

# FPGAに実装したCPUと複数のデジタルPID制御器 による機構制御システム

富重 晃季<sup>\*1</sup> 山田 健仁<sup>\*2</sup>

## Mechanical Control System using Microprocessor and Multiple Digital PID Controllers on FPGA

Koki TOMISIGE <sup>\*1</sup> and Takehito YAMADA <sup>\*2</sup>

### Abstract

Precision mechanical control system is needed a high speed processing controller. Digital control system is usually constructed by using micro-processor. So, the calculation performance of micro-processor is important for the precision mechanical controller. However, a single chip micro-processor clock frequency is becoming limited in terms of power consumption. In this study, we have realized a high-speed compact size controller by implementing PID processing devices with a micro-processor on a Field-Programmable Gate Array under relatively low clock frequency. The control system was applied to an inverted pendulum mechanism to confirm the controller performance and an amount of logic circuits.

**Key Words :** Mechanical control, Inverted pendulum, Digital PID control, FPGA

### 1. はじめに

磁気ディスク装置の磁気ヘッド位置決め制御など、機構制御には高速・高精度が求められ、高度な制御手法の実装が重要な技術課題となっている<sup>1)</sup>。一般にデジタル制御器を実装するには、マイクロプロセッサを利用することが多く、その演算速度が制御器実装の上で重要となる<sup>2)</sup>。しかし、マイクロプロセッサのクロック周波数の増加による演算速度の高速化は、消費電力増大の観点から限界になりつつある。このため制御演算機構をハードウェアにより実現し、比較的低いクロック周波数で実現することは、有効な制御器実装手法と考えられる。制御演算機構をハードウェア化することで、複数の制御演算を並列に実行できるため、総合的な演算速度はシングルコアのCPUに比較して優れていることは明らかである。

本研究では、電子素子の分野で一般的となってきたプログラム可能なロジックデバイスであるField-Programmable Gate Array (FPGA)を活用し、FPGA

上に制御器全体の統合調整機能を担うCPUと制御パラメータを調整可能なデジタルPID制御器を複数搭載した、柔軟で汎用性のある制御器を実現することを目的としている。

提案する構成での制御器の有効性を実証するために、制御器の性能評価に一般的に使用される倒立振子を制御対象の機構とし、デジタルPID制御器を2セットとCPUを実装した実験を行った。FPGA上に構成したデジタルPID制御器の制御パラメータ(比例・積分時間・微分時間)は、CPUからプログラムにより変更できる構成となっており、可変制御器としての基本機能を実現している。

本論文では、デジタルPID制御器の設計例とデジタルPID制御器及びその周辺回路のVHDL記述例を示すとともに、実装に必要な論理素子数を見積もることで、このようなCPUとデジタル制御器を同時に搭載する統合型制御器のFPGA上への実装について検討した結果を述べる。

<sup>\*1</sup> 情報電子工学専攻

<sup>\*2</sup> 情報電子工学科

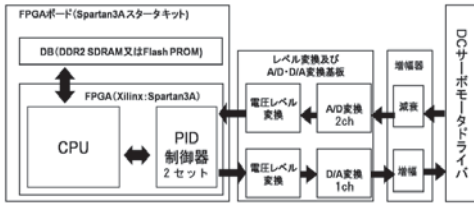


図1 FPGAを中心とした制御システムの構成図

表1 実験装置の各要素

FPGA ボード	FPGA	Spartan3A XCS700A
	CPU	MicroBlaze
電圧レベル変換, A/D・D/A変換基板	電圧レベル変換	TC74LVX4245FS (東芝社製)
	A/D コンバータ (12bit)	MCP3204 (Microchip Technology 社製)
	D/A コンバータ (12bit)	DAC7611 (Burr Brown 社製)
増幅器	OP アンプ	TL074CN (Texas Instrument 社製)
制御対象 (DCサーボモータ)	倒立振り子制御実験装置	STTS-1 (サーボテクノ社製)

### 2. 制御システムの構成

図1にFPGAを中心とした制御システムの構成図を示す。FPGAにはXilinx社製のSpartan3Aを使用した。このFPGA上にCPUとデジタルPID制御器2セットを実装した。なお、CPUはXilinx社より提供されているソフト・マイクロプロセッサのMicroBlazeを使用した。

表1に実験装置の各要素を示す。FPGAボード上には、FPGAの他に、外部RAM(DDR2 SDRAM)、外部ROM(Flash PROM)、汎用IOポート等が搭載されている。また、FPGAには、CPU、デジタルPID制御器回路の他に、A/D・D/A変換器制御回路、CPUとデジタルPID制御器回路との通信回路を実装する。CPUにデジタルPID制御器パラメータの調整機能を持たせることで、制御系の自動最適化や可変制御器が実現できる。なお、このような構成でのPID制御器パラメータの最適化技術<sup>3)</sup>に関しては多くの手法があるが、今回の構成及び制御対象での有効性に関しての検証は、今後の課題とする。

### 3. 倒立振り子用制御器の設計

倒立振り子機構の制御手法は、二足歩行ロボット、ロケットの発射台、クレーンなどに応用されている。制御手法としては、状態フィードバック制御、PID制御、ファジィ制御、スライディングモード制御など多々あげられるが、本研究では、最も多く使用されている制御器であるPID制御器をFPGA上に複数実装することに

よってコンパクトで高速な制御器を実現することを検討する。

倒立振り子は、本質的に不安定な機構であり、フィードバック制御により安定化する必要がある、制御手法の有効性を検証するのによく使われる制御対象である。

### 3.1 倒立振り子のモデル

図2に制御対象である台車型一次元倒立振り子の模式図を示す。各記号を以下とすると物理モデルは、式(1)、式(2)で与えられる。

- $L$  : 回転軸から振り子の重心までの距離 [m]
  - $m$  : 振り子の質量 [kg]
  - $J$  : 重心回りの慣性モーメント [kgm<sup>2</sup>]
  - $\mu_\theta$  : 回転軸回りの粘性抵抗係数 [Nm/s]
  - $M$  : 台車の質量 [kg]
  - $\mu_z$  : 台車の移動に対する粘性抵抗係数 [Nm/s]
  - $\xi$  : 台車系の物理定数 [m/s<sup>2</sup>]
  - $\zeta$  : 台車系の物理定数 [1/s]
  - $g$  : 重力加速度 (=9.81 [m/s<sup>2</sup>])
- なお、振り子の慣性モーメントは、 $J = mL^2/3$  である。

- ・台車と振り子の台車移動方向の運動方程式

$$(M+m)\frac{d^2z}{dt^2} + mL\cos\theta\frac{d^2\theta}{dt^2} - mL\left(\frac{d\theta}{dt}\right)^2\sin\theta + \mu_z\frac{dz}{dt} = f \tag{1}$$

- ・振り子の回転軸周りの運動方程式

$$mL\cos\theta\frac{d^2z}{dt^2} + (J+mL^2)\frac{d^2\theta}{dt^2} - mgL\sin\theta + \mu_\theta\frac{d\theta}{dt} = 0 \tag{2}$$

倒立状態に近い場合、 $\sin\theta \approx \theta$ 、 $\cos\theta \approx 1$ 、 $\left(\frac{d\theta}{dt}\right)^2 \approx 0$ と近似できるため、以下の線形連立微分方程式となる。

$$(M+m)\frac{d^2z}{dt^2} + mL\frac{d^2\theta}{dt^2} + \mu_z\frac{dz}{dt} = f \tag{3}$$

$$mL\frac{d^2z}{dt^2} + (J+mL^2)\frac{d^2\theta}{dt^2} - mgL\theta + \mu_\theta\frac{d\theta}{dt} = 0 \tag{4}$$

さらに、台車とその駆動系をサーボ系として速度フィードバック制御すると、質量や台車の粘性抵抗は補償されて、その運動方程式は式(5)と単純化できる。

$$\frac{d^2z}{dt^2} = -\zeta\frac{dz}{dt} + \xi u \tag{5}$$

また、入力電圧  $u$  から台車位置  $z$  までの伝達関数は、

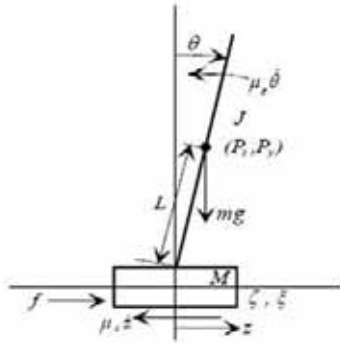


図2 台車型一次元倒立振子の模式図

表2 倒立振子物理パラメータ

振り質量	m	0.023	[kg]
回転軸から重心までの距離	L	0.2	[m]
重心回りの慣性モーメント	J	$3.20 \times 10^{-4}$	[kgm <sup>2</sup> ]
回転軸回りの粘性摩擦抵抗	$\mu_\theta$	$27.41 \times 10^{-6}$	[Nm/s]
台車の物理定数パラメータ	$\zeta$	240	
台車の物理定数パラメータ	$\xi$	90	

$Z(s) = \frac{\xi}{s(s+\zeta)}$  の2次系で与えられることになる。以上  $U(s)$

より、倒立振子の物理モデルは、式(6)に示す状態方程式、及び出力方程式で表せる。

$$\frac{d}{dt} \begin{bmatrix} z \\ \theta \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\zeta & 0 \\ 0 & \frac{mLg}{J+mL^2} & \frac{mL\zeta}{J+mL^2} & -\frac{\mu_\theta}{J+mL^2} \end{bmatrix} \begin{bmatrix} z \\ \theta \\ \dot{z} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \xi \\ -\frac{mL\xi}{J+mL^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \theta \\ \dot{z} \\ \dot{\theta} \end{bmatrix} \quad (6)$$

なお、倒立振子の物理パラメータ値を表2に示す。

### 3.2 状態フィードバックによる制御

2. で述べたように制御器は汎用性を考え、デジタル PID 制御器として実装するが、基準となる PID 制御器のパラメータを求めめるため、状態フィードバックによる倒立振子の安定化を考える。この時、操作量  $u$  は式(7)で与えられる。

$$u = [k1 \ k2 \ k3 \ k4] \cdot [z \ \theta \ \dot{z} \ \dot{\theta}]^T \quad (7)$$

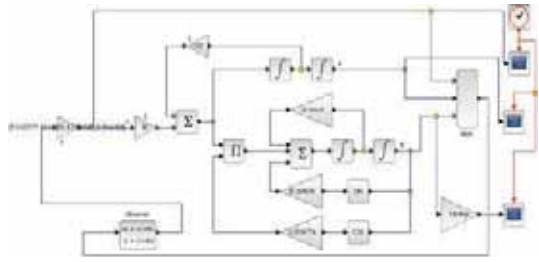


図3 倒立振子状態フィードバック制御のシミュレーションプログラム (Xcos)

$[k1 \ k2 \ k3 \ k4]$  は状態フィードバックゲインで、LQR 最適制御により決定できる。この状態フィードバックゲインは、Scilab<sup>4)</sup> の関数 lqr を使用して求めた。

今回の制御対象では、状態変数の  $z, \theta$  は直接観測できないので、オブザーバを使用して推定する。このときオブザーバは、式(8)で表せる。

$$\dot{\hat{x}} = (A - HC)\hat{x} + Bu + Hy \quad (8)$$

オブザーバゲインのHは、Scilabの関数ppolを使用し、オブザーバの極指定により決定できる。なお、分離定理により、状態フィードバックゲインはオブザーバの設計とは独立に決定してよいので、lqr で求めたものを使用する。表2を基にXcosによりオブザーバを利用した状態フィードバックのシミュレーションプログラムを作成すると図3となる。なお、状態フィードバックゲインは、 $[3.16 \ 13.11 \ 6.73 \ 2.38]$ である。

### 3.3 PID 制御器の設計とシミュレーション

3.2 で求めた、状態フィード制御系を基にPID制御器の構造と初期パラメータを決定する。台車目標位置を  $z_r$ 、偏差を  $e$  とすると  $e = z_r - z$  より、 $z_r = 0$  とすることで、式(7)は式(9)に書き換えることができる。

$$u = -k1e - k3 \frac{de}{dt} + k2\theta + k4 \frac{d\theta}{dt} \quad (9)$$

式(9)の前の2項は、台車の位置に対するPD制御であり、後の2項は振り角度に対するPD制御と解釈できる。

以上より、台車に用いる制御器の伝達関数は式(10)であり、式(10)を双一次Z変換を用いて離散系の伝達関数に変換すると式(11)で表せる。ただし、 $C1(s)$ を安定な制御器とするため、極を調整している。

$$C1(s) = \frac{U1(s)}{E(s)} = -k1 - k3s \cdot \frac{100}{s+100} \quad (10)$$

$$C1(z) = -\frac{\beta z + \alpha}{z + \gamma} \quad (11)$$

サンプリング周期を  $T=100$  [ $\mu$ s] とすると,  
 $\alpha = 672.78, \beta = 672.81, \gamma = -0.99$  と求まる。  
 さらに式(11)を差分方程式で表すと式(12)となる。

$$u1(n) = -\alpha e(n-1) - \beta e(n) - \gamma u1(n-1) \quad (12)$$

この式(12)が本システムの台車に対するデジタル制御演算則である。同様に振子に対するデジタル制御演算則を求めると式(13), 伝達関数は式(14)になる。

$$u2(n) = \kappa \theta(n-1) + \lambda \theta(n) - \mu u2(n-1) \quad (13)$$

$$\kappa = -249.80, \lambda = 249.93, \mu = 0.99$$

$$C2(z) = \frac{\lambda z + \kappa}{z + \mu} \quad (14)$$

また, 式(9)より, 操作量は式(15)となる。

$$u(n) = u1(n) + u2(n) \quad (15)$$

Xcos によるシミュレーションプログラムを図4に示す。また, 図5に台車目標位置  $z_r$  を  $0.05$  [m] 変化させた時の操作量  $u$ , 台車位置  $z$ , 倒立振子角度  $\theta$  のシミュレーション結果を示す。

シミュレーション結果は, 状態フィードバック制御の場合とほぼ等価な動作が実現できていることを示している。この設計手法によれば, 比較的簡易に構造が簡単なデジタルPID制御器が設計できる。しかしながら, 制御対象の物理モデルが明確でない場合やモデル化誤差が大きい場合は, この方法では十分ではない。これに対処するためには, 制御器パラメータをCPUから調整する機構は不可欠である。

#### 4. FPGA への実装と評価

3. で述べた制御器の設計, 及びシミュレーションより, 状態フィードバック制御と等価な動作をデジタルPID制御で実現する場合, サンプリング時間が重要であることが分かる。従って, コンピュータでディ

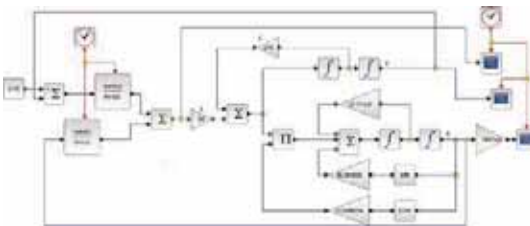


図4 倒立振子デジタルPID制御のシミュレーションプログラム (Xcos)

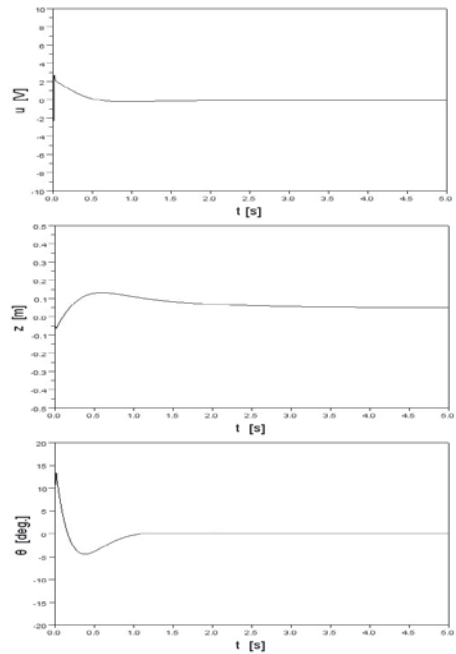


図5 倒立振子のデジタルPID制御 (シミュレーション)

ジタルPID制御を実装する場合, コンピュータのCPU性能が大きく影響することになる。FPGAによりデジタルPID制御器を実装した場合, その演算時間は非常に短く, また並列に実行されるため, サンプリング時間の影響を小さくできる<sup>5)</sup>。

#### 4.1 A/D, D/A 変換制御部

A/D変換器として使用しているMCP3204のタイミングチャートを図6に示す。MCP3204はSPI互換シリアルインターフェースを使用しており,  $\overline{CS}$  をLowにすることで通信を開始する。 $\overline{CS}$  がLowになると4ビットの制御ビット選択信号を変換器の  $D_{IN}$  へ出力する。これにより入力チャンネル構成を選択する。AD変換器は開始ビットを受け取った後, CLKの4番目の立ち上りでアナログ入力のサンプリングを開始し, 5番目のCLKの立ち下りでサンプリング期間を終了する。制御ビット選択信号の部分のVHDLは以下の記述となる。

```

if(CLK'event and CLK = '0) then  --制御ビット出力
  if(CS_ad = '0)then             --CSがLowか
    Din_ad <= Din_ad(3 downto 0) & '0;  --制御ビット選択出力
  else
    Din_ad <= SEL;                --制御ビット選択(SEL)
  end if;
end if;

```

D0 ビットが入力されると変換器は次のクロックで Low の Null Bit を出力し、その後、入力電圧を 12 ビットのデジタル値に変換した値を MSB から 1 ビットずつシリアルに出力していく。この信号を 1 ビットずつ取り込む VHDL 記述を以下に示す。

```

when S3 =>
  if (clk_count = "1100") then
    ad_finish <= '1';
    RD_STATE <= S4;
  else
    Dout_ad <= Dout_ad(10 downto 0) & Dout_bitin;
    clk_count <= clk_count + '1';
    RD_STATE <= S3;
  end if;

```

D/A 変換器 DAC7611 のインタフェースは、SSI (Synchronous Serial Interface) で、そのタイミングチャートを図 8 に示す。A/D 変換器のインタフェース回路と同様に VHDL で記述する。

### 4.2 デジタル PID 制御器部

本実験での制御対象は、1 入力 (u) 2 出力 (z,  $\theta$ ) であり、操作量 u は、式(12)、式(13)、式(15)となる。FPGA 上への実装では、式(12)と式(13)を実現する 2 組の PID 制御器を作製し、その出力 u1, u2 の和を操作量 u とする。この様に複数の PID 制御器を実装することは、

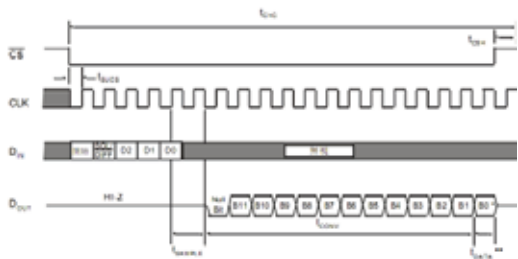


図 6 A/D 変換器の動作タイミングチャート

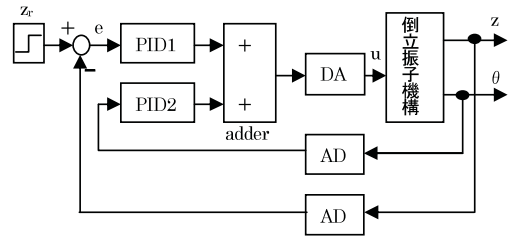


図 7 制御ブロック図

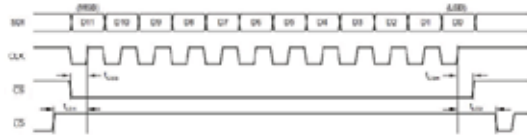


図 8 D/A 変換器の動作タイミングチャート

多軸機構の制御に展開できることを示している。特に、CPU による制御演算ではなく、独立した回路による並列制御演算となるため、演算時間に基づくサンプリング時間の設定を設計し易い。

図 7 に FPGA への実装を中心とした制御ブロック図を示す。

各制御器の係数 ( $\alpha, \beta, \gamma, \kappa, \lambda, \mu$ ) は、FPGA 上の CPU から与える。操作量 u の算出の部分の VHDL 記述を以下に示す。

```

SET_1:set_com
portmap(set_enable1,Xn,coef1,add_input,result1);
--coef1×X(n)+add_input(値 '0')を result1 に格納
SET_2:set_com
Portmap(set_enable2,Xn_1,coef2,result1,result2);
--coef2×X(n-1)+result1 を result2 に格納
SET_3:set_com
portmap(set_enable3,Yn_1,coef3,result2,result3);
--coef3×Y(n-1)+result2 を result3 に格納

```

set\_com にはそれぞれ、乗算器、加減算器、また桁あふれが生じたときに 32 ビットにクリップする回路がコンポーネントとして結合してある。乗算器、加減算器の VHDL 記述の一部を以下に示す。演算は、固定小数点 (整数部 10bit, 小数部 5bit, 符号 1bit) で実装した。

```

entity mul is
  generic (Int : integer:=10; De : integer:=5);
  port (A_IN : in std_logic_vector( (Int+De+1) - 1 downto 0);
        B_IN : in std_logic_vector( (Int+De+1) - 1 downto 0);
        C_OUT : out std_logic_vector( (2*(Int+De+1)) - 1 downto 0));
end mul;
architecture Behavioral of mul is
begin
  C_OUT <= signed(A_IN)*signed(B_IN);
end Behavioral;

```

```

entity add is
  generic (Int : integer:=10; De : integer:=5);
  port (A_IN,B_IN: in std_logic_vector(2*(Int+De+1) - 1
    downto 0);
        C_OUT : out std_logic_vector(2*(Int+De+1) downto 0);
        EN : in std_logic);
end add;
architecture Behavioral of add is
  signal cout : std_logic_vector(2*(Int+De+1)-1 downto 0);
  process(A_IN,B_IN,EN)
  begin
    if (EN = '1') then
      cout <= signed(A_IN) + signed(B_IN);
    end if;
  end process;
end Behavioral;

```

また、制御器には遅延器が必要なため、現在の値  $u(n)$  をレジスタに格納しておき、次のサイクルで  $u(n-1)$  として読み出すことで実現する。  $e(n), \theta(n)$  についても

同様に、それぞれレジスタを指定することで読み出せるようにする。

### 4.3 制御の実行

CPU からデジタル PID 制御器にパラメータが付与されると、CPU は A/D 変換制御部に動作開始命令を出す。A/D コンバータは SPI シリアルインタフェースを使用し、チャンネル選択をして台車と振子の位置情報を取り込む。取り込みが完了すると A/D 変換された値はそれぞれの制御器に伝送され、偏差の算出をおこない操作量を求める。制御演算の算出は並列に処理が行われる。両方の制御器から操作量が出力されるのを待ち、出力が完了するとそれらの和を  $u(n)$  とし D/A 変換制御部に転送する。D/A 変換制御部は D/A コンバータを動作させ、入力された操作量を実際に制御対象に出力する電圧値に変換し制御対象へ伝送する。その後、サンプリング周期調整部がサンプリング周期に応じて次のサンプル点までのクロックを一定数カウントし、カウントが終了すると、CPU へ演算終了の割り込み信号を送り 1 サンプル周期分の制御を終了する。なお、全体のクロック周波数は 1MHz である。

### 4.4 使用ロジックユニット消費量の評価

FPGA (XCS700A) における、デジタル PID 制御器 1 つあたりのロジックユニット消費量を表 3 に示す。また、表 4 に FPGA 上に図 6 の構成を実装したときのロジックユニット消費量を示す。表 3 からは、デジタル制御器が最も大きな割合で消費するのは BUFGMUX (グローバルバッファ) であるように読み取れる。しかし、BUFGMUX はクロック信号の遅延を軽減するためのものであり、FPGA 内の配置配線によって必要量が変わるため、BUFGMUX の消費量から FPGA 上に搭載可能デジタル制御器の個数を予測することはできない。図 6 の構成での実装結果から、FPGA 上に搭載可能なデジタル制御器の個数を検討した。表 4 より、FPGA 上には 1 つの CPU に加え、最大で 4 つのデジタル制御器を搭載可能なことが分かった。

現在、FPGA の大容量化は益々加速しており、CPU と複数のデジタル制御器を搭載する統合型制御器の実

表 3 デジタル PID 制御器 1 つのロジックユニット消費量

Logic Utilization	Used	Available	Utilization
Number of occupied Slices	129	5,888	2%
Total Number Slice Registers	211	11,776	1%
Total Number of 4 input LUTs	181	11,776	1%
Number of BUFGMUXs	4	24	16%
Number of MULT18X18SIOs	3	20	15%

表 4 図 7 の構成でのロジックユニット消費量

Logic Utilization	Used	Available	Utilization
Number of occupied Slices	2,788	5,888	47%
Total Number Slice Registers	3,165	11,776	26%
Total Number of 4 input LUTs	3,561	11,776	30%
Number of BUFGMUXs	12	24	50%
Number of MULT18X18SIOs	9	20	45%

装は十分に実用的である。今回例示した、制御器の設計手順は、更に、それぞれのデジタル制御器に対する制御器パラメータの付与、制御応答結果の記録などを、リアルタイム OS を用いて一つのタスクとして CPU に行わせることができれば、本システムを多入力多出力システムに応用することが可能である。

## 5. まとめ

現在、制御アルゴリズムのソフトウェア化が、一般的な制御器実装手法となっているが、高速、高精度な機構制御には、制御器のハードウェアによる実装技術が重要となると考えられる。

本論文では、FPGA 上に複数の PID 制御器と制御器のパラメータ調整などに使用する CPU を搭載した機構制御用の制御器構成を提案した。また、実装実験により必要なロジックユニット量を見積もった。制御性能の評価と制御器パラメータ調整の手法開発が今後の課題である。

## 謝辞

本研究は、財団法人メカトロニクス技術高度化財団の研究助成の成果です。研究助成に対し深く感謝の意を表します。

## 文献

- 1) 山口高司, 平田光男, 藤本博志: ナノスケールサーボ制御, 東京電機大学出版局, (2007)
- 2) Dogan Ibrahim: Microcontroller Based Applied Digital Control, John Wiley & Sons, Ltd, (2006)
- 3) Karl J. Astrom and Tore Hagglund: Advanced PID Control, ISA, pp.293-328, (2006)
- 4) Scilab: <http://www.scilab.org/> 2011.09.01 確認
- 5) 竹光考昭, 中澤雅史, 山田健仁: FPGA によるデジタル PI 温度制御器の統合調整システム, 電気・情報関連学会中国支部第 60 回連合大会, (2009)

(2011.9.16 受理)