Vol 53 No 4, November 2019

ANALOG DEVICES
想像を超える可能性を
AHEAD OF WHAT'S POSSIBLE™

AnalogDialogue

モータ・コントロール向けに最適化されたΣΔ ADCによる電流測定【Part 2】

著者：Jens Sorensen、システム・アプリケーション・エンジニア
Shane O'Meara、システム・アプリケーション・エンジニア
Dara O'Sullivan、システム・アプリケーション・マネージャ

本稿では、シグマ・デルタ型のA/Dコンバータ（ΣΔ ADC）を利用したモータ・コントロール・アプリケーションについて、2部構成で解説を進めています。今回はそのPart 2です。前回のPart 1では、モータ・コントロール・アプリケーションにおいて、ΣΔ変調によって符号化されたデータをsincフィルタで復調する方法について説明しました。その中で、sincフィルタのインパルス応答とPWM（パルス幅変調）の同期をとる（ロックする）ことが重要であるということを明らかにしました。但し、実際には、それらの同期をとれるようにシステムを適切に構成するのは容易ではありません。

Part 2では、上述した同期を容易に実現できるよう最適化された新たなsincフィルタを紹介します。それを使用すれば、フィードバック・チェーンにおいて厳密なタイミング制御を必要とするアプリケーションの測定性能を高めることができます。また、そのsincフィルタをHDL（ハードウェア記述言語）でコーディングする際の注意点と、そのコードをFPGAに実装する場合の最適化手法について解説します。更に、そのようにして実装したFPGAをベースとする3相サーボ・ドライブの性能を測定した結果を示します。

同期に向けて最適化されたsincフィルタ

Part 1では、ΣΔ ADCを使用して測定を行う場合、sincフィルタのインパルス応答とPWM周期の同期をとることで、エイリアスの発生を抑えられることを明らかにしました。この方法の考え方自体は単純明快なものです。しかし、望ましい結果が得られるようにシステムを構成するのは難しく、実現が不可能なことも少なくありません。ここで、sincフィルタとPWM回路がシステム・クロック源を共有しており、共に周波数f_{sys}で動作するケースを考えます。ΣΔ ADCが内蔵する変調器で使われるクロックf_{mclk}は、次式で表されます[1]。

$$f_{mclk} = \frac{f_{sys}}{D_{mclk}} \quad (1)$$

ここで、D_{mclk}は変調器のクロックの分周比です。同様に、PWM処理の周波数f_{pwm}は、次式で表されます。

$$f_{pwm} = \frac{f_{sys}}{D_{pwm}} \quad (2)$$

D_{pwm}は、PWMの周波数を定めるクロック分周比です。

sincフィルタのデシメーション・レート（データ・レート）は、以下の式で決まります。

$$f_{dec} = \frac{f_{mclk}}{D_{dec}} = \frac{f_{sys}}{D_{mclk} \times D_{dec}} \quad (3)$$

D_{dec}は、デシメーション・クロックの分周比です。ここで、インパルス応答とPWM周期の間でドリフトが生じるのを防ぐためには、PWM周期内のデシメーション・サイクル数は整数でなければなりません（以下参照）。

$$\frac{f_{dec}}{f_{pwn}} = N \quad (4)$$

ここで、Nは整数です。式 (2)、式 (3)、式 (4) を組み合わせると、以下の式が得られます。

$$\frac{D_{pwm}}{D_{mclk} \times D_{dec}} = N \quad (5)$$

この式を満たすクロック分周比の組み合わせは、明らかに限られています。また、ほとんどの場合、クロック分周比の選択には、厳しい制約が課せられます。例えば、PWM周期や変調器のクロックは、システムとして特定の値（それぞれ10kHzや20MHzなど）に定められていることがあります。もう1つの問題は、変調器のクロックの選択肢が限定されることです。例えば、f_{sys}が100MHzである場合、D_{mclk}の値として選択できるのは、5～10の整数（20MHz～10MHz）に限られます。

こうした制約を考慮すると、インパルス応答とPWM周期の同期を望ましい形で実現できるクロック分周比を見つけるのは、不可能とまではいかないまでも、非常に難しいことがわかります。一般的には、望ましいPWM周期、変調器のクロック、S/N比が得られるように分周比を選択するということではなく、式（5）を満たせる分周比を何とか見つけ出し、それを選択するしかないという状況になるでしょう。また、時間の経過に伴っていずれかの周波数が変化すると、適切な構成は実現できなくなります。しかし、1つのモーション・コントローラがネットワーク内の複数のモータ・コントローラと同期をとる多軸システムでは、時間の経過に伴って周波数を変更するというのはよくあることです。

インパルス応答とPWM周期の同期を実現できれば、優れた測定性能が得られます。しかし、この方法は実用的ではない可能性があります。次のセクションでは、この問題に対処できる新たな種類のsincフィルタを紹介します。そのフィルタを使用すれば、最適な測定性能が得られると共に、あらゆるクロック分周比をそれぞれ個別に選択できます。

フラッシング型sincフィルタ

図1は、従来の3次sincフィルタの概念図です。このフィルタでは、システム・クロックをスケーリングすることによって、ADCの変調器で使用するクロックを生成します。一方、ADCの変調器は、アナログ信号を1ビットのデジタル値に変換し、そのデータ・ストリームをフィルタに受け渡します。フィルタの機能そのものは、変調器と同じクロック・レートで動作する3つの積分器とデシメーション・レートで動作する3つの微分器で実現されます。積分器の伝達関数は$1/(1 - z^{-1})$、微分器の伝達関数は$(1 - z^{-1})$です。3つの積分器と3つの微分器は、それぞれカスケード接続されています。

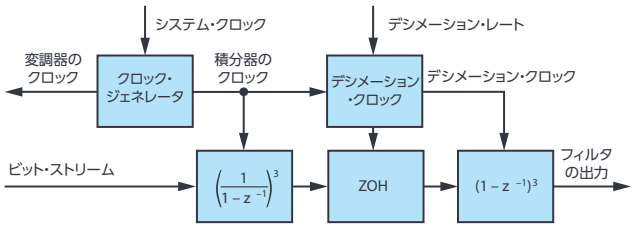

図1. 従来の3次sincフィルタ（連続動作型sincフィルタ）

変調器とsincフィルタは、それぞれが必要とするクロックを連続的に印加することによって動作します。その結果、フィルタはデシメーション・クロックに基づく固定レートでデータを連続的に出力します。通常、フィルタからのデータ・レートは、モータ・コントロールに使用するアルゴリズムの更新レートよりも高いので、フィルタ出力のうち多くは破棄されることになります。インパルス応答の中心が理想的な測定ポイントと合致した場合のみ出力が取得され、フィードバック用の情報として使用されます。

空間ベクトル変調を使用する場合、位相電流が平均値になるのはPWM周期ごとに2回しかありません。したがって、エイリアスを生じさせないsincフィルタのデータ出力は、PWM周期ごとに2つしか存在し得ないということになります。また、これは、フィルタ処理は連続的に実行する必要はないということを意味します。実際には、フィードバックが必要なときだけ測定を有効にし、それ以外は測定を無効にしておいて構わないということです。このことから、従来のADCとは異なり、オン／オフ・モードで動作させて測定を行う手法を適用できるという結論が得られます。

オン／オフ・モードの動作を実現するためには、従来の回路に対し、変更を加えなければなりません。従来の回路では、変調器とフィルタのクロックが同じシステム・クロックに基づいて動作していました。これをそのまま踏襲すると、変調器とフィルタの両方がオン／オフ・モードで動作することになります。そうすると、ADCの性能が低下してしまいます。ADCの変調器は、特定のセトリング時間とダンピング特性を備える高次システムであるからです。この構成では、ADCに最初にクロックを印加するときに、変調器がセトリングしていなければ、信頼できる出力ビット・ストリームを得ることはできません。

上記の問題を解消する新たなフィルタの構成を図2に示しました。本稿では、図1に示した従来のフィルタを連続動作型sincフィルタ、図2の新たなフィルタをフラッシング型sincフィルタと呼ぶことにします。ご覧のように、フラッシング型sincフィルタの中核部分は、連続動作型sincフィルタと同様に、3つの積分器と3つの微分器をカスケード接続したもので構成されています。それとは別の部分に、新しい動作モードを実現するための複数の工夫が盛り込まれています。1つは、新たなクロック生成回路です。この回路では、変調器のクロックと積分器のクロックを独立させています。それにより、ADCの変調器には連続的にクロックを供給しつつ、積分器のクロックは測定値を取得するときだけ有効にするという動作を実現します。もう1つの工夫は、新たなフィルタ制御回路を設けていることです。

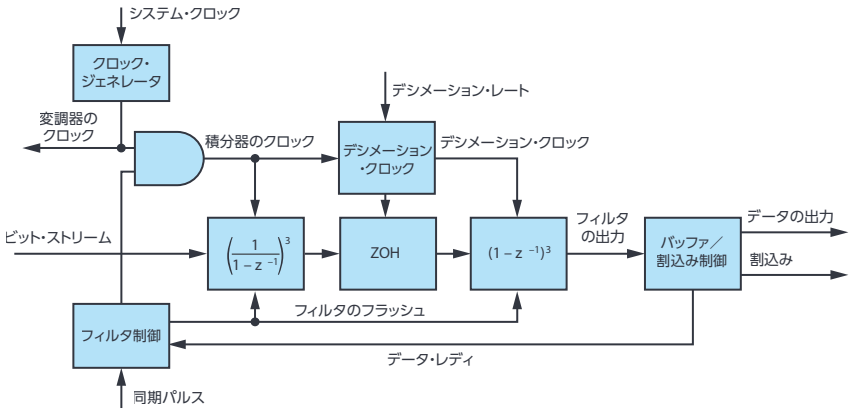

図2. オン／オフ・モードの動作に対応するフラッシング型sincフィルタ。
すべての状態をフラッシュするように設計されています。

この回路は同期パルスを基準とし、フィルタの動作に必要なすべてのタイミングとトリガを処理します。この回路の主要な機能は、フィルタのフラッシュ（flush）です。これには、すべてのフィルタの状態を初期化し、測定の開始に先立ってフィルタのタイミングを設定して、正しいタイミングで積分器のクロックを有効／無効にする処理が含まれます。最後の工夫は、新たなバッファ／割込み制御回路を設けていることです。この回路により、出力データをふるい分けて正しい測定値を取得します。同回路は、割込みにより、新たな測定値を取得できる状態になったことをモータ・コントロール・アプリケーションに通知する処理も担います。

図3に、このフラッシング型sincフィルタの動作を表すタイミング・チャートを示しました。測定を開始する際には、フィルタ制御回路に同期パルスが印加されます。通常、このパルスは、新たなPWM周期の開始を表します。同期パルスは、タイマーをスタートする処理も担います。タイマーは、望ましい測定ポイントの1.5デシメーション・サイクル分より前のタイミングで切れるように構成しておきます。タイマーが切れた時点で、積分器のクロックとデシメーション・クロックが有効になり、フィルタの処理が開始します。バッファ／割込み制御回路は、3デシメーション・サイクル後（3次sincフィルタのセトリング時間に相当）に出力データを取得し、割込み信号を発行します。図3において、測定は同期パルスに同期した状態で行われている点に注目してください。次の同期で同じシーケンスが繰り返されますが、変調器のクロックは、いったんフィルタの処理が開始されるオンのままになります。

この新たなフラッシング型sincフィルタにより、従来の連続動作型sincフィルタが抱える同期の問題が解消されます。フラッシング型sincフィルタとその動作モードは、PWM周期、変調器のクロック、デシメーション・レートについて一切の制約を課しません。あらゆる構成のシステムに対して、同じように適切に動作します。例えば、PWM周期が時間の経過と共に変化したとしても、問題なく機能します。また、このフィルタは、測定を実施するたびにリセットされます。そのため、クロック間のドリフトの影響も受けません。

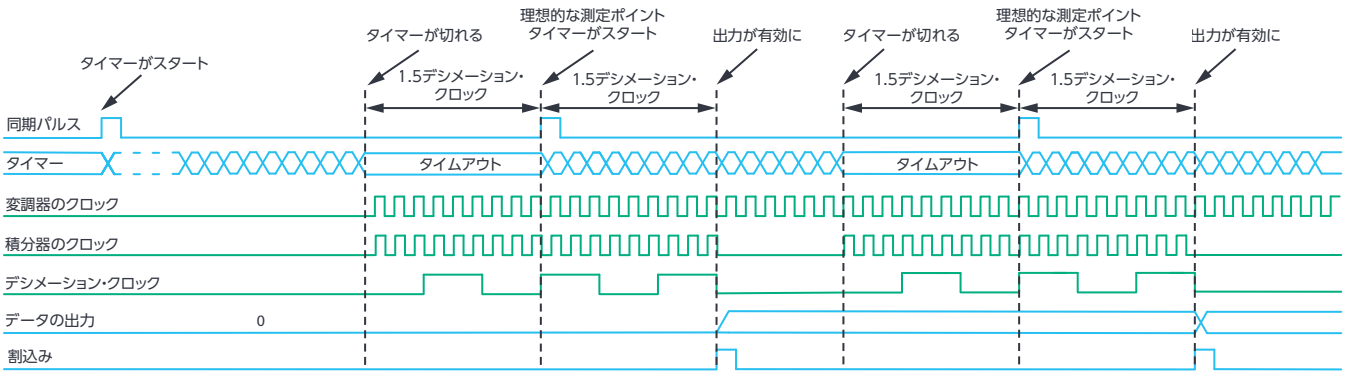

図3. オン／オフ・モードで動作するフラッシング型
sincフィルタのタイミング・チャート

HDLによる新たなsincフィルタの実装

世の中には、sincフィルタをHDLで記述した例がいくつも公開されています。それらのうちいくつかを確認してみたところ、フィルタの性能に悪影響を及ぼしたり、予期せぬ動作を引き起こしたりする可能性があるものが存在することがわかりました。以下では、フラッシング型sincフィルタを実装する際に生じるいくつかの問題を指摘します。その上で、FPGA上で最適な性能を得るためにはHDLコードをどのように記述すればよいのか解説します。

積分器

図1に示したように、sinc3フィルタの中核にあるのは、カスケード接続された3つの積分器と3つの微分器です。ここで、純粋な積分器をZ領域で表してみます（以下参照）[2]。

$$\frac{y(z)}{u(z)} = \frac{1}{1 - z^{-1}} \tag{6}$$

上式において、uは入力、yは出力です。この積分器の差分方程式は、次のようになります。

$$y[n] = u[n] + y[n-1] \tag{7}$$

この1次の式は、アキュムレータに相当します。アキュムレータは、FPGAなどにおいてはクロックド・ロジックとして実装するのに非常に適しています。図4に示すように、アキュムレータは、D型フリップフロップ（以下、DFF）を使用して実装するのが一般的です。この回路は、ごく少数の論理ゲートを使うだけでFPGA上に実装できます。

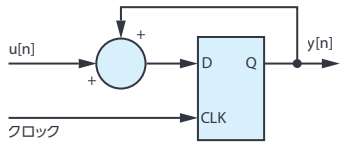

図4. DFFによるアキュムレータの実装

続いて、3つの純粋な積分器をカスケード接続すると、Z領域の伝達関数は、次式のようになります。

$$\frac{y(z)}{u(z)} = \left(\frac{1}{1 - z^{-1}}\right)^3 = \frac{1}{1 - 3z^{-1} + 3z^{-2} - z^{-3}} \tag{8}$$

カスケード接続した3つの積分器の差分方程式は、次のようになります。

$$y[n] = u[n] + 3y[n-1] - 3y[n-2] + y[n-3] \tag{9}$$

サンプルnの入力がサンプルnの出力にどのように影響を与えるかに注目してください。

図4のDFFベースのアキュムレータを使って3つの積分器を実装すると、図5のようになります。

これはクロックド回路であり、入力の変化が出力に影響を及ぼすまでには数クロックを必要とします。式（10）として、カスケード接続したアキュムレータの差分方程式を示しました。

$$\begin{aligned} y_1[n] &= u[n] + y_1[n-1] \\ y_2[n] &= y_1[n-1] + y_2[n-1] = u[n-1] + y_1[n-2] + y_2[n-1] \\ y_3[n] &= y_2[n-1] + y_3[n-1] = u[n-2] + y_1[n-3] + y_2[n-2] + y_3[n-1] \end{aligned} \tag{10}$$

これを見れば、どのような動作になるのか明確に理解できるでしょう。ご覧のように、この差分方程式は、純粋な積分器に対応する式（9）とはかなり異なります。純粋な積分器では、入力の影響が直ちに出力に現れます。それに対し、カスケード接続したアキュムレータでは、入力が出力に反映されるまでに2クロックを要します。

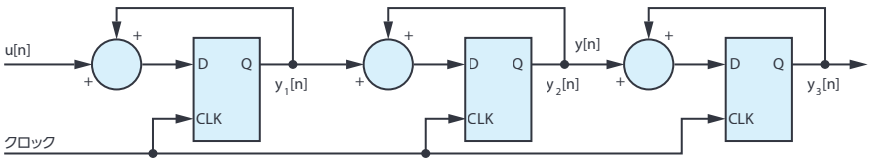

図5. DFFベースのアキュムレータを使って実装した3つの積分器

この違いを確認するために、5番目のサンプルとして単位ステップを印加した場合の式（9）、式（10）のステップ応答を図6に示しました。ご覧のように、アキュムレータをベースとする回路では、純粋な積分器をカスケード接続した場合と比べて2サンプル分の遅延が生じることがわかります。

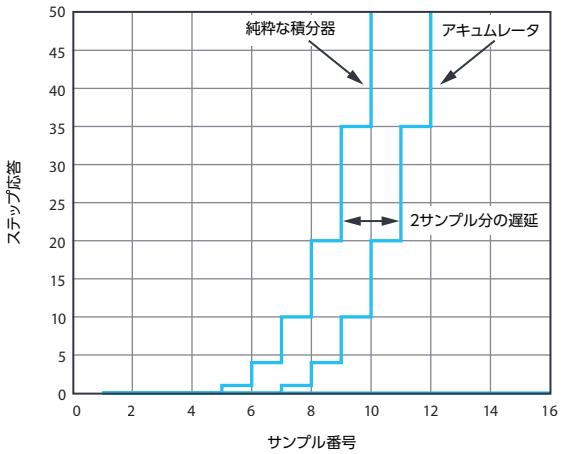

図6. 純粋な積分器とアキュムレータを使用した場合のステップ応答の違い

ほとんどの場合、公開されているsincフィルタの実装例では、DFFをベースとするアキュムレータを使用して積分器を実装することが推奨されています。ゲート数を少なく抑えられることが最大の理由です。しかし、この軽量な実装には欠点もあります。フィルタの群遅延と比べれば、変調器のクロック2つ分の遅延が加わることは大した問題ではないと思われるかもしれません。しかし、この遅延は、高い周波数成分を減衰するフィルタの能力に影響を及ぼします。その結果、アキュムレータを使用した実装では、純粋な積分器を使用する場合と比べて有効ビット数が少なくなります。また、本稿で紹介したフラッシング型sincフィルタは、理想的な伝達関数に従わなければ正しく機能しません。このような理由から、sincフィルタの積分器は、アキュムレータをベースとして実装するべきではないということになります。

理想的なsinc3フィルタの応答を得るためには、図7のように式（9）の差分方程式を直接的に実装するべきです。このブロック図は、クロックド・ロジック部（DFF）と組み合わせ部（加算部）の2つから成る点に注目してください。この実装には、より多くの論理ゲートが必要になりますが、望ましいフィルタ性能と遅延が得られます。

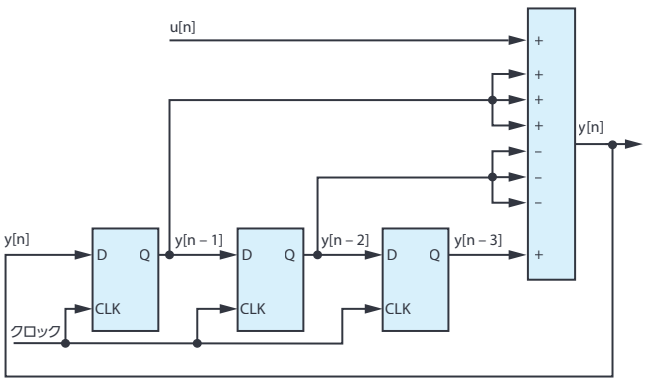

図7. カスケード接続された3つの積分器の実装例

微分器

公開されている多くのsincフィルタの実装例を見ると、積分器と同様に、微分器についても問題があることがわかりました。微分器が不適切に実装されていると、フィルタの性能の低下や予期せぬ遅延が生じる可能性があります。ここでは、まず問題のある微分器の実装について説明します。その上で、FPGA上で最適な性能を得るための実装方法を紹介します。まず、Z領域の純粋な微分器は、以下の式で表されます。

$$\frac{y(z)}{u(z)} = 1 - z^{-1} \quad (11)$$

差分方程式は次式のとおりです[2]。

$$y[n] = u[n] - u[n-1] \quad (12)$$

FPGAにおける微分器の実装としては、図8のようにDFFを使用する方法が最も一般的です。

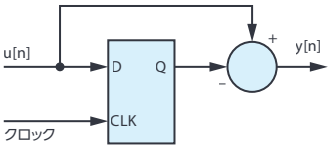

図8. DFFを使って実装された微分器

3つのDFFを使用した微分器は、一般的には、HDLでは図9のようにコーディングされます。ここではVerilogを模したコードを使用していますが、他の言語でも同じようにコーディングできます。

```
always @(posedge clock)
begin
     y1[n] <= u[n] – u[n–1];
     y2[n] <= y1[n] – y1[n–1];
     y3[n] <= y2[n] – y2[n–1];
     u[n–1]  <= u[n];
     y1[n–1] <= y1[n];
     y2[n–1] <= y2[n];
end
```

図9. クロックド・ロジックとしてコーディングされた3つの微分器

クロックド・アサインメントでは、必ず、右辺のすべてのステートメントが評価された後に、左辺への代入が実行されます[3]。すべてがクロックに従い、すべての代入が並列に実行されるということです。しかし、出力項yx[n]は、先に更新される遅延項u[n-1]、yx [n-1]に依存するため、問題が生じます。図9のHDLコードは、図10のような回路として実装されるからです。

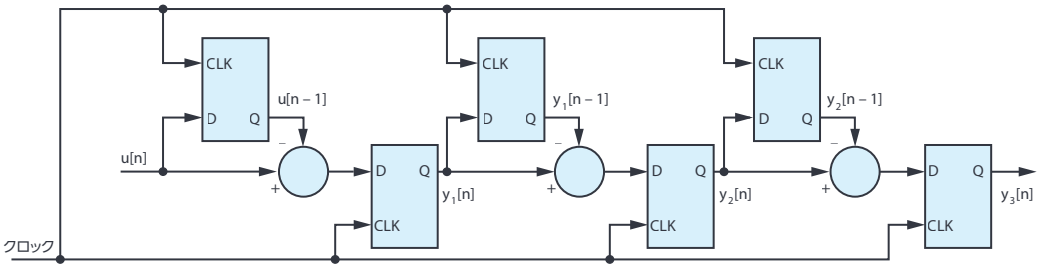

図10. クロックド・アサインメントに従って実装された微分器

クロックド・アサインメントを使用すると、微分器の遅延は、想定していた3クロック分ではなく6クロック分になります。微分器はデシメーション・クロックで動作するので、フィルタの群遅延とセトリング時間は実質的に2倍になります。また、フィルタの減衰特性にも影響が生じ、周波数応答は、理想的な3次sincフィルタのものとは異なる状態になります。図10に示した回路は、sincフィルタの実装例としてよく見られるものです。しかし、このように実装するのではなく、理想的な微分器を模して実装することを強くお勧めします。

図10のHDLコードは、現在の出力を計算する組み合わせ部と、遅延の状態を更新するクロックド・ロジックの2つに分けることができます。そのようにすれば、図11に示すコードのように、クロックに従うブロックの外に組み合わせ部を移すことができます。

```
        u[n-1] <= u[n];
        y1[n-1] <= y1[n];
        y2[n-1] <= y2[n];
    end
    assign y1[n] = u[n] – u[n-1];
```

図11. 3つの微分器をクロックド・ロジックと組み合わせ部に分けてコーディングした例

図11のようなコンビナトリアル・アサインメントの場合、yxの計算によって遅延が追加されることはありません。そのため、トータルの遅延は、6クロック分ではなく理想的な3クロック分に抑えられます。図12に、微分器の理想的なブロック図を示しました。

ここまでに説明した方法で実装した積分器と微分器を組み合わせれば、減衰と遅延の面で理想的な特性を備えるsincフィルタを実現できます。そのようにしてフラッシング型sincフィルタを実装することにより、フィルタの遅延について正確に把握することが可能になります。その結果、ΣΔ ADCを使用したあらゆる測定においてメリットを享受できるようになります。

測定結果

筆者らは、ここまでに説明した内容を考慮して、ΣΔ ADCによって測定を行うシステムを実装しました。その上で、Xilinx®の「Zynq®-7020 SoC」[4]をベースとするサーボ・モータ・コントローラと組み合わせてテストを実施しました。開発したシステムは、永久磁石をベースとし、60Vに対応する3相サーボ・モータ(Kinco Automationの「SMH40S[5]」)と、3相スイッチング電圧源インバータで構成されます。Zynq-7020 SoCは、フィールド指向のモータ・コントロール用アルゴリズムと、リアルタイムに測定データを取得するためのソフトウェアを実行します。

このシステムでは、2個の絶縁型ΣΔ ADC「ADuM7701」の後段に、2個の3次sincフィルタを配置します。それにより、相電流を測定します。sincフィルタは、フラッシングの動作モードを含めて、本稿で説明した内容に即して実装しました。比較を行うために、従来の連続動作型sincフィルタを使用した場合の測定結果と、フラッシングsincフィルタを使用した場合の測定結果を示します。

制御システムは、クローズドループのフィールド指向制御(FOC：Field-oriented Control)に対応しますが、測定はすべてオープンループ制御で行います。クローズド電流ループは、測定時にノイズの影響を受けやすく、電流ループを介してノイズが結合します。オープンループで測定を行えば、電流コントローラからの影響はすべて除去されるので、結果を直接比較することができます。

測定は、モードの設定とPWMとの同期処理を除き、すべて同一の設定の下で実施しました。デシメーション・レートも125に統一しています。したがって、測定結果に違いが生じるとすれば、sinc3のインパルス応答とPWM周期の同期が正確に実現されているか否かが原因だと見なすことができます。制御アルゴリズムは10kHzで実行し、変調器のクロックは12.5MHzとしました。

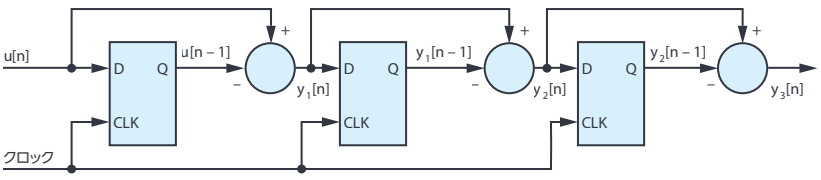

図12. 3つの微分器をクロックド・ロジックと組み合わせロジックによって実装した例

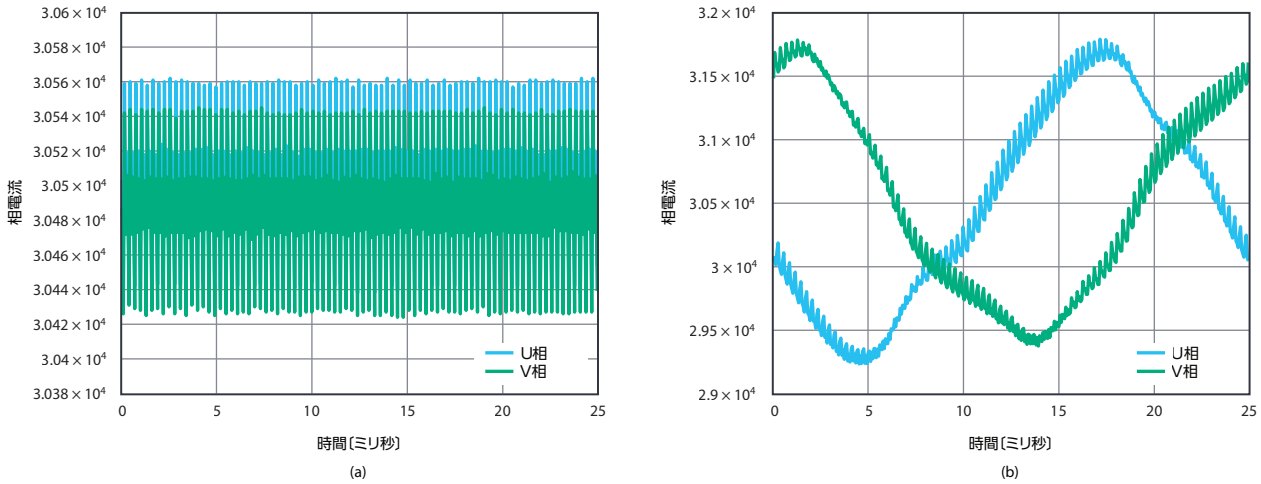

図13. 連続動作型sincフィルタを使用した場合の測定結果。
インパルス応答とPWM周期の同期はとられていません。

連続動作型sincフィルタを使用した場合の測定結果――インパルス応答とPWM周期は非同期

図13（a）は、インパルス応答とPWM周期の同期をとっていない場合の相電流の測定結果です。図13（b）の測定結果は、モータは停止させた状態で、パワー・インバータにおいて全相を50%のデューティ・サイクルでスイッチングさせて2つの相電流を測定した結果です。この動作モードでは、ノイズのレベルが測定結果に現れます。この相電流は、モータが600rpmでオープンループ制御を実行するという条件で測定しました。モータは4つの極のペアを持ち、電気的な周期は25ミリ秒です。どちらのグラフにも大きなノイズが現れており、クローズドループの電流制御回路では、性能に深刻な影響が生じると考えられます。ノイズ・レベルは相電流の大きさとは無関係であり、負荷が軽い場合のノイズは比較的深刻な問題になります。この例の場合、ノイズは、sincフィルタのインパルス応答とPWM周期の同期がとられていないことが原因で発生します。sincフィルタのデシメーション・レート（減衰率）を高めても、結果はほぼ変化しません。

連続動作型sincフィルタを使用した場合の測定結果――インパルス応答とPWM周期は同期

図14は、PWM周期あたりのデシメーション・サイクル数が整数で、インパルス応答が理想的な測定ポイントに対応している場合の測定結果です。図14の結果は、図13の結果と直接比較することができます。そうすると、フィルタのデシメーション・レートは同じなのに、図14ではノイズのレベルが非常に小さく抑えられていることがわかります。これらの測定結果は、ΣΔ ADCを含むシグナル・チェーンの性能を最大限に活かすには、システムの構成と同期がいかに重要であるかということを表しています。

フラッシングsincフィルタを使用した場合の測定結果

図14の測定結果が得られるのであれば、性能としては十分です。しかし、先述したとおり、連続動作型sincフィルタを使用する場合、同期を実現できるようにフィルタを構成するのは容易ではありません。

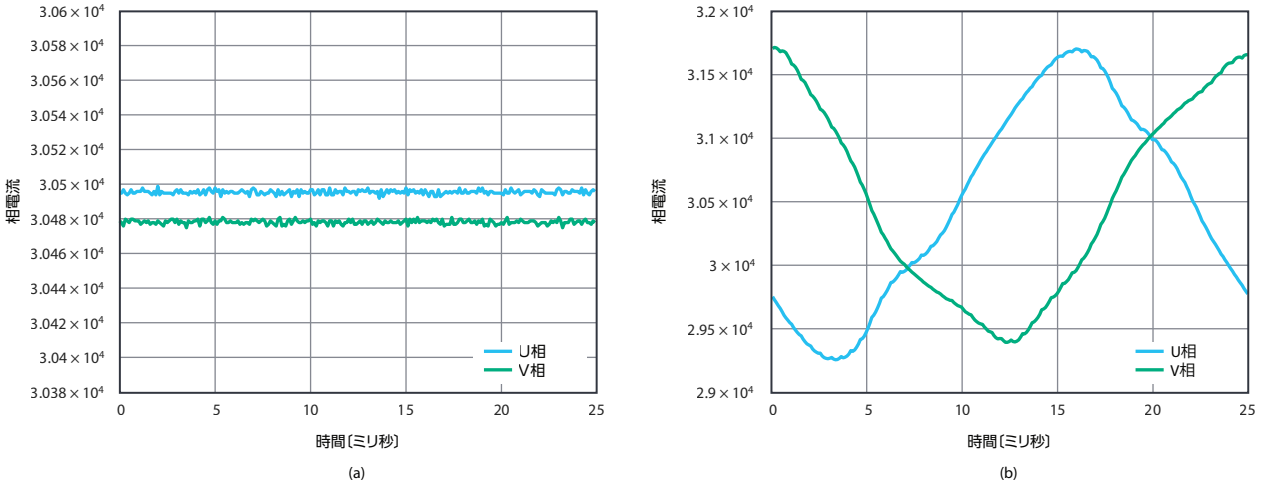

図14. 連続動作型sincフィルタにおいて、インパルス応答と
PWM周期の同期をとった場合の測定結果

連続動作型sincフィルタとPWM周期の同期をとるのは不可能ではありませんが、実用的ではないケースが少なくないということです。この問題を解決するのが、フラッシング型sincフィルタです。

図15は、フラッシング型sincフィルタを使用した場合の測定結果です。このフィルタでは、理想的な測定ポイントを中心とする3デシメーション・サイクルだけフィルタ処理を実行するように構成してあります。想定どおり、図14の連続動作型フィルタを使った場合と同等の性能が得られています。

直接的に比較が行えるように、フラッシング型sincフィルタと連続動作型sincフィルタは、同じ条件に基づいて構成しています。両者の違いは、連続動作型sincフィルタを使用する場合、この構成でなければ図13の測定結果のように性能が低下してしまう点にあります。一方、フラッシング型sincフィルタであれば、任意のシステム設計において最適な性能を得ることが可能です。

インパルス応答とPWM周期の同期がとられていない場合、連続動作型sincフィルタのノイズの大きさは、図13（a）に示すように、16ビットの信号の約120LSBに相当するレベルにまで達します。つまり、ノイズによって下位7ビット分の精度が失われることになります。図15（a）に示したように、フラッシング型sincフィルタのノイズは、16ビットの信号の約5LSBに相当するレベルに抑制されます。つまり、ノイズによって失われる精度は、3ビット分を下回るレベルに抑えられます。

まとめ

ΣΔ ADCによって相電流を測定する手法は、モータ駆動アプリケーションで広く使われています。但し、最適な性能を得るには、システム全体を適切に構成する必要があります。本稿では、性能の低下につながる要因を明らかにすると共に、システムの適切な構成方法について説明しました。

最適な電流フィードバック性能が得られるようにシステムを構成するのは容易ではありません。場合によっては、それが不可能なケースもあります。この問題を解決するものとして、本稿ではフラッシング型sincフィルタという新たな手法を紹介しました。このフィルタはオン／オフ・モードで動作し、任意の構成のシステムにおいて最適な性能が得られることを保証します。

FPGA上にsincフィルタを実装するには、HDLコードの開発が必要です。本稿では、フィルタの遅延を低減し、大きな減衰量が得られるようにするためのコーディング手法も紹介しました。

最後に、フラッシング型sincフィルタがもたらす重要な効果を明らかにする測定結果も示しました。

参考資料

[1] Dara O'Sullivan、Jens Sorensen、Aengus Murray「ADSP-CM402F/ADSP-CM403F/ADSP-CM407F/ADSP-CM408F SINC フィルタとAD7401A を使用した絶縁型モーター・コントロール帰還」Analog Devices、2015年4月

[2] Alan Oppernheim、Ronald Schafer「Discrete-Time Signal Processing. Third edition（離散時間信号処理 第3版）」Prentice Hall Inc.、2010年

[3] Rajeev Madhavan「Quick Reference for Verilog HDL（Verilog HDLクイック・リファレンス）」Automata Publishing Company、1995年

[4]「Zynq-7000 SoC Data Sheet: Overview（Zynq-7000 SoC データシート：概要）」Xilinx, Inc.、2018年7月

[5]「KNC-SRV-SMH40S Servo（KNC-SRV-SMH40S サーボ）」Anaheim Automation, Inc.、2019年4月

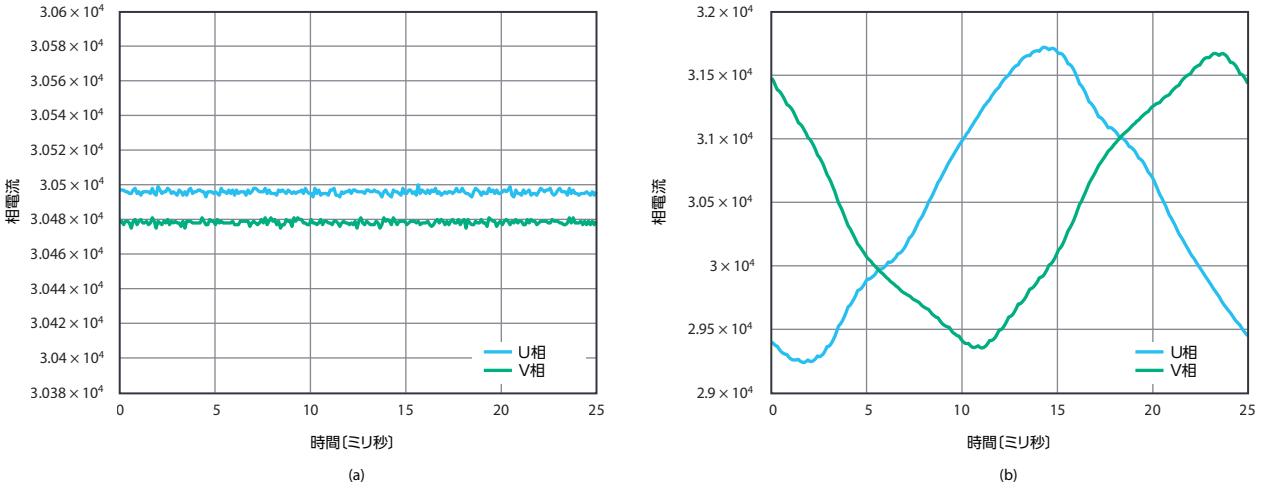

図15. フラッシング型sincフィルタを使用した場合の測定結果。
インパルス応答とPWM周期の同期がとられています。

著者について

Jens Sorensen（jens.sorensen@analog.com）は、アナログ・デバイセズのシステム・アプリケーション・エンジニアです。産業用アプリケーション向けのモータ・コントロール・ソリューションを担当しています。制御用のアルゴリズム、パワー・エレクトロニクス、制御用のプロセッサに関心を持っています。デンマークのオールボー大学で工学分野の修士号を取得しています。

著者について

Shane O'Meara（shane.omeara@analog.com）は、アナログ・デバイセズのコネクテッド・モーション／ロボティクス・チームに所属するシステム・アプリケーション・エンジニアです。産業用モーション制御の分野で制御／監視に使用する高精度の変換技術とシグナル・チェーンに関する技術が専門です。リムリック大学で工学分野の学士号を取得。2011年にアナログ・デバイセズに入社しました。

著者について

Dara O'Sullivan（dara.osullivan@analog.com）は、アナログ・デバイセズのシステム・アプリケーション・マネージャです。オートメーション／エネルギー事業部門のコネクテッド・モーション／ロボティクス・チームに所属しています。産業用モーション制御の分野における電力変換、制御、監視が専門です。アイルランドのユニバーシティ・カレッジ・コークで工学分野の学士号、修士号、博士号を取得しています。2001年から、産業分野や再生可能エネルギーの分野で研究、コンサルティングなどの業務に従事しています。

アナログ・デバイセズ株式会社

AD5311-0-11/19

お住いの地域の本社、販売代理店などの情報は、analog.com/jp/contact をご覧ください。

オンラインサポートコミュニティEngineerZoneでは、アナログ・デバイセズのエキスパートへの質問、FAQの閲覧ができます。

ANALOG DEVICES
想像を超える可能性を
AHEAD OF WHAT'S POSSIBLE™

VISIT ANALOG.COM/JP