

リアルタイム・ネットワークを介した多軸モーション制御の同期確立

著者: Jens Sorensen, Dara O'Sullivan, Christian Aaen

概要

EtherCATなどのリアルタイム対応かつデタミニスティックなイーサネット・プロトコルを利用すれば、多軸モーション制御システムの同期動作が可能になります¹⁾。この同期には2つの側面があります。1つは、多様な制御ノードの間でコマンドとリファレンスを伝送する際、共通クロックとの同期をとらなければならないということです。もう1つは、制御アルゴリズムとフィードバック機能は、同一のクロックと同期をとって実行しなければならないということです。前者の同期は、ネットワーク・コントローラの本質的な機能であり、十分に理解されています。一方、後者の同期については、これまで放置されてきたというのが実態です。結果として、現在では、その点がモーション制御の観点から見た性能上のボトルネックとなっています。

本稿では、ネットワーク・コントローラからモータの端子、センサーまで、モータの駆動に関連するあらゆる要素の同期をとるための新たな手法を紹介します。それを活用することにより、格段に高いレベルで同期を確立し、制御の性能を大幅に高めることが可能になります。

問題の明確化、最先端技術の状況

ここでは、まず各種の最先端ソリューションにおける制約について明らかにします。それに向けて、図1に示すような、ネットワークに接続された2軸のモーション制御システムを例にとります。

モーション制御におけるマスタは、2つのサーボ・コントローラに対し、リアルタイム・ネットワークを介してコマンドとリファレンスを送信します。各サーボ・コントローラは、ネットワーク上でスレーブ・ノードを構成しているということになります。サーボ・コントローラの構成要素は、ネットワーク・コントローラ、モータ・コントローラ、パワー・インバータ、モータ/エンコーダです。

リアルタイム・ネットワーク・プロトコルの種類により、マスタとスレーブ・ノードの同期方法は異なります。よく使われるのは、各ノードにローカルの同期クロックを用意する方法です。そのようにして共通の時間を認識できるようにすることで、サーボの全軸に対するリファレンスとコマンドの同期を厳密に確立します。つまり、リアルタイム・ネットワークに接続されたすべてのネットワーク・コントローラの同期がとられるということです。

通常、ネットワーク・コントローラとモータ・コントローラの間には、2本の割り込み（IRQ）信号線があります。1本は、入力信号を収集してネットワーク上に送信するタイミングをモータ・コントローラに通知するためのものです。もう1本は、ネットワークからデータを読み込むタイミングをモータ・コントローラに通知する役割を果たします。この方法に従えば、モーション・コントローラとモータ・コントローラの間で行われるデータ交換の同期がとられ、タイミングについて非常に高い精度を得ることができます。

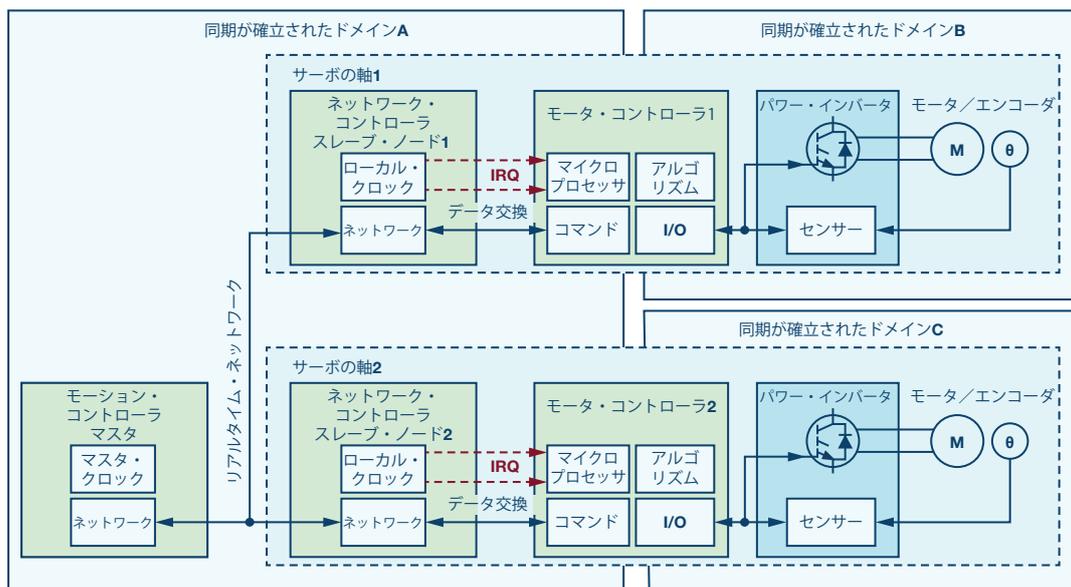


図1. ネットワークに接続された標準的な2軸のモーション制御システム

しかし、それだけでは、両モータ・コントローラ間におけるデータの同期を実現することはできません。両モータ・コントローラも同期をとってデータにตอบสนองすることが必要です。それができなければ、モータ・コントローラはネットワークのタイミング精度を活かすことができません。リファレンスやコマンドに対する応答については、モータ・コントローラのI/Oが問題をもたらすことになります。

PWM（パルス幅変調）タイマーやA/Dコンバータ（ADC）など、モータ・コントローラの各I/Oには本質的な遅延が存在すると共に、時間の量子化に伴う誤差が発生します。例として、パワー・インバータ用のゲート駆動信号を生成するPWMタイマーを考えます（図2）。このタイマーは、リファレンス M_x をアップダウン・カウンタの出力と比較することにより、ゲート駆動信号を生成します。制御アルゴリズムによって M_x が変更された場合、新しいデューティ・サイクルは、PWMの次の周期まで有効になりません。これには、ゼロ次ホールドと同様の効果があります。つまり、デューティ・サイクルはPWMの1周期 T につき1回しか更新されません（ダブルアップデート・モードを使用する場合は2回）。

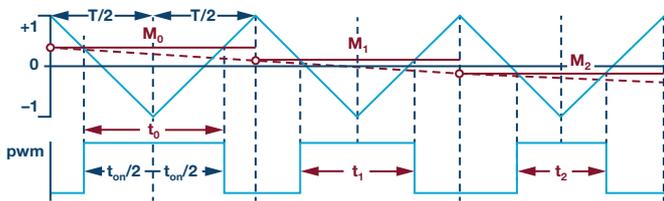


図2. PWMタイマーのデューティ・サイクルの更新

リアルタイム・ネットワーク上でどれだけ厳密にデータ交換の同期がとられていたとしても、PWMタイマーにおける時間の量子化により、軸の同期には決定的なずれが生じます。新たにリファレンスを受信しても、新たなデューティ・サイクルが有効になるまでは応答できないからです。それにより、最大でPWMの1周期分（一般的には50マイクロ秒～100マイクロ秒）の不確実性が生じます。実際には、ネットワークの同期周期とPWM周期の間には、未定義でまちまちな位相関係が存在します。リアルタイム・ネットワークの不確実性が1マイクロ秒未満であることと比較すると、ネットワークを介したモーション制御の同期において、モータ・コントローラのI/Oが重要な要素であることは明らかです。同期の精度を決定するのは、リアルタイム・ネットワークではなく、システムのI/Oだということです。

改めて、図1を見ると、このシステムには、A、B、Cという互いに独立した3つのドメインがあります。各ドメイン内では同期が確立されています。しかし、各ドメインの間は事実上、非同期です。その不確実性は変動し、最大でPWMの1周期分に達します。

同期の不確実性とアプリケーションへの影響

タイミングの不確実性の影響は、ロボットによる制御や機械による加工といったアプリケーションで使われる高性能の多軸サーボ・システムにおいて顕著に現れます。モータの制御軸の間で、時間オフセットがI/Oのレベルにおいてまちまちだと、ロボットや加工ツールの最終的な3次元位置の精度に直接的かつ測定可能なレベルの影響が及びます。

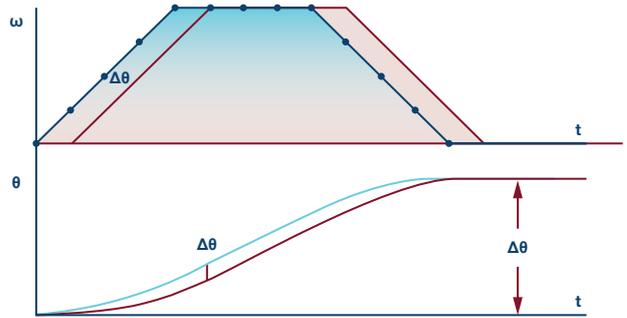


図3. システム内の遅延が位置の精度に及ぼす影響

ここでは、図3に示したシンプルなモーション・プロファイル为例にとります。この例では、モータの速度のリファレンス（赤色）が上昇した後、再び下降します。上昇するレートは電気機械システム的能力の範囲内にあり、実際の速度はリファレンスに従うと予想されます。しかし、システムのどこかに遅延があると、実際の速度（青色）はリファレンスよりも遅れ、位置の誤差 $\Delta\theta$ が生じます。

多軸装置において、目標とする位置 (x, y, z) は、装置の機械的な構造に基づいて、角度軸のプロファイル $(\theta_1, \theta_2, \dots, \theta_n)$ に変換されます。そのプロファイルにより、各軸に向けたコマンドのシーケンスが定義されます。つまり、一定の時間おきの位置/速度の設定が定められるということです。ところが、軸間にタイミングの差が存在すると、装置の精度が低下します。図4に示した2軸の例をご覧ください。一連の (x, y) 座標により実線で示されているのが、装置が目標とするパスです。しかし、システム内に遅延が存在すると、 y 軸に対するコマンドにタイミング誤差が生じます。その結果、実際のパス（破線）は不規則な形になっています。

遅延の大きさが一定であれば、適切な補償を適用することによって、その影響を最小限に抑えられるかもしれません。より深刻なのは、遅延の値が未知であり、なおかつ変動するケースです。その場合に適切な補償を適用するのは不可能でしょう。また、遅延の値が変動することは、制御ループのゲインも変動するという意味を意味します。したがって、最適な性能を得るためにループのチューニングを実施することも困難になります。

システムのどこかに遅延が存在すると、装置全体の精度が低下します。逆に、遅延を最小化または排除すれば、生産性と最終製品の品質の向上を実現することができます。

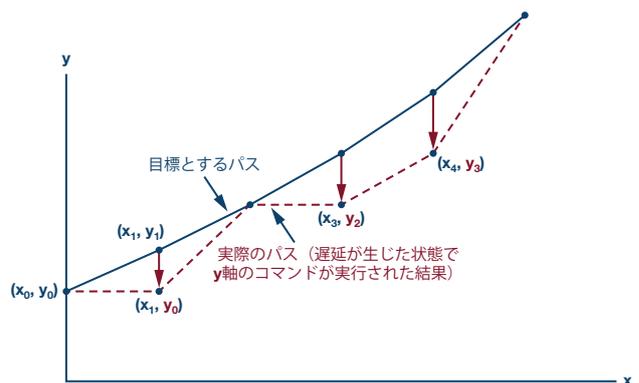


図4. システム内の遅延が位置の精度に及ぼす影響

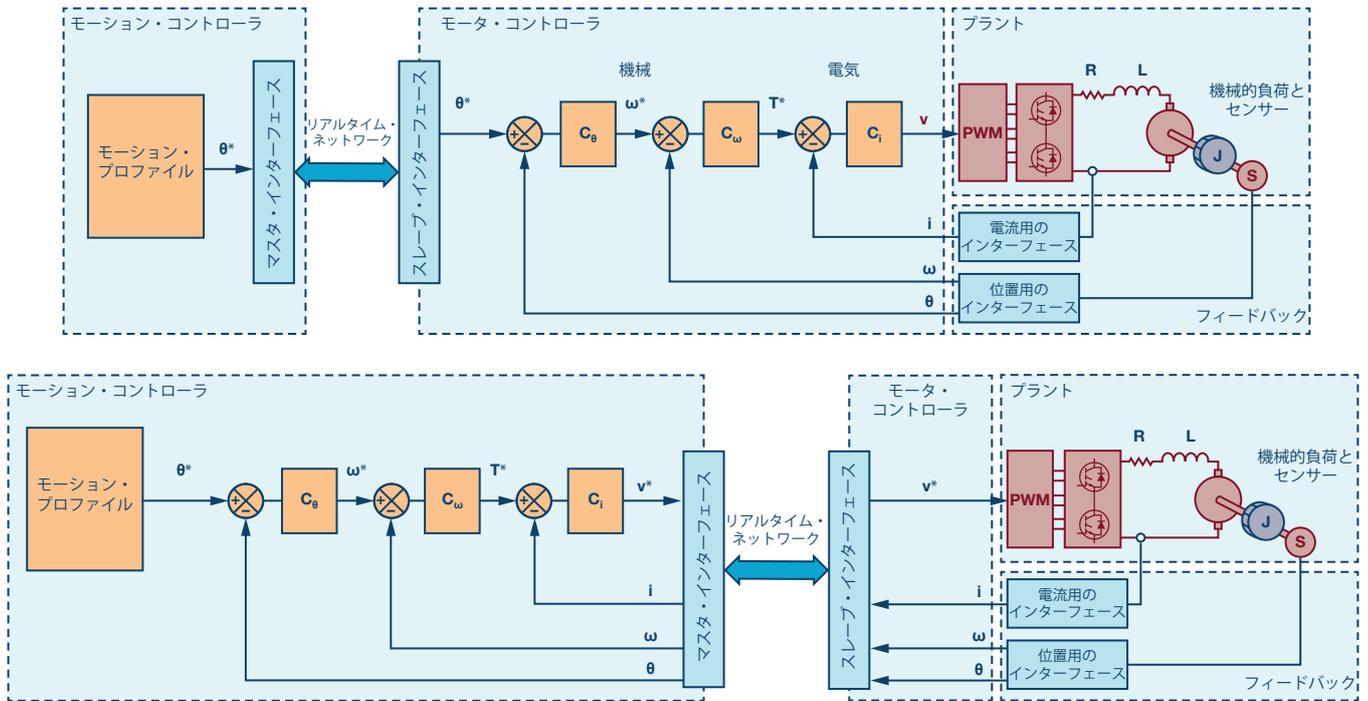


図5. モーション制御のトポロジ。
 (上)は従来のトポロジで、(下)は新たなトポロジです。

同期と新たな制御トポロジ

図5(上)は、従来のモーション制御の方法を示したものです。一般に、モーション・コントローラとしては、PLC(プログラマブルロジックコントローラ)が使われます。このモーション・コントローラは、リアルタイム・ネットワークを介して位置に関するリファレンス(θ^*)をモータ・コントローラに送信します。モーション・コントローラは、カスケード型の3つのフィードバック・ループによって構成されています。3つのループとは、トルク/電流(T/i)を制御する内側のループ、速度(ω)を制御する中央のループ、位置(θ)を制御する外側のループです。トルクを制御するループは帯域幅が最も広く、位置を制御するループは帯域幅が最も狭くなっています。プラントからのフィードバックは、モーション・コントローラでローカルに保持され、制御アルゴリズムの実行ならびにPWM回路の動作との間で厳密な同期がとられます。

このトポロジでは、モーション・コントローラとモータ・コントローラの間で位置のリファレンスが交換されることによって、各軸の同期がとられます。モータ・コントローラのI/O(フィードバック回路とPWM回路)との同期に対する相関は、CNC(コンピュータ数値制御)加工などの非常に高精度なアプリケーションにおいてのみ問題になります。通常、位置用の制御ループは帯域幅がかなり狭いため、I/Oの同期の影響はさほど大きくありません。したがって、ネットワークとI/Oが、同期が確立された異なるドメイン内にあっても、一般的にはリファレンスのレベルでノードの同期をとることより、許容範囲内の性能を得ることができます。

図5(上)に示したのは一般的な制御トポロジです。ただ、それとは異なる形態で制御を分割している例もあります。それは、位置用の制御ループと速度用の制御ループをモーション・コントローラ側の実装し、ネットワークを介して速度/トルクのリファレンスを送信するというものです。その一方で、業界では図5(下)のような分割方法への移行がトレンドとなっています。すなわち、すべての制御ループをモータ・コントローラからネットワークのマスタ側にある高性能のモーション・コントローラに移すというものです。リアルタイム・ネットワーク上で交換されるのは、モータ・コントローラ用の電圧リファレンス(v^*)と、モーション・コントローラ用のプラントからのフィードバック(i, ω, θ)のデータです。この制御トポロジは、マルチコア・プロセッサを採用した高性能のPLCとリアルタイム・ネットワークによって実現されます。これによってもたらされるメリットはいくつかあります。1つは、アーキテクチャのスケールビリティが非常に高いことです。軸の追加/削除も容易であり、モータ・コントローラの処理能力について配慮する必要がありません。また、軌道の計画とモーション制御の両方が1カ所(中央)で行われるため、高い精度が得られます。

但し、この新たな制御トポロジには欠点もあります。それは、制御アルゴリズムをモータ・コントローラから排除したことで、コードの実行とI/Oの厳密な同期が損なわれるというものです。制御ループの帯域幅が広いほど、I/Oの同期が損なわれることは大きな問題になります。特にトルク/電流用の制御ループは同期に対して敏感です。

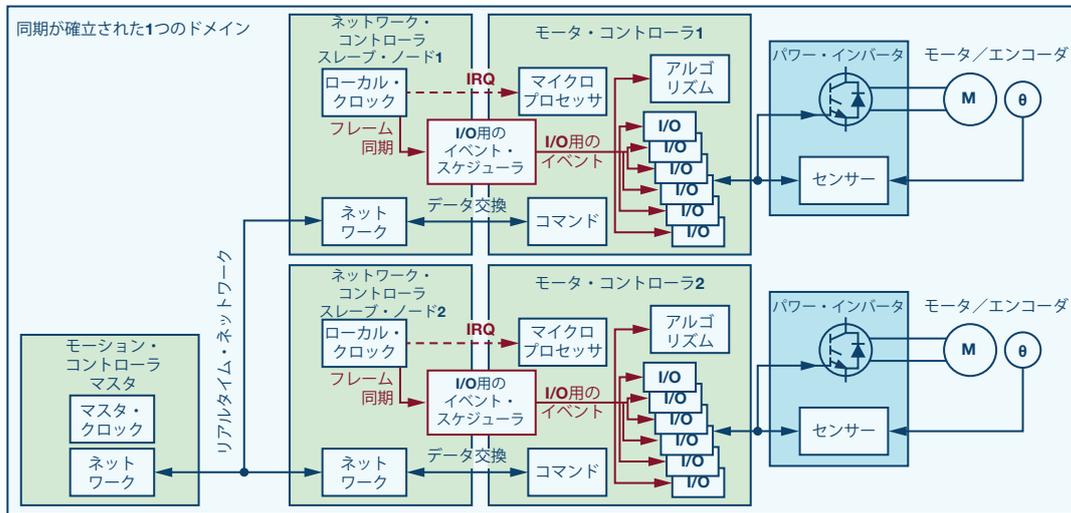


図6. I/O用のスケジューラ役割。このスケジューラにより、同期の確立された1つのドメインを実現します。

提案するソリューション

高速な制御ループをモーション・コントローラに移すと、ネットワークのマスタからモータの端子に至るまでのすべての同期を確立する必要が生じます。

本稿で提案するのは、各軸に対応するI/Oとネットワークの同期をとることで、あらゆるものの同期が確立された1つのドメイン内で処理を実行するという考え方です。それに向けては、ネットワーク・コントローラとモータ・コントローラの間にはI/O用のイベント・スケジューラを配置します(図6)。このイベント・スケジューラの主な機能は、すべてのペリフェラルに対する同期/リセット・パルスを生成することです。それにより、各ペリフェラルとネットワーク・トラフィックの同期を確立することが可能になります。同スケジューラは、ネットワーク・コントローラのローカル・クロックに基づくフレーム同期信号を受け取り、ネットワークと同期をとる必要があるすべてのI/Oに対して適切なハードウェア・トリガを出力します。

各I/Oには、それぞれに独自のタイミング/リセットの要件があります。したがって、I/O用のイベント・スケジューラは、各I/Oの要件に合致するトリガを供給しなければなりません。トリガに対する要件もそれぞれに異なりますが、一般的な原則はすべてのI/Oに共通です。まず、すべてのトリガはフレーム同期信号を基準にする必要があります。また、トリガごとに遅延/オフセットが伴います。その遅延は、I/Oの本質的な遅延(ADCの変換時間やsincフィルタの群遅延など)に関連して生じます。加えて、I/Oには応答時間(ADCからのデータ転送など)が存在します。I/O用のイベント・スケジューラは、各I/Oのタイミングに関する要件を把握し、トリガ/リセット用のパルスをローカル・クロックに合わせて連続的に調整します。図7に、I/O用のイベント・スケジューラにおいて各出力パルスが生成される原理を示しました。

ネットワークに接続されたほとんどのモーション制御システムでは、フレーム・レート(フレーム同期レート)は、モータ・コントローラにおけるPWMの更新レートと同じかそれ以下です。

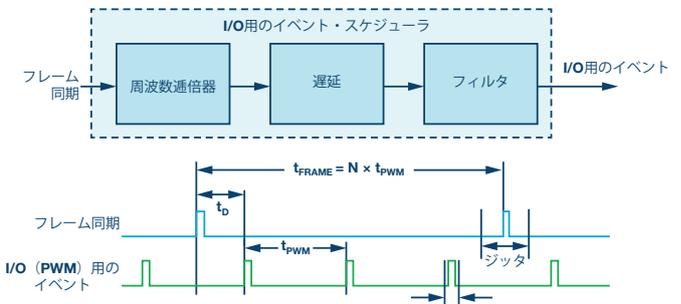


図7. I/O用のスケジューラによるトリガ・パルスの生成

つまり、I/O用のイベント・スケジューラは、フレームの周期ごとに少なくとも1つのトリガ・パルスを供給する必要があります。

例えば、フレーム・レートが10kHzでPWMの更新レートが10kHzである場合、I/O用のイベント・スケジューラは、ネットワークのフレームごとに1つのトリガ・パルスを供給しなければなりません。同様に、フレーム・レートが1kHzでPWMの更新レートが10kHzである場合には、ネットワークのフレームごとに10個のトリガ・パルスを供給する必要があります。この処理に利用されるのが図7の周波数通倍器です。続いて、各I/Oの本質的な遅延を補償するために、遅延 t_b が各同期パルスに適用されます。I/O用のイベント・スケジューラ最後の要素は、インテリジェントなフィルタ機能です。どのようなネットワークにも、トラフィック上にはいくらかのジッタが生じます。図7のフィルタは、トリガ・パルス上のジッタを抑えると共に、フレーム同期信号の周波数が急激に変化しないように制限する役割を果たします。

図7(下)には、PWM用の同期信号に関するタイミング・チャートを示しています。フレーム同期の周波数がPWMの周波数の倍数になっている点、ならびにI/O向けのトリガ信号上でジッタが抑えられている点に注目してください。

実装

図8に示したのは、本稿で提案する同期機構の実装例です。ネットワークに接続されたモーション制御システムに実装し、テストも実施済みです。ネットワークのマスターとしてはBeckhoff AutomationのPLC「CX2020」を採用しました。このPLCは、プログラムを開発／配備するためのPCに接続されています。リアルタイム・ネットワーク（赤色の矢印）のプロトコルとしてはEtherCATを採用しています。

モータ・コントローラの主要な要素はアナログ・デバイセズの「fido5200」と「ADSP-CM408」です。集積度の高いこれらのICを組み合わせれば、ネットワークに接続されたモータの駆動に使用するチップ・セットが実現されます。

fido5200は、2つのイーサネット・ポートを備えるリアルタイム・イーサネット・マルチプロトコル（REM）スイッチです。ホストとなるプロセッサと産業用イーサネットの物理層との間の柔軟なインターフェースを提供します。同ICは、様々な産業用イーサネット・プロトコルに対して高度な同期機構を実装可能にするタイマー制御ユニット（TCU）を備えています。入力信号のキャプチャや、方形波信号の出力といった機能も、専用のタイマー・ピンによって実現できます。タイマーの入出力は、同期の対象となるローカルの時間に合わせて実行されるため、ネットワークのトラフィックとの同期も確立されます。それにより、単一のスレーブ・ノードに含まれるI/Oの同期だけでなく、ネットワークに接続されている全スレーブ・ノードとの同期をとることができます。

REMスイッチにはイーサネット・ポートが2つあるので、2つの物理層（PHY1とPHY2）に接続できます。このトポロジは、リング型とライン型の両方のネットワークに対応可能です。但し、このテストにおいては、本稿で提案する同期機構について理解しやすくするために、1つのスレーブ・ノードのみを使用し、1つのイーサネット・ポートのみをアクティブにしました。

REMスイッチは、パラレルのメモリ・バスを介してホスト・プロセッサと通信します。これにより、スループットの向上と遅延の削減が図れます。

ADSP-CM408は、モータ・コントローラのホスト・プロセッサとして使用しました。同ICは、制御およびアプリケーション機能を実現するために「ARM® Cortex®-M4F」をベースとして実装された特定用途向けのプロセッサです。

インバータの制御用のPWM回路、サンプリングを実施するためのADC、位置エンコーダのインターフェース用タイマーなど、産業分野の制御アプリケーションに必要なペリフェラルを内蔵しています。すべてのペリフェラルとネットワークの同期をとるためには、柔軟性の高いトリガ・ルーティング・ユニット（TRU）が使用されます。TRUは、fido5200のTCUによって生成されたトリガ信号を、ADSP-CM408のタイミング・クリティカルなすべてのペリフェラルにリダイレクトします。対象となるペリフェラルとしては、PWM回路、位相電流の測定に使用するsincフィルタ、ADC、アブソリュート・エンコーダ用のインターフェースが挙げられます。

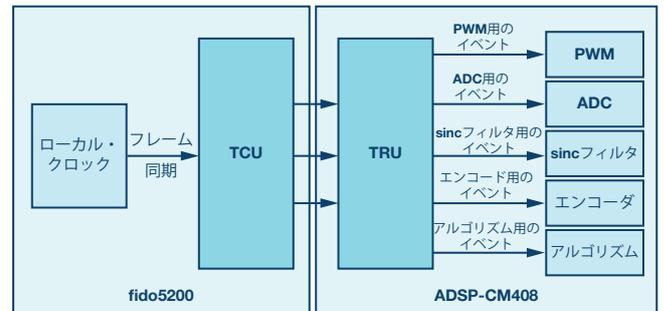


図9. I/O用の同期イベントの生成

I/Oの同期を確立するための原理を図9に示しました。この図において、I/O用のイベント・スケジューラは、REMスイッチが備えるTCUとモータ・コントローラのプロセッサが備えるTRUによって実現されている点に注目してください。つまり、この機能は、2つの集積回路にまたがって実装されています。

モータ・コントローラに対してフィードバックされるのは、電流の位相と3相サーボ・モータにおけるロータの位置の情報です。電流の位相は独立したシグマ・デルタ（ $\Sigma\Delta$ ）方式のADCによって測定され、ロータの位置はEnDatに対応するアブソリュート・エンコーダによって測定されます。 $\Sigma\Delta$ ADCもエンコーダも、外部のFPGAやCPLDを経由することなく、ADSP-CM408に直接接続されています。

PWM回路のスイッチング周波数は10kHzで、制御アルゴリズムはPWMの1周期ごとに1回実行されます。上述したとおり、TCUはPWMの1周期ごとに1回、ADSP-CM408に同期パルスを供給します。

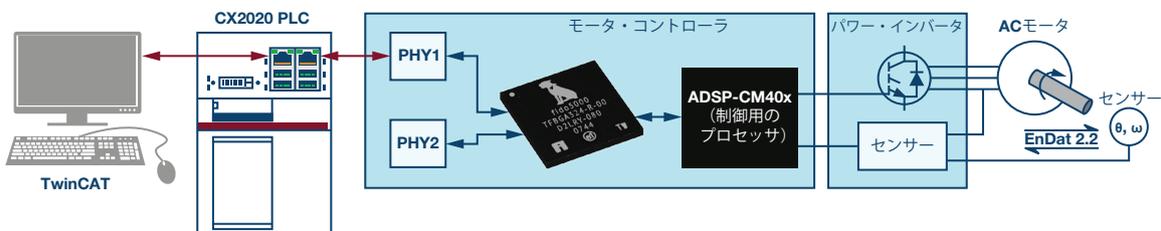


図8. 同期機構の実装

テストの結果

テスト用回路の外観を図10に示しました。システムにおける同期の状況を示すために、PLCは200マイクロ秒のタスク時間でプログラムを実行するように設定しました。このタスク時間によって、EtherCATベースのネットワークのフレーム・レートも決まります。モータ・コントローラは、100 μ s (10kHz) というPWM/制御の更新周期で処理を実行し、そのレートで同期パルスを生成する必要があります。図11にテストの結果を示しました。

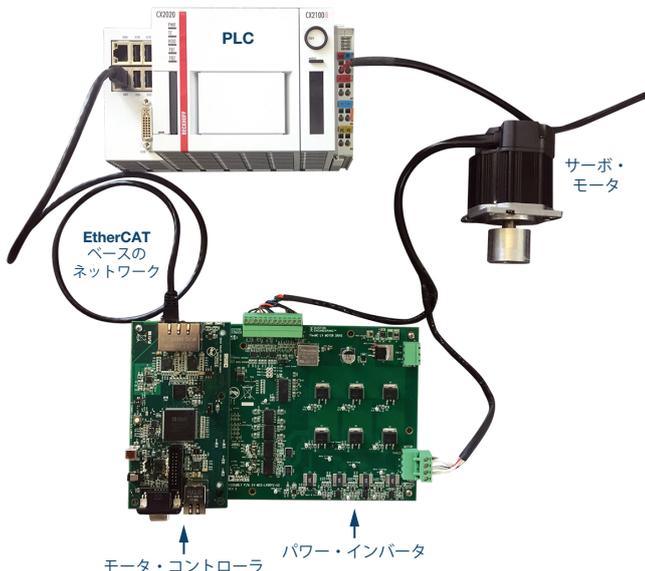


図10. テスト用回路の外観

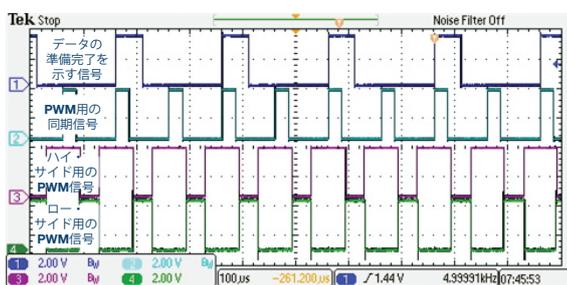


図11. I/O用の同期イベントの生成

図11のData Ready信号は、REMスイッチがモータの制御用のデータをネットワークに送信する準備を終えたことを表します。この信号は、EtherCATのフレーム・レートに相当する200マイクロ秒ごとに発生します。

PWMの同期信号もREMスイッチによって生成され、モータ・コントローラのI/Oとネットワークのトラフィックの同期をとるために使用されます。PWMの周期は100マイクロ秒なので、REMスイッチは、PWM用の同期パルスをEtherCATの1フレームあたり2個生成します。図11(下)のHS PWMとLS PWMは、1つのモータの位相に対応するハイ・サイドのPWM信号とロー・サイドのPWM信号です。PWM信号とネットワークのトラフィックの同期が確立されている点に注目してください。

まとめ

リアルタイム・イーサネットは、モーション制御システムで広く使用されています。なかには、1マイクロ秒未満の精度で時間同期を実現するプロトコルもあります。しかし、その同期は、ネットワークのマスタとスレーブの間でやり取りされるデータのトラフィックのみを対象としています。既存のネットワーク・ソリューションでは、モーション制御用のI/Oの同期については考慮されていません。そのため、達成可能な制御性能に制限が加わっています。

本稿で提案した同期機構は、ネットワークのマスタからモータの端子まで、すべての要素の同期を確立することを可能にします。それにより、制御性能を大幅に高めることができます。この同期機構は、多軸間のシームレスな同期も実現します。軸数が増えた場合にも容易に対応することが可能であり、個々のモータ・コントローラの要件に応じて同期が確立されます。

同期の確立を実現するための主要な要素は、ネットワーク・コントローラとモータ・コントローラの間位置するI/O用のイベント・スケジューラです。このスケジューラは適時プログラムすることが可能であり、ジッタや周波数の変化の影響が最小限になるように調整できます。

本稿では、提案した同期機構を実装し、そのテスト結果も示しました。実験では通信プロトコルとしてEtherCATを使用しましたが、本稿で示した考え方はリアルタイム対応の任意のイーサネット・プロトコルに適用できます。

参考資料

¹ Jie Ma 「Multi-DOF Motion Control System Design and Realization Based on EtherCAT (EtherCATをベースとする多自由度のモータ・コントロール・システムの設計/構築)」 2016 Sixth International Conference on Instrumentation and Measurement, Computer, Communication, and Control, 2016年7月

著者：

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。



Jens Sorensen

Dara O'Sullivan (dara.osullivan@analog.com) は、アナログ・デバイセズのシニア・システム・アプリケーション・エンジニアです。オートメーション/エネルギー/センサー事業部門のMPC（モータ/パワー・コントロール）チームに所属しています。専門は、ACモータ・コントロール・アプリケーションにおける電力変換/制御です。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。



Dara O'Sullivan

Christian Aaen (christian.aaen@analog.com) は、アナログ・デバイセズのデタミニスティック・イーサネット（Deterministic Ethernet）テクノロジー・グループに所属するソフトウェア・システム設計エンジニアです。専門は組み込みソフトウェアの設計であり、電力変換やモータ駆動に関する業務経験を有しています。デンマークのオールボー大学で理学分野の学士号と修士号を取得しています。



Christian Aaen