論　文

FPGA を用いた高精度微分器の実現

金田　泰昌＊1）　　入月　康晴＊1）

Realization of a High Precision Differentiator Using FPGA

Yasuaki Kaneda[*1)], Yasuharu Irizuki[*1)]

In this paper, we reduce velocity sensors from control systems by implementing velocity estimators in FPGA. In order to implement other peripherals in FPGA, we require differentiators with small circuit size and high precision. Recently, a pseudo differentiator has been proposed with a very simple structure that could also have a small circuit size. However, this pseudo differentiator has large discretization errors. In this paper, we reduce the discretization errors of the pseudo differentiator through Richardson extrapolation (RE) and fractional delay (FD). In general, FD is implemented by approximate methods, such as Lagrange FIR interpolators. In this paper, we show that the implementation of FD is equivalent to the realization of a high sampling rate (HSR) system, and we realize the HSR system using FPGA. Numerical simulations and experiments show the effectiveness of the proposed differentiator.

キーワード：微分器，速度推定，リチャードソン補外，非整数遅延，FPGA

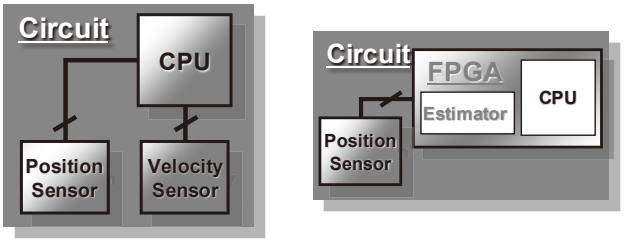
Keywords：Differentiator, Velocity estimation, Richardson extrapolation, Fractional delay, FPGA

1. はじめに

制御システムを構築するためには，CPU をはじめ，センサなど様々な周辺 IC を必要とする。その周辺 IC の中でも，速度センサはフィードバック制御を行う上で欠かすことができない重要な IC の一つである。しかしその一方で，速度センサは高精度なものは比較的高価であり，制御システムのコストを上げる原因の一つにもなっている。また物理的な制約で取り付けが困難なケースも存在する。例えば小型のロボットを考えた場合，各関節にエンコーダーとジャイロセンサを取り付けることは，スペース等の物理的な制約や，重量的な問題で困難である場合もある。

この問題に対して従来までは，CPU で速度推定器を実行し，位置から速度を推定している。仮にこの速度推定器を回路として FPGA で実現することができれば，CPU をはじめ速度推定器を一つの IC で実現することができるため，部品点数・実装面積の削減が可能となる。また，CPU の負荷を下げることができるため，同じスペックの CPU でより高度な制御アルゴリズムを実行することができる（図 1）。

これまでに，数多くの速度推定器が提案されている[(1)(2)]。最も代表的なものとして，「疑似微分器」と呼ばれるものがある。これは微分特性にローパスフィルタを加えたものである。そのため，ローパスフィルタで設定したカットオフ周波数までは微分特性を示し，カットオフ周波数以上の周波数帯域ではローパスフィルタの効果で信号の増幅を抑えることができる。そのため，ノイズに対してある程度のロバスト性を持つ。しかしながら，疑似微分器をデジタル実装するためには，疑似微分器を離散化する必要があり，その離散化誤差が大きいことが知られている。



（a）　従来までの制御系　　（b）センサレス制御系

図1.　FPGA を用いたセンサレス制御系

本研究では，疑似微分器の離散化誤差低減手法を考える。離散化誤差を低減するために，本研究ではリチャードソン補外（以下，RE）と非整数遅延（以下，FD）を FPGA で実装することを検討する[(3)]。離散化誤差はサンプリング周期の級数で表されることが知られている。RE は級数の収束を速くする効果があるため，これを離散化された疑似微分器に適用することで，離散化誤差がより速く収束することが期待される。また，FD は信号を内挿する効果があるため，大きなサンプリング周期でも精度が高くなることが知られている。しかしながら，FD を FPGA で実装するためには，通常近似が必要となる。本研究では，できるだけ近似をせずに，FD を実装する手法を検討する。

事業名　平成 23 年度　基盤研究
＊1）情報技術グループ

2. RE及び FD を用いた疑似微分器

一次疑似微分器の伝達関数は次式で与えられる。

$$D(s)=\frac{s}{Ts+1}. \tag{1}$$

ここで， T は時定数を表し，このときのカットオフ周波数は $f_c=1/2\pi T$ [Hz]となる。ゼロ次ホールドによる式(1)のパルス伝達関数は，

$$D(z)=\frac{1}{T}\frac{1-z^{-1}}{1-e^{-\frac{T_s}{T}}z^{-1}}. \tag{2}$$

ただし， T_s はサンプリング周期を表す。今， $\alpha\in[0\quad 1]$ としたとき，非整数遅延（FD）である $z^{-\alpha}$ を用いた疑似微分器のパルス伝達関数は次式で与えられる。

$$D(z,\alpha)=\frac{1}{T}\frac{1-z^{-\alpha}}{1-e^{-\frac{\alpha T_s}{T}}z^{-\alpha}}. \tag{3}$$

ここで，$\alpha=0$ 周りでのテーラー展開を考えと，式(3)は次式として表すことができる[3]。

$$\begin{aligned} D(z,\alpha)\big|_{z=e^{j\omega T_s}} &= \frac{j\omega}{j\omega T+1}+\sum_{k=1}^{\infty}g_k\alpha^k \\ &= \frac{j\omega}{j\omega T+1}+O(\alpha). \end{aligned} \tag{4}$$

ただし，g_k は T 及び T_s の関数である。式 (4) の右辺第一項は，式(1)の連続時間系の伝達関数を表し，第二項が誤差項を表す。つまり式(4)は，疑似微分器の離散化誤差は非整数 α に依存し， α が小さいほど離散化誤差が小さくなることを意味する。このため，FD を導入することで，離散化誤差を低減することができる。

次に RE を用いた疑似微分器を導出する。式 (4) より次の二つの式が得られる。

$$D(z,\alpha)\big|_{z=e^{j\omega T_s}}=\frac{j\omega}{j\omega T+1}+g_1\alpha+g_2\alpha^2+\cdots, \tag{5}$$

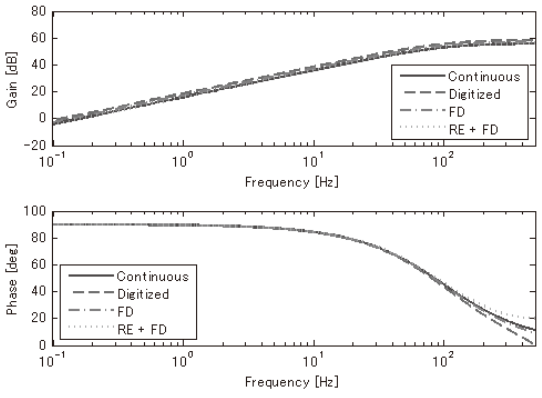
$$D(z,2\alpha)\big|_{z=e^{j\omega T_s}}=\frac{j\omega}{j\omega T+1}+2g_1\alpha+4g_2\alpha^2+\cdots. \tag{6}$$

よって，α に関する項を次のように削除することができる。

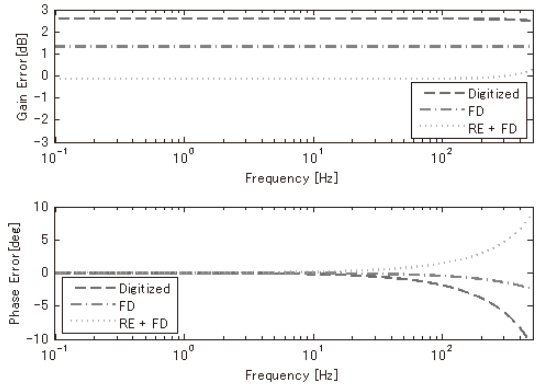
$$\begin{aligned} D^1(z,\alpha)\big|_{z=e^{j\omega T_s}} &= 2D(z,\alpha)\big|_{z=e^{j\omega T_s}}-D(z,2\alpha)\big|_{z=e^{j\omega T_s}} \\ &= \frac{j\omega}{j\omega T+1}-2g_2\alpha^2-6g_3\alpha^3+\cdots \\ &= \frac{j\omega}{j\omega T+1}+O(\alpha^2). \end{aligned} \tag{7}$$

式(7)における $D^1(z,\alpha)$ が RE を用いた疑似微分器であり，誤差項の次数が増えていることがわかる。すなわち，離散化誤差が少なくなり，離散化精度が高くなっていることがわかる。

時定数をカットオフ周波数が 100Hz となるように $T=0.0016$ とし，また $\alpha=0.5$ としたときの疑似微分器のボード線図を図 2に示す。ゲインにおいて，FD を適用することで疑似微分器の離散化誤差が全帯域で小さくなり，また RE を適用することでさらに改善されていることがわかる。位相もまた，FD を適用することで改善されている。ただし，RE を適用した場合，ナイキスト周波数付近で誤差が大きくなっていることがわかる。このように FD と RE の両者を適用した場合，位相の一部分で誤差が出るものの，ゲインは全帯域で改善しており，総合的には優れていると言える。



(a)　ボード線図



(b)　理想ボード線図との誤差

図2.　疑似微分器(Digitized)，FD を用いた疑似微分器(FD)，　FD 及びREを用いた疑似微分器(RE+FD)のボード線図

3. FD の実装手法

3. 1　Lagrange 補間を用いた近似手法（従来手法）　通常，FD の実装には次式の Lagrange 補間が用いられる。

$$z^{-(l+\alpha)}\approx\sum_{n=0}^{L}h_n(\alpha)z^{-n}. \tag{8}$$

ここで，l は正の定数であり，非整数 α が非常に小さい場合は $z^{-\alpha}$ を z^{-l}/z^{-l} でスケーリングする必要がある[3]。L は Lagrange 補間の次数を表わす。また $h_n(\alpha)$ は次式で与えられる。

$$h_n(\varepsilon)=\prod_{l=0,l\neq n}^{L}\frac{l+\alpha-l}{n-l}. \tag{9}$$

FD の代わりに Lagrange 補間を用いた場合のボード線図を図 3に示す。ここで $l=30$， $L=50$ である。この結果より，Lagrange 補間を用いた場合は，低周波数領域では離散化誤差が低減されているが，高周波数帯域では離散化誤差が非常に大きくなっていることがわかる。さらに，Lagrange 補

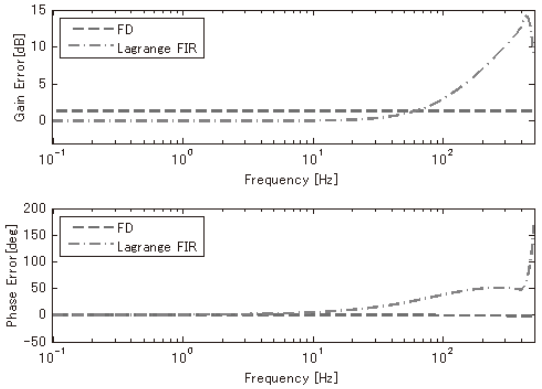


図3.　Lagrange 補間を用いた場合(Lagrange FIR)の，理想ボード線図との誤差

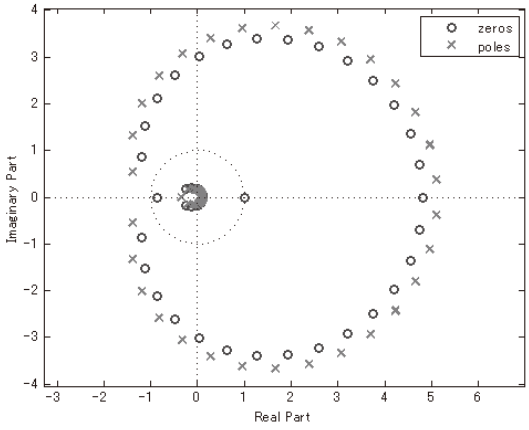


図4.　Lagrange 補間を用いた場合の極(poles)と零(zeros)の配置

間を用いた場合のパルス伝達関数の極零配置を図 4に示す。この結果より，Lagrange 補間を用いた場合のパルス伝達関数は，単位円外に多く極を持つため，不安定なシステムになっていることがわかる。

3. 2　高速サンプリング周期システムによる実現手法

遅延演算子 z^{-1} について考える。今，サンプリング周期 T_s における遅延演算子を z_s^{-1} とする。このとき，

$$z_s^{-\alpha} = e^{-j\omega\alpha T_s} = e^{-j\omega T_\alpha} := z_\alpha^{-1}. \tag{10}$$

ここで，$T_\alpha = \alpha T_s$ である。式(10)は，サンプリング周期 T_s における FD $z_s^{-\alpha}$ は，新しいサンプリング周期 T_α における遅延 z_α^{-1} と等価であることを意味している。ここで α は 1 以下であるため，FD を実装することは，速いサンプリング周期を用いることと等価であることを意味する。

今，サンプリング周期 $T_{2\alpha}$ のときの遅延演算子を $z_{2\alpha}^{-1}$ とすると，式(7)は次式として与えられる。

$$D^1(z,\alpha) = 2D(z_\alpha) - D(z_{2\alpha}). \tag{11}$$

これは，サンプリング周期 T_α，$T_{2\alpha}$ のマルチレートシステムを意味する。一方，z_α^{-1} と $z_{2\alpha}^{-1}$ との間には以下の関係が成り立つ。

$$z_{2\alpha}^{-1} = e^{-j\omega(2\alpha T_s)} = \left(e^{-j\omega\alpha T_s}\right)^2 = z_\alpha^{-2}. \tag{12}$$

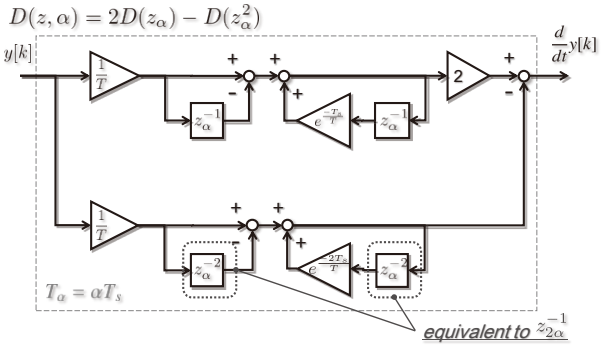


図5.　提案微分器のアーキテクチャ

表1. ハードウェアの仕様

Hardware	Specifications
FPGA	Altera Cyclone III (EP3C25F324C8N)
Clock	50MHz
A/D	8ch 13bit 1Msps (±10 V)
D/A	8ch 12bit 2Msps (±10 V)

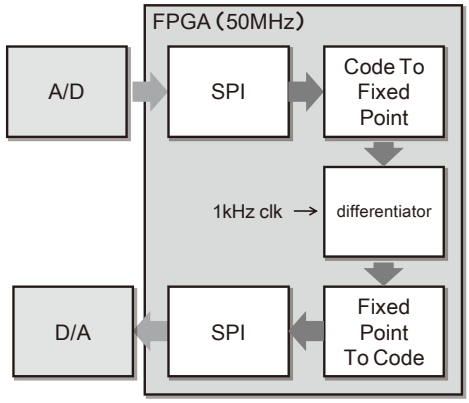


図6.　主要コンポーネント

表2.　実験条件

Input signal of SG	Type	Sinusoidal wave
	Frequency	10Hz
	Amplitude	1.0V
	Variance	2.0×10^{-3}
Original sampling rate, T_s		1ms
Number of FD, α		0.5
Length of fixed point		64bits (fraction: 32bits)

よって，

$$D^1(z,\alpha) = 2D(z_\alpha) - D(z_\alpha^2). \tag{13}$$

この式はサンプリング周期 T_α のシングルレートシステムであり，マルチレートシステムに比べて実装が容易となる。シングルレートシステムのアーキテクチャを図 5に示す。

4.　実験

4. 1　条件　本研究では，高速サンプリング周期システムを実現するために FPGA を利用する。使用する FPGA ボードの仕様を表 1に，FPGA 内部の主要コンポーネントを図 6に示す。FPGA として Altera 社の Cyclone III を用い，SPI 通信を用いて A/D で信号を取り込み，固定小数点に変換した後，微分器にデータを送る。微分器より推定された微分値をコードへ変換し，SPI 通信を用いて D/A で出力する。

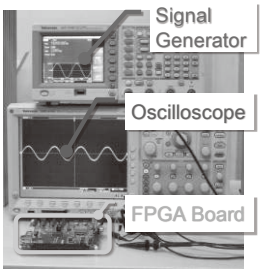


図7. 実験環境

表3. ハードウェアリソース

	OSR/HSR	RE + HSR
Total LEs	3922 (16%)	4425 (18%)
Total registers	1494 (6%)	1685 (7%)
9k bit dual port memory unit (M9k)	6 (9%)	6 (9%)
Total memory bits	51712 (9%)	51712 (9%)
Embedded multiplier 9bit elements	86 (65%)	102 (77%)

実験環境を図 7に，実験条件を表 2に示す。10Hz の正弦波をシグナルジェネレータから発生させ, FPGA ボードに入力する。そして FPGA ボードから出力された推定信号をオシロスコープで取得し，理想信号との誤差を比較する。

4. 2 結果 表 3に実装に必要なハードウェアリソースを示す。高速サンプリング周期（HSR）を用いた場合と，元々のサンプリング周期（OSR）を用いた場合とでは，クロックが異なるだけで，ハードウェアリソースには変化が無い。一方，RE を適用することで，若干ハードウェアリソースが増加する。

図 8に微分値の推定結果の時間応答を，表 4にそのときの二乗平均誤差を示す。この結果より，疑似微分器のみでは離散化誤差が大きいため推定誤差も大きくなっているが，FD を用いることで推定誤差が小さくなり，さらに RE を用いることでより性能が改善されていることがわかる。

5. まとめ

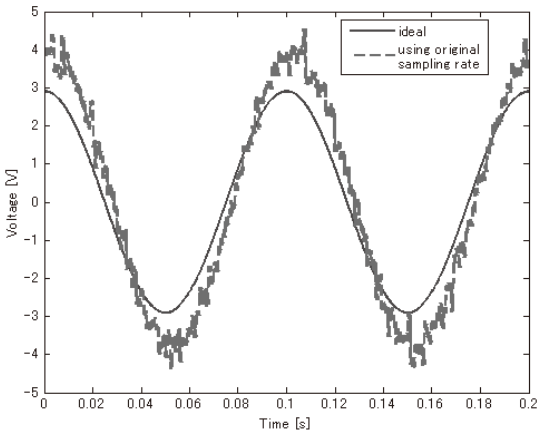
本論文では，RE 及び FD を用いることで，疑似微分器の離散化誤差を低減した。また FD の実装は高速サンプリング周期システムの実現と等価であることを示し，FPGA を用いて実現した。これにより，安定でかつハードウェアリソースの少ない高精度な微分器を実現した。

今後の課題は，ハードウェアリソースのさらなる低減や，よりノイズにロバストな推定器の開発があげられる。

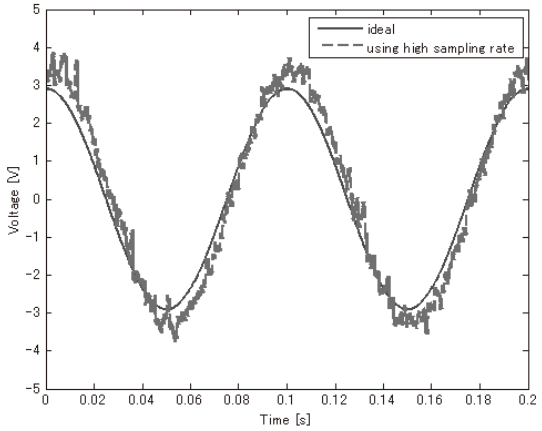
（平成 24 年 5 月 18 日受付，平成 24 年 8 月 1 日再受付）

表4. 推定結果の二乗平均誤差

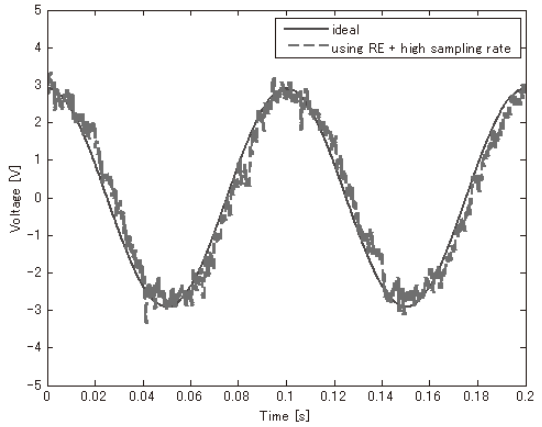
OSR	HSR	RE + HSR
0.89088	0.53118	0.42219



(a) 疑似微分器による推定結果（OSR）



(b) FD を用いた疑似微分器による推定結果（HSR）



(c) RE 及び FD を用いた疑似微分器による推定結果（RE+HSR）

図8. 実験結果

文 献

(1)M. A. Al-Alaoui : “Novel Digital Integrator and Differentiator,” *IEE Electronics Letters*, Vol. 29, No. 4, pp. 376-378 (1993).

(2)C. C. Tseng, et al. : “Design of Digital Differentiator Using Difference Formula and Richardson Extrapolation,” *IET Signal Process.*, Vol. 2, No. 2, pp. 177-188 (2008).

(3)Y. Kaneda, et, al. : “FPGA Implementation of Digital Differentiator Using Richardson Extrapolation and High Sampling Rate Acting Like Fractional Delay,” Proc. of SICE Annual Conf. (2011).