の平均値などが、昇圧型コンバータの応答特性を大きく変化させていることがわかる。

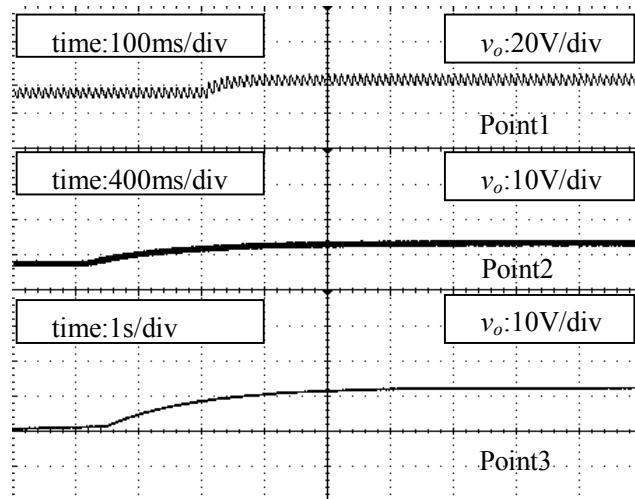
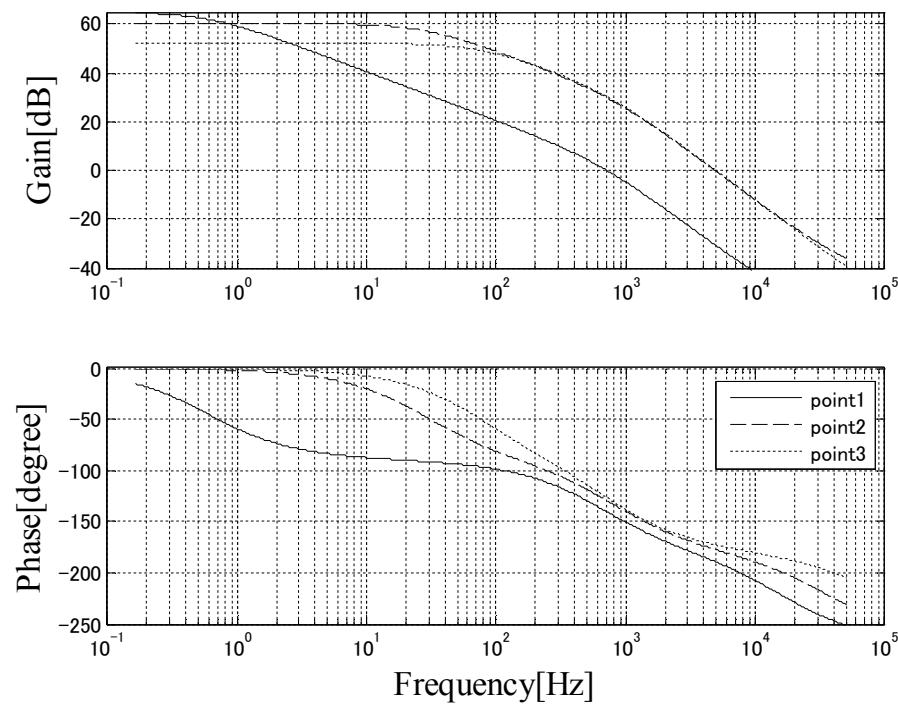
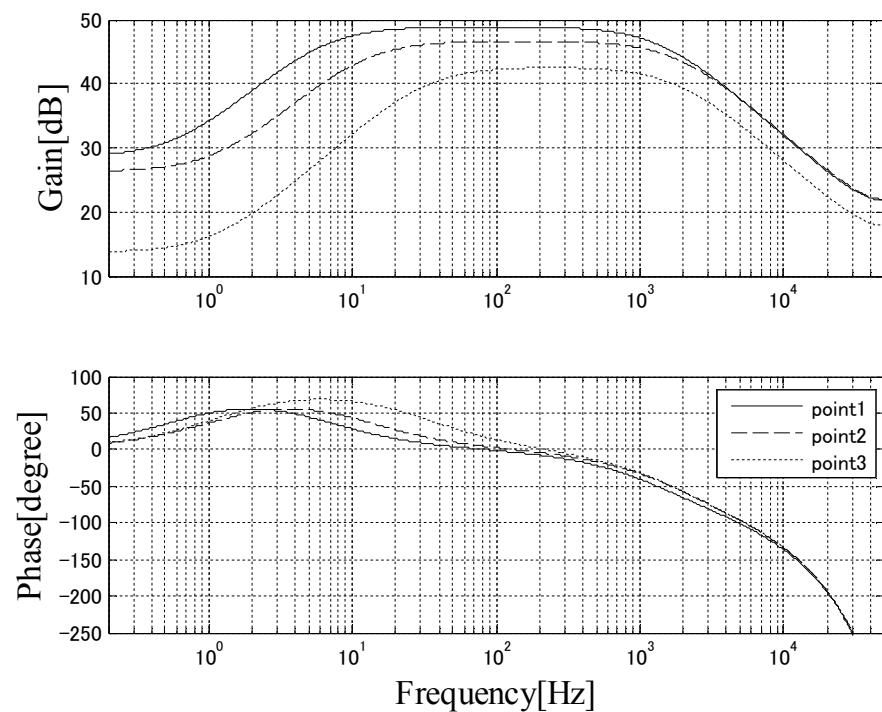


図 2.4 本研究で用いた PFC 昇圧型コンバータの各動作点におけるステップ応答



(a)操作量入力から出力電圧までのボード線図



(b)操作量入力からインダクタ電流までのボード線図

図 2.5 本研究で用いた PFC 昇圧型コンバータのボード線図

こういった、パラメータ変化による、静特性・動特性の変化は、詳しくは4章で述べるが、式(2.8)に示すような、制御対象の入力と出力に加えられる、等価外乱  $q_u$  と  $q_y$  に置き換えることができる。

$$\begin{aligned}\dot{x}(t) &= A_c x(t) + B_c u(t) + q_u(t) \\ y(t) &= C_c x(t) + q_y(t)\end{aligned}\tag{2.8}$$

制御系を設計する際は、パラメータの変動による影響を減らし、ロバストにするため、この等価外乱から出力までの伝達関数のゲインを出来る限り小さくすることが重要である。

## 2.3 状態方程式の離散化

デジタル制御系を構築するために、制御対象の状態方程式を離散化し、離散系の差分方程式を得る。式(2.7)のシステムに於いて、入力  $u(t)$  をサンプリング時間  $T_s$  で零次ホールドした信号  $u(k)$  が状態方程式に入力されるとする。時刻  $t_0$  の時の状態  $x(t_0)$  及び入力  $u(t_0)$  を初期値とした時の、式(2.7)式に示す状態方程式の解は次式で与えられる。

$$x(t) = e^{A_c(t-t_0)} x(t_0) + \int_0^t e^{A_c(t-\tau)} B_c u(\tau) d\tau\tag{2.9}$$

ただし、 $e^{At} = I + At + \frac{A^2 t^2}{2!} + \cdots + \frac{A^n t^n}{n!} + \cdots = \sum_{n=0}^{\infty} \frac{(At)^n}{n!}$  ( $I$  は単位行列) である。

ここで、 $e^{A(t-t_0)}$  は、状態ベクトル  $x(t_0)$  を  $x(t)$  に変換するものである。これを用いて、零次ホールドにて離散化した時の差分方程式を導く。上式より、時刻  $(k+1)T_s$  時の状態  $x_d(k+1)$  を時刻  $kT_s$  時の状態  $x_d(k)$  及び入力  $u_d(k)$  を用いて表すと、次式となる。

$$x_d(k+1) = e^{A_c T_s} x_d(k) + \int_{kT_s}^{(k+1)T_s} e^{A_c((k+1)T_s-\tau)} B_c u_d(k) d\tau\tag{2.10}$$

ここで  $(k+1)T_s - \tau = \eta$  をすると、次式が得られる。

$$x_d(k+1) = e^{A_c T_s} x_d(k) + \int_0^{T_s} e^{A_c \eta} B_c u_d(k) d\eta = A_d x_d(k) + B_d u_d(k)\tag{2.11}$$

上式は連続系と同様に、右辺第一項は時刻  $kT_s$  時の値  $x_d(k)$  によって決まる自由応答であり、第二項は入力  $u_d(k)$  による強制応答である。 $A_d$ 、 $B_d$  はテイラー展開によって次式のように求められる。

$$A_d = e^{A_c T_s} = \sum_{n=0}^{\infty} \frac{(A_c T_s)^n}{n!}\tag{2.12}$$

$$B_d = \int_0^{T_s} e^{A_c \eta} B_c d\eta = A_c^{-1} \left[ \sum_{n=0}^{\infty} \frac{(A_c T_s)^n}{n!} - I \right] B_c\tag{2.13}$$

以上のように、式(2.11)～式(2.13)を用いて、連続系状態方程式の離散化ができる。

また、出力される状態は変わらないものとすると、 $C$ 行列は変わらず  $C_d = C_c$  であり、出力方程式を零次ホールドで離散化すると、次式となる。

$$y(k) = C_d x_d(k) \quad (2.14)$$

## 2.4 遅れ時間の考慮と制御対象の伝達関数

デジタル制御では、A/D コンバータでサンプリングを行った後、制御演算を行って、PWM 信号のデューティ比を更新するまでには、A/D 変換時間や制御演算時間等による遅れ時間が存在する。遅れ時間は制御系に影響をおよぼすため、これを考慮して制御系設計を行う。遅れ時間を、1サンプル周期よりも短い入力むだ時間  $L_d$  ( $L_d \leq T_s$ ) とし、入力  $u(t)$  をサンプル周期  $T_s$  で零次ホールドし、時間  $L_d$  後に更新される信号  $u_d(k)$  が状態方程式に入力されるとする。

まず、 $kT_s$  から  $kT_s + L_d$  の間は、 $u_d(k)$  の前サンプルの状態( $(k-1)T_s$  でサンプルされた状態)  $\xi_1(k)$  が入力されている。したがって、式(2.12)から次式が得られる。

$$x_d(kT_s + L_d) = e^{A_c L_d} x_d(kT_s) + \int_0^{L_d} e^{A_c \tau} B_c \xi_1(kT_s) d\tau \quad (2.15)$$

次に、 $kT_s + L_d$  から  $(k+1)T_s$  までの間は、 $u_d(k)$  の  $kT_s$  でサンプルされた状態  $\xi_2(k)$  が入力される。従って次式が得られる。

$$x_d((k+1)T_s) = e^{A_c(T_s - L_d)} x_d(kT_s + L_d) + \int_0^{T_s - L_d} e^{A_c \tau} B_c \xi_2(kT_s) d\tau \quad (2.16)$$

上 2 式から、次式が得られる。

$$\begin{aligned} x_d((k+1)T_s) &= e^{A_c T_s} x_d(kT_s) \\ &+ e^{A_c(T_s - L_d)} \int_0^{L_d} e^{A_c \tau} B_c \xi_1(kT_s) d\tau + \int_0^{T_s - L_d} e^{A_c \tau} B_c \xi_2(kT_s) d\tau \end{aligned} \quad (2.17)$$

よって、入力むだ時間  $L_d$  ( $L_d \leq T_s$ ) を考慮した差分方程式は次式となる。

$$\begin{aligned} x_d(k+1) &= e^{A_c T_s} x_d(k) \\ &+ e^{A_c(T_s - L_d)} \int_0^{L_d} e^{A_c \tau} B_c \xi_1(k) d\tau + \int_0^{T_s - L_d} e^{A_c \tau} B_c \xi_2(k) d\tau \end{aligned} \quad (2.18)$$

$$\xi_1(k+1) = \xi_2(k) \quad (2.19)$$

但し、式(2.18)の右辺第二項、第三項は、テイラー展開により、次式となる。

$$e^{A_c(T_s - L_d)} \int_0^{L_d} e^{A_c \tau} B_c d\tau = \sum_{n=0}^{\infty} \frac{[A_c(T_s - L_d)]^n}{n!} A_c^{-1} \left[ \sum_{n=0}^{\infty} \frac{(A_c L_d)^n}{n!} - I \right] B_c \quad (2.20)$$

$$\int_0^{T_s-L_d} e^{A_c \tau} B_c d\tau = A_c^{-1} \left[ \sum_{n=0}^{\infty} \frac{[A_c(T_s - L_d)]^n}{n!} - I \right] B_c \quad (2.21)$$

ここで、状態  $x_d(k)$  と  $\xi_1(k)$  を合わせて、新たな状態  $x_{dt}(k) = [v_o(k) \ i_L(k) \ \xi_1(k)]^T$  とし、 $\xi_2(k)$  を新たな入力  $v(k)$  とすると、式(2.18)・式(2.19)は、次式となる。

$$x_{dt}(k+1) = A_{dt}x_{dt}(k) + B_{dt}v(k) \quad (2.22)$$

ただし、

$$A_{dt} = \begin{bmatrix} e^{A_c T_s} & e^{A_c(T_s - L_d)} \int_0^{L_d} e^{A_c \tau} B_c d\tau \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{dt} = \begin{bmatrix} \int_0^{T_s - L_d} e^{A_c \tau} B_c d\tau \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ 1 \end{bmatrix}$$

である。また、この時、状態  $\xi_1(k)$  は出力に関わらないため、 $C_{dt} = [C_d \ 0]$  となり、出力方程式は次式となる。

$$y(k) = C_{dt}x_{dt}(k) \quad (2.23)$$

以上により、式(2.22)、式(2.23)を用いて、入力むだ時間  $L_d$  を考慮した連続系状態方程式の離散化ができる。この離散化された制御対象に対し、制御系を設計することで、デジタル制御における遅れ時間の悪影響を取り除くことができる。

制御対象の状態方程式を、1入力1出力システムとして、伝達関数で表す場合は、連続系では、式(2.7)のラプラス変換をとり

$$X(s) = (sI - A_c)^{-1}x(0) + (sI - A_c)B_c U(s) \quad (2.24)$$

$$Y(s) = C_c X(s) \quad (2.25)$$

$x(0)=0$  とすれば、伝達関数  $G_c(s)$  は

$$G_c(s) = \frac{Y(s)}{U(s)} = C_c (sI - A_c)^{-1} B_c \quad (2.26)$$

と表される。一方、離散系の場合の差分方程式は、式(2.22)、式(2.23)の z 変換をとると、

$$X_{dt}(z) = (zI - A_{dt})^{-1}x_{dt}(0) + (zI - A_{dt})B_{dt}V(z) \quad (2.27)$$

$$Y_{dt}(z) = C_{dt}X_{dt}(z) \quad (2.28)$$

で表される。 $x(0)=0$  とすれば、パルス伝達関数  $G_{dt}(z)$  を、以下の式で得ることができる。

$$G_{dt}(z) = \frac{Y_{dt}(z)}{V_{dt}(z)} = C_{dt}(zI - A_{dt})^{-1}B_{dt} \quad (2.29)$$

# 第3章 実験基板とディジタルコントローラの製作

## 3.1 マイクロコントローラ

### 3.1.1 マイクロコントローラの概要

マイクロコントローラ(Micro Controller,マイコン)とは、CPU機能とメモリや各種ペリフェラルを内蔵した集積回路を指す。一般には、家電製品や電子機器等の内部に組み込まれていて、それらの制御に利用される。本研究では、マイクロコントローラとして、ルネサスエレクトロニクス社のSH2A-FPU(SH7216)を使用した。SH2A-FPUはルネサスエレクトロニクス社のマイクロコントローラ SH-2 をベースに、スーパースカラ方式を導入して高速化し、浮動小数点演算器を搭載したものである。SH7216は、汎用インバータやUPS等のパワーエレクトロニクスへの応用を考慮され開発されたSH2A-FPUシリーズのマイクロコントローラであり、動作周波数 200[MHz]で 400[MIPS]の演算性能を持つなど、コントローラ系のマイクロコントローラながら十分高速な処理が可能である。また、周辺機能として、PWM出力機能を持つタイマや、12bit 分解能の A/D 変換器を内蔵している。このマイクロコントローラに、コンピュータからデバッグを通してプログラムを書き込むことができる。



図 3.1 マイクロコントローラ(SH7216)が搭載された基板

### 3.1.2 マイクロコントローラの動作

本研究で使用したマイクロコントローラは、分解能  $\sigma_{adc} = 12[\text{bit}]$  の分解能を持つ A/D 変換器を 8ch 搭載している。変換時間は 1chあたり約  $1.1[\mu\text{s}]$  である。今回は、A/D 変換器の電源を  $5[\text{V}]$ としたため、 $0 \sim 5[\text{V}]$ を  $12[\text{bit}]$  のデジタル値に変換する。この A/D 変換に関するゲイン  $K_{adc}$  は

$$K_{adc} = \frac{2^{12}}{5} = 819.2 \quad (3.1)$$

となる。この A/D 変換器を、出力電圧  $v_o$ 、インダクタ電流  $i_L$ 、全波整流された入力電圧  $v_{ac}$  の 3 つの物理量の変換に用いる。変換時間は合計約  $3.3[\mu\text{s}]$  である。また、マイクロコントローラは、条件付きで  $100[\text{MHz}]$  のクロックで動作する PWM 出力機能付きタイマ MTU2S を 3ch 搭載している。このタイマのうち 2ch を使用し、相補 PWM モード動作させることで、 $100[\text{MHz}]$  クロックの PWM パルス出力を行う。さらに、それと同期した A/D 変換の開始処理を行う。今回はサンプリング周波数  $100[\text{kHz}]$  であるため、PWM 分解能  $\sigma_{pwm}$  は、

$$\sigma_{pwm} = \frac{100 \times 10^6}{100 \times 10^3} = 1000 \approx 10 [\text{bit}] \quad (3.2)$$

である。タイマは、振幅  $C_m = \sigma_{pwm}$  のキャリア波と操作量  $u$  とを比較して、PWM 信号を生成すると考える。

このタイマを用いた時の、キャリア波と PWM 波形、A/D 変換と割り込み、デューティ比更新のタイミングは、図 3.2 のようになる。

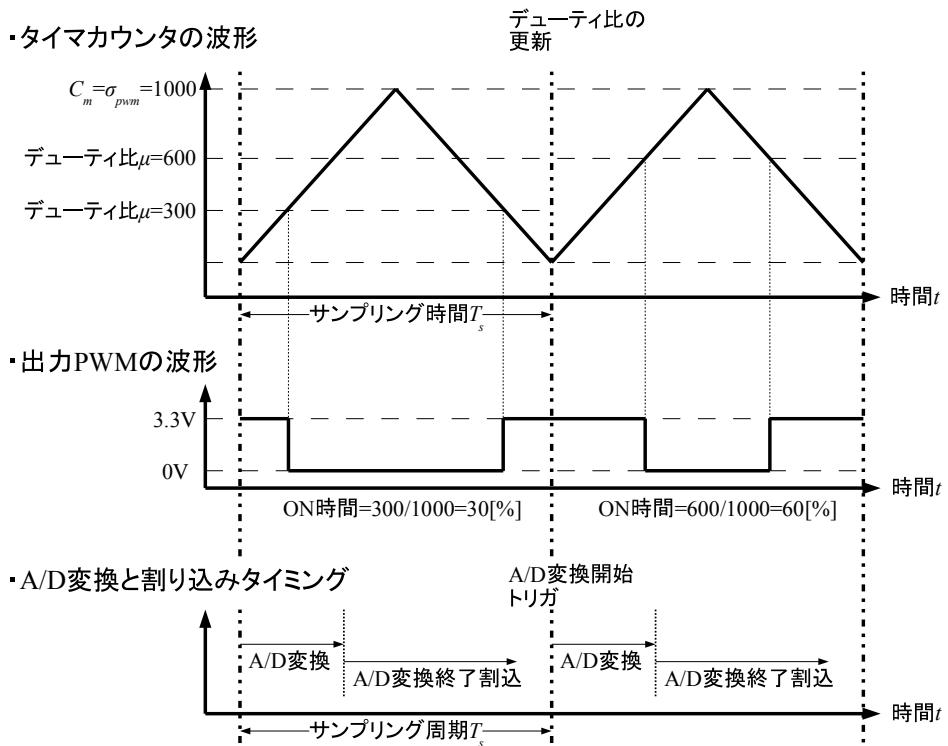


図 3.2 キャリア波と PWM 出力の関係図

一定の周期で制御演算処理を行うため、A/D 変換終了と同時に割り込み処理を行い、その中で制御演算を行う。図 3.2において、A/D 変換を開始してから、終了後に割り込みを行い、制御演算をして、操作量である PWM デューティ比が更新されるまでに、サンプリング周期と等しい  $T_s$  [s] の時間がかかる。今回は制御演算を、この 1 回の割り込みの時間内で終わらせるものとし、2.4 節で述べた、入力むだ時間  $L_d$  は  $L_d=0.99T_s$  とする。

割り込み中の制御演算処理の例を図 3.3 に示す。

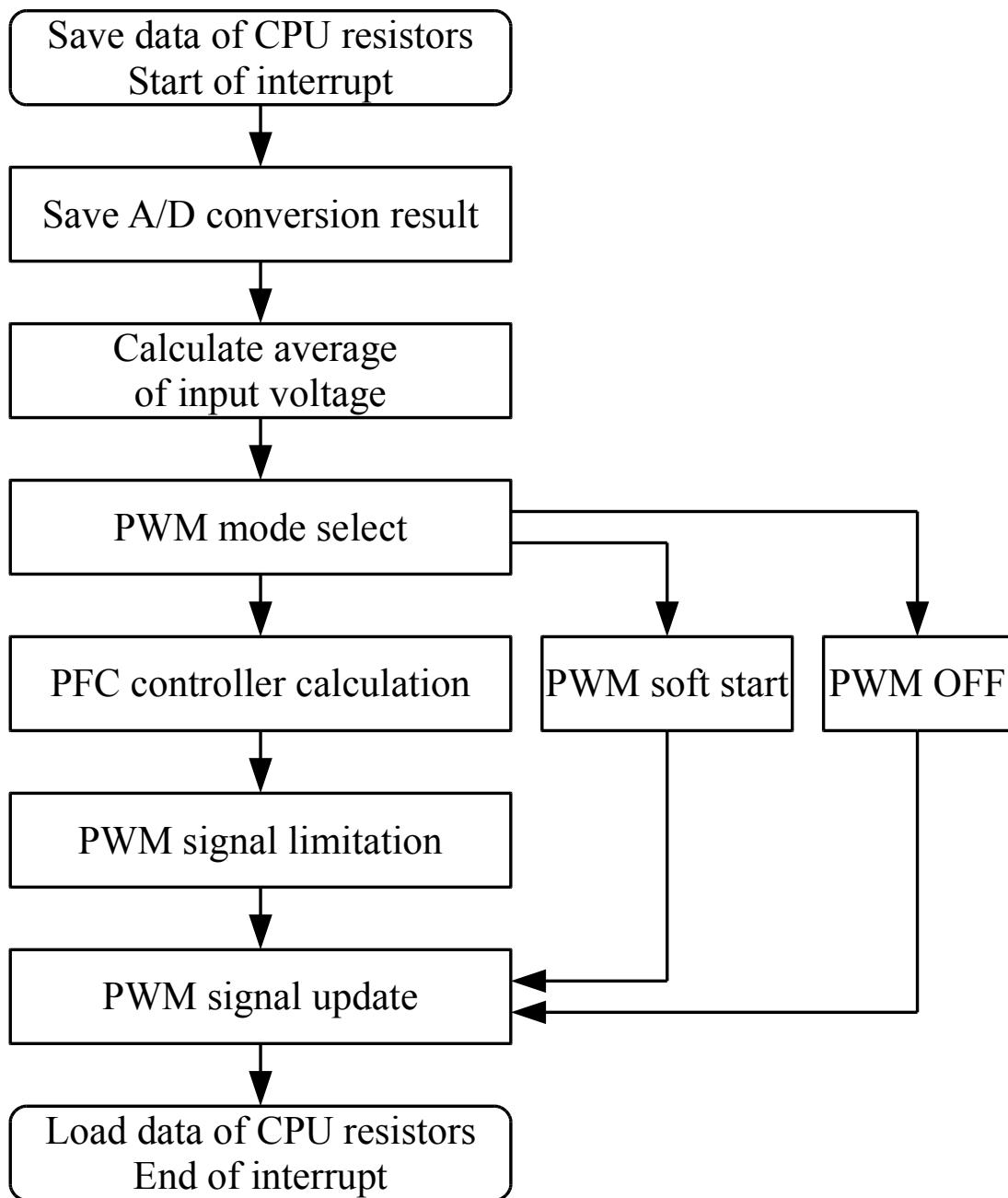


図 3.3 制御ルーチンの例

## 3.2 PFC 昇圧型コンバータ基板

### 3.2.1 PFC 昇圧型コンバータ基板の概要

本研究では、PFC 昇圧型コンバータ基板として、ルネサスエレクトロニクス社のアナログ PFCIC である HA16174 の評価基板(HA16174 EVB)を用いた。この基板は、入力電圧範囲が AC85~AC265[V]、最大出力電圧 DC385[V]で、最大出力電力 500[W]の、AC-DC コンバータである。

この基板の構成図を図 3.3 に示す。パワー系は、ダイオードブリッジによる全波整流回路と、平滑コンデンサ、連続電流モードの昇圧型コンバータで構成されている。制御には、HA16174 が使用されているが、基板は、パワー系の素子が配置されているメインボードと、PFCIC とその周辺回路が配置されているサブボードに分かれており、サブボードを取り外して、SH7216 と適切な接続回路で接続することで、デジタル制御 PFC 昇圧型コンバータとして扱うことができる。

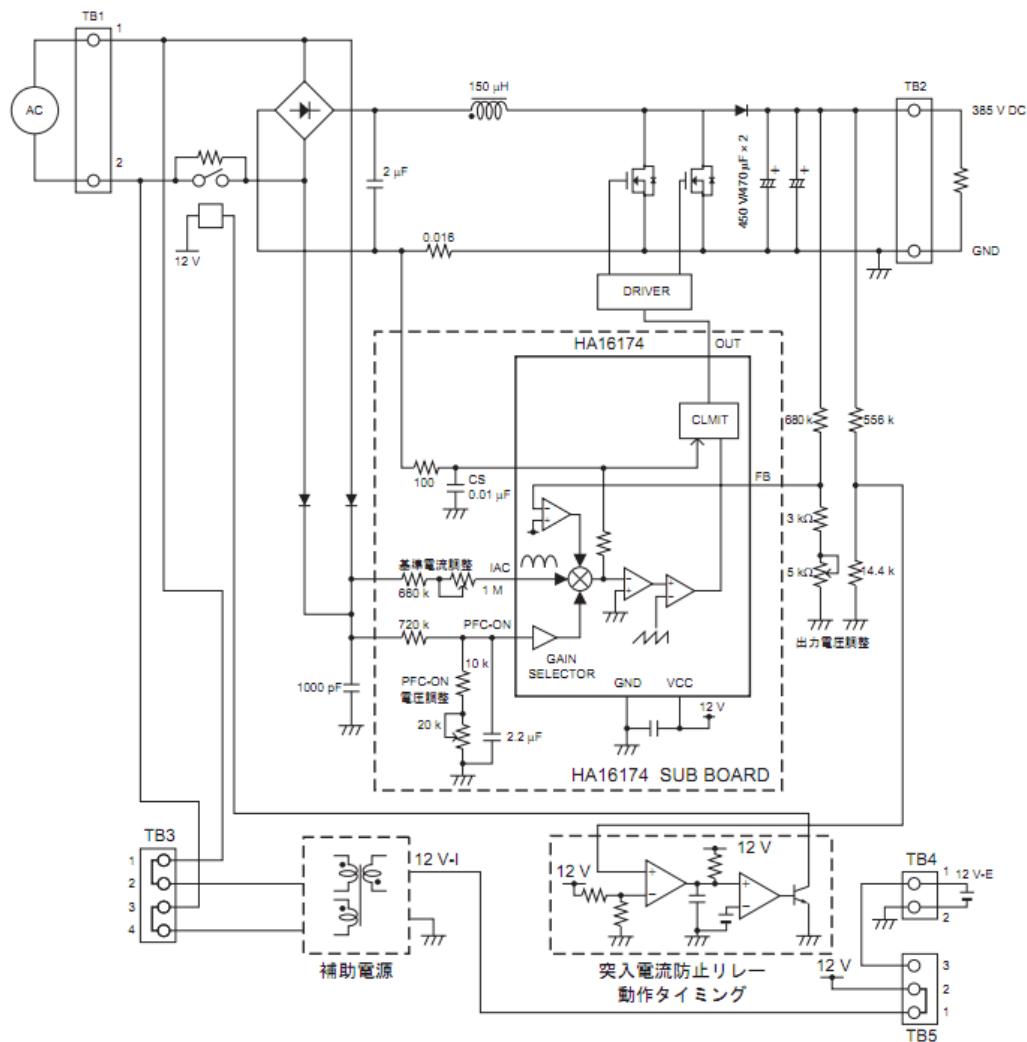


図 3.4 PFC 昇圧型コンバータ基板の構成図

### 3.2.2 PFC 昇圧型コンバータ基板とマイクロコントローラの接続

PFC 昇圧型コンバータのパワー部の電圧や電流は、そのままでは直接マイクロコントローラの A/D コンバータに入力することができないため、適切な電圧に変換して、入力する必要がある。また、マイクロコントローラの出力する PWM 信号も、H の電圧がマイクロコントローラのバス電圧であるため、これを、FET を駆動できる電圧にしなければならない。本研究では、付録 D にあるような接続回路を作成し、コンバータとマイクロコントローラを接続した。

PWM 信号のデューティ比はこの回路で変化することはないが、A/D 変換前の信号は振幅が変化するため、これを DC ゲインとして、考慮する必要がある。2.1 節で述べたように、出力電圧・インダクタ電流・入力電圧の 3 つを A/D 変換器に入力する。

出力電圧は、抵抗で分圧した後、フィルタを介して入力される。この抵抗のゲインとフィルタゲインは、

$$k_{v_{out}} = \frac{4.7}{684.7} \times 1 = 0.00686432\cdots \quad (3.3)$$

である。

インダクタ電流は、電流検出抵抗を介して電圧にされた後、フィルタで増幅されるため、

$$k_{i_L} = 0.0165 \times \frac{3000}{220} = 0.225 \quad (3.4)$$

がゲインとなる。

入力電圧は全波整流され、これも抵抗で分圧された後、フィルタを介して入力されるので、ゲインは、

$$k_{v_{ac}} = \frac{20}{700} \times 1 = 0.02857142\cdots \quad (3.5)$$

となる。

# 第4章 出力電圧のみを制御する近似的 2 自由度制御系の設計法

## 4.1 近似的 2 自由度制御による目標

### 4.1.1 等価外乱

2.2.3 項で述べた、等価外乱について、再度詳しく述べる。図 4.1 に示すフィードバック制御系において、制御器の伝達関数を  $C(z)$ 、制御対象の伝達関数を  $P(z)$  とする。出力は  $y$ 、目標値は  $r$ 、操作量は  $u$  である。ここで、2.2.3 項でも挙げたパラメータ変化による、静特性・動特性の変化により、制御対象の伝達関数が  $P_1(z)$  となった時の操作量を  $u_1$ 、出力を  $y_1$  とする。このような特性変化は、図 4.2 に示すように、制御対象の入出力に外乱  $q_u = u - u_1$  と  $q_y = y - y_1$  を加えることと等価である。したがって、制御対象の特性変化は、等価外乱  $Q = [q_u, q_y]$  に置き換えることができる。

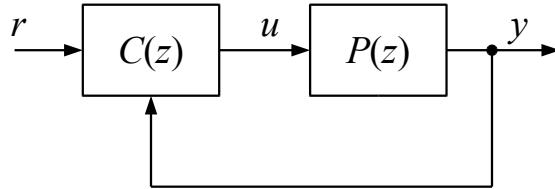


図 4.1 フィードバック制御系

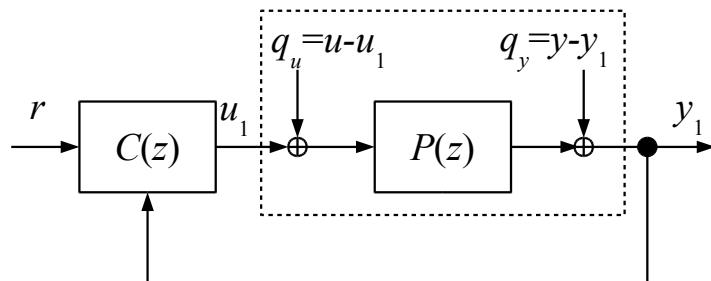


図 4.2 等価外乱を含むフィードバック制御系

近似的 2 自由度制御系の設計では、目標値  $r$  から出力  $y$  の特性と、外乱  $Q$  から出力  $y$  までの特性を、近似的に 2 自由度系とし、それぞれ独立に設計できるようにすることで、制御系設計を簡易にしている。 $r \rightarrow y$  の特性の帯域幅をより広くし、 $Q \rightarrow y$  の特性をより低感度にすること、パラメータを設定することが、近似的 2 自由度制御系の設計目標である。

### 4.1.2 離散系制御対象

昇圧型コンバータの、インダクタ電流と出力電圧両方を制御する前に、まず、図 4.3 のように、出力電圧のみをフィードバックし制御する、近似的 2 自由度制御器を設計する。出力電圧のみを制御するとき、第 1 状態変数を  $v_o$  にとり、出力  $y$  として  $v_o$  のみを得るものとすると、制御対象の動作点近傍の状態方程式は、式(2.7)となる。また、この式を遅れ時間  $L_d$  を考慮して離散化すると、式(2.22)となる。今回は、後述のとおり、電流フィードバックを電圧フィードバックに変換するため、1 サンプルの周期遅れ要素を結合し、図 4.4 の系を構成する。図 4.4 の系は、次式である。

$$\begin{aligned} x_{dw}(k+1) &= A_{dw}x_{dw}(k) + B_{dw}w(k) \\ y(k) &= C_{dw}x_{dw}(k) \end{aligned} \quad (4.1)$$

$$\text{但し, } A_{dw} = \begin{bmatrix} A_{dt} & B_{dt} \\ 0 & 0 \end{bmatrix}, \quad B_{dw} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_{dw} = \begin{bmatrix} C_{dt} & 0 \end{bmatrix}, \quad x_{dw} = \begin{bmatrix} x_{dt}(k) \\ \xi_2(k) \end{bmatrix} = \begin{bmatrix} v_o(k) \\ i_L(k) \\ \xi_1(k) \\ \xi_2(k) \end{bmatrix} = \begin{bmatrix} v_o(k) \\ i_L(k) \\ \xi_1(k) \\ v(k) \end{bmatrix}$$

である。4 章では、これを制御対象とし、近似的 2 自由度制御器を適用していく。

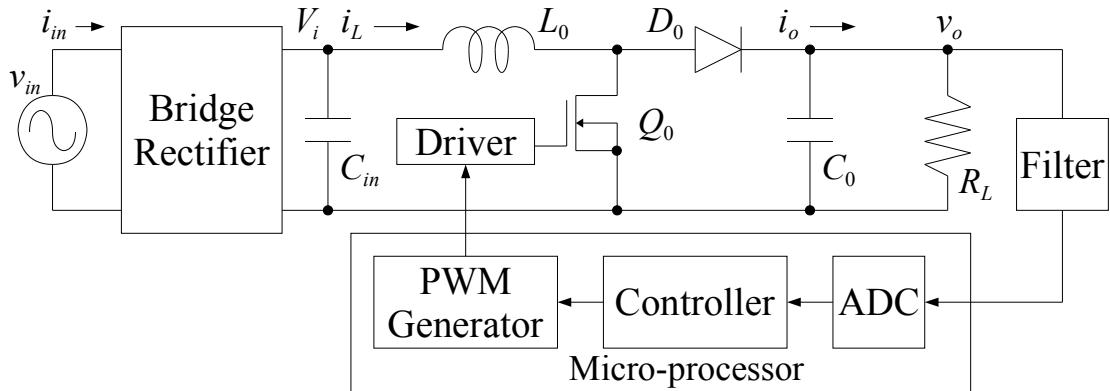
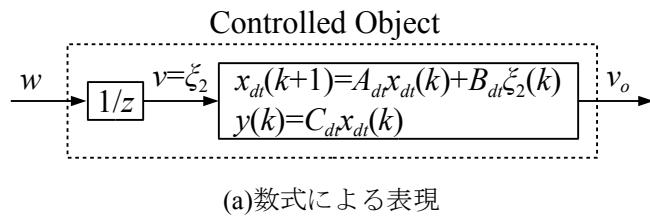
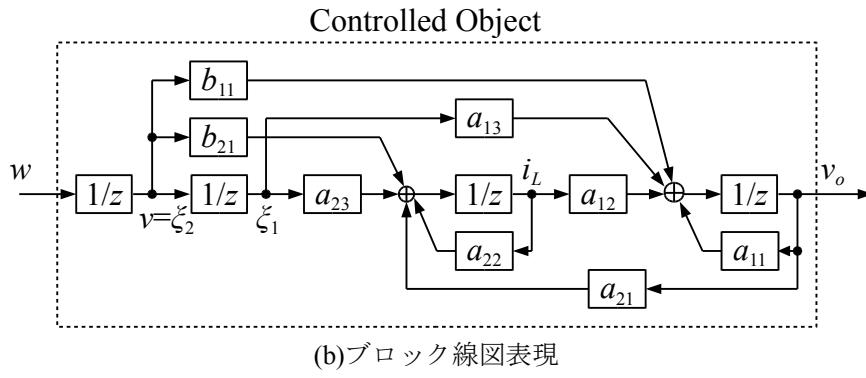


図 4.3 出力電圧のみをフィードバックする昇圧型コンバータ



(a) 数式による表現



(b) ブロック線図表現

図 4.4 入力無駄時間  $L_d$  と 1 サンプル周期遅れを持つ制御対象

#### 4.1.3 昇圧型 DC-DC コンバータの仕様と動作点の決定

昇圧型コンバータは、前述のとおり、目標値の変化や負荷変化等のパラメータ変化により、動作点が移動し、その特性が大きく変化する。負荷近くの降圧コンバータと違い、立ち上がり時間等に厳しい制約があるわけではないが、動作点の変化に対しても、同じ立ち上がり応答を保ち、かつオーバーシュートを生じないステップ応答特性が重要である。また、同様にして動作点の変動の大きな要因となる、負荷急変応答に対する出力電圧変動の特性が重要である。よって、以下の仕様(1)～(4)を満たすディジタル制御昇圧型 DC-DC コンバータを設計・製作する。

- (1)入力電圧は 100[VAC]、出力電圧は 240[VDC]から 385[VDC]まで変化する
- (2)ステップ応答は負荷抵抗が 300[Ω]から 5[kΩ]の範囲で変わらないものとする
- (3)ステップ応答におけるオーバーシュートはすべての負荷と出力電圧範囲において 10[%]以内とする。
- (4)負荷が 300[Ω]から 5[kΩ](またはその反対)に急変した時の出力電圧変動は 5[%]以下とする。

これらの仕様を満たすように制御器を設計する際、評価すべき動作点を、負荷抵抗が最小と最大の点、出力電圧が最小と最大の点から、以下のように 3 つ定める。

- ①出力電圧 385[VDC]、負荷抵抗 5[kΩ]
- ②出力電圧 385[VDC]、負荷抵抗 300[Ω]
- ③出力電圧 240[VDC]、負荷抵抗 300[Ω]

これら 3 つのすべての動作点で、同じステップ応答で安定な制御器を設計することが出来れば、仕様で定められるすべての動作範囲で、安定であり、また  $Q$  から  $y$  までの伝達特性を低感度にすることが出来れば、動作点間の変動の影響を抑制できる。以下では、これらの 3 つの動作点での特性を考慮し、外乱の影響を抑制する簡単な昇圧型 DC-DC コンバータの制御器の設計法を提案する。

## 4.2 昇圧型 DC-DC コンバータの近似的 2 自由度制御器の設計

### 4.2.1 状態フィードバック則の適用

制御対象に状態フィードバック則を適用し、伝達関数  $W_{ry}(z)$ を持つモデルマッチングシステムを構築する。式(4.1)の離散時間制御対象に状態フィードバック則  $w(k) = -Fx_{dw} + G(z-H_4)r$  を適用すると、図 4.5 のようになる。 $r$  から  $y$  までの伝達関数は次式のように表される。

$$W_{ry}(z) = \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-H_3)}{(z-H_3)} \frac{(z-H_4)}{(z-H_4)} \frac{(z-n_1)}{(1-n_1)} \frac{(z-n_2)}{(1-n_2)} \quad (4.2)$$

このシステムは状態フィードバックであるため、システムの  $r$  から  $y$  の伝達関数の極は極配置方で任意の値に指定できる。 $n_1$  と  $n_2$  は離散時間制御対象の零点である。また、システムの定常時に  $r$  と  $y$  を等しくし、新たに追加した  $\zeta_2$  による極  $H_4$  の影響を打ち消すために、フィードフォワード  $G(z-H_4)$  が追加されている。 $G$  は  $r$  から  $y$  までの直流ゲインを 1 とするためのゲインで、以下のように指定できる。

$$G = \frac{(1-H_1)(1-H_2)(1-H_3)}{(1-n_1)(1-n_2)} \times \frac{1}{C_{dw}A_{dw}B_{dw}} \quad (4.3)$$

ここで、 $|n_1|, |n_2| \gg 1$  とする。 $H_1$  を代表根、即ち、 $|H_3| \ll |H_1|, |H_2| \ll |H_1|$  となるように指定すると、 $W_{ry}(z)$  は次式のような 1 次モデルに近似される。

$$W_{ry}(z) \approx W_m(z) = \frac{1-H_1}{z-H_1} \quad (4.4)$$

$n_1, n_2$  は、 $|n_1|, |n_2| \gg 1$  であるため無視する。この目標特性  $W_{ry}(z) \approx W_m(z)$  は仕様(2), (3)を満たすように指定する。

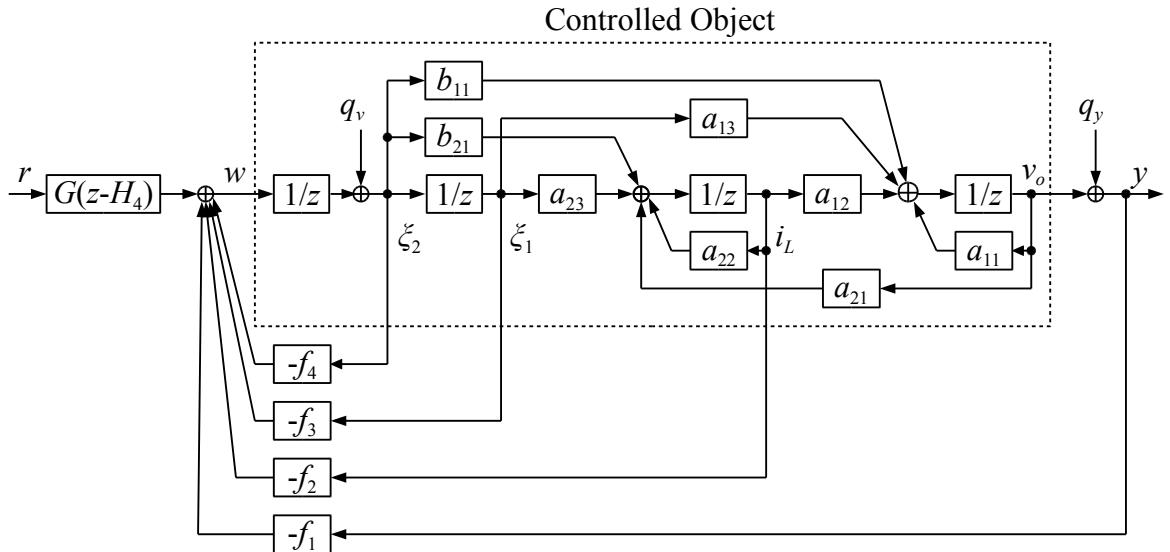


図 4.5 離散時間制御対象に状態フィードバックを備えたモデルマッチングシステム

#### 4.2.2 モデルマッチングシステムの等価変換

本研究で用いる昇圧型コンバータは、交流入力を整流して直流入力電圧を得ているため、図4.5のシステムにおける、インダクタ電流が定常状態で大きく変化する。そのため、インダクタ電流を必要としないモデルマッチングシステムが望ましい。そこで、1サンプル遅れ $\xi_2$ を追加し、電流フィードバックを電圧フィードバックに置き換える。図4.5のシステムを等価変換し、電流フィードバック無しで実現できるシステムにする。

出力電圧の差分方程式を $z$ 変換すると次式が得られる。

$$zv_o(z) = a_{11}v_o(z) + a_{12}i_L(z) + a_{13}\xi_1(z) + b_{11}\xi_2(z) \quad (4.5)$$

これを式変形すると、

$$i_L(z) = \frac{z}{a_{12}}v_o(z) - \frac{a_{11}}{a_{12}}v_o(z) - \frac{a_{13}}{a_{12}}\xi_1(z) - \frac{b_{11}}{a_{12}}\xi_2(z) \quad (4.6)$$

となるため、図4.5は、電流フィードバックを使用しない、図4.6のシステムに置き換えられる。

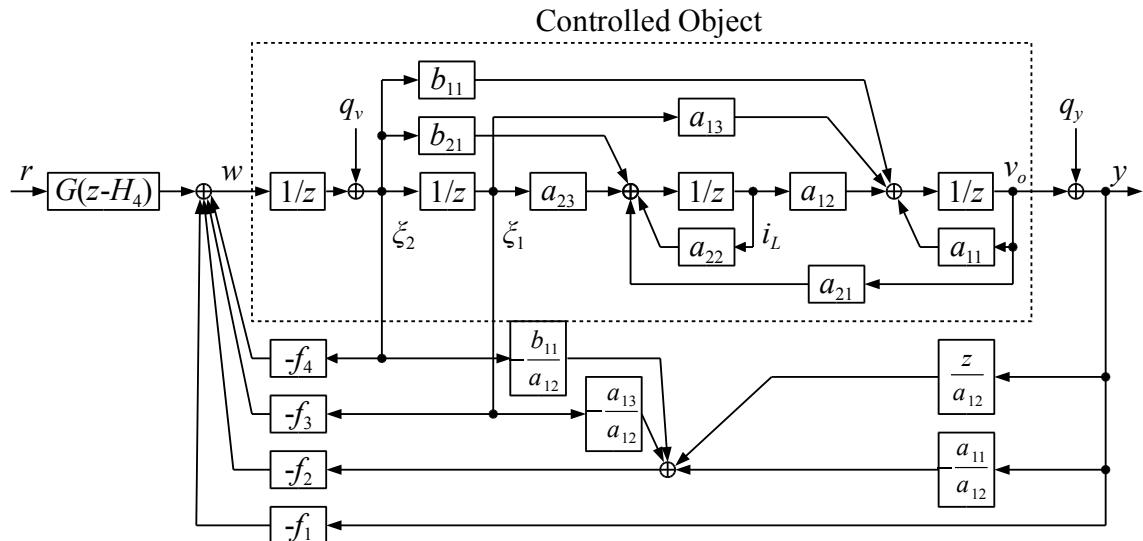


図4.6 電流フィードバックを他の状態フィードバックに置き換えたモデル

しかし、ここには、分子に $z$ が含まれる項があるため、これを等価変換する必要がある。入力 $w$ の差分方程式を $z$ 変換すると、次式が得られる。

$$w(z) = -f_1v_o(z) - f_2i_L(z) - f_3\xi_1(z) - f_4\xi_2(z) + G(z - H_4)r \quad (4.7)$$

また、

$$w(z) = z\xi_2(z) \quad (4.8)$$

である。式(4.7)に、式(4.6)、(4.8)を代入すると、次式となる。

$$\begin{aligned}
z\xi_2(z) &= v_o \left( -f_1 + \frac{f_2 a_{11}}{a_{12}} - \frac{f_2}{a_{12}} z \right) + \xi_1 \left( -f_3 + \frac{f_2 a_{13}}{a_{12}} \right) + \xi_2 \left( -f_4 + \frac{f_2 b_{11}}{a_{12}} \right) + G(z - H_4)r \\
&= v_o (ff_1 + ff_2 z) + ff_3 \xi_1 + ff_4 \xi_2 + G(z - H_4)r
\end{aligned} \tag{4.9}$$

$$\xi_2(z) = \frac{1}{z - ff_4} (ff_1 v_o + ff_2 z v_o + ff_3 \xi_1 + G z r - G H_4 r) \tag{4.10}$$

$$\text{ただし, } ff_1 = -f_1 + \frac{f_2 a_{11}}{a_{12}}, \quad ff_2 = -\frac{f_2}{a_{12}}, \quad ff_3 = -f_3 + \frac{f_2 a_{13}}{a_{12}}, \quad ff_4 = -f_4 + \frac{f_2 b_{11}}{a_{12}}$$

である。ここで、

$$\frac{ff_2 z v_o + G z r}{z - ff_4} = \frac{(z - ff_4)(ff_2 v_o + Gr) + ff_2 ff_4 v_o + G ff_4 r}{z - ff_4} \tag{4.11}$$

であることを考慮すると、式(4.10)は次式となる。

$$\xi_2(z) = \frac{1}{z - ff_4} (ff_1 v_o + ff_2 ff_4 v_o + ff_3 \xi_1 + G ff_4 r - G H_4 r) + ff_2 v_o + Gr \tag{4.12}$$

$\frac{1}{z - ff_4}$  は、図 4.7 のブロック線図で表されることから、図 4.6 のシステムは、図 4.8 のシステムに等価変換される。

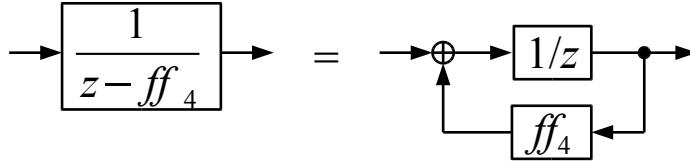


図 4.7 ブロック線図の等価変換

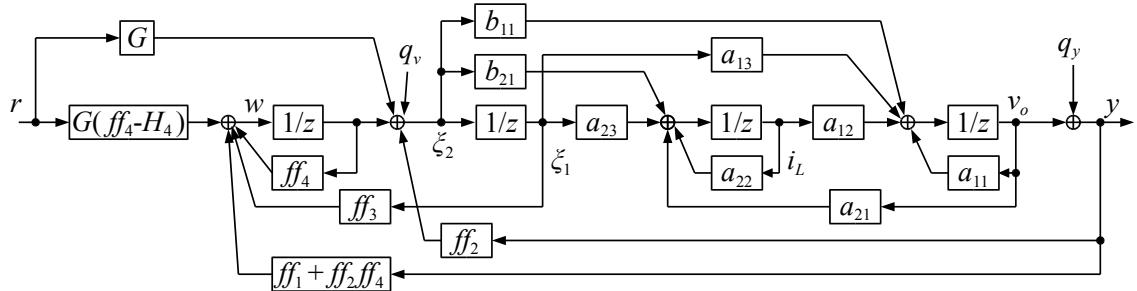


図 4.8 電圧フィードバックのみを用いたモデルマッチングシステム

### 4.2.3 逆システムとフィルタを用いたロバスト制御器

図4.8のシステムの外乱  $Q$  を  $Q = [q_v \ q_y]$  とし, 外乱  $Q$  から出力  $y$  までの伝達関数を  $W_{Qy}(z) = [W_{qwy}(z) \ W_{qyy}(z)]$  とし, 逆システムとフィルタを結合したシステムを図4.9に示す. ここで, フィルタ  $K(z)$  は次式で示す1次系のローパスフィルタとなる.

$$K(z) = \frac{k_z}{z - 1 + k_z} \quad (4.13)$$

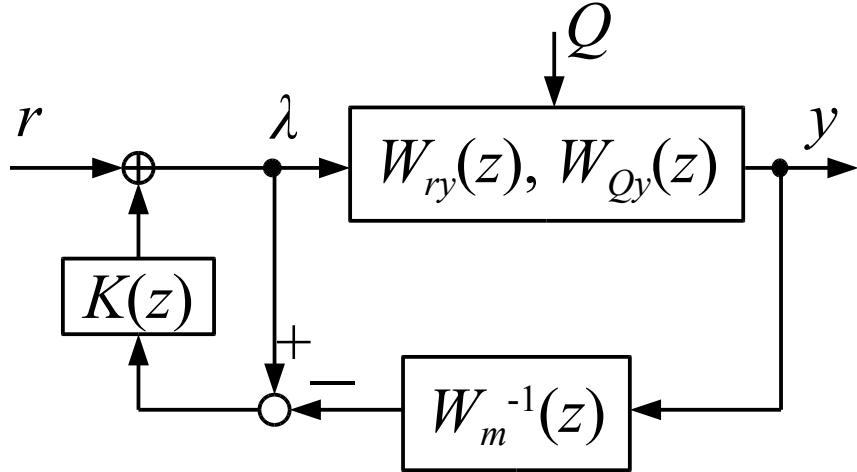


図4.9 逆システムとフィルタを結合したロバスト補償器

このシステムの  $r \rightarrow y$  及び  $Q \rightarrow y$  伝達関数は次式となる.

$$y = W_m(z) \frac{z - 1 + k_z}{z - 1 + k_z W_s(z)} W_s(z) r \quad (4.14)$$

$$y = \frac{z - 1}{z - 1 + k_z} \frac{z - 1 + k_z}{z - 1 + k_z W_s(z)} W_{Qy}(z) Q \quad (4.15)$$

ただし,

$$W_s(z) = \frac{(1 - H_2)}{(z - H_2)} \frac{(1 - H_3)}{(z - H_3)} \frac{(z - n_1)}{(1 - n_1)} \frac{(z - n_2)}{(1 - n_2)}$$

である. ここで,  $W_s(z) \approx 1$  であれば,

$$y \approx W_m(z) r \quad (4.16)$$

$$y \approx \frac{z - 1}{z - 1 + k_z} W_{Qy}(z) Q \quad (4.17)$$

となる.  $r \rightarrow y$  の特性は,  $H_1$  によって,  $Q \rightarrow y$  までの特性は  $k_z$  によってそれぞれ独立的に指定できることがわかる. つまり, 図4.9のシステムは, 1次モデルを近似的に実現する2自由度系であり,  $k_z$  を大きくするほど, パラメータ変動による特性変動, すなわち外乱に対し, 低感度になる.

図 4.9 の逆システムは、 $W_m^{-1}(z)$ の分子の次数が分母よりも高いため、このままでは実現できない。そこで、以下の等価変換を行い、実現できるシステムに置き換える。図 4.9 の  $\lambda(z)$  の差分方程式を  $z$  変換すると、次式が得られる。

$$\lambda(z) = r(z) + K(z) [\lambda(z) - W_m^{-1}(z)y(z)] \quad (4.18)$$

これを式変形すると、

$$\lambda(z) = r(z) + K(z)\lambda(z) - K(z)W_m^{-1}(z)y(z) \quad (4.19)$$

$(z)$ を略して、

$$\lambda(1-K) = r - KW_m^{-1}y \quad (4.20)$$

$K$  と  $W_m$  を展開すると、

$$\lambda\left(1 - \frac{k_z}{z-1+k_z}\right) = r - \frac{k_z}{z-1+k_z} \frac{z-H_1}{1-H_1} y \quad (4.21)$$

$$\begin{aligned} \lambda\left(\frac{z-1}{z-1+k_z}\right) &= r - \frac{k_z}{z-1+k_z} \frac{z-1+1-H_1}{1-H_1} y \\ &= r - \frac{k_z}{z-1+k_z} \frac{z-1}{1-H_1} y - \frac{k_z}{z-1+k_z} \frac{1-H_1}{1-H_1} y \\ &= r - \frac{k_z}{z-1+k_z} \frac{z-1}{1-H_1} y - \frac{k_z}{z-1+k_z} y \end{aligned} \quad (4.22)$$

左辺の  $( )$  内を右辺に移して、

$$\begin{aligned} \lambda &= r \frac{z-1+k_z}{z-1} - \frac{k_z}{1-H_1} y - \frac{k_z}{z-1} y \\ &= r + \frac{k_z}{z-1} r - \frac{k_z}{1-H_1} y - \frac{k_z}{z-1} y \end{aligned} \quad (4.23)$$

と表される。これより次式が得られる。

$$\lambda(z) = r(z) + \frac{k_z}{z-1} (r(z) - y(z)) - \frac{k_z}{1-H_1} y(z) \quad (4.24)$$

よって、図 4.9 のシステムは、図 4.10 のシステムに置き換えることができる。図 4.10 と図 4.8 のモデルマッチングシステムをまとめると、図 4.11 に示すデジタル積分形制御システムが得られる。

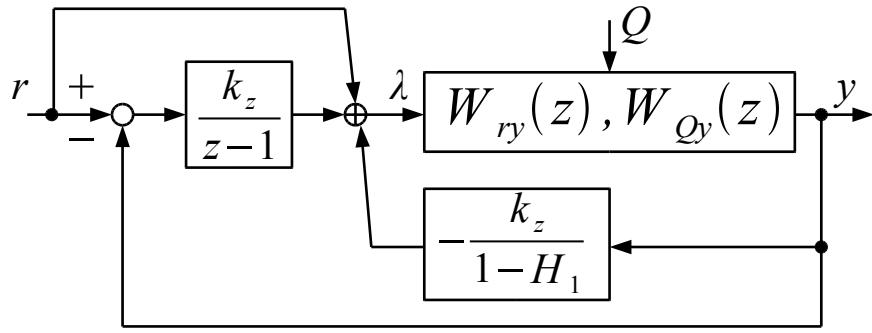


図 4.10 等価変換されたロバスト補償器

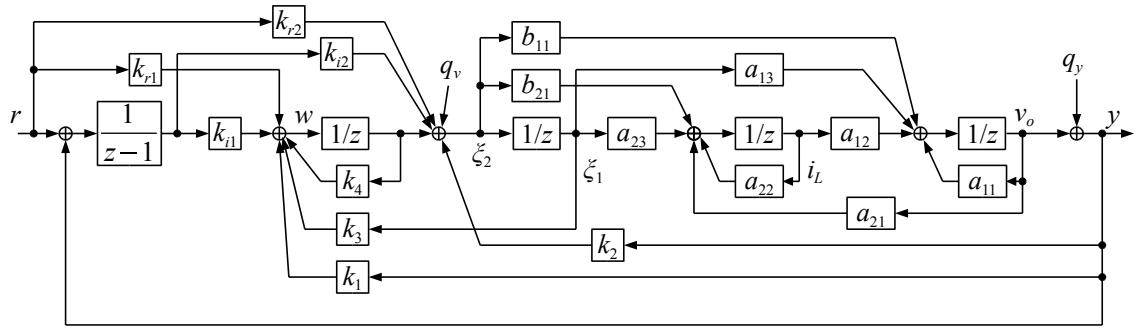


図 4.11 近似的 2 自由度デジタル積分形制御系

各パラメータは以下となる。

$$\begin{aligned}
 k_1 &= -\frac{k_z}{1-H_1} G(f\bar{f}_4 - H_4) + f\bar{f}_1 + f\bar{f}_2 f\bar{f}_4 \\
 k_2 &= f\bar{f}_2 - \frac{k_z}{1-H_1} G, \quad k_3 = f\bar{f}_3, \quad k_4 = f\bar{f}_4 \\
 k_{i1} &= k_z G(f\bar{f}_4 - H_4), \quad k_{i2} = k_z G, \quad k_{rl} = G(f\bar{f}_4 - H_4), \quad k_{r2} = G
 \end{aligned} \tag{4.25}$$

図 4.11 のシステムは、以下の制御アルゴリズムで表され、これをマイクロコントローラに実装して実行することで、システムの制御演算を行うことができる。

$$\begin{aligned}
 \xi_2 &= k_{i2} * v_i + w + k_2 * y + k_{rl} * r \\
 w &= k_{i1} * v_i + k_1 * y + k_3 * \xi_1 + k_4 * w + k_{r2} * r \\
 v_i &= r - y + v_i \\
 \xi_1 &= \xi_2
 \end{aligned}$$

また、式(4.14)において、 $W_s(z) \approx 1$  のとき、 $k_z$  が十分大きく、 $|H_1| \gg |1+k_z|$  であれば、次の近似式が成り立つ。

$$\frac{z-1+k_z}{z-1+k_z} = \frac{k_z}{z-1+k_z} \tag{4.26}$$

このとき、 $\lambda$ は、式(4.23)より、

$$\begin{aligned}\lambda &= r \frac{z-1+k_z}{z-1} - \frac{k_z}{1-H_1} y - \frac{k_z}{z-1} y \\ &= \frac{k_z}{z-1} r - \frac{k_z}{1-H_1} y - \frac{k_z}{z-1} y = \frac{k_z}{z-1} (r-y) - \frac{k_z}{1-H_1} y\end{aligned}\quad (4.27)$$

となる。したがって、 $k_z$ が十分大きく、式(4.26)が成り立つとき、図4.10における $r$ から $\lambda$ のフィードフォワードと、その等価変換である、図4.11の $k_{r1}$ ,  $k_{r2}$ のフィードフォワードは必ずしも必要ではない。

### 4.3 制御器パラメータの決定法

提案した図4.11のシステムでは、広い周波数帯域において $W_s(z) \approx 1$ とすることで、式(4.14), (4.15)を(4.16), (4.17)に近づけ、近似度を上げることが重要となる。また、外乱に対して低感度にするため、 $k_z$ を大きな値にしなければならない。しかし、 $k_z$ を大きくすると、式(4.14), (4.15)の

$$z-1+k_z W_s(z)=0 \quad (4.28)$$

の根が代表根 $H_1$ に近づき、 $W_s(z) \neq 1$ となって、近似度が悪くなる可能性がある。

式(4.28)において、 $k_z$ を0から大きくしたとき、根がどのように動くかをプロットしたものを根軌跡と呼び、以下の図4.12にその例を示す。根は $1$ ,  $-H_2$ ,  $-H_3$ より出発し、 $k_z$ が任意の値の時、 $p_{z1}, p_{z2}, p_{z3}$ の位置に移動する。 $k_z$ が十分大きな値で、そのときの $p_{z1}, p_{z2}, p_{z3}$ が

$$|p_{z1}|, |p_{z2}|, |p_{z3}| \ll |H_1| \quad (4.29)$$

となれば、その位置で近似度はよくなり、かつ外乱に対して低感度にことができる。

そこで、制御器パラメータの決定手順を以下のように定める。

- (1) 公称モデルとなる動作点をひとつ決定する。
- (2) 代表根 $H_1$ を、任意の立ち上がり時間となるように定める。
- (3)  $H_4$ が代表根に影響を与えないよう、 $|H_4| \approx 0.5|H_1|$ と指定する。
- (4)  $k_z$ を $k_z \approx 0.5$ と指定する。
- (5)  $|H_3| \ll |H_1|$ ,  $|H_2| \ll |H_1|$ を満たすように、 $H_2$ ,  $H_3$ を指定する。
- (6) シミュレーションによって、全ての動作点で仕様を満たすことを確認する。
- (7) 仕様を満たさない場合、以下の優先順位でパラメータを調整する。
  - i. 根軌跡を見て、 $|p_{z1}|, |p_{z2}|, |p_{z3}| \ll |H_1|$ を満たすよう $H_2, H_3$ を変更する。
  - ii.  $k_z$ を変更する(範囲は $0 < k_z < 1$ )。
  - iii.  $H_1, H_4$ もしくは公称モデルを変更して、iとiiを繰り返す。

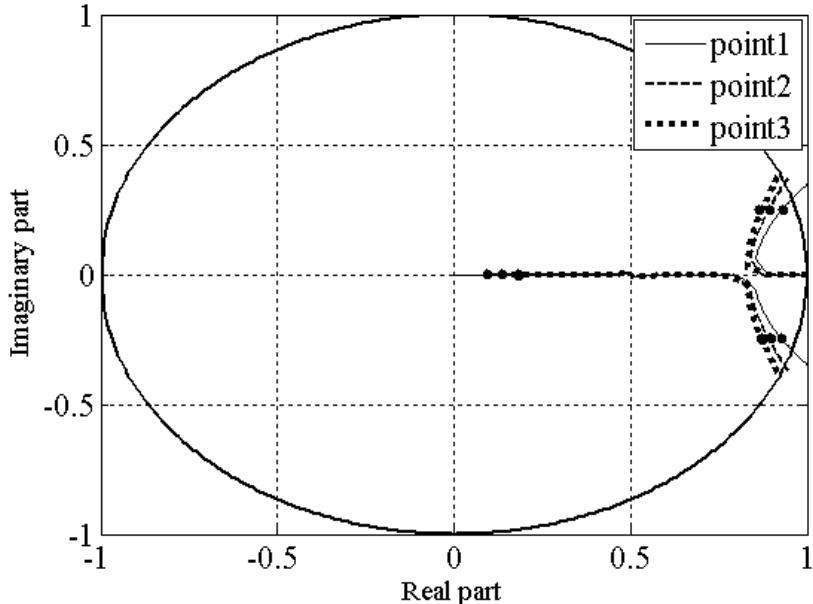


図 4.12 根軌跡

#### 4.4 制御器の設計

実際の設計した値をここでは述べる。以降の数値計算とシミュレーションには、科学計算ソフトウェアの MATLAB を用いている。また、MATLAB のプログラムは付録 A に付す。

今回、公称モデルは、出力電圧の時定数が一番長く、入出力間の定常ゲインが最大である、動作点①を公称モデルとして用いる。

まず、サンプリング周波数 100[kHz]、遅れ時間  $0.99T_s$  の時の、制御対象となる昇圧コンバータの離散伝達関数は、以下の式となる。

$$G_{wv_o} = -0.000027416 \frac{(z - 2.5985)(z + 52.469)}{z^2(z - 0.99998)(z - 0.91395)} \quad (4.29)$$

ここで、 $H_1$  は、出力電圧のステップ応答をおよそ 100[ms] とするため、 $H_1=0.9999$  とする。また  $H_4=0.3$ 、 $k_z=0.5$  と設定し、 $|H_3| \ll |H_1|$ 、 $|H_2| \ll |H_1|$  となるよう、 $H_2, H_3=0.1 \pm 0.1i$  と指定した。このとき、得られた制御器では、オーバーシュートが大きく、仕様を満たさなかった。図 4.12 の根軌跡を見ると、根の移動距離を大きくし  $k_z$  を大きくするためには、 $H_2, H_3$  の実部が大きく、かつ虚部は小さく、安定性のために右半平面に置く必要があることがわかった。そのため、 $H_2, H_3$  を設定し直し、すべての動作点で安定で、i の条件を満たすようパラメータを微調整した結果、

$$H_1 = 0.99973, \quad H_2, H_3 = 0.47 \pm 0.01i, \quad H_4 = 0.1, \quad k_z = 0.3 \quad (4.30)$$

と指定した。このとき、式(4.25)より、制御器パラメータは以下のように得られる。

$$\begin{aligned} k_1 &= 86.57063, \quad k_2 = -117.2187, \quad k_3 = -0.3079751 \\ k_4 &= 0.07642718, \quad k_i = 0.0002289656, \quad k_{iz} = 0.009713121 \end{aligned} \quad (4.31)$$

ここで、式(4.26)で得られる、移動後の根は、代表根  $H_1$  よりも小さい。このため、式(4.27)の近似により、 $k_{r1}$ ,  $k_{r2}$  のフィードフォワードゲインは 0 に設定した。

## 4.5 シミュレーションとその結果

図 4.13 に示す SIMULINK モデルを用いて、前項で得られた式(4.31)の制御器パラメータを用いた時の、各動作点における、ステップ応答、 $r \rightarrow y$  間ボード線図、開ループボード線図、 $q_y \rightarrow y$ ,  $q_v \rightarrow y$  間ボード線図、それぞれのシミュレーションを行った。ステップ応答結果を図 4.14、ボード線図を図 4.15～18 に示す。いずれの動作点に置いても、出力電圧のステップ応答波形とステップ応答時間はほぼ同じであり、またオーバーシュートはほぼ無く、仕様を満たすことがわかる。

また、非線形な昇圧コンバータの負荷急変応答(動作点変化)をシミュレーションするため、SimPowerSystems ツールボックスを用いて、図 4.19 のモデルを作成し負荷急変応答シミュレーションを行った。この結果を図 4.20 に示す。負荷急変時の出力電圧変動幅は 5[V]程度に抑えられており、出力電圧に対し 1.1[%]の変動であるため、これも仕様を満たす。

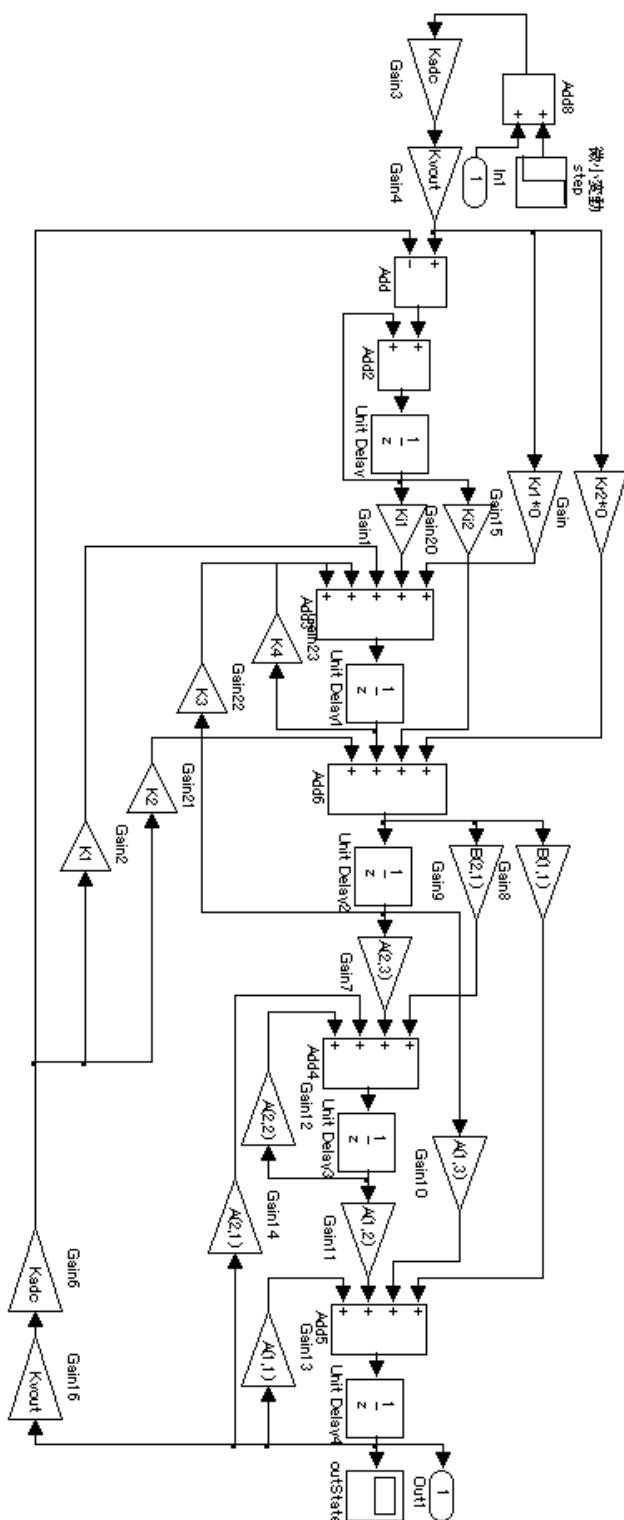


図 4.13 近似的 2 自由度ディジタル積分形制御系の MATLAB/SIMULINK モデル

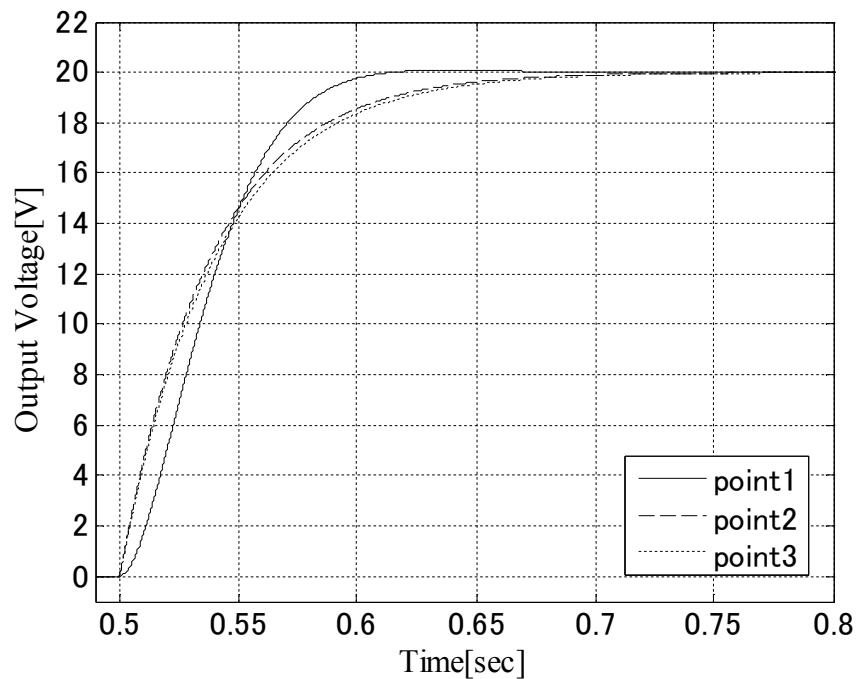


図 4.14 近似的 2 自由度ディジタル積分形制御系の Step 応答シミュレーション

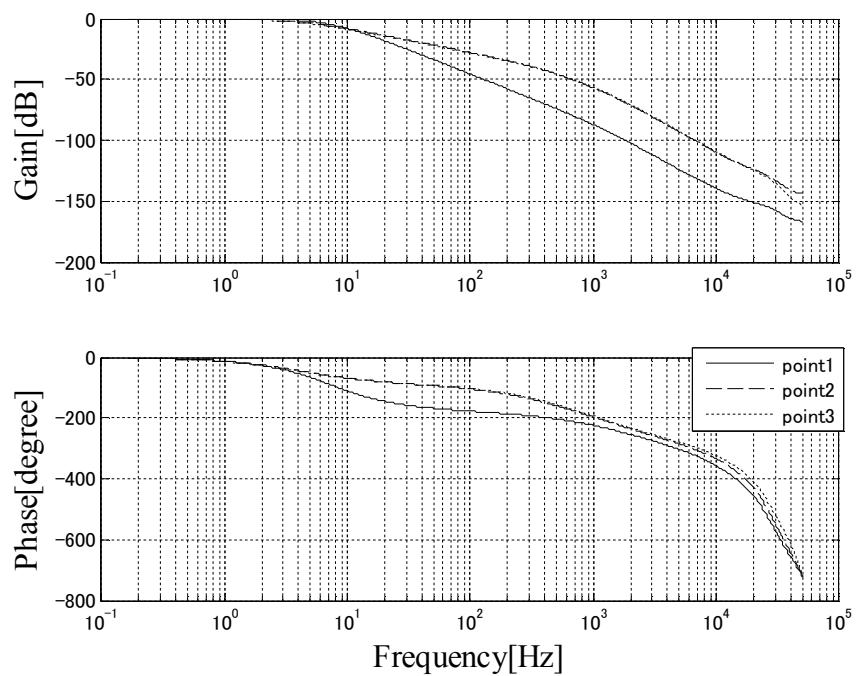


図 4.15 図 4.13 の制御器の  $r \rightarrow y$  間ボード線図

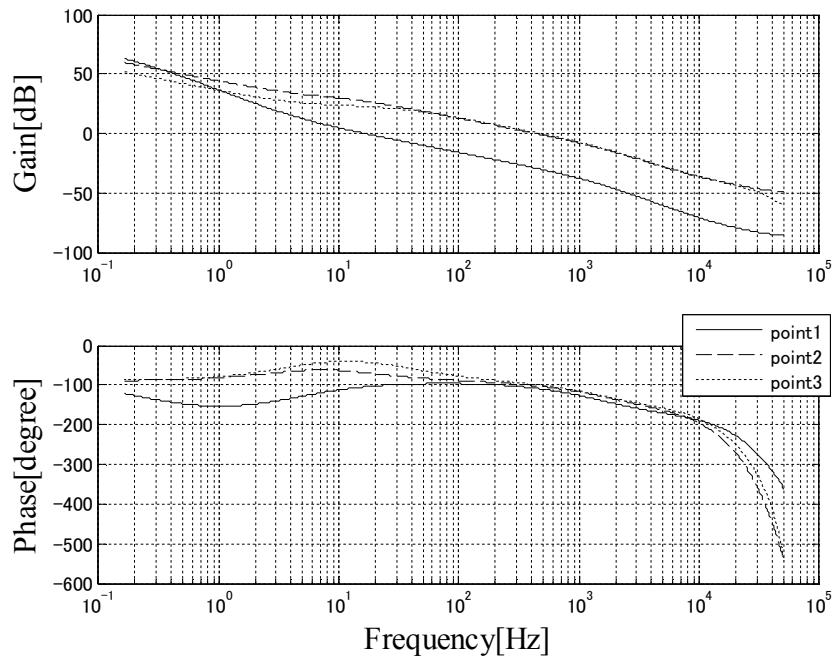


図 4.16 図 4.13 の制御器の開ループボード線図

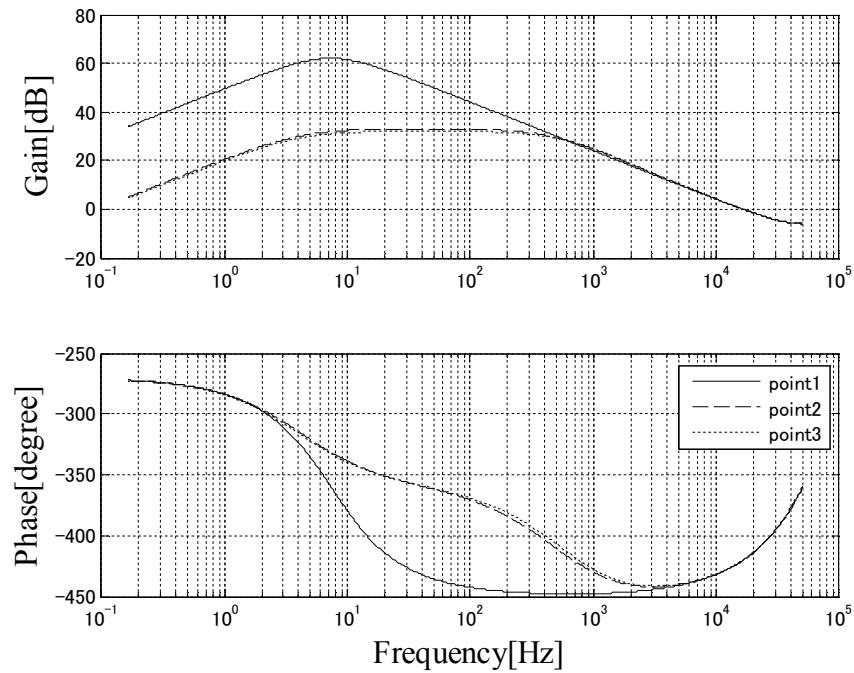


図 4.17 図 4.13 の制御器の  $q_y \rightarrow y$  間ボード線図

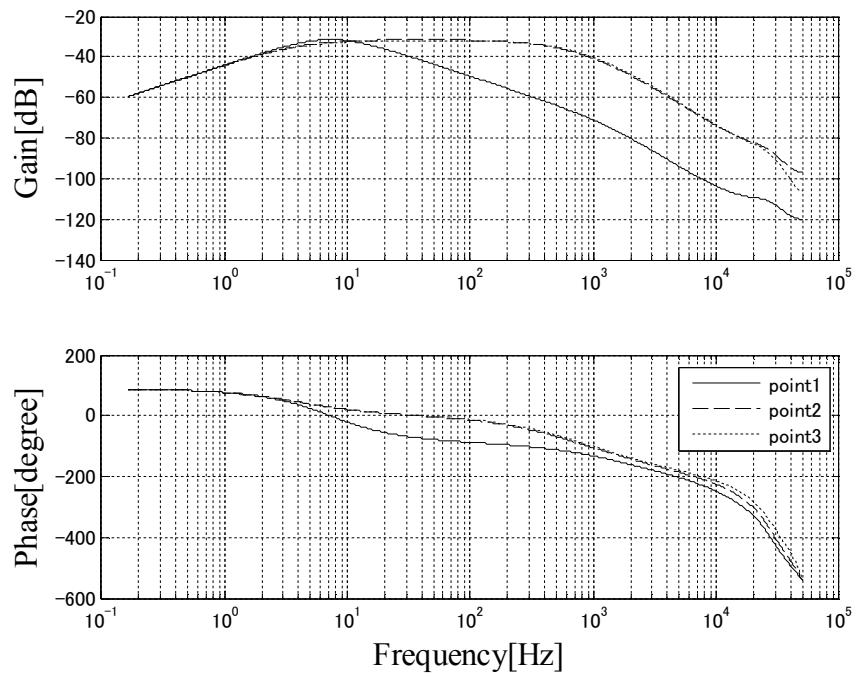


図 4.18 図 4.13 の制御器の  $q_v \rightarrow y$  間ボード線図

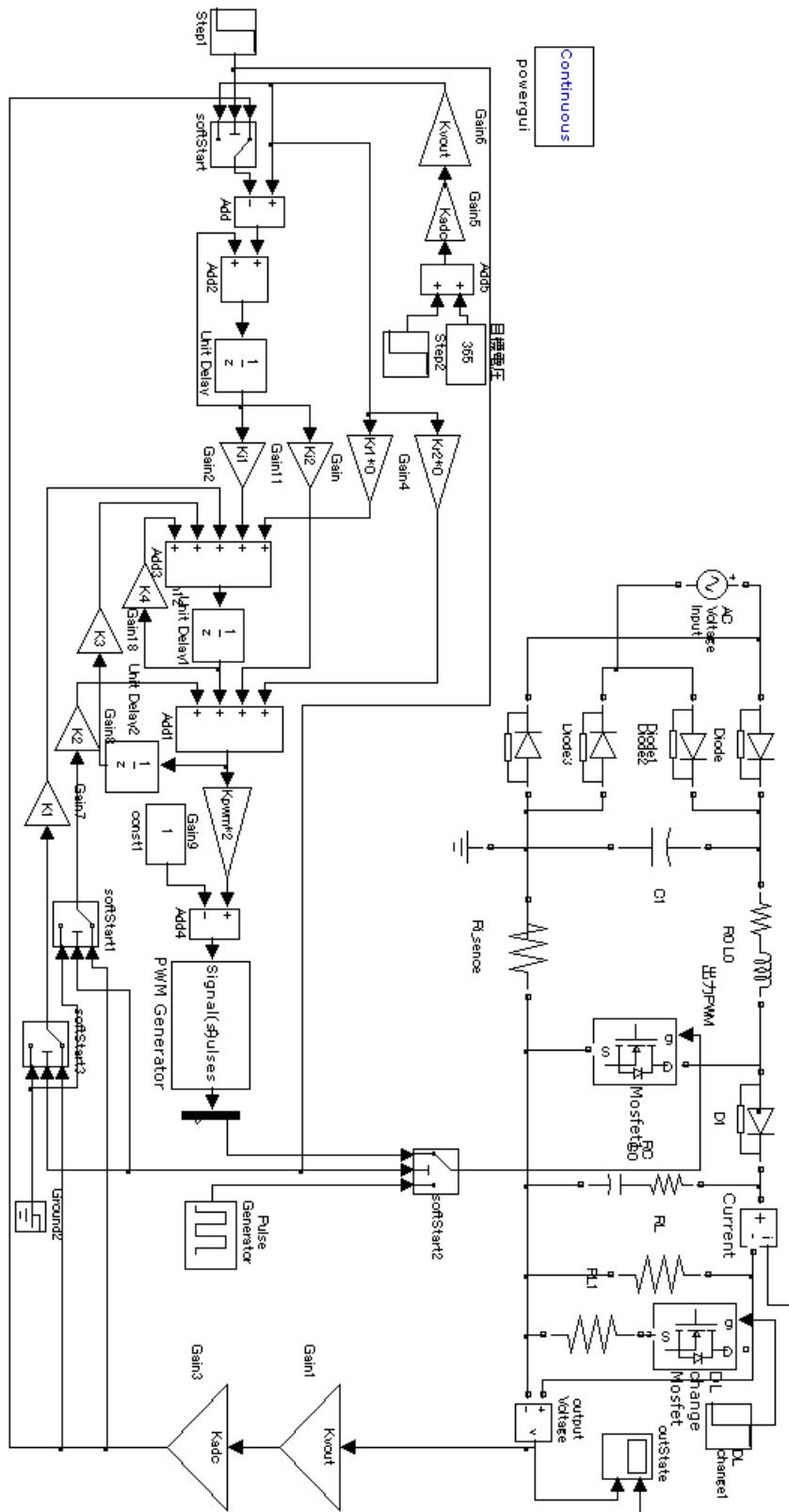


図 4.19 SimPowerSystems を用いた昇圧コンバータの負荷急変シミュレーションモデル

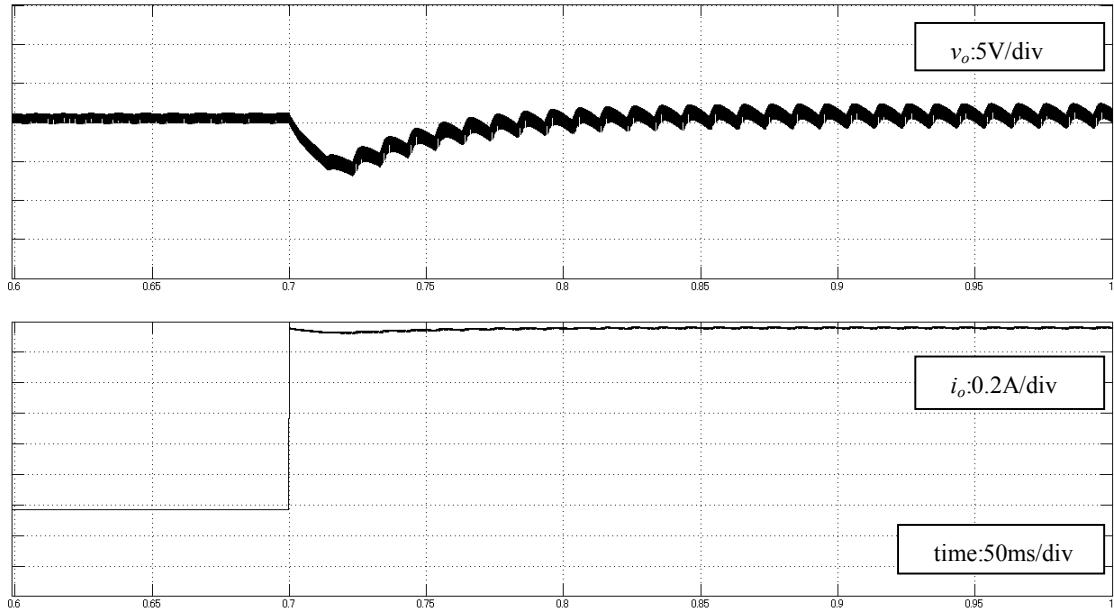


図 4.20 SimPowerSystems を用いた昇圧コンバータの負荷急変シミュレーション結果

## 4.6 実験結果

式(4.31)の制御器パラメータを、前述の制御アルゴリズムを用いマイクロコントローラに実装して実行し、実機実験を行った。図 4.21 に各動作点におけるステップ応答結果を示す。シミュレーションどおり、それぞれ同様の立ち上がり波形で、約 80[ms]で立ち上がっており、動作点の変化に対しロバストな結果が得られた。

また、付録 E に示す、負荷急変装置を用いて、負荷急変時の出力電圧変動を測定した際の応答結果を、図 4.22 に示す。これもシミュレーションと同様、出力電圧の変動幅が 5[V]程度となり、十分ロバストな結果が得られ、すべての仕様を満たすことが確認された。

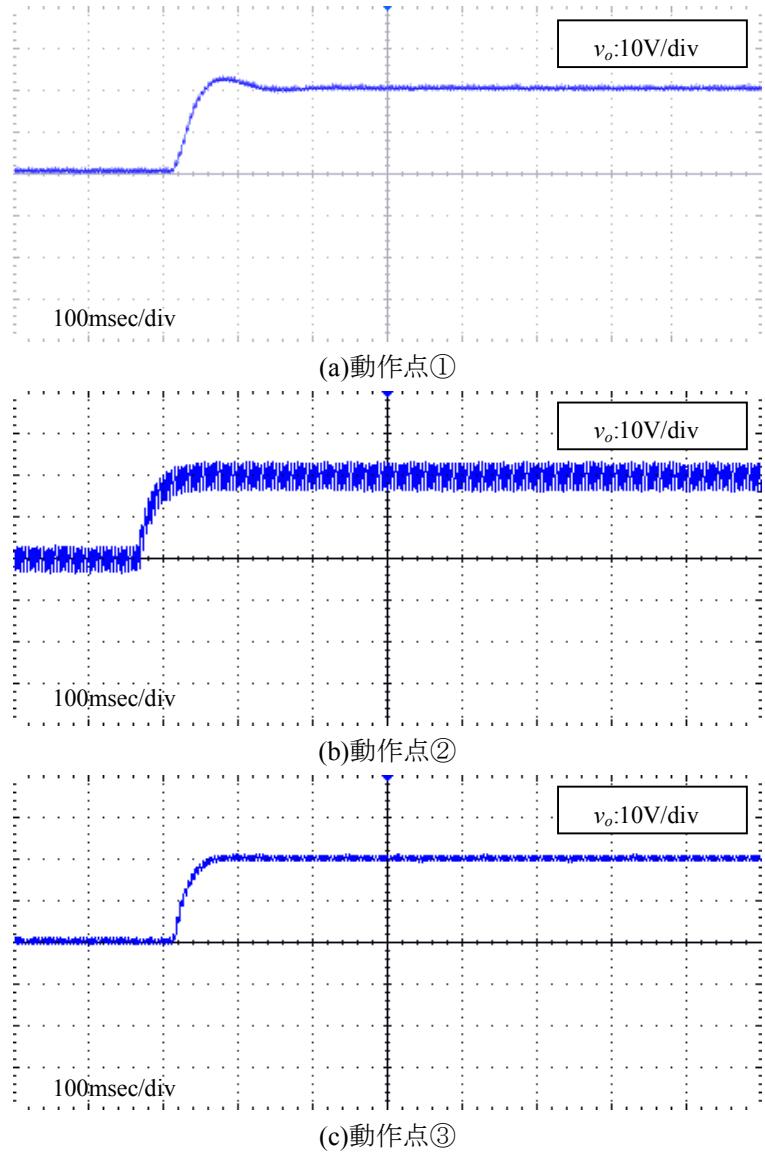


図 4.21 近似的 2 自由度ディジタル積分形制御系の各動作点のステップ応答結果

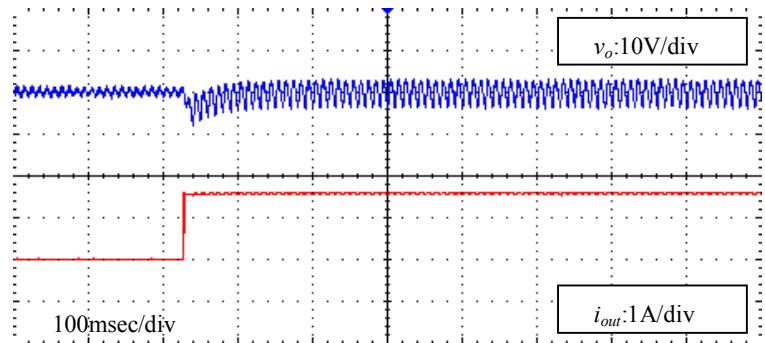


図 4.22 近似的 2 自由度ディジタル積分形制御系の負荷急変応答結果

ここで、従来手法との比較として、図 4.23 に示す、PI 制御器を用いた場合の、実機実験を行った。パラメータは、すべての動作点のステップ応答でオーバーシュートが小さく、かつ高速に応答する限界をトライアンドエラーで  $K_p=1$ ,  $K_i=0.0003$  に決定した。この PI 制御器のステップ応答の結果を図 4.24 に、負荷急変応答結果を図 4.25 にそれぞれ示す。

PI 制御器を用いた場合、動作点の違いにより、ステップ応答の立ち上がり時間が違ってしまう。また負荷変動時の出力電圧変動幅は約 20[V]と、大きな変動を生じてしまう。負荷急変時の出力電圧変動幅を、パラメータの変更により 15[V]程度まで小さくすることはできるが、その場合、ステップ応答で大きなオーバーシュートを生じ、安定して立ち上がらない場合があるなど、仕様を満たすことが困難である。

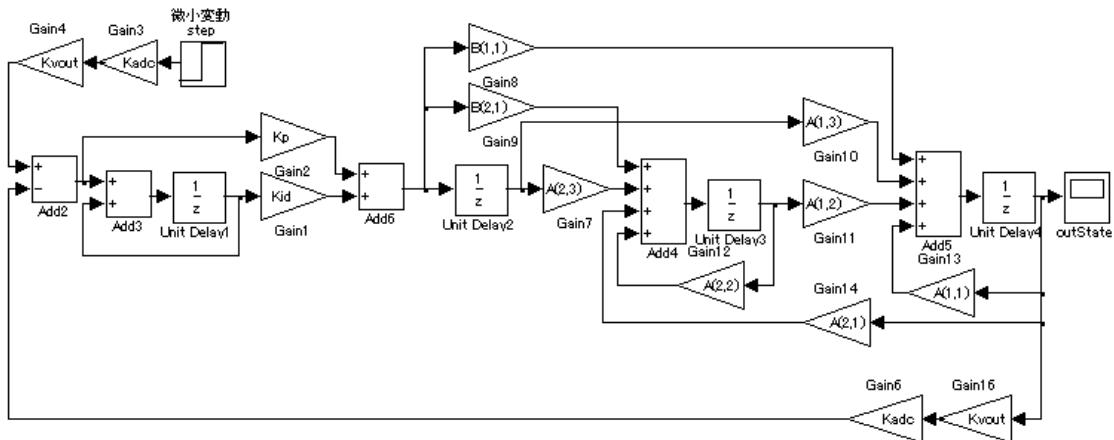


図 4.23 PI 制御器の MATLAB/SIMULINK シミュレーションモデル

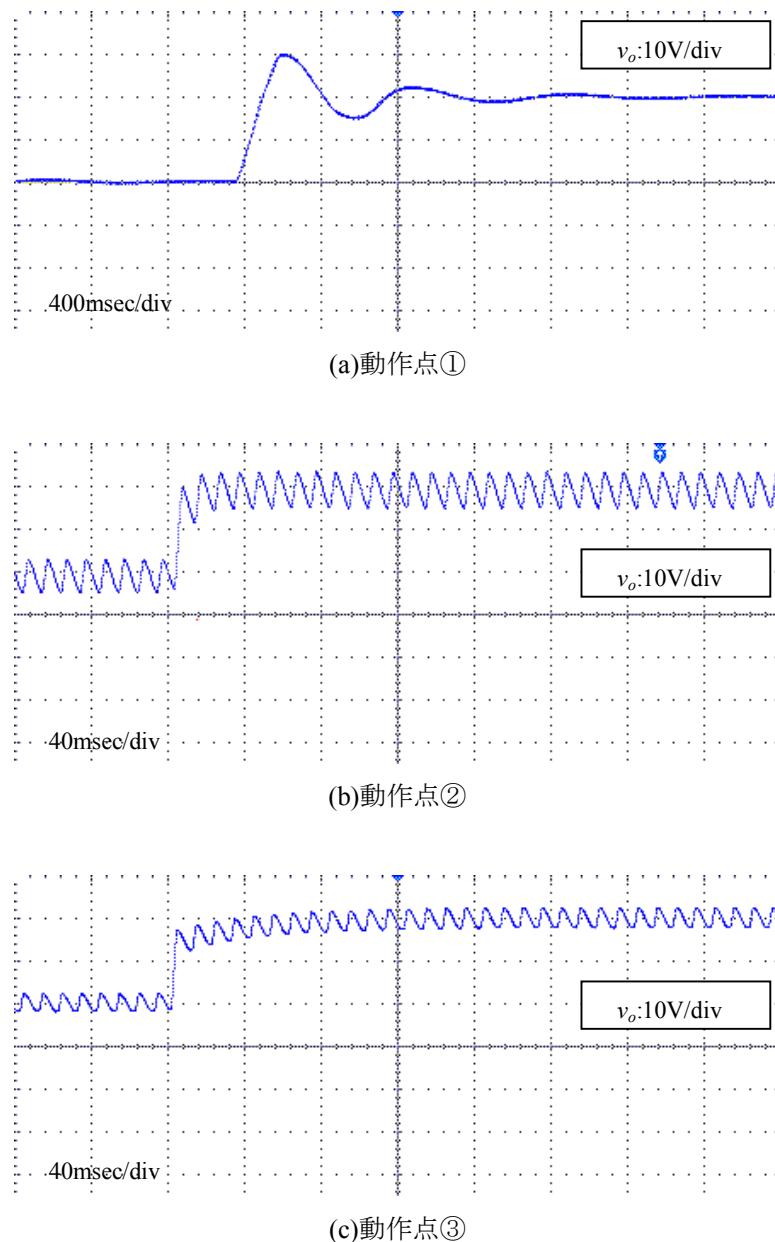


図 4.24 PI 制御器の各動作点のステップ応答結果

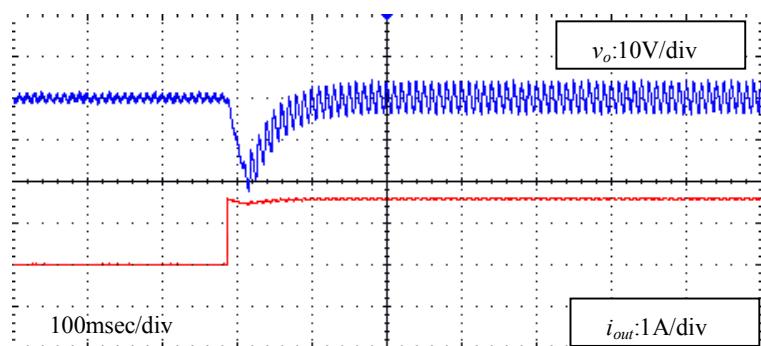


図 4.25 PI 制御器の負荷急変応答結果

# 第5章 力率改善を行う近似的 2 自由度制御系の設計法

## 5.1 力率改善手法の概要と目標

一般的な、力率改善制御器の構成を図 5.1 に示す。力率改善制御器の構成手順としては、まず、はじめに、デューティ比  $\mu$  を入力、インダクタ電流  $i_L$  を出力とする系を制御対象として、電流制御器を設計する。これで、電流は  $r_i$  に追従するようになる。ここで  $r_i$  に  $v_{ac}$  を乗算して、 $i_L$  を、整流された入力電圧に追従させるようとする。こうすると、入力電流  $i_{ac}$  は入力電圧  $v_i$  に追従する事になり、力率が改善され、高調波特性が改善される。またここで、 $v_{ac}$  の振幅が変動した場合に、出力電圧とインダクタ電流にその影響が出ることを抑制する補償として、 $v_{ac}$  の平均値  $v_{avg} = \frac{2}{\pi} V_i$  の逆数の二乗  $\frac{1}{v_{avg}^2}$  を計算し、これも乗算する。

さらに、乗算器入力  $u_v$  を入力、出力電圧  $v_o$  を出力とする系を新たな制御対象として、電圧制御器を設計する。こうすると、入力電流は入力電圧に追従したまま、出力電圧を一定に保つようデューティ比の平均値を操作するような力率改善制御器が得られる。

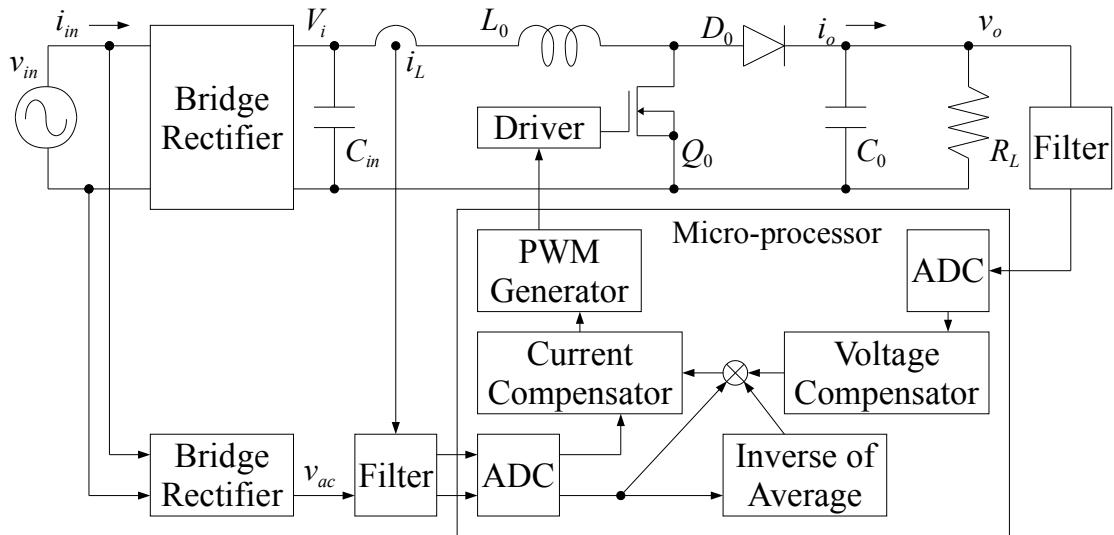


図 5.1 一般的な力率改善制御器の構成

以上のように、力率改善制御器は、出力電圧とインダクタ電流の両方を制御する必要があるため、2重のループを持つ。この力率改善制御に、近似的 2 自由度制御を適用する場合

でも、一般的な制御器同様、電流制御器を設計した後、乗算器を追加し、電圧制御器を構成する。

単純に近似的 2 自由度制御器で 2 重ループを構成すると、図 5.2 に示すように、外側の電圧制御ループが複雑になり、設計が難しく、マイクロコントローラにおける制御演算に多くの時間がかかるてしまう。そこで、本章では、内側の電流制御器を、パラメータの設定条件によって 1 次モデルに近似し、電圧制御器の設計を簡易にした、新しい近似的 2 自由度力率改善制御器の設計手法を提案する。

力率改善制御では、4 章で述べた、パラメータ変化等による特性変動等は、電流制御器の操作量・制御量に入る外乱  $Q_y = [q_v, q_{yi}]$  と、電圧制御器の操作量・制御量に入る外乱  $Q_v = [q_v, q_{vv}]$  を加えることと等価であると考えられる。そのため、これらの外乱から、出力までの伝達関数をなるべく小さくなるよう、提案手法を用いて、合理的にパラメータを設定することで、多動作点において、同一の目標値応答、高力率と高調波電流の低減を実現し、かつ動作点の変動に対してロバストな制御器を得ることが目標である。

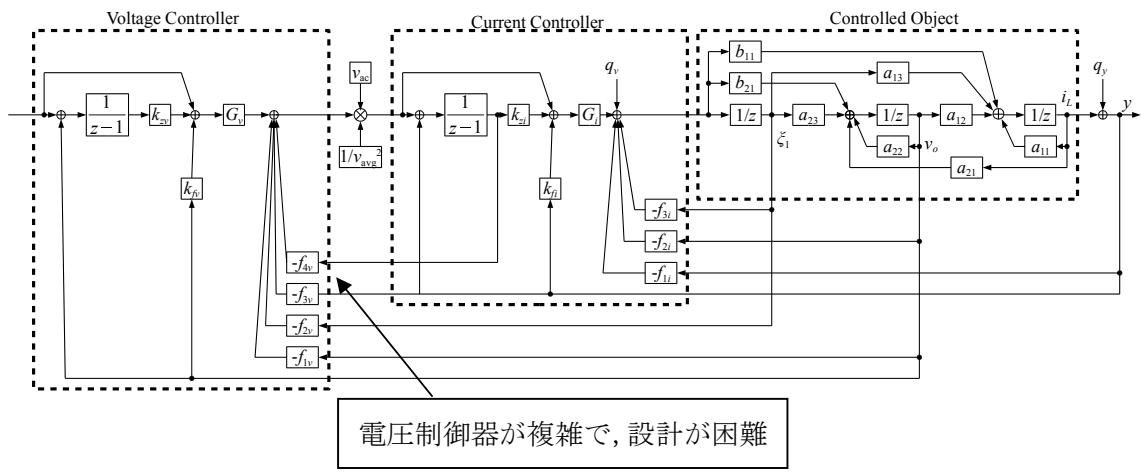


図 5.2 近似的 2 自由度制御を力率改善制御に適用する際の問題点

## 5.2 PFC 昇圧型コンバータ

### 5.2.1 離散系制御対象

昇圧型コンバータの、インダクタ電流と出力電圧両方を制御することを考え、5.1 節で述べた手順で、近似的 2 自由度 PFC 制御器を設計する。このとき、まず、インダクタ電流制御器を設計し、その後出力電圧制御器を設計するため、第 1 状態変数に  $i_L$  を、第 2 状態変数に  $v_o$  をとり、出力  $y$  として双方を得るものとすると、制御対象の動作点近傍の状態方程式は、以下の式となる。

$$\begin{aligned}\dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t)\end{aligned}\tag{5.1}$$

ただし,

$$A_c = \begin{bmatrix} -\frac{R_0}{L_0} & -\frac{1-\mu_s}{L_0} \\ \frac{1-\mu_s}{C_0} & -\frac{1}{R_L C_0} \end{bmatrix}, \quad B_c = \begin{bmatrix} \frac{V_s}{L_0} \\ -\frac{I_s}{C_0} \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad x(t) = \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix}$$

$$u(t) = \mu(t), \quad y(t) = \begin{bmatrix} y_i(t) \\ y_v(t) \end{bmatrix} = \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix}$$

である。この式を遅れ時間  $L_d$  を考慮して離散化すると、式(2.22)から、以下となる。

$$\begin{aligned} x_{dt}(k+1) &= A_{dt}x_{dt}(k) + B_{dt}v(k) \\ y(k) &= C_{dt}x_{dt}(k) \end{aligned} \quad (5.2)$$

ただし,

$$A_{dt} = \begin{bmatrix} e^{A_c T_s} & e^{A_c(T_s-L_d)} \int_0^{L_d} e^{A_c \tau} B_c d\tau \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{dt} = \begin{bmatrix} \int_0^{T_s-L_d} e^{A_c \tau} B_c d\tau \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ 1 \end{bmatrix}$$

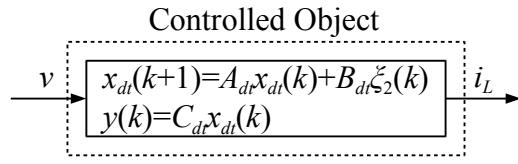
$$C_{dt} = [C_c \quad 0], \quad x_{dt}(k) = \begin{bmatrix} i_L(k) \\ v_o(k) \\ \xi_1(k) \end{bmatrix}, \quad v(k) = \xi_2(k) = \xi_1(k+1)$$

である。この時の制御対象は、図 5.3 のようになる。5 章では、これを制御対象とし、力率改善制御器を構成していく。

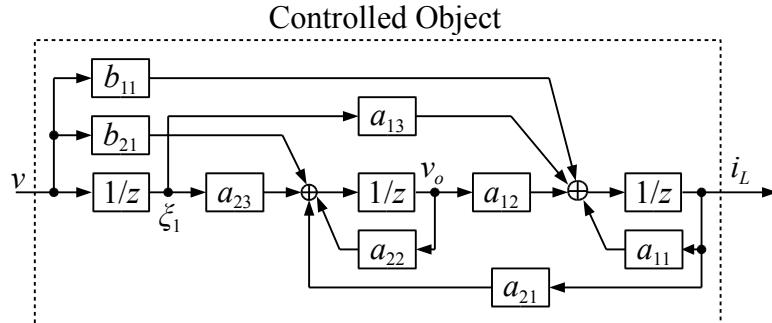
### 5.2.2 PFC 昇圧型コンバータの仕様と動作点の決定

PFC 昇圧型コンバータにおいても、多動作点に対して、同じ立ち上がり応答を保ち、かつオーバーシュートを生じない目標値ステップ応答特性や、動作点の変動に対する応答を見るための、負荷急変応答特性が重要である。また、常に高い力率を保つために、それぞれの動作点における、力率特性と高調波電流特性が重要となってくる。よって、以下の仕様(1)～(6)を満たすデジタル制御 PFC 昇圧型 DC-DC コンバータを設計・製作する。

- (1)入力電圧は 100[VAC]、出力電圧は 240[VDC]から 385[VDC]まで変化する
- (2)ステップ応答は負荷抵抗が 300[Ω]から 5[kΩ]の範囲で変わらないものとする
- (3)ステップ応答におけるオーバーシュートはすべての負荷において 10[%]以内とする。
- (4)負荷が 300[Ω]から 5[kΩ](またはその反対)に急変した時の出力電圧変動は 5[%]以下とする。
- (5)最大負荷時の力率は 0.99 以上とする。
- (6)入力電流の高調波電流は国際規格である IEC/EN61000-3-2 を満たす。



(a)数式による表現



(b)ブロック線図表現

図 5.3 入力無駄時間  $L_d$ を持つ制御対象

これらの仕様を満たすように制御器を設計する際、評価すべき動作点を、負荷抵抗が最小と最大の点、出力電圧が最小と最大の点から、以下のように 3 つ定める。

- ①出力電圧 385[VDC]、負荷抵抗 5[kΩ]
- ②出力電圧 385[VDC]、負荷抵抗 300[Ω]
- ③出力電圧 240[VDC]、負荷抵抗 300[Ω]

これら 3 つのすべての動作点で、同じステップ応答で安定な制御器を設計することが出来れば、仕様で定められるすべての動作範囲で、安定であり、また外乱  $Q_i, Q_v$  から  $y$  までの伝達特性を低感度にすることが出来れば、動作点間の変動の影響を抑制できる。以下では、これらの 3 つの動作点での特性を考慮し、外乱の影響を抑制する簡単な PFC 昇圧型コンバータの制御器の設計法を提案する。

## 5.3 PFC 昇圧型コンバータの近似的 2 自由度制御器の設計

### 5.3.1 電流制御器の設計

状態フィードバック則を用いて、制御対象の応答特性を任意のものに変更し、モデルマッチングシステムを構築する。式(5.2)の制御対象に、状態フィードバック  $v(k) = -Fx_{dt}(k) + Gr(k)$  を適用すると、モデルマッチングシステムは図 5.4 のようになる。この時の、入力  $r_i'$  から出力  $i_L$  までの応答特性は以下の式になる。

$$W_{ryi}(z) = \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-H_3)}{(z-H_3)} \frac{(z-n_{1i})}{(1-n_{1i})} \frac{(z-n_{2i})}{(1-n_{2i})} \quad (5.3)$$

ここで、定常状態において、入出力間の直流ゲインを 1 とするためのゲイン  $G$  は以下の式で導出される。

$$G = \frac{1}{C_{dt}} \frac{(1-H_1)(1-H_2)(1-H_3)}{I - A_{dt}} \quad (5.4)$$

式(5.3)において、昇圧コンバータのインダクタ電流を第一状態変数にとった場合、零点  $n_{2i}$  は  $|n_{2i}| \gg 1$  となり、無視しても構わないが、零点  $n_{1i}$  は、時定数  $\tau = (R_L C_0)$  が小さい場合、単位円内の中心近くに(例・ $\tau = 2.82e-8$  の時  $n_{1i} = 0.177$ ,  $\tau = 9.4e-6$  の時  $n_{1i} = 0.3683$ ,  $\tau = 9.42e-5$  の時  $n_{1i} = 0.80729$ ), 時定数  $\tau$  が大きい場合、単位円付近に(例・ $\tau = 0.2745$  の時  $n_{1i} = 0.999927$ ), それぞれ存在し、特に後者の場合、オーバーシュートを大きく生じ、伝達特性への影響が生じる。

モデルマッチングシステムの極  $H_1, H_2, H_3$  を決定する際には、 $n_{1i}$  はこれを除去しないと安定した特性が得られないため、 $H_2 \approx n_{1i}$  とする。その他の極は、 $H_1$  を代表根つまり、 $|H_1| \gg |H_3|$  となるように設定する。このとき、入力  $r$  から出力  $y$  までの伝達関数は、以下の 1 次モデルに近似できる。

$$W_{ry}(z) = \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-H_3)}{(z-H_3)} \frac{(z-n_{1i})}{(1-n_{1i})} \frac{(z-n_{2i})}{(1-n_{2i})} \approx \frac{(1-H_1)}{(z-H_1)} = W_m \quad (5.5)$$

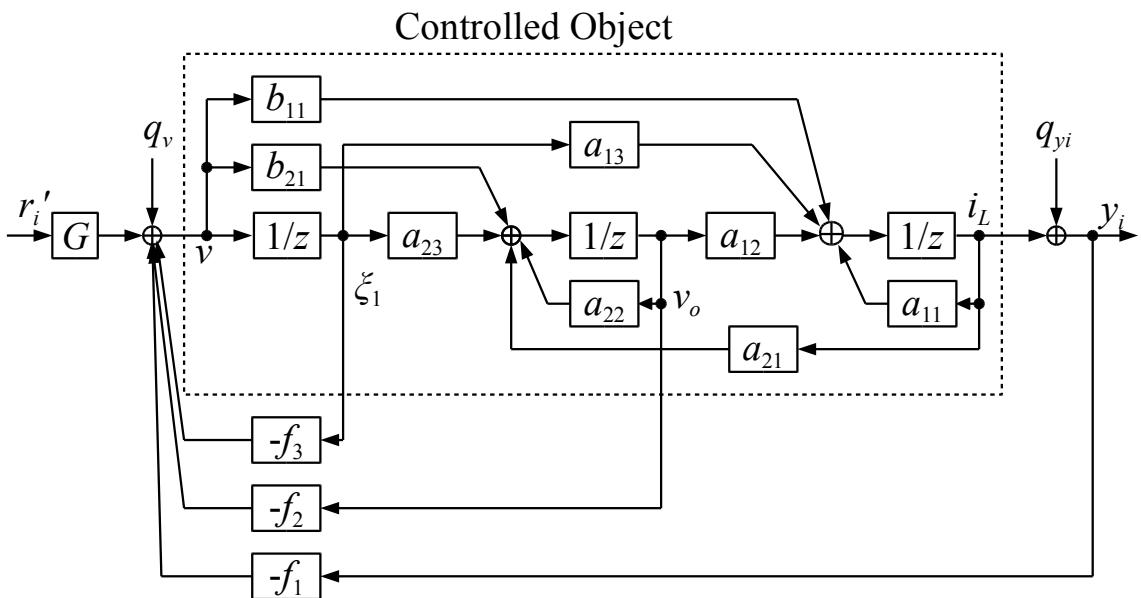


図 5.4 モデルマッチングシステム

続いて、このモデルマッチングシステムに、逆システムとフィルタを結合した、ロバスト補償器を接続する。ロバスト補償器は、4章で用いた図4.9と同じ構成の図5.5のシステムである。

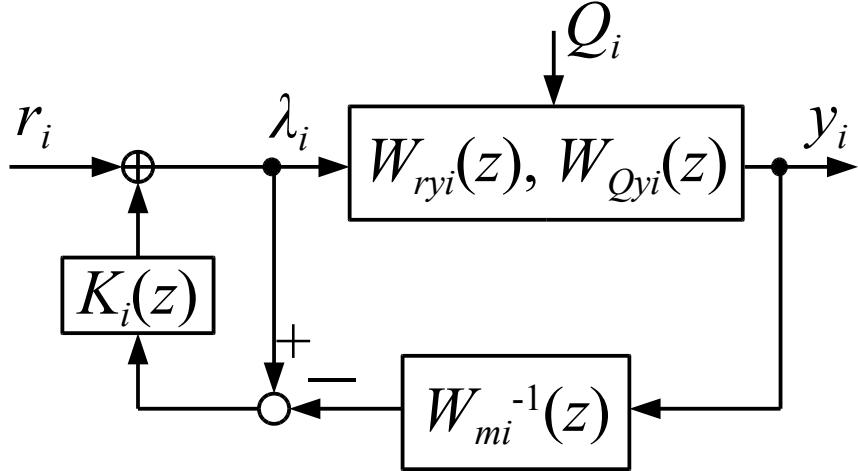


図5.5 逆システムとフィルタを結合したロバスト補償器

ここで、フィルタ  $K_i(z)$  は、

$$K_i(z) = \frac{k_{zi}}{z - 1 + k_{zi}} \quad (5.6)$$

である。

このシステムにおいて、 $r_i \rightarrow y_i$  及び  $Q_i \rightarrow y_i$  の伝達関数は以下のようになる

$$y_i = W_{mi}(z) \frac{z - 1 + k_{zi}}{z - 1 + k_{zi}W_{si}(z)} W_{si}(z)r_i \quad (5.7)$$

$$y_i = \frac{z - 1}{z - 1 + k_{zi}} \frac{z - 1 + k_{zi}}{z - 1 + k_{zi}W_{si}(z)} W_{Qyi}(z)Q_i \quad (5.8)$$

ただし、

$$W_{Qyi}(z) = [W_{qyyi}(z) \quad W_{qyyi}(z)] , \quad W_{si}(z) = \frac{(1 - H_3)(z - n_{2i})}{(z - H_3)(1 - n_{2i})}$$

である。ここで、 $W_{si}(z) \approx 1$  であれば、

$$y_i = W_{mi}(z)r_i \quad (5.9)$$

$$y_i = \frac{z - 1}{z - 1 + k_{zi}} W_{Qyi}(z)Q_i \quad (5.10)$$

となり、 $r_i \rightarrow y_i$  特性は  $H_1$  のみで、 $Q_i \rightarrow y_i$  特性は  $k_{zi}$  のみで、それぞれ独立的に指定できる、近似的に2自由度系を構成していることがわかる。4.2節で述べた等価変換手法により、このロバスト補償器は以下のモデルに変換される。

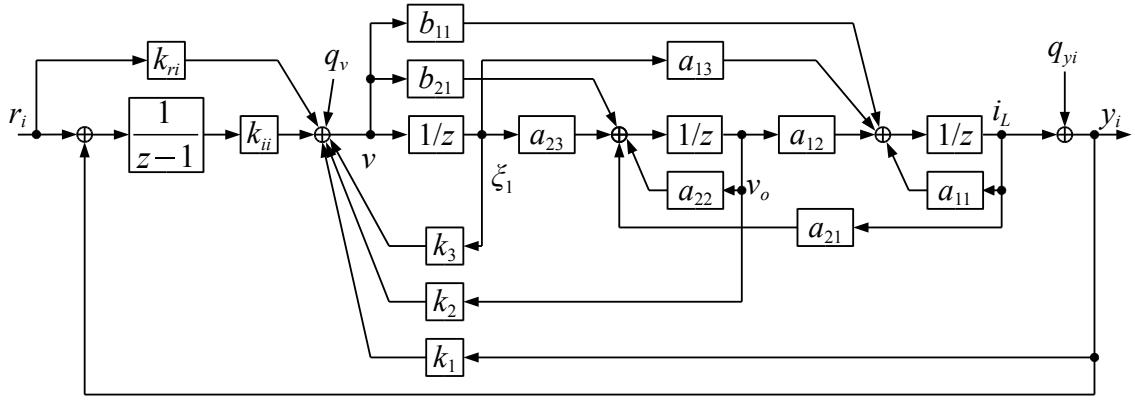


図 5.6 近似的 2自由度ディジタル積分形インダクタ電流制御器

ここで、各パラメータは以下である。

$$k_1 = -f_1 - \frac{Gk_{zi}}{1-H_1}, \quad k_2 = -f_2, \quad k_3 = -f_3 \quad (5.11)$$

$$k_{ii} = Gk_{zi}, \quad k_{ri} = G$$

図 5.6において、入力 \$r\$ から出力 \$y\$ までの伝達関数は、ロバスト補償器を接続したことにより追加された極 \$1\$ と \$H\_3\$ が、\$k\_{zi}\$ の大きさによって、別の極 \$p\_{1i}, p\_{2i}\$ (特性方程式 \$z-1+k\_{zi}W\_{si}(z)=0\$ の解) へ移動し、以下となる。

$$y_i = \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-p_{1i})}{(z-p_{1i})} \frac{(1-p_{2i})}{(z-p_{2i})} \frac{(z-n_{1i})}{(1-n_{1i})} \frac{(z-n_{2i})}{(1-n_{2i})} r_i \quad (5.12)$$

この時、\$|H\_1| \gg |p\_{1i}|, |p\_{2i}|\$ であるように \$k\_{zi}\$ と \$H\_3\$ を決めておくと、式(5.12)は、以下の 1 次式となる。

$$y_i \approx \frac{(1-H_1)}{(z-H_1)} r_i \quad (5.13)$$

### 5.3.2 電流制御器の近似と乗算器の接続

図 5.1において、インダクタ電流の平均値は、入力電流 \$i\_{ac}\$ を全波整流したものになる。そのため、インダクタ電流を、整流された入力電圧 \$v\_{ac}\$ に追従させれば、入力電流 \$i\_m\$ は、入力電圧 \$v\_{in}\$ に追従することになり、力率が改善され、高調波電流の除去が期待される。そのため、電流目標値 \$r\_i\$ に \$v\_{ac}\$ を乗算して、\$i\_L\$ を \$v\_{ac}\$ に追従させるようとする。また、そのままでには、電流目標値 \$r\_i\$ が大きくなりすぎてしまうため、\$v\_{ac}\$ の平均値 \$v\_{avg}\$ の逆数の二乗も同時に乗算器に入力する。

乗算器を接続したモデルを図 5.7 に示す。この図において、\$u\_v\$ を入力、\$v\_o\$ を出力とおいた、新たな制御対象を考え、それを用いて電圧制御器を設計する。新たな制御対象は、昇圧コンバータの離散化モデル 3 次と、制御器 1 次の計 4 次のシステムであるが、これに対し、

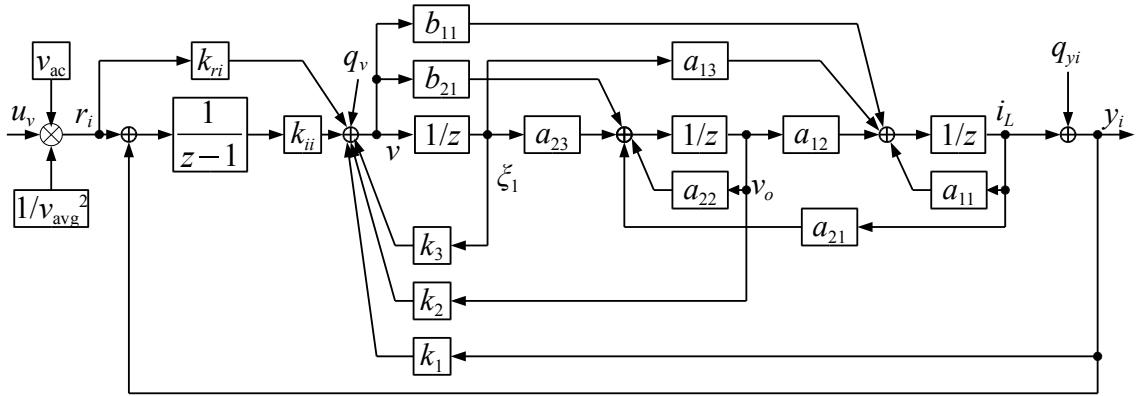


図 5.7 乗算器を付加した電流制御器

電流制御器同様にして、再度状態フィードバックをかけると、システムが複雑になるとともに、パラメータが増え、設計が困難になる。そこで、新たな制御対象に対する、出力電圧制御器を簡易に設計する手法が必要となる。

5.3.2 項で述べたように、電流制御器の  $r_i$  から  $i_L$  までの伝達関数は、式(5.12)である。このとき、 $r_i$  から  $v_o$  までの伝達関数は、式(5.12)に対し、極は等しく、零点と直流ゲインが異なるだけであるから、電流制御器の設計におけるパラメータの決定時に条件をつけることで、ある程度任意の位置に、極を配置することができる。よって、電流制御器の設計時に、 $r_i$  から  $v_o$  までの伝達関数も、1次モデルに近似できるよう、極配置を考慮することで、図 5.7 の新たな制御対象に直接ロバスト補償器を接続することができる。

では、新たな制御対象の特性を 1 次モデルに近似する手法を説明する。昇圧コンバータの離散化状態方程式では、状態変数  $v_o$  と  $i_L$  とを入れ替えると、直流ゲインと零点のみが変化する。そのため、 $u_v$  を入力、 $v_o$  を出力とした、新たな制御対象の伝達特性は、以下の式で表される。

$$v_o = G_{riv} \frac{v_{ac}}{v_{avg}^2} \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-p_{z1})}{(z-p_{z1})} \frac{(1-p_{z2})}{(z-p_{z2})} \frac{(z-n_{1v})}{(1-n_{1v})} \frac{(z-n_{2v})}{(1-n_{2v})} u_v \quad (5.13)$$

ただし、 $G_{riv}$  は、インダクタ電流目標値から出力電圧までの直流ゲイン、 $n_{1v}, n_{2v}$  は、 $v \rightarrow v_o$  特性の零点である。 $G_{riv}$  は、状態方程式の入力  $v$  から  $i_L$  までの直流ゲイン  $G_{vi} = C_{dt}(I - A_{dt})^{-1}B_{dt}$  と、状態方程式を用いると、以下の式で表される。

$$G_{riv} = \frac{a_{21} + \frac{a_{23} + b_{21}}{G_{vi}}}{1 - a_{22}} \quad (5.14)$$

また、 $v_{ac}$  に関しては、電流目標値を操作するものであり、制御器設計には関わらない。そこで、これを 0 から 1 に正規化するため、 $v_{ac}$  のピーク値  $V_i$  と置き換える。すると、式(5.13)は、以下となる。

$$v_o = G_{riv} \frac{V_i}{v_{avg}^2} \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-p_{z1})}{(z-p_{z1})} \frac{(1-p_{z2})}{(z-p_{z2})} \frac{(z-n_{1v})}{(1-n_{1v})} \frac{(z-n_{2v})}{(1-n_{2v})} u_v \quad (5.15)$$

ここで、零点  $n_{1v}$ ,  $n_{2v}$  は単位円外に出て、 $|n_{1v}|, |n_{2v}| >> 1$  となるため、伝達特性には影響を与えない。直流ゲインを打ち消して、入出力のゲインを 1 とするゲイン  $G_v$  を追加し、入力  $r_v'$  から出力  $v_o$  の伝達特性を考えると、極は、 $H_1$ ,  $H_2$ , と、 $H_3$ ,  $k_z$  によって値が決まる  $p_{1i}$ ,  $p_{2i}$  に置かれる。一般に、5.3.1 項に述べた、昇圧コンバータの時定数は大きく、電流制御器の零点  $n_{1i}$  は単位円に近い位置にあるため、 $H_2(\approx n_{1i})$  は  $|H_2| >> |H_1|$  となるから、先のパラメータ決定条件から、 $|H_1| >> |H_3|, |p_{z1}|, |p_{z2}|$  であり、同時に  $|H_2| >> |H_1|, |H_3|, |p_{z1}|, |p_{z2}|$  となり、伝達関数は次の 1 次モデル  $W_{mv}$  に近似される。

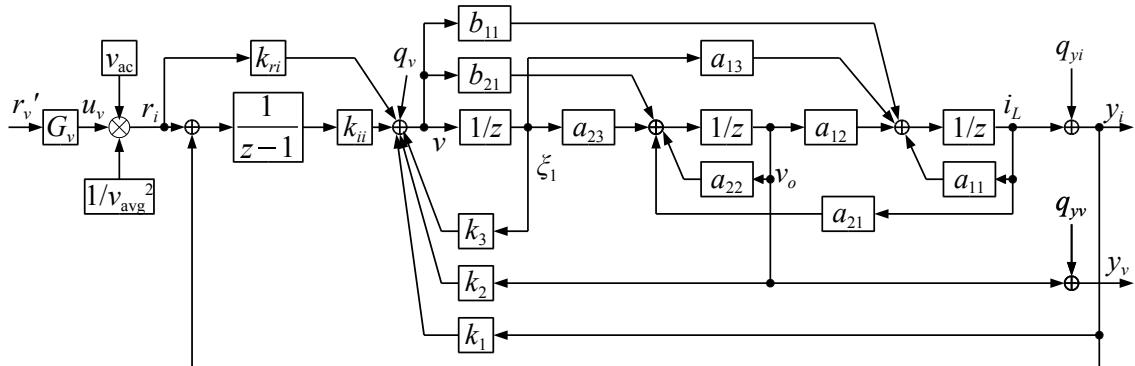
$$\begin{aligned} v_o &= W_{rv} r_v' = \frac{(1-H_1)}{(z-H_1)} \frac{(1-H_2)}{(z-H_2)} \frac{(1-p_{1i})}{(z-p_{1i})} \frac{(1-p_{2i})}{(z-p_{2i})} \frac{(z-n_{1v})}{(1-n_{1v})} \frac{(z-n_{2v})}{(1-n_{2v})} r_v' \\ &\approx \frac{1-H_2}{z-H_2} r_v' = W_{mv}(z) r_v' \end{aligned} \quad (5.16)$$

ただし、

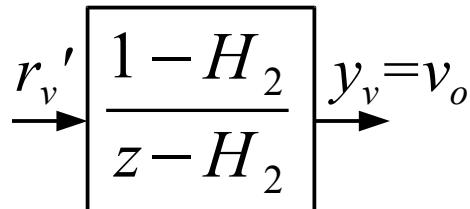
$$G_v = \frac{v_{avg}^2}{V_i G_{riv}}$$

である。

以下に、新たな入力  $r_v'$  と出力  $y_v$  を設けた、新たな制御対象を示す。



(a) 出力電圧を制御量とする新たな制御対象モデル



(b) (a)の近似表現

図 5.8 出力電圧を制御量とした新たな制御対象のモデルとその近似表現

### 5.3.3 電圧制御器の設計

図 5.7 の新たな制御対象の伝達関数が 1 次モデルに近似されたので、このモデルに、以下のロバスト補償器を接続する。

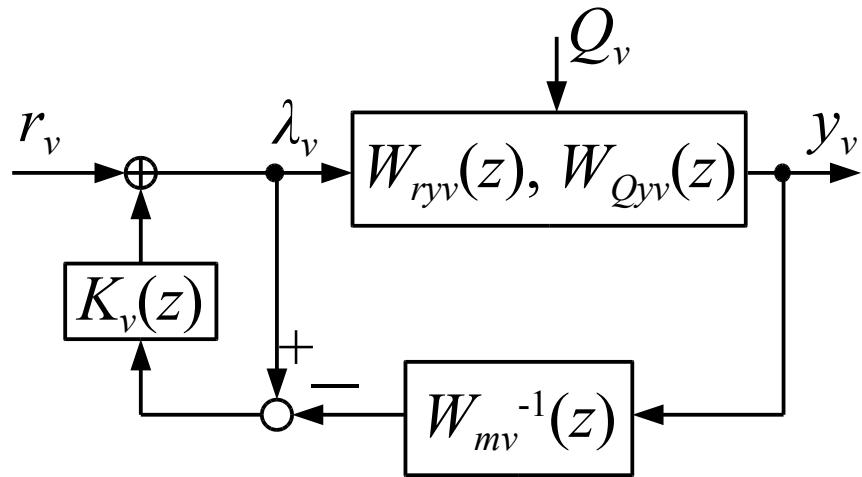


図 5.9 ロバスト電圧補償器

ここで、 $K_v$  は以下の式で表されるフィルタである。

$$K_v(z) = \frac{k_{zv}}{z - 1 + k_{zv}} \quad (5.17)$$

図 5.9 を等価変換して図 5.8 のモデルに接続すると、以下の図 5.10 の、近似的 2 自由度デジタル積分形出力電圧制御器が得られる。

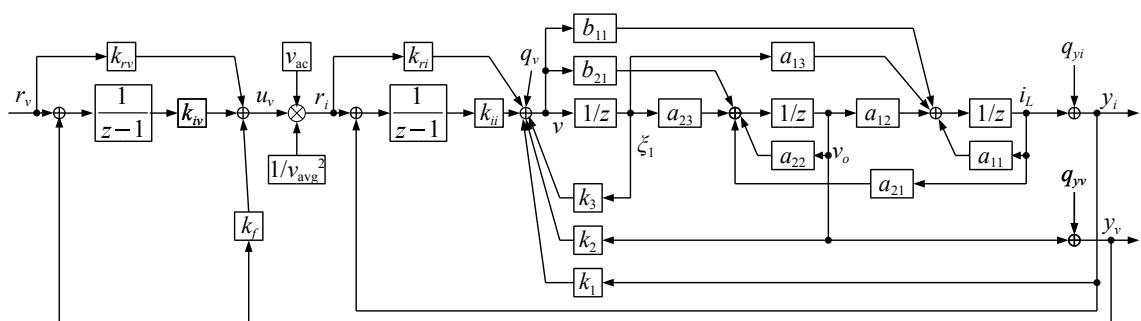


図 5.10 力率改善をする近似的 2 自由度デジタル積分形出力電圧制御器

ここで、各パラメータは、

$$k_f = -\frac{G_v k_{zv}}{1 - H_2}, \quad k_{iv} = G_v k_{zv}, \quad k_{rv} = G_v \quad (5.18)$$

である。

この図において、 $r_v$ ,  $v_o$ を入力とし、 $u_v$ を出力する部分が、出力電圧補償器であり、 $r_i$ と制御対象の各状態  $i_L$ ,  $v_o$ ,  $\xi_1$ を入力とし、 $v$ を出力する部分がインダクタ電流補償器である。

このシステムにおいて、 $r_v \rightarrow y_v$ 及び  $Q_v \rightarrow y_v$ の伝達関数は以下のようになる。

$$y_v = W_{mv}(z) \frac{z-1+k_{zv}}{z-1+k_{zv}W_{sv}(z)} W_{sv}(z) r_v \quad (5.19)$$

$$y_v = \frac{z-1}{z-1+k_{zv}} \frac{z-1+k_{zv}}{z-1+k_{zv}W_{sv}(z)} W_{Qyv}(z) Q_v \quad (5.20)$$

ただし、

$$W_{Qyv}(z) = [W_{Qyv}(z) \quad W_{Qyv}(z)]$$

$$W_{sv}(z) = \frac{(1-H_1)}{(z-H_1)} \frac{(1-p_{1i})}{(z-p_{1i})} \frac{(1-p_{2i})}{(z-p_{2i})} \frac{(z-n_{1v})(z-n_{2v})}{(1-n_{1v})(1-n_{2v})}$$

である。

ここで、 $W_{sv}(z) \approx 1$ であれば、

$$y_v = W_{mv}(z) r_v \quad (5.21)$$

$$y_v = \frac{z-1}{z-1+k_{zv}} W_{Qyv}(z) Q_v \quad (5.22)$$

となり、 $r_v \rightarrow y_v$ 特性は  $H_2$ のみで、 $Q_v \rightarrow y_v$ の伝達特性は、 $k_{zv}$ で決定される。ここから、図 5.10 のシステムは、出力電圧の制御に関しても近似的に 2 自由度系を構成している。

図 5.10 のシステムは、以下の制御アルゴリズムで表され、これをマイクロコントローラに実装して実行することで、システムの制御演算を行うことができる。

$$\begin{aligned} u_v &= k_{iv} * v_{iv} + k_f * v_o + k_{rv} * r_v \\ v_{iv} &= ar_v - v_o + v_{iv} \\ r &= u_v * v_{ac} / v_{avg}^2 \\ v &= k_{ii} * v_{ii} + k_1 * i_L + k_2 * v_o + k_3 * \xi_1 + k_{ri} * r \\ v_{ii} &= r - i_L + v_{ii} \\ \xi_1 &= v \end{aligned}$$

また、4 章と同様に、 $k_{zi}$ ,  $k_{zv}$ が十分大きければ、図 5.10 の  $k_r$ ,  $k_{rv}$ のフィードフォワードは必ずしも必要ではない。

## 5.4 制御器パラメータの決定法

提案した図 5.10 のシステムでは、近似度を高め、かつ、外乱に対して低感度にするため、 $k_{zi}$ ,  $k_{zv}$ を大きな値にしなければならない。しかし、4.4 節と同様に、 $k_{zi}$ ,  $k_{zv}$ を大きくすると、式(5.7),(5.8),(5.19),(5.20)の

$$z - 1 + k_{zi} W_{si}(z) = 0 \quad (5.23)$$

の根が、電流制御器の代表根  $H_1$  に近づき、また、

$$z - 1 + k_{zv} W_{sv}(z) = 0 \quad (5.24)$$

の根が、電圧制御器の代表根  $H_2$  に近づき、 $W_{si}(z), W_{sv}(z) \neq 1$  となって、近似度が悪くなる可能性がある。さらに、電流制御器においては、電圧制御器を設計するための近似条件を満たさず、電圧制御器を正しく設計できなく可能性がある。また、電圧制御器においては、帯域が広がりすぎてしまい、電流制御器の動作に影響を与える可能性がある。

式(5.23)において、 $k_{zi}$  を 0 から大きくしたときの公称モデルの根軌跡の例を以下の図 5.11(a)に示す。根は 1,  $H_3$  より出発し、 $k_{zi}$  が任意の値の時、 $p_{1i}, p_{2i}$  の位置に移動する。 $k_{zi}$  が十分大きな値で、そのときの  $p_{1i}, p_{2i}$  が

$$|p_{1i}|, |p_{2i}| \ll |H_1| \quad (5.25)$$

となれば、その位置で近似度はよくなり、かつ外乱に対して低感度にすることができ、さらに電圧制御器設計の条件を満たす。

また、公称モデル以外の場合は、状態フィードバックによる極配置が正しく行われず、根は 1,  $H_1, H_3$  より出発し、 $k_{zi}$  が任意の値の時、 $p_{1i}, p_{2i}, p_{3i}$  の位置に移動する。このときの根軌跡の例を以下の図 5.11(b)に示す。このときは、 $k_{zi}$  が十分大きな値で、移動後の極(例では、 $|p_{2i}|, |p_{3i}| \gg |p_{1i}|$  であるため、特に  $p_{2i}, p_{3i}$ )による特性が、 $H_1$  のみの 1 次モデルの特性となるべく近くなるような位置にあり、かつ電圧制御器設計の条件を満たすため、

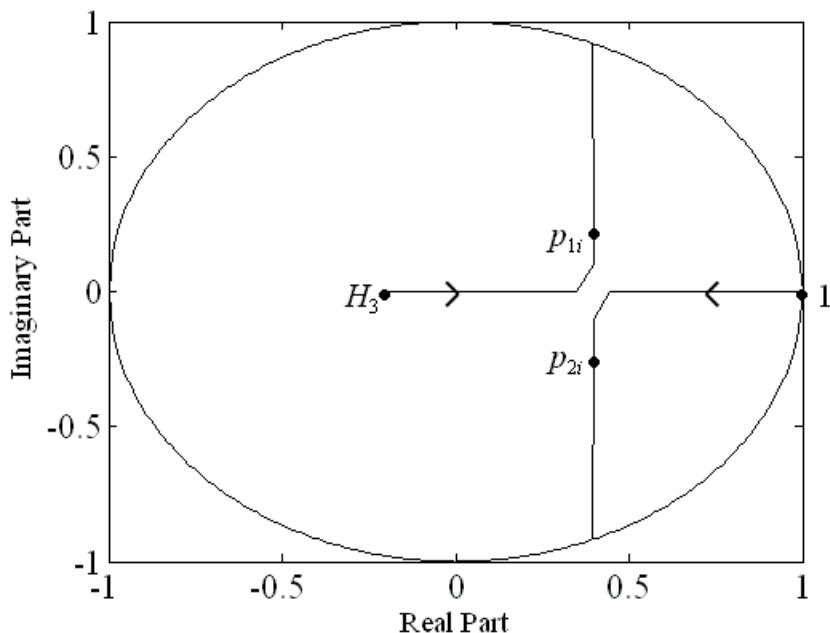
$$|H_2| \gg |p_{1i}|, |p_{2i}|, |p_{3i}| \quad (5.26)$$

となるようにパラメータを設定する必要がある。

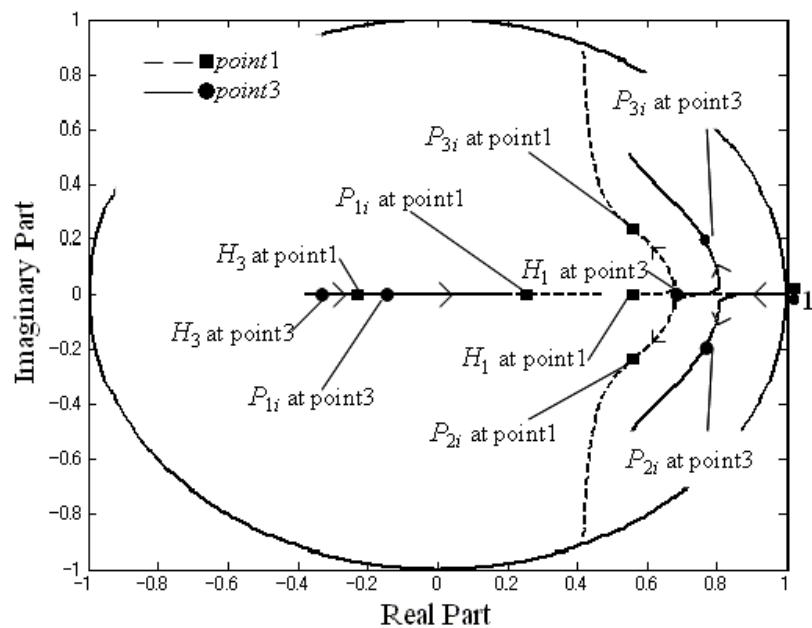
また、式(5.25)において、 $k_{zv}$  を 0 から大きくしたときの根軌跡の例を図 5.12 に示す。根は 1,  $H_1(p_{3i})$ ,  $p_{1i}, p_{2i}$  より出発し、 $k_{zv}$  が任意の値の時、 $p_{1v}, p_{2v}, p_{3v}, p_{4v}$  の位置に移動するが、力率改善制御器においては、電流制御器の帯域が電圧制御器よりも十分高いため、根はほとんど動かない。また、電流制御器の設計時に  $|H_2| \gg |H_1|, |p_{1i}|, |p_{2i}|, |p_{3i}|$  を満たしているため、 $k_{zv}$  が十分大きな値で、1 から出発し移動した先の極  $p_{4v}$  が

$$|p_{4v}| \ll |H_2| \quad (5.27)$$

となれば、その位置で近似度はよくなり、かつ外乱に対して低感度にすることができる。ただし  $k_{zv}$  が大きすぎると電流制御器に影響し、電流波形が歪んで高調波電流が増加するため、5.6 節の図 5.26 のように、2 重ループを構成した際の  $r_t-y_i$  のボード線図を見て、低域のカットオフ周波数が、入力電圧周波数以上にならないよう  $k_{zv}$  を調整する必要がある。



(a) 公称モデルの場合



(b) 公称モデル以外の動作点の場合

図 5.11 電流制御器の根軌跡の例

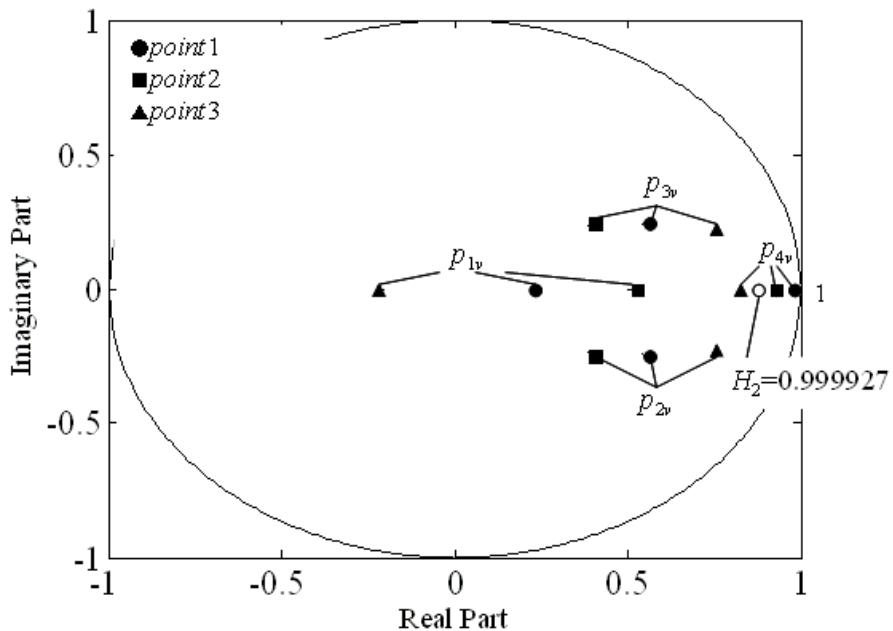


図 5.12 電圧制御器の根軌跡の例

そこで、制御器パラメータの決定手順を以下のように定める。

- (1) まず、公称モデルとなる動作点をひとつ決定する。
- (2) 電流制御器の代表根  $H_1$  を、入力電圧変化に追従できる任意の値に定める。
- (3)  $k_{zi}$  を  $k_{zi} \approx 0.5$  と指定する。
- (4)  $|H_3| << |H_1|$ ,  $H_2 \approx n_1$  を満たすように、 $H_2$ ,  $H_3$  を指定する。
- (5) シミュレーションによって、電流制御器が仕様を満たすことを確認する。
- (6) 仕様を満たさない場合、以下の優先順位でパラメータを調整する。
  - i. 根軌跡を見て、公称モデルにおいて  $|H_1| >> |p_{1i}|, |p_{2i}|$ , 他の動作点において  $|H_2| >> |p_{1i}|, |p_{2i}|, |p_{3i}|$  を満たすよう  $H_1, H_3$  を変更する。
  - ii.  $k_{zi}$  を変更する(範囲は  $0 < k_{zi} < 1$ )。
  - iii. 公称モデルを変更して、i と ii を繰り返す。
- (7) 続いて、電圧制御器の  $k_{zv}$  を  $k_{zv} \approx 0.5$  と指定する。
- (8) シミュレーションによって、電圧制御器が仕様を満たし、電流制御器に大きな影響を与えないことを確認する。
- (9) 仕様を満たさない場合、以下の優先順位でパラメータを調整する。
  - iv.  $k_{zv}$  を変更する(範囲は  $0 < k_{zv} < 1$ )。
  - v. 根軌跡と制御帯域を見て、 $|H_2| >> |p_{1v}|, |p_{2v}|, |p_{3v}|, |p_{4v}|$  を満たすよう電流制御器のパラメータを変更し、iv を繰り返す。

## 5.5 制御器の設計

力率改善制御器の設計にあたり、最大出力の高調波電流を低減することが求められるため、今回は公称モデルとして、最大出力の状態である、動作点②を用いる。

まず、サンプリング周波数 100[kHz]、遅れ時間  $0.99T_s$  の時の、制御対象となる昇圧コンバータの離散伝達関数は、以下の式となる。

$$G_{vi_L} = 0.25698 \frac{(z - 0.999927)(z + 93.29)}{z(z - 0.99924)(z - 0.88756)} \quad (5.28)$$

ここで、インダクタ電流を商用交流周波数 50[Hz]を整流した 100[Hz]の全波整流波形に追従させるため、 $H_1$ は、 $H_1=0.5$ とした。また  $H_3=0.1$ 、 $k_z=0.5$ と設定し、 $H_2 \approx n_1=0.999927$ と指定した。このとき、得られた制御器では、多少オーバーシュートを生じ、また、式(5.25)を満たさなかった。図 5.11 の根軌跡を見ると、根の移動距離を大きくし  $k_z$ を大きく、かつ式(5.25)を満たすようにするために、 $H_3$ の実部がなるべく左にある必要があることがわかった。そのため、 $H_3$ を左半平面で設定し直し、すべての動作点で安定であり、式(5.25)の条件を満たすようパラメータを微調整した結果、

$$H_1 = 0.5, \quad H_2 = 0.999927, \quad H_3 = -0.2, \quad k_z = 0.35 \quad (5.29)$$

と指定した。このパラメータは他の動作点において式(5.26)を満たした。このとき、式(5.11)より、制御器パラメータは以下のように得られる。

$$\begin{aligned} k_1 &= -0.034705, & k_2 &= 0.0014317 \\ k_3 &= -0.58241, & k_{ii} &= 0.0086668 \end{aligned} \quad (5.30)$$

また、式(5.25)を満たしているため、 $k_{ri}$ のフィードフォワードゲインは 0 に設定した。

続いて、電圧制御器のパラメータ  $k_{zv}=0.5$  と設定した。このとき得られた制御器は、安定であり、 $H_1$ にあった極も大きく移動しておらず、式(5.27)の条件を満たした。しかしながら、図 5.26 の  $r_i \rightarrow i_L$  のボード線図から、低域カットオフ周波数が 50[Hz]を超えて、電流制御器の特性に影響を与えることが懸念された。実際の、SimPowerSystems のシミュレーションと実機実験結果から、電流波形が歪み、高調波電流が増加することが確認されたため、制御帯域を抑え、電流制御器に影響を与えることなく、 $k_z$ の値を調整した結果、

$$k_{zv} = 0.25 \quad (5.31)$$

と指定した。このとき、式(5.18)より、制御器パラメータは以下のように得られる。

$$k_f = -45.153, \quad k_{iv} = 0.00329 \quad (5.32)$$

また、このとき、式(5.27)の条件を満たしたため、 $k_{rv}$ のフィードフォワードゲインは 0 に設定した。

## 5.6 シミュレーションとその結果

図 5.13 に示す SIMULINK モデルを用いて、前項で得られた式(5.29)のパラメータを用いた電流制御器の、各動作点における、ステップ応答,  $r_i \rightarrow y_i$  間ボード線図, 開ループボード線図,  $q_{yi} \rightarrow y_i$ ,  $q_v \rightarrow y_i$  間ボード線図, それぞれのシミュレーションを行った。ステップ応答結果を図 5.14, ボード線図を図 5.15~18 に示す。

また、図 5.19 に示す SIMULINK モデルを用いて、式(5.31)のパラメータを用いた電圧制御器の、ステップ応答,  $r_v \rightarrow y_v$ ,  $r_v \rightarrow y_v$  間ボード線図, 開ループボード線図,  $q_{yv} \rightarrow y_v$ ,  $q_v \rightarrow y_v$  間ボード線図, それぞれのシミュレーションを行った。ステップ応答結果を図 5.20, ボード線図を図 5.21~25 に示す。

いずれの動作点に置いても、出力電圧のステップ応答波形とステップ応答時間はほぼ同じであり、またオーバーシュートはほぼ無く、仕様を満たすことがわかる。

また、SimPowerSystems を用いて、図 5.26 のモデルを作成し、定常状態の電流追従波形と負荷急変応答波形のシミュレーションを行った。この結果を図 5.27, 5.28 に示す。定常状態において、インダクタ電流は入力電圧の全波整流波形によく追従しており、力率の改善が期待される。また、負荷急変時の出力電圧変動幅は 15[V]程度に抑えられており、出力電圧 385[V]に対して、約 3.9 [%]の変動であるため、これも仕様を満たす。

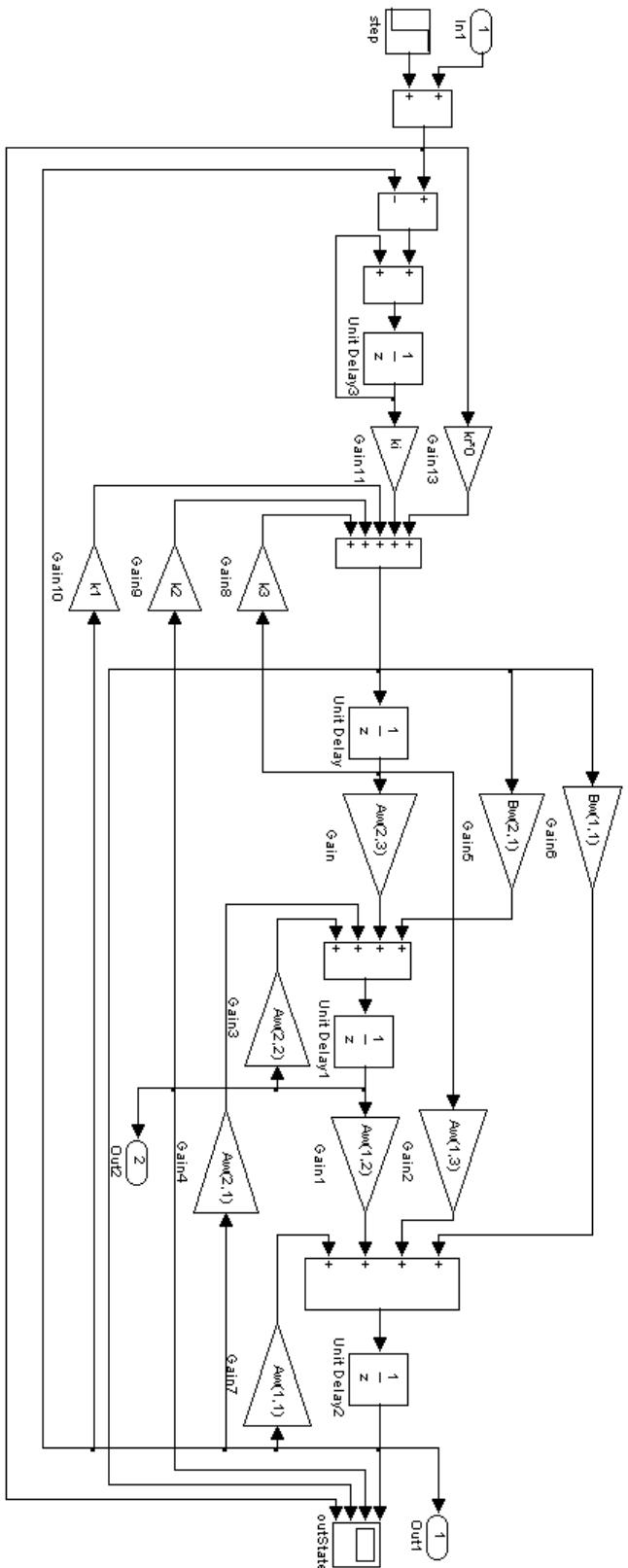


図 5.13 電流制御器の MATLAB/SIMULINK モデル

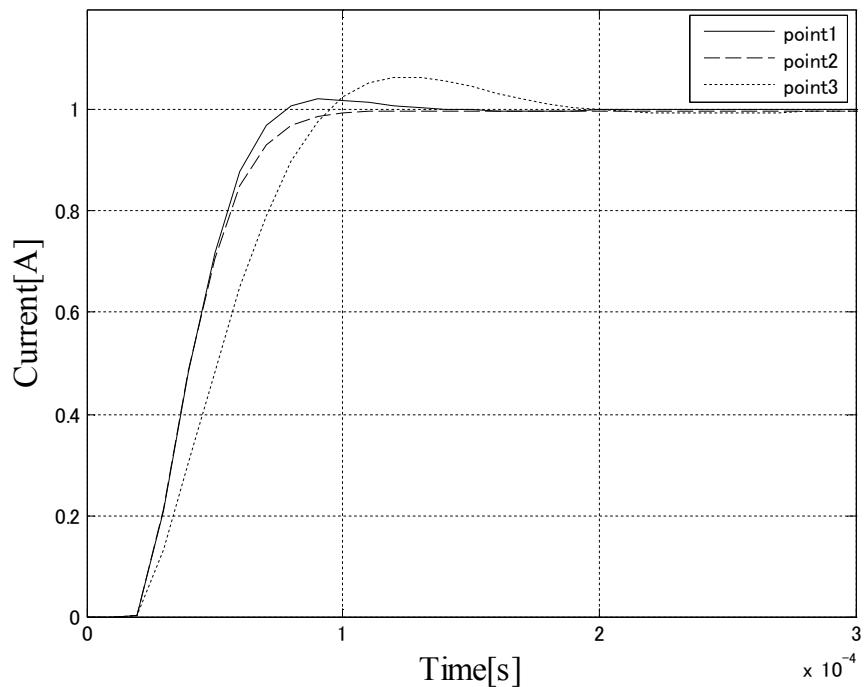


図 5.14 図 5.13 のモデルのステップ応答

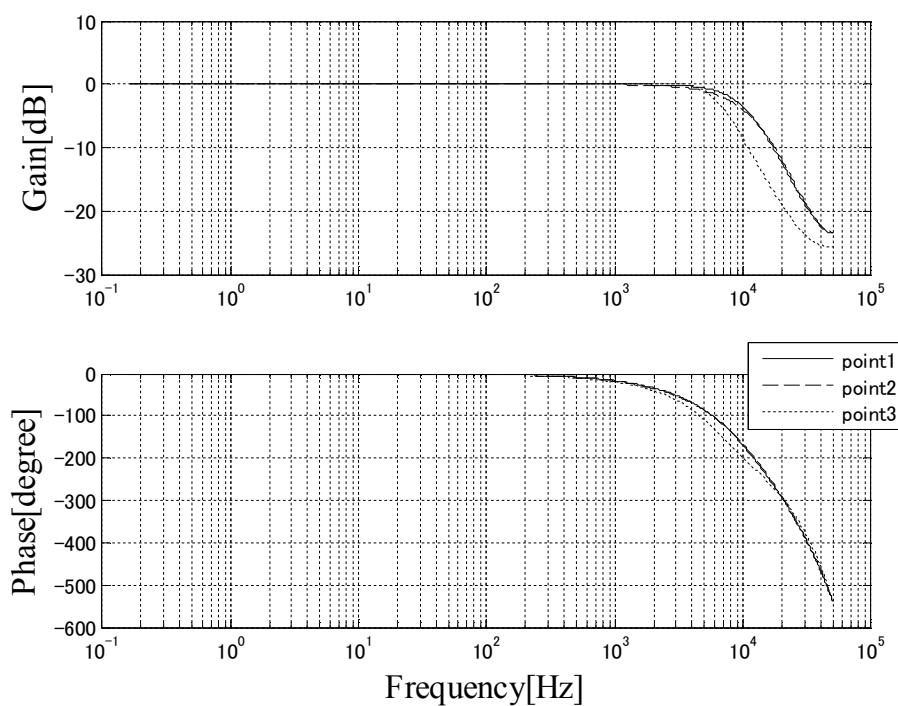


図 5.15 図 5.13 のモデルの閉ループボード線図

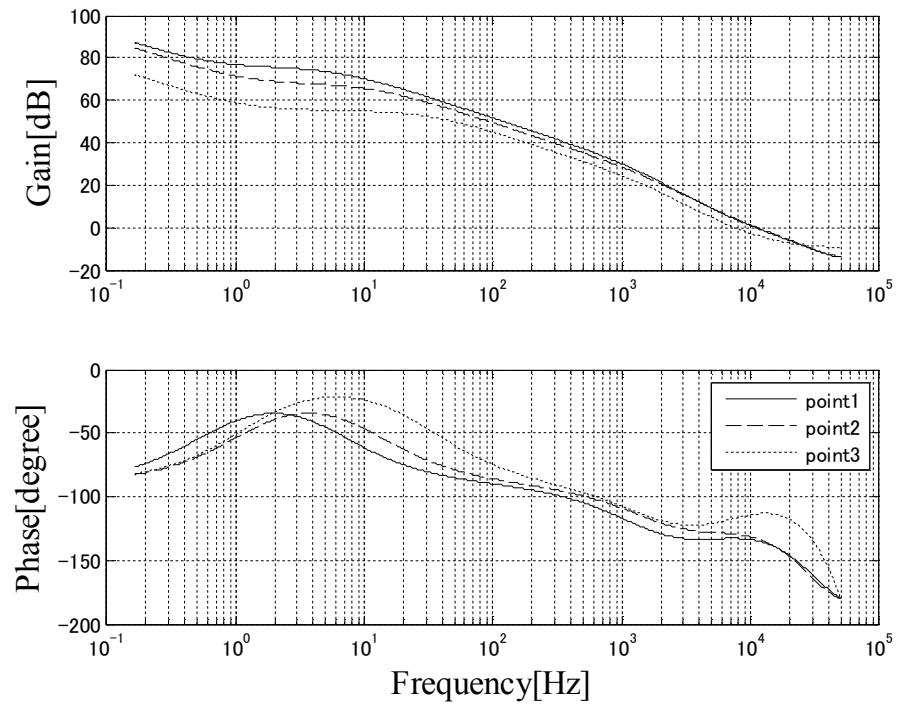


図 5.16 図 5.13 のモデルの開ループボード線図

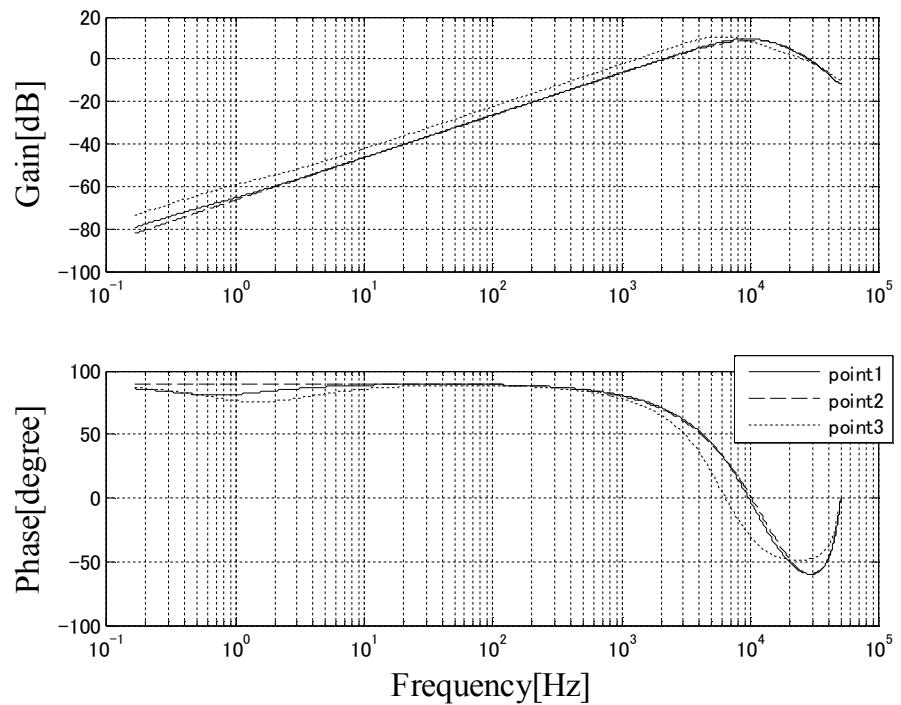


図 5.17 図 5.13 のモデルの  $q_{yi} \rightarrow y_i$  ボード線図

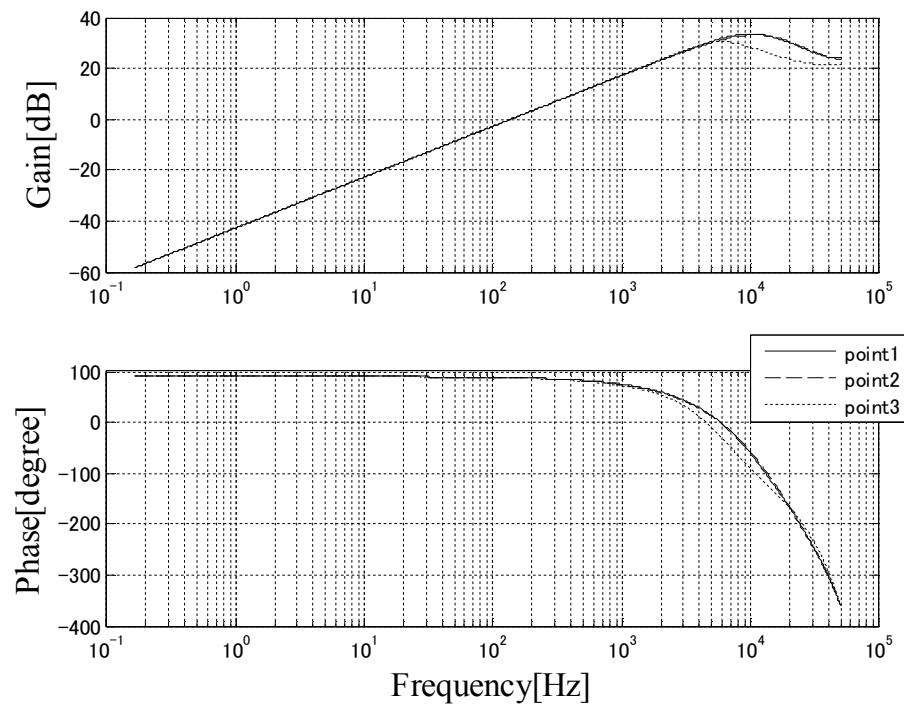


図 5.18 図 5.13 のモデルの  $q_v \rightarrow y_i$  ボード線図

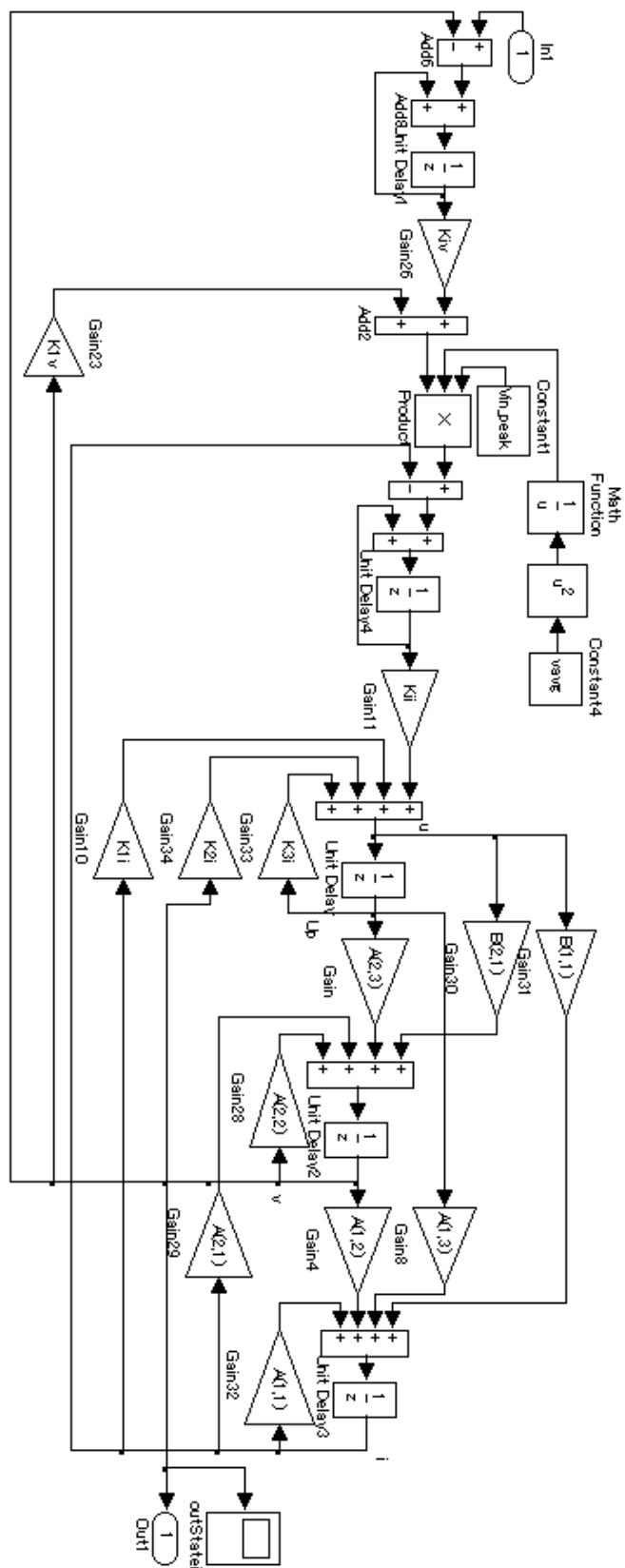


図 5.19 力率改善電圧制御器の MATLAB/SIMULINK モデル

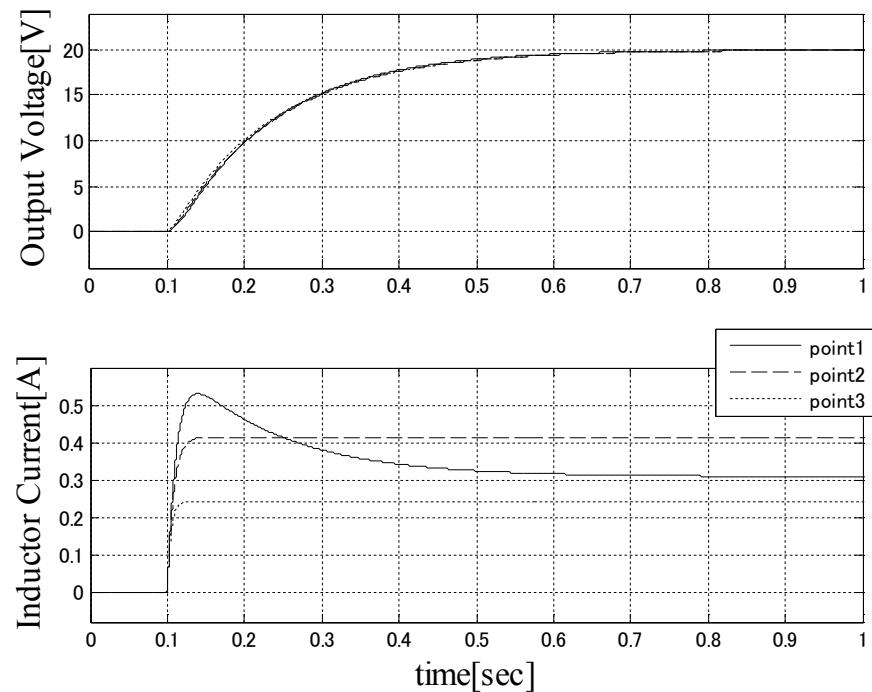


図 5.20 図 5.19 のモデルのステップ応答

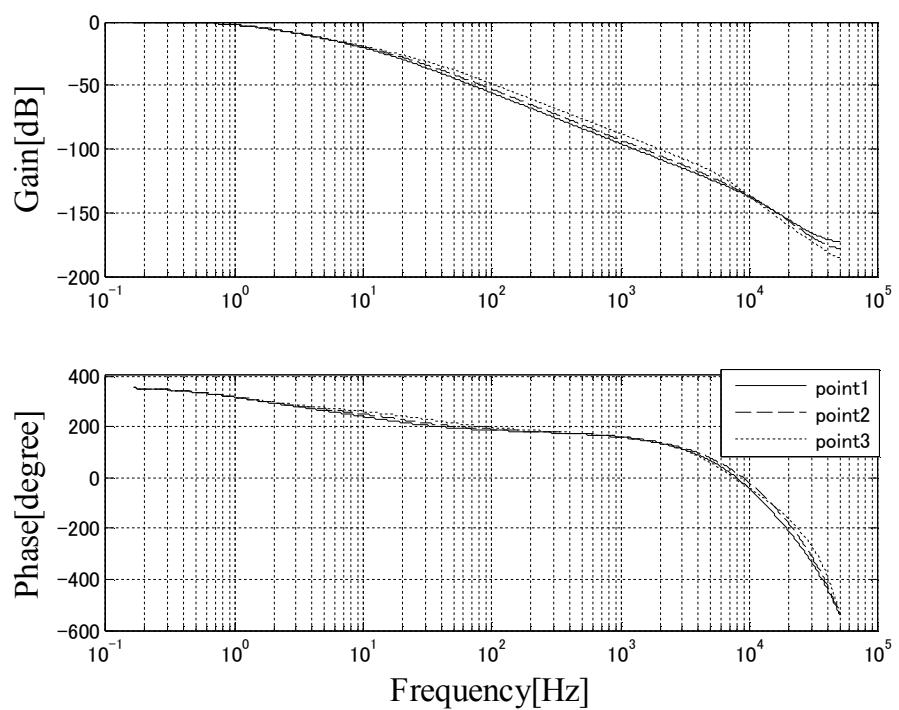


図 5.21 図 5.19 のモデルの  $r_v \rightarrow y_v$  閉ループボード線図

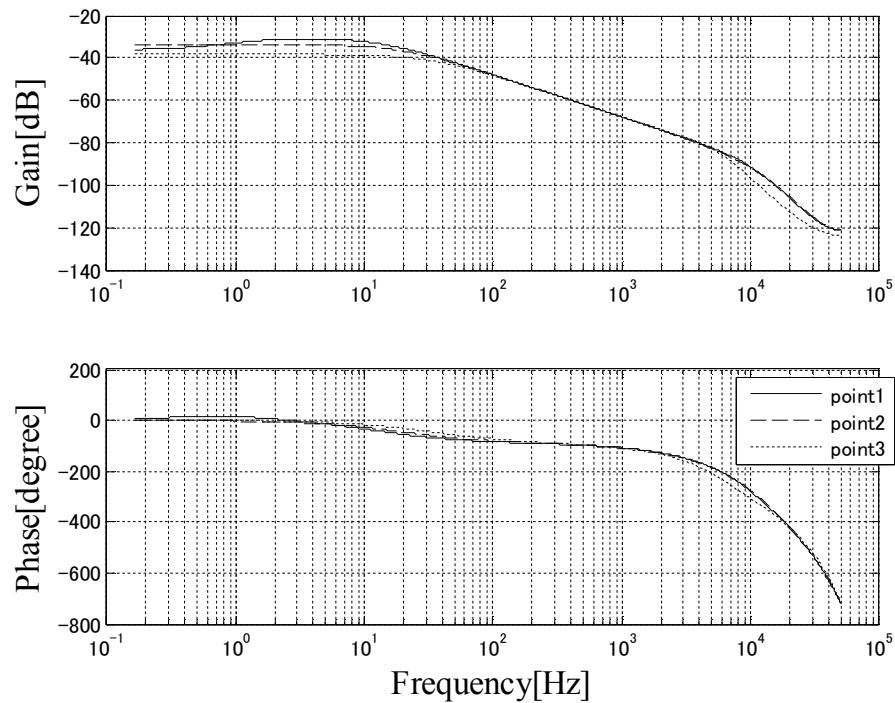


図 5.22 図 5.19 のモデルの  $r_v \rightarrow y_v$  閉ループボード線図

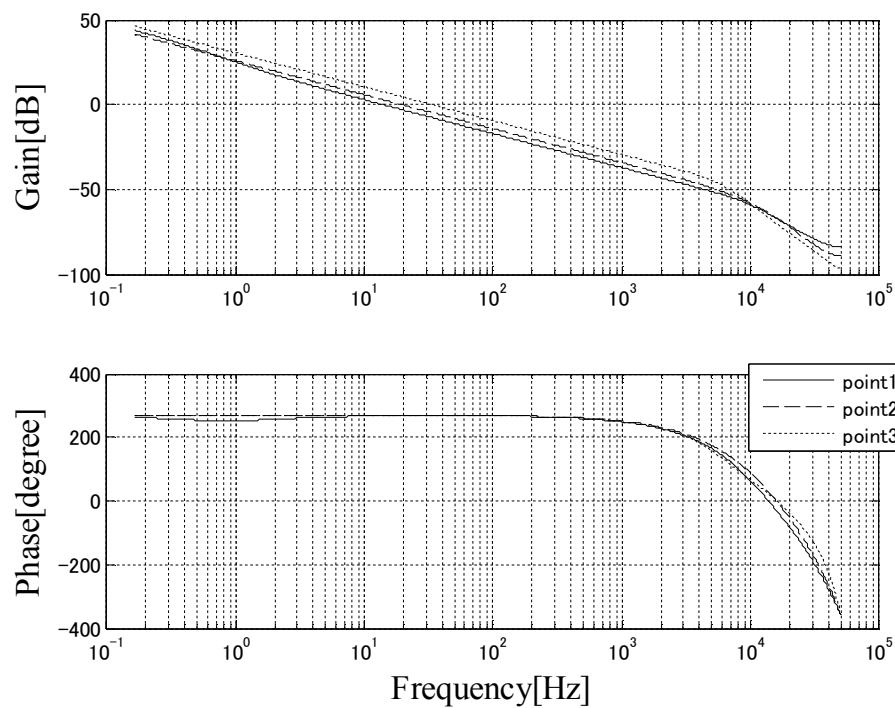


図 5.23 図 5.19 のモデルの開ループボード線図

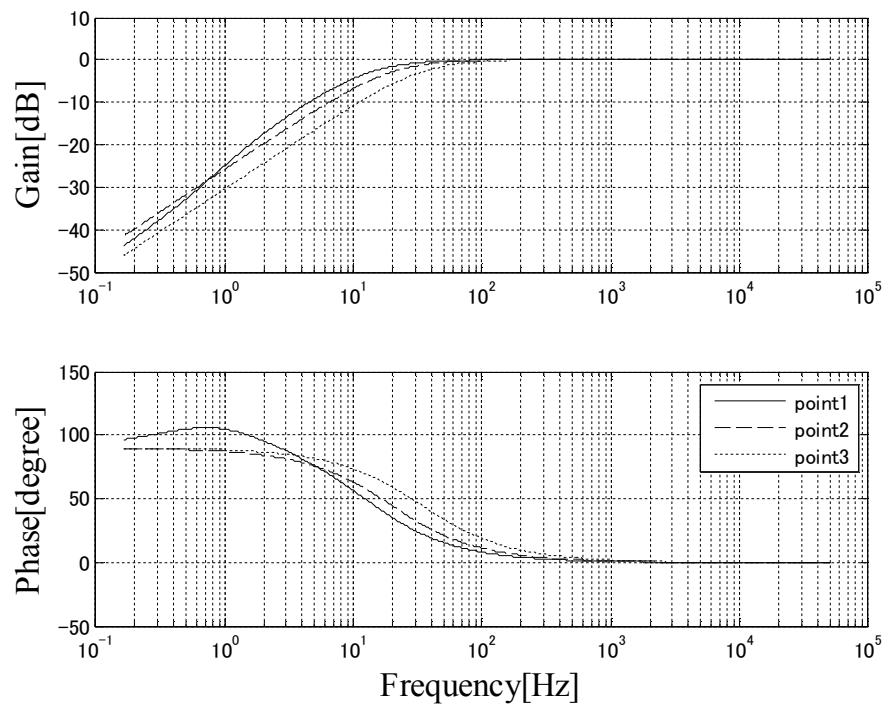


図 5.24 図 5.19 のモデルの  $q_{yv} \rightarrow y_v$  ボード線図

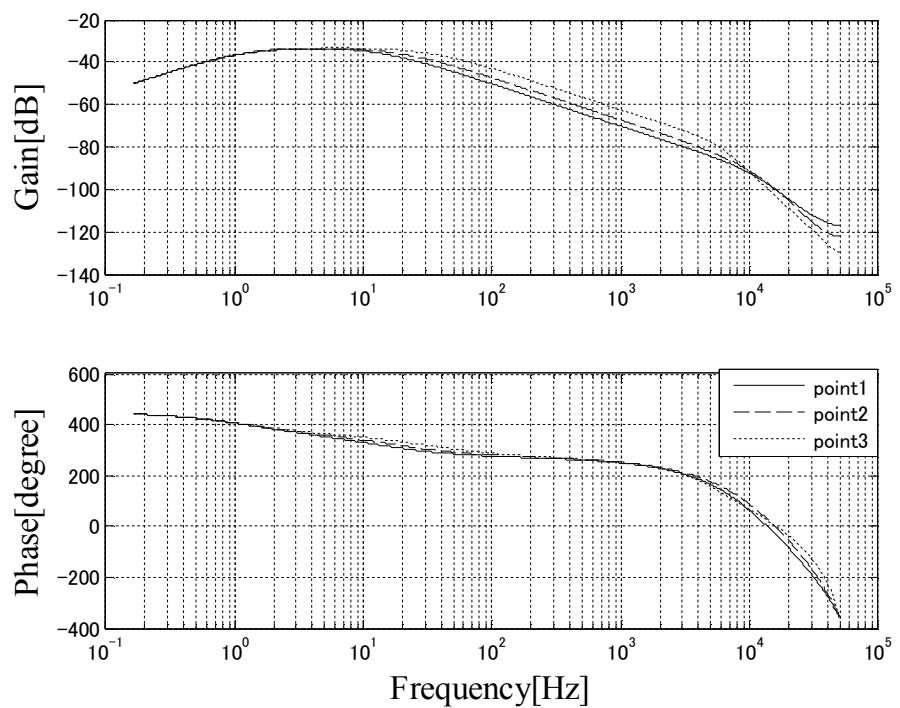


図 5.25 図 5.19 のモデルの  $q_v \rightarrow y_v$  ボード線図

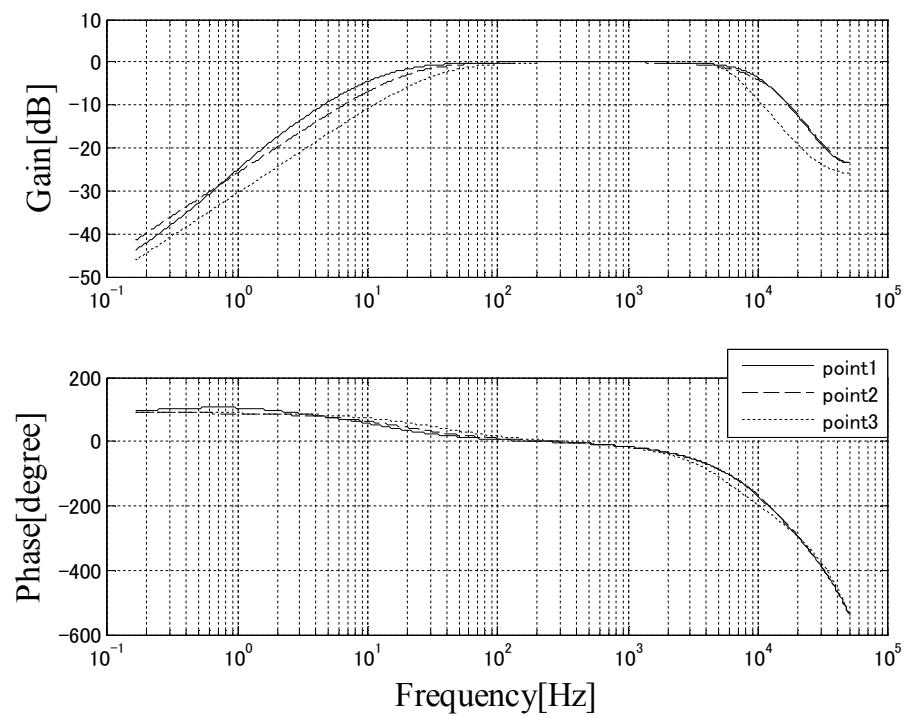


図 5.26 図 5.19 のモデルの  $r_i \rightarrow y_i$  ボード線図

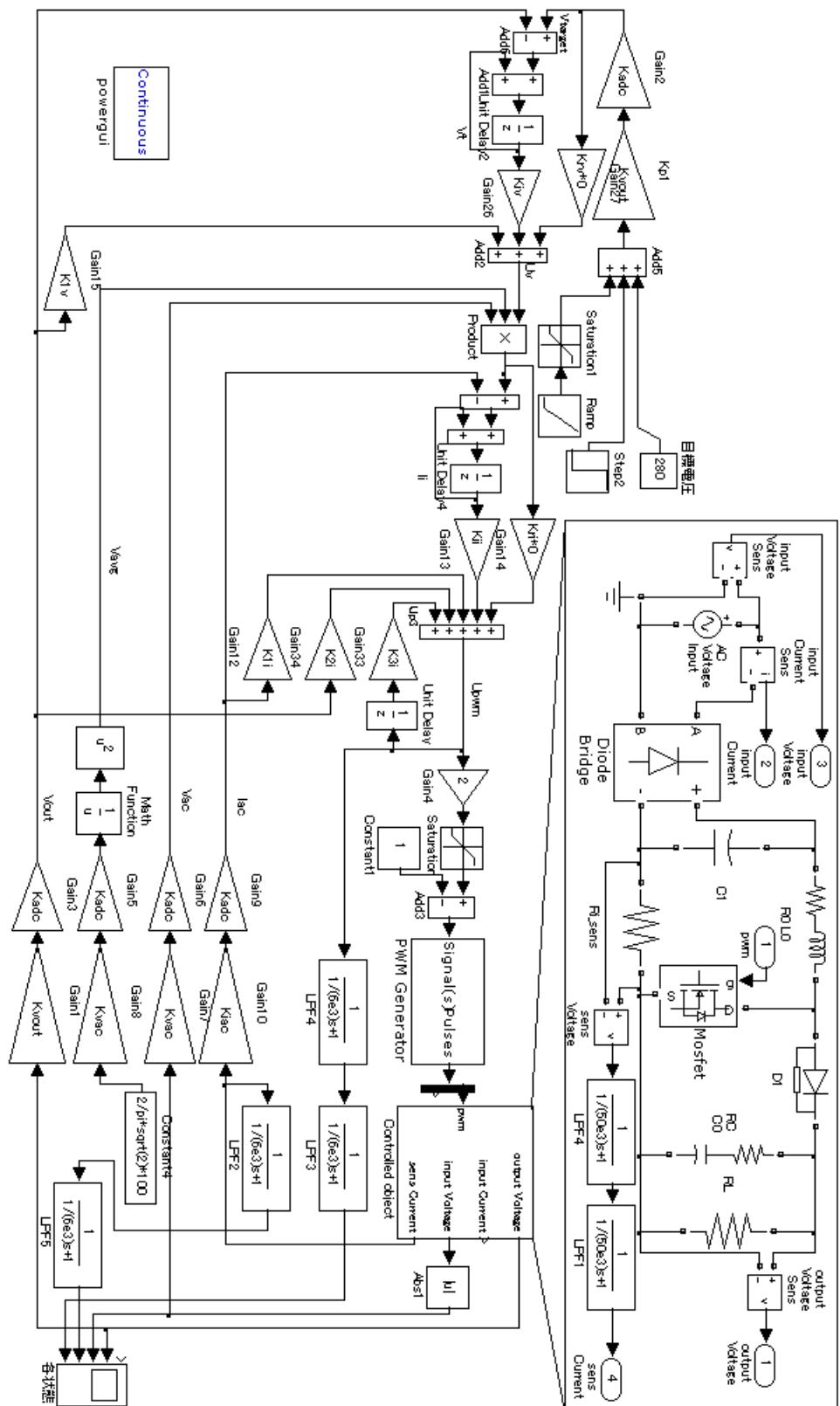


図 5.27 力率改善制御器の SimPowerSystems モデル

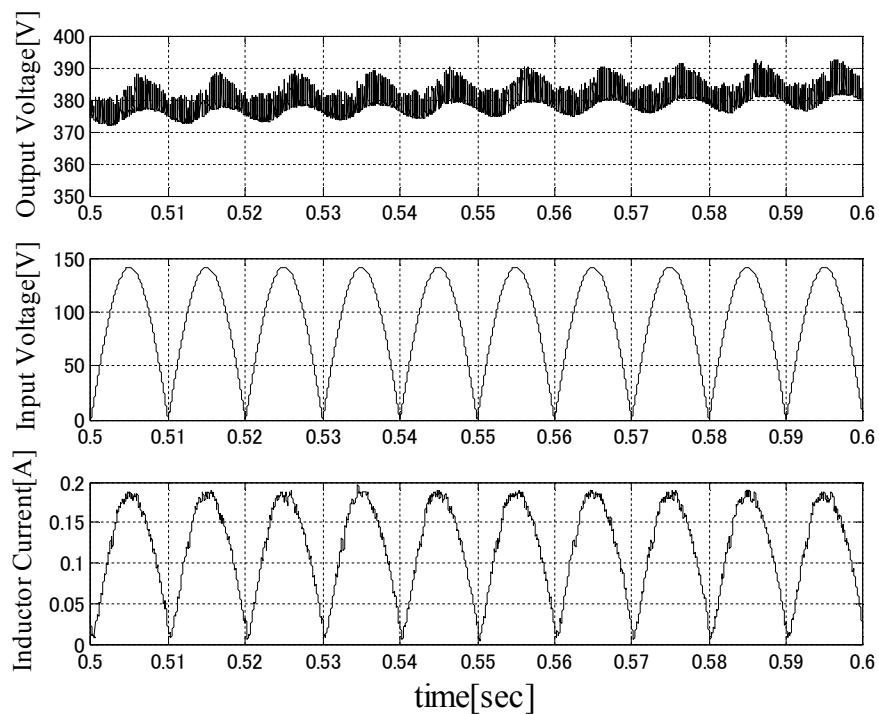


図 5.28 図 5.27 のモデルで SimPowerSystems を用いた定常状態力率改善シミュレーション結果

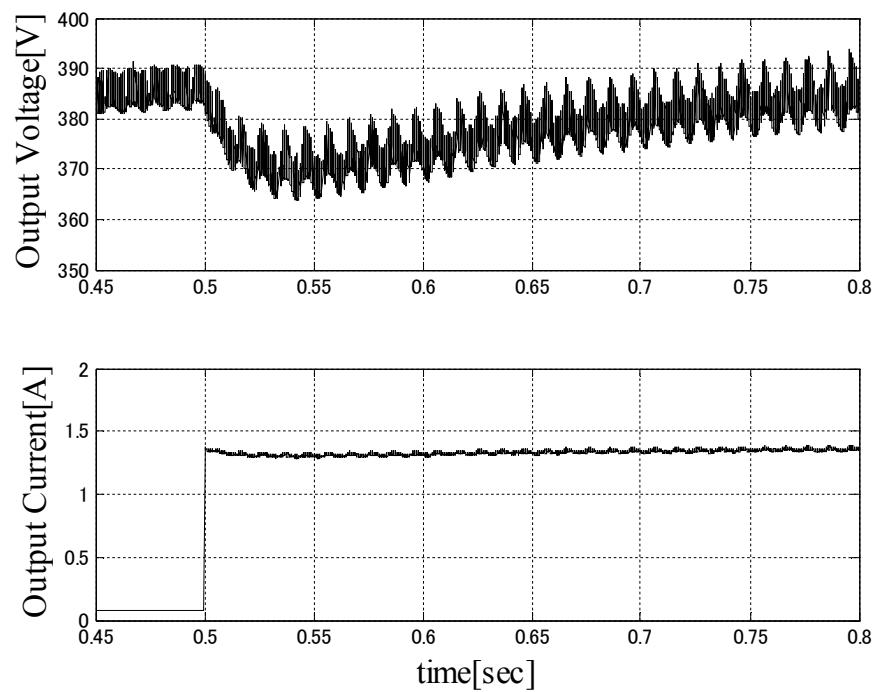


図 5.29 図 5.27 のモデルで SimPowerSystems を用いた負荷急変応答結果

## 5.7 実験結果

式(5.30), (5.32)の制御器パラメータを、前述の制御アルゴリズムを用いマイクロコントローラに実装して実行し、実機実験を行った。図 5.30 に、定常状態の各動作点における、各部波形を、図 5.31 に各動作点におけるステップ応答結果を、図 5.32 に、各動作点における電源投入時の立ち上がり波形を、それぞれ示す。電源投入時には、大電流が流れるため、AC 電源を投入してから、しばらくして制御を開始し、目標値を徐々に大きくするソフトスタートを用いた。

定常状態では、どの動作点においても、安定して立ち上がり、電流の追従ができていた。しかしながら、動作点①のような軽負荷の場合では、インダクタ電流が歪み、高調波成分が増えてしまった。

ステップ応答は、シミュレーションどおり、それぞれ同様の立ち上がり波形で、約 350[ms] で立ち上がっており、動作点の変化に対しロバストな結果が得られた。

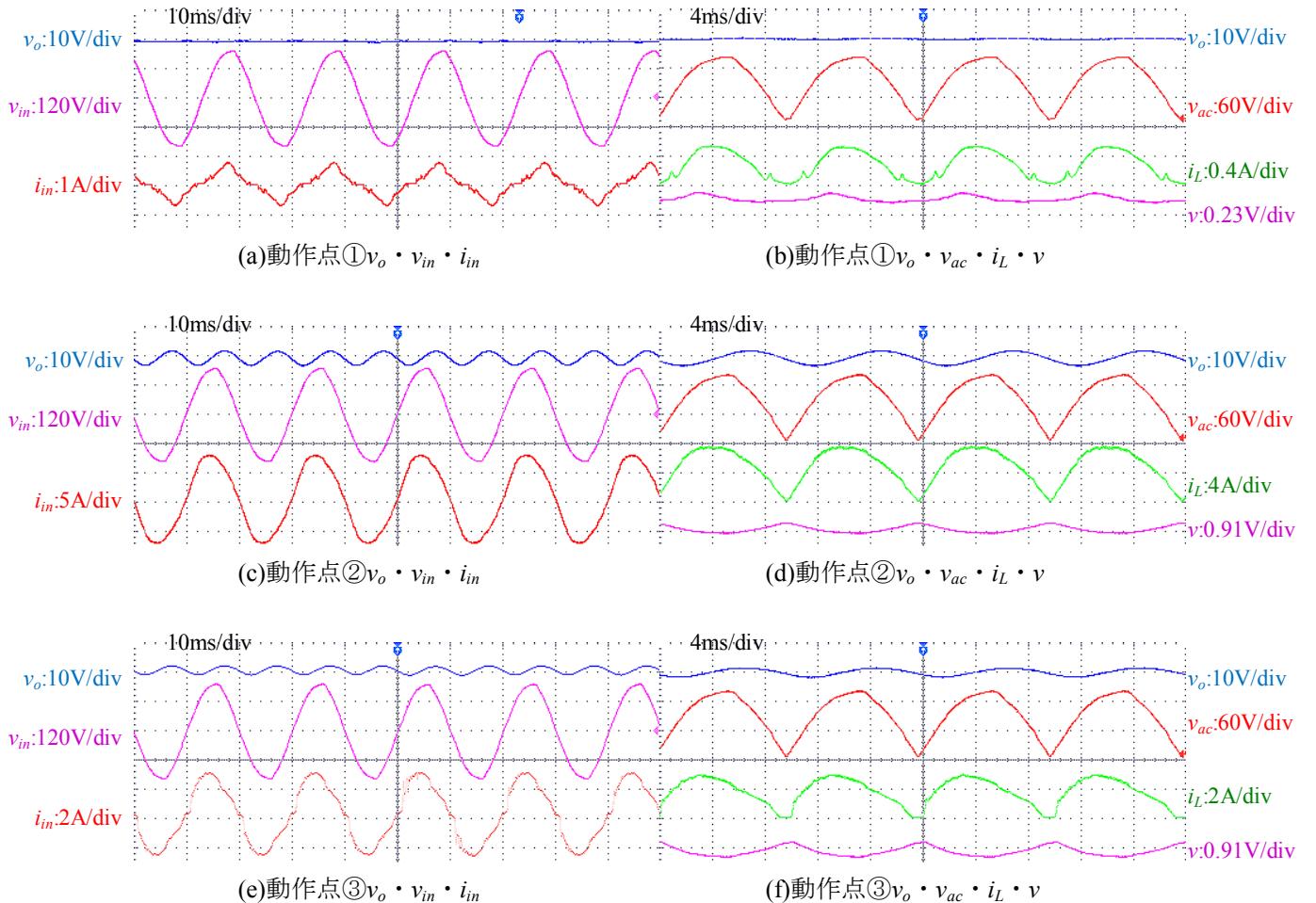


図 5.30 近似的 2 自由度力率改善制御器の各動作点の定常状態波形

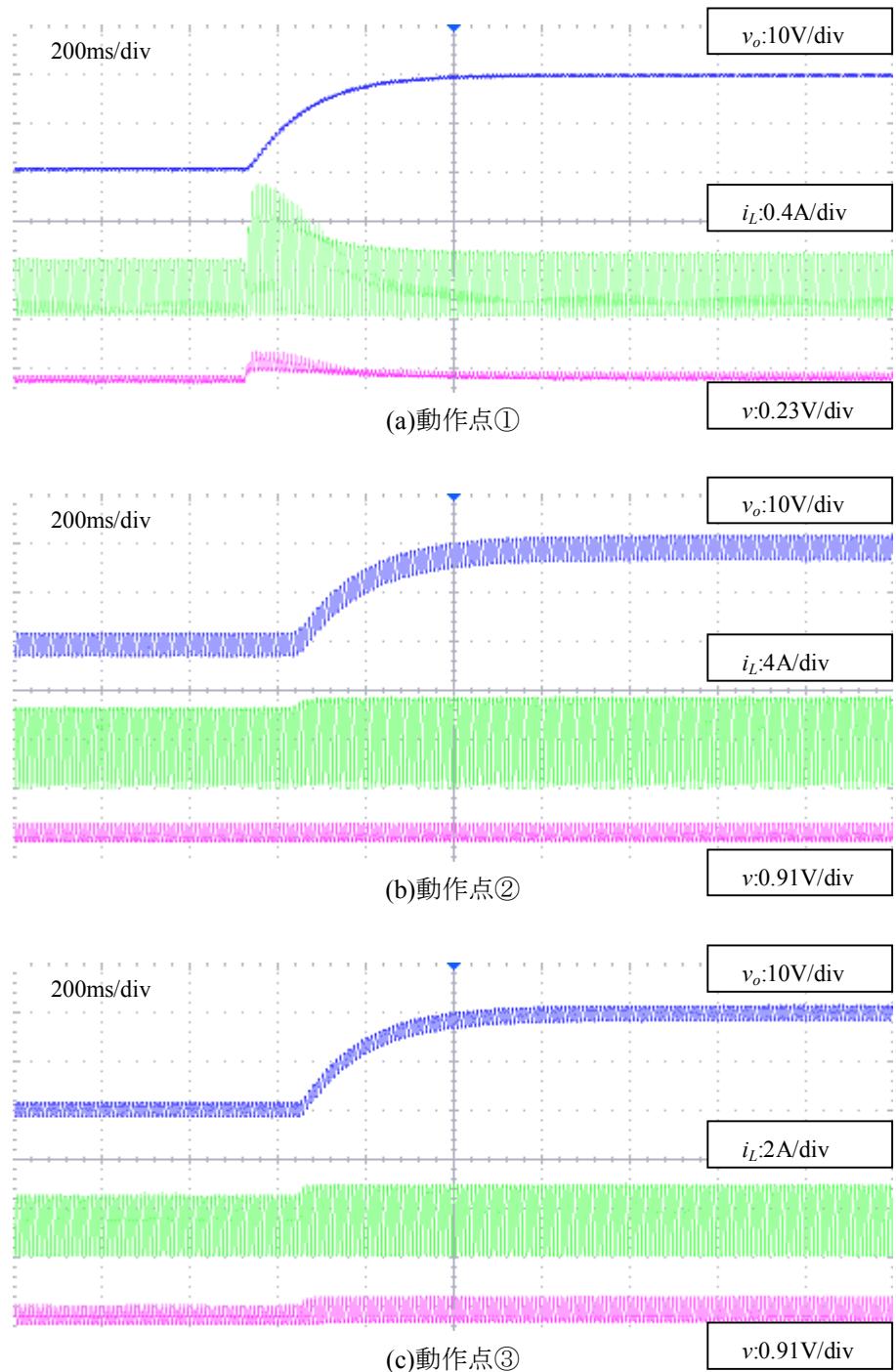


図 5.31 近似的 2 自由度力率改善制御器の各動作点のステップ応答結果

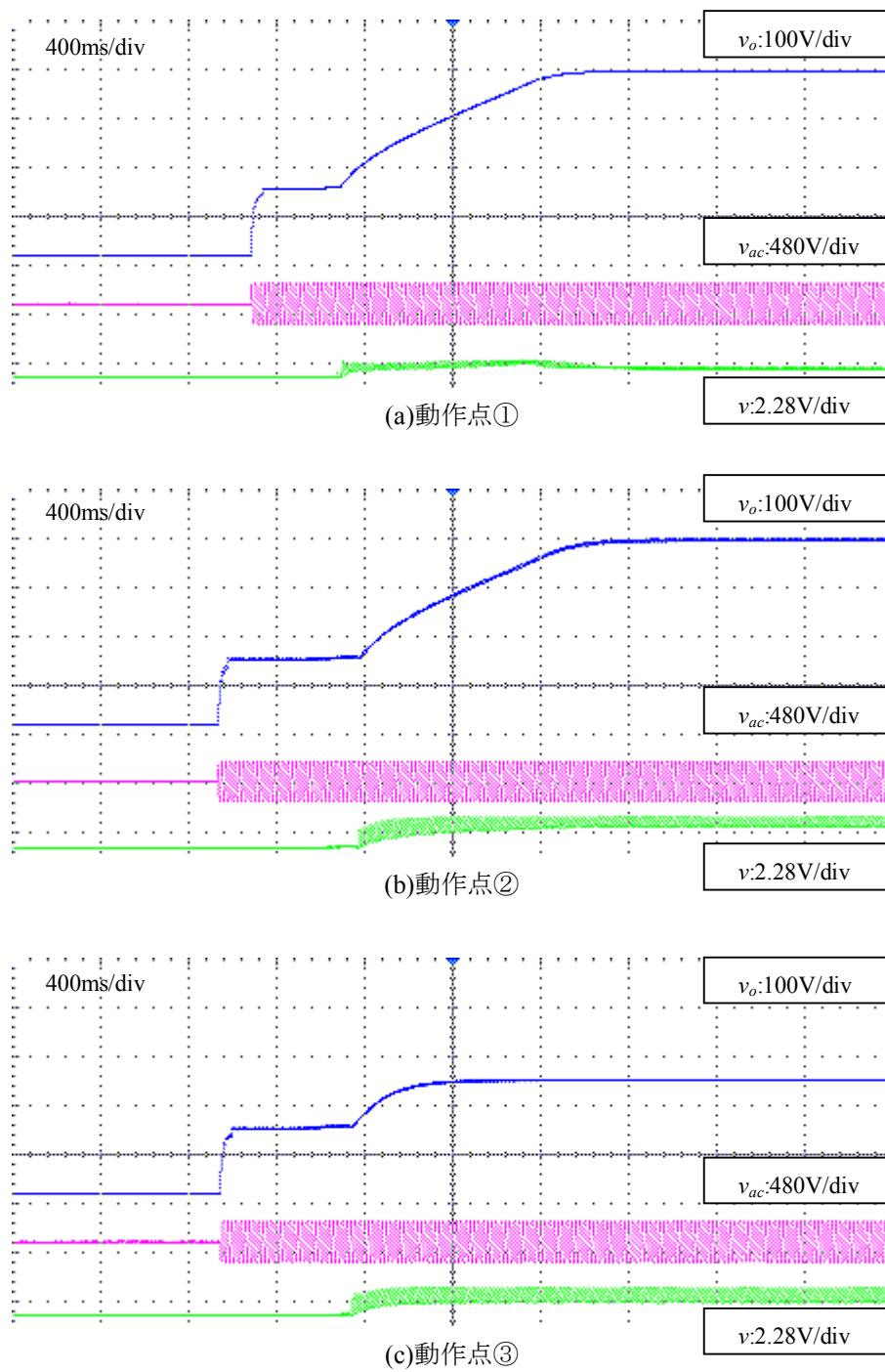


図 5.32 近似的 2 自由度力率改善制御器の各動作点の立ち上がり波形

次に、負荷急変時の出力電圧変動を測定した際の応答結果を、図 5.33 に示す。シミュレーションと同様、出力電圧の変動幅が 15[V]程度となり、十分ロバストな結果が得られ、仕様を満たした。 $1[\text{k}\Omega] \rightarrow 282[\Omega]$  の負荷急変では 10[V]以下にすることができた。さらに、効率特性、力率特性と、各動作点における入力電流の高調波特性を、それぞれ図 5.34, 35, 36 に示す。効率には補助電源の消費電力は考慮していない状態で、概ね 90 [%]程度であった。定常状態波形から、軽負荷の場合は電流波形が歪み、力率特性の悪化と、高調波電流の増加が見られる。しかしながら、力率は悪化しても、50[W]以上で 0.9 以上を保っており、全ての動作点で、IEC/EN 61000-3-2 に定められる規制値を大幅に下回り、仕様を満足した。

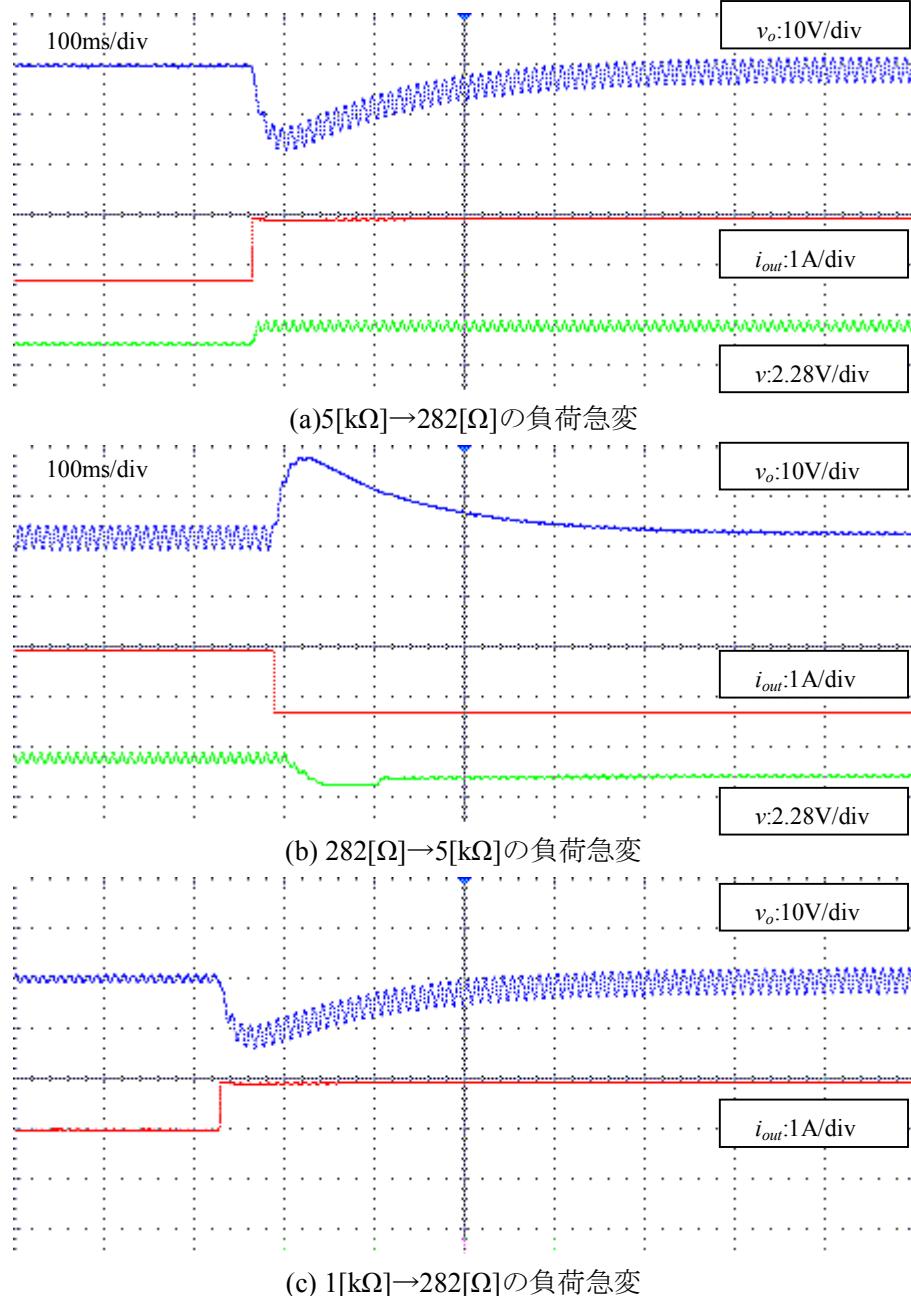


図 5.33 近似的 2 自由度力率改善制御器の負荷急変応答結果

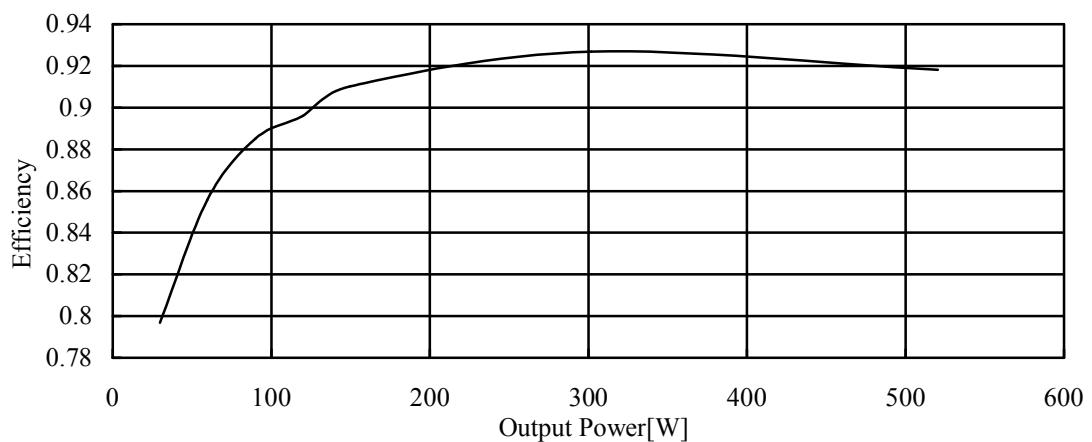


図 5.34 近似的 2 自由度力率改善制御器の効率特性

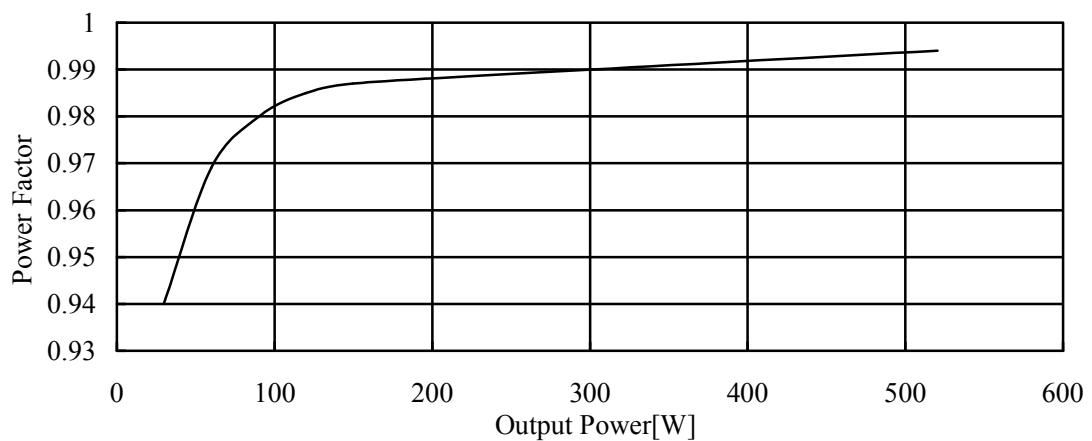
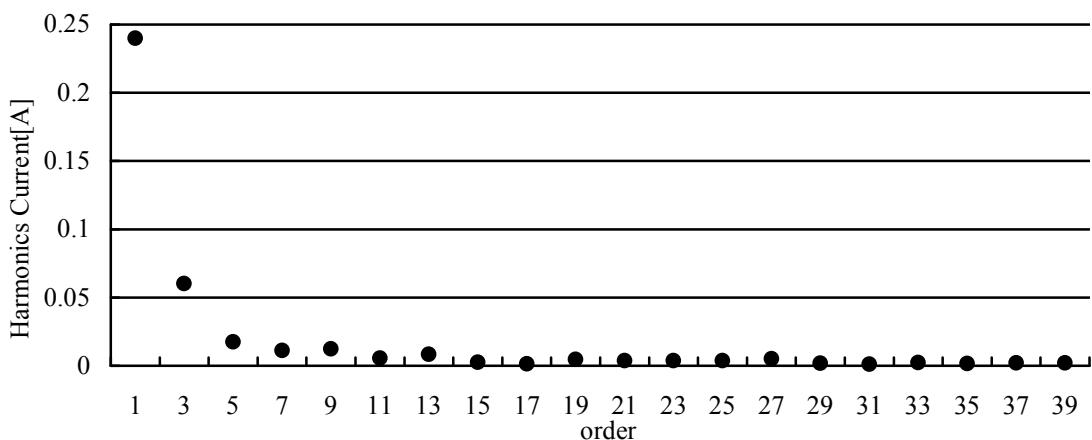
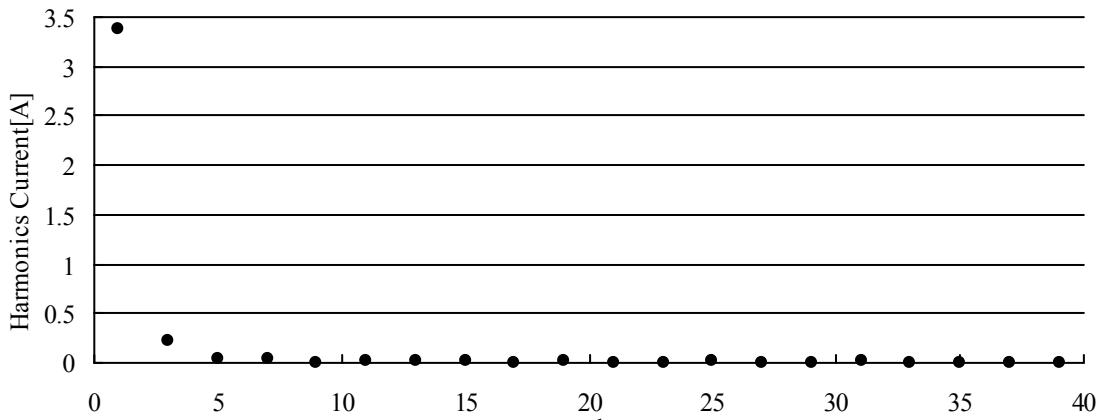


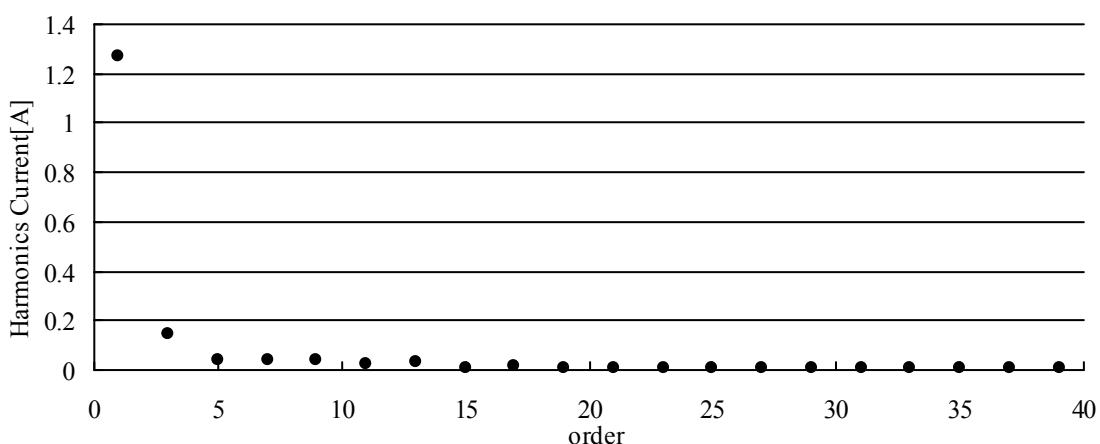
図 5.35 近似的 2 自由度力率改善制御器の力率特性



(a)動作点①



(b)動作点②



(c)動作点③

図 5.36 近似的 2 自由度力率改善制御器の各動作点の高調波電流特性

ここで、従来手法との比較として、アナログ IC : HA16174 を用いた場合と、図 5.37 に示す、デジタル PI 制御器を用いた場合の、実機実験を行った。アナログ IC のパラメータは、HA16174 評価ボードのアプリケーションノートを参考に決定した。デジタル PI パラメータは、すべての動作点で立ち上がり、かつ負荷急変に対し高速に応答する限界をトライアンドエラーで  $K_{pi}=0.05$ ,  $K_{ii}=0.01$ ,  $K_{pv}=3$ ,  $K_{iv}=0.001$  に決定した。アナログ制御器は目標値設定の変更がし辛いため、今回は動作点①と②の場合のみの比較を行った。

このアナログ制御器の定常状態波形、負荷急変応答波形、力率特性、高調波特性の結果を図 5.38 から図 5.41 に、デジタル PI 制御器の、ステップ応答、定常状態波形、負荷急変応答波形、力率特性、高調波特性の結果を図 5.42 から図 5.46 に、それぞれ示す。

アナログ制御器を用いた場合、負荷急変に対し高速に応答し、出力電圧変動は 5[V]以下と、大変良好な結果を得ることができているが、定常状態において、電流波形に大きなゼロクロス歪を生じ、電流波形が三角波に近い波形となっている。高調波特性は、規制値を下回って入るもの、全体的に大きな高調波を生じていることがわかる。

デジタル PI 制御器を用いた場合、電流波形と高調波特性は良い結果を得ることができているが、動作点の違いにより、ステップ応答の立ち上がり時間が違ってしまい、またオーバーシュートと、軽負荷では発振を生じる。さらに、動作点① $\leftrightarrow$ ②間の負荷変動時では出力電圧の変動幅が大きく実験出来なかった。予備的に負荷変動範囲を狭めて行った、負荷  $1[k\Omega]\leftrightarrow 282[\Omega]$  の負荷急変実験における出力電圧変動幅でも約 20[V]と、大きな変動を生じてしまった。

以上から、アナログ制御器では、仕様を満たすことは可能であったが、1.2 節で述べたように、機能拡張性が悪く、高調波電流が多く発生し、高性能化が困難であり、デジタル PI 制御器では、負荷急変時の出力電圧変動において仕様を満たすことが困難であることがわかった。近似的 2 自由度制御を力率改善制御器に適用することで、機能拡張性を有しながら、デジタル PI 制御器とほぼ変わらない計算時間で、高性能化し、仕様を満たすことができる。

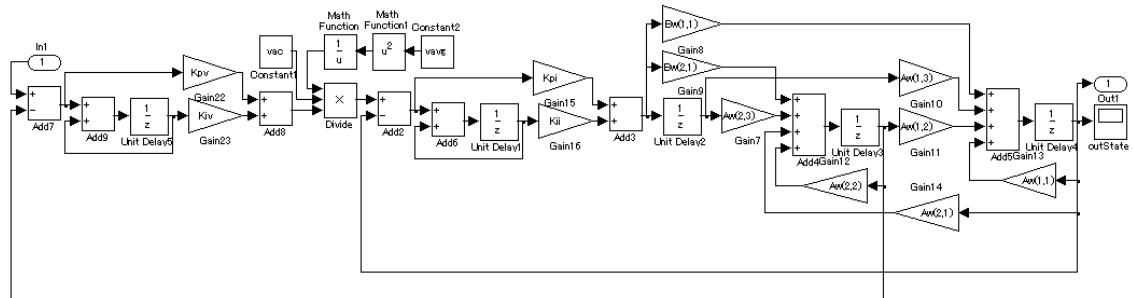


図 5.37 デジタル PI 力率改善制御器の MATLAB/SIMULINK モデル

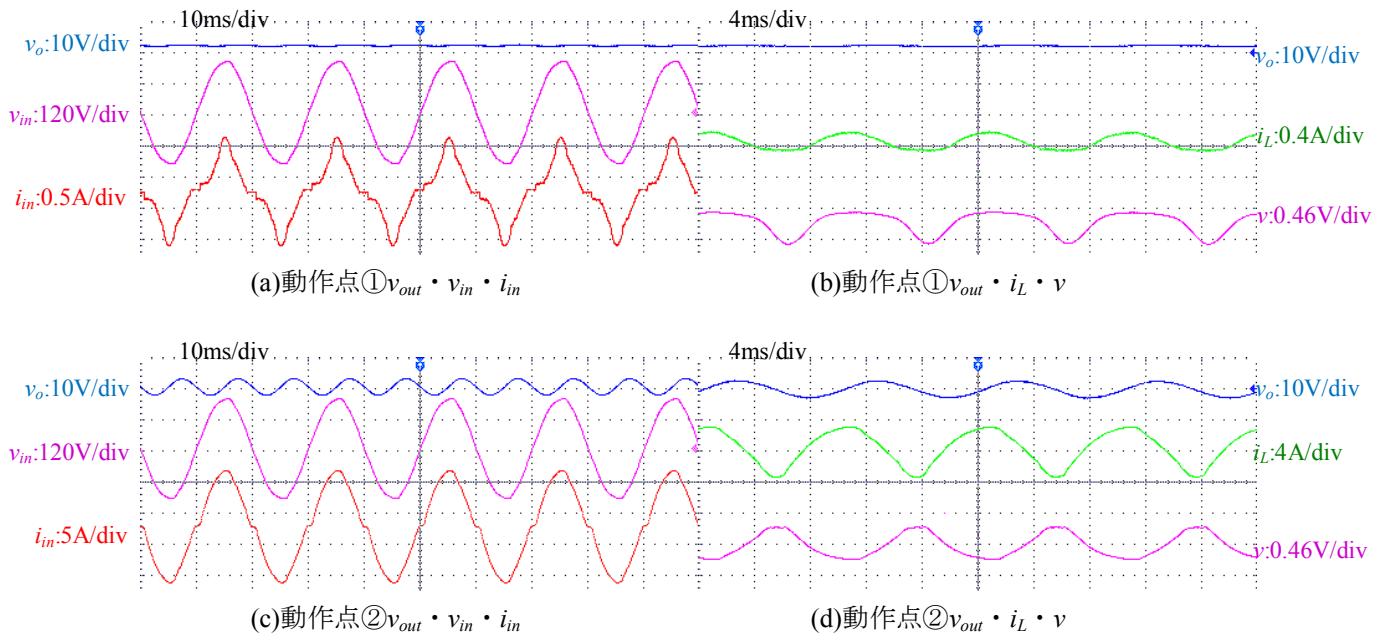


図 5.38 アナログ力率改善制御器の各動作点の定常状態波形

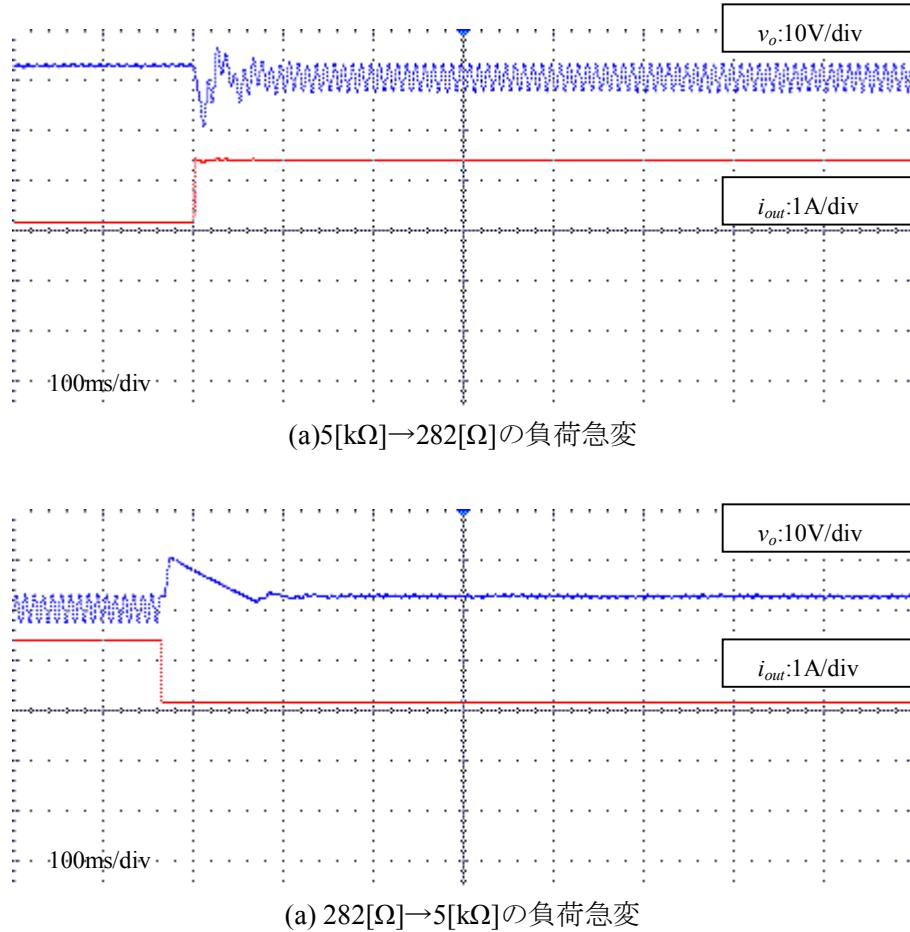


図 5.39 アナログ力率改善制御器の負荷急変応答結果

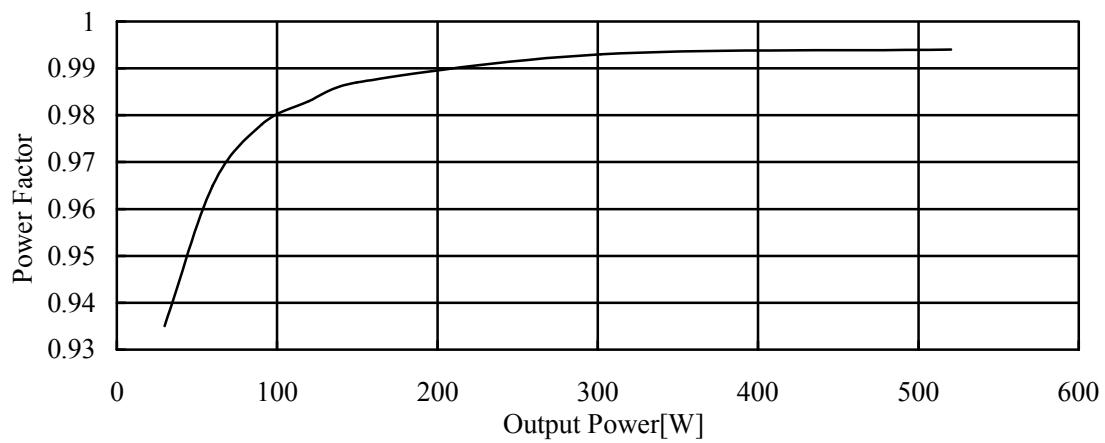


図 5.40 アナログ力率改善制御器の力率特性

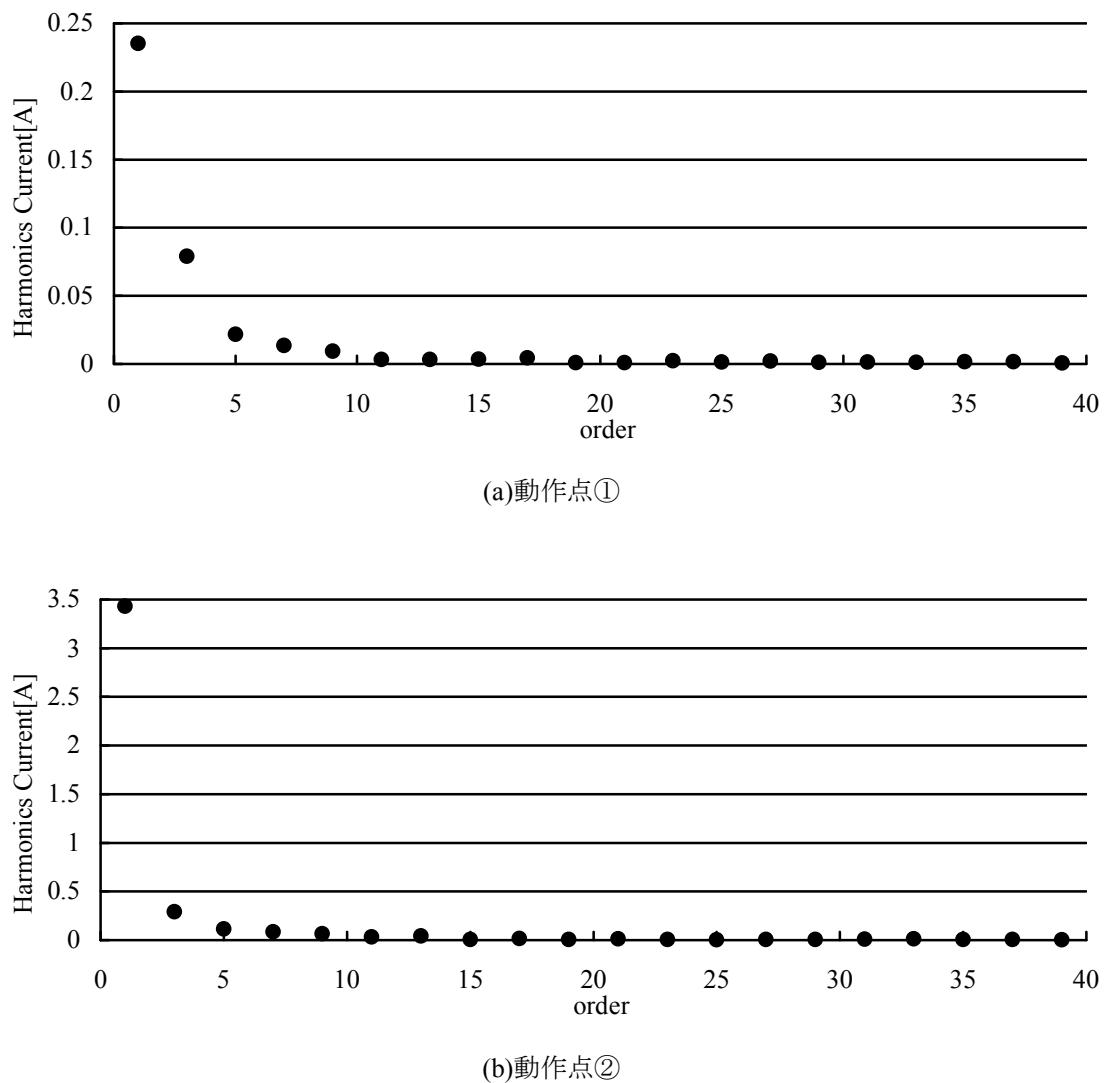


図 5.41 アナログ力率改善制御器の各動作点の高調波電流特性

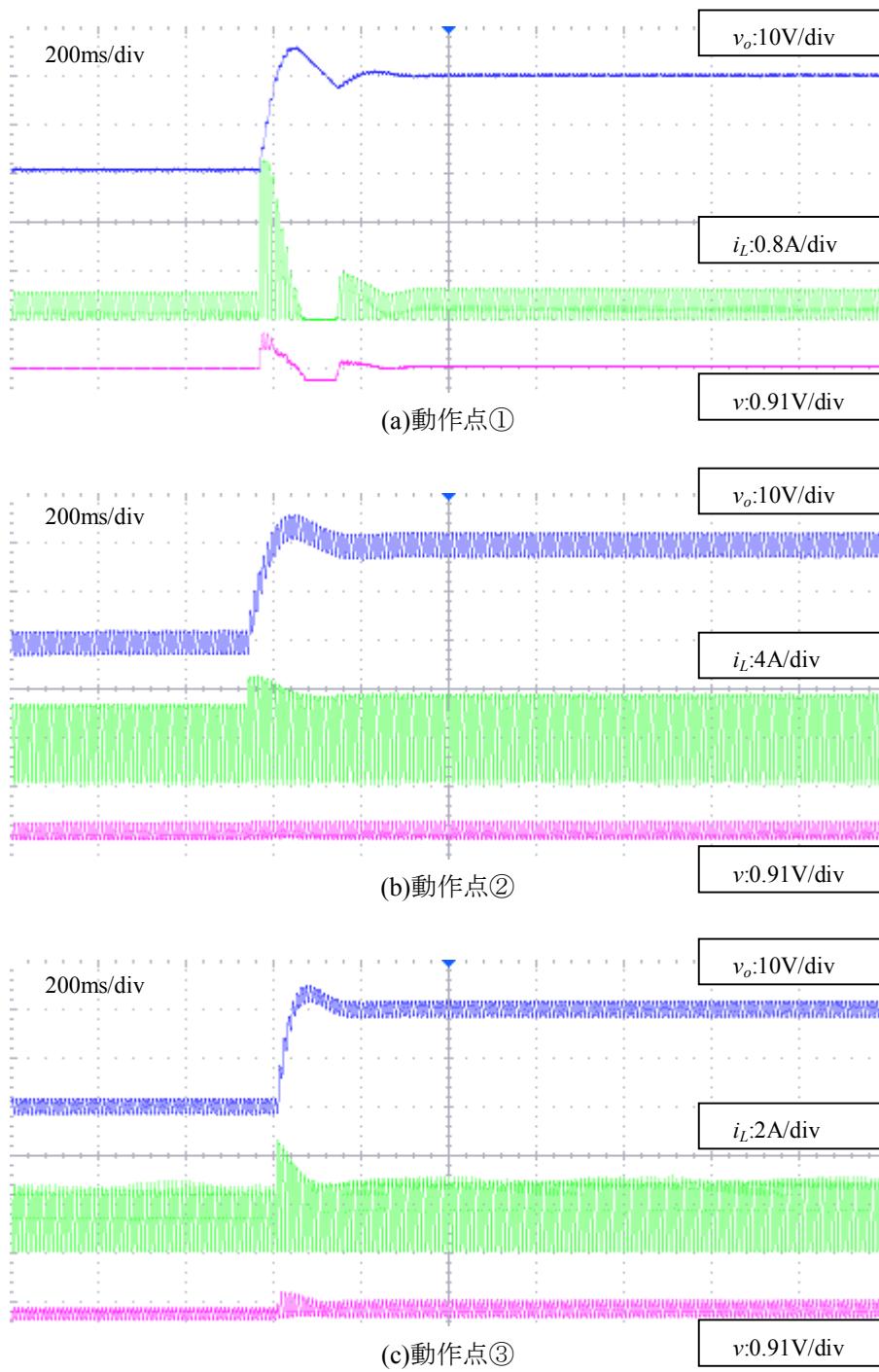


図 5.42 ディジタル PI 力率改善制御器の各動作点のステップ応答結果

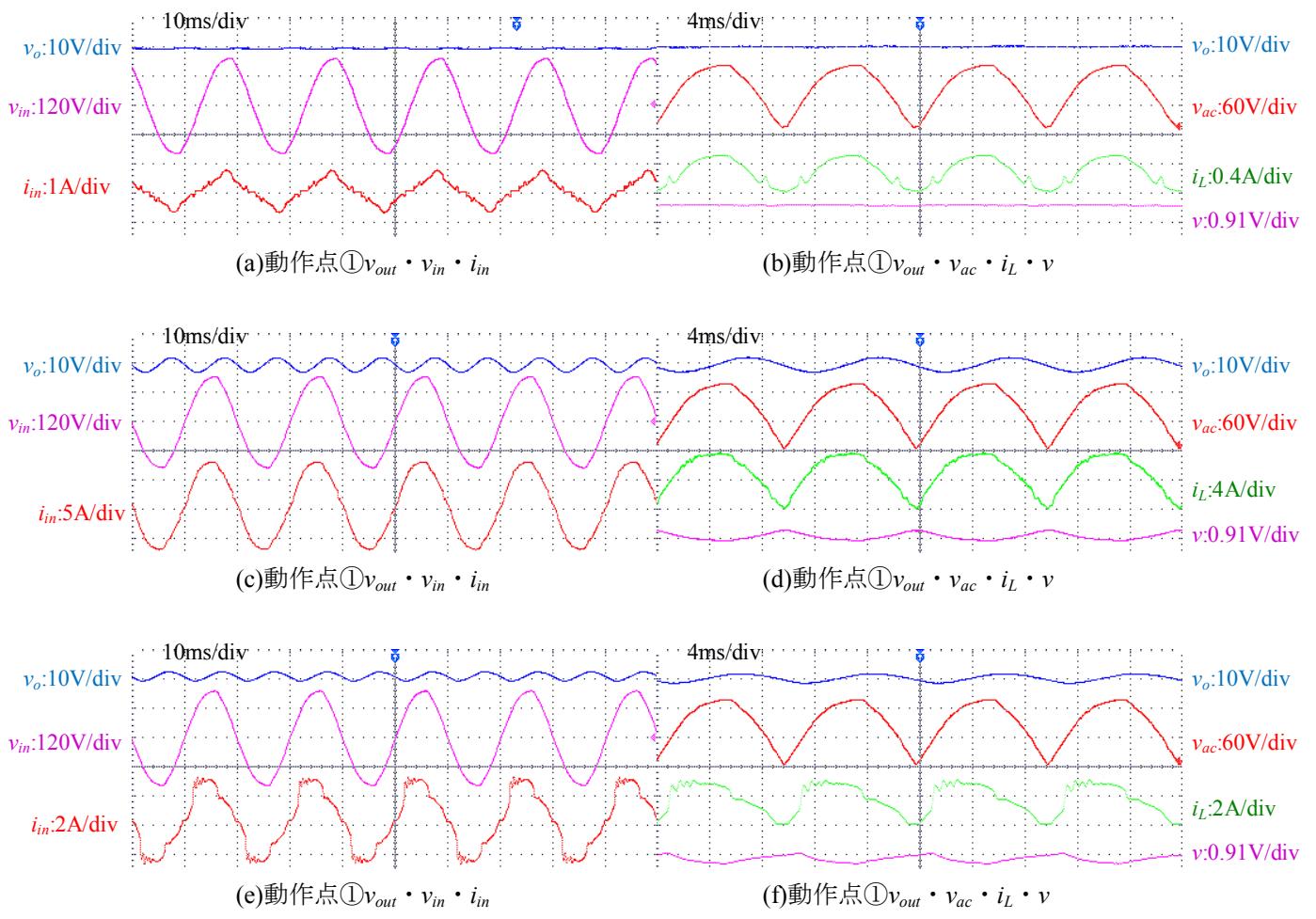
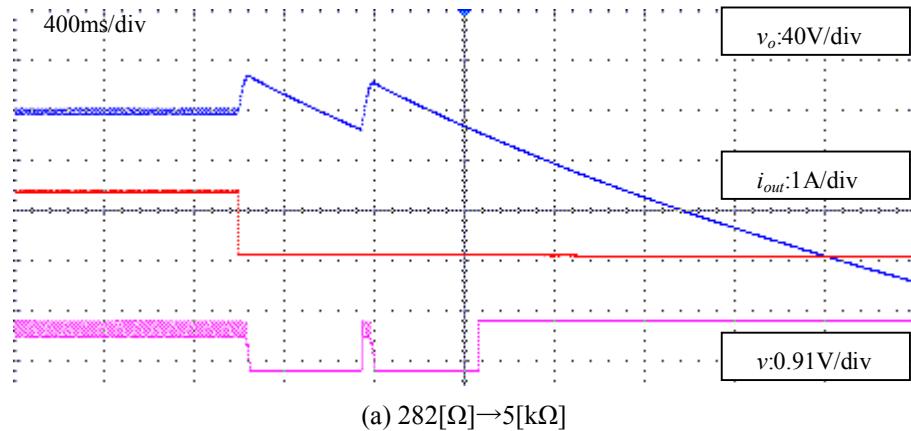


図 5.43 ディジタル PI 力率改善制御器の各動作点の定常状態波形



(a)  $282[\Omega] \rightarrow 5[k\Omega]$

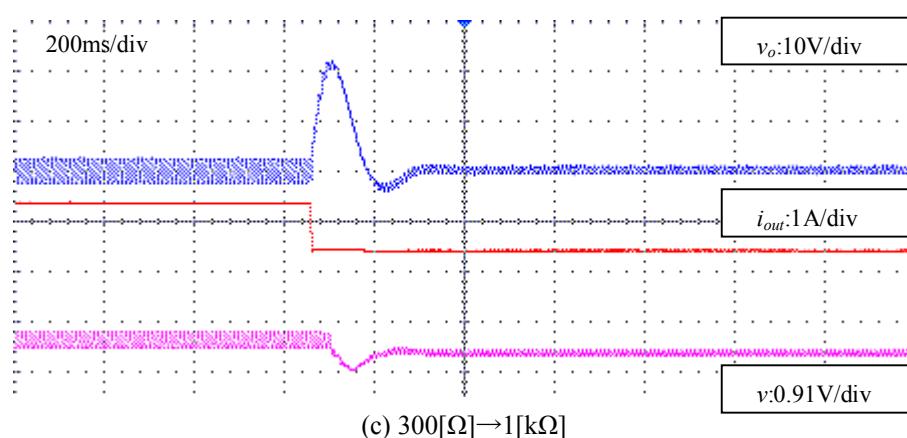
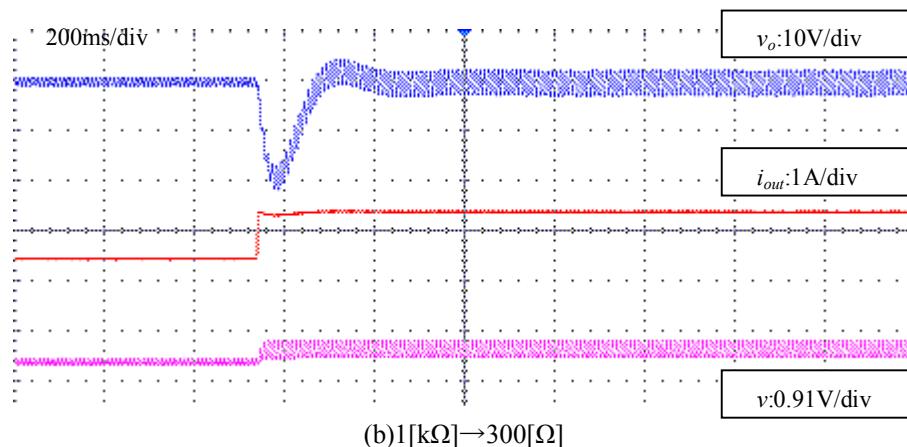


図 5.44 ディジタル PI 力率改善制御器の負荷急変応答結果

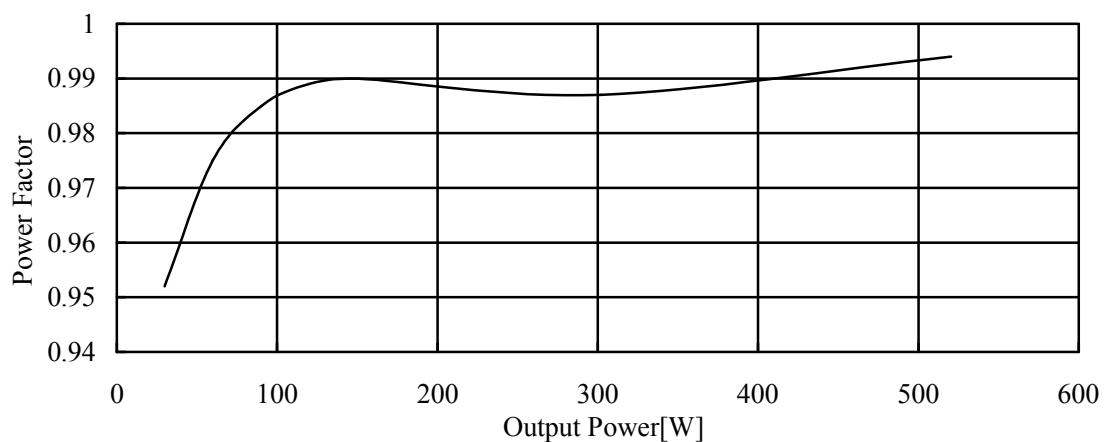
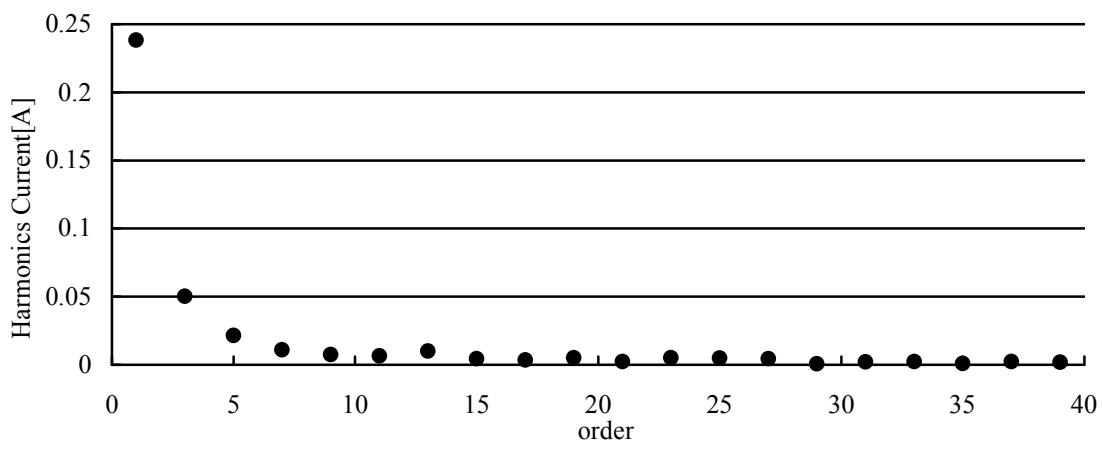
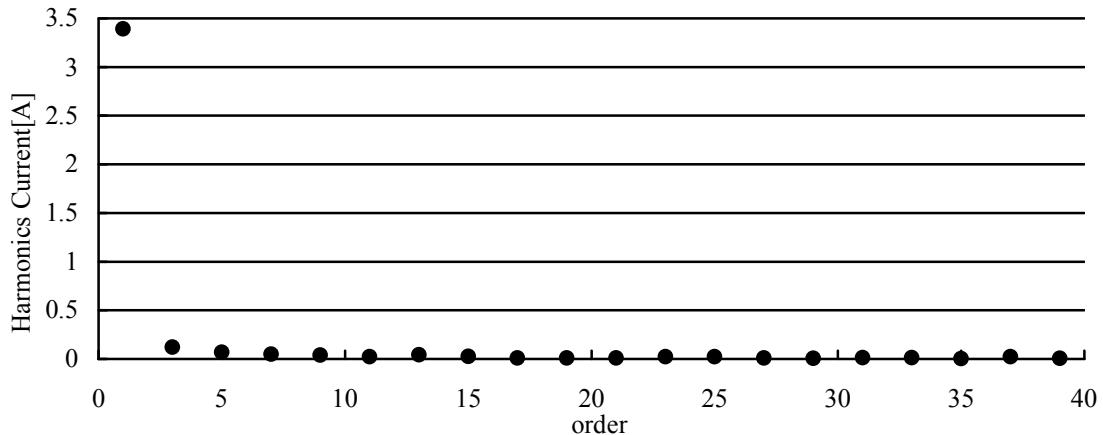


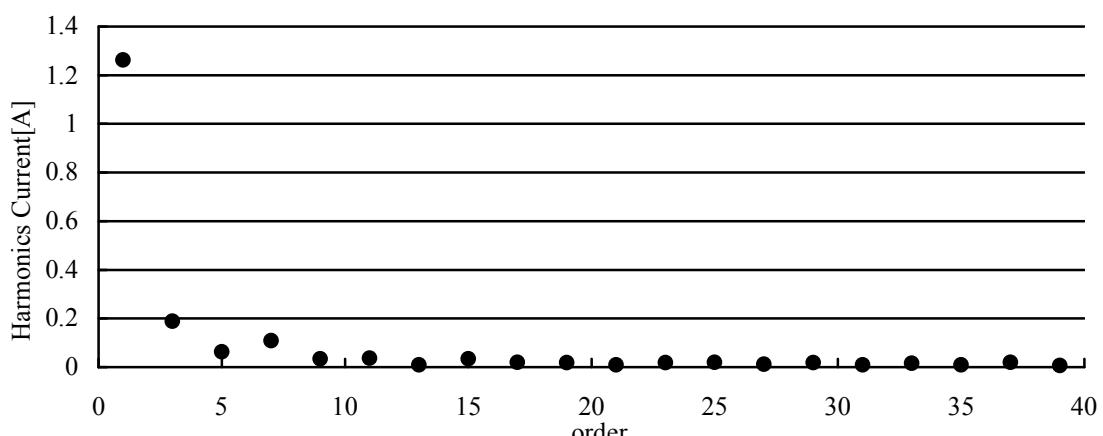
図 5.45 ディジタル PI 力率改善制御器の力率特性



(a)動作点①



(b)動作点②



(c)動作点③

図 5.46 ディジタル PI 力率改善制御器の各動作点の高調波電流特性

## 5.8 入力電圧の変動

### 5.8.1 入力電圧変動の影響

今までの結果は、入力電圧が AC100[V]から変わらないものとして、シミュレーションと実験を行ってきた。しかしながら、入力の家庭用交流電源等の電圧は、周囲の機器や使用される地域の電力系統の状況により、変動する場合がある。そこで、入力電圧が変動した場合の、各状態への影響と、 $1/v_{avg}^2$  の項による、入力電圧変動に対する補償の効果について、考察する。

まず、力率改善回路の入力電力の平均値  $P_{inavg}$  は、入力電圧の平均値  $v_{avg} \times$  入力電流の平均値  $i_{avg}$  で、さらにそれぞれの全波整流後の平均値の積に等しいから、

$$P_{inavg} = v_{avg} \times i_{avg} = v_{ac} \times I_s \quad (5.32)$$

である。

ここで、負荷  $R_L$  とコンバータの損失  $\eta$  は一定であるときに、入力電圧  $v_i$  の振幅  $V_i$  が変動した場合、式(2.5), (2.6)の昇圧コンバータの静特性方程式から、インダクタ電流、出力電圧両方にも、 $V_i$  に比例した変動が生じる。出力電圧を一定に保ちたいとすると、出力電力  $P_{outavg}=V_s^2/R_L$  を一定にするようにする必要がある。これは、入力電圧  $v_i$  の振幅が変化した場合、入出力電力の関係  $P_{outavg}=\eta \times P_{inavg}$  から、入力電力  $P_{inavg}$  を一定に保つ必要があることを意味する。入力電圧  $v_i$  の振幅が  $\alpha$  倍されたとすると、 $v_{ac}$  の平均値  $v_{avg}$  も同じ比率で  $\alpha$  倍される。このとき、入力電力  $P_{inavg}$  が一定であるためには、以下の式から  $I_s$  が  $1/\alpha$  倍になればよい。

$$P_{inavg} = v_{ac} \times I_s = \alpha v_{ac} \times \frac{I_s}{\alpha} \quad (5.33)$$

さて、PFC 制御器の乗算器に、 $1/v_{avg}^2$  を掛けていると、乗算器の出力=インダクタ電流の目標値の平均値は、

$$v_{PL} \times v_{ac} \times \frac{1}{v_{avg}^2} = \frac{v_{PL}}{v_{ac}} \quad (5.34)$$

となる。 $v_{ac}$  の平均値が  $\alpha$  倍変化した場合では、

$$v_{PL} \times \alpha v_{ac} \times \frac{1}{(\alpha v_{avg})^2} = \frac{v_{PL}}{\alpha v_{ac}} \quad (5.35)$$

となって、変動がない場合に対し、インダクタ電流が  $1/\alpha$  倍となるため、入力電力は等しくなり、出力電圧変動の抑制につながる。

### 5.8.2 制御器への補正

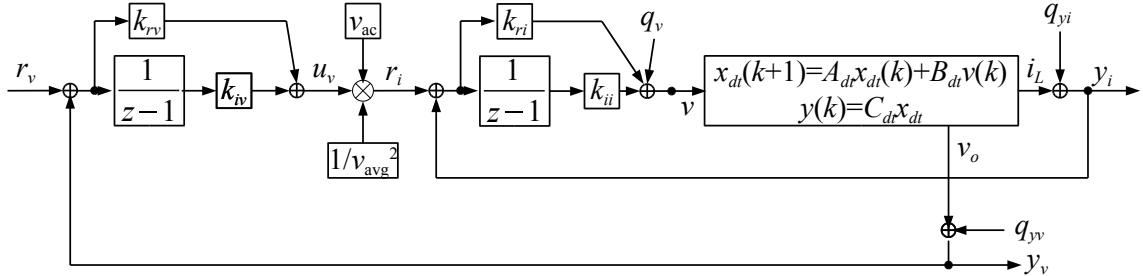


図 5.47 ディジタル PI 力率改善制御器の構成

PI 制御の場合、電圧制御器、電流制御器は、目標値と測定値の誤差信号から、制御を行う。電流制御器の目標値は、電圧制御器の操作量であるが、時定数の長い電圧制御器の操作量は、入力電圧変動に対し応答が遅い。そのため、入力電圧が変動すると、インダクタ電流も大きく変動し、その後結果的に出力電圧変動が大きく生じてしまう。そこで、この  $1/v_{avg}^2$  をかけることで、電圧制御器の操作量にフィードフォワードのような補償を働かせて、インダクタ電流の補正、入力電圧の補正を行うことが必須である。

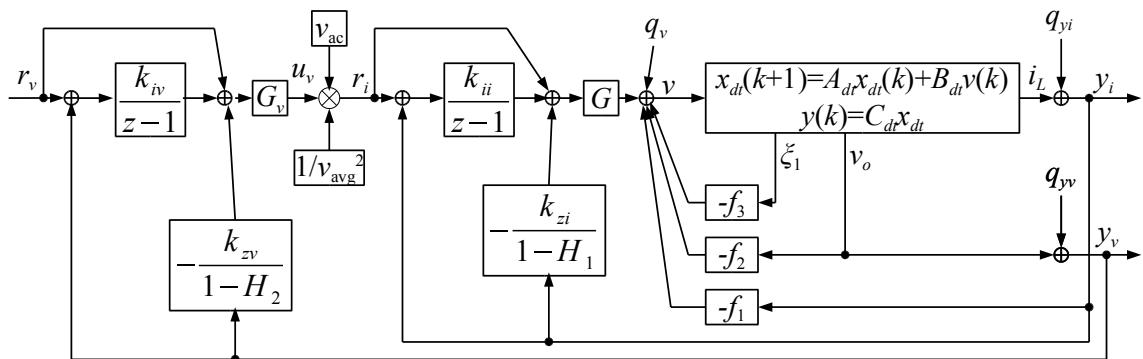


図 5.48 近似的 2 自由度力率改善制御器の構成

近似的 2 自由度制御の場合、エラーアンプの誤差信号だけでなく、常に状態 FB 信号が電圧・電流制御器それぞれの操作量に直接入っており、電流制御器の目標値が変わらなくても、電流制御器の操作量が変化し、インダクタ電流の変動が抑えられる。

かつ、近似的 2 自由度制御は、外乱に強い特性であり、入力電圧の振幅変動は、実質パラメータ変動による外乱であるため、それ単体でも振幅変動の影響を抑えることができ、 $1/v_{avg}^2$  は必ずしも必要ではないと考えられる。

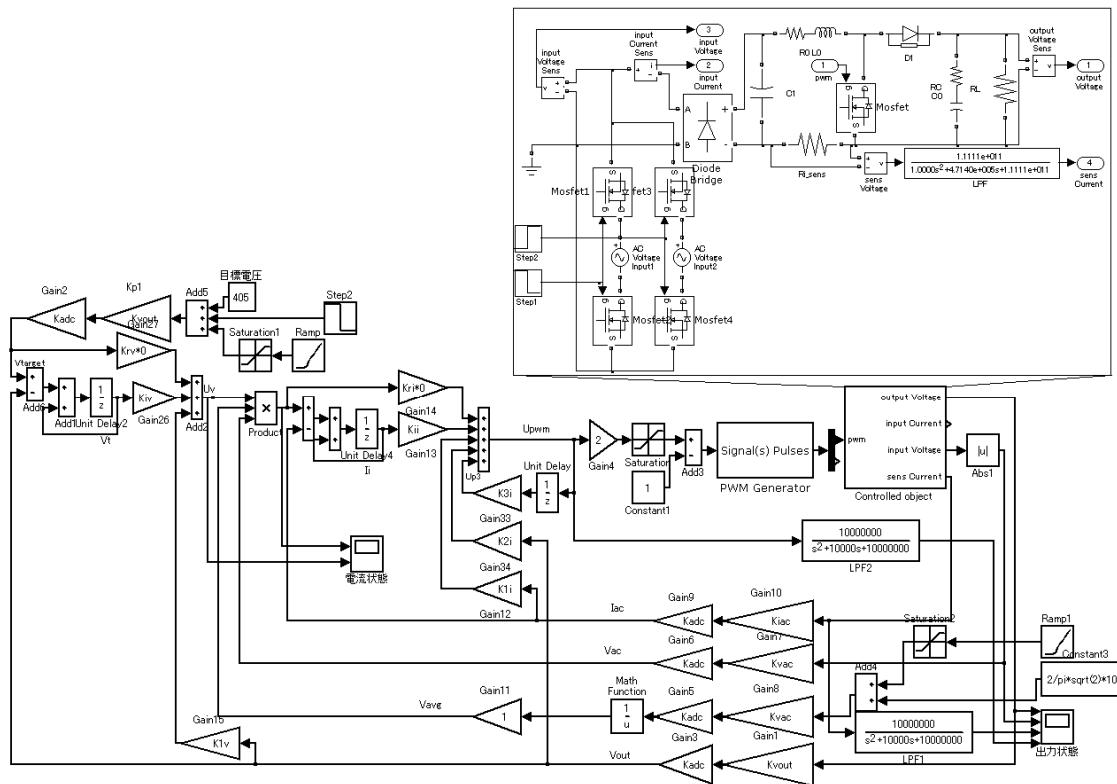
### 5.8.3 入力電圧急変シミュレーション

入力電圧が変動した場合の、出力電圧等への影響を検証するために、以下の図 5.49 の SimPowerSystems のモデルを使って、入力電圧が AC100[V]から AC220[V]へ変化した場合の  $1/v_{avg}^2$  の有無による、出力電圧変動への影響をシミュレーションする。

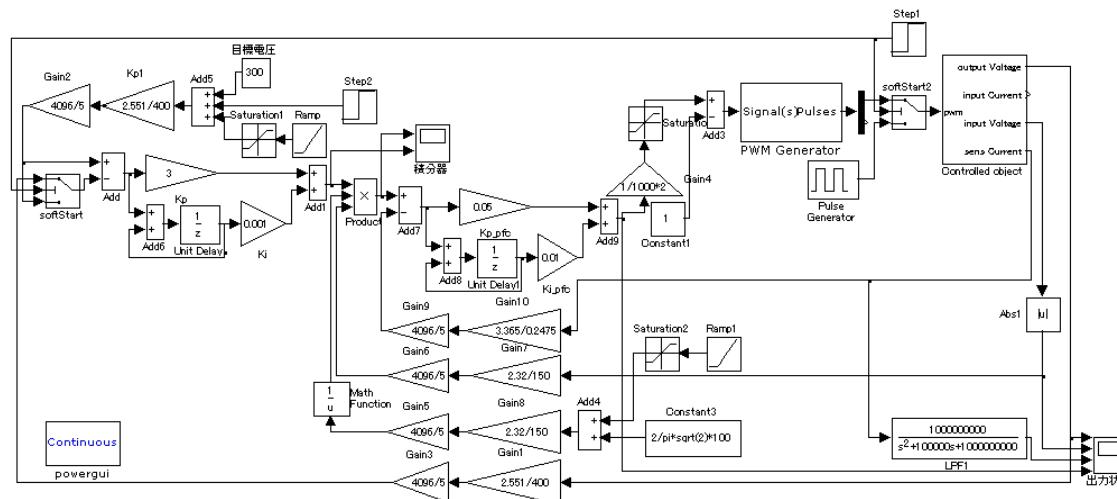
コンバータの電源を立ち上げて、出力電圧が 385[V]に到達してから、入力電圧を変動させる。 $v_{avg}$  は入力電圧の過去 1 周期の平均を計算するように設定した。

この SimPowerSystems モデルを使ってシミュレーションした時の、近似的 2 自由度制御の  $v_{avg}$  補償無しの結果を図 5.50 に、 $v_{avg}$  補償有りの結果を図 5.51 にそれぞれ示す。近似的 2 自由度制御の場合、入力電圧が変動したとき、 $v_{avg}$  の有無に関わらず、約 20[V]程度の出力電圧変動が見られ、 $v_{avg}$  がない場合とある場合では、大きな差は見られなかった。

続いて、デジタル PI 制御の場合の、シミュレーションの、 $v_{avg}$  補償無しの結果を図 5.52 に、 $v_{avg}$  補償有りの結果を図 5.53 に示す。デジタル PI 制御の場合、 $v_{avg}$  の補償があるときは、約 20[V]の出力電圧変動であったが、 $v_{avg}$  補償がない場合、約 40[V]と大きく変動してしまっていることがわかる。ここから、近似的 2 自由度制御では、 $v_{avg}$  補償が無くとも、そのロバスト性により、入力電圧急変のような外乱に対しても、出力電圧変動を抑えられるが、デジタル PI 制御などの 1 自由度制御では、 $v_{avg}$  の補償を行わないと、大幅に出力電圧変動が増えてしまうことがわかる。

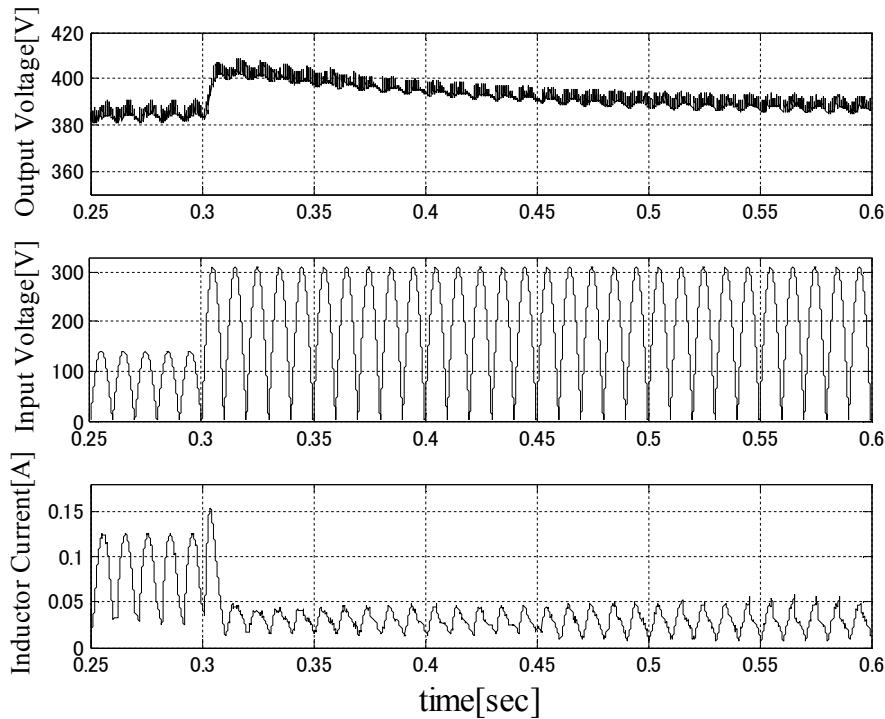


(a)近似的2自由度力率改善制御器

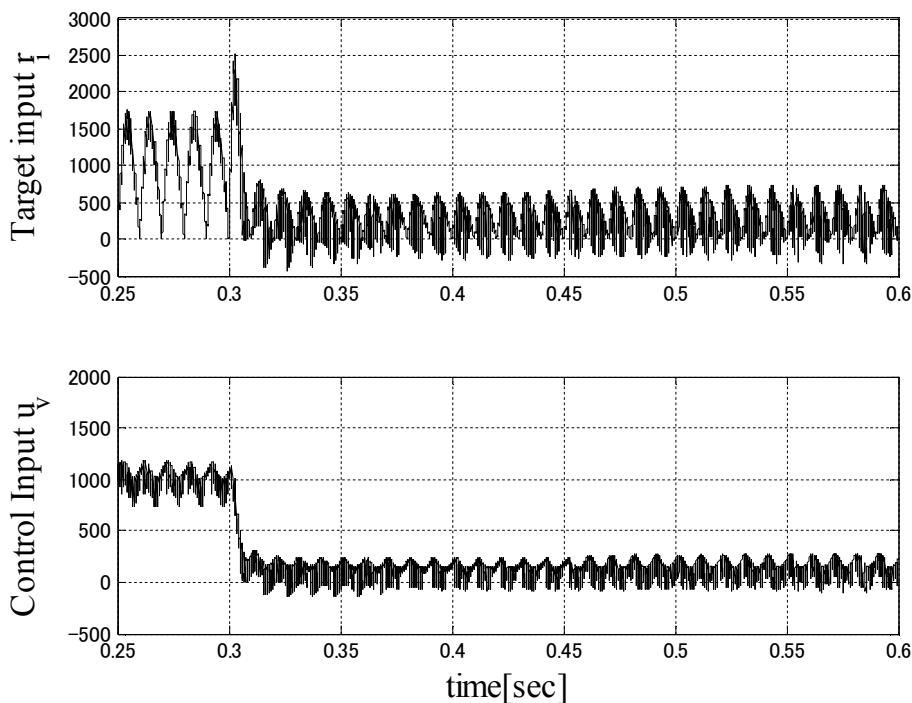


(b)デジタルPI力率改善制御器(サブシステムの中は(a)と同一)

図 5.49 力率改善制御器の入力電圧急変シミュレーションを行う SimPowerSystems モデル

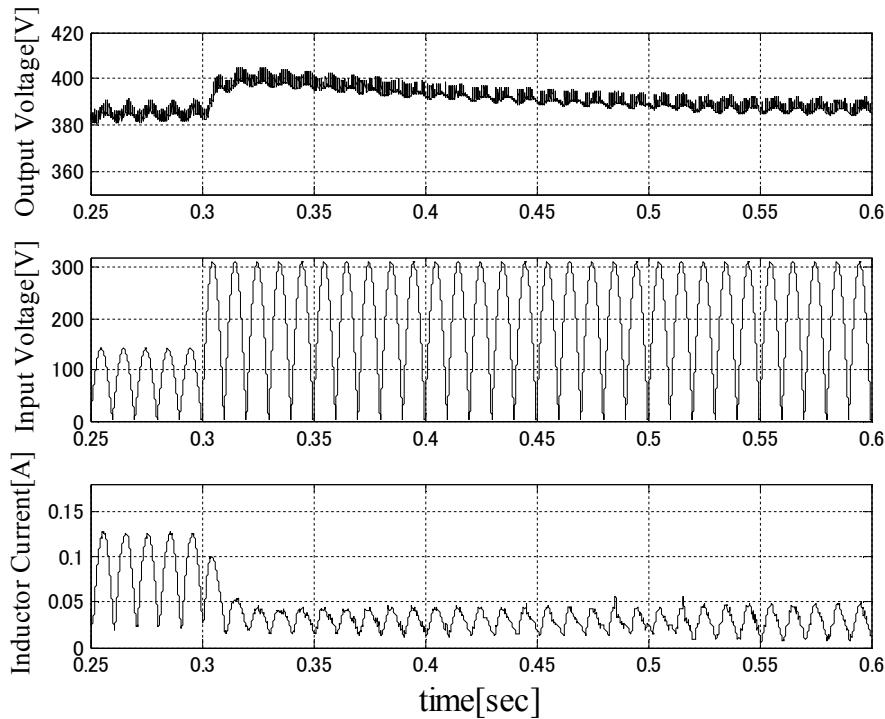


(a)出力電圧, 入力電圧, インダクタ電流

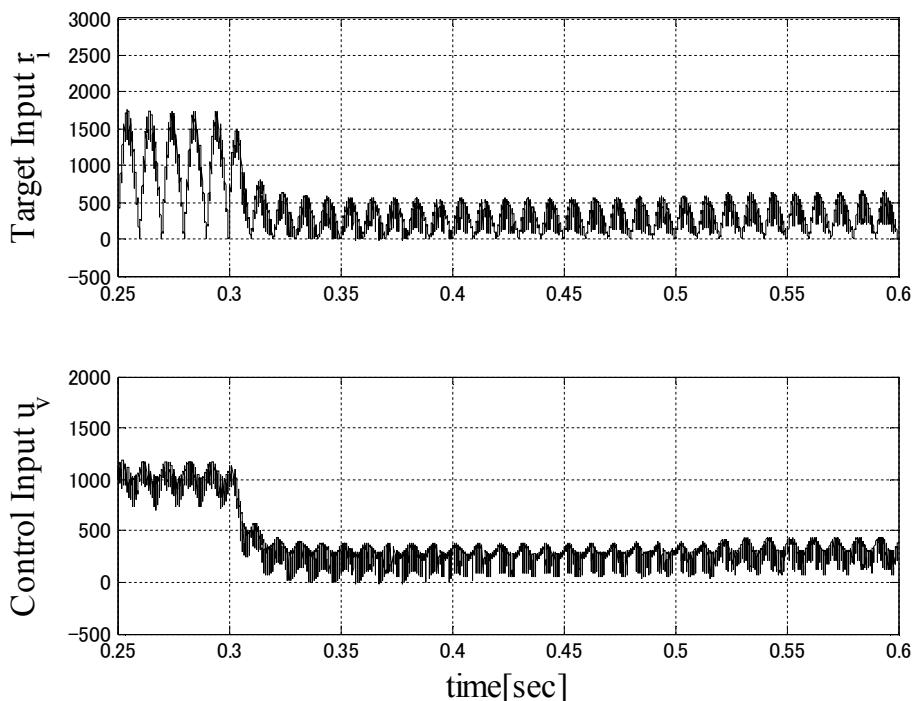


(b)電流目標値と出力電圧制御器の操作量

図 5.50 近似的 2 自由度力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$ 無し)

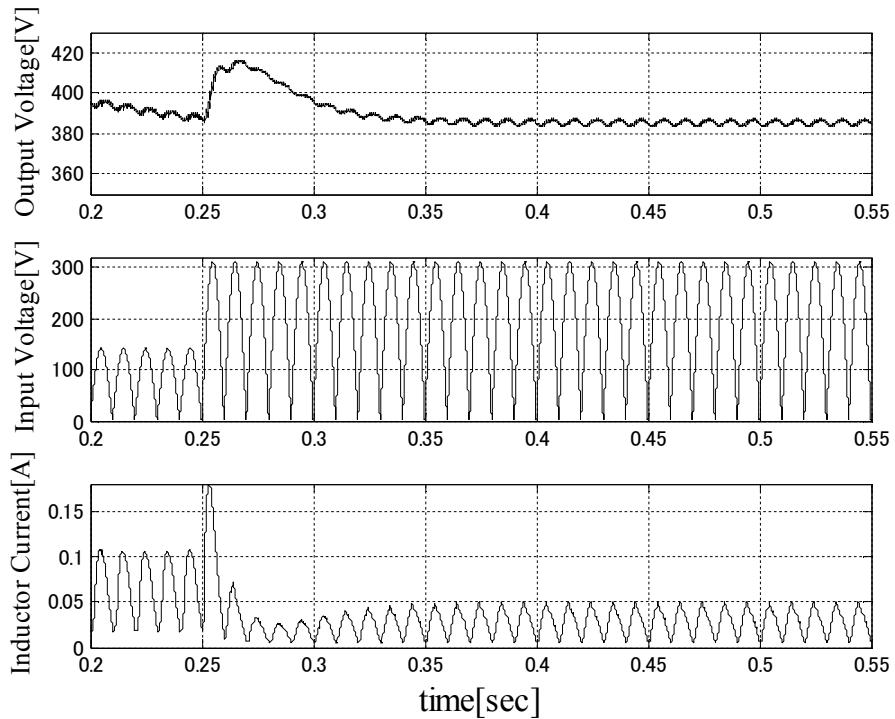


(a)出力電圧, 入力電圧, インダクタ電流

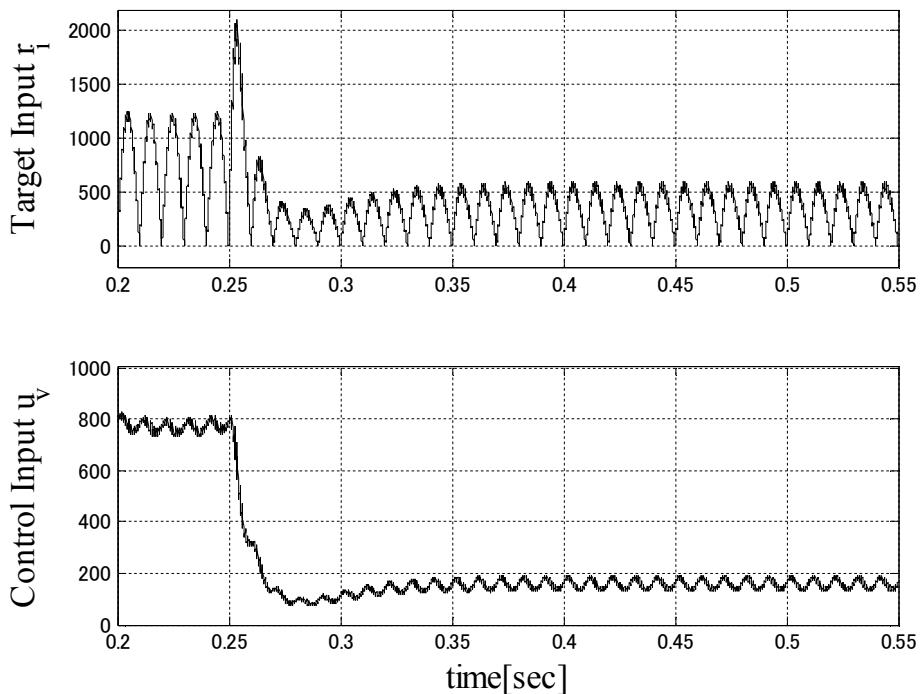


(b)電流目標値と出力電圧制御器の操作量

図 5.51 近似的 2 自由度力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$  有り)

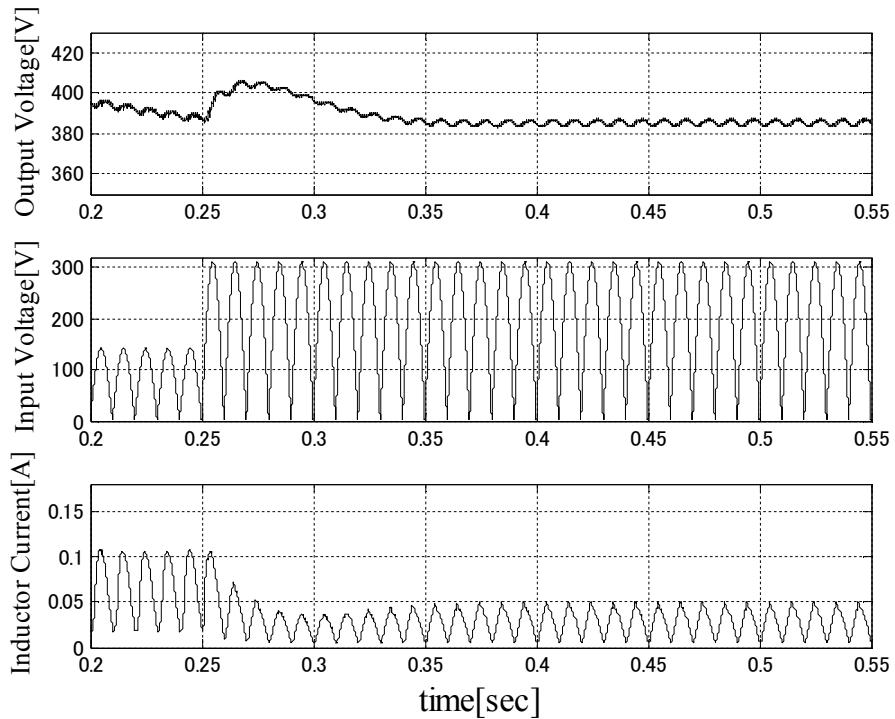


(a)出力電圧, 入力電圧, インダクタ電流

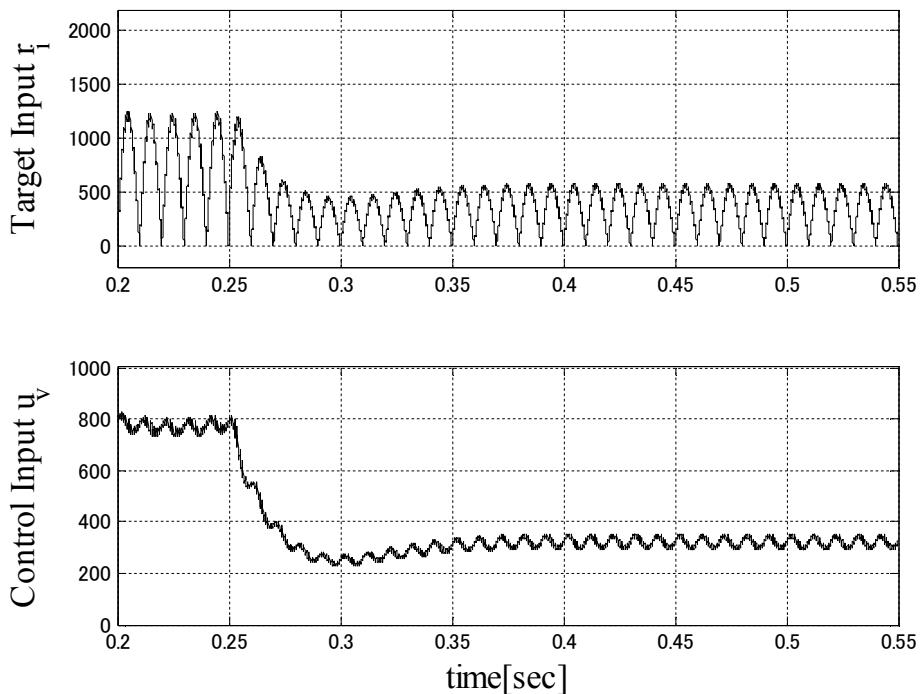


(b)電流目標値と出力電圧制御器の操作量

図 5.52 ディジタル PI 力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$ 無し)



(a)出力電圧, 入力電圧, インダクタ電流



(b)電流目標値と出力電圧制御器の操作量

図 5.53 ディジタル PI 力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$  有り)

# 第6章 結論

本研究では、電流導通モードのPFC昇圧コンバータを制御対象とし、インダクタ電流制御器と出力電圧制御器の2重ループで制御することで、力率を改善し、高調波を十分低減しながら、負荷の変動に対して出力電圧の変動を十分小さく抑え、簡易に設計できる、ロバストな近似的2自由度デジタル積分形制御器の設計法を提案した。

本論文においては、マイクロコントローラ(SH7216)を使用したデジタル制御PFC昇圧型コンバータシステムを設計・製作し、その制御に取り組んだ。

第2章では、力率改善回路の基本構成と昇圧型コンバータの動作について、また、モデルリングと、作成したモデルの離散化について述べた。昇圧型コンバータは非線形システムであり、負荷などの変化に対し、静特性と動特性が大きく変化する。そのため、その制御には、仕様の範囲で動作点を定め、全ての動作点における特性を考慮する必要がある。また、デジタル制御の際には、A/D変換からPWMのデューティ比を更新するまでの遅れが生じるため、それを考慮する必要がある。

第3章では、マイクロコントローラの動作と、パワーパートの構成について述べた。実際のマイクロコントローラの1周期ごとに行う割り込みルーチンの構成をし、遅れ時間を $0.99T_s$ に定めた。また、パワーパートの実際の回路と、マイクロコントローラとの接続をするための回路を製作した。

第4章では、力率改善はせず、出力電圧のみを制御することを目的とし、非線形システムの動作点変化を考慮した、近似的2自由度制御器の設計法と、パラメータの決定法、シミュレーションと実験結果について述べた。非線形システムを制御するにあたり、仕様から、動作点3つを定めた。制御器の設計には、公称モデルとなる動作点を1つ選択し、それに対して制御器を設計し、かつ他の動作点における特性を考慮しながら、パラメータを導出する手法を述べた。導出されたパラメータを用いた制御器を実装して、昇圧型コンバータのステップ応答と負荷急変時の出力電圧応答を、シミュレーションと実験によって検証した。シミュレーション結果と実験結果はよく一致し、動作点の変動に対し十分ロバストで、与えられた仕様を満足した。デジタルPI制御を用いた場合、動作点の変化にロバストではなく、仕様を満たすことが出来なかった。

第5章では、力率改善制御器の一般的な構成法と、そこに近似的2自由度制御を適用する際の問題点を挙げ、それを解決する、力率改善近似的2自由度制御器の設計法と、パラメータの決定法、シミュレーションと実験結果について述べた。近似的2自由度制御を力率改善制御に直接適用すると、電圧制御器が複雑になり設計が困難になる。そこで、電流制御器の設計時に条件をつけることで、電流制御ループを1次モデルに近似し、電圧制御器の設計を簡易にした。4章と同様に、公称モデルとなる動作点に対して制御器を設計し、他の動作点における特性を考慮しながら、かつ、電圧制御器が電流制御器に影響を与えない

いよう、パラメータを決定した。決定パラメータを用いた制御器を実装して、力率改善昇圧型コンバータの立ち上がり波形、ステップ応答、負荷急変応答、力率特性、高調波特性を、シミュレーションと実験によって検証した。シミュレーション結果と実験結果はよく一致し、提案した制御器は、動作点の変動に対し十分ロバストで、力率を改善しながら、高調波を大きく除去することができ、与えられた仕様を満足した。アナログ IC を用いた場合、負荷急変応答は良好であったが、電流波形に歪を生じ、高調波特性が悪く、高性能化が望めない。デジタル PI 制御を用いた場合、動作点の変化にロバストではなく、仕様を満たすことが出来なかった。従来手法との比較から、今回提案した制御器のほうがより高調波電流を低減でき、ロバストであると言え、提案法の有用性と有効性が示された。また、シミュレーションにより、提案した近似的 2 自由度制御器は、入力電圧の変動に対しても、補償を行わなくともロバストであることが示された。

今後の課題は、今回行えなかった、 $1/v_{avg}^2$  の有無による入力 AC 電圧の変動に対するロバスト性の影響の実機実験による検証を行い、またそれを考慮して、制御アルゴリズムやパラメータを改善することや、さらなる高調波電流除去のために、入力電圧波形をソフトウェアで生成すること、広範囲動作をさせるため、制御器やパラメータを動的に切り替えること、などが挙げられる。近年では、大出力化と低ノイズ化のために、パワー部を多相化してインターリーブ動作させることも活発に研究がなされており、インターリーブ動作をする力率改善コンバータにも、提案手法を適用することが望まれる。

また、今回は、昇圧型コンバータの出力の電解コンデンサの等価直列抵抗(ESR)は、制御系には影響がないものとして無視して設計したが、実際に、電解コンデンサは大きな ESR を持つため、それも考慮して、制御器設計を行うことも課題である。

# 謝辞

最後に、本研究を行うにあたり、多大なるご指導、ご鞭撻を頂いた樋口幸治准教授、中野和司教授、梶川竜義技官、並びに、ルネサスソリューションズ株式会社の松島正さん、鈴木元章さんに深く感謝致します。また、共に研究を行った、鶴堀君、山岸さん、深石君、西村君、神田君、鈴木君、足達君、野谷君、近藤君、Savio さんにも感謝致します。

# 参考文献

- [1] 竹上栄治, "DSP を用いた DC-DC コンバータのロバスト制御に関する研究", 電気通信大学大学院博士論文, 2008
- [2] 竹上栄治, 橋口幸治, 中野和司, 富岡聰, 渡辺一史, "二次モデル実現近似的 2 自由度ディジタル積分形制御器による DCDC コンバータのロバスト制御", 電子情報通信学会論文誌, Vol.J88-C, No.9, pp.724-736, 2005
- [3] 橋口幸治, 中野和司, 荒木邦彌, 茅野文穂, "電圧フィードバックのみを用いた近似的 2 自由度積分形制御によるロバスト PWM 電力増幅器の設計", 電子情報通信学会論文誌, Vol.J85-C, No.10, pp.906-916, 2002
- [4] Koji Higuchi, Eiji Takegami, Kazushi Nakano, Satoshi Tomioka and Kazushi Watanabe, "New Design and Manufacture of Robust DC-DC Converter Using Approximate 2DOF Digital Controller", ICIT 2005, pp.840-845, 2005
- [5] 佐々木清吾, 渡辺晴夫, "スイッチング電源の動的制御のための多動作点解析", 信学技報, pp.33-38, 2005
- [6] M. Xie, "Digital Control For Power Factor Correction", Unpublished master's thesis, Virginia Polytechnic Institute and State University, 2003.
- [7] 松下宏治, "トランジスタ技術 2009 年 9 月号 ディジタル制御 PFC 設計初めの一歩", pp145-153, 2009
- [8] 河村篤男, "現代パワーエレクトロニクス", サイエンス社, 2005
- [9] "SH7216 グループハードウェアマニュアル", ルネサスエレクトロニクス, 2009
- [10] "SH7216CPU ボード R0K572167C001BR ユーザーズマニュアル", ルネサスエレクトロニクス, 2009
- [11] "HA16174 EVB PFC コントロール IC HA16174 評価ボード アプリケーションノート", ルネサスエレクトロニクス, 2007
- [12] "HA16174P/FP データシート", ルネサスエレクトロニクス, 2005
- [13] 野波健蔵, 西村秀和, "MATLAB による制御理論の基礎", 東京電機大学出版局, 1998
- [14] 足立修一, "MATLABによる制御工学", 東京電機大学出版局, 1999
- [15] 相良節夫, 和田清, 中野和司, "デジタル制御の基礎", コロナ社, 1992
- [16] 木村英紀, "デジタル信号処理と制御", 昭晃堂, 1982
- [17] 本田潤, "D 級/デジタルアンプの設計と製作", CQ 出版社, 2004
- [18] 兼田雅弘, 山本幸一郎, "デジタル制御工学", 共立出版, 1989
- [19] A. Kugi and K. Schlacher, "Nonlinear  $H^\infty$ -Controller Design for a DC-to-DC Power Converter", IEEE transactions on control systems technology, vol.7, No.2, pp.230-237, 1999.

- [20] L. Rossetto, G. Spiazzi, and P. Tenti, “Control Techniques for Power Factor Correction Converters”, Proc. PEMC’94, pp.1310-1318, 1994
- [21] M. Fu and Q. Chen, “A DSP Based Controller for Power Factor Correction (PFC) in a Rectifier Circuit”, IEEE APEC 2001, pp.144-149, 2001
- [22] E. Figueres, J. M. Benavent, G. Garcera and M. Pascual, “A Control Circuit With Load-Current Injection for Single-Phase Power-Factor-Correction Rectifiers”, IEEE Transactions on Industrial Electronics, Vol. 54, No. 3, pp.1272-1281, 2007

## 本研究で使用した実験装置

メーカー	名称	機器名	Serial No.
[1]NF 回路設計ブロック	プログラマブル AC/DC 電源	EC1000S	9103558
[2]ADVANTEST	デジタルマルチメータ	R6450	16141589
[3]HIOKI	AC/DC パワーハイテスタ	3334	091017711
[4]TAKASAGO	直流安定化電源	TPM O16-2	971401
[5]TAKASAGO	直流安定化電源	TPM O16-2	971402
[6]Tektronix	Mixed Signal Oscilloscope	MSO2014	C000480
[7]Agilent	電流プローブ用電源	N2775A	1100057
[8]Agilent	電流プローブ	N2774A	1200031
[9]Renesas Electronics	マイコンエミュレータ E10A	HS0005KCU01H	04194C

## 本研究で使用したソフトウェア

- [1]MATLAB Version 7.8.0.347 (R2009a) 32bit,  
 SIMULINK Version 7.3 (R2009a),  
 SimPowerSystems Toolbox Version 5.1
- [2]High-performance Embedded Workshop Version 4.08.00.011

# 付録 A MATLAB プログラム

## A.1 出力電圧のみ制御する昇圧型コンバータの MATLAB プログラム

ム

```
format short e; %小数点表示桁指定
format compact; %余計な改行を省く
clc; %コマンドウインドウのクリア
clear all; %ワークスペースのクリア
close all; %図をすべて閉じる
%%%%%%%%%%%%% COMMENT %%%%%%%%%%%%%%
% boost_3point_2DOFcontrol_characteristics_noload.m %
% Renesas PFCIC HA16174評価ボードの昇圧コンバータ %
% 無負荷時や5kΩ負荷時を考慮した時の2DOF制御時の特性 %
% copyright (c) Yoshihiro Ota, Kohji Higuchi %
% The University of Electro-Communications %
%%%%%%%%%%%%% %%%%%%%%%%%%%%
%STEP.1%%%%%_SAMPLING FREQUENCY_%%%%%%%%%%%%%%
fs=100e3; %[Hz] e3=(*100)
Ts = 1/fs; %sampling frequency [Hz]→ time[sec]
%STEP.2%%%%%_CHANGE PARAMETER_%%%%%%%%%%%%%%
No = 13; %
H1 = -0.47 + 0.01i; %
H2 = -0.99973;
H3 = -0.47 - 0.01i; %
H4 = 0.1; %
Kz = 0.3;
%注.H1H3を左半平面に持ってくと発振する
%OPTION.0%%%%%_SIMULATION PARAM_%%%%%%%%%%%%%%
simTime = 1;
changeTime = 0.5;
%OPTION.1%%%%%_CONTROLED OBJECT PARAM_%%%%%%%%%%%%%%
Tdelay = Ts ; %delay time
L = 0.99*Ts ; %delay time
Vin = 100; %入力電圧AC100V
Vin_peak = Vin*sqrt(2); %入力のピーク電圧
fvin = 50; %入力AC周波数
Vout = [500,385,240]; %出力電圧目標値
%Vout = [240,509,385]; %出力電圧目標値
%Mu = 0.68 %目標出力電圧でのデューティ
L0 = 150e-6; %インダクタンス
C0 = (470e-6)*2; %出力キャパシタ
R0 = 1.8; %インダクタの等価直列抵抗
RC = 0.3; %キャパシタの等価直列抵抗
C1 = 1e-3; %入力キャパシタ
Risence = 0.033/2; %電流検出抵抗
RL0 = 292; %負荷抵抗
%RL0 = 500; %負荷抵抗
RL1 = 5000;
COL =(8000e-6)*2;
LOL = 150e-6; %インダクタンス
%2
RL1 = 5000;
COL =(12000e-6)*2;
LOL = 200e-6; %インダクタンス
RL_DL0 = 500; %負荷急変用の抵抗
RL_DL1 = 650; %負荷急変用の抵抗
```

```

Kvac = 2.3125/150;      %入力電圧分圧ゲイン
Kiac = 3000/200;        %入力電流検出ゲイン
Kadc = 4096/5;          %ADCゲイン {2^(12)/5[V]}
Kvout= 5.08/Kadc;
%Kvout= 2.551/400;      %出力電圧分圧ゲイン
cpuClock    = 200e6;      %SH7216 200[MHz]
mtuClock   = cpuClock/2;  %MTU2SのクロックPΦ
mtuTime     = 1/mtuClock; %MTUの1ステップ時間
step        = round(Ts/mtuTime);%ステップ数
Kpwm = 1/step;
h=1;      %電圧フィードバック係数
%CALC_0%%%%%_STATE EQUATION_%%%%%%%%%%%%%%%
%StaticCaracteristy平均出力電圧の計算とプロット
for i= 1:999
Mu = i/1000;
%Vs(i) = 1 / (1+R0/(RL0*(1-Mu)^2)) * Mu * Vin_peak /(1-Mu) + Vin_peak;
%Vs(i) = ( Vin_peak*RL0*(1-Mu)) / ( R0+RL0*((1-Mu)^2) );
Vs(i) = Vin_peak* ((RL0*(1-Mu))/(RL0*(1-Mu)^2 + R0));
%Vs(i) = Vin_peak* ((RL0*(1-Mu))/(RL0*(1-Mu)^2 + R0))*(1/exp(RL0/1e5)+Mu)*exp(RL0/1e5);
Is(i) = 1 / (1-Mu) / RL0 * Vs(i);
end
figure(100)
for i=1:99
    Vss(i)=Vs(round(i*10));
end
plot(Vss,'k-');
%title('デューティ 対出力電圧特性','FontSize',14);
xlabel('Average of duty ratio[%]','FontName','Times New Roman','FontSize',14);
ylabel('Average of output voltage[V]','FontName','Times New Roman','FontSize',14);
%axis([0,81.6,0,700]);
axis([0,100.6,0,950]);
%3つの動作点におけるデューティを計算
for i= 2:99
    for j= 1:3
        if(Vs(i)>Vout(j) && Vs(i-1)<=Vout(j))
            Mu(j) = i/1000;
        end
    end
end
%それぞれのデューティでIsとVsを求める
%Vs = 1 / (1+R0/(RL0*(1-Mu)^2)) * Mu * Vin_peak /(1-Mu) + Vin_peak;
%Is = 1 / (1-Mu) / RL0 * Vs;
for j=1:3
    Is(j) = Vin_peak / ( R0+RL0*((1-Mu(j))^2) );
    Vs(j) = ( Vin_peak*RL0*(1-Mu(j))) / ( R0+RL0*((1-Mu(j))^2) );
end
%昇圧コンバータの状態方程式
%動作点①5kohm
Ac1=[-1/(COL*RL1)      1/COL*(1-Mu(1))
      -1/(LOL)*(1-Mu(1)) -R0/L0];
Bc1=[-Is(1)/COL
      Vs(1)/LO];
%動作点②
Ac2=[-1/(C0*RL0)      1/C0*(1-Mu(2))
      -1/(L0)*(1-Mu(2)) -R0/L0];
Bc2=[-Is(2)/C0
      Vs(2)/LO];
%動作点③
Ac3=[-1/(C0*RL0)      1/C0*(1-Mu(3))
      -1/(L0)*(1-Mu(3)) -R0/L0];
Bc3=[-Is(3)/C0
      Vs(3)/LO];

```

```

Cc=[1 0];
Dc=[0];
o3=[0;0;0];
%伝達関数モデルに変換
[num_c1,den_c1] = ss2tf(Ac1,Bc1,Cc,Dc,1);
[num_c2,den_c2] = ss2tf(Ac2,Bc2,Cc,Dc,1);
[num_c3,den_c3] = ss2tf(Ac3,Bc3,Cc,Dc,1);
%CALC.1%%%%%%%%_DISCRETIZATION_%%%%%%%%%%%%%%%%
%離散時間モデルに変換
[Ad1,Bd1]=c2d(Ac1,Bc1,Ts);
[Aw1,Bw1,Cw,Dw]=c2dt(Ac1,Bc1,Cc,Ts,L);
[Ad2,Bd2]=c2d(Ac2,Bc2,Ts);
[Aw2,Bw2,Cw,Dw]=c2dt(Ac2,Bc2,Cc,Ts,L);
[Ad3,Bd3]=c2d(Ac3,Bc3,Ts);
[Aw3,Bw3,Cw,Dw]=c2dt(Ac3,Bc3,Cc,Ts,L);
[num_d1,den_d1] = ss2tf(Ad1,Bd1,Cc,Dc,1);
[num_d2,den_d2] = ss2tf(Ad2,Bd2,Cc,Dc,1);
[num_d3,den_d3] = ss2tf(Ad3,Bd3,Cc,Dc,1);
[num_w1,den_w1] = ss2tf(Aw1,Bw1,Cw,Dw,1);
[num_w2,den_w2] = ss2tf(Aw2,Bw2,Cw,Dw,1);
[num_w3,den_w3] = ss2tf(Aw3,Bw3,Cw,Dw,1);
%極を計算
[nwd01,dewd1,kd1]=ss2zp(Aw1,Bw1,Cw,Dw);
nwd1(1,1)=nwd01(2,1);
nwd1(2,1)=nwd01(1,1);
[nwd02,dewd2,kd2]=ss2zp(Aw2,Bw2,Cw,Dw);
nwd2(1,1)=nwd02(2,1);
nwd2(2,1)=nwd02(1,1);
[nwd03,dewd3,kd3]=ss2zp(Aw3,Bw3,Cw,Dw);
nwd3(1,1)=nwd03(2,1);
nwd3(2,1)=nwd03(1,1);
%%%%%%%%%%%%%%%
Cww=[Cw 0];
Aww1=[Aw1 Bw1;o3' 0];
Bww1=[o3;1];
[nwwd01,dewwd01,kd001]=ss2zp(Aww1,Bww1,Cww,Dw);
Aww2=[Aw2 Bw2;o3' 0];
Bww2=[o3;1];
[nwwd02,dewwd02,kd002]=ss2zp(Aww2,Bww2,Cww,Dw);
Aww3=[Aw3 Bw3;o3' 0];
Bww3=[o3;1];
[nwwd03,dewwd03,kd003]=ss2zp(Aww3,Bww3,Cww,Dw);
%CALC.2%%%%%%%%_POLE PLACEMENT&MODEL MATCHING_%%%%%%%%%%%%%%%%
P=[-H2 -H1 -H3 -H4];
F=place(Aww1,Bww1,P); %極配置
eig(Aww1-Bww1*F); %
H=(1+H1)*(1+H2)*(1+H3)/((1-nwd1(1,1))*(1-nwd1(2,1)));
E=Cww*Aww1*Bww1;
G=1/E;
%[nwwF1,dewwF1,kwwF1]=ss2zp(Aww1-Bww1*F,Bww1*G*H,Cww,D)
A=Aww1;
B=Bww1;
%%%%%%%%%%%%%%%
FF(1,4)=-B(1,1)/A(1,2);
FF(1,2)=A(1,2);
FF(1,1)=-A(1,1)/A(1,2);
FF(1,3)=-A(1,3)/A(1,2);
Fz=-F(1,4)-F(1,2)*FF(1,4);
%CALC.3%%%%%%%%_ROBUST COMPENSATOR&EQUIVALENT CONVERSION_%%%%%%%%%%%%%%%
%K1=Kz/(1-Kz);
%K2=-2/(1+H2);
k1=-F(1,1)-F(1,2)*FF(1,1)+((-F(1,4)-F(1,2)*FF(1,4))*(-F(1,2)/FF(1,2)))-(G*H*H4+G*H*Fz)*(Kz)*((1)/(1+H2));

```

```

k2=-F(1,2)/FF(1,2)-G*H*(Kz)*((1)/(1+H2));
k3=-F(1,3)-F(1,2)*(FF(1,3));
k4=Fz;
ki=h*(G*H*H4+G*H*Fz)*(Kz)*(1);
kiz=h*G*H*Kz;

k1r=h*G*H;
k2r=h*(G*H*H4+G*H*Fz);
%CALC.4%%%%_DSP PARAMETER DEFINITION%%%%%%%%%%%%%
K1 = real(k1)/(Kadc*Kvout);
K2 = real(k2)/(Kadc*Kvout);
K3 = real(k3);
K4 = real(k4);
Ki1 = real(ki)/(Kadc*Kvout);
Ki2 = real(kiz)/(Kadc*Kvout);
Kr1 = real(k1r)/(Kadc*Kvout);
Kr2 = real(k2r)/(Kadc*Kvout);
%CALC.5%%%%_BODE DIAGRAM %%%%%%%%%%%%%%
%10^-2~10^5.5まで対数的に等間隔な2000点からなる行ベクトルを作成
w=logspace(0.01,5.5,2000);
%①離散系近似的2自由度制御器接続閉ループ特性
A=Aw1;
B=Bw1;
[a,b,c,d] = dlinmod('boost_3point_2dofcontrol_open_dlinmod',Ts);
[mag_do1,phase_do1] = dbode(a,b,c,d,Ts,1,w);
A=Aw2;
B=Bw2;
[a,b,c,d] = dlinmod('boost_3point_2dofcontrol_open_dlinmod',Ts);
[mag_do2,phase_do2] = dbode(a,b,c,d,Ts,1,w);
A=Aw3;
B=Bw3;
[a,b,c,d] = dlinmod('boost_3point_2dofcontrol_open_dlinmod',Ts);
[mag_do3,phase_do3] = dbode(a,b,c,d,Ts,1,w);
figure1=figure(1);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(mag_do1),'k',...
w./(2*pi),20*log10(mag_do2),'k',...
w./(2*pi),20*log10(mag_do3),'k'),grid;
ylabel('Gain[dB]',FontSize,14,FontName,'Times New Roman');
subplot(2,1,2),semilogx(w./(2*pi),phase_do1-360,'k',...
w./(2*pi),phase_do2-360,'k',...
w./(2*pi),phase_do3-360,'k'),grid;
xlabel('Frequency[Hz]',FontSize,14,FontName,'Times New Roman');
ylabel('Phase[degree]',FontSize,14,FontName,'Times New Roman');
legend('point1','point2','point3');
%離散系のゲイン余裕と位相余裕
[Gmd1,Pmd1,Wcgd,Wcpd] = margin(mag_do1,phase_do1,w);
[Gmd2,Pmd2,Wcgd,Wcpd] = margin(mag_do2,phase_do2,w);
[Gmd3,Pmd3,Wcgd,Wcpd] = margin(mag_do3,phase_do3,w);
Gmd1_dB = 20*log10(Gmd1);
Gmd2_dB = 20*log10(Gmd2);
Gmd3_dB = 20*log10(Gmd3);
fprintf('Discrete Gain&Phase margin:$\n',Mu(1)*100,Mu(2)*100,Mu(3)*100);
fprintf('Gmgn=%2.1fdb %2.1fdb %2.1fdb$\n',Gmd1_dB,Gmd2_dB,Gmd3_dB);
fprintf('Pmgn=%2.1f° %2.1f° %2.1f° $\n',Pmd1,Pmd2,Pmd3);

%②離散系近似的2自由度制御器接続閉ループ特性
A=Aw1;
B=Bw1;
[a,b,c,d] = dlinmod('boost_3point_2dofcontrol_closed_dlinmod',Ts);
[mag_dc1,phase_dc1] = dbode(a,b,c,d,Ts,1,w);
[mag_dqyy1,phase_dqyy1] = dbode(a,b,c,d,Ts,2,w);
[mag_dquy1,phase_dquy1] = dbode(a,b,c,d,Ts,3,w);
A=Aw2;
B=Bw2;

```

```

[a,b,c,d] = dlinmod('boost_3point_2dofcontrol_closed_dlinmod',Ts);
[mag_dc2,phase_dc2] = dbode(a,b,c,d,Ts,1,w);
[mag_dqyy2,phase_dqyy2] = dbode(a,b,c,d,Ts,2,w);
[mag_dquy2,phase_dquy2] = dbode(a,b,c,d,Ts,3,w);
A=Aw3;
B=Bw3;
[a,b,c,d] = dlinmod('boost_3point_2dofcontrol_closed_dlinmod',Ts);
[mag_dc3,phase_dc3] = dbode(a,b,c,d,Ts,1,w);
[mag_dqyy3,phase_dqyy3] = dbode(a,b,c,d,Ts,2,w);
[mag_dquy3,phase_dquy3] = dbode(a,b,c,d,Ts,3,w);
figure2=figure(2);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(mag_dc1),'k-',...
w./(2*pi),20*log10(mag_dc2),'k-',...
w./(2*pi),20*log10(mag_dc3),'k'),grid;
ylabel('Gain[dB]', 'FontName','Times New Roman', 'FontSize',14);
subplot(2,1,2),semilogx(w./(2*pi),phase_dc1-360,'k-',...
w./(2*pi),phase_dc2-360,'k-',...
w./(2*pi),phase_dc3-360,'k'),grid;
xlabel('Frequency[Hz]', 'FontName','Times New Roman', 'FontSize',14);
ylabel('Phase[degree]', 'FontName','Times New Roman', 'FontSize',14);
legend('point1', 'point2', 'point3');
%③離散系近似的2自由度制御器qyy特性
figure3=figure(3);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(mag_dqyy1),'k-',...
w./(2*pi),20*log10(mag_dqyy2),'k-',...
w./(2*pi),20*log10(mag_dqyy3),'k'),grid;
ylabel('Gain[dB]', 'FontName','Times New Roman', 'FontSize',14);
subplot(2,1,2),semilogx(w./(2*pi),phase_dqyy1-360,'k-',...
w./(2*pi),phase_dqyy2-360,'k-',...
w./(2*pi),phase_dqyy3-360,'k'),grid;
xlabel('Frequency[Hz]', 'FontName','Times New Roman', 'FontSize',14);
ylabel('Phase[degree]', 'FontName','Times New Roman', 'FontSize',14);
legend('point1', 'point2', 'point3');
%④離散系近似的2自由度制御器quy特性
figure4=figure(4);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(mag_dquy1),'k-',...
w./(2*pi),20*log10(mag_dquy2),'k-',...
w./(2*pi),20*log10(mag_dquy3),'k'),grid;
ylabel('Gain[dB]', 'FontName','Times New Roman', 'FontSize',14);
subplot(2,1,2),semilogx(w./(2*pi),phase_dquy1-360,'k-',...
w./(2*pi),phase_dquy2-360,'k-',...
w./(2*pi),phase_dquy3-360,'k'),grid;
xlabel('Frequency[Hz]', 'FontName','Times New Roman', 'FontSize',14);
ylabel('Phase[degree]', 'FontName','Times New Roman', 'FontSize',14);
legend('point1', 'point2', 'point3');
%⑤根軌跡
%CALC.5.5%%%%_ROOT LOCUS%%%%%%%
num1=((1+H1)*(1+H3))/((1-nwd1(1,1))*(1-nwd1(2,1)))*[1 -(nwd1(1,1)+nwd1(2,1)) nwd1(1,1)*nwd1(2,1)];
den1=[1 H1+H3-1 H1*H3-H1-H3 -H1*H3]; %Kz/(z-1) (z-1)(z+H1)(z+H3)
num2=((1+H1)*(1+H3))/((1-nwd2(1,1))*(1-nwd2(2,1)))*[1 -(nwd2(1,1)+nwd2(2,1)) nwd2(1,1)*nwd2(2,1)];
den2=[1 H1+H3-1 H1*H3-H1-H3 -H1*H3]; %Kz/(z-1) (z-1)(z+H1)(z+H3)
num3=((1+H1)*(1+H3))/((1-nwd3(1,1))*(1-nwd3(2,1)))*[1 -(nwd3(1,1)+nwd3(2,1)) nwd3(1,1)*nwd3(2,1)];
den3=[1 H1+H3-1 H1*H3-H1-H3 -H1*H3]; %Kz/(z-1) (z-1)(z+H1)(z+H3)
%Kzの変化量定義
kwz=0.01; %Kzの最大値は0.37くらい
r=tf(num1,den1);
[r1,kwz1]=rlocus(r,kwz); %根軌跡の計算
r1 = rot90(r1); %出力が横向の行列になっているので縦に変える
r= tf(num2,den2);
[r2,kwz2]=rlocus(r,kwz); %根軌跡の計算
r2 = rot90(r2); %出力が横向の行列になっているので縦に変える
r= tf(num3,den3);
[r3,kwz1]=rlocus(r,kwz); %根軌跡の計算

```

```

r3 = rot90(r3); %出力が横向の行列になっているので縦に変える
%単位円の計算
xk=-1:0.01:1;
for k=1:201
y1(k)=(1-xk(k)^2)^0.5;
y2(k)=-(1-xk(k)^2)^0.5;
end
%根軌跡と単位円のプロット
figure5=figure(5);
axes5 = axes('Parent',figure5,'FontSize',14);
plot(real(r1),imag(r1),'b-',real(r2),imag(r2),'r-',real(r3),imag(r3),'g-',xk,y1,xk,y2,...
    'LineWidth',2,'Parent',axes5'),grid
axis([-1,1,-1,1]);
xlabel('Real part','FontName','Times New Roman','FontSize',14);
ylabel('Imaginary part','FontName','Times New Roman','FontSize',14);
%CALC.6%%%%_STEP RESPONCE_%%%%%%%%
%⑥離散系ステップ応答
A=Aw1;
B=Bw1;
sim('boost_3point_2dofcontrol_closed_dlinmod');
timew1= outState.time;
resw1 = outState.signals(1,1).values;
A=Aw2;
B=Bw2;
sim('boost_3point_2dofcontrol_closed_dlinmod');
timew2= outState.time;
resw2 = outState.signals(1,1).values;
A=Aw3;
B=Bw3;
sim('boost_3point_2dofcontrol_closed_dlinmod');
timew3= outState.time;
resw3 = outState.signals(1,1).values;
figure6=figure(6);
axes6 = axes('Parent',figure6,'FontSize',14);
plot(timew1,resw1,'k-',timew2,resw2,'k-',timew3,resw3,'k',...
    'Parent',axes6'),grid;
xlabel('Time[sec]','FontName','Times New Roman','FontSize',14);
ylabel('Output Voltage[V]','FontName','Times New Roman','FontSize',14);
axis([0.49,0.8,-1,22]);
legend('point1', 'point2', 'point3');
%負荷急変応答
%連続時間
Ac = Ac1;
Bc = Bc1;
A1_varRL1 = Ac3;
B1_varRL1 = Bc3;
A = Ad1;
B = Bd1;
Adl = Ad3;
Bdl = Bd3;
num_dl = [0.1];
den_dl = [1 -0.9];
num_dl = [1];
den_dl = [1];
%連続
num = [1];
den = [1/4e4 1];
%OUT.1%%%%% SHOW CONTROLLER PARAMETER_%%%%%%%%
t = clock; %現在時刻を格納
%C言語記述用出力
fprintf('↓ OUTPUT FOR HEW ↓\n');
fprintf('//No.%d\n',No);
fprintf('//at %4d/%02d/%02d:%02d:%02d.0f\n',t(1,1),t(1,2),t(1,3),t(1,4),t(1,5),t(1,6));

```

```

fprintf('//H1=%1.2f+%1.2fi\n',real(H1),imag(H1));
fprintf('//H2=%1.5f\n',H2);
fprintf('//H3=%1.2f+%1.2fi\n',real(H3),imag(H3));
fprintf('//H4=%1.5f\n',H4);
fprintf('//Kz=%1.3f\n',Kz);
#define K1 %d
#define K2 %d
#define K3 %d
#define K4 %d
#define Ki1 %d
#define Ki2 %d
#define KR1 %d
#define KR2 %d

```

## A.2 力率改善制御する PFC 昇圧型コンバータの MATLAB プログラム

format short e; %小数点表示桁指定

```

format compact; %余計な改行を省く
clc;           %コマンドウインドウのクリア
clear all;     %ワークスペースのクリア
close all;     %図をすべて閉じる
warning('off'); %警告を出さない
%%%%%%%%%%%%_COMMENT_%%%%%%%%%%%%%%%%
% pfc_2DOFcontrol_characteristics_v2.m
% Renesas PFCIC HA16174評価ボードを使った力率改善
% 電流ループと電圧ループに2DOF制御を使った時の特性
% 無負荷時や5kΩ負荷時を考慮
% copyright (c) Yoshihiro Ohta, Kohji Higuchi
% The University of Electro-Communications
%STEP.1%%%_SAMPLING FREQUENCY_%%%%%%%%%%%%%%
fs=100e3; %[Hz] e3=(*100)
Ts = 1/fs; %sampling frequency [Hz]→ time[sec]
%STEP.2%%%_CHANGE PARAMETER_%%%%%%%%%%%%%%
%極とパラメータの指定
%H2iはそのまま、H2vだけ変えてみる
%H2vを下げるとき電流波形電圧応答とともに悪化
%H2vをあげると電流はましになって電圧は早く復帰
No = 129;
H1i = -0.50;
H2i = -0.999927137130156;
H3i = 0.2;
kzi = 0.35;
H2v = -0.999927137130156;
kzv = 0.25;
%OPTION.0%%%_SIMULATION PARAM_%%%%%%%%%%%%%%
simTime = 1;
changeTime = 0.3;
wend = log10(fs*2*pi/2); %周波数の終点を計算
w=logspace(0.01,wend,2000);
%OPTION.1%%%_CONTROLED OBJECT PARAM_%%%%%%%%%%%%%%
Tdelay = Ts ; %delay time
L = 0.99*Ts ; %delay time
Vin      = 100;      %入力電圧AC100V
Vin_peak = Vin*sqrt(2); %入力のピーク電圧
fvin    = 50;        %入力AC周波数
%Vout   = 240;       %出力電圧目標値

```

```

%Mu      = 0.43;          %目標出力電圧でのデューティ
Vout    = 385;            %出力電圧目標値
Vout    = [500,385,240];   %出力電圧目標値
Mu      = 0.65;            %目標出力電圧でのデューティ
L0      = 150e-6;          %インダクタンス
C0      = (470e-6)*2;     %出力キャパシタ
R0      = 1.8;              %インダクタの等価直列抵抗
RC      = 0.3;              %キャパシタの等価直列抵抗
C1      = 1e-3;             %入力キャパシタ
Risence = 0.033/2;        %電流検出抵抗
RL0     = 292;              %負荷抵抗
%軽負荷時の特性変化考慮用パラメータ
RL1     = 5000;
COL     = (12000e-6)*2;
L0L     = 200e-6;           %インダクタンス
RL_DL0  = 500;              %負荷急変用の抵抗
RL_DL1  = 650;              %負荷急変用の抵抗
Kvac   = 2.3125/150;       %入力電圧分圧ゲイン
Kiac   = 3000/200;          %入力電流検出ゲイン
Kadc   = 4096/5;            %ADCゲイン {2^(12)/5[V]}
Kvout= 2.4805/400;         %出力電圧分圧ゲイン
vavg  = 2/pi*sqrt(2)*100;
Vavg  = 1/((vavg*Kvac*Kadc)^2);  %平均電圧(sinの平均は2/pi)
cpuClock = 200e6;           %SH7216 200[MHz]
mtuClock = cpuClock/2;       %MTU2SのクロックPΦ
mtuTime  = 1/mtuClock;       %MTUの1ステップ時間
step     = round(Ts/mtuTime); %ステップ数
Kpwm = 1/step;

%CALC.0%%%%_STATE EQUATION%%%%%%%
%平均出力電圧と電流の対デューティ比プロット
for i= 1:999
    Mu = i/1000;
    Vs(i) = Vin_peak* ((RL0*(1-Mu))/(RL0*(1-Mu)^2 + R0));
    Is(i) = 1 / (1-Mu) / RL0 * Vs(i);
end
for i=1:99
    Vss(i)=Vs(round(i*10));
    lss(i)=Is(round(i*10));
end
figure(1)
plot(1:99,lss,'k-',1:99,Vss,'k-');
xlabel('duty ratio[%]', 'FontSize', 14);
ylabel('Output Voltage[V], Inductor Current[A]', 'FontSize', 14);
grid;

%実測値csvデータを読み込み表示
figure(1001);
ActResult = csvread('ActResult.csv');
duty292 = ActResult(1:26,1);
data292 = ActResult(1:26,2);
duty500 = ActResult(1:18,3);
data500 = ActResult(1:18,4);
duty1k  = ActResult(1:30,5);
data1k  = ActResult(1:30,6);
duty5k  = ActResult(1:15,7);
data5k  = ActResult(1:15,8);
dutynon = ActResult(1:9,9);
datanon = ActResult(1:9,10);
plot(duty292./10,data292,'k-' ,duty500./10,data500,'k:' ,...
      duty1k./10,data1k , 'k-' ,duty5k./10 ,data5k , 'k-x' ,...
      dutynon./10,datanon , 'k-' );
axis([0,70,0,450]);
grid;
xlabel('duty ratio[%]', 'FontSize', 14);

```

```

ylabel('OutputVoltage[V]', 'FontName', 'Times New Roman', 'FontSize', 14);
legend('RL=300', 'RL=500', 'RL=1000', 'RL=5000', 'RL=No load');
%3つの動作点におけるデューティを計算
for i= 2:999
    for j= 1:3
        if(Vs(i)>Vout(j) && Vs(i-1)<=Vout(j))
            Mu(j) = i/1000;
        end
    end
end
%それぞれのデューティでIsとVsを求める
%Vs = 1 / (1+R0/(RL0*(1-Mu)^2)) * Mu * Vin_peak /(1-Mu) + Vin_peak;
%Is = 1 / (1-Mu) / RL0 * Vs;
for j=1:3
    Is(j) = Vin_peak / ( R0+RL0*((1-Mu(j))^2));
    Vs(j) = ( Vin_peak*RL0*(1-Mu(j))) / ( R0+RL0*((1-Mu(j))^2));
end
%%%%%%%%%%%%%
%最初は電流フィードバックループを作る%
%%%%%%%%%%%%%
%微小変動に対する回路状態方程式
%動作点無印385V300ohm
Ac=[-R0/L0 -(1-Mu(2))/(L0)
    (1-Mu(2))/C0 -1/(C0*RL0)];
Bc=[ Vs(2)/L0
    -Is(2)/C0];
Cc=[1 0];
Dc=[0];
%動作点①385V5kohm
Ac1=[-R0/L0 -(1-Mu(1))/(L0)
    (1-Mu(1))/C0 -1/(C0*RL1)];
Bc1=[ Vs(1)/L0
    -Is(1)/C0];
%動作点③240V300ohm
Ac3=[-R0/L0 -(1-Mu(3))/(L0)
    (1-Mu(3))/C0 -1/(C0*RL0)];
Bc3=[ Vs(3)/L0
    -Is(3)/C0];
%状態方程式を時間遅れ付きで離散化
[Aw ,Bw ,Cw,Dw] = c2dt(Ac ,Bc ,Cc,Ts,L);
[Aw1,Bw1,Cw,Dw] = c2dt(Ac1,Bc1,Cc,Ts,L);
[Aw3,Bw3,Cw,Dw] = c2dt(Ac3,Bc3,Cc,Ts,L);
%離散化状態方程式のボード線図を描画
[mag_w ,phase_w ] = dbode(Aw ,Bw ,Cw,Dw,Ts,1,w);
[mag_w1,phase_w1] = dbode(Aw1,Bw1,Cw,Dw,Ts,1,w);
[mag_w3,phase_w3] = dbode(Aw3,Bw3,Cw,Dw,Ts,1,w);
figure(2);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(mag_w1),'k-',...
    w./(2*pi),20*log10(mag_w ),'k-',...
    w./(2*pi),20*log10(mag_w3),'k:'),grid;
ylabel('Gain[dB]', 'FontName', 'Times New Roman', 'FontSize', 14);
axis([0.1*2,fs/2,10,50]);
subplot(2,1,2),semilogx(w./(2*pi),phase_w1,'k-',...
    w./(2*pi),phase_w , 'k-',...
    w./(2*pi),phase_w3,'k:'),grid;
xlabel('Frequency[Hz]', 'FontName', 'Times New Roman', 'FontSize', 14);
ylabel('Phase[degree]', 'FontName', 'Times New Roman', 'FontSize', 14);
axis([0.1*2,fs/2,-250,100]);
legend('point1', 'point2', 'point3');
%連続系のステップ応答を記録
sim('pfc_controlledobject_linmod');
timec= outState.time;
resc = outState.signals(1,1).values;

```

```

%CALC.1%%%%%%%%_DISCRETIZATION_%%%%%%%%%%%%%%%%
%状態方程式を遅れ考慮で離散化
%[Ad,Bd] = c2d(Ac,Bc,Ts);
%[Aw,Bw,Cw,Dw] = c2dt(Ac,Bc,Cc,Ts,L);
[numw,denw] = ss2tf(Aw,Bw,Cw,Dw,1);
%極を計算
[nwd01,dewd1,kd1]=ss2zp(Aw,Bw,Cw,Dw,1);
nwd1(1,1)=nwd01(2,1);
nwd1(2,1)=nwd01(1,1);
%離散系のボード線図
figure(6)
[magd,phased] = dbode(Aw,Bw,Cw,Dw,Ts,1,w);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(magd),'b-'),grid;
title('離散系の状態方程式のr-iボード線図','FontSize',14);
ylabel('Gain[dB]', 'FontSize',14);
subplot(2,1,2),semilogx(w./(2*pi),phased,'b-'),grid;
xlabel('Frequency[Hz]', 'FontSize',14);
ylabel('Phase[degree]', 'FontSize',14);
%CALC.2%%%%%%%%_POLE PLACEMENT&MODEL MATCHING_%%%%%%%%%%%%%%%%
%状態フィードバック則を適用し
%モデルマッチング系を作る
%極配置
P=[-H2i -H1i -H3i];
F=place(Aw,Bw,P);
eig(Aw-Bw*F); %極が正しく配置されているか確認
[nwd00,dewd0,kd0]=ss2zp(Aw,Bw,Cw,Dc); %状態空間から零点・極に変換(s-z1)(s-z2).../(s-p1)(s-p2)...
nwd0(1,1)=nwd00(2,1);
nwd0(2,1)=nwd00(1,1);
H=(1+H1i)*(1+H2i)*(1+H3i)/((1-nwd0(1,1))*(1-nwd0(2,1)));
I=eye(3);
G = 1/(Cw*(I/(I-(Aw-Bw*F))*Bw));%モデルマッチング系の定常ゲインの求め方2
[a,b,c,d] = dlinmod('pfc_statefeedback_dlinmod',Ts);
[nnn,ddd,kkk]=ss2zp(a,b,c,d,1)
%CALC.3%%%%%%%%_ROBUST COMPENSATOR&EQUIVALENT CONVERSION_%%%%%%%%%%%%%%
%ロバスト制御器の等価変換後のパラメータを決定
k1=-( F(1,1)+G*kzi/(1+H1i) );
%k1=-( F(1,1)+G*kzi/(1-H2i) ); %間違い
k2=-F(1,2);
k3=-F(1,3);
ki=G*kzi;
kr=G;
%周辺ゲインの補正
K1i = k1/(Kiac*Kadc*Risence);
K2i = k2/(Kvout*Kadc);
K3i = k3;
Kii = ki/(Kiac*Kadc*Risence);
Kri = kr/(Kiac*Kadc*Risence);
%ゲイン補正した制御器のステップ応答
sim('pfc_2DOF_currentcontrol_coefficient');
timeg= outState.time;
resg = outState.signals(1,1).values;
figure(11)
plot(timeg,resg,'b-'),grid
title('ゲイン補正した2dof制御器のr-iステップ応答','FontSize',14);
axis([0,0.003,0,1.2]);
%制御器の閉ループr-i特性
figure(12)
[a,b,c,d] = dlinmod('pfc_2DOF_currentcontrol_coefficient',Ts);
[magi,phasei] = dbode(a,b,c,d,Ts,1,w);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(magi),'b-'),grid;
title('2dof制御閉ループr-i特性','FontSize',14);
ylabel('Gain[dB]', 'FontSize',14);
subplot(2,1,2),semilogx(w./(2*pi),phasei,'b-'),grid;

```

```

xlabel('Frequency[Hz]', 'FontSize', 14);
ylabel('Phase[degree]', 'FontSize', 14);
%制御器のopenループr-i特性
figure(22)
[a,b,c,d] = dlinmod('pfc_2DOF_currentcontrol_coefficient_openloop', Ts);
[magjo,phaseio] = dbode(a,b,c,d,Ts,1,w);
subplot(2,1,1), semilogx(w./((2*pi)),20*log10(magjo), 'b-'), grid;
title('2dof制御開ループr-i特性', 'FontSize', 14);
ylabel('Gain[dB]', 'FontSize', 14);
subplot(2,1,2), semilogx(w./((2*pi)),phaseio, 'b-'), grid;
xlabel('Frequency[Hz]', 'FontSize', 14);
ylabel('Phase[degree]', 'FontSize', 14);
%根軌跡-----
%Kzの変化量定義
kwz=0:0.01:kz;
A=Aw;B=Bw;
[a,b,c,d] = dlinmod('pfc_2DOF_currentcontrol_rlocus_v2', Ts);
[r,kwz]=rlocus(a,b,c,d,kwz); %根軌跡の計算
%z-1+kz*Ws=0の根の表現
num1=((1+H2i)*(1+H3i))/((1-nwd1(1,1))*(1-nwd1(2,1)))*[1 -(nwd1(1,1)+nwd1(2,1)) nwd1(1,1)*nwd1(2,1)];
den1=[1 H2i+H3i-1 H2i*H3i-H2i-H3i -H2i*H3i]; %Kz/(z-1) (z-1)(z+H1)(z+H3)
rr=tf(num1,den1);
[r,kwz]=rlocus(rr,kwz); %根軌跡の計算
r = rot90(r); %出力が横向の行列になっているので縦に変える
A=Aw1;B=Bw1;
[a,b,c,d] = dlinmod('pfc_2DOF_currentcontrol_rlocus_v2', Ts);
[r1,kwz1]=rlocus(a,b,c,d,kwz); %根軌跡の計算
A=Aw3;B=Bw3;
[a,b,c,d] = dlinmod('pfc_2DOF_currentcontrol_rlocus_v2', Ts);
[r3,kwz3]=rlocus(a,b,c,d,kwz); %根軌跡の計算
%単位円の計算
xk=-1:0.01:1;
for k=1:201
y1(k)=(1-xk(k)^2)^0.5;
y2(k)=-(1-xk(k)^2)^0.5;
end
%根軌跡と単位円のプロット
figure(21);
plot(real(r),imag(r), 'k-',xk,y1, 'k-',xk,y2, 'k-'), grid
plot(real(r),imag(r), 'b-',xk,y1, 'g-',xk,y2, 'g-'), grid
axis([-1,1,-1,1]);
xlabel('Real Part', 'FontName', 'Times New Roman');
ylabel('Imaginary Part', 'FontName', 'Times New Roman');
%根軌跡ここまで-----
%制御器の入力に入力電圧補正乗算器をつけた時の閉ループr-v特性
figure(13)
[Awv,Bwv,Cwv,Dwv] = dlinmod('pfc_2DOF_currentcontrol_rvcharacteristics', Ts);
dbode(Awv,Bwv,Cwv,Dwv,Ts,1,w), grid;
title('入力に乗算器をつけた2dof制御閉ループr-v特性', 'FontSize', 14);
[nnn,ddd,kkk]=ss2zp(Awv,Bwv,Cwv,Dwv,1)
abs(ddd)
%入力電圧補正乗算器をつけた電流制御器のr-vステップ応答
sim('pfc_2DOF_currentcontrol_rvcharacteristics');
timerv= outState.time;
resrv = outState.signals(1,2).values;
figure(14)
plot(timerv,resrv, 'b-'), grid
title('入力に乗算器をつけた2dof制御器r-vステップ応答', 'FontSize', 14);
axis([0,1,0,0.001]);
%%%%%%%%%%%%%%%
%次に、 figure11のAwv,Bwv,Cwv,Dwvを新たな制御対象とみなし%
%電圧ループニ自由度制御器を設計する %
%%%%%%%%%%%%%%%

```

```

%1次モデル近似した時のH2=0.999927
%極配置するわけではないから、H1とH3、H4は不要
vavg = Vin_peak*2/pi;
%ri-v定常ゲインを求める
%zv=Aw(2,2)*v+Aw(2,1)*i+Aw(2,3)*u/z+Bw(2,1)*uで、
%定常ゲインはz=1,i=1のときのものを求めるので
[numw,denw] = ss2tf(Aw,Bw,Cw,Dw,1);
Gui = sum(numw) / sum(denw); %u-i定常ゲイン
%とすると、
%v=( Aw(2,1)+Aw(2,3)/Gui+Bw(2,1)/Gui ) /( 1-Aw(2,2) )であるから
Guv=( Aw(2,1)+Aw(2,3)/Gui+Bw(2,1)/Gui ) /( 1-Aw(2,2) );
%ri-vのゲイン
Griv= Guv; %これにKadc*Kiac*Risenceは含まれている
%乗算器をつけて、その前からvまでのゲイン
Guvv = ( 1/ (vavg*Kvac*Kadc) *Vin_peak*Kvac*Kadc) *Griv;
Gv = 1/Guvv;
%ゲイン補正をした場合
Krv = Gv/(Kadc*Kvout);
Kiv = Gv*kzv/(Kadc*Kvout);
K1v = -Gv*kzv/(1+H2v)/(Kadc*Kvout);
%電圧制御器も2dofのステップ応答
%sim('pfc_2DOF_ivcontrol_app_ev_v2');
A=Aw; B=Bw ;
sim('pfc_2DOF_ivcontrol_app_ev_v3');
timeiva = outStatei.time;
resivai = outStatei.signals(1,1).values;
resivav = outStatei.signals(1,2).values;
A=Aw1;B=Bw1;
sim('pfc_2DOF_ivcontrol_app_ev_v3');
timeiva1 = outStatei.time;
resivai1 = outStatei.signals(1,1).values;
resivav1 = outStatei.signals(1,2).values;
A=Aw3;B=Bw3;
sim('pfc_2DOF_ivcontrol_app_ev_v3');
timeiva3 = outStatei.time;
resivai3 = outStatei.signals(1,1).values;
resivav3 = outStatei.signals(1,2).values;
figure(19)
plot(timeiva,resivav,'b-',timeiva1,resivav1,'r-',timeiva3,resivav3,'g-'),grid
ylabel('Output Voltage[V]',FontSize',14);
legend('point1','point2','point3');
xlabel('time[sec]',FontName,'Times New Roman',FontSize',14);
%根軌跡-----
%Kzの変化量定義
kwzv=0.01:kzv;
A=Aw;B=Bw;
[a,b,c,d] = dlinmod('pfc_2DOF_ivcontrol_rlocus',Ts);
[rv,kwzv]=rlocus(a,b,c,d,kwzv); %根軌跡の計算
%単位円の計算
xk=-1:0.01:1;
for k=1:201
y1(k)=(1-xk(k)^2)^0.5;
y2(k)=-(1-xk(k)^2)^0.5;
end
%根軌跡と単位円のプロット
figure(31);
plot(real(rv),imag(rv),'b-',xk,y1,'k-',xk,y2,'r-'),grid
axis([-1,1,-1,1]);
xlabel('Real Part',FontName,'Times New Roman');
ylabel('Imaginary Part',FontName,'Times New Roman');
%根軌跡ここまで-----
%制御器の閉ループr-v特性
figure(23)

```

```

[a,b,c,d] = dlinmod('pfc_2DOF_ivcontrol_app_ev_v2',Ts);
[magv,phasev] = dbode(a,b,c,d,Ts,1,w);
subplot(2,1,1),semilogx(w./(2*pi),20*log10(magv),'b-'),grid;
title('2dof制御閉ループrv-v特性','FontSize',14);
ylabel('Gain[dB]','FontSize',14);
subplot(2,1,2),semilogx(w./(2*pi),phasev,'b-'),grid;
xlabel('Frequency[Hz]','FontSize',14);
ylabel('Phase[degree]','FontSize',14);
[nnn,ddd,kkk]=ss2zp(a,b,c,d,1)
abs(ddd)

%OUT.1%%%%_SHOW CONTROLLER PARAMETER_%%%%%%%%%%%%%%%
%制御器パラメータのC言語開発用表示
t = clock; %現在時刻を格納
%C言語記述用出力
fprintf('↓ OUTPUT FOR HEW ↓\n');
fprintf('//No.%d\n',No);
fprintf('//at %4d/%02d/%02d/%02d:%02d.%02f\n',t(1,1),t(1,2),t(1,3),t(1,4),t(1,5),t(1,6));
fprintf('//H1i=%1.2f+%1.2fj\n',real(H1i),imag(H1i));
fprintf('//H2i=%1.6f\n',H2i);
fprintf('//H3i=%1.2f+%1.2fj\n',real(H3i),imag(H3i));
fprintf('//Kzi=%1.4f\n',kzi);
fprintf('#define K1i %d\n',K1i);
fprintf('#define K2i %d\n',K2i);
fprintf('#define K3i %d\n',K3i);
fprintf('#define KUI %d\n',Kii);
fprintf('#define KRI %d\n',Kri);
fprintf('//H2v=%1.6f\n',H2v);
fprintf('//Kzv=%1.4f\n',kzv);
fprintf('#define K1V %d\n',K1v);
fprintf('#define KUV %d\n',Kiv);
fprintf('#define KRV %d\n',Krv);

%グラフ上のフォントサイズとグラフの線幅の指定
FontSize = 16;
LineWidth = 2;
FontName = 'Times New Roman';

```

# 付録 B マイクロコントローラプログラム

主要プログラム main.c と interrupt.c から抜粋

## main.c

```
//----- プログラムここから -----//
//ヘッダとか
#include <machine.h> //nop();
#include "iodefine.h"
#include "interrupt.h"
#include "uart.h"
#include "macro.h"
#include "param.h"

//関数プロトタイプ宣言
void main(void); //メインプログラム
void initADC0(void); //ADC の初期設定
void initMTU2S(void); //MTU2S の初期設定
void initPORT(void); //Port の初期設定

//ペリフェラル
void wait(unsigned int w); //ウェイト関数
//-----
// エントリーポイント
//-----
void main(void)
{
    int i;
    //Uart の初期設定
    initSCI1();
    //テストメッセージ
    puts("\r\n");puts("\r\n");puts("\r\n");
    puts("SH_PFC_Control_Program\r\n");
    puts(PRJNAME); puts("\r\n");
    puts("target output voltage is ");
    putn( (unsigned long)((TRUE_TARGET+19.19)/5.143) );
    puts("[V]\r\n");
    puts("\r\n");
    //周辺モジュールの初期設定
    CPG.MCLKCR.BYTE = 0x40; //MTU2S のクロックを 100MHz に設定
    initValue(); //制御変数の初期化
    initPORT(); //Port の初期設定
    initADC0(); //AD の初期設定
    initMTU2S(); //MTU2S の初期設定

    for(i = 0; i<16 ; i++) AnalogData.vbuf[i] = 0;
    AnalogData.v_idx = 0;
    AnalogData.vp_idx = 1023;
    AnalogData.b_idx = 0;
    AnalogData.bp_idx = 0;
    //メインループ
    while(1){
    }
}
//-----
```

## interrupt.c

```
//ヘッダ
#include "iodefine.h"
#include "interrupt.h"
#include "uart.h"
#include "macro.h"
#include "param.h"
#include "sin_wave.h"
//-----制御用変数-----
int Vrv1; //可変抵抗電圧・RV1
int lac; //入力電流・AN1
int Vac; //入力電圧・AN2
int Vout; //出力電圧・AN3
//2dof用
volatile int Vtrg; //目標値電圧
volatile float Vdif; //差分電圧
volatile float Itrg; //目標値電流
volatile float Idif; //差分電流
volatile float Upwm; //出力デューティ比
//#define Vavg 8.794769e-004 //平均入力電圧(AC100V時)=1/(2/pi*sqrt(2)*100*Kvac*Kadc)
#define Vavg 4.3974e-004 //平均入力電圧(AC200V時)=1/(2/pi*sqrt(2)*200*Kvac*Kadc)
//本来は Vac から計算して導出する

//各状態パラメータ
volatile float Uv;
volatile float Uvi;
volatile float Uviz;
volatile float Ua;
volatile float Ua4;
volatile float Up3;
unsigned long cnt; //ソフトスタート用カウント変数
unsigned long mtd; //使用する制御ルーチン選択変数
unsigned int state=OFF; //PWM出力モード変数
//OFF =0 OFF モード :duty0.1%
//OPEN =1 OPEN モード :duty=RV1 の値
//SOFTSTART =2 SOFTSTART モード :徐々に大きく
//PI =3 PI モード :PI 制御ルーチン
//TDOF =4 2DOF モード :2DOF 制御ルーチン
//PIPFC =5 PIPFC モード :PI の力率改善制御ルーチン
//TDOFPFC=6 TDOFPFC モード :2DOF の力率改善制御ルーチン
//-----制御用変数ここまで-----
//目標値設定
//ソフトスタート終了電圧
#define SOFT_TARGET CONST150
//最終目標値の前の目標値とリミッタ
//#if (TRUE_TARGET == CONST385)
//    #define BEFORE_TARGET CONST365
//    #define DUTY_MAX 950
#if (TRUE_TARGET == CONST400)
    #define BEFORE_TARGET CONST385
    #define DUTY_MAX 950
#elif (TRUE_TARGET == CONST300)
    #define BEFORE_TARGET CONST280
    #define DUTY_MAX 900
#elif (TRUE_TARGET == CONST240)
    #define BEFORE_TARGET CONST220
    #define DUTY_MAX 880
#endif //#if (TRUE_TARGET == CONST385)

//リミッタ定数
#define UV_LIMIT 7400000
#define DUTY_MIN 1
TAnalogData AnalogData;
```

```

//-----
///// 割り込み関数群
/////
//ADC 終了割り込み処理内容
//ここで PWM デューティ比の更新を行う
//注：最後に必ず TGRD4 を更新すること
void intrADC0(void){
    LED3_ON();           //割り込み確認の為 LED3=PE13 を ON
    if(ADC0.ADSR.BIT.ADF != 0)          //ADC 終了フラグを解除
        ADC0.ADSR.BIT.ADF = 0;

    //AD 結果を格納
    Vrv1 = ADC0.ADDR0; //可変抵抗電圧・RV1
    lac      = ADC0.ADDR1; //入力電流・AN1
    Vac      = ADC0.ADDR2; //入力電圧・AN2
    Vout = ADC0.ADDR3; //出力電圧・AN3

    //----保護プログラム・変更不可----//
    //過電圧保護(出力電圧 420V 以上で強制停止)
    if(Vout >= CONST420){
        state = OFF;
        LED0_OFF();
        LED1_OFF();
        LED2_OFF();
        LED5_ON();
        initialValue();
        puts("VOLTAGE_LIMIT!!");
    }
    //過電流保護(インダクタ電流 15A 以上で強制停止)
    if(lac >= C_CONST15A){
        state = OFF;
        LED0_OFF();
        LED1_OFF();
        LED2_OFF();
        LED5_ON();
        initialValue();
        puts("CURRENT_LIMIT!!");
    }
    //----保護プログラム・変更不可----//
    //----入力電圧波形生成----//
    //16 点の移動平均して、同期電圧目標値波形を生成する
    //データバッファに入力電圧を格納
    AnalogData.vbuf[AnalogData.b_idx] = Vac;
    if(AnalogData.b_idx++ >= 15)    AnalogData.b_idx = 0;
    if(AnalogData.bp_idx++ >= 15)   AnalogData.bp_idx = 0;
    //16 点の和を計算
    sum = AnalogData.vbuf[0] + AnalogData.vbuf[1] + AnalogData.vbuf[2] + AnalogData.vbuf[3] +
          AnalogData.vbuf[4] + AnalogData.vbuf[5] + AnalogData.vbuf[6] +
          AnalogData.vbuf[7] + AnalogData.vbuf[8] + AnalogData.vbuf[9] + AnalogData.vbuf[10] +
          AnalogData.vbuf[11] + AnalogData.vbuf[12] + AnalogData.vbuf[13] + AnalogData.vbuf[14] +
          AnalogData.vbuf[15];
    //和から 16 点の平均を算出(4bit 右シフト→x/2^4)
    AnalogData.vac[AnalogData.v_idx] = (sum >> 4);
    if(AnalogData.v_idx++ >= 1023)      AnalogData.v_idx = 0;
    if(AnalogData.vp_idx++ >= 1023)     AnalogData.vp_idx = 0;
    if(Vac > VAC_THRESHOLD){
        resetFlg = 0;
        //割り込み確認の為 LED3=PE13 を OFF
    }
    //データが閾値より小さく、ゼロ検出した場合サイン波をリセット
    if((resetFlg == 0)&& (Vac < VAC_THRESHOLD) && (AnalogData.vac[(AnalogData.vp_idx)] <
    AnalogData.vac[AnalogData.v_idx]) )
    {

```

```

        sincnt = 965;
        resetFlg = 1;
    //        LED3_ON();           //割り込み確認の為 LED3=PE13 を ON
    }

//----入力電圧波形生成----//
//PWM デューティ出力モードの場合分け
switch( state ){
    case OFF: //PWM 出力しない(state = 0=OFF)
        Upwm      = 1;
        break;

    case OPEN: //制御ルーチンを使わない(state = 1=OPEN)
        Upwm = ((unsigned int)Vrv1>>2); //PWM 出力は CPU ボード上の RV1 に依存
        if(Upwm>650)          Upwm = 650;           //リミッタ
        break;

    case SOFTSTART: //ソフトスタート(state = 2=SOFTSTART duty50%制限・出力 150V で制御ルーチンへ
    移行)
        Upwm = (cnt++>>10);
        if(Upwm>500)          Upwm = 500;           //リミッタ
        if(Vout>SOFT_TARGET){//出力が 220V になったら制御ルーチンに移行
            state = mtd;//指定した制御器のルーチンへ
            cnt = 0;
            LED1_ON();
            puts("Control Mode.\r\n");
        }
        break;

    case PI://2dof 制御ルーチンを使う(stepState = 3=PI)
        stepCount();
        //-----PI 制御ルーチン-----//
        Vdif = (volatile float)(Vtrg - Vout);
        Upwm = (volatile float)(Vdif*KP + Ua*KI);
        Ua      = (volatile float)(Ua+Vdif);
        if(Upwm>800)          Upwm = 800;
        currentLimiter();
        //-----PI 制御ルーチンここまで-----//
        break;

    case TDOF: //2dof 制御ルーチンを使う(stepState = 4=TDOF)
        stepCount();
        //---boost-1DVFB2DOF 制御ルーチン-----//
        Vdif = (volatile float)(Vtrg - Vout);
        Upwm = (volatile float)(Uvz+Ua+K2*Vout);
        Ua      = (volatile float)(Uvi+K1*Vout+Up3+Ua4);
        Uv      = (volatile float)(Vdif + Uv);
        Uvz = (volatile float)(K12* Uv);
        Uvi     = (volatile float)(KI1* Uv);
        Ua4     = (volatile float)(K4 * Ua);
        Up3     = (volatile float)(K3 * Upwm);
        if(Upwm>800)          Upwm = 800;
        currentLimiter();
        //---boost-1DVFB2DOF 制御ルーチンここまで-----//
        //変数リミッタ
        if(Uv<UV_LIMIT)       Uv = UV_LIMIT;           //5k Ω 以上の負荷 ⇔ 無負荷の急変時に必要
        if(Ua4>0)              Ua4 = 0;
        if(Up3>0)              Up3 = 0;
        break;

    case PIPFC:
        stepCount();
        //-----PFC-PI 制御ルーチン-----//

```

```

        Vdif = (volatile float)(Vtrg - Vout);
        Ua      = (volatile float)(Vdif*KPV + Uv*KIV);
        Uv      = (volatile float)(Uv + Vdif);
    //
        ltrg = (volatile float)(Ua*sin_wave[sincnt]*Vavg - lac);
        ltrg = (volatile float)(Ua*Vac*Vavg - lac);
        Upwm = (volatile float)(ltrg*KPI + Uvi*KII);
        Uvi      = (volatile float)(Uvi + ltrg);
        if(Upwm>910)          Upwm = 910;
        if(Uv <0)              Uv  = 0; //出力電力操作量が負にならないようにする.
        currentLimiter();
        //-----PFC-PI 制御ルーチン-----//
        break;

    case TDOFPFC:
        stepCount();
        //-----PFC-2DOF 制御ルーチン-----//
        //FF あり
        Upwm      = (volatile float)(KRI*ltrg + KUI*Uvi + K1I*lac + K2I*Vout + Up3);
        Uv         = (volatile float)(KRV*Vtrg + KUV*Ua + K1V*Vout);
        //FF なし
        Upwm      = (volatile float)(KUI*Uvi + K1I*lac + K2I*Vout + Up3);
        Uv         = (volatile float)(KUV*Ua + K1V*Vout);
        //
        ltrg      = (volatile float)(sin_wave[sincnt] * Vavg * Uv); //生成した sin 波を使う
        ltrg      = (volatile float)(Vac * Vavg * Uv);
        Uvi      = (volatile float)(Uvi + ltrg - lac);
        Ua       = (volatile float)(Ua + Vtrg - Vout);
        Up3      = (volatile float)(K3I * (Upwm));
        Upwm      = (volatile float)(Upwm*1000);
        if(Upwm>950)          Upwm = 950;
        currentLimiter();
        //-----PFC-2DOF 制御ルーチンここまで-----//
        break;

    }
    //state 判定終了
//-----保護・デューティ更新・変更不可-----//
//デューティ制限
if(Upwm>DUTY_MAX)  Upwm = DUTY_MAX;
if(Upwm<DUTY_MIN)  Upwm = DUTY_MIN;
//サイン波計算値を更新
if(sincnt++ > 1998) sincnt = 0;
//R2RDAC 出力の更新
// PC.DR.WORD      = sin_wave[sincnt];           //生成 sin 波の出力
// PC.DR.WORD      = (unsigned int)(Upwm*4);        //操作量の出力
// PWM の更新
MTU2S4.TGRD = ((volatile int)Upwm);           //PWM 値の出力
//-----保護・デューティ更新・変更不可-----//
LED4_TOGGLE(); //サンプリング周波数確認の為 LED4=PE14 を反転
LED3_OFF();   //割り込み確認の為 LED3=PE13 を OFF
}
//-----
//NMI 割り込み処理内容
void intrNMI(void){
    MTU2S4.TGRD = (1); //PWM 値の初期化

    if( state != OFF ){ //PWM 出力を停止 OFF にする
        puts("RESET\r\n");
        state = OFF;
        LED0_OFF();
        LED1_OFF();
        LED2_OFF();
        initValue(); //パラメータの初期化
    }else{ //PWM 出力を再開

```

```

        puts("SOFTSTART¥r¥n");
        state = SOFTSTART; //ソフトスタート開始
        LED0_ON();
        LED1_OFF();
        LED2_OFF();
        LED5_OFF();
    }
}

//-----
//IRQ6 割り込み処理内容
void intrIRQ6(void){
//PI          =3      PI モード          :PI 制御ルーチン
//TDOF         =4      2DOF モード       :2DOF 制御ルーチン
//PIPFC        =5      PIPFC モード      :PI の力率改善制御ルーチン
//TDOFPFC=6   TDOFPFC モード     :2DOF の力率改善制御ルーチン

    //使用する制御ルーチンの変更
    mtd++;
    if(mtd>6) mtd=3;
    switch( mtd ){
        case PI:
            Vtrg=CONST280;
            Ua=0;
            Uvi=0;
            puts("PI¥r¥n");
            break;

        case TDOF:
            puts("2DOF¥r¥n");
            break;

        case PIPFC:
            puts("PIPFC¥r¥n");
            break;

        case TDOFPFC:
//            Vtrg=CONST280;
//            Ua=1e7; //電圧積分器の初期値(これがないと立ち上がり時大電流)
//            Ua=9.86e6; // 電 圧 積 分 器 の 初 期 値 (AC100V
            Ua=100*sqrt(2)*Kvout*Kadc*K1v/Kiv*(-1);
            Ua=1.972e+7; // 電 圧 積 分 器 の 初 期 値 (AC200V
            Ua=200*sqrt(2)*Kvout*Kadc*K1v/Kiv*(-1);
            puts("2DOFPFC¥r¥n");
            break;
    } //switch(mtd)
}

//-----
//MTU2S 割り込み処理内容
void intrMTU2S(void){
    if(MTU2S4.TSR.BIT.TCFV == 1)
        MTU2S4.TSR.BIT.TCFV =0; //割り込みフラグの消去
        //何もない
}

//-----
//グローバル変数の初期化
void initValue(void){
    //パラメータすべて初期化
    mtd          = TDOF;
    puts("2DOF¥r¥n");
    cnt          = 1024;
    Vrv1        = 4;
    Vtrg        = CONST280;
    Vdif        = 0.0;
}

```

```

Itrg      = 0.0;
Idif      = 0.0;
Upwm      = 1.0;
//各状態パラメータ
Uv          = 0.0;
Uvi         = 0.0;
Uviz        = 0.0;
Ua          = 0.0;
Ua4         = 0.0;
Up3          = 0.0;
AnalogData.calc = 0;
}

//-----
void stepCount(void){
//目標値を 365-385 にする
if( Vtrg != TRUE_TARGET ){
    if( Vtrg != BEFORE_TARGET ){ //365 以下なら徐々に上げる
        if( cnt++ > STARTUP_TIME ){
            Vtrg++;
            cnt = 0;
        }
    }else{
        if( cnt++ > 200000 ){
            cnt = 0;
            Vtrg = TRUE_TARGET;
            LED2_ON();
        }
    }
}
}

//制御中の電流制限(インダクタ電流 10A 以上でデューティ 0.1%)
void currentLimiter(void){
    if(lac >= C_CONST15A){
        Upwm      = 1;
        LED5_ON();
        puts("CURRENT_LIMITTER");
    }
}

```

# 付録 C 数式の導出

## C.1 $W_{ry}$ の導出

$(z - n_1)(z - n_2)$  は、制御対象の元々の零点、 $(z - H_1)(z - H_2)(z - H_3)$  は極配置後の極である。

ここで、1次遅れ系  $\frac{1}{z - H_1}$  を考える。

連続システムの1次遅れ系  $G_c(s)$  は、直流ゲイン  $G_{dc}$  と時定数  $\tau_d$  を用いて

$$G_c(s) = \frac{G_{dc}}{\tau_d s + 1}$$

と表記することができる。この1次遅れ系のステップ応答は、ステップ入力のラプラス変換  $\frac{1}{s}$  をかけて、

$$G_{cstep}(s) = \frac{1}{s} \frac{G_{dc}}{\tau_d s + 1} = \frac{1}{s} \frac{\tau_d}{s + \frac{1}{\tau_d}}$$

となる。これをサンプリング周期  $T_s$  で

$$\frac{a}{s(s+a)} \Rightarrow \frac{(1-e^{-aT_s})z}{(z-1)(z-e^{-aT_s})}$$

の公式を使って  $z$  変換する。すると、

$$G_{cstep}(z) = \frac{z}{z-1} \frac{G_{dc}(1-e^{\frac{-1}{\tau_d}T_s})}{z-e^{\frac{-1}{\tau_d}T_s}}$$

となり、ステップ入力の  $z$  変換  $\frac{z}{z-1}$  を除くと、離散1次遅れ系のパルス伝達関数は

$$G_c(z) = \frac{G_{dc}(1-e^{\frac{-1}{\tau_d}T_s})}{z-e^{\frac{-1}{\tau_d}T_s}} = \frac{G_{dc}(1-H_1)}{z-H_1}$$

で表される事がわかる。ただし  $e^{\frac{-1}{\tau_d}T_s} = H_1$  である。

この式から、離散システムでは、極を  $H_1$  においていた場合、直流ゲインとして  $G_{dc}$  の他に  $(1-H_1)$  が必要である事がわかる。

全ての極・零点に対して、 $G_{dc}=1$  とし、同様の直流ゲインを追加すると、 $W_{ry}$  は以下の式となる。

$$W_{ry}(z) = \frac{(1-H_1)(1-H_2)(1-H_3)(z-n_1)(z-n_2)}{(z-H_1)(z-H_2)(z-H_3)(1-n_1)(1-n_2)}$$

## C.2 G の導出

まず、状態方程式

$$\begin{aligned} x_w(k+1) &= A_w x_w(k) + B_w v(k) \\ y(k) &= C_w x_w(k) \end{aligned}$$

の直流ゲインを求める。この状態方程式は、1 入力 1 出力の伝達関数に置き換えると、

$$G_w = C_w (zI - A_w)^{-1} B_w$$

となる。伝達関数の直流ゲインは、 $z=1$  の場合に相当するため、伝達関数  $G_w$  に  $z=1$  を代入すると、

$$G_{wdc} = C_w \frac{I}{I - A_w} B_w$$

が、先の状態方程式の直流ゲインとなる。

ここで、状態フィードバックは、状態フィードバックゲイン  $F$  を使って、系の  $A$  行列を  $(A-BF)$  に変えることで、 $A$  行列の固有値(極)を移動する手法であるから、状態フィードバックを適用した系の直流ゲインは、状態方程式の直流ゲインの  $A$  行列を  $A-BF$  したものとなる。よって、状態フィードバック  $v(k) = -Fx_w(k) + r$  を適用した系の直流ゲインは、

$$G_{wfdc} = C_w \frac{I}{I - (A_w - B_w F)} B_w$$

であり、これを打ち消して、直流ゲインを 1 とするためのゲイン  $G$  は、その逆数の

$$G = \frac{1}{G_{wfdc}} = \frac{1}{C_w \frac{I}{I - (A_w - B_w F)} B_w}$$

となる。

### C.3 $r \rightarrow y$ 伝達関数の導出

図 4.9 の  $r \rightarrow y$  の伝達関数を導出する。ここで、 $Q \rightarrow y$  の特性は関係ない為、 $Q=0$  とおく。この時、 $\lambda$  は、

$$\begin{aligned}\lambda &= r + K(z)(\lambda - W_m^{-1}y) \\ &= r + K(z)(\lambda - W_m^{-1}W_{ry}\lambda) \\ &= r + K(z)(\lambda - W_m^{-1}W_m W_s \lambda) \\ &= r + \lambda K(z)(1 - W_s)\end{aligned}$$

だから、

$$\begin{aligned}\lambda[1 - K(z)(1 - W_s)] &= r \\ \lambda \left( \frac{z-1+k_z - k_z + k_z W_s}{z-1+k_z} \right) &= r \\ \lambda = r \left( \frac{z-1+k_z}{z-1+k_z W_s} \right) &\end{aligned}$$

となる。ここで、 $y$  は  $y = W_{ry}\lambda$  だから、

$$\begin{aligned}y &= W_{ry} \frac{z-1+k_z}{z-1+k_z W_s} r \\ &= W_m \frac{z-1+k_z}{z-1+k_z W_s} W_s r\end{aligned}$$

となる。

## C.4 $Q \rightarrow y$ 伝達関数の導出

図 4.9 の  $Q \rightarrow y$  の伝達関数を導出する。ここで  $r \rightarrow y$  の特性は関係ないため、 $r=0$  とおく。  
この時  $\lambda$  は

$$\lambda = K(z)(\lambda - W_m^{-1}y) \text{ より}$$

$$\lambda(1 - K(z)) = -K(z)W_m^{-1}y$$

$$\lambda = \frac{-K(z)W_m^{-1}y}{1 - K(z)}$$

である。ここで、 $y$  を、図 7 から、

$$y = W_{ry}\lambda + W_{Qy}Q$$

とすると、これを式変形して、

$$\begin{aligned} W_{Qy}Q &= y - W_{ry}\lambda \\ &= y + \frac{W_{ry}K(z)W_m^{-1}y}{1 - K(z)} \\ &= y \left( 1 + \frac{W_s k_z}{1 - \frac{k_z}{z-1+k_z}} \right) \\ &= y \left( \frac{z-1}{z-1} + \frac{W_s k_z}{z-1} \right) \\ &= y \left( \frac{z-1 + W_s k_z}{z-1} \right) \end{aligned}$$

となり、

$$\begin{aligned} y &= W_{Qy}Q \frac{z-1}{z-1 + W_s k_z} \\ &= \frac{z-1}{z-1 + k_z} \frac{z-1 + k_z}{z-1 + W_s k_z} W_{Qy}Q \end{aligned}$$

となる。

# 付録 D 実験装置と回路図

## D.1 マイクロコントローラ SH7216 と昇圧コンバータボードの接続

### 回路

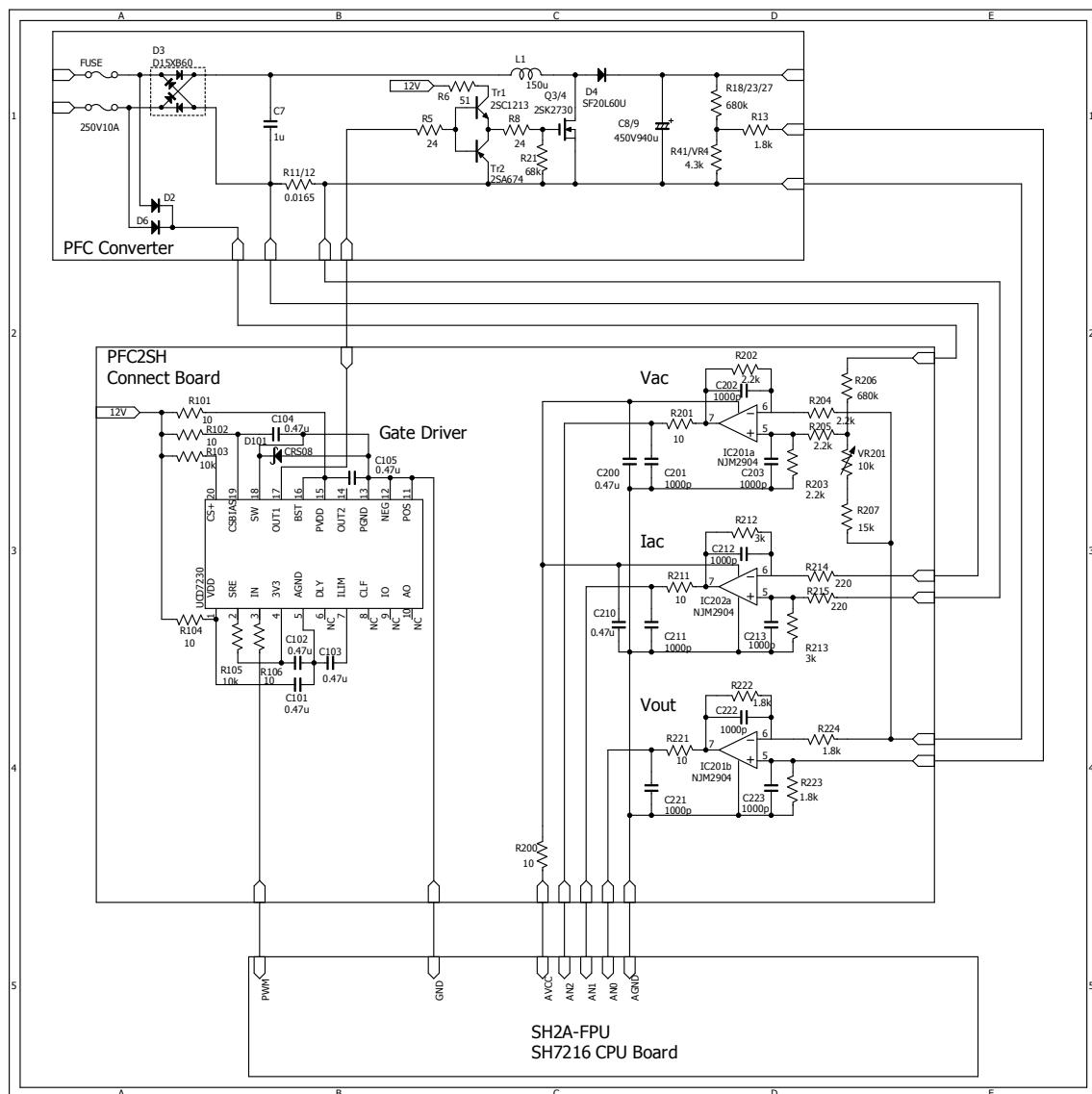


図 D.1 マイクロコントローラと昇圧コンバータの接続回路

## D.2 実験装置の外観

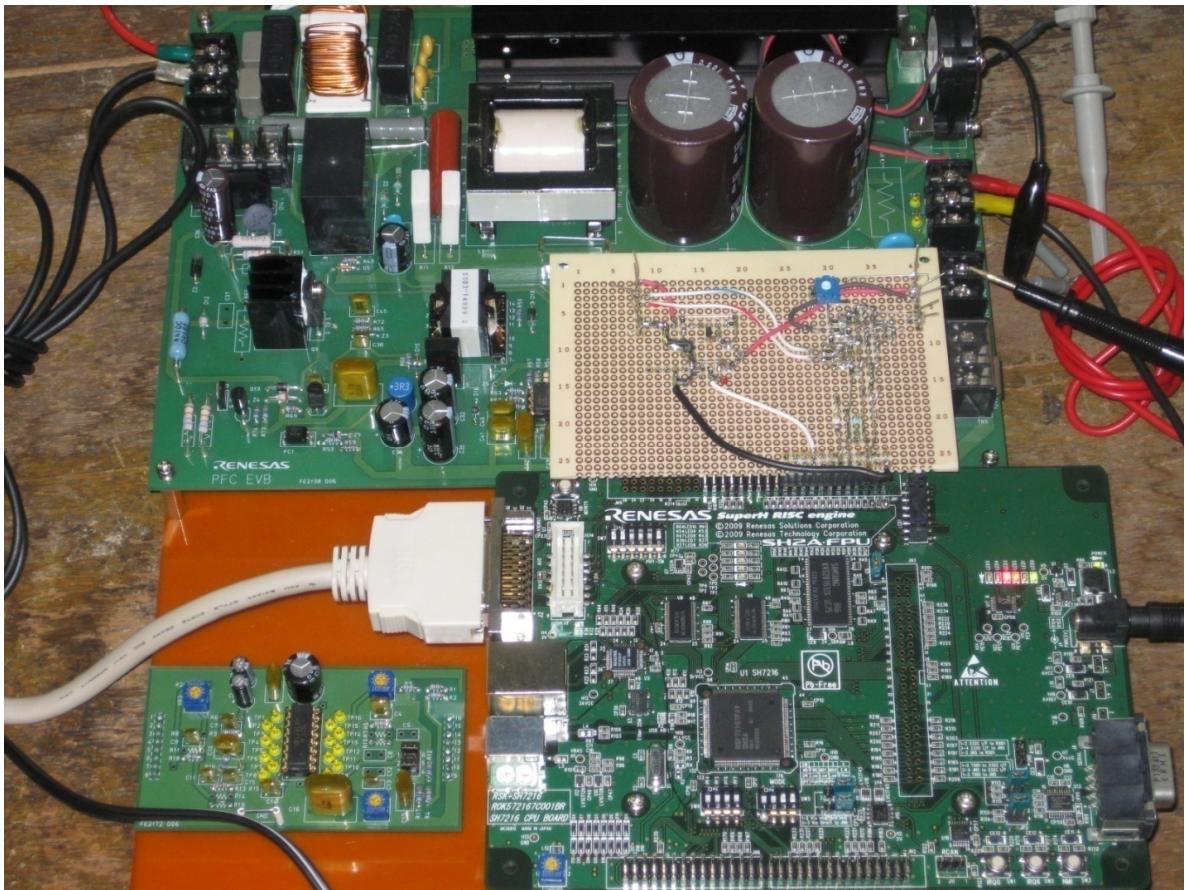


図 D.2 マイクロコントローラと PFC 昇圧コンバータ・接続回路の外観

# 図目次

1.1 機器の入力電圧と入力電流波形の例	• • • 4
1.2 PFC機能付きAC-DCコンバータのブロック図	• • • 5
2.1 PFC昇圧型 DC-DCコンバータの構成	• • • 9
2.2 制御対象の回路モデル	• • • 11
2.3 本研究で用いたPFC昇圧型コンバータの静特性	• • • 12
2.4 本研究で用いたPFC昇圧型コンバータの各動作点におけるステップ応答	• • • 13
2.5 本研究で用いたPFC昇圧型コンバータのボード線図	• • • 14
3.1 マイクロコントローラ(SH7216)が搭載された基板	• • • 18
3.2 キャリア波と PWM出力の関係図	• • • 19
3.3 制御ルーチンの例	• • • 20
3.4 PFC昇圧型コンバータ基板の構成図	• • • 21
4.1 フィードバック制御系	• • • 23
4.2 等価外乱を含むフィードバック制御系	• • • 23
4.3 出力電圧のみをフィードバックする昇圧型コンバータ	• • • 24
4.4 入力無駄時間 $L_d$ と1サンプル周期遅れを持つ制御対象	• • • 25
4.5 離散時間制御対象に状態フィードバックを備えたモデルマッチングシステム	• • • 26
4.6 電流フィードバックを他の状態フィードバックに置き換えたモデル	• • • 27
4.7 ブロック線図の等価変換	• • • 28
4.8 電圧フィードバックのみを用いたモデルマッチングシステム	• • • 28
4.9 逆システムとフィルタを結合したロバスト補償器	• • • 29
4.10 等価変換されたロバスト補償器	• • • 31
4.11 近似的2自由度ディジタル積分形制御系	• • • 31
4.12 根軌跡	• • • 33
4.13 近似的2自由度ディジタル積分形制御系の MATLAB/SIMULINKモデル	• • • 35
4.14 近似的2自由度ディジタル積分形制御系の Step応答シミュレーション	• • • 36
4.15 図 4.13の制御器の $r \rightarrow y$ 間ボード線図	• • • 36
4.16 図 4.13の制御器の 開ボード線図	• • • 37
4.17 図 4.13の制御器の $q_y \rightarrow y$ 間ボード線図	• • • 37
4.18 図 4.13の制御器の $q_v \rightarrow y$ 間ボード線図	• • • 38
4.19 SimPowerSystemsを用いた昇圧コンバータの負荷急変シミュレーションモデル	• • • 39

4.20 SimPowerSystemsを用いた昇圧コンバータの負荷急変シミュレーション結果	• • • 40
4.21 近似的2自由度ディジタル積分形制御系の各動作点のステップ応答結果	• • • 41
4.22 近似的2自由度ディジタル積分形制御系の負荷急変応答結果	• • • 41
4.23 PI制御器の MATLAB/SIMULINKシミュレーションモデル	• • • 42
4.24 PI制御器の各動作点のステップ応答結果	• • • 43
4.25 PI制御器の負荷急変応答結果	• • • 43
 5.1 一般的な力率改善制御器の構成	• • • 44
5.2 近似的2自由度制御を力率改善制御に適用する際の問題点	• • • 45
5.3 入力無駄時間 $L_d$ を持つ制御対象	• • • 47
5.4 モデルマッチングシステム	• • • 48
5.5 逆システムとフィルタを結合したロバスト補償器	• • • 49
5.6 近似的2自由度ディジタル積分形インダクタ電流制御器	• • • 50
5.7 乗算器を附加した電流制御器	• • • 51
5.8 出力電圧を制御量とした新たな制御対象のモデルとその近似表現	• • • 52
5.9 ロバスト電圧補償器	• • • 53
5.10 力率改善をする近似的2自由度ディジタル積分形出力電圧制御器	• • • 53
5.11 電流制御器の根軌跡の例	• • • 56
5.12 電圧制御器の根軌跡の例	• • • 57
5.13 電流制御器のMATLAB/SIMULINKモデル	• • • 60
5.14 図5.13のモデルのステップ応答	• • • 61
5.15 図5.13のモデルの閉ループボード線図	• • • 61
5.16 図5.13のモデルの開ループボード線図	• • • 62
5.17 図5.13のモデルの $q_y \rightarrow y$ ボード線図	• • • 62
5.18 図5.13のモデルの $q_v \rightarrow y$ ボード線図	• • • 63
5.19 力率改善電圧制御器のMATLAB/SIMULINKモデル	• • • 64
5.20 図5.19のモデルのステップ応答	• • • 65
5.21 図5.19のモデルの $r_v \rightarrow y_v$ 閉ループボード線図	• • • 65
5.22 図5.19のモデルの $rv \rightarrow y$ 閉ループボード線図	• • • 66
5.23 図5.19のモデルの開ループボード線図	• • • 66
5.24 図5.19のモデルの $q \rightarrow y$ ボード線図	• • • 67
5.25 図5.19のモデルの $qv \rightarrow yv$ ボード線図	• • • 67
5.26 図5.19のモデルの $ri \rightarrow yi$ ボード線図	• • • 68
5.27 力率改善制御器のSimPowerSystemsモデル	• • • 69
5.28 図5.26のモデルでSimPowerSystemsを用いた定常状態力率改善シミュレーション結果	• 70
5.29 図5.26のモデルでSimPowerSystemsを用いた負荷急変応答結果	• • • 70

5.30 近似的2自由度力率改善制御器の各動作点の定常状態波形	• • • 71
5.31 近似的2自由度力率改善制御器の各動作点のステップ応答結果	• • • 72
5.32 近似的2自由度力率改善制御器の各動作点の立ち上がり波形	• • • 73
5.33 近似的2自由度力率改善制御器の負荷急変応答結果	• • • 74
5.34 近似的2自由度力率改善制御器の効率特性	• • • 75
5.35 近似的2自由度力率改善制御器の力率特性	• • • 75
5.36 近似的2自由度力率改善制御器の各動作点の高調波電流特性	• • • 76
5.37 ディジタルPI力率改善制御器のMATLAB/SIMULINKモデル	• • • 77
5.38 アナログ力率改善制御器の各動作点の定常状態波形	• • • 78
5.39 アナログ力率改善制御器の負荷急変応答結果	• • • 78
5.40 アナログ力率改善制御器の力率特性	• • • 79
5.41 アナログ力率改善制御器の各動作点の高調波電流特性	• • • 79
5.42 ディジタルPI力率改善制御器の各動作点のステップ応答結果	• • • 80
5.43 ディジタルPI力率改善制御器の各動作点の定常状態波形	• • • 81
5.44 ディジタルPI力率改善制御器の負荷急変応答結果	• • • 82
5.45 ディジタルPI力率改善制御器の力率特性	• • • 82
5.46 ディジタルPI力率改善制御器の各動作点の高調波電流特性	• • • 83
5.47 ディジタルPI力率改善制御器の構成	• • • 85
5.48 近似的2自由度力率改善制御器の構成	• • • 85
5.49 力率改善制御器の入力電圧急変シミュレーションを行うSimPowerSystemsモデル	• • • 87
5.50 近似的2自由度力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$ 無し)	• • • 88
5.51 近似的2自由度力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$ 有り)	• • • 89
5.52 ディジタルPI力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$ 無し)	• • • 90
5.53 ディジタルPI力率改善制御器の入力電圧急変シミュレーション結果( $v_{avg}$ 有り)	• • • 91
D.1 マイクロコントローラと昇圧コンバータの接続回路	• • • 121
D.2 マイクロコントローラとPFC昇圧コンバータ・接続回路の外観	• • • 122