

平成 2 1 年度 卒業論文

DC-DC コンバータの
SH マイコンによる高度デジタル制御

学 籍 番 号	0 6 1 2 1 5 7
氏 名	太田 恵大
電子工学科	電子システム工学講座
指導教官	樋口幸治 准教授
提 出 日	平成 2 2 年 1 月 2 9 日

電気通信大学 電気通信学部 電子工学科

概要

近年, 社会の IT 化が進む中で, 急速に広まっている情報通信ネットワーク機器や高速サーバなどにおいて, 広範囲な負荷変動に対して過渡応答特性の劣化しないロバストな DC-DC コンバータが必要とされている.

ロバストであるとは,

- ・系を安定させた上で, 制御量を目標値に精度良く追従または一致させること
- ・外乱の悪影響を除去抑制すること
- ・制御対象内のパラメータ変動や負荷変動に対して, 共に不感であること

を満たすことを言う.

本研究では, 近似的 2 自由度デジタル積分形制御器を用いて DC-DC コンバータの特性, とくに, スタートアップ特性と負荷急変応答特性を改善することを目的とする. このデジタル制御器は, 制御帯域幅をより広くし, 同時に出力電圧の変動を, 負荷抵抗の急激な変化に対して十分小さく抑えることのできる, ロバストなものである. この制御器の設計法では, スタートアップ特性の目標モデルに 1 次近似モデルを用い, この一次近似に対する外乱を制御対象のパラメータ変動とみなしている. また, この設計法では, 外乱と出力間の伝達関数に任意の零点を設定できるため, 近似 2 次微分特性を実現することが出来る. これにより, 外乱から出力間でのゲインは大幅に小さくなり, 負荷抵抗の変動に対して出力電圧の変動を小さくすることが出来る.

本研究では, スタートアップ特性の目標モデルに用いた 1 次近似モデルの十分な近似と, 外乱から出力電圧への近似 2 次微分特性を得るための制御器の設計法を提案する. 提案された方法によって得られたデジタル制御器のシミュレーションを行い, 負荷抵抗の変動およびスタートアップ時の出力電圧の動作を確認し, 与えられた仕様を満たす適切な制御パラメータを得る. また, このデジタル制御器および制御パラメータを SH マイコンに実装する. そして, 製作したシステムの動作実験を行い, 負荷変動に対して出力電圧変動を小さく抑えたまま, スタートアップ特性を改善できることを検証している.

目次

概要.....	i
目次.....	ii
第1章 序論.....	1
第2章 デジタル制御系における制御器の設計.....	2
2.1 状態方程式の離散化.....	2
2.2 制御対象の伝達関数と遅れ時間の考慮.....	3
第3章 DC-DC コンバータ.....	5
3.1 DC-DC コンバータの基本構成.....	5
3.2 制御対象のモデル化.....	7
第4章 零点を付加した近似的二自由度制御系の設計法.....	9
4.1 零点の付加方法.....	9
4.2 近似的二自由度制御系の設計.....	12
第5章 デジタルコントローラの製作.....	14
5.1 SH マイコンの概要.....	14
5.2 演算時間について.....	14
5.3 デジタルコントローラのアロリズムの導出.....	15
第6章 実験方法とシミュレーション.....	18
6.1 実験システムと実験方法.....	18
6.2 シミュレーションとパラメータの決定.....	19
第7章 実験結果と考察.....	21
第8章 結論.....	23
謝辞.....	24
参考文献.....	25
付録A MATLAB シミュレーションプログラム.....	26
付録B SH マイコンメインプログラム.....	30
付録C 入力電圧を変化させたときのスタートアップ応答結果.....	37
付録D DC-DC コンバータボードおよびその回路図.....	38
付録E 卒業研究発表会で挙げられた質問とその回答.....	40

第1章 序論

DC-DC コンバータの多くの応用では、負荷は前もって特定することが出来ない。負荷の大きさは 0 から最大定格まで急に變更される。従来用いられてきた制御法では、個々の負荷に対して設計条件を変えて個々の制御器を設計し、制御系を構成して対応している。そこで、このような広範な負荷変動に対して、も一個の制御器で対応できるいわゆるロバストな DC-DC コンバータが必要とされている。

通常は、DC-DC コンバータの制御には、簡単な積分制御を実行するアナログ制御 IC が使用されている。DC-DC コンバータの需要が高まりつつある近年では、PID や根軌跡法などによって設計されたデジタル制御器の DC-DC コンバータへの適用が検討されている。しかしながら、これらの技術によって DC-DC コンバータの十分なロバスト性を達成することは困難である。

本研究では、制御帯域幅をより広くし、同時に出力電圧の変動を抵抗負荷の急変に対して十分小さく抑える、ロバストな近似的 2 自由度デジタル制御器の設計法を提案する。またこの制御器を PWM を搭載する SH マイコン(SH7216)に実装して実験を行い、理想状態のシミュレーションと比較・検討する。

本設計法では、スタートアップ特性の目標モデルに 1 次近似モデルを用いており、この一次近似に対する外乱を制御対象のパラメータ変動とみなしている。そして、この設計法では、外乱と出力間の伝達関数に任意の零点を設定できるため、近似 2 次微分特性を実現することが出来る。これにより、外乱から出力間でのゲインは大幅に小さくなり、負荷抵抗の変動に対して出力電圧の変動を小さくすることが出来る。

本設計法における、1 次近似モデルを実現する 2 自由度制御器は以下のように構成される。まず、外乱からのフィードフォワードと状態フィードバックからなるモデルマッチング系を、出力電圧フィードバックのみを用いた系に等価変換する。次に、モデルマッチング系を 1 次近似モデルに誘導する。そして、1 次近似モデルの逆システムおよび逆システムを実現するためのフィルタをモデルマッチング系に結合する。最後に、制御器を等価変換し、1 次近似モデルを実現する 2 自由度デジタル制御器を得る。

このモデルの動作を MATLAB によるシミュレーションによって確認する。そして、シミュレーションによって仕様を満たすパラメータを決定する。またデジタル制御器として SH マイコンを用いたシステムを製作する。システムにはシミュレーションによって得られた制御アルゴリズムと制御パラメータを実装する。そして、製作したシステムにおいて実験を行い、負荷急変による出力電圧変動を小さく抑えたまま、スタートアップ応答特性を改善できたことを確認する。

第2章 デジタル制御系における制御器の設計

2.1 状態方程式の離散化

連続時間における1入力システムを表現する状態方程式は以下のように表される.

$$\dot{x}(t) = A_c x(t) + b_c u(t) + b_c q_u \quad (2.1)$$

$$y(t) = C_c x(t) + q_y \quad (2.2)$$

このシステムの応答は, 次の式で表される.

$$x(t) = e^{A_c(t-t_0)} x(t_0) + \int_0^t e^{A_c(t-\tau)} b_c u(\tau) d\tau \quad (2.3)$$

$$e^{At} = I + At + \frac{A^2 t^2}{2!} + A + \frac{A^n t^n}{n!} + A \quad (2.4)$$

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.

デジタル制御では, システム全体を離散時間システムとして表現する必要がある. そこでシステムの入力として0次ホールド(各サンプル時点の値をサンプル周期だけ保持する)を用い, システムを離散化することを考える. システムの入力は

$$u(t) = u(kT), \quad kT \leq t \leq (k+1)T \quad (2.5)$$

となるので, $t = (k+1)T$ における状態 $x[(k+1)T]$ は直前の $x(kT)$ を用いて

$$x[(k+1)T] = e^{A_c T} x(kT) + \int_{kT}^{(k+1)T} e^{A_c[(k+1)T-\tau]} b_c u(kT) d\tau \quad (2.6)$$

で与えられる.

ここで $\eta = (k+1)T - \tau$ と変数変換を行うと $d\eta = -d\tau$ となるから,

$$x[(k+1)T] = e^{A_c T} x(kT) + \int_{kT}^{(k+1)T} e^{A_c \eta} b_c u(kT) d\eta \quad (2.7)$$

となる.

よって離散時間におけるシステム方程式は次のように記述される.

$$\dot{x}(k+1) = Ax(k) + bu(k) + bu(k) + bq_u \quad (2.8)$$

$$y(k) = C_c x(k) + q_y \quad (2.9)$$

ただし,

$$A = e^{A_c T} = I + A_c T + \frac{A_c^2 T^2}{2!} + A + \frac{A_c^n T^n}{n!} + A \quad (2.10)$$

$$b = \int_0^T e^{A_c \eta} b_c d\eta = \left[I + \frac{A_c T}{2!} + A + \frac{A_c^n T^n}{n!} + A \right] T b_c \quad (2.11)$$

である.

2.2 制御対象の伝達関数と遅れ時間の考慮

制御対象の伝達関数は連続系で与えられるため、これを離散系のパルス伝達関数に変換しなければならない。そこで、連続系のシステム方程式のラプラス変換を取り、

$$X(s) = (sI - A_c)^{-1} x(0) + (sI - A_c)^{-1} b_c U(s) \quad (2.12)$$

$$Y(s) = C_c X(s) \quad (2.13)$$

$x(0)=0$ とすれば、伝達関数 $G(s)$ は、

$$G(s) = \frac{Y(s)}{X(s)} = C_c (sI - A_c)^{-1} b_c \quad (2.14)$$

一方、離散系のシステム方程式は、

$$X(z) = (zI - A)^{-1} x(0) + (zI - A)^{-1} b U(z) \quad (2.15)$$

$$Y(z) = C X(z) \quad (2.16)$$

であらわされる。 $x(0)=0$ とすれば、伝達関数 $G(z)$ は、

$$G(z) = \frac{Y(z)}{X(z)} = C (zI - A)^{-1} b \quad (2.17)$$

となる。

ディジタル制御において、サンプリング開始から操作量が出力されるまでに一定の遅れ時間 L が存在する。そこで、この遅れ時間を制御対象に存在する入力無駄時間と等価であると考え、このとき、遅れた一つの状態を導入すると、システム方程式は次のように表される。

$$\begin{bmatrix} x(k+1) \\ \xi(k+1) \end{bmatrix} = \begin{bmatrix} e^{AT} & \int_{T-L}^T e^{A\eta} b d\eta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \xi(k) \end{bmatrix} + \begin{bmatrix} \int_0^{T-L} e^{A\eta} b d\eta \\ 1 \end{bmatrix} v(k) \quad (2.18)$$

$$y(k) = \begin{bmatrix} C_c & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \xi(k) \end{bmatrix} \quad (2.19)$$

よって得られた式も差分方程式をなしているので、 $\begin{bmatrix} x & \xi \end{bmatrix}^T = \mathbf{x}'$ と置き換えることにより、式(2.18)(2.19)は次のように書き換えられる。

$$\mathbf{x}'(k+1) = A' \mathbf{x}(k) B' v(k) \quad (2.20)$$

$$y(k) = C' \mathbf{x}'(k) \quad (2.21)$$

これより, パルス伝達関数は,

$$G(z) = C'(zI - A)^{-1}B' \quad (2.22)$$

となる.

ただし

$$A' = \begin{bmatrix} e^{AT} & \int_{T-L}^T e^{A\eta} b d\eta \\ 0 & 0 \end{bmatrix} \quad (2.23)$$

$$B' = \begin{bmatrix} \int_0^{T-L} e^{A\eta} b d\eta \\ 1 \end{bmatrix} \quad (2.24)$$

$$C' = [C_c \quad 0] \quad (2.25)$$

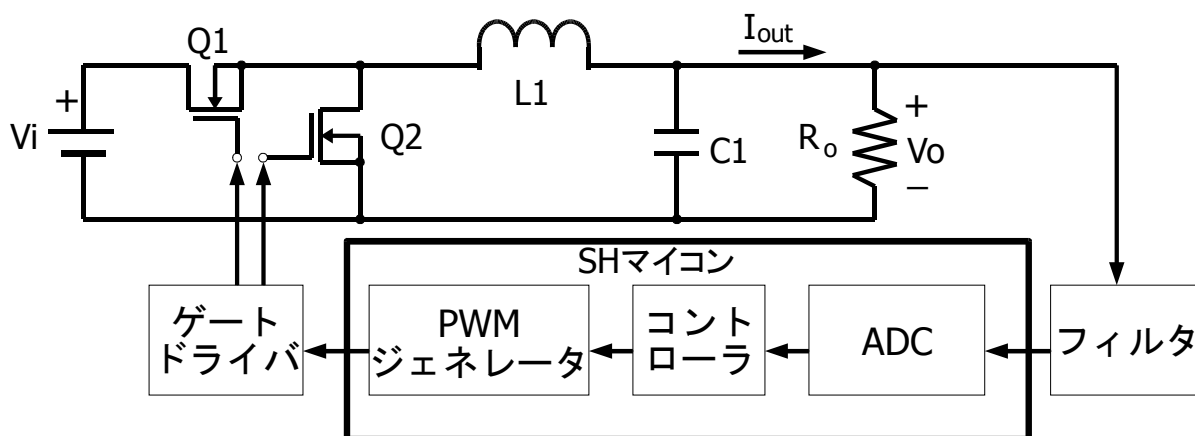
である.

上式の表現により, 演算時間による遅れ L は入力無駄時間として正確に表現されるため, DSC による応答は指定した特性のものとなり, 演算時間の悪影響を取り除くことが出来る.

第3章 DC-DCコンバータ

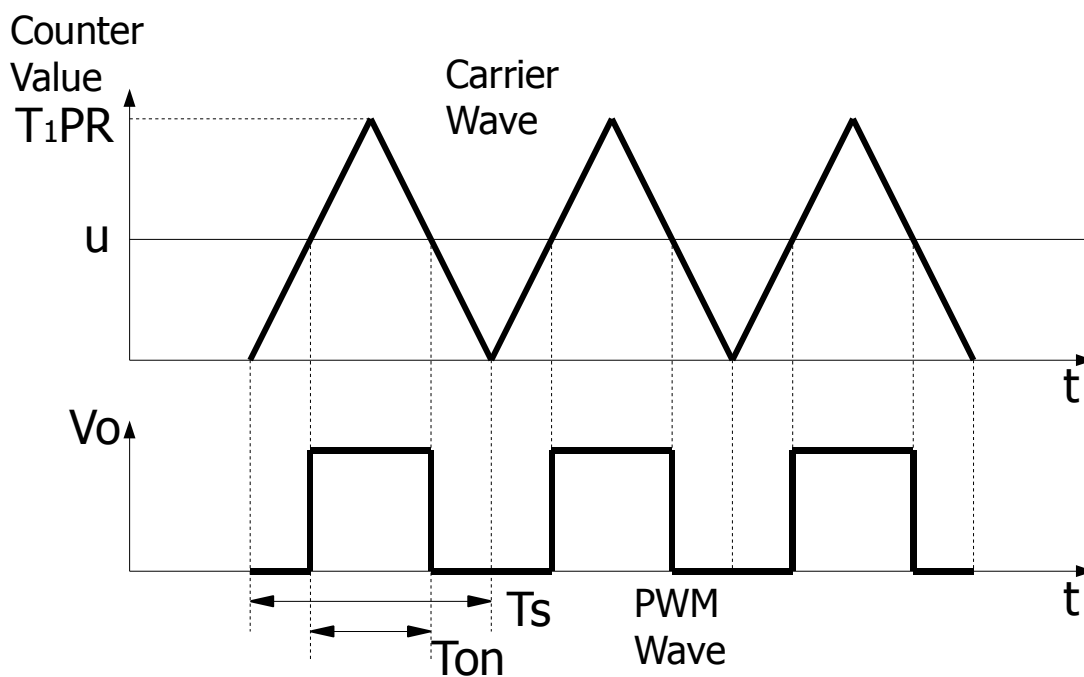
3.1 DC-DCコンバータの基本構成

図 3.1 に PWM 搭載 SH マイコン(SH7216)を用いた DC-DC コンバータの基本構成を示す.



本研究では、降圧型の DC-DC コンバータとし、入力電圧 9.6[V], 出力電圧 3.3[V]とした. 図の LCR 回路は、キャリアやスイッチングノイズを除去するためのフィルタであり、 $L=0.5[\mu\text{H}]$, $C=235[\mu\text{F}]$ とした.

今回使用した DC-DC コンバータでは、キャリア波として連続アップダウンカウントモードを用い、図 3.2 に示すようにキャリア波が操作入力よりも上にあるときに出力する変調方式を採用している. 尚、キャリア波の周波数は 450[kHz]とした.



ここで、FET からLCフィルタに入力される電圧について考える。まず、スイッチによってフィルタに入力される電圧 V_{in} について考える。入力電圧を V_i とし、2つの FET によってスイッチングされると、電圧 V_{in} は方形波状となる。そこで T_s と T_{on} の比を求める。図 3.2 より、三角形の相似から

$$\frac{T_{on}}{T_s} = \frac{T_1 PR - u}{T_1 PR} \quad (3.1)$$

となる。

ここで、カウンタ値 $T_1 PR$ は、サンプリング周期 2.22[us]、カウンタクロックの刻み幅 $T_{dk} = 10[\text{ns}]$ より、

$$T_1 PR = \frac{T_s}{2} / T_{dk} = \frac{2.22\mu}{2 \times 10\text{n}} = 111 \quad (3.2)$$

ゆえに、フィルタに入力される電圧の平均値は、

$$V_{in} = \frac{T_{on}}{T_s} V_i = -0.086 \cdot (u - 111) \quad (3.3)$$

となり、 $K_p = -0.086$ とおくことで、系のゲインとする。

また、ADC の変換ゲイン K_{adc} は、ADC の電源電圧 5[V] を 12 ビットで分割しているので、

$$K_{adc} = 2^{12} / 5 = 819.2 \quad (3.4)$$

となる。

さらに、ADC 前のフィルタによるゲイン K_{filter} は、OP アンプに接続した抵抗から、

$$K_{filter} = \frac{2.2\text{k}}{2\text{k}} = 1.1 \quad (3.5)$$

である。

3.2 制御対象のモデル化

図 3.1 で示された制御対象のコンバータ部をモデル化すると、図 3.3 のようになる。

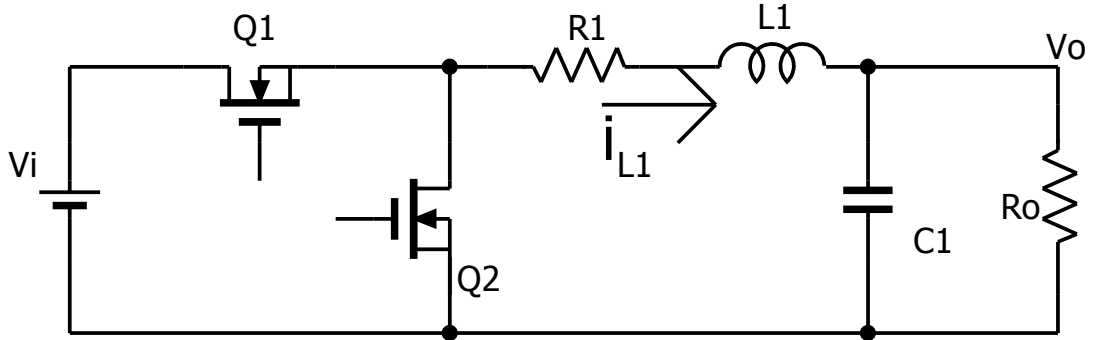


図 3.3:コンバータ部のモデル

図 3.3 での無負荷時の状態方程式は、

$$\dot{x} = Ax + Bu \quad (3.6)$$

$$y = Cx \quad (3.7)$$

$$x = \begin{bmatrix} V_o \\ i_{L1} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & \frac{1}{C1} \\ \frac{-1}{L1} & \frac{-R1}{L1} \end{bmatrix} \quad (3.8)$$

$$B = \begin{bmatrix} 0 \\ \frac{K_{pwm} V_{in}}{L1} \end{bmatrix}, \quad C = [1 \quad 0] \quad (3.9)$$

となる。ここで、R1 はコイルやFET のオン抵抗などの合成抵抗である。

また、抵抗負荷時はシステム行列が、

$$A = \begin{bmatrix} \frac{-1}{R_o C_1} & \frac{1}{C1} \\ \frac{-1}{L1} & \frac{-R1}{L1} \end{bmatrix} \quad (3.10)$$

となる。

また、抵抗および容量負荷時は、システム行列は、

$$A = \begin{bmatrix} \frac{-1}{R_o(C1+C0)} & \frac{1}{(C1+C0)} \\ \frac{-1}{L1} & \frac{-R1}{L1} \end{bmatrix} \quad (3.10)$$

となる。

このとき、演算時間遅れを考慮し制御対象を離散化すると、

$$A_d = \begin{bmatrix} e^{AT} & \int_{T-L}^T e^{AT} b d\eta \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & 0 \end{bmatrix} \quad (3.11)$$

$$B_d = \begin{bmatrix} \int_0^{T-L} e^{AT} b d\eta \\ 0 \end{bmatrix} = \begin{bmatrix} B_{11} \\ B_{21} \\ 1 \end{bmatrix} \quad (3.12)$$

となる。

制御対象の負荷の変動は、制御対象の A 行列の $A(1,1)$ と $A(1,2)$ 要素のパラメータ変化とみなすことが出来る。このパラメータ変動は、離散化すると、 A_d と B_d のパラメータ変動になることがわかる。ここで、図 3.4 において A_d のパラメータ変動は等価外乱 q_u に、 B_d のパラメータ変動は q_y に置き換えることが出来る。従って、これらのパラメータ変動、すなわち負荷変動の影響を抑える、つまりロバストにするには、等価外乱 q_u 、 q_y から y への伝達関数になるべく小さくなるような制御系を構成すればよい。目標特性を保持したまま、これらの等価外乱の影響を抑制できる設計法を次に提案する。

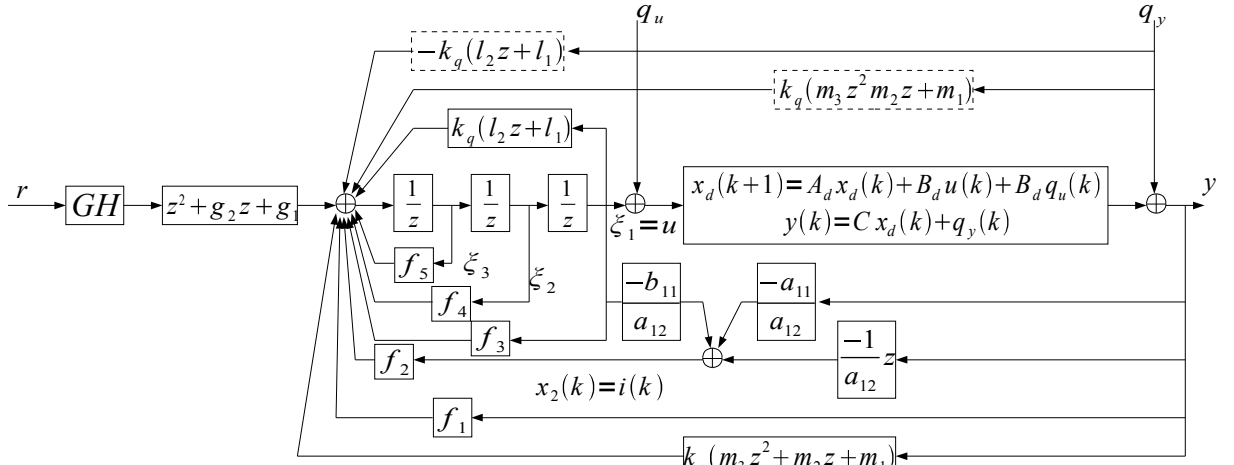


図 3.4: 負荷変動に対する等価外乱と状態フィードバックのみを用いたモデルマッチング

第4章 零点を付加した近似的二自由度制御系の設計法

4.1 零点の付加方法

式(2.8)の出力より次式を得る.

$$Y = O^* x_d(k) + U \bar{u}(k) + U \bar{q}_u + \bar{q}_y \quad (4.1)$$

ただし,

$$Y = \begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix} O^* = \begin{bmatrix} C \\ C A_d \\ C A_d^2 \end{bmatrix} U = \begin{bmatrix} 0 & 0 \\ C B_d & 0 \\ C A_d B_d & C B_d \end{bmatrix} \quad (4.2)$$

$$\bar{u} = \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \bar{q}_u = \begin{bmatrix} \bar{q}_u(k) \\ \bar{q}_u(k+1) \end{bmatrix} \bar{q}_y = \begin{bmatrix} \bar{q}_y(k) \\ \bar{q}_y(k+1) \\ \bar{q}_y(k+2) \end{bmatrix} \quad (4.3)$$

である.

式(4.1)の両辺に左側から \bar{I}_2 を乗じ, x_d について解くことにより次式が得られる.

$$x_d(k) = (\bar{I}_2 O^*)^{-1} \bar{I}_2 Y - (\bar{I}_2 O^*)^{-1} \bar{I}_2 U \bar{u}(k) - (\bar{I}_2 O^*)^{-1} \bar{I}_2 U \bar{q}_u(k) - (\bar{I}_2 O^*)^{-1} \bar{I}_2 \bar{q}_y \quad (4.4)$$

ただし,

$$\bar{I}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.5)$$

である.

これを式(4.1)に代入することにより, 次式を得る.

$$\begin{aligned} & (I_3 - O^* (\bar{I}_2 O^*)^{-1} \bar{I}_2) U \bar{q}_u + (I_3 - O^* (\bar{I}_2 O^*)^{-1} \bar{I}_2) q_y \\ & = (I_3 - O^* (\bar{I}_2 O^*)^{-1} \bar{I}_2) Y + (I_3 - O^* (\bar{I}_2 O^*)^{-1} \bar{I}_2) U \bar{u} \end{aligned} \quad (4.6)$$

ここで, \bar{I}_3 は 3×3 の単位行列である. これによって q_u, q_y は Y と $\bar{u}(k)$ に置き換えることができ, 式(4.6)は式(4.7)に書き換えられる.

$$\begin{aligned} & -(l_2 z + l_1) \bar{q}_u(k) + (z^2 + m_2 z + m_1) q_y(k) \\ & = (l_2 z + l_1) u(k) + (z^2 + m_2 z + m_1) y(k) \end{aligned} \quad (4.7)$$

ただし

$$\begin{aligned} [l_1 \quad l_2] &= -(I_3 - O^* (\bar{I}_2 O^*)^{-1} \bar{I}_2) U \\ [m_1 \quad m_2 \quad m_3] &= (I_3 - O^* (\bar{I}_2 O^*)^{-1} \bar{I}_2) \end{aligned} \quad (4.8)$$

である.

制御対象に対して無駄時間, 電流推定, 零点の付加を考慮したシステムを次式のように表す.

$$x_{dw}(k+1) = A_{dw} x_{dw}(k) + B_{dw} v(k) \quad (4.9)$$

$$y(k) = C_{dw} x_{dw}(k) \quad (4.10)$$

ただし,

$$x_{dw}(k) = \begin{bmatrix} x_d(k) \\ \xi_1(k) \\ \xi_2(k) \\ \xi_3(k) \end{bmatrix} \quad A_{dw} = \begin{bmatrix} A_d & B_d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B_{dw} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.11)$$

$$C_{dw} = [C \quad 0 \quad 0 \quad 0] \quad \xi_1(k) = u(k) \quad (4.12)$$

である.

式(4.3)で表されるシステムに q_u, q_y, r からのフィードフォワードと状態フィードバックを適用することにより, 図 3.4 でしめされるシステムを得ることが出来る. これをあらわすと

$$\begin{aligned} v(k) = & -k_q(l_2 z + l_1) q_u(k) + k_q(z^2 + m_2 z + m_1) q_y(k) \\ & + (z^2 + g_2 z + g_1) r(k) + [f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6] x_{dw}(k) \end{aligned} \quad (4.13)$$

となる.

図 3.4 において破線で示されている部分が, \bar{q}_u, \bar{q}_y のフィードフォワードの係数である. 式(4.7)により式(4.13)は次のように書き換えられる.

$$\begin{aligned} v(k) = & k_q(l_2 z + l_1) u(k) + k_q(z^2 + m_2 z + m_1) y(k) \\ & + (z^2 + g_2 z + g_1) r(k) + [f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6] x_{dw}(k) \end{aligned} \quad (4.14)$$

これによってフィードフォワード部は図 4.1 で示される図に等価変換することが出来る. 図 4.1 から次式を得る.

$$ff_1 = f_1 - f_2(a_{11}/a_{12}) + k_q m_1 + f_5(k_q m_2 + f_2/a_{12}) + f_5^2 k_q m_3 + (f_4 + k_q l_2) k_q m_3 \quad (4.15)$$

$$ff_2 = k_q m_2 + f_2/a_{12} + f_5 k_q m_3 \quad (4.16)$$

$$ff_3 = k_q m_2 + f_2/a_{12} + f_5 k_q m_3 \quad (4.17)$$

$$ff_4 = f_4 + k_q l_2 \quad (4.18)$$

$$k_5 = f_5 \quad (4.19)$$

$$k_6 = k_q m_3 \quad (4.20)$$

また, 図 4.1 において $r-y, q_u-y, q_y-y$ 間の伝達関数はそれぞれ次のように表される.

$$W_{ry}(z) = N_{ry}(z)/D(z) \quad (4.21)$$

$$W_{quy}(z) = N_{quy}(z)/D(z) \quad (4.22)$$

$$WW_{qyy}(z) = N_{qyy}(z)/D(z) \quad (4.23)$$

ただし,

$$N_{ry}(z) = GH(z^2 + glz + g0)(b_{11}z + b_{21}a_{12} - a_{22}b_{11}) \quad (4.24)$$

$$W_{quy}(z) = N_{qz} N_p \quad (4.25)$$

$$W_{qy}(z) = N_{qz} D_p \quad (4.26)$$

$$N_{qz} = a_{12}z^3 - a_{12}f_5z^2 + (a_{12}b_{11}k_q - a_{12}f_4)z - f_3a_{12}f_2b_{11} - a_{12}a_{22}b_{11}k_q + a_{12}^2b_{21}k_q \quad (4.27)$$

$$\begin{aligned}
D(z) = & z^5 + (-f_5 - a_{22} - a_{11})z^4 + (a_{11}f_5 + a_{11}a_{22} - a_{21}a_{12} + a_{22}f_5 - f_4)z^3 \\
& + (a_{21}a_{12}f_5 - f_3 + a_{11}f_4 - a_{11}a_{22}f_5 + a_{22}f_4)z^2 \\
& + (a_{22}f_3 + a_{21}a_{12}f_4 + a_{11}f_3 - b_{11}f_1 - f_2b_{21} - a_{11}a_{22}f_4)z \\
& + f_2a_{11}b_{21} - a_{21}f_2b_{11} + a_{21}a_{12}f_3 + f_1a_{22}b_{11} - f_1a_{12}b_{21} - a_{11}a_{22}f_3
\end{aligned} \tag{4.28}$$

である.

$D(z)$ により, 全体の極は $f1, f2, f3, f4, f5$ となる. $N_{ry}(z)$ により g_0 と g_1 によつて 2 つの零点が任意に設定できる. $N_{qz}(s)$ と $N_{qy}(s)$ に共通な Nq_z によつて $q_u - y$ と $q_y - y$ 間の一つの零点が k_q によつて同じ値に任意に設定できる. つまり, W_{qy} に任意の零点を一つ加えることが出来る.

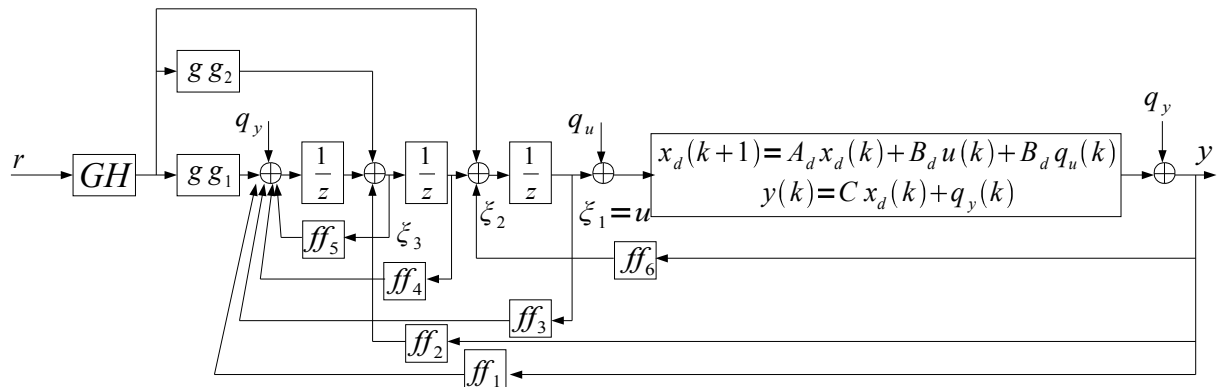


図 4.1:出力電圧フィードバックのみを用いたモデルマッチングシステム

4.2 近似的二自由度制御系の設計

ここで、目標値 r ，制御量 y 間の伝達関数 W_{ry} を

$$W_{ry} = \frac{(1+H_1)(1+H_2)(1+H_3)(z-n_1)(z-n_2)(z+H_4)(z+H_5)}{(1-n_1)(1-n_2)(z+H_1)(z+H_2)(z+H_3)(z+H_4)(z+H_5)} \quad (4.29)$$

と指定する. ただし, n_1 と n_2 は制御対象の零点である. ここで q_u , q_y から y への伝達関数は次のよう
にあらわされる.

$$W_{q_u y}(z) = (z-1) \bar{N}_{qz} N_p / D_H(z) \quad (4.30)$$

$$W_{q_y y}(z) = (z-1) \bar{N}_{qz} N_p / D_H(z) \quad (4.31)$$

$N_{qz}(1)=0$ の解により k_q によって零点を 1 と指定すると k_q は次のように表される.

$$k_q = \frac{-a_{12} + a_{12}f_5 + a_{12}f_4 + a_{12}f_3 - b_{11}f_2}{a_{12}(b_{11} - a_{22}b_{11} + b_{21}a_{12})} \quad (4.32)$$

ここで, $|H_2| \gg |Re(H_1)|$, $|H_2| \gg |Re(H_3)|$ とすることにより, $W_{ry}(z)$ は次の一次モデルに近似で
きる.

$$W_{ry}(z) \approx W_m(z) = \frac{(1+H_2)}{(z+H_2)} \quad (4.33)$$

これに対してインバースシステムとフィルタを付加した図 4.2 の系を構成する. $K(z)$ は次式で表される.

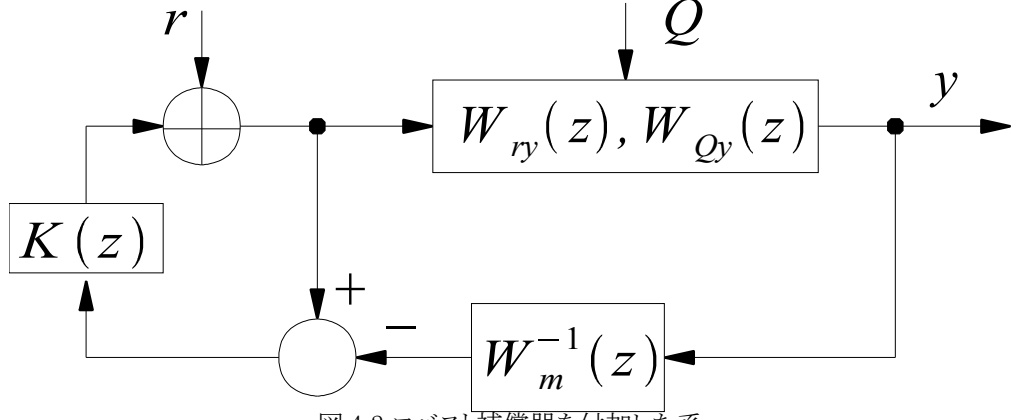


図 4.2: ロバスト補償器を付加した系

$$K(z) = \frac{k_z}{z-1+k_z} \quad (4.34)$$

$r-y$, q_u-y , q_y-y 間の伝達関数はそれぞれ次のよう
にあらわされる.

$$y = \frac{1+H_2}{z+H_2} \frac{z-1+k_z}{z-1+k_z} W_s(z) r \quad (4.35)$$

$$y = \frac{(z-1)^2}{z-1+k_z} \frac{z-1+k_z}{z-1+k_z} \frac{\bar{N}_{qz} N_p}{D_H(z)} q_u y \quad (4.36)$$

$$y = \frac{(z-1)^2}{z-1+k_z} \frac{z-1+k_z}{z-1+k_z} \frac{\bar{N}_{qz} N_p}{D_H(z)} q_y y \quad (4.37)$$

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

ここで $W_s(z) \approx 1$ となるようにすれば, 式はそれぞれ次のように表される.

$$y = \frac{(z-1)^2}{z-1+k_{\bar{z}}} \frac{\bar{N}_{qz} N_p}{D_H(z)} q_u y \quad (4.40)$$

$$y = \frac{(z-1)^2}{z-1+k_\tau} \frac{\bar{N}_{qz} N_p}{D_H(z)} q_y y \quad (4.41)$$

13

第5章 デジタルコントローラの製作

5.1 SH マイコンの概要

マイコン(マイクロコントローラ:Micro Controller)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, それらの制御に利用される.

SH マイコンはルネサステクノロジ社の 32bit 組み込み機器向けシングルチップ型 RISC マイコンである. 本研究では, コントローラ系 SH マイコン 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, SH2A-FPU(SH7216)を使用した. SH7216 は動作周波数 200[MHz]であり 400 [MFLOPS]の演算性能を持つ. また, 周辺機能として, PWM タイマや 12bit A/D 変換器を内蔵している. このマイコンに, コンピュータからエミュレータを通してプログラムを書き込む.

また, この SH マイコンは, コントローラ系ながら十分高速な処理速度を持ち, 家電製品などの制御と並行して, コンバータ制御を行うことも可能である. 一方, 内部処理が高速であるために周辺回路も十分高速である必要がある. そのため, SH マイコンに接続する IC の応答速度や遅延時間を考慮した設計が必要であることに注意する.

5.2 演算時間について

スイッチング電源の制御には, その基本的な構造として, 零次ホールドのサンプリング特性を持っている. 一般的にアナログ制御 IC は, サンプル周期よりも遅れ時間が十分小さいため, 遅れ時間を考慮しなくても良い. これに対し, デジタル制御は演算遅れが発生し, 高域の特性が悪化するため, 演算遅れに対する補償をする必要がある.

本研究で使用する SH マイコンでは, 下図のような演算時間や A/D 変換・PWM 出力の時間遅れが存在する. マイコンの演算遅れは, 等価的に入力遅れとみなすことができ, これを考慮したフィードバックを適用する. 下図より, 演算遅れは $0.99 T_s$ とした. なお, サンプル周期 T_s は $2.22[\mu s]$ である. このサンプリング周期の決定には, SH マイコンに内蔵される PWM タイマを利用している.

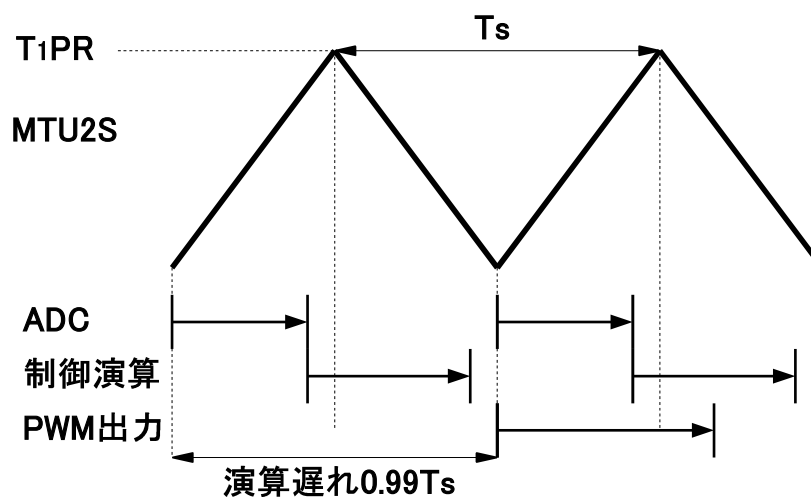


図 5.1:SH マイコンのタイムチャート

5.3 デジタルコントローラのアルゴリズムの導出

デジタルコントローラ的设计に関しては、MATLAB というソフトウェアを使って、シミュレーションを行い適切なパラメータを设计する。

デジタル制御器の実行においては、演算時間や A/D・D/A 変換による時間遅れが問題となる。この遅れは制御対象の離散化と同時に、無駄時間として考慮されているが、できるだけ小さくするために、制御器アルゴリズムを工夫する。

まず、図 4.1 より導出した MATLAB によるブロック線図を図 5.2 に示す。パラメータ変動の無いブロックをまとめることにより演算時間を短縮した。図 5.2 の制御器に用いたパラメータはそれぞれ次のように指定した。

$$\begin{aligned}
 k_1 = & f_1 - \frac{f_2 a_{11}}{a_{12}} + k(-a_{11}^2 - a_{21} a_{12} + \frac{(a_{12} a_{11} + a_{12} a_{22}) a_{11}}{a_{12}}) \\
 & + f_5(-k)(\frac{a_{12} a_{11} + a_{12} a_{22}}{a_{12}} + \frac{f_2}{a_{12}}) + f_5^2 k + (f_4 - k b_{11}) k \\
 & - GHK_z \frac{g_0 + (g_1 + f_5) f_5 + f_4 - k b_{11}}{1 + H_2}
 \end{aligned} \tag{5.1}$$

$$k_2 = k - \frac{GHK_z}{1 + H_2} \tag{5.2}$$

$$k_3 = -k \frac{a_{11} a_{12} + a_{12} a_{22}}{a_{12}} + \frac{f_2}{a_{12}} + f_5 k - \frac{(g_1 - f_5) GHK_z}{1 + H_2} \tag{5.3}$$

$$k_4 = f_3 - \frac{f_2 b_{11}}{a_{12}} + k(\frac{(a_{12} a_{11} + a_{12} a_{22}) b_{11}}{a_{12}} - a_{11} b_{11} - b_{21} a_{12}) \tag{5.4}$$

$$k_5 = f_4 - k b_{11} \tag{5.5}$$

$$k_6 = f_5 \tag{5.6}$$

$$k_{i1} = (g_0 + (g_1 + f_5) f_5 + f_4 - k b_{11}) GHK_z \tag{5.7}$$

$$k_{i2} = (g_1 + f_5) GHK_z \tag{5.8}$$

$$k_{i3} = GHK_z \tag{5.9}$$

$$k_{1r} = GH \tag{5.10}$$

$$k_{2r} = (g_1 + f_5) GH \tag{5.11}$$

$$k_{3r} = (g_0 + (g_1 + f_5) f_5 + f_4 - k b_{11}) GH \tag{5.12}$$

さらに図 5.2 は A/D 変換によるゲイン K_{adc} を考え、等価変換されている。これによって得られるパラメータは次のようになる。

$$\begin{array}{lll} K1=k1/Kadc & K2=k3/Kadc & K3=k2/Kadc \\ K4=k4 & K5=k5 & K6=k6 \\ K1r=k1r/Kadc & K2r=k2r/Kadc & K3r=k3r/Kadc \\ Ki1=ki1/Kadc & Ki2=ki2/ki1 & Ki3=ki3/ki1 \end{array}$$

図 5.2 より制御アルゴリズムが得られる。これを SH マイコンに実装するために、これを C 言語で記述したものを次に示す。この演算をサンプリング周期に対応した割り込み時間ごとに行う。

$$\begin{aligned} U_b &= U_a + U_{v2} + K2 * V_{s_B} \\ U_a &= U_{v3} + U_{c1} + U_{b1} + U_{a1} + K1 * V_{s_B} \\ U_v &= U_v + E_B \\ U_{c1} &= K4 * U_c \\ U_{b1} &= K5 * U_b \\ U_{a1} &= K6 * U_a \\ U_{v1} &= Ki1 * U_v \\ U_{v2} &= Ki2 * U_v \\ U_{v3} &= Ki3 * U_v \\ V_{s_B} &= \text{ADC 結果} \\ E_B &= \text{Target_B} - V_{s_B} \\ U_c &= U_b + K3 * V_{s_B} + U_{v1} \end{aligned}$$

上記の計算部分のプログラムの詳細は、巻末に付録として載せた。

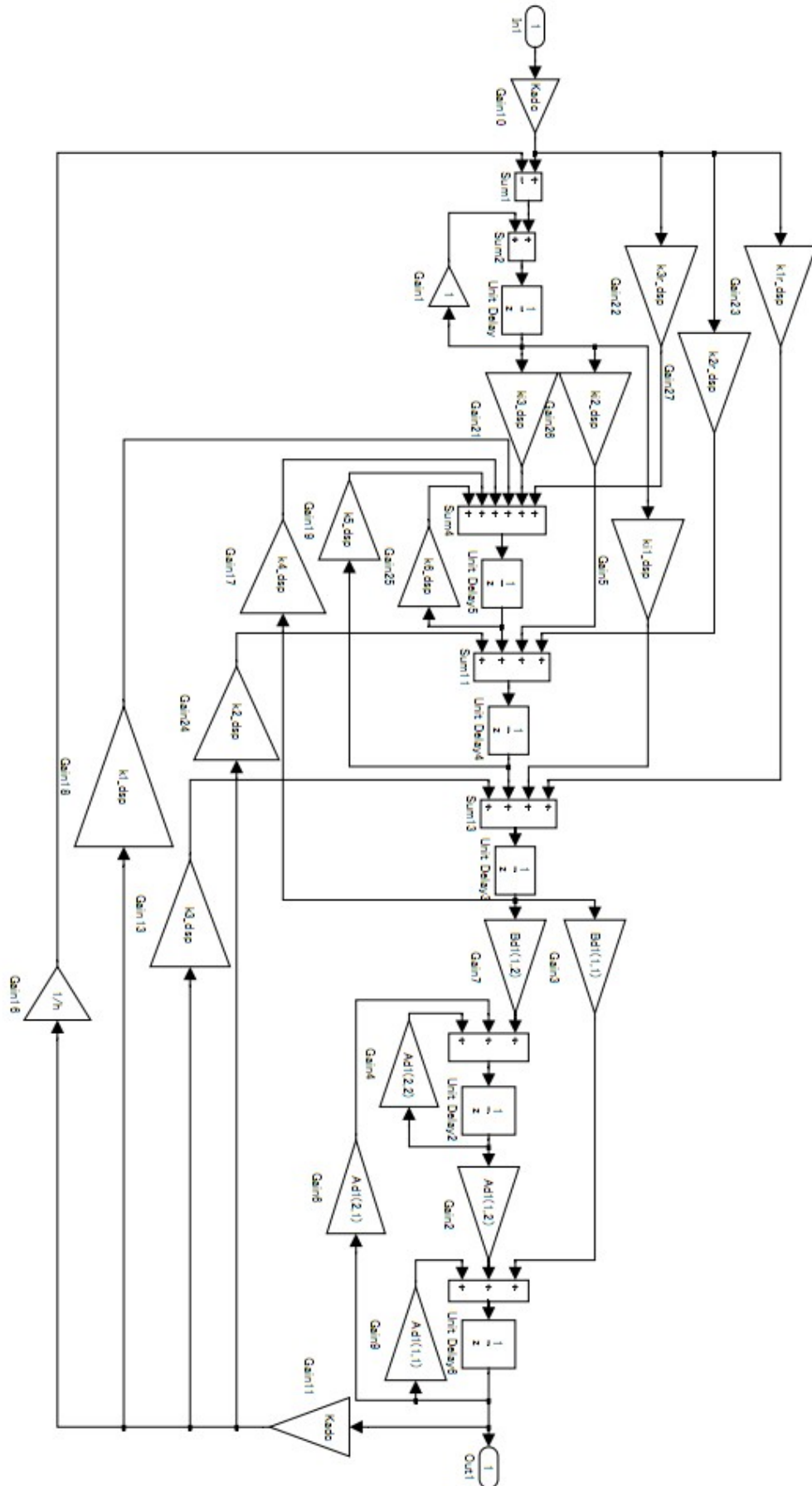


図 5.2:制御対象とコントローラのブロック線図(Kadc 付加等価変換後)

第 6 章 実験方法とシミュレーション

6.1 実験システムと実験方法

制御対象に対して5章で求めた制御アルゴリズムと制御パラメータを実現するシステムを図 6.1 に示す.

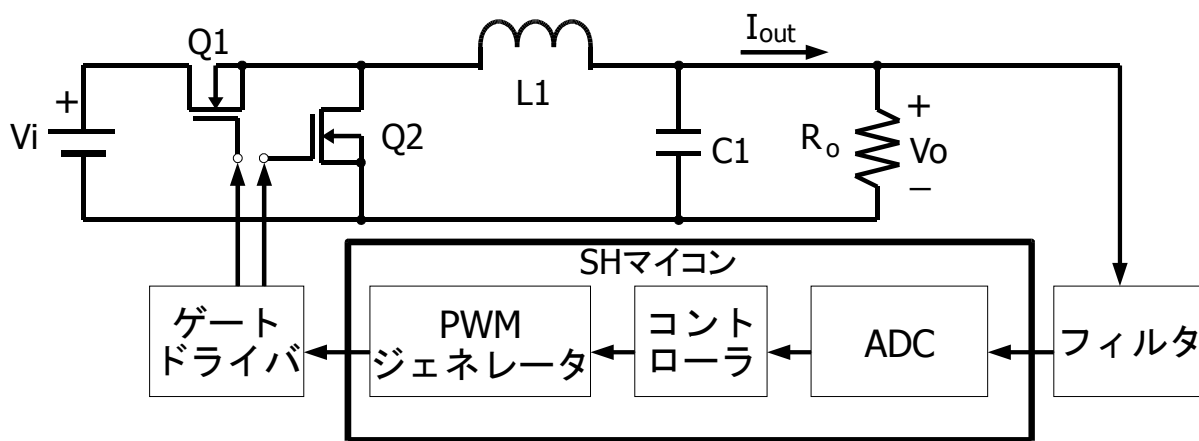


図 6.1: 実験システム

実験方法

スタートアップ応答実験では、コンバータを立ち上げ動作させたときの出力電圧の過渡状態の電圧波形をオシロスコープで測定する。

負荷急変実験では、定常状態時に、電子負荷を用いて出力電流を 10[A] から 20[A] (等価的に負荷抵抗が $0.33[\Omega] \Leftrightarrow 0.165[\Omega]$) に変化するように設定し、出力電圧と出力電流の波形を測定する。

6.2 シミュレーションとパラメータの決定

デジタルコントローラの設計にあたり, 必要なパラメータの設計は MATLAB によって行う. 次に MATLAB を用いた設計の手順を説明する.

MATLAB の Simulink を用いて作成した図 5.2 のブロック線図を, MATLAB のプログラムに使用し, それに設計パラメータを入力してシミュレーションを行う. 設計パラメータの値を変化させながらシミュレーションを繰り返し, 適切な制御器パラメータを得る.

図 6.2, 6.3 に MATLAB による立ち上がり応答・負荷急変応答のシミュレーション結果を示す. 立ち上がり時間は約 $45[\mu s]$, 負荷急変時の出力電圧変動は約 $[20mV]$ となっている.

シミュレーションで使用した設計パラメータ, およびそこからシミュレーションによって得られる制御器のパラメータを示す. 実験ではこの制御器パラメータの値を実装する.

〈設計パラメータ〉

$$\begin{array}{lll} H1=-0.1+0.5i & H2=-0.885 & H3=-0.1-0.5i \\ H4=-0.3+0.3i & H5=-0.3-0.3i & \\ kz=0.17 & kq=-1 & \end{array}$$

〈制御器パラメータ〉

$$\begin{array}{lll} K1=1.204541e+000 & K2=-1.520982e+000 & K3=8.436175e-001 \\ K4=8.491034e-001 & K5=3.733202e-001 & K6=-2.224235e-001 \\ Ki1=-7.282576e-003 & Ki2=5.989362e-003 & Ki3=-5.361771e-003 \end{array}$$

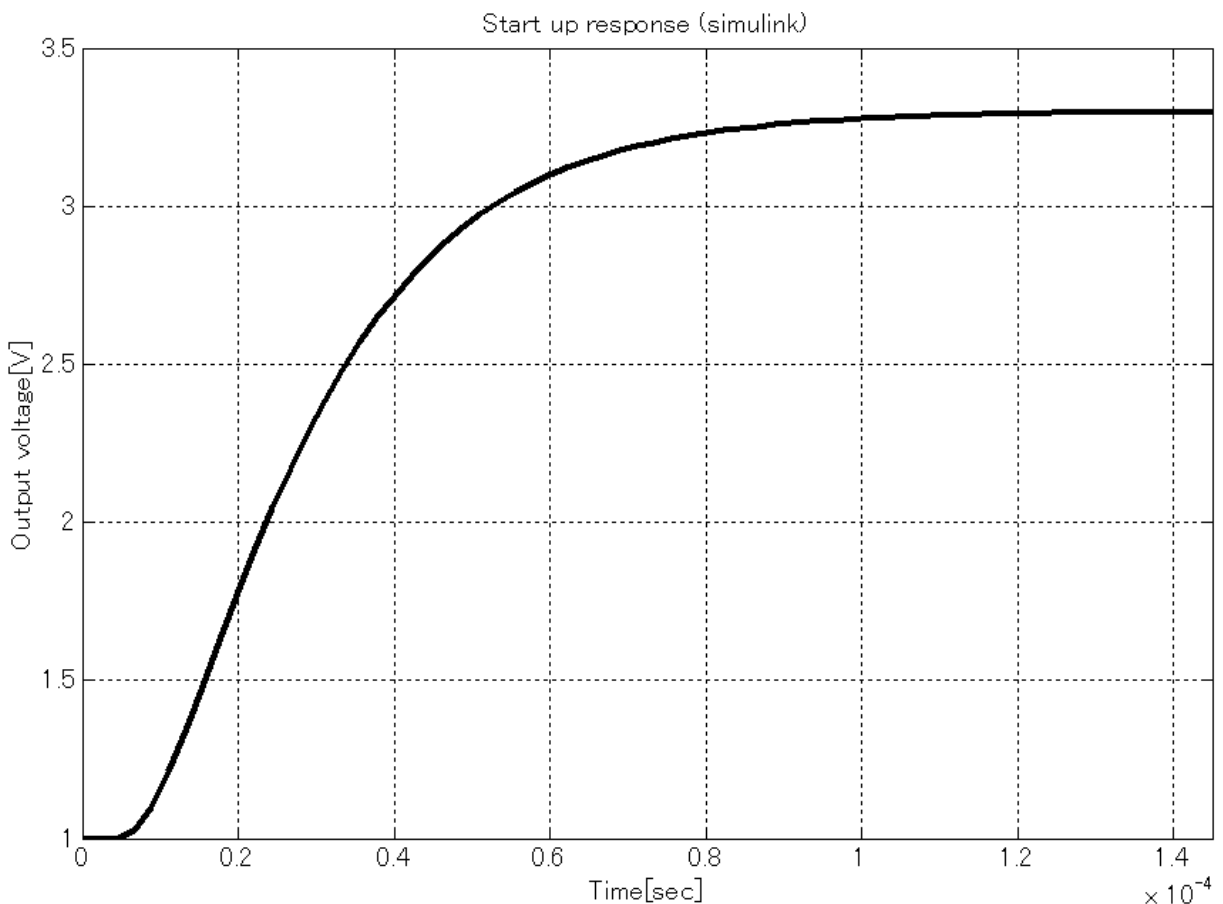


図 6.2: 立ち上がり応答のシミュレーション結果

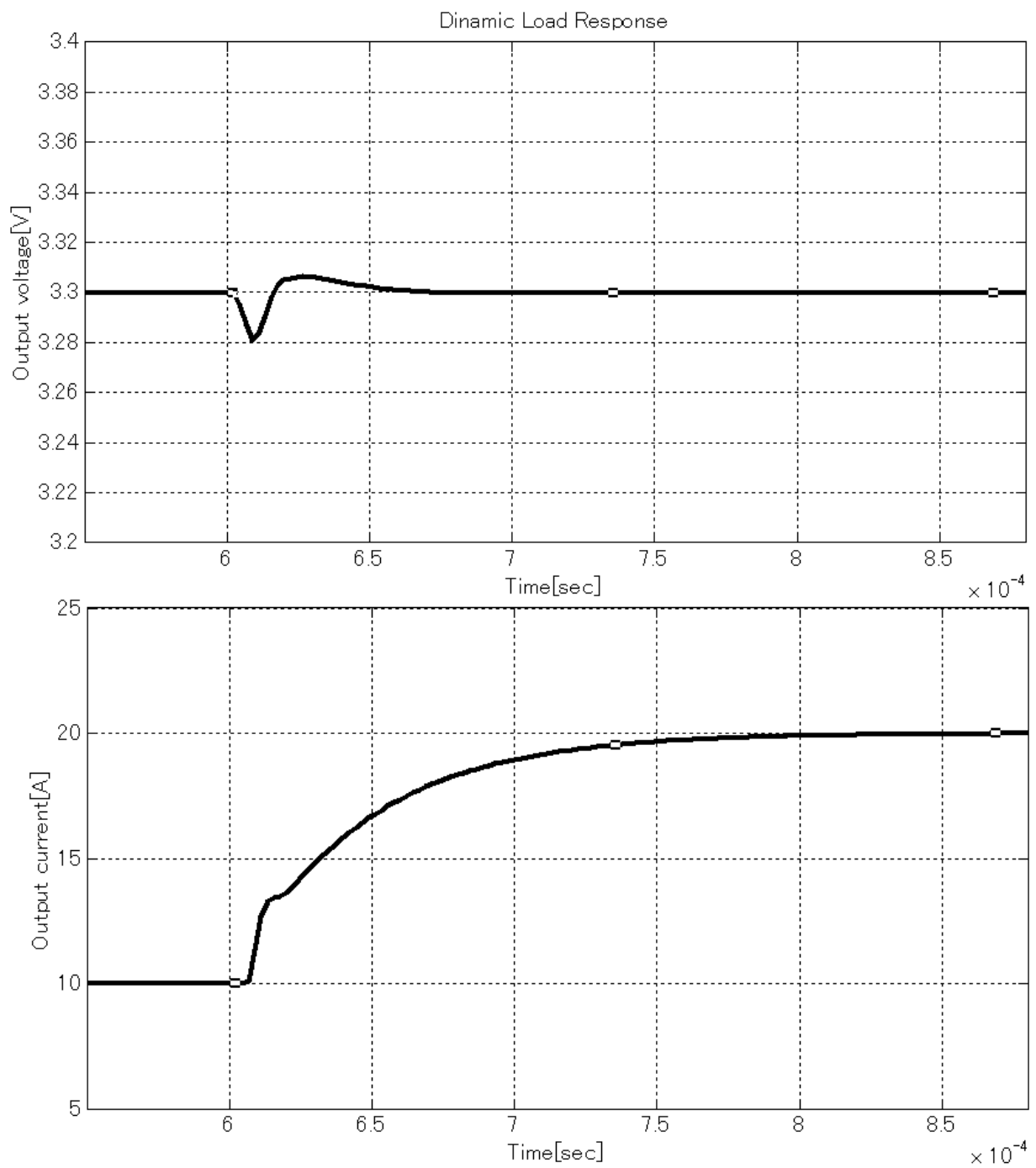


図 6.3: 負荷急変応答(10A→20A)のシミュレーション結果

第7章 実験結果と考察

制御アルゴリズムと制御器パラメータを SH マイコンに実装したときのスタートアップ応答の実験結果を図 7.1～7.3 に示す. すべて目標値は 3.3[V]で, 入力電圧は 9.6[V]である. 図 7.1 は無負荷時, 7.2 は抵抗負荷 $R=0.33[\Omega]$ 時, 7.3 は抵抗負荷 $R=0.165[\Omega]$ 時の結果である.

これらの出力電圧波形はシミュレーションの結果とほぼ同様の結果を得られており, また, 仕様を満たしている.

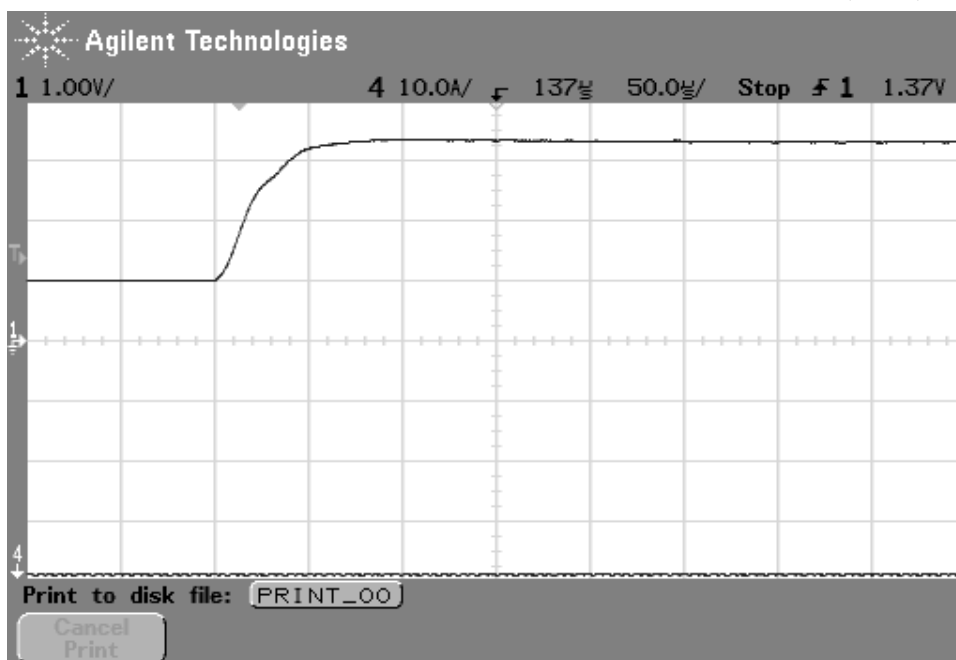


図 7.1:立ち上がり応答の実験結果(無負荷)

縦軸: Ch1=出力電圧(1[V/div])

横軸: 時間(40[us/div])

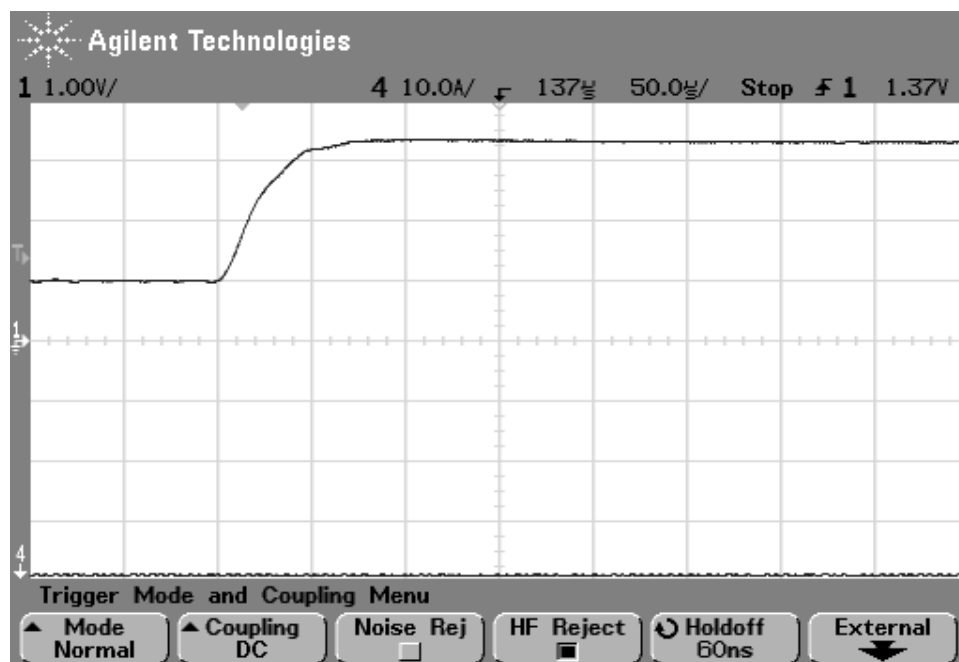


図 7.2:立ち上がり応答の実験結果(0.33[Ω]負荷)

縦軸: Ch1=出力電圧(1[V/div])

横軸: 時間(40[us/div])

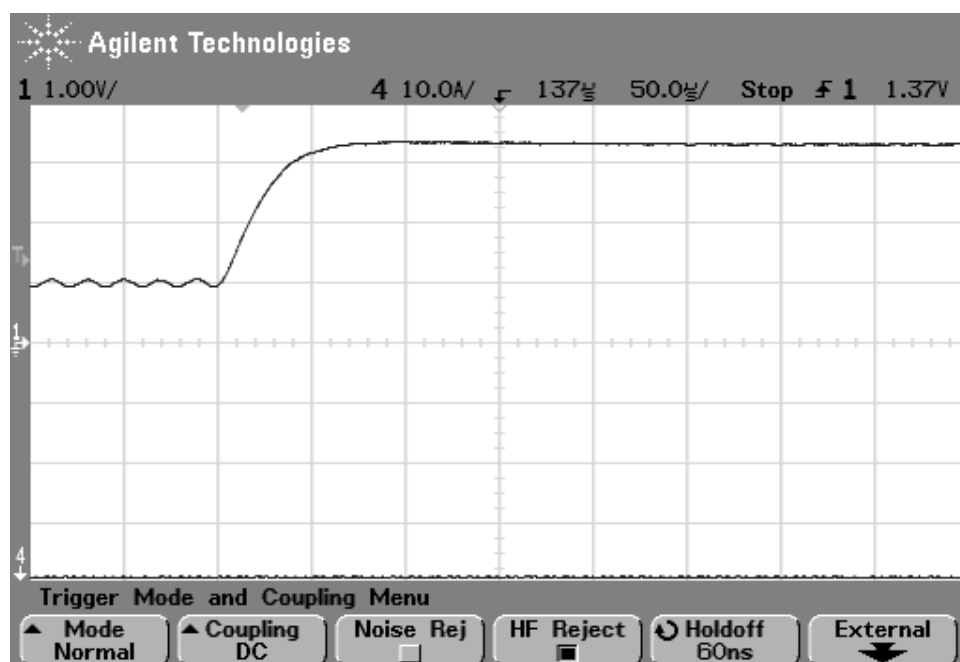


図 7.1:立ち上がり応答の実験結果(0.165[Ω]負荷)

縦軸:Ch1=出力電圧(1[V/div])

横軸:時間(40[us/div])

また、このほかに、負荷抵抗の値を $R=\infty[\Omega]$, $0.33[\Omega]$, $0.165[\Omega]$ のときにおいて入力電圧を $V_{in}=7.6[V]$, $11.6[V]$ と変化させた場合も、シミュレーションとほぼ同様の結果が得られた。

次に、負荷急変の実験結果を示す。抵抗負荷を電子負荷を用いて 1[kHz]おきに $0.33[\Omega] \Leftrightarrow 0.165[\Omega]$ と切り替えた。抵抗を切り替えたときの出力電圧の変動は 30[mV]以内に抑えることができたが、シミュレーションより大きい値となった。これは、新規に制作した H/W の不具合が原因と考えられる。

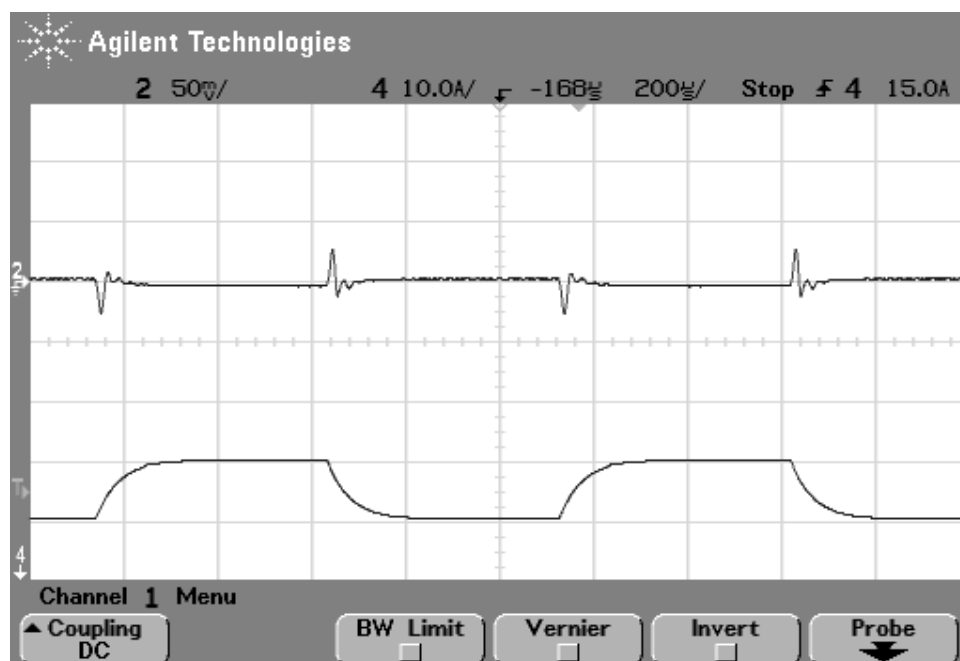


図 7.4:負荷急変応答の実験結果(負荷 $R_0=0.33\Omega \Leftrightarrow 0.165\Omega$)

縦軸:Ch1=出力電圧(50[mV/div])

Ch2=出力電流(10[A/div])

横軸:時間(200[us/div])

第 8 章 結論

本研究では, 出力電圧の変動を, 抵抗負荷の変化に対してより小さくするべく, 近似的二自由度制御器を設計し, これを実現する方法として状態フィードバックからなるモデルマッチング系を, 一次近似を用いて実現する積分形制御器に変換する方法を示した.

そして, このモデルの動作をMATLABによるシミュレーションによって確認し, スタートアップ応答が従来よりも改善されていること, 負荷変動時における出力電圧の変動が十分小さくなっていることを確認した.

またデジタル制御器として PWM 搭載 SH マイコン(SH7216)を用いたシステムを製作し, シミュレーションによって得られた制御器パラメータと, 制御器アルゴリズムを実装した. このシステムの動作実験によって次の結果が得られた.

スタートアップ時の特性は負荷抵抗を $R=\infty[\Omega]$, $0.33[\Omega]$, $0.165[\Omega]$, 入力電圧を $7.6[V]$, $9.6[V]$, $11.6[V]$ とするすべての組み合わせにおいてシミュレーションとほぼ同様の結果を得られた.

負荷変動実験では, 負荷抵抗を $0.33[\Omega] \leftrightarrow 0.165[\Omega]$ と急変させ, 出力電圧変動を $30[mV]$ 以内に収めることができたが, シミュレーションよりやや大きい値になることを確認した.

これによってすべての実験で仕様に近い値を得られることが確認され, 提案法の有用性と有効性が示された.

今後の課題は, H/W を改善して負荷変動実験のシミュレーション結果と実験結果を近づけること, 制御アルゴリズムを改善することでスイッチング周波数をさらに上げること, $K(z)$ を変更して, 現在2次微分特性となっている式(4.40)・(4.41)を3次微分特性として, 負荷変動に対する出力電圧変動の振幅をより小さくすること, 設計パラメータの改善によりさらに制御帯域を広くすること, などが挙げられる. また, 本研究では行わなかった, 入力変動についてもシミュレーションを行い, 実機で十分な実験を行うことや, フィードフォワードを利用したコントローラを設計することも考えられる.

謝辞

最後に、本研究を進めるにあたり、親切なご指導と的確なアドバイスを賜った樋口幸治准教授、梶川竜義技官、並びに中野和司教授に深く感謝致します。

また、ご指導いただいた羽根田さん、高仲さん、Savioさん、藤田さん、岩田さん、季さん、林さん、小林さん、野村さんにも深く感謝いたします。そして、研究生活を共に過ごした鶴堀君、山岸さん、小村君、平尾君にも感謝致します。

参考文献

- [1]竹上 栄治, 樋口 幸治, 中野 和司: “二次微分外乱特性付加近似的 2DOF デジタル制御器による DC-DC コンバータのロバスト制御”, 電気学会論文誌D, Vol. 129, No. 12, pp.1137-1146 2009
- [2]樋口 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
- [3] “SH7216 グループハードウェアマニュアル”, ルネサステクノロジ 2009
- [4] “SH7216CPU ボード R0K572167C001BR ユーザーズマニュアル”, ルネサステクノロジ 2009
- [5]石田義久, 鎌田弘之: “デジタル制御のポイント”, 産業図書 1990
- [6]三上直樹: “デジタル信号処理とDSP”, CQ 出版社 1999

本研究で使用したソフトウェア

- [1]MATLAB Version 7.5.0(R2007b):数値計算シミュレーションソフトウェア
- [2]High-performance Embedded Workshop Version 4.07:ルネサスマイコン統合開発環境
- [6]OpenOffice.org Version 3.1.1:オープンソースオフィスソフトウェア
- [7]BSch3V 080112 版:フリー回路図エディタ

本研究で使用した実験機器

- [1]KIKUSUI 直流安定化電源 PAN16-30A JM001291
- [2]TAKASAGO 電子負荷 FK-200L 5220199020
- [3]Agilent オシロスコープ 54624A MY40001889
- [4]HIOKI クランプオンプローブ 3273-50 090919702
- [5]HIOKI クランプオンプローブ用電源 3272 091010817
- [6]Renesas エミュレータ E10A 05606E
- [7]パソコン WindowsXP HP Compaq6715s CPU Mobile AMD Sempron 3500+ 1.79GHz memory 896MB HDD 65.8GB

付録 A MATLAB シミュレーションプログラム

今回シミュレーションを行った MATLAB のシミュレーションプログラムを以下に示す.

```
format long e ;
clear all;close all;
clc;
%STEP.1%%_SAMPLING FREQUENCY_%%%%%%%%%%
fs=450e3 ; %[Hz] e3=(*100)
%STEP.2%%_CHANGE PARAMETER_%%%%%%%%%%
H1 = -0.1+0.5i
H2 = -0.885
H3 = -0.1-0.5i
H4 = -0.3+0.3i
H5 = -0.3-0.3i
Kz = 0.17
kq = -1
%STEP.3%%_OPTION_%%%%%%%%%%
%Endofgraph = 0.4e-3;
% If it happened something errors, please change large number
% FOR EXAMPLE : fs=900[KHz]→ Endofgraph=0.6e-3
%%%%%%%%%
S_FREQUENCY = round(fs) %output for command window
Ts = 1/fs %sampling frequency [Hz]→ time[sec]
Tdelay = Ts ; %delay time
Vi = 9.6 ; %input voltage [V]
L = 0.999*Ts ;%2/24,26 実験使用 inductor[H]
L0 = 0.555e-6 ;%inductor[H]
C0 = 235e-6 ; %capacitor[F]
RL = 0.33 ; %resistor[Ω]
R0 = 7e-3 ; %resistor[Ω]
M_clock = 100e6; % SH7216 M clock for MTU2S 100[MHz]
M_clock_time=1/M_clock; %fewquency[Hz]→time[sec]
step=Ts/(M_clock_time*2) ;%[μs/ns]
one_cycle_step=round(step) %output for command window
one_step_of_voltage = Vi/one_cycle_step %[mV] output for command window
Kadc=4096/4.66 ;%ADC gain {2^(12)[bit]/4.5[V]}for SIMULINK4.762
Vout_d=round(3.3*Kadc) % output voltage analog 3.3 to digital
h=1; %for SIMULINK
Kpwm=-1/step ;
```

```

Gp=((RL+R0)/(RL))*Kpwm*Vi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ac=[-1/(C0*RL) 1/C0
-1/(L0) -R0/L0];
Bc=[0
Gp/(L0)];
C=[1 0];
[Aw1,Bw1,Cw,D]=c2dt(Ac,Bc,C,Ts,L);
%連続状態空間モデルを入力に時間遅れをもたせながら離散時間モデルに変換
[Ad1,Bd1]=c2d(Ac,Bc,Ts) ;%continuous time to discrete time
o2=[0 0];
o3=[0
0
0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Aww1=[Ad1 Bd1 o2' o2'
o3' 1 0
o3' 0 1
o3' 0 0];%転置演算子
Bww1=[o3
0
1];
Cww=[C 0 0 0];
[nwwd0,dewwd0,kd00]=ss2zp(Aww1,Bww1,Cww,D);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P=[-H2 -H1 -H3 -H4 -H5];
F=place(Aww1,Bww1,P) ;%極配置
H=(1+H1)*(1+H2)*(1+H3)/(1-nwwd0(1,1));%零点注意
E=Cww*Aww1^3*Bww1 ;
G=1/E ;
[nwwF1,dewwF1,kwwF1]=ss2zp(Aww1-Bww1*F,Bww1*G*H,Cww,D);
f1=-F(1,1);
f2=-F(1,2);
f3=-F(1,3);
f4=-F(1,4);
f5=-F(1,5);
a11=Ad1(1,1);
a12=Ad1(1,2);
a21=Ad1(2,1);

```

```

a22=Ad1(2,2);
b11=Bd1(1,1);
b21=Bd1(2,1);
g0=H4*H5;
g1=H4+H5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k =(f3*a12+a12*kq^3+a12*kq^2*f5-a12*kq*f4-f2*b11)/((a12*b21-kq*b11-a22*b11)*a12) ;
k1=real((f1-f2*a11/a12+k*(-a11^2-a21*a12+(a12*a11+a12*a22)*a11/a12)+f5*(-
k*(a12*a11+a12*a22)/a12+f2/a12)+f5^2*k+(f4-k*b11)*k)-G*H*Kz*(g0+(g1+f5)*f5+f4-k*b11)/(1+H2));
k2=real(k-G*H*Kz/(1+H2));
k3=real(-k*(a11*a12+a12*a22)/a12+f2/a12+f5*k-(g1+f5)*G*H*Kz/(1+H2));
k4=real(f3-f2*b11/a12+k*((a12*a11+a12*a22)*b11/a12-a11*b11-b21*a12));
k5=real(f4-k*b11);
k6=real(f5);
ki1=real((g0+(g1+f5)*f5+f4-k*b11)*G*H*Kz);
ki2=real((g1+f5)*G*H*Kz);
ki3=real(G*H*Kz);
k1r=real(G*H);
k2r=real((g1+f5)*G*H);
k3r=real((g0+(g1+f5)*f5+f4-k*b11)*G*H);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% 負荷急変システム %%%%%%%%%
RL_var1=0.165 ;
A1_varRL1=[-1/(C0*RL_var1) 1/C0
-1/(L0) -R0/L0];
[Ad1_varRL1,Bd1_varRL1]=c2d(A1_varRL1,Bc,Ts) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1_dsp=k1/Kadc;
k2_dsp=k3/Kadc;
k3_dsp=k2/Kadc;
k4_dsp=k4;
k5_dsp=k5;
k6_dsp=k6;
ki1_dsp=ki3/Kadc;
ki2_dsp=ki2/Kadc;
ki3_dsp=ki1/Kadc;
k1r_dsp=k1r/Kadc;
k2r_dsp=k2r/Kadc;
k3r_dsp=k3r/Kadc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

sim('takanaka004');
figure;
plot(Vout_data(:,1),Vout_data(:,2)),...
    xlabel('Time[sec]'),ylabel('Output voltage[V]'),grid ;
title('Start up response (simulink)');
axis([0.03e-4,0.1e-3,1,3.5]);
figure;
plot(U_B_Data(:,1),U_B_Data(:,2)),...
    xlabel('Time[sec]'),ylabel('Output voltage[V]'),grid ;
title('操作量 u');
axis([0,0.2e-3,-50,0]);
figure;
plot(Vout_data(:,1),Vout_data(:,2)),...
    xlabel('Time[sec]'),ylabel('Output voltage[V]'),grid ;
title('Dinamic Load Response');
axis([0.48e-3,0.75e-3,3.2,3.4]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ↓ output for command window
fprintf('%n ↓ OUTPUT TO HEW%n\n#define S_FREQUENCY %d // Hz %n',S_FREQUENCY);
fprintf('%n#define K1 %d\n',k1_dsp);
fprintf('#define K2 %d\n',k2_dsp);
fprintf('#define K3 %d\n',k3_dsp);
fprintf('#define K4 %d\n',k4_dsp);
fprintf('#define K5 %d\n',k5_dsp);
fprintf('#define K6 %d\n',k6_dsp);
fprintf('#define Ki1 %d\n',ki1_dsp);
fprintf('#define Ki2 %d\n',ki2_dsp);
fprintf('#define Ki3 %d\n',ki3_dsp);
fprintf('//H1 = %1.2f+%1.2fi\n',real(H1),imag(H1));
fprintf('//H2 = %1.2f\n',H2);
fprintf('//H3 = %1.2f+%1.2fi\n',real(H3),imag(H3));
fprintf('//H4 = %1.2f+%1.2fi\n',real(H4),imag(H4));
fprintf('//H5 = %1.2f+%1.2fi\n',real(H5),imag(H5));
fprintf('//Kz = %1.2f\n',Kz);
fprintf('//kq = %1.2f\n\n',kq);

```


付録 B SH マイコンメインプログラム

以下に SH マイコンに使用したメインプログラムを示す.

```
#include <machine.h>
#include "vect.h"
#include "iodefine.h"
#define ENABLE 1
#define DISABLE 0
// ==== Parameters from MATLAB ====
#define S_FREQUENCY 450000 // Hz
#define K1 1.204541e+000
#define K2 -1.520982e+000
#define K3 8.436175e-001
#define K4 8.491034e-001
#define K5 3.733202e-001
#define K6 -2.224235e-001
#define Ki1 -7.282576e-003
#define Ki2 5.989362e-003
#define Ki3 -5.361771e-003
//H1 = -0.10+0.50i
//H2 = -0.885
//H3 = -0.10+-0.50i
//H4 = -0.30+0.30i
//H5 = -0.30+-0.30i
//Kz = 0.17
//kq = -1.00
/* ==== CMP constant value ==== */
#define PWM_CONST DISABLE
#define DUTY_CYCLE 34 /*[%]*/
/* ==== prototype declaration ==== */
void main(void);
void mtu2S_init(void);
void pfc_init(void);
void io_ad_init(void);
void control_routine(void);
/* ==== symbol definition ==== */
#define TRM_CONST 2900
#define TRM_START_CONST 860
#define MTU2S_TIMER_TGRA (1000000000)/((S_FREQUENCY)*(10)*(2))
```

```

#define MTU2S_TGRA_CONST      ( (MTU2S_TIMER_TGRA)*(100-(DUTY_CYCLE))/100 )
#define CARRIER_CYCLE      ( (float)(MTU2S_TIMER_TGRA) )
#define DEAD_TIME      ((short)1)
#define PUL_CYCLE      (CARRIER_CYCLE+DEAD_TIME)
/* ==== RAM allocation variable declaration ==== */
volatile float Target_B = TRM_START_CONST ;
volatile float Ua ;
volatile float Ub ;
volatile float Uc ;
volatile float Uv ;
volatile float Uc1 ;
volatile float Ub1 ;
volatile float Ua1 ;
volatile float Uv1 ;
volatile float Uv2 ;
volatile float Uv3 ;
volatile float Vs_B ;    // output voltage (in AD converter)
volatile float E_B ;    // difference voltage (Target B - Vs B)
volatile float U_B ;    // OFF part of DUTY-CYCLE
void main(void)
{
    /* ==== Setting of MTU2 ==== */
    mtu2S_init();
    /* ==== interrupt priority register ==== */
    INTC.IPR02.BIT._IRQ6 = 0x0A;          /* IRQ6 switch interrupt level =10 */
    INTC.ICR1.BIT.IRQ6S = 0x01;          /* IRQ6 switch down egde */
    /* ==== Setting of PFC ==== */
    pfc_init();
    /* ==== Initial setting of ADC ==== */
    io_ad_init();
    /* ==== timer start register(TSTR) ==== */
    MTU2S.TSTR.BYTE = 0xC0;    /* Start counting on channels 3 and 4 */
    /* ==== clear the interrupt mask ==== */
    set_imask(0);
    while(1)                /* loop */
    {
        if(ADC0.ADSR.BIT.ADF == 0x01 )
        {
            /* ==== B0 set {check operation time} ==== */

```

```

        PE.DR.BIT.B15 = 1;
    ADC0.ADSR.BIT.ADF = 0;
    control_routine ();
        PE.DR.BIT.B15 = 0 ;
    } //if end
} //while end
} //main end

void control_routine (void)
{
//***** Control routine B *****
    Vs_B = (ADC0.ADDR2) ; /* ----ADC ---- */
    E_B = (Target_B) - (Vs_B);
    Uc = (Ub) + ( (K3) * (Vs_B) ) + (Uv1);
    U_B = (Uc) + ( CARRIER_CYCLE );

    /* ==== set PWM duty ==== */
    #if PWM_CONST == DISABLE // feedback control (closed roop)
        MTU2S4.TGRD = U_B;
    #elif PWM_CONST == ENABLE //no use feedback control (open roop)
        MTU2S4.TGRD = MTU2S_TGRA_CONST;
    #endif

//***** Control routine A *****
    Ub = (Ua) + (Uv2) + K2 * (Vs_B) ;
    Ua = (Uv3) + (Uc1) + (Ub1) + (Ua1) + ( (K1) * (Vs_B) ) ;
    Uv = (Uv) + (E_B) ;
    Uc1 = (K4) * (Uc) ;
    Ub1 = (K5) * (Ub) ;
    Ua1 = (K6) * (Ua) ;
    Uv1 = (Ki1) * (Uv) ;
    Uv2 = (Ki2) * (Uv) ;
    Uv3 = (Ki3) * (Uv) ;
}

void mtu2S_init(void)
{
    unsigned short Dead_time; /* dead time */
    unsigned short C_cycle; /* carrier frequency/2 */
    unsigned short Pul_cycle; /* arrier frequency/2 + dead time */
    unsigned short Ad_start; /* ad start desire count */

    /* ==== Setting of variable ==== */
    Dead_time = DEAD_TIME;

```

```

C_cycle = CARRIER_CYCLE;
Pul_cycle = PUL_CYCLE;
//      Ad_start = AD_START;
/* ==== Setting of MTU2 CH3,4 ==== */
/* ---- Stop operating the timer counter ---- */
MTU2S.TSTR.BIT.CST4 = 0x00;
      MTU2S.TSTR.BIT.CST3 = 0x00;
/* ---- timer control register 3(TCR) ---- */
MTU2S3.TCR.BIT.CCLR = 0x00;
      MTU2S3.TCR.BIT.CKEG = 0x00;
      MTU2S3.TCR.BIT.TPSC = 0x00;
/* ---- timer control register 4(TCR) ---- */
MTU2S4.TCR.BIT.CCLR = 0x00;
      MTU2S4.TCR.BIT.CKEG = 0x00;
      MTU2S4.TCR.BIT.TPSC = 0x00;
/* ---- timer counter 3(TCNT) ---- */
MTU2S3.TCNT = Dead_time;      /* dead time */
/* ---- timer counter 4(TCNT) ---- */
MTU2S4.TCNT = 0x0000;      /* Clear the timer counter */
MTU2S4.TGRB = MTU2S_TGRA_CONST ;      /* compare register for PWM3 output */
MTU2S4.TGRD = MTU2S_TGRA_CONST ;      /* buffer register for TGRB_4 */
MTU2S3.TGRA = Pul_cycle;      /* Toggle wave output that
                                synchronized at compare match cycle */
MTU2S3.TGRC = Pul_cycle;      /* buffer register for TGRA_3 */
/* ---- timer dead time data register(TDDR) ---- */
MTU2S.TDDR = Dead_time;      /* dead time */
/* ---- timer cycle data register(TCDR) ---- */
MTU2S.TCDR = C_cycle;      /* 1/2 carrier cycle */
/* ---- timer cycle buffer register(TCBR) ---- */
MTU2S.TCBR = C_cycle;      /* Buffer register for TCDR_3 */
/* ---- timer output control register 1(TOCR1) ---- */
MTU2S.TOCR1.BIT.PSYE = 0x01; /* Toggle output enable */
/* ---- timer mode register(TMDR) ---- */
MTU2S3.TMDR.BIT.BFB = 0x01;
      MTU2S3.TMDR.BIT.BFA = 0x01;
      MTU2S3.TMDR.BIT.MD = 0x0f;
/* ---- timer output master enable register(TOER) ---- */
MTU2S.TOER.BIT.OE4D = 0x01;
/* ---- timer interrupt enable register(TIER) ---- */

```

```

    MTU2S4.TIER.BIT.TTGE2 = 0x01; /*A/D Start Request Enable TTGE=1(TCNT_4 underflow)*/
}

void io_ad_init(void)
{
    /* ===== Setting of ADC ===== */
    /* ----- A/D Control Register(ADCR) ----- */
    ADC0.ADCR.BIT.ADST = 0x00; /* 7 = b'0 : A/D start */
    ADC0.ADCR.BIT.ADCS = 0x00; /* 6 = b'0 : Single-cycle scan */
    ADC0.ADCR.BIT.ACE = 0x00; /* 5 = b'0 : Automatic clearing of ADDR */
    ADC0.ADCR.BIT.ADIE = 0x00; /* 4 = b'0 : Generation of A/D conversion end
                                interrupt is disabled */
    ADC0.ADCR.BIT.TRGE = 0x01; /* 1 = b'1 : A/D conversion start by the external trigger
                                or an A/D conversion start trigger from
                                the MTU or MTU2S is disabled */
    ADC0.ADCR.BIT.EXTRG = 0x00; /* 0 = b'0 : A/D converter is started by the A/D conversion
                                start trigger from the MTU2 or MTU2S */
    /* ----- A/D Start Trigger Select Register(ADSTRGR) ----- */
    ADC0.ADSTRGR.BIT.STR6 = 0x01; /*select TRGAn from MTU2S*/
    /* ----- A/D Analog Input Channel Select Register(ADANSR) ----- */
    ADC0.ADANSR.BIT.ANS2 = 0x01; /* AN2 */
    /* ----- A/D Bypass Control Register(ADBYPSCR) ----- */
    ADC0.ADBYPSCR.BIT.SH = 0x00; /* not use S&H */
    /* ----- A/D Status Register (ADSR) ----- */
    ADC0.ADSR.BIT.ADF = 0; /* ADF clear */
}

void pfc_init(void)
{
    /* ===== Setting of PFC ===== */
    /* ----- port A control register H2 ----- */
    PFC.PACRH2.BIT.PA20MD = 0x03; /* IRQ6 SET */
    /* ----- port E control register L4 ----- */
    PFC.PECRL1.BIT.PE3MD = 0x05; /* TIOC4DS SET (PWM OUTPUT)*/
    /* ----- port D control register H2 ----- */
    PFC.PDCRL3.BIT.PD8MD = 0x05; /* TIOC3AS SET : toggle output*/
    /* ----- port D IO register L ----- */
    PFC.PDIORL.BIT.B8 = 0x01; /* TIOC3A output : toggle output*/
    /* ----- port E IO register L ----- */
    PFC.PEIORL.BIT.B3 = 0x00; /* WAIT! TIOC4DS input (PWM OUTPUT)*/
    /* ===== change output {check calculate time} ===== */

```

```

        PFC.PEIORL.BIT.B15 = 0x01; /*check opelation time */
        PFC.PEIORL.BIT.B9  = 0x01; /*LED7 OUTPUT */
        PFC.PEIORL.BIT.B11 = 0x01; /*LED8 OUTPUT */
    }
#pragma section IntPRG
// 11 NMI
void INT_NMI(void)
{
    int j=0;
    MTU2S4.TGRD = MTU2S_TGRA_CONST;
    for(j=0;j<600000;j++)
    {
        nop();
        nop();
        nop();
    }
    PE.DR.BIT.B9 ^= 0x01; //LED
    PFC.PEIORL.BIT.B3 ^= 0x01; // TIOC4DS input
    Ua  = 0;
    Ub  = 0;
    Uc  = 0;
    Uv  = 0;
    Uv1 = 0;
    Uv2 = 0;
    Uv3 = 0;
    Ua1 = 0;
    Ub1 = 0;
    Uc1 = 0;
    Vs_B = 0;
    E_B  = 0;
    U_B  = 0;
}
// 70 Interrupt IRQ6
void INT_IRQ6(void)
{
    if(Target_B == TRM_START_CONST)
    {
        Target_B = TRM_CONST;
        PE.DR.BIT.B11 = 0x01;
    }
}

```

```
    }  
    else  
    {  
        Target_B = TRM_START_CONST;  
        PE.DR.BIT.B11 = 0x00;  
    }  
}  
/* End of File */
```

付録 C 入力電圧を変化させたときのスタートアップ応答結果

以下に, 入力電圧を変化させたときのスタートアップ応答結果を示す.

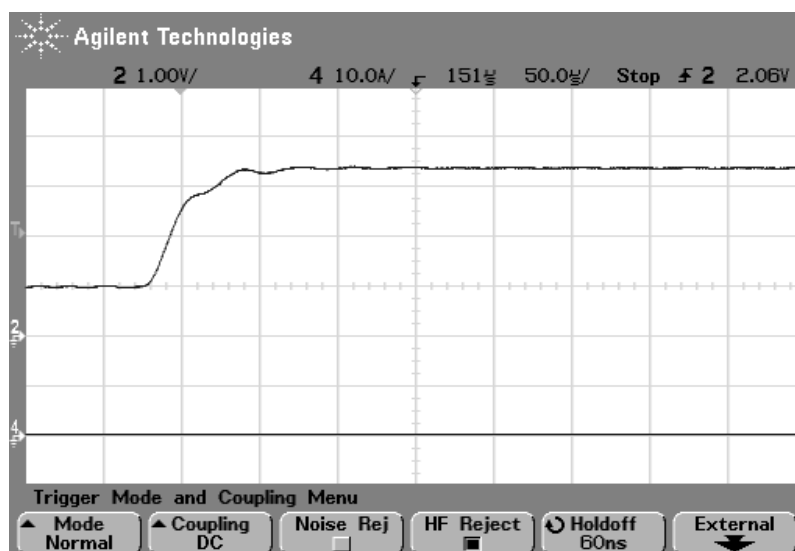


図 C.1: 入力電圧が 7.6V の時のスタートアップ応答結果(負荷 0.33 Ω)

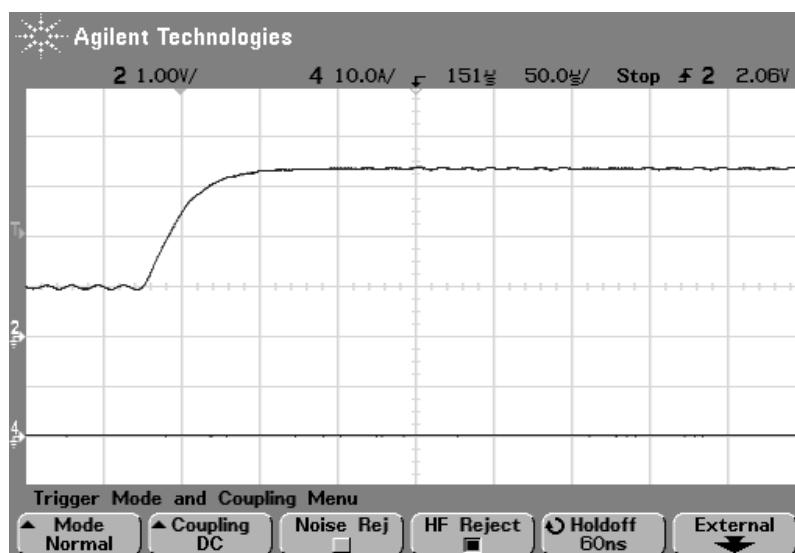


図 C.2: 入力電圧が 11.6V の時のスタートアップ応答結果(負荷 0.33 Ω)

付録 D DC-DC コンバータボードおよびその回路図

以下に DC-DC コンバータシステムとその回路図を示す。図で分かるように、SH マイコンはルネサステクノロジ製 CPU ボードを使用したため、その部分の回路図は CPU ボードのユーザーズマニュアルを参照されたい。

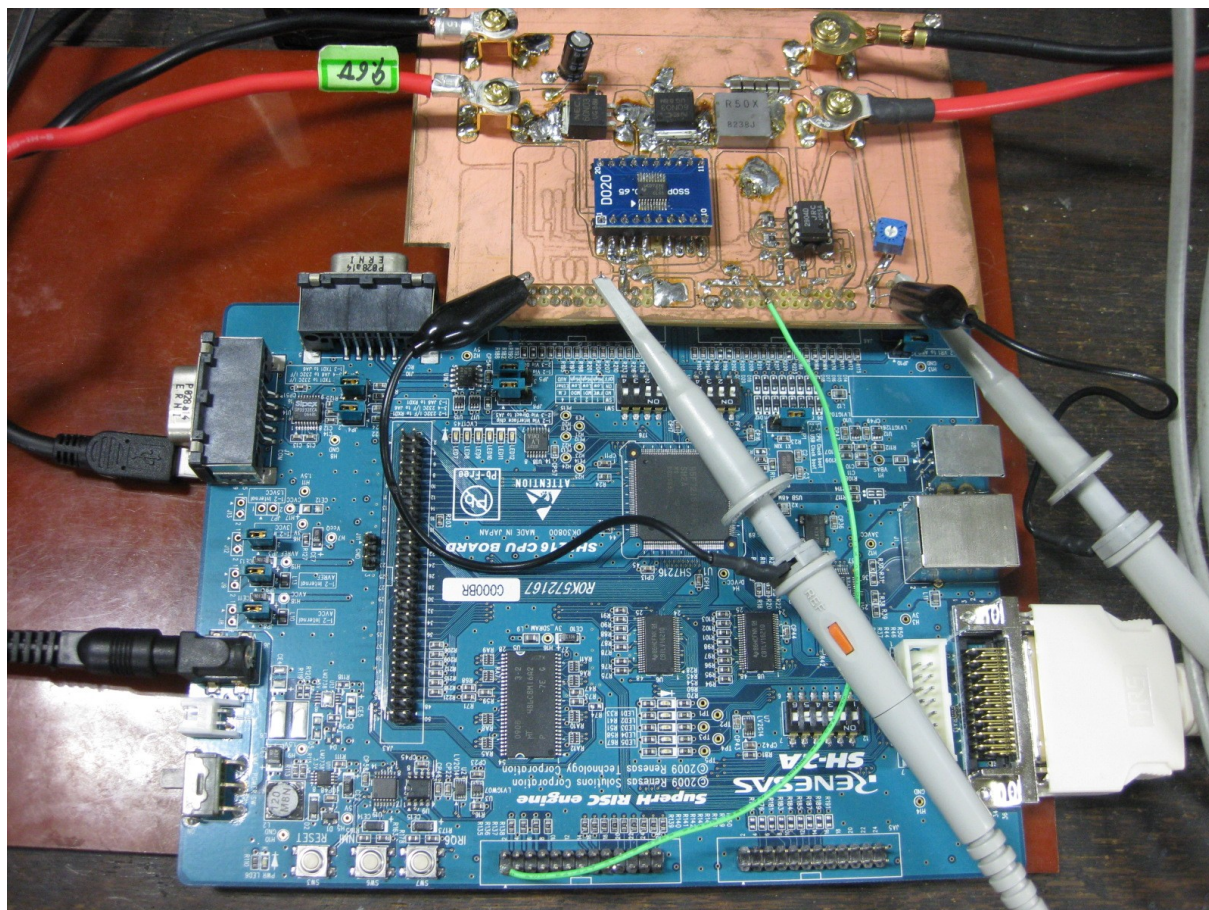


図 D.1:DC-DC コンバータシステム

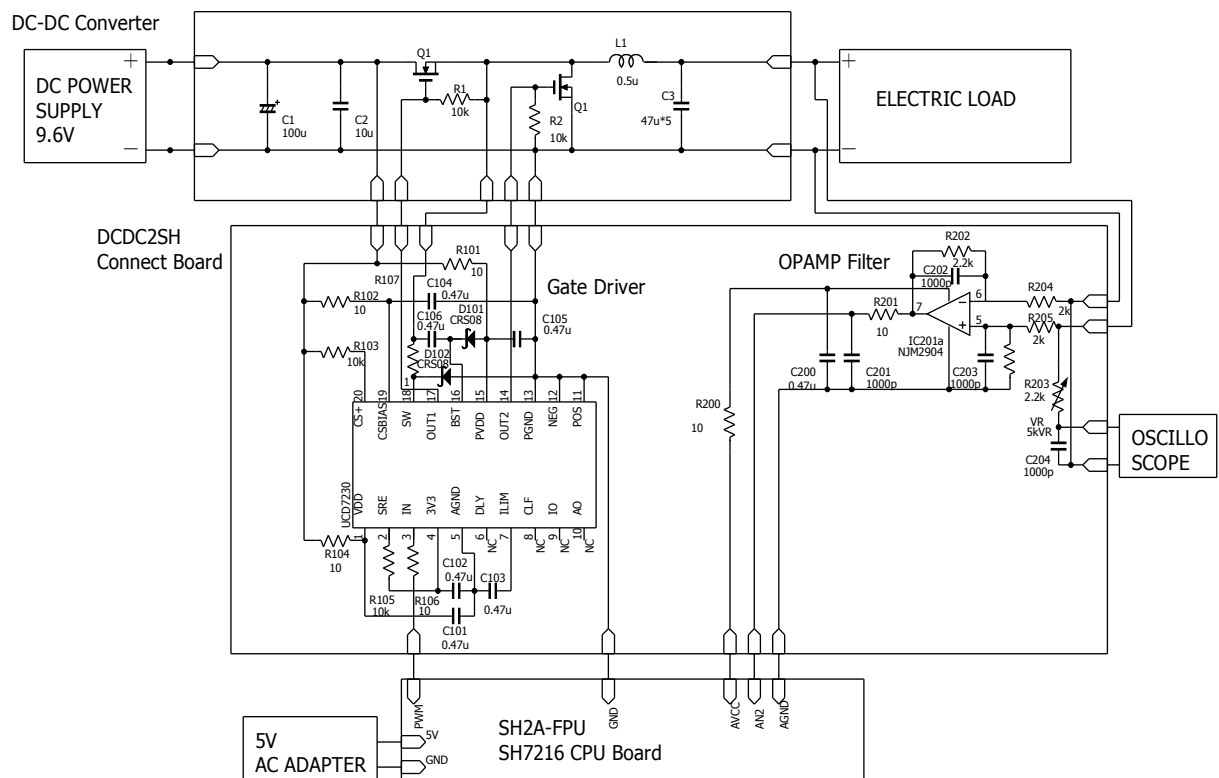


図 D.2:DC-DC コンバータシステム全体の回路図

付録 E 卒業研究発表会で挙げられた質問とその回答

2010 年 2 月 15 日に行われた平成 21 年度卒業研究発表会において、私の発表後の質疑応答の際に挙げられた質問について述べる。

①負荷急変実験において、シミュレーションの結果と実記実験の結果が違うのはなぜか

Ans.

今回、新規にハードウェアを製作した点や、実機で動作させた場合どうしても発生してしまう演算誤差などをシミュレーションで再現できていないことが原因であると考えられる。たとえば、今回演算に使った SH マイコンは浮動小数点演算をしているが、これには桁落ちや情報落ち、丸め誤差、打ち切り誤差などの誤差が発生する。さらに、実際の FET のスイッチングには、スイッチング損失や制御対象の非線形性などが問題となるが、これらの要素はシミュレーションでは考慮していない。すると、シミュレーションでは電流波形が少しがたついていたり、また、シミュレーションでは出なかった出力電圧のリングングが実機では発生するなどの違いが出てしまう。

こういった違いを改善するために、シミュレーションを見直す必要があると考えられる。

②先行研究と対比して、SH マイコンを使用したことにより、制御器の部分で違いが出たのはどこか

Ans.

式(2.18)などに使用される、遅れ L の値を DSP の場合は $T_s/2$ だったのに対し、今回は $0.99T_s$ と、大きくした。また、先行研究では、 q - y の伝達特性を決める k_z の値が 0.1 程度だったのに対し、今回は 0.17 と大きめにし、それに伴い $H1 \sim H5$ までのパラメータを調整して、さらに外乱に対してロバストになるようなパラメータとした。

③DSP と SH マイコンでは、公称の演算性能は SH マイコンのほうが上回っているのに、実際の演算時間はなぜ DSP のほうが短いのか

Ans.

DSP と SH マイコンでそれぞれ採用しているアーキテクチャの構造上、このような結果になってしまう。たとえば、ただの浮動小数点の加減算を延々と繰り返すような計算なら、公称性能に見合った演算時間で演算が終わるが、今回用いた制御ルーチンの演算においては、多数の分岐演算や積和演算を含むため、高速乗算器を持ち、分岐によるオーバーヘッドの少ない DSP のほうが、同じルーチンでも短い演算時間で処理を終える結果となる。