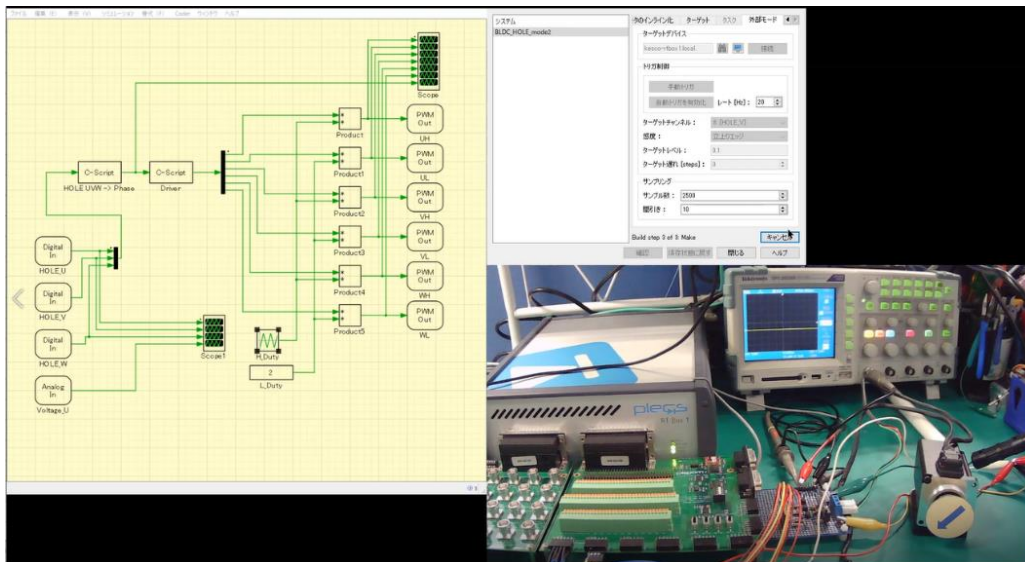


## PLECS-RT BOX を使ったブラシレスモーターの駆動



この資料は、PLECS-RT-BOX を使用してブラシレスモーターを 120 度通電制御にて回転させるデモの動画ではご紹介できなかった部分を補足します。

## 1. デモの概要

PLECS-RT-BOX は PLECS で作成したモデルをハードウェアとして実行できるユニットです

PLECS で作成したモデルをビルドして組み込む事でコントローラーとして使用することができ、また制御対象をモデル化し組み込む事として対象物のエミュレーションをすることもできます。今回のデモは、以下の二段階でモデルから実機検証への移行を紹介していきます。

- ①すべてモデルの状態で行う回転制御を行う (MIL:Model In the Loop Simulation)
- ②制御部のモデルを RT-BOX で動作させ、実機のモーターの回転制御を行う (CIL:Component In the Loop Simulation)

## 2. 構成及び実装する機能の説明

### 2.1 機能の概要

今回実装する 120 度通電制御は、ローターの位置を 6 セクションに分け、その位置にあった回転磁界を発生させるために励磁する相を決めます。簡単なイメージとして Figure 1 を見てください。コイルが 3 つあり、電流を流すときにローターの磁石の方向に対して 90 度の位置で磁束を生成するように制御します。図の場合、反時計回りに回転しようとしています。ローターが回転し②の領域に入った場合、③の領域に磁束が生成されるようにコイルへの電流を変化させます。この一連の動きを繰り返すことで、ローターが回転します。これらの詳細については BLDC の専門書を参照してください。

この制御を行うためにはローターの位置を把握する必要があります、そのためモーターモデルでは回転角情報を利用し、実機ではホールセンサを使用します。

作成するモデルのブロック図は以下のようになりました。

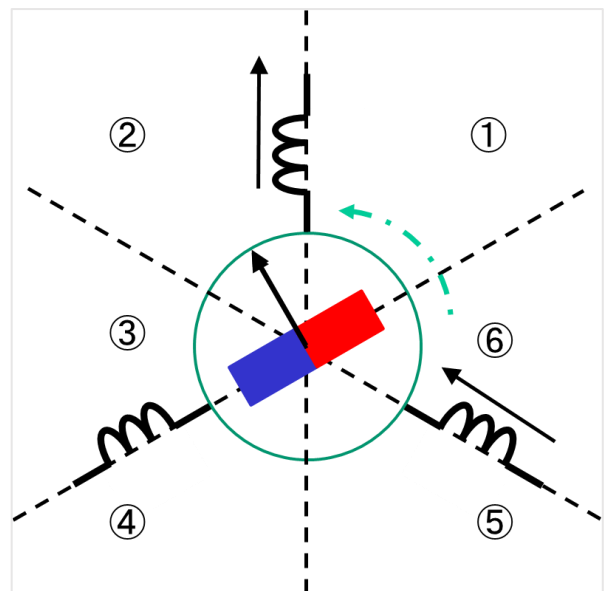


Figure 1

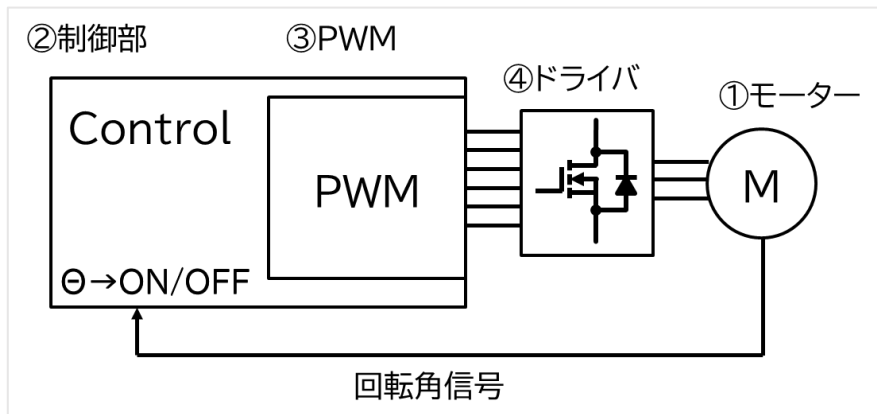


Figure 2

モーターの回転角情報から、ローターの位置を判定しどのドライバを ON するかを決めます。ON するドライバに対して PWM を使ってその印可電流を調整します。

このブロック図からモデルを以下のように作成しました。

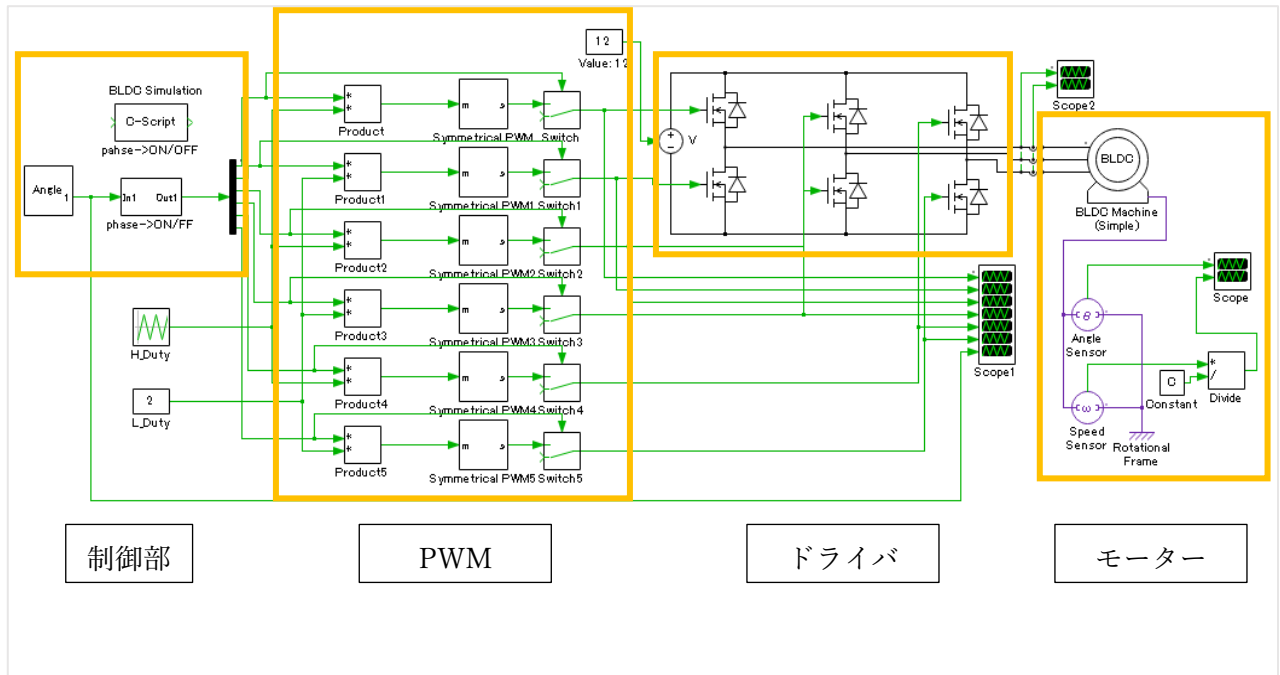


Figure 3

## 2.2 機能の実装 (モデル化)

### (1) モーター

モーターは PLECS の標準ライブラリ内の「BLDC Machine(simple)」を使用します。  
このモデルは内部パラメーターとして以下の設定が可能です。(Figure 4)

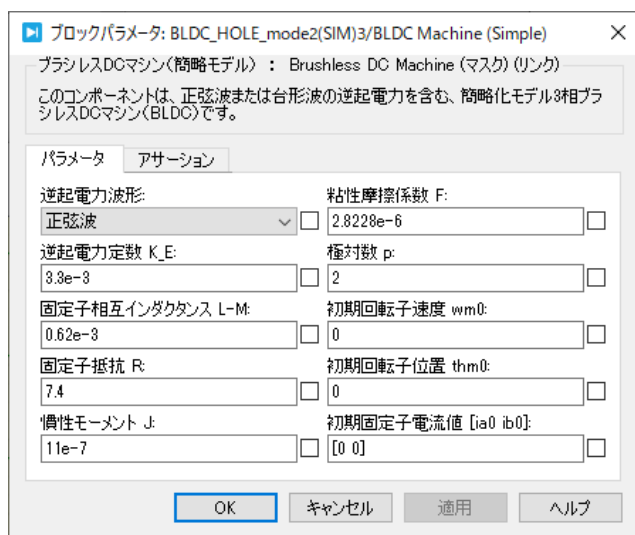


Figure 4

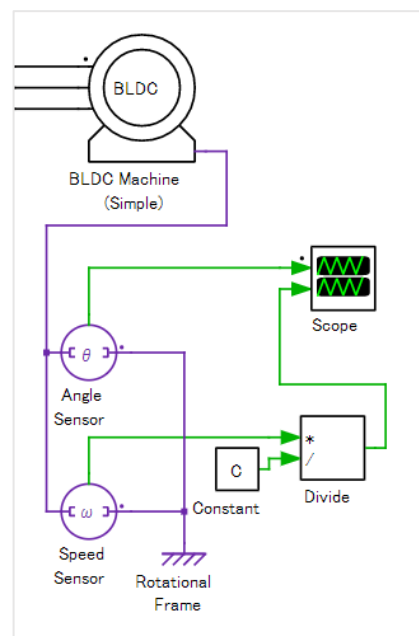


Figure 5

このモデルからはスピードセンサーとアングルセンサー出力があり、この信号を利用して制御を行います。スピードセンサーの出力は単位が[rad/sec]のため Constant コンポーネントで  $2\pi$  を指定し、その後演算コンポーネントで除算することで[r/sec]に換算して波形表示しています。

### (2) 制御部

Angle コンポーネントは PLECS のサブシステムというコンポーネントを使用しています。これは内部にコンポーネントを含めることができます。中は Figure 7 のようになっています。

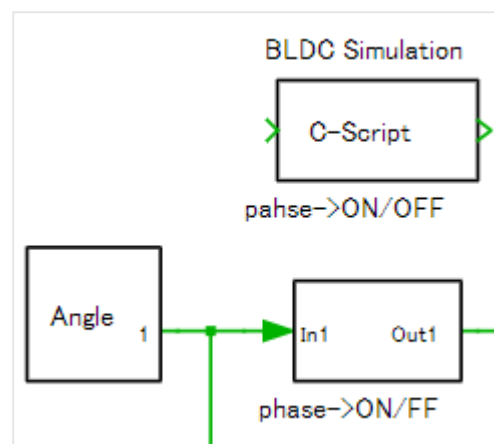


Figure 6

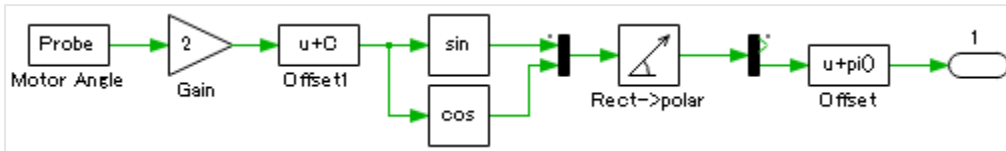


Figure 7

Probe コンポーネントはモデル内の指定信号を抽出することができます。今回はモーターモデルからアングルセンサー出力を取り込みしています。(モーターモデルからの結線で構成も可能です)

次の段の Gain コンポーネントでは、電気角を一致させるため 2 倍しています。今回のモーターモデルは 2 極にしているため機械角 1 回転で電気角は 2 回転となるためです。

アングルセンサーの出力は回転角に応じてインクリメントするので同じ回転方向の場合、数値は増加します。そのため、一度三角関数の計算を入れることで回転系の座標に変換し、その後極座標変換で現在の位相値を求めています。

次の [phase->ON/OFF] ブロックもサブシステムコンポーネントとなっています。

各相の ON/OFF の条件分岐については、Table 1 のようになります。一回転を  $2\pi$  として 6 分割し、各位置にて ON する相を示しています。

この表を基に、条件分岐を使い実装したのが Figure 8 になります。

モデルは比較器コンポーネントと論理演算子を使用して実装しています。

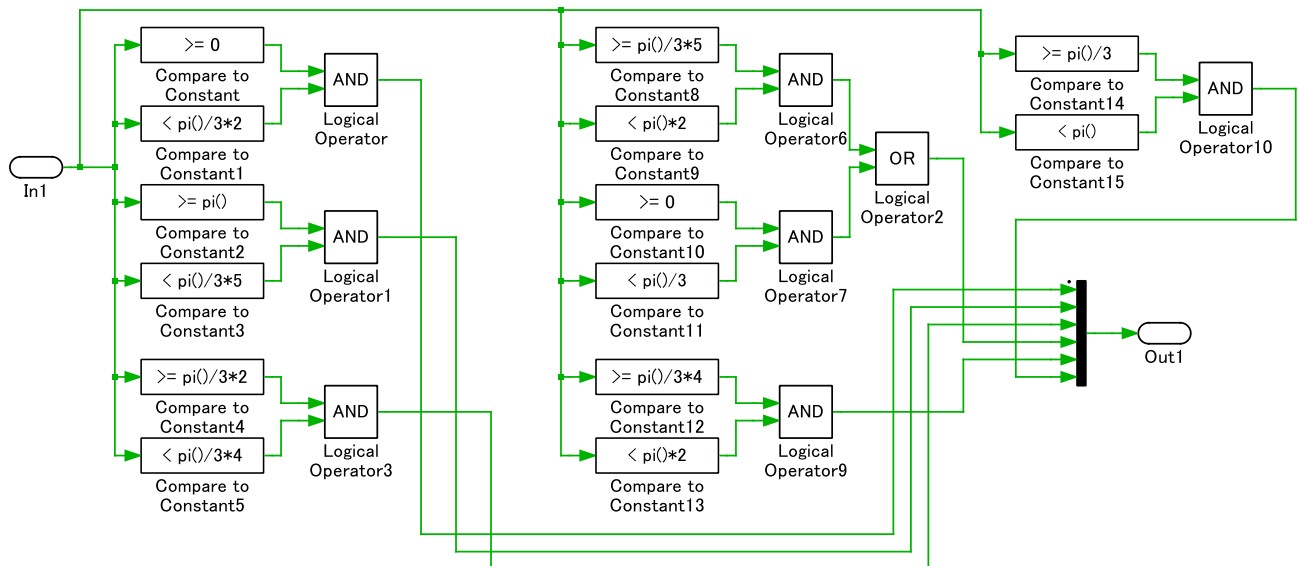


Figure 8

Table 1

位相	ドライバ段 各スイッチ					
	U相 H側	U相 L側	V相 H側	V相 L側	W相 H側	W相 L側
0 ~60°	ON	OFF	OFF	ON	OFF	OFF
60° ~120°	ON	OFF	OFF	OFF	OFF	ON
120° ~180°	OFF	OFF	ON	OFF	OFF	ON
180° ~240°	OFF	ON	ON	OFF	OFF	OFF
240° ~300°	OFF	ON	OFF	OFF	ON	OFF
300° ~360°	OFF	OFF	OFF	ON	ON	OFF

出力する信号は6線ありますが、デマルチプレクサコンポーネントにより1本にまとめています。配線は外観上同じですが、Out 端子に接続されている配線は6種類の信号が含まれる信号線となります。

このような PLECS のライブラリにあるコンポーネントを使用した場合、条件増えると、モデルの規模が大きくなってしまいます。こういった場合、PLECS では C-Script コンポーネントを使って、C 言語を実装することもできます。

C-Script コンポーネントを使うことで、すでにできあがっている数式などを流用することも可能です。

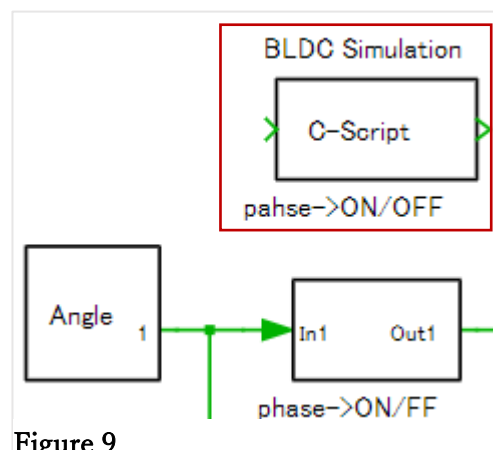


Figure 9

```

C言語パラメータ: BLDC_HOLE_mode2(SIM)3/pahse->ON/OFF
指定 コード入力
出力開数
1 float phase=0;
2 int sector=0;
3
4 //convert phase to step
5 phase = PLACE ; //input rad
6 if (phase>=0 && phase <(PI/3)) (sector = 1;
7 if (phase)>=(PI/3) && phase <(PI/3*2)) (sector = 2;
8 if (phase)>=(PI/3*2) && phase <PI) (sector = 3;
9 if (phase)>=PI && phase <(PI/3*4)) (sector = 4;
10 if (phase)>=(PI/3*4) && phase <(PI/3*5)) (sector = 5;
11 if (phase)>=(PI/3*5) && phase <(2*PI)) (sector = 6;
12
13 //Set Output Value
14
15 if (sector == 1){
16 UH=PWM_matrix[0][0];
17 UL=PWM_matrix[0][1];
18 VH=PWM_matrix[0][2];
19 VL=PWM_matrix[0][3];
20 WH=PWM_matrix[0][4];
21 WL=PWM_matrix[0][5];
22 }
23 else if (sector == 2){
24 UH=PWM_matrix[1][0];
25 UL=PWM_matrix[1][1];
26 VH=PWM_matrix[1][2];
27 VL=PWM_matrix[1][3];
28 WH=PWM_matrix[1][4];

```

Figure 11

```

C言語パラメータ: BLDC_HOLE_mode2(SIM)3/pahse->ON/OFF
指定 コード入力
宣言文
1 #include <float.h>
2 #include <math.h>
3
4
5 #define PLACE Input(0)
6
7 #define UH Output(0)
8 #define UL Output(1)
9 #define VH Output(2)
10 #define VL Output(3)
11 #define WH Output(4)
12 #define WL Output(5)
13
14 #define PI 3.14159265358979323846
15
16 //PWM Matrix
17
18 int PWM_matrix[6][6]=
19 {{1,0,0,1,0,0},
20 {1,0,0,0,1,1},
21 {0,0,1,0,0,1},
22 {0,1,1,0,0,0},
23 {0,1,0,0,1,0},
24 {0,0,0,1,1,0}};
25
26
27

```

Figure 10

Figure 8 は Figure 10 と Figure 11 のように記述することが可能です

### (3) PWM

モーターに印可する電流は PWM 制御で調整します。今回は速度制御を含まず、無負荷の状態でも回転させています。その為、PWM の ON Duty が大きいとモーター電流が大きくなりトルクが増加します。負荷が一定であれば回転させる力が大きくなるため回転数が高くなります。

PWM の信号の生成には PWMOUT コンポーネントを使用します。このコンポーネントは、設定した三角波の幅に対して入力した値がクロスした時点で ON します。入力は 0 ~ 2 の範囲で入力し、2 の時に Duty が 100% となります。しかし、このコンポーネントでは 0 を入力しても、三角波の下端に位置しているため一瞬 ON してしまいます。

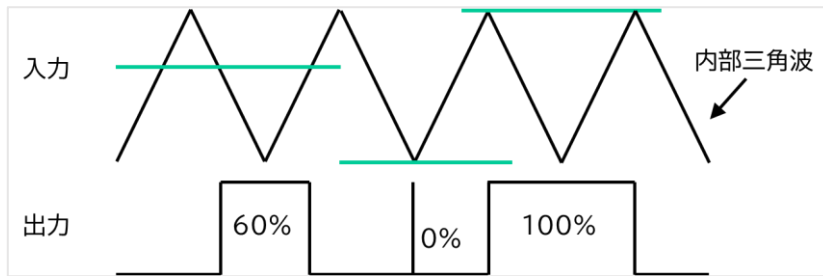


Figure 12

後段のドライバは H 側と L 側が同じタイミングで動作しているため、一瞬でも H と L が同時 ON すると短絡になりシミュレーションはエラーを出します。今回は、入力が「0」以外は PWMOUT の出力を切断し「0」をドライバに入力するようにスイッチをいれました。

PWM レベルが 0 の場合の条件分岐で、PWMOUT のスイッチを ON/OFF 制御しています。

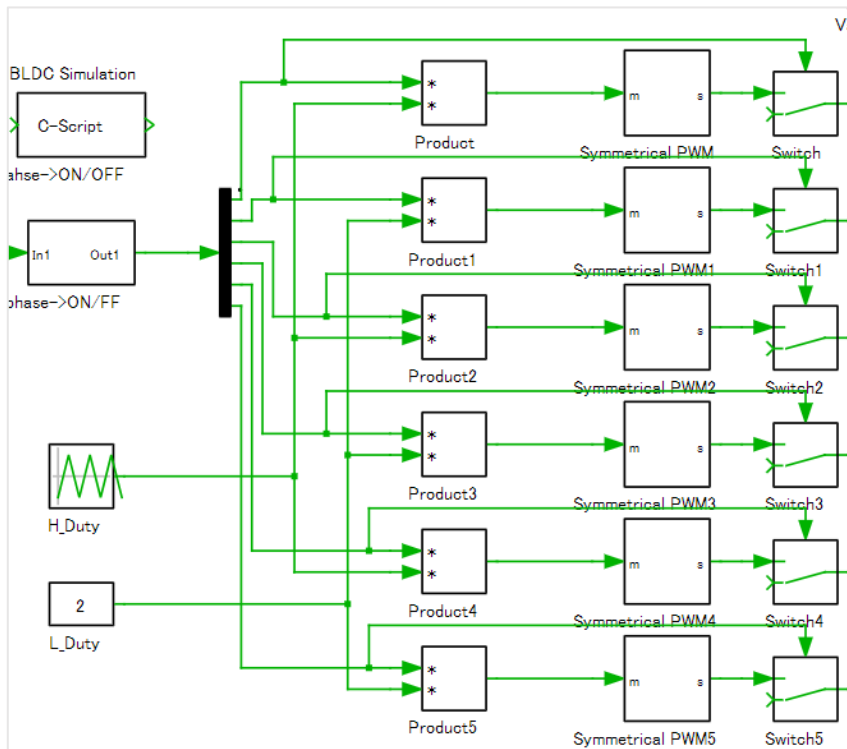


Figure 13

PWMDuty について、今回のような 6 セクションの場合、L 側は ON 固定で H 側だけ PWM 制御することで励磁電流をコントロールできます。各相を ON するか OFF するかは 1 か 0 のため、ON する場合は PWMDuty を定数で設定し、かけ算した結果をドライバへ入力しています。

(4) ドライバ

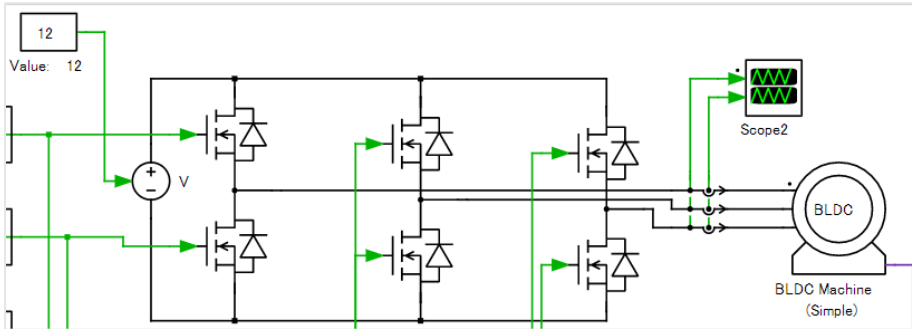


Figure 14

ドライバは三相モーターの為 3レグ6アームのハーフブリッジ回路を使用します。PLECS 標準コンポーネントの三相ハーフブリッジドライバを使います。

このコンポーネントは各アームにたいして「1」を入れると ON し「0」は OFF になります。

前段の PWMOUT からの信号を受けて、実際にモーターに対して PWM の信号を与えます。

3. シミュレーション

3.1 シミュレーション設定

メニューよりシミュレーションパラメーターを選択すると、実行するシミュレーションの設定をすることができます。

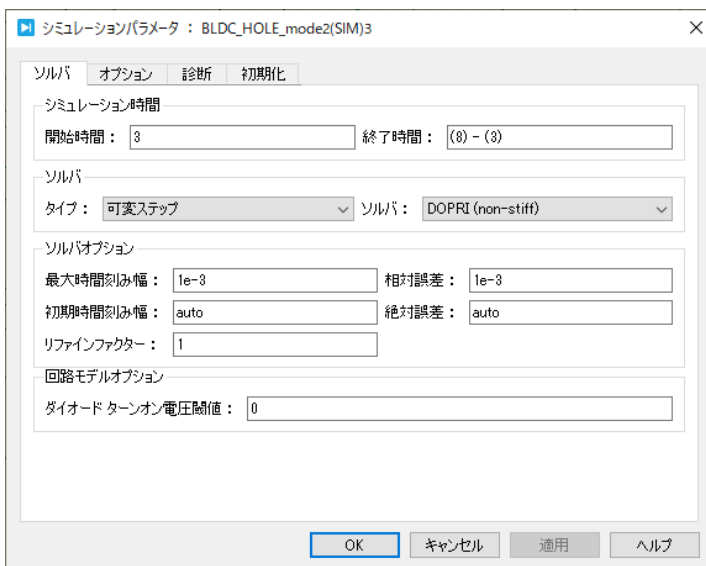
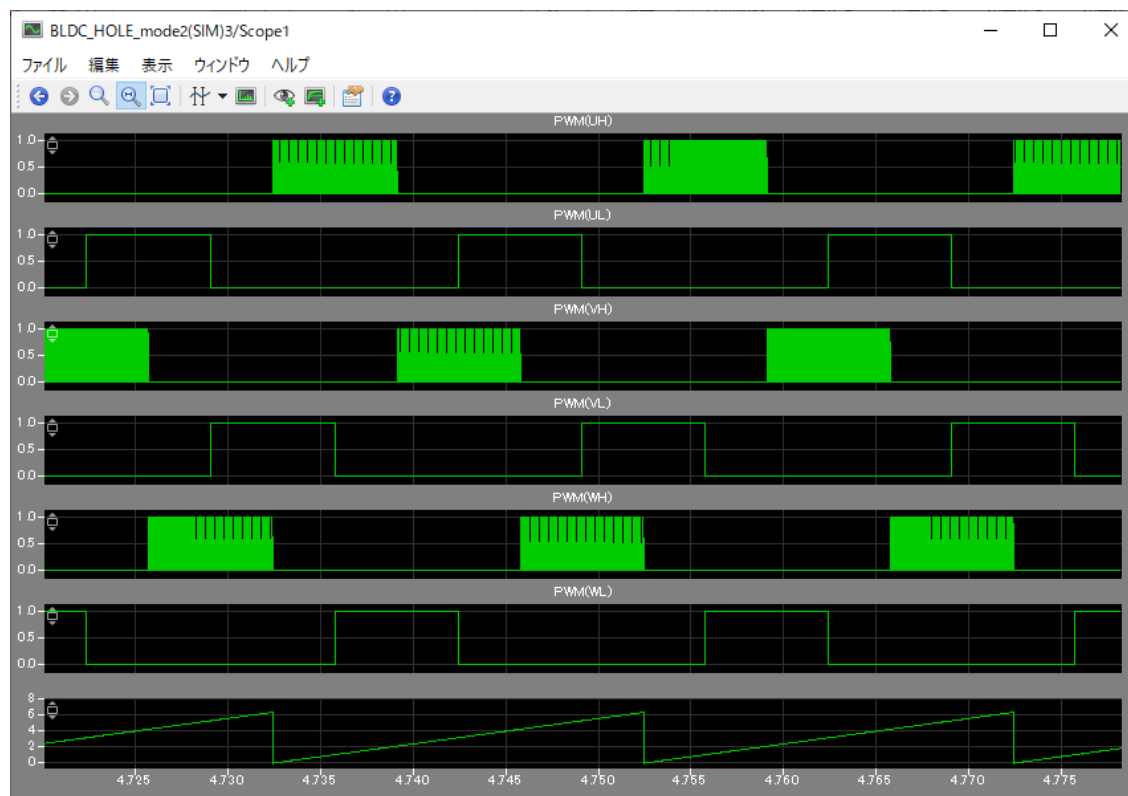


Figure 15

### 3.2 シミュレーション結果

PLECS は Scope コンポーネントに接続されたノードの波形を見ることができ、シミュレーション実行中もリアルタイムで波形を確認することができます。このシミュレーションは CPU:Corei5-8256U RAM:8G で約 20 秒でした。



**Figure 16**

最下段は位相角の波形で、この信号から 6 セクションのどの位置にいるか把握し各相の ON/OFF を決定しているのが観察できます。

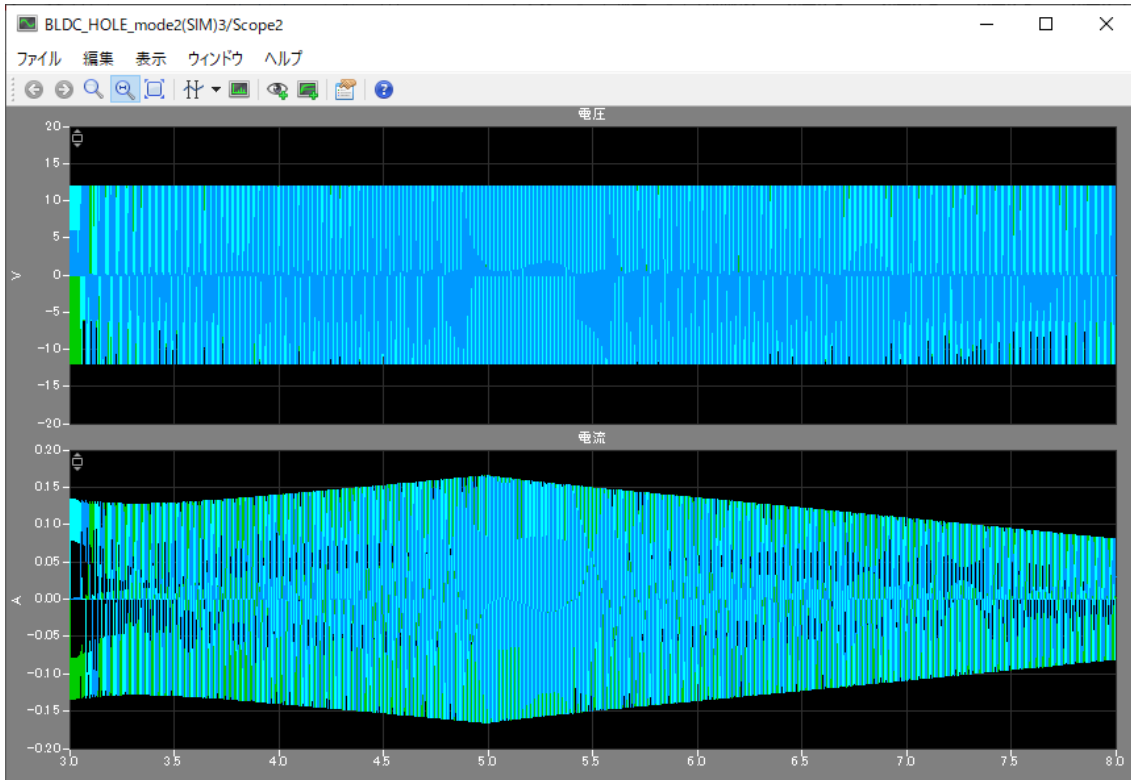


Figure 17

モーター電流は Duty によって変化しているのが観察できます。

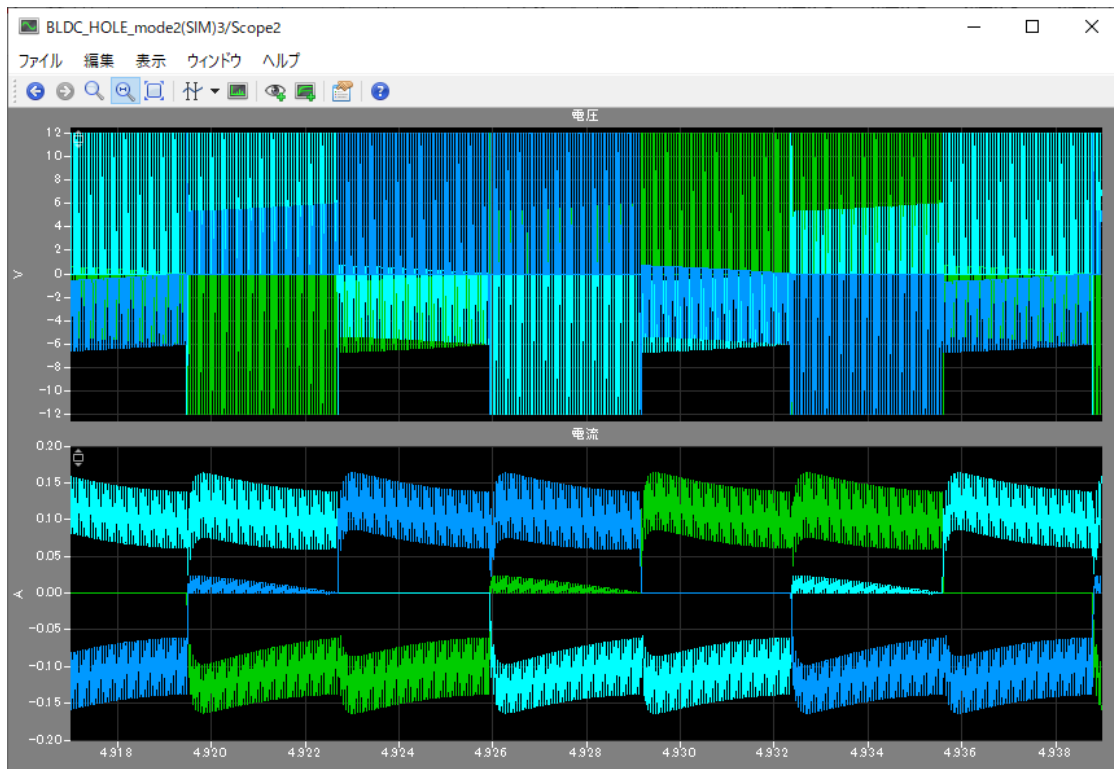
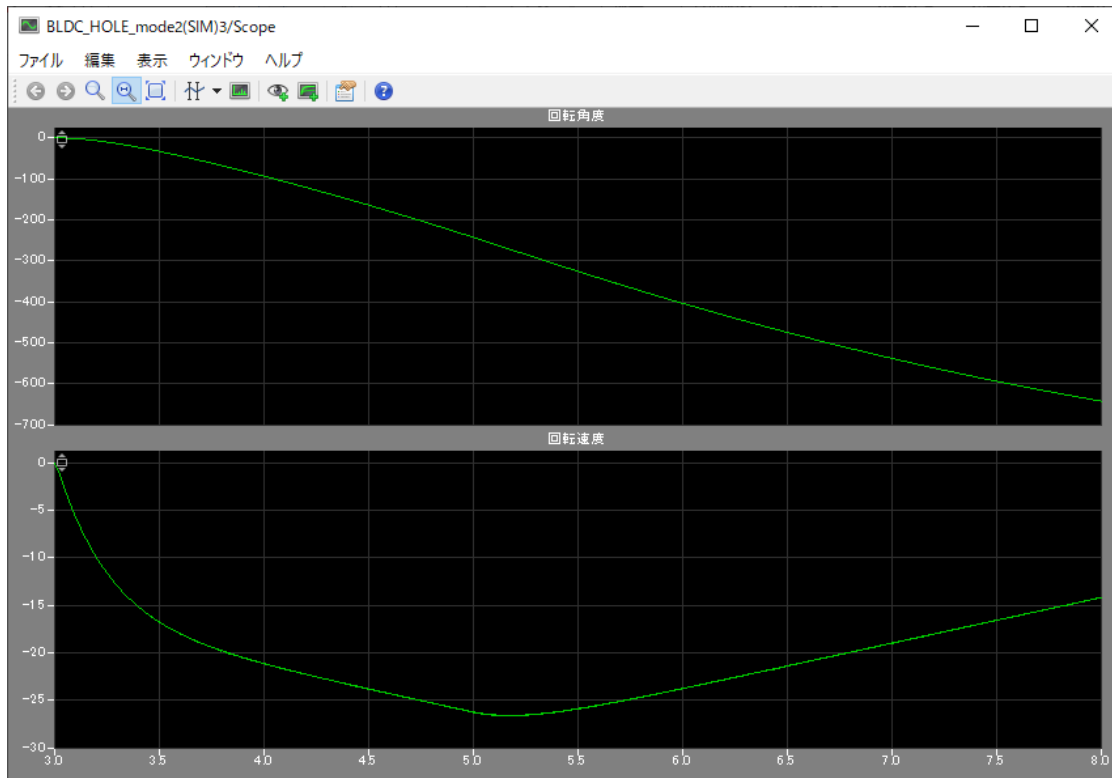


Figure 18

スコープ上で拡大したい部分をドラッグすることで波形の拡大も可能です。



**Figure 19**

PWMDuty の変化に伴って、回転速度が変化しているのが確認できました。

#### 4. SIL (Simulation In the Loop) の実行

モデル上でモーターが回転できることが確認できましたので、次にコントローラーを PLECS RT-BOX とし、実際にモーターを回転させる SIL を実行していきます。

##### 4.1 モデルの変更

実機を動作させるため、このモデルのドライバ部と位置検出を外部ドライバとホールセンサを使用する形にモデルを変更します。外部との接続部は RT-BOX のコンポーネント Digital IN と PWMOUT を使用します。そして、3 相ドライバとモーター、回転角を得るためのホールセンサに置き換えます。

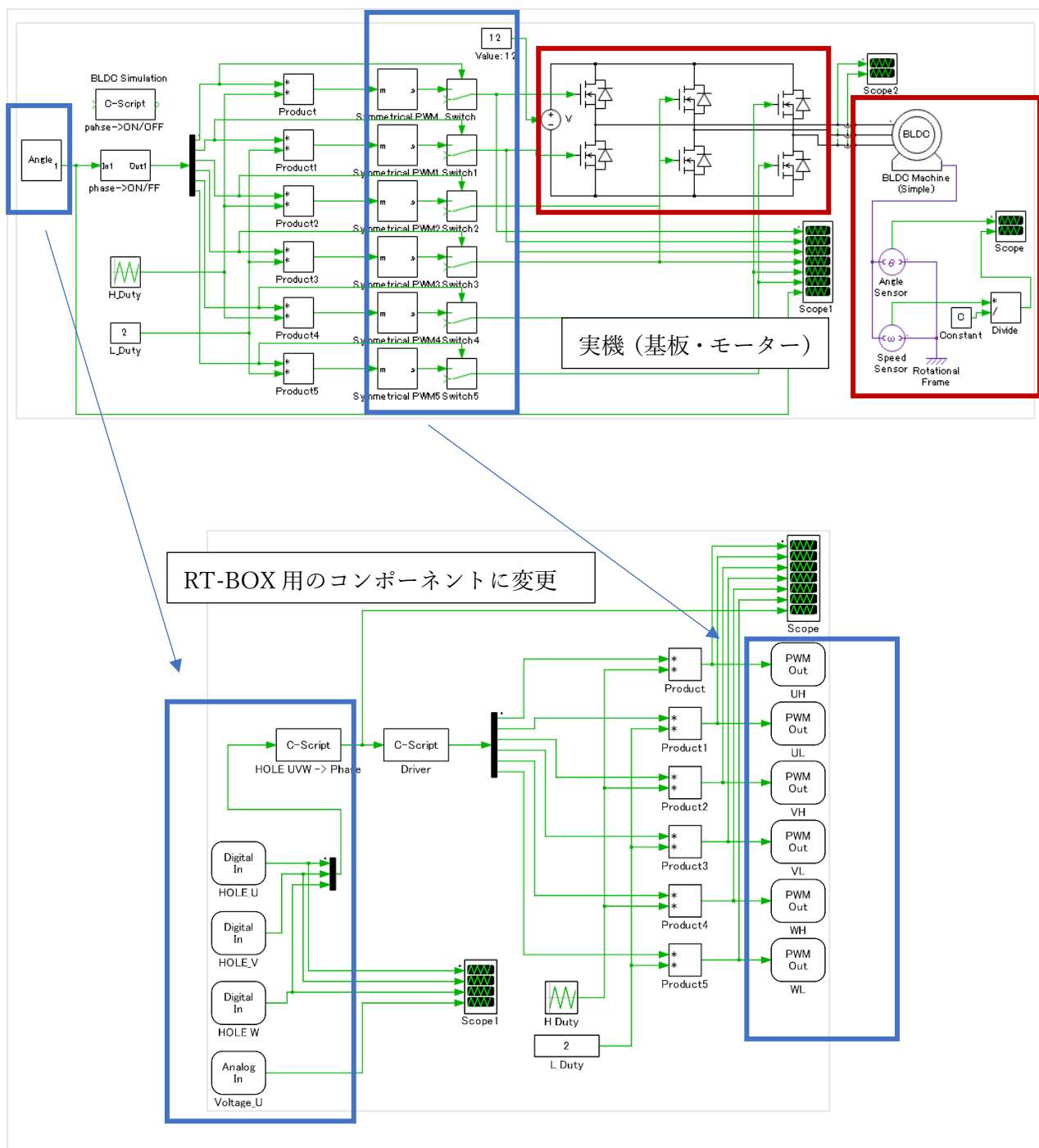


Figure 20

モーター内のホールセンサは RT-BOX の Digital In に接続され、角度情報を演算します。基板は RT-BOX のデジタル入出力が 5V か 3.3V の選択となるため、それに合うように製作しています。

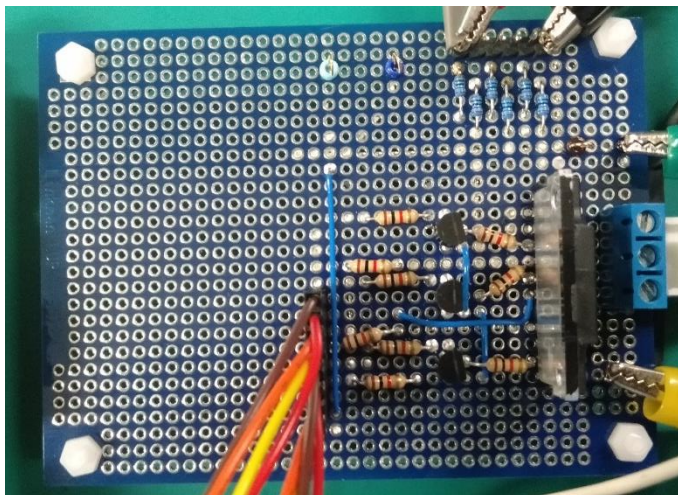
また、ドライバ制御は各相の ON/OFF 指令に対して、PWM の Duty 情報を付加し PWMOUT コンポーネントに入力します。PWMOut はシミュレーションの離散ステップとは独立して PWM の波形を出力します。RT-BOX の Digital Out 端子からの信号はドライバ基板に接続され、電源電圧が 12V のドライバでモーターを駆動します。

## 4.2 モーターの選定・基板の製作

### (1) モーター

モーターは コアレスモーター社製の CPH50 を使用しました

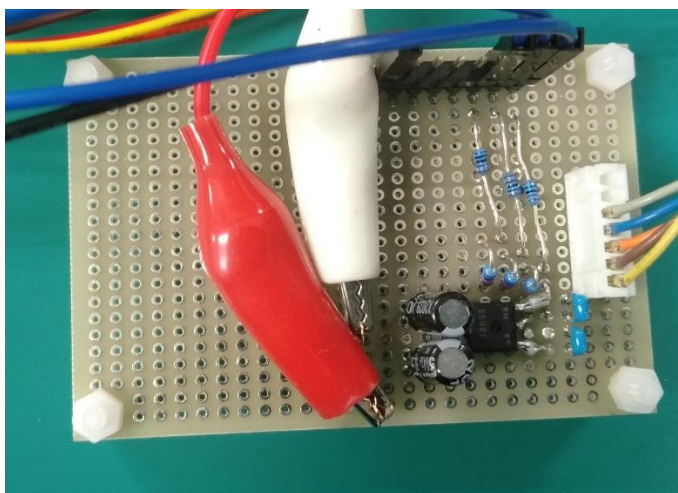
### (2) 三相ドライバ基板



**Figure 21**

三相ドライバは汎用のドライバ IC を使用しています。(使用 IC: S L A 6 0 2 0)

### (3) ホールセンサインターフェース基板



**Figure 22**

モーターに内蔵されているホールセンサは 5V 電源のオープンコレクタ出力のため、プルアップ抵抗を基板上に配置しました。RT-BOX のデジタル入出力は 3.3V か 5V 系か選択可能です。入力についてはトレラント機能があるため 5V を入力することができます。

## 4.3 モデルの設定

### (1) PWM Out

PWM Out コンポーネントの設定は以下のようになっています。

RT-BOX の PWM 機能はシミュレーションの離散ステップとは独立して動作するため、キャリア周波数を設定するとシミュレーション設定のステップに影響されず設定した周波数で動作します。

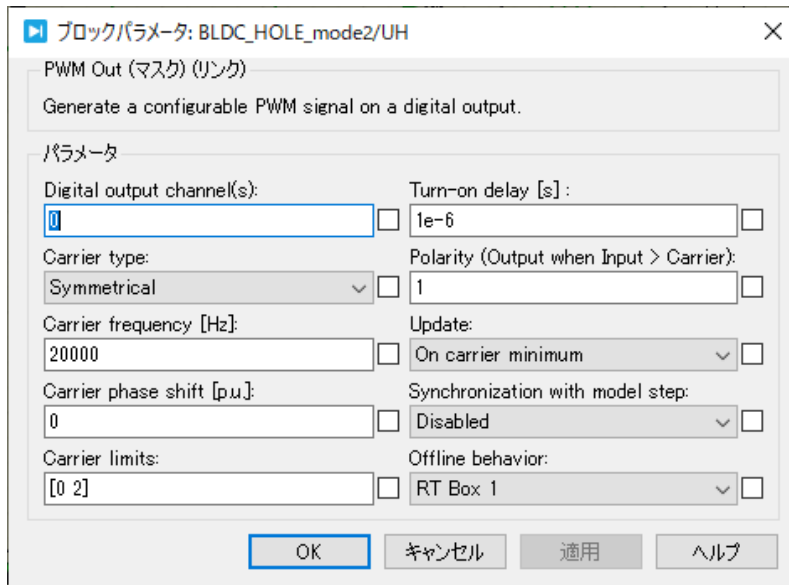


Figure 23

### (2) Digital IN

Digital IN 端子は離散ステップごとに端子の状態を読み込みます。

チャンネルの指定および、プルアップやプルダウン設定をすることができます。

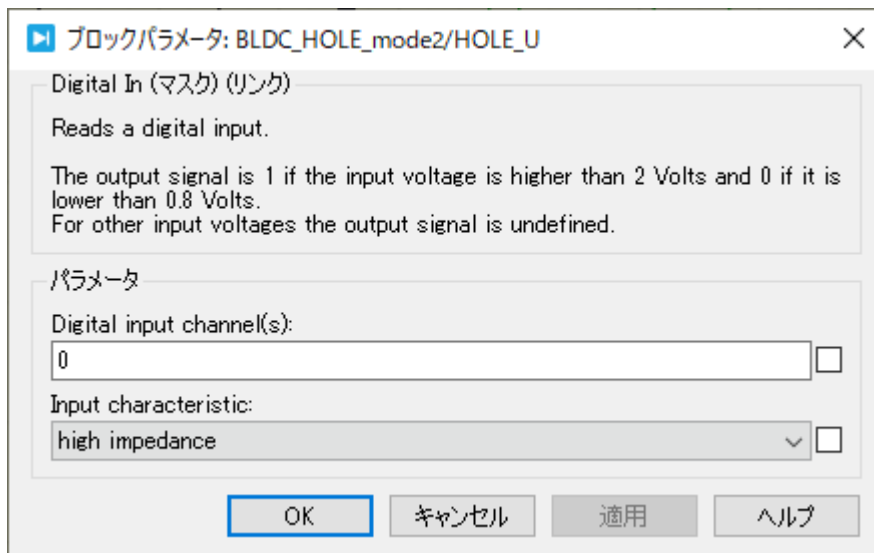


Figure 24

## 4.4 SIL シミュレーションの実行

### (1) シミュレーション設定

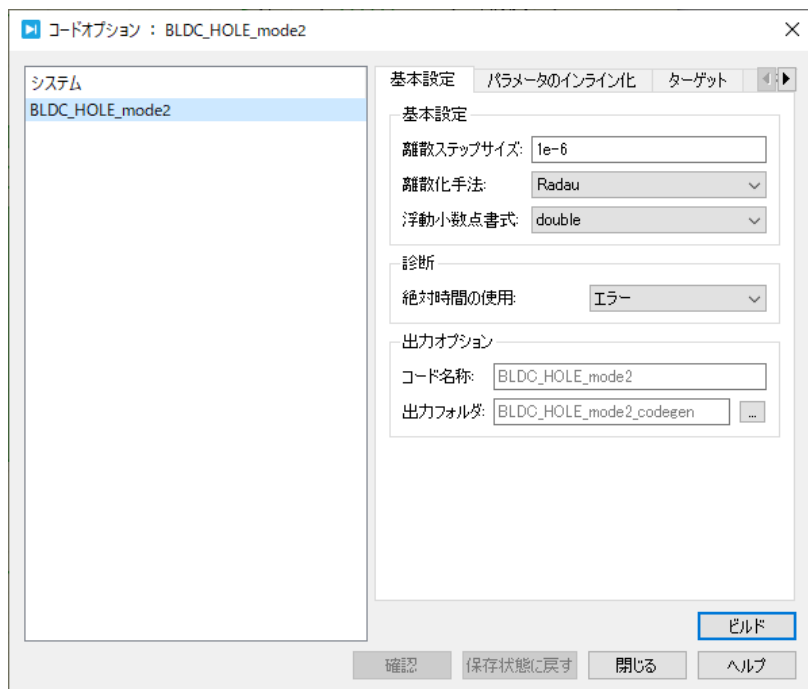
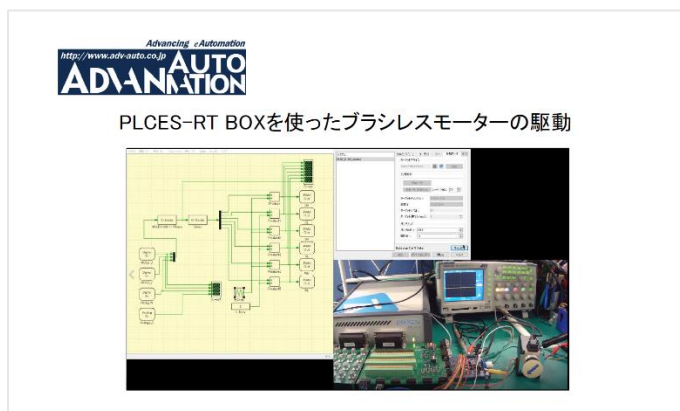


Figure 25

分散ステップサイズは RT-BOX の動作する周期を指定します。モデル内の演算がこの周期内に完了するように設定する必要があります。前項で記載していますが、PWM の出力はこの分散ステップとは独立しているため、分散ステップが長くなったとしても PWM の動作に影響はありません。

### (2) シミュレーション結果

実行したシミュレーションについて、動画にて掲載をしています。ぜひご覧ください。



## 5. おわりに

今回のデモでは PLECS で簡単にモデルの作成ができ、また RT-BOX を使用することで簡単にシミュレーション、実機動作テストが可能となることをご紹介しました。

今回はコントローラー側を PLECSRT-BOX で実装しましたが逆にモーターのモデルを使用し、コントローラーを実機でシミュレーションすることも可能です。これにより、実機では大電力となるようなシステムのコントローラーも検証・テストすることが可能です。

PLECS は 90 日間の試用もしくは機能制限状態で利用することができます。

また、弊社アドバンオートメーションのページにチュートリアル日本語字幕版や日本語版マニュアルもございます。是非、操作をお試しく下さい。

(日本語字幕チュートリアルと日本語版マニュアルへのコンテンツアクセスにはパスワードが必要です。アドバンオートメーション web サイトよりご請求ください)

### ■アドバンオートメーション PLECS 関連ページ

<https://www.adv-auto.co.jp/products/plexim/>

### ■PLEXIM 社 Web ページ

<https://www.plexim.com/ja>

本技術情報は、お客様にシステムのご紹介をすることが目的です。本技術情報が提供する回路、シミュレーション結果、部品表などは、あくまでも参考であり、例示又は推奨するものではありません。