

反復動作用途: 精密周波数制御 & 振動制御

By Mike McDonald, アプリケーションエンジニア

FL テクノロジーリンク株式会社
TECHNOLOGY LINK, LTD.
〒171-0022 東京都豊島区南池袋 3-18-35
OKビル 2階
Tel: 03-5924-6750 Fax: 03-5924-6751
E-mail: sales@technology-l.com
URL: <http://www.technology-link.jp>

初めに

アクチュエータとステージの一般的な用途として、反復動作、言い換えると連続的或いは設定された回数で2点間を行き来する動作があります。通常、これらの反復動作の用途には、運動のプロファイル、移動の振幅または距離、およびサイクルの頻度または周期を含む多くの特性があります。要件に応じて、Zaber デバイスから最良の結果を得るためのさまざまな最適な方策があります。また、デバイスができることには限界もあります。

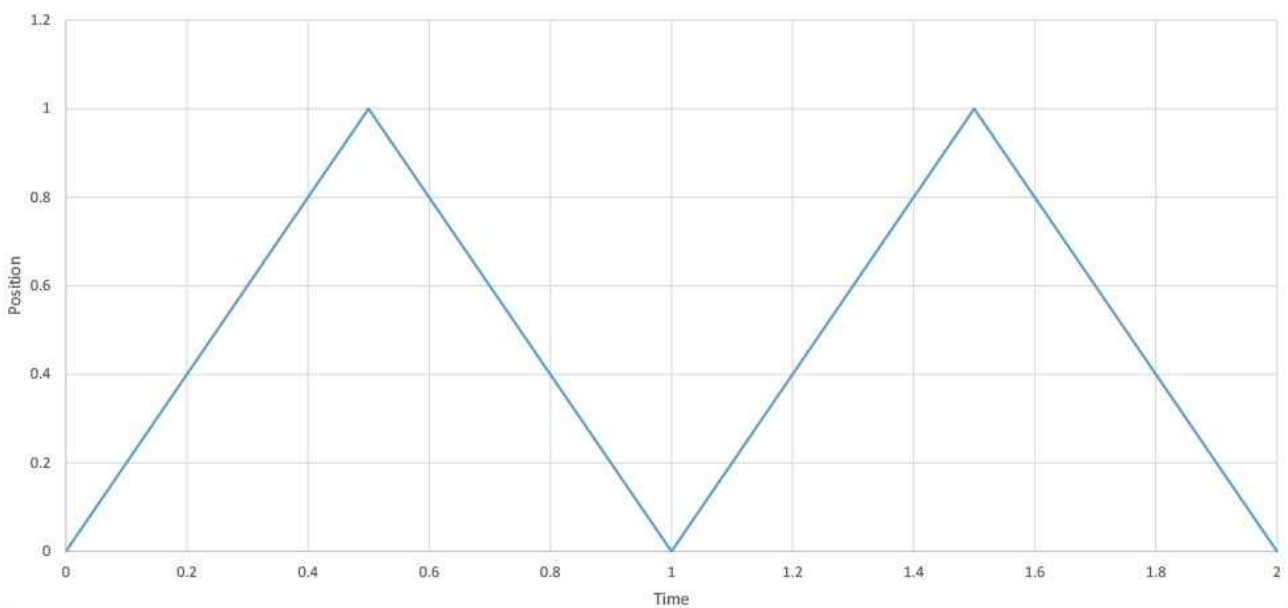
一般的なプロファイルタイプ

三角波形

この動作は、アクチュエータが2ポイント間を一定の速度で動く最も単純なタイプの波形です。方向が変わる時に急激な加速と減速が発生することを想定し、急峻点を作成します。急な加速を想定した場合、振幅と周波数を非常に簡単に制御できるため、使いやすい波形です。

$$\text{速度} = \text{周波数} \times \text{振幅}$$

三角波形



```
#template(Simple)
//triangle waveform example for X-LSM100A for 1 hz, 0.5 mm amplitude
//microstep size 0.047625 μm
//set the speed to 34402, which works out to 1mm/s
Conversation.Request(Command.SetTargetSpeed, 34402);

//set the acceleration to infinite
Conversation.Request(Command.SetAcceleration, 0);

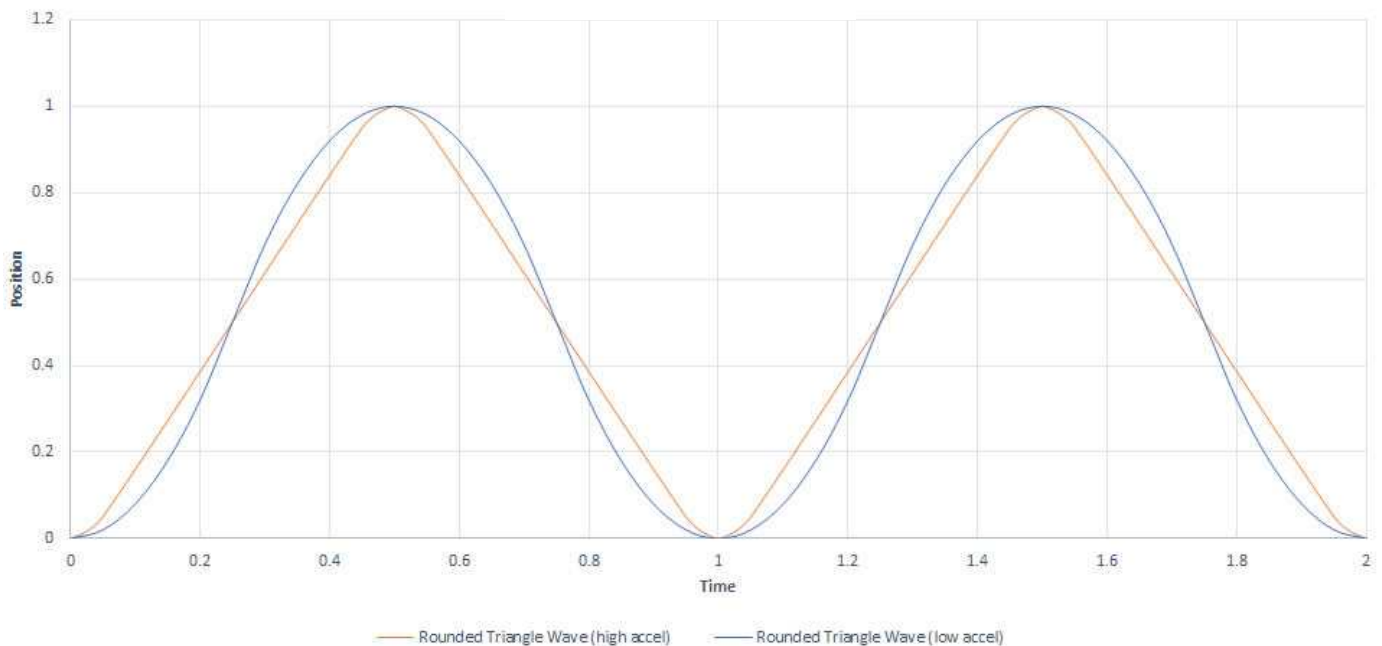
//100 cycles with 0.5 mm amplitude
for(var i = 0; i < 100; i++)
{
  Conversation.Request(Command.MoveRelative, 10499);
  Conversation.Request(Command.MoveRelative, -10499);
}
```

Binary コマンドプロトコルを使用して Zaber Console 用に作成された三角波プログラムの例。Zaber Console のスクリプト作成の詳細については、[http : //www.zaber.com/wiki/Software/Zaber_Console/Scripting](http://www.zaber.com/wiki/Software/Zaber_Console/Scripting) を参照してください。

理論的には最も簡単な波形ですが、無限の加速は実用的ではありません。各デバイスは負荷を加速するための推力に制限があります。つまり、モータが停止する前に設定できる最大加速度があることを意味します。

減速して再び加速すると、三角波のピークを丸める効果があります。加速度が低下するにつれて、ピークはより丸くなります。同じ周期と振幅を維持するには、加速時間を補正するために目標速度を調整する必要があります。各サイクルには、加速、減速、および一定速の 3 つの移動領域があります。ある値の加速に対して、所望の周期および周波数を達成するための目標速度を決定するために、各領域に対して運動学的方程式を使用しなければなりません。

丸み付けされた三角波形



- 与えられた加速に対して目標速度が増加すると、最終的に目標速度に達しない点に到達する。走行の前半は一定の加速になり、後半は一定の減速になります。2領域を活用すると、モーション（動作）がより簡単に判別できます。

$$\text{acceleration} = \text{amplitude} / (\text{period} / 4)^2$$

- 目標速度が以下の値である場合、全体の動きは加速度で構成されます。

$$\text{speed} > \text{acceleration} * (\text{period} / 4)$$

この例の場合、加速値は 14 に丸められています（設定加速度は整数値でなければなりません）。この例では、実際には、 4mm/s^2 の代わりに約 4.07mm/s^2 の加速度と約 0.504Hz の周波数で動作します。デバイスの加速度の分解能は、この方法での精度の制限要因となる可能性があります。

```
#template(Simple)
//rounded waveform example for X-LSM100A for 0.5 hz, 1 mm amplitude
//microstep size 0.047625 μm
//set the acceleration to (1 mm / (0.5s)^2) = 4 mm/s^2
Conversation.Request(Command.SetAcceleration, 14)

//set the speed to be above (4 mm/s^2 * 0.5 s) = 4 mm/s
Conversation.Request(Command.SetTargetSpeed, 137608);

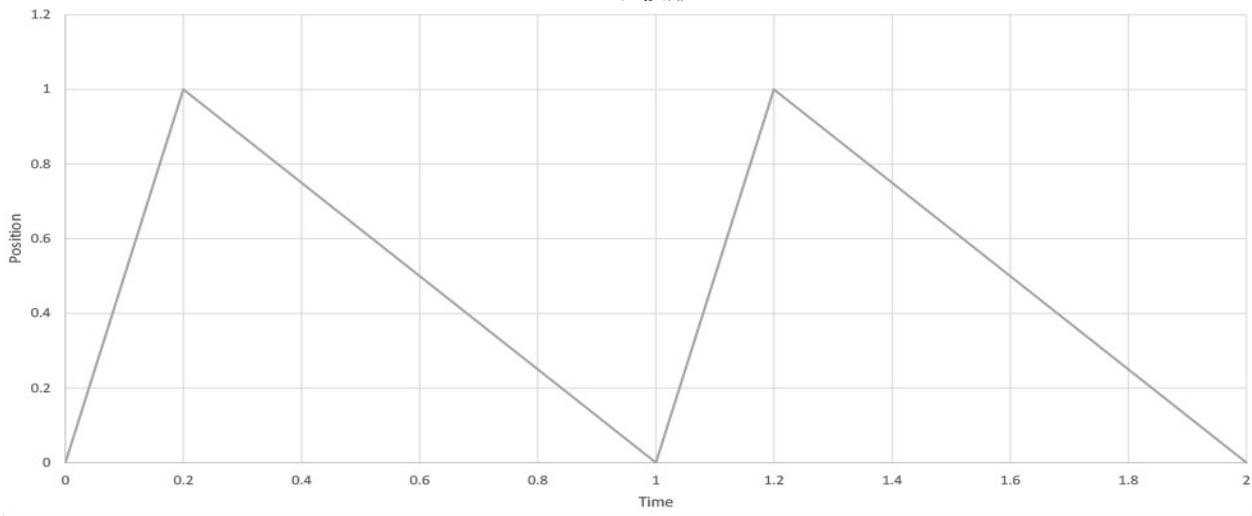
//100 cycles with
for(var i = 0; i < 100; i++)
{
  Conversation.Request(Command.MoveRelative, 20997);
  Conversation.Request(Command.MoveRelative, -20997);
}
```

- Zaber Binary にて Zaber Console 用に作成された丸められた三角波プログラムの例。

ノコギリ波形

鋸歯状のプロファイルは三角波に非常に似ていますが、伸長と収縮の間に異なる速度が使用されます。ある方向には遅いが正確な速度を達成し、次に他の方向には早い速度に達するのが実用的な波形です。例えば、注入のために低速で高精度の流速が必要とされる注射器アプリケーションで使用することができ、注射器は迅速に位置をリセットする必要があります。これは、制御が容易なもう1つの波形で、移動コマンドの前に速度を変更するコマンドを追加するだけです。それは、三角波と同じ加速上の限界が生じます。

ノコギリ波形



```
#template(Simple)
//Sawtooth waveform example for X-LSM100A for 1 hz, 0.5 mm amplitude
//microstep size 0.047625 μm
//set the acceleration to infinite
Conversation.Request(Command.SetAcceleration, 0);

//100 cycles with 0.5mm amplitude, switching between 4 mm/s and 0.57 mm/s
for(var i = 0; i < 100; i++)
{
  Conversation.Request(Command.SetTargetSpeed, 137608);
  Conversation.Request(Command.MoveRelative, 10499);
  Conversation.Request(Command.SetTargetSpeed, 19609);
  Conversation.Request(Command.MoveRelative, -10499);
}
```

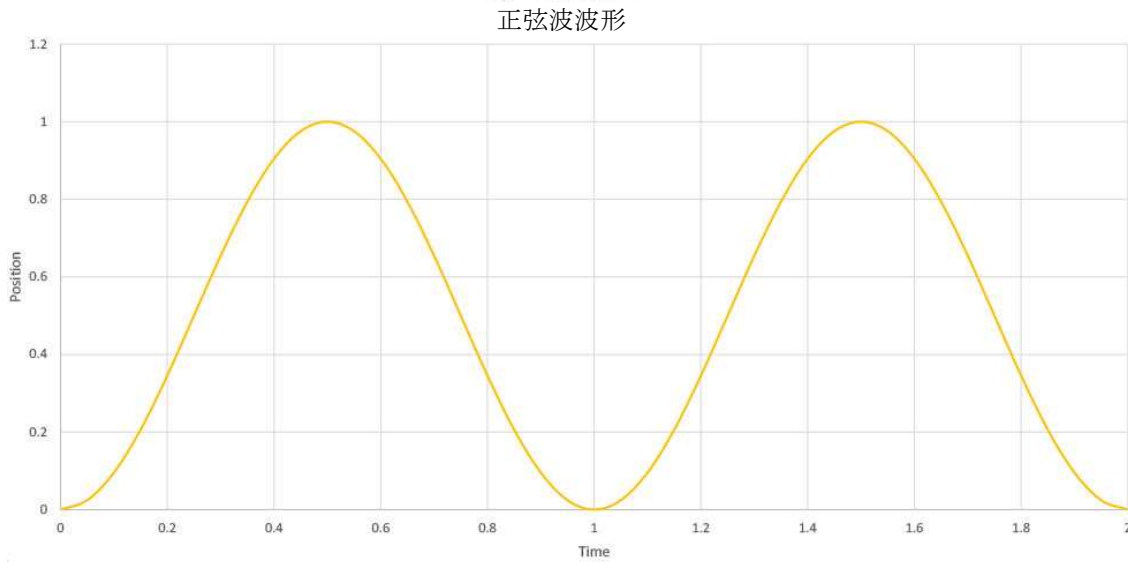
、バイナリで Zaber Console 用に作成されたノコギリ波プログラム例。

正弦波波形

正弦運動は、2点間反復動作のために役立つ共通のパターンです。波は常に変化する速度と加速度を持ち、滑らかで予測可能な動きを提供します。

頻繁に使用されるプロファイルであるため、Zaber デバイスでは **move sin (制限は動作)** コマンドは **Zaber ASCII** プロトコルで実行します。このコマンドユーザーは振幅と周期を定義することができ、デバイスはこれらの入力に基づいてサイクルを開始します。ユーザは、実行するサイクル数を指定することもできます。このコマンドは複雑なタスクを大幅に簡素化すると同時に、タイミングと位置精度の最適なソリューションを提供します。

ASCII プロトコルは、すべての A シリーズおよび X シリーズデバイスで使用できます。T シリーズデバイス (ASCII プロトコル使用不可) を使用して正弦波運動をシミュレートするには、前述の丸めた三角波を使用すると妥当な近似が得られます。



```
#template(Simple)
//Run 200 cycles of sinusoidal motion on an X-LSM100A with 1 hz, 0.5mm amplitude (i.e. 1 mm peak to peak motion) //microstep
size 0.047625 µm
Conversation.Request("move sin 10499 1000 200");
```

・ Zaber Console 用に ASCII で書かれた正弦波動作を開始するプログラムの例。

正確な増幅 VS. 正確なタイミング

留意すべき点は、用途上タイミング精度または位置精度のいずれがより重要かどうかである。そのうちのひとつ正弦波運動指令である **sin 動作** を使用する理由は、両方の精度が保証されるという点で役立ちます。他の解決策は、どちらか一方を達成することに限られます。

この制限の原因は、所望の位置に達したときに次の場所へ移動するように指示するために、装置とソフトウェアとの間の通信の必要性です。デバイスの構成、接続、コマンド、およびソフトウェアに応じて、通信には **5~50** ミリ秒かかることがあります。上記のサンプルスクリプトは、位置精度を達成することに的を絞りました。プログラムは、移動が完了するまで待機してから、次の移動を装置に通信します。これにより、最終目標位置に到達することが保証されますが、タイミングにいくらかの遅延と不確実性が生じます。目標の速度と加速度の設定を調整することで、これを部分的に補うことができます。

精密な動きを使用し、また周波数の速度と加速度に頼る代わりに、正確なタイミングを使用し、且つ振幅の速度と加速度に頼ることができます。ほとんどの動作コマンドのタイミングを制御する最も簡単な方法は、コンピュータのシステム時間を使用することです。この方法では、ポイント・ツー・ポイントの移動コマンドを送信してそれらが完了するのを待つ代わりに、間に遅延を含んだ交互方向に定速度コマンドを送信します。

```
#template(Simple)
//triangle waveform example for X-LSM100A for 1 hz, 0.5 mm amplitude
//microstep size 0.047625 µm
//set the acceleration to infinite
Conversation.Request(Command.SetAcceleration, 0);

//100 cycles with 0.5 mm amplitude, 0.5 s wait between movements (speed 1 mm/s)
for(var i = 0; i < 100; i++)
{
    Conversation.Request(Command.MoveAtConstantSpeed, 34402);
    Sleep(500);
    Conversation.Request(Command.MoveAtConstantSpeed, -34402);
    Sleep(500);
}
```

Binary で Zaber Console の正確なタイミングを使用する三角波の例。

同じ方法を適用して、いずれかの他の波形を生成することもできます。この方法の欠点は、正確な位置を満たさなければならないという条件がプログラムに存在しないため、位置決めの確実性が失われることである。また、コマンドからコマンドの送信までの時間も如何に早く送信するかで少し変わるので、時間的にはミリ秒単位の精度は得られません。

move sin コマンドに加えて、**トリガーとストリームされたコマンド**は、ASCIIプロトコルで利用可能な別のツールで、通信による潜在的なタイミングエラーに対処することができます。トリガーツールは、自動的にアクションを発生させる条件をデバイスに設定するために使用されます。この一例として、ある位置に達してから、その装置は別の位置に移動しなければならないことです。トリガーの詳細については、技術資料「**トリガーを使用したオートメーション**」を参照してください。ストリームコマンドは、送信時にキューに追加され、デバイスはキューに入れられたコマンドを順番に実行します。これは、**2つの動きの間に通信時間がないことを意味します**。ユーザは、デバイスに2つのポイント間を移動するように指示するストリーミングされたモーションコマンドを継続的に送信することができ、ユーザがキューをいっぱいに保つ限り、デバイスは行ったり戻ったりを繰り返します。

ストリームは複数の座標を一連化することもできるので、2つの軸を一緒に循環させるのに便利なツールです。ストリームの詳細については、[ストリームと補間の多軸モーション技術記事](#)を参照してください。

I 高周波数の制限

サイクリングアプリケーションには、しばしば満たすべき周波数範囲があります。最適なデバイスが確実に選択されるように、ユーザはデバイスが達成できる最大周波数を知る必要があります。これを決定する際に考慮すべきいくつかの要素があります。

第1の考慮事項は、前述の通信時間です。ほとんどの Zaber デバイスでは、デバイスにコマンドを送信するのに約 10 ミリ秒かかり、毎秒 100 コマンドが送信されます。各サイクルには、少なくとも 2つのコマンド（選択された波形に依存）が必要です。つまり、50 Hz は、デバイスを制御するコンピュータで達成できる上限値です。上記の移動ツール、ストリームツール、またはトリガーツールを使用すると、通信時間は重要な要素にはなりません。

小振幅用途では、最大加速度が次の制限要因になります。加速が 1 サイクルで利用可能な距離でその速度に達することができない場合、デバイスの最大速度が何であるかは問題ではありません。ステージの最大加速度は、デバイスが加速している速度とデバイスへの負荷の両方に依存するため、指定された仕様ではありません。試行錯誤は、特定のアプリケーションで達成可能な最大加速度を見つける最善の方法です。Zaber のアプリケーションエンジニアは、喜んで、特定の加速度または周波数が必要な場合にデバイスを推奨し、最大加速テストのお手伝いを致します。

一例として、ステージモデル : **X-LSQ150B** は、約 $10,000 \text{ mm/s}^2$ の最大加速度を得るために、1 kg の荷重で（水平に取り付けて）テストしました。ステージがその加速度で 0.50mm の振幅で反復運動をしている場合、それは 70mm/s のピーク速度に達し、各サイクルで加/減速に費やされる時間は約 28ms となります。1 サイクルあたり約 20ms の通信時間で、これは約 20Hz の最大周波数まで動作します

走行距離が長くなるにつれて、最大速度もまた要因になってきます。上記の例に戻って、X-LSQ150B は、 $10,000 \text{ mm/s}^2$ で加速すると約 3.9 mm で最高速度 280 mm/s に達することになります。これは、10mm を超える場合、7.8mm の加減速と 2.2mm の一定速度が存在することを意味します。これは、動作の周期当たり約 128ms、通信時間約 20ms に及びます。この場合、最大周波数は約 6Hz となります。

ステージの負荷が大きければ大きいほど、加速度を低く設定して最大周波数を下げます。ステージが最大加速度の限界に近づくと、ステッピングモーターが停止し、いくつかのサイクルがスキップされる可能性があります。

ほとんどの場合、最良の解決策は、デバイスを選択する際には目標に余裕を持たせ、周波数が達成可能かどうか確認のためにアプリケーションエンジニアにご相談ください。

Mike McDonald は、Zaber Technologies Inc. のアプリケーションエンジニアです。Zaber 社は、光学、フォトニクス、産業オートメーション、生物医学、および他の多くのアプリケーションに使用される精密リニアアクチュエータ、リニアステージ、およびその他のモーションコントロール製品を設計、製造しています。詳細については、www.zaber.com をご覧ください。

If you found the above information interesting, consider subscribing to our newsletter to receive product announcements, user tips, and special promotions (typically worth \$100 off a selected product). Subscribe online at www.zaber.com.