



# PLECS Tutorial

## Modeling an Electric Vehicle Using the PLECS Mechanical Domain

PLECSの機械回路ブロックによる電気自動車のモデリング

Tutorial Version 1.0

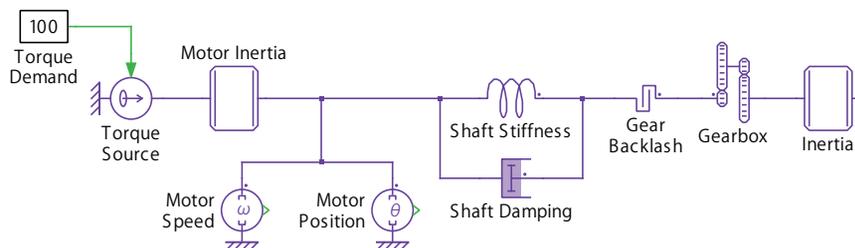
# 1 はじめに

この演習では、PLECSの機械モデリング機能を使用して、電気モータで駆動する電気自動車を表現する複合電気機械システムを作成する方法を学習します。この演習の具体的な学習目標は次のとおりです：

- PLECSの機械回路ブロックの回転ライブラリと並進ライブラリについて理解を深めます。これらの各ライブラリは、回転および並進の機械的ダイナミクスをモデリングするために使用する外力・速度および回路要素のサブカテゴリで構成されています。さらに、各ライブラリには、速度、位置、力/トルクなどの対象となる機械的変数を測定するための測定センサーカテゴリがあります。
- 車両のモータと電気駆動系をモデリングするために必要なコンポーネントを学習します。モータと回転駆動系の機械モデルを開発します。
- 走行抵抗とタイヤの摩擦力を受ける集中車両システムの影響を観察するための並進車両モデルを開発します。

## 2 モータとドライブトレインシャフトのモデリング

図1: モータとドライブトレイン



電気自動車を前進させるために使用するモータは、慣性ブロックを使用して、理想的なトルク源と回転質量として簡単な方法でモデリングできます。モータのダイナミクスは次のように表されます：

$$\frac{d\omega}{dt} = \frac{1}{J_{\text{motor}}} \sum \tau \quad (1)$$

ここで、トルク $\tau$ は車体に対して正に作用します。

駆動系シャフトはモータをギアボックスに接続します。シャフトのダイナミクスは、2つのPLECSコンポーネントを並列に使用してモデリングします。トーションバネブロックはシャフトの剛性をモデリングし、回転ダンパーブロックはシャフトの減衰損失をモデル化します。ギアボックスは、高速回転シャフトを低速回転シャフトに接続して、モータの速度をホイールの速度まで下げるために使用します。ギアボックスの特性は次のとおりです：

$$\omega_{\text{fast}} = g \omega_{\text{slow}} \quad \text{および} \quad \tau_{\text{fast}} = \frac{\tau_{\text{slow}}}{g} \quad (2)$$

ここで $g$ はギア比です。

ギアコンポーネントを接続するときは、高速回転シャフトに接続されたフランジにドットが付いているため、注意が必要です（ギア比が1より大きい場合）。さらに、回転バックラッシュコンポーネントを使用して、ギアの非理想バックラッシュをモデリングすることもできます。モータと駆動系シャフトのモデルを[図1](#)に示します。

駆動系シャフトの角速度とロータ位置は、それぞれ回転速度センサブロックと角度センサブロックを使用して測定できます。各回転(並進)機械センサブロックは、回転(並進)参照フレームに対して、またはセンサの2つのフランジ間で測定を行うことができます。2つのフランジ間の量を測定する場合、マークされていないフランジに対して、ドットでマークされたフランジで測定が行われます。



**あなたのタスク:**

- 1 コンポーネントライブラリ"機械回路ブロック"のサブライブラリ"回転"の"回転要素"にある慣性ブロックを使用してMotor Inertiaをモデリングします。
- 2 トルク(非定常)ブロックを追加し、定数ブロックをその入力に接続して、ドライバが要求(Torque Demand)する100Nmのトルクを生成します。トルク(非定常)をダブルクリックし、第2フランジのドロップダウンメニューから参照座標と連結を選択して、ロータの参照フレームに対するトルクを生成します。トルク(非定常)をMotor Inertia(慣性)に接続します。
- 3 トーションバネブロックと回転ダンパーブロックを使用して、Shaft StiffnessとShaft Dampingをそれぞれモデリングします。[図1](#)のように、これらのコンポーネントをMotor Inertiaに接続します。
- 4 歯車コンポーネントをGearboxとして追加します。回転バックラッシュブロックをGear Backlashとして追加し、ShaftとGearbox間を接続します。
- 5 [表1](#)の値を使用して、モデルのパラメータを入力します。

表1: モータと駆動系のパラメータ

コンポーネント	パラメータ	値
Motor Inertia (慣性)	慣性モーメント	0.014 kgm <sup>2</sup> /rad <sup>2</sup>
	初期速度	0 rad/s
	初期角度	0 rad
Shaft Stiffness (トーションバネ)	バネ定数	45.45Nm/rad
	平衡変位(伸びなし)	0 rad
	初期変位	0 rad
Shaft Damping (回転ダンパー)	減衰定数	0.5Nm/rad
	初期変位	0 rad
Gearbox(歯車)	ギア比	10
Rotational Backlash (回転バックラッシュ)	全バックラッシュ量	0 rad
	初期変位	0 rad

- 6 各センサーをダブルクリックし、第2フランジのドロップダウンメニューから参照座標と連結を選択して、ロータの参照フレームに対する駆動系を測定します。これらを駆動系のシャフトに接続します。
- 7 駆動系の固有振動数 $f_n$ も重要です。車両の慣性モーメントがモータに比べて大きいと仮定すると、シャフトの剛性と減衰によって生じる微分方程式を解くことによって、ドライブラインの固有振動数を計算できます。駆動系の固有振動数の式は次のとおりです:

$$f_n = \frac{1}{2\pi} \sqrt{\frac{c}{J_{\text{motor}}}} \tag{3}$$

ここで $c$ はバネ定数、 $J_{\text{motor}}$ はMotor Inertiaの慣性モーメントです(~9 Hz)。

- 8 車両の慣性を表すために、慣性ブロックをGearboxの二次側に接続し、値を $10^4 \text{kgm}^2/\text{rad}^2$ とします。これによりシステムが減衰し、数サイクルで駆動系の振動が治まります。PLECSスコープを回転速度センサーブロックに接続し、デフォルト設定のままシミュレーションを実行し、モータの速度を監視します。データカーソルを使用して、1回の完全な振動が発生する周期を測定します(約0.116秒)。この周期から固有振動数を計算し、この値を前の手順で計算した周波数と比較します。

 この段階では、モデルは参照モデルmechanical\_domain\_1.plecsと同じになっているはずです。

## 3 車両のモデリング

### 3.1 車両と外力のモデル

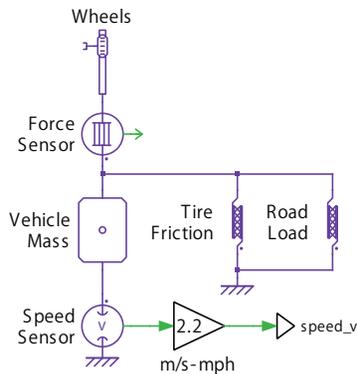
PLECSで電気自動車をモデリングするには、質量と摩擦の2つのコンポーネントが必要です。並進 -> 回路要素 -> 質量ブロックは、端子に接続された車軸からの力を吸収し、質量の速度は次の方程式を積分することによって決定します:

$$\frac{dv}{dt} = \frac{1}{m} \sum F \quad (4)$$

ここで、力 $F$ は車体に対して正に作用します。

滑り摩擦ブロックは、タイヤの摩擦と車両が受ける走行抵抗の両方をモデリングするために使用します。滑り摩擦ブロックの"静摩擦力"パラメータは、摩擦力の最大値です。さらに、滑り摩擦ブロックは、並進参照フレームに対するクーロン摩擦力と粘性摩擦係数をモデリングします。これらの力は車軸に作用するため、[図2](#)に示すように、Vehicle Massと平行に車軸に接続されます。

図2: 単純な電気自動車モデル



#### あなたのタスク:

- 1 Gearboxの二次側に接続している慣性ブロックとPLECSスコープを取り外します。
- 2 コンポーネント ライブラリの"機械回路ブロック"のサブライブラリ並進 -> 回路要素 -> 質量ブロックを使用して車両をモデリングします。測定センサーリストから、並進速度センサーブロックを追加します。センサをダブルクリックし、ドロップダウンメニューから参照座標と連結を選択して、並進参照フレームに対する車両の速度を測定します。

- 並進速度センサーブロックを使用して、並進参照座標を基準として再び車両の並進速度を測定します。ゲイン(利得)ブロックを追加してゲイン値2.2とし、測定された速度をメートル/秒(m/s)からマイル/時(mph)に変換します。
- コンポーネントリストの2つの滑り摩擦ブロックを使用して、タイヤの摩擦と走行抵抗をモデル化します。並進参照座標ブロックを使用して、並進参照フレームに対するタイヤの摩擦力(Tire Friction)と走行抵抗(Road Load)を生成します。
- 表2の値を使用して、車両モデルのパラメータを入力します。使用しているRoad Loadの値は、急な坂道を上る車両を表していることに注意してください。

表2: 電気自動車のパラメータ

コンポーネント	パラメータ	値
Vehicle Mass (質量)	質量	1200kg
	初期速度	0m/s
	初期位置	0m
Tire Friction (滑り摩擦)	静摩擦力	12N
	クーロン摩擦力	12N
	粘性摩擦係数	0 Ns/m
Road Load (滑り摩擦)	静摩擦力	200N
	クーロン摩擦力	200N
	粘性摩擦係数	2000 Ns/m

- 車両システムの回転コンポーネントと並進コンポーネントを統合するには、車輪を介して相互に結合する必要があります。車輪は、コンポーネントライブラリの"機械回路ブロック"のサブライブラリ並進 -> 回路要素 -> ラックアンドピニオンブロックを使用してモデリングします。ラックアンドピニオンブロックを介して、回転駆動系を並進車両モデルに接続します。"歯車半径"を0.28mに設定します。
- 並進 -> 測定センサー -> 外力センサーブロックを追加し、車輪から車両モデルに伝達される力を測定します。他の信号と同じようにPLECSスコープで外力を監視します。

### 3.2 車両システムのシミュレート



#### あなたのタスク:

- 他のすべてのシミュレーションパラメータはデフォルトのまま、シミュレーションを5秒間実行します。単一のPLECSスコープで、Vehicle Speed(車両速度)、Motor Speed(モータ速度)、Motor Position(モータ角度位置)、Force Sensor(伝達された外力)をプロットします。車両の最終的な移動速度、モータの角速度、および伝達力の値はどれくらいですか?(3.7mph、60 rad/s、3571N)。
- モータのトルクを1Nmに変更して、シミュレーションを再実行します。車両の最終的な移動速度、モータの角速度、モータの角度位置、および伝達力の新しい値はどれくらいですか?(0 mph、0 rad/s、0.022 rad、35.7N)。PLECSスコープの現在のトレースを保持ボタンをクリックして完了したトレースを保存し、"Run1"というラベルを付けます。
- Gear Backlashの"全バックラッシュ量"を0.006radに変更して、シミュレーションを再実行します。車両の最終的な移動速度、モータの角速度、モータの角度位置、および伝達力の最新の値はどれくらいですか?(0 mph、0 rad/s、0.025 rad、35.7N)。トレースを"Run2"というラベルを付けて保存します。

- 4 "Run1"と"Run2"の最終的なモータの位置と初期の伝達力を比較します。モータの位置が異なるのはなぜでしょう? 力の伝達に遅れが生じるのはなぜでしょう? "Run1"と"Run2"のモータの最高速度はどれくらいでしょう(それぞれ0.83 rad/s、0.96 rad/s)? なぜこれらは異なるのでしょうか?

 この段階では、モデルは参照モデルmechanical\_domain\_2.plecsと同じになっているはずです。

## 4 高度なモータシステムモデル

### 4.1 実用的な電気駆動システム

モーターシンプルなモデルでは、理想的なトルクソースを使用して、回転するドライブシャフトにドライバの一定トルクの要求(Torque Demand)を生成します。実際には、モータは電気回路によって駆動する電気機械装置です。ドライバが要求するトルクは、三相インバータシステムの変調方式に変換されます。ファイルelectric\_drive.plecs内のElectric Driveサブシステムは、典型的な電気機械システムを表します。これは、3相インバータ、DC電圧源、および3相永久磁石同期機(Permanent Magnet Synchronous Machine: PMSM)で構成されています。ドライバが要求するトルクを生成するために、フィールド指向制御(FOC)アルゴリズムが実装されています。

#### あなたのタスク:

- 1 簡略化されたモータモデルを表すトルク(非定常)ブロックと慣性ブロックを、electric\_drive.plecsファイルのElectric Driveブロックと置き換えます。定数ブロックであるTorque Demandからの値を"T\*"端子に接続します。Torque Demandを100Nmに戻します。
- 2 マルチプレクサブブロックを使用して、駆動系シャフトのMotor Speed信号とMotor Position信号を結合し、Electric Driveブロックの"m"端子に入力します。
- 3 駆動系シャフトをElectric Driveブロックの"s"端子に接続します。
- 4 Torque Demand信号と3相永久磁石同期機(PMSM)によって生成されるトルクを多重化し、2つ目のPLECSスコープを追加して結合した信号をプロットします。
- 5 メニューシミュレーション -> シミュレーションパラメータ... -> 初期化タブで、完全な電気駆動装置を構成するコンポーネントのパラメータ値が含まれている、mechanical\_domain\_init.mファイルを読み込みます。これは、コマンド ウィンドウで次のようにファイル名を呼び出すだけで実行されます(PLECS Blockset): Mechanical\_domain\_init;。  
PLECS Standaloneの場合、テキストエディタでmechanical\_domain\_init.mを読み込み、すべてを選択してコピーして**モデル初期化**コマンドウィンドウに貼り付けます。
- 6 シミュレーションを再実行し、電気回路がPMSMによって生成されるトルクに与える影響を観察します。

## 4.2 センサ誤差

ハードウェアシステムでは、センサによって測定された電流と電圧に誤差が生じます。測定された三相固定子電流は、PMSMのFOC制御アルゴリズムに基づいてインバータのスイッチング信号を実装するために使用されます。相電流の測定にゲイン誤差がある場合、相電流の基本周波数の2倍でトルクリップルが発生します。この効果は、Electric Driveサブシステム内のSensor Errorブロックを介して測定された電流に誤差を導入することで実現できます。この誤差により、測定された電流は次のようになります:

$$i_{a, \text{measured}} = (1+e/100) \cdot i_{a, \text{real}} \quad (5)$$

$$i_{b, \text{measured}} = (1+e/100) \cdot i_{b, \text{real}} \quad (6)$$

$$i_{c, \text{measured}} = i_{c, \text{real}} \quad (7)$$

ここで、 $e$ はSensor Errorブロックに設定された測定誤差の割合です。



### あなたのタスク:

- 1 Sensor Errorブロックをダブルクリックして、5%のGain errorを導入します。
- 2 シミュレーションを実行し、システムに取り入れる低周波トルクリップルを測定します。Electric Driveサブシステムの相電流の基本周波数を測定します。トルクリップルの周波数が相電流の基本周波数の2倍であることを確認するには、Electric Driveサブシステム内のPLECSスコープのカーソルを使用します。



この段階では、モデルは参照モデルmechanical\_domain\_3.plecsと同じになっているはずです。

## 5 まとめ

この演習では、機械回路ブロックライブラリの基本的な並進コンポーネントを使用して、単純な並進車両モデルを作成するための段階的な手法を示しました。さらに、回転ライブラリのコンポーネントを使用して回転駆動系のモデルを開発しました。当初は、ドライバが要求するトルクを車両システムに適用するために、理想的なトルクソースを使用していました。次に、トルクリップルのセンサ誤差を観察するための、より現実的な電気機械システムを導入しました。この演習の拡張として、起動時に駆動系で発生する速度の振動を動的減衰させるコントローラを追加設計することもできます。

PLECSの機械回路ブロックは、エンジニアが機械システムと電気システム間の相互作用による影響を観察するのに役立つ便利なツールです。たとえば、トルク制御が不完全な場合、負荷速度に影響を与え、場合によっては自然共鳴を励起する可能性があります。同様に機械的な非理想性、特に非線形的な非理想性は、電気システムと関連する制御に影響を与えるでしょう。設計者は、故障モード(電気および制御的な性質)の影響を調査することもできます。

完全なシステムレベルモデルを備えた機械回路ブロックでは、パワーエレクトロニクスの完全な使用サイクル(車両の運転サイクルなど)でシミュレーションが可能となり、特に熱モデリングや信頼性解析に興味深い課題となります。最後に、複雑な動作制御システムの場合、完全な機械システムをモデリングすることで、非常に高度な制御アルゴリズムを設計およびテストすることができます。

改訂履歴:

Tutorial Version 1.0 初版

**plexim**

☎ +41 44 533 51 00

+41 44 533 51 01

✉ Plexim GmbH

Technoparkstrasse 1

8005 Zurich

Switzerland

@ info@plexim.com

<http://www.plexim.com>

**Pleximへの連絡方法:**

Phone

Fax

Mail

Email

Web

**KESCO** KEISOKU ENGINEERING SYSTEM

計測エンジニアリングシステム株式会社

<https://kesco.co.jp>

*PLECS Tutorial*

© 2002–2021 by Plexim GmbH

このマニュアルに記載されているソフトウェアPLECSは、ライセンス契約に基づいて提供されています。ソフトウェアは、ライセンス契約の条件の下でのみ使用またはコピーできます。Plexim GmbHの事前の書面による同意なしに、このマニュアルのいかなる部分も、いかなる形式でもコピーまたは複製することはできません。

PLECSはPlexim GmbHの登録商標です。MATLAB、Simulink、およびSimulink Coderは、The MathWorks、Inc.の登録商標です。その他の製品名またはブランド名は、それぞれの所有者の商標または登録商標です。