

令和 元年度 卒業論文

OpenModelica のシミュレーション結果を 用いたモータ特性表自動生成ツールの試作

指導教員 片山 徹郎 教授

宮崎大学 工学部 情報システム工学科

原田 海人

2020 年 1 月

目次

1	はじめに	1
2	研究の準備	2
2.1	ブラシ付き DC モータ	2
2.2	対応するモデル	2
2.2.1	モータ単体の Modelica モデル	2
2.2.2	モータ単体の Modelica モデルをサブシステムとするモデル	3
2.2.3	モデル作成時の制約	4
2.3	OpenModelica	4
2.3.1	出力	4
2.3.2	グラフィカルモデリング	5
2.4	Modelica 言語	6
2.4.1	Modelica 標準ライブラリ (MSL)	6
2.5	モータ特性表	6
2.6	Python	7
3	機能	8
3.1	モータ特性表生成	8
3.1.1	特性表	8
3.1.2	特性グラフ	10
3.2	ツールの実行	11
3.3	エラー表示	14
4	実装	15
4.1	csv ファイルの読み込み	15
4.2	特性表の各要素を算出するために必要なデータを取得	15
4.3	特性表の各要素を算出	16
4.3.1	電圧	16

4.3.2	始動電流	16
4.3.3	停動トルク	16
4.3.4	最大効率	17
4.3.5	定格トルク	17
4.3.6	定格回転数	17
4.3.7	定格電流	17
4.3.8	定格出力	17
4.3.9	最大回転数	18
4.4	特性表を生成	18
5	適用例	19
5.1	モータ単体のモデル	19
5.2	モータ単体の Modelica モデルをサブシステムとするモデル	19
6	考察	20
6.1	評価	20
6.1.1	評価方法	20
6.1.2	結果	20
6.2	関連研究	20
6.3	ツールの問題点	20
7	おわりに	21
	謝辞	22
	参考文献	23

第 1 章

はじめに

近年、モータは、エアコン・洗濯機・掃除機などの家電製品をはじめ、自動車関係、医療関係など様々な分野に用いられており [1]、社会に必要不可欠な存在となっている。

～はじめに 流れ 案 1～

シミュレーションを行った場合、期待通りか確認する。

確認する際は、シミュレーション結果から目的のグラフや値を計算等して作成しなければならない。

今回試作したツールで、グラフや値を作成する手間を省くことで、モータ開発の効率化を図る。

本論文の構成は、以下の通りである。

第 2 章では、モータ特性表自動生成ツールを試作するために必要となる前提知識について説明する。

第 3 章では、試作したモータ特性表自動生成ツールの機能について説明する。

第 4 章では、モータ特性表自動生成ツールの実装について説明する。

第 5 章では、試作したモータ特性表自動生成ツールが正しく動作することを検証する。

第 6 章では、試作したモータ特性表自動生成ツールについて考察する。

第 7 章では、本論文のまとめと今後の課題を述べる。

第 2 章

研究の準備

本章では、本研究で必要となる前提知識を説明する。

2.1 ブラシ付き DC モータ

今回試作するツールでは、ブラシ付きモータのシミュレーション結果に対応する。ブラシ付きモータとは、――

2.2 対応するモデル

試作するモータ特性表自動生成ツールでは、以下の Modelica モデルのシミュレーション結果に対応する。

- モータ単体の Modelica モデル !!!
- モータ単体の Modelica モデルをサブシステムとするモデル

以降、上記のモデルについて具体的に説明する。

2.2.1 モータ単体の Modelica モデル

!!! モータ単体の話ではなくブラシ付き DC モータの話になっている。モータ単体の Modelica モデルとは、電源部品、抵抗部品、インダクター部品、起電力部品、慣性部品、接地部品を持つモデルのことである。

表 2.1: MSL 対応表

部品名	使用する MSL
電源部品	Modelica.Electrical.Analog.Sources
抵抗部品	Modelica.Electrical.Analog.Basic
インダクター部品	Modelica.Electrical.Analog.Basic
起電力部品	Modelica.Electrical.Analog.Basic
慣性部品	Modelica.Mechanics.Rotational.Components
接地部品	Modelica.Electrical.Analog.Basic

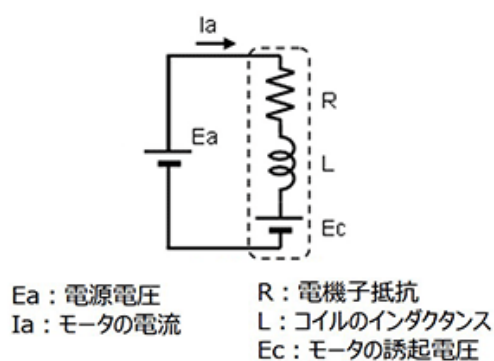


図 2.1: ブラシ付き DC モータの等価回路

上記 6 つの部品が必要な理由は、ブラシ付き DC モータの等価回路 [2] を Modelica 言語で表す際に、使用する部品 [3] だからである。

各部品で使用する MSL を表 2.1 に、ブラシ付き DC モータの等価回路を図 2.1 に、モータ単体の Modelica モデルの例を図 2.2 に、図 2.2 の Modelica コードを図 2.3 に、それぞれ示す。

2.2.2 モータ単体の Modelica モデルをサブシステムとするモデル

モータ単体の Modelica モデルをサブシステム [3] とするモデルとは、2.2.1 節で説明したモータ単体の Modelica モデルを 1 つのサブシステムとして扱い、他の部品と合わせたモデルのことである。

例として、DC モータのサブシステムを用いた DC モータサーボのモデルを図 2.4 に、図 2.4 の

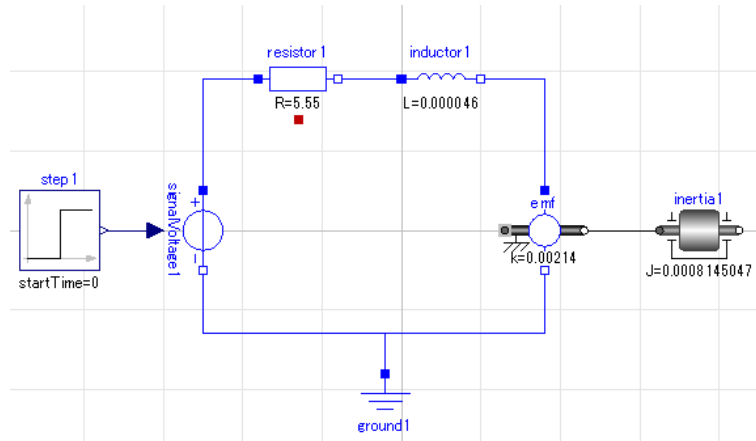


図 2.2: モータ単体の Modelica モデルの例

Modelica コードを図 2.5 に、それぞれ示す。

2.2.3 モデル作成時の制約

2.2.1 章、2.2.2 章で説明した制約の他に、以下の制約も満たしていなければならない。

- 入力は 0 秒スタート
- 電圧値は一定
- 各部品に名前をつける際は、デフォルトの名前にする
- モータのモデルは 1 つまで

2.3 OpenModelica

Modelica 言語に対応した OSS である [4]。

2.3.1 出力

OpenModelica では、シミュレーション結果の保存先を、以下の 3 つの形式から選択することができる。

- mat ファイル

```

1 model DCmotor
2 Modelica.Electrical.Analog.Basic.Resistor resistor1(R = 5.55, T = 283.15) annotation(
3   Placement(visible = true, transformation(origin = {-26, 38}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
4 Modelica.Electrical.Analog.Basic.Inductor inductor1(L = 0.000046) annotation(
5   Placement(visible = true, transformation(origin = {10, 38}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
6 Modelica.Electrical.Analog.Basic.Ground ground1 annotation(
7   Placement(visible = true, transformation(origin = {-4, -46}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
8 Modelica.Mechanics.Rotational.Components.Inertia inertial(J = 0.0008145047, a(start = 0), phi(start = 0), w(start = 0)) annotation(
9   Placement(visible = true, transformation(origin = {75, 0}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
10 Modelica.Electrical.Analog.Basic.EMF emf(k = 0.00214, useSupport = false) annotation(
11   Placement(visible = true, transformation(origin = {36, 0}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
12 Modelica.Electrical.Analog.Sources.SignalVoltage signalVoltage1 annotation(
13   Placement(visible = true, transformation(origin = {-50, 0}, extent = {{-10, 10}, {10, -10}}, rotation = -90)));
14 Modelica.Blocks.Sources.Step step1(height = 2.4, startTime = 0) annotation(
15   Placement(visible = true, transformation(origin = {-86, 0}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
16 equation
17 connect(step1.y, signalVoltage1.v) annotation(
18   Line(points = {{-75, 0}, {-62, 0}}, color = {0, 0, 127}));
19 connect(emf.flange, inertial.flange_a) annotation(
20   Line(points = {{46, 0}, {65, 0}}));
21 connect(signalVoltage1.p, resistor1.p) annotation(
22   Line(points = {{-50, 10}, {-50, 38}, {-36, 38}}, color = {0, 0, 255}));
23 connect(resistor1.n, inductor1.p) annotation(
24   Line(points = {{-16, 38}, {0, 38}}, color = {0, 0, 255}));
25 connect(inductor1.n, emf.p) annotation(
26   Line(points = {{20, 38}, {36, 38}, {36, 10}}, color = {0, 0, 255}));
27 connect(ground1.p, signalVoltage1.n) annotation(
28   Line(points = {{-4, -36}, {-4, -26}, {-50, -26}, {-50, -10}}, color = {0, 0, 255}));
29 connect(emf.n, signalVoltage1.n) annotation(
30   Line(points = {{36, -10}, {36, -26}, {-50, -26}, {-50, -10}}, color = {0, 0, 255}));
31 annotation(
32   uses(Modelica(version = "3.2.3")));
33 end DCmotor;

```

図 2.3: 図 2.2 の Modelica コード

- plt ファイル

- csv ファイル

csv ファイルに書かれている内容について説明！

csv ファイルの名前の生成規則について説明！

図 2.2 のモデルをシミュレーションした時に、OpenModelica から出力される csv ファイルの一部を、図 3.1 に示す。

また、OpenModelica では、シミュレーション結果を、グラフとして画面上に描画することが可能である。

2.3.2 グラフィカルモデリング

サブセクションで書くなれば何か他にも書く。

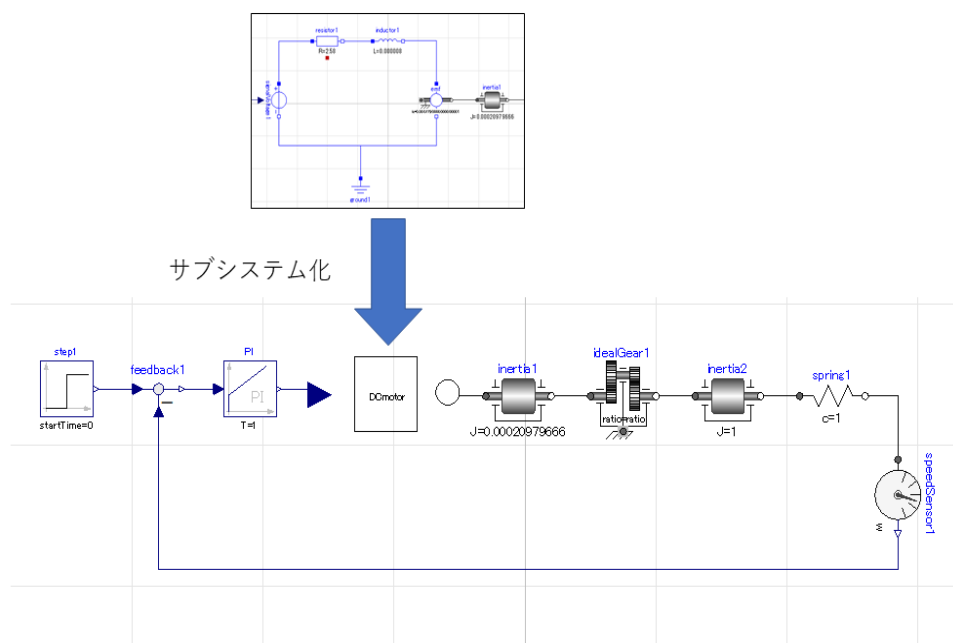


図 2.4: DC モータサーボのモデル

2.4 Modelica 言語

微分代数方程式を用いた複合領域の物理システムモデリングのために開発されたオブジェクト指向言語である。

2.4.1 Modelica 標準ライブラリ (MSL)

Modelica 言語による様々な物理領域のモデルライブラリを開発しており、数学、機械、電気、熱、流体、制御系、状態遷移機械などを含んだフリーのライブラリがリリースされている。

2.5 モータ特性表

モータ特性表とは、モータを選定する際に、参考にする資料である。一般的に決まった形式はなく、各会社によって書いている要素は異なるため、10社のモータ特性表をもとに、作成する

```

1 model submodel
2   Modelica.Blocks.Sources.Step step1(height = 1.5) annotation(
3     Placement(visible = true, transformation(origin = {-70, 48}, extent = {{-4, -4}, {4, 4}}, rotation = 0)));
4   Modelica.Blocks.Math.Feedback feedback1 annotation(
5     Placement(visible = true, transformation(origin = {-56, 48}, extent = {{-4, -4}, {4, 4}}, rotation = 0)));
6   Modelica.Blocks.Continuous.PI PI(T = 1) annotation(
7     Placement(visible = true, transformation(origin = {-42, 48}, extent = {{-4, -4}, {4, 4}}, rotation = 0)));
8   Modelica.Mechanics.Rotational.Components.IdealGear idealGear1 annotation(
9     Placement(visible = true, transformation(origin = {13, 49}, extent = {{-5, -5}, {5, 5}}, rotation = 0)));
10  Modelica.Mechanics.Rotational.Components.Inertia inertia2(J = 1) annotation(
11    Placement(visible = true, transformation(origin = {31, 49}, extent = {{-5, -5}, {5, 5}}, rotation = 0)));
12  Modelica.Mechanics.Rotational.Components.Spring spring1(c = 1) annotation(
13    Placement(visible = true, transformation(origin = {47, 49}, extent = {{-5, -5}, {5, 5}}, rotation = 0)));
14  Modelica.Mechanics.Rotational.Components.Inertia inertia3(J = 0.00020979666) annotation(
15    Placement(visible = true, transformation(origin = {194, 14}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
16  Modelica.Mechanics.Rotational.Sensors.SpeedSensor speedSensor1 annotation(
17    Placement(visible = true, transformation(origin = {57, 33}, extent = {{-5, -5}, {5, 5}}, rotation = -90)));
18  pack_iner1 pack_iner1 annotation(
19    Placement(visible = true, transformation(origin = {-16, 48}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
20 equation
21  connect(idealGear1.flange_a, pack_iner1.flange_b) annotation(
22    Line(points = {{8, 49}, {8, 49.5}, {-6, 49.5}, {-6, 48}}));
23  connect(idealGear1.flange_b, inertia2.flange_a) annotation(
24    Line(points = {{18, 49}, {26, 49}}));
25  connect(spring1.flange_b, speedSensor1.flange) annotation(
26    Line(points = {{52, 49}, {57, 49}, {57, 38}}));
27  connect(inertia2.flange_b, spring1.flange_a) annotation(
28    Line(points = {{36, 49}, {42, 49}}));
29  connect(PI.y, pack_iner1.u) annotation(
30    Line(points = {{-38, 48}, {-28, 48}}, color = {0, 0, 127}));
31  connect(speedSensor1.w, feedback1.u2) annotation(
32    Line(points = {{57, 27}, {57, 22.5}, {-56, 22.5}, {-56, 45}}, color = {0, 0, 127}));
33  connect(feedback1.y, PI.u) annotation(
34    Line(points = {{-52.4, 48}, {-46.8, 48}}, color = {0, 0, 127}));
35  connect(step1.y, feedback1.u1) annotation(
36    Line(points = {{-65.6, 48}, {-59.2, 48}}, color = {0, 0, 127}));
37  annotation(
38    uses(Modelica(version = "3.2.3"));end submodel;

```

図 2.5: 図 2.4 の Modelica コード

モータ特性表の要素を決定した。

以下にモータ特性表の構成と、要素を示す。

2.6 Python

!!! 未調査 !!!

第 3 章

機能

本章では、本研究で試作するモータ特性表自動生成ツールの機能について説明する。

Modelica 言語で作成したモータのモデルを、OpenModelica でシミュレーションした際に csv ファイルが出力される。

今回試作するモータ特性表自動生成ツールは、OpenModelica が出力した csv ファイルを入力として読み込み、モータ特性表を出力として生成する。

3.1 モータ特性表生成

今回試作したモータ特性表自動生成ツールは、9 個の要素を持つ特性表と、4 つの特性グラフを作成し、それらを 1 つの PDF ファイルにまとめ、モータ特性表として出力する。

3.1.1 特性表

特性表を構成する 9 個の要素は、以下の通りである。

- 電圧 V
- 始動電流 mA
- 停動トルク mNm
- 最大効率 %

- 定格トルク mNm
- 定格回転数 rpm
- 定格電流 mA
- 定格出力 W
- 最大回転数 rpm

以降、各要素が表す内容について述べる。

電圧

電圧とは、シミュレーション時に、回路に印加された電圧値を表す。

単位は、V(ボルト) である。

始動電流

始動電流とは、モータの起動時に流れる電流値を表す。

単位は、mA(ミリアンペア) である。

停動トルク

停動トルクとは、モータが出しうる最大トルクで、このトルク以上の負荷がかかれば、モータは停止する値を表す。

単位は、mNm(ミリニュートンメートル) である。

最大効率

効率とは、入力電力に対する機械出力の比を百分率 [%] で表したものであり、最大効率は、その中でも一番大きい値を表す。

単位は、%(パーセント) である。

定格トルク

定格トルクとは、最大効率時のトルク値を表す。

単位は、mNM(ミリニュートンメートル) である。

定格回転数

定格回転数とは、最大効率時の回転数値を表す。

単位は、rpm(アールピーエム) である。

定格電流

定格電流とは、最大効率時の電流値を表す。

単位は、mA(ミリアンペア) である。

定格出力

定格出力とは、最大効率時の出力値を表す。

単位は、W(ワット) である。

最大回転数

最大回転数とは、回転数値の中でも一番大きい値を表す。

単位は、rpm(アールピーエム) である。

3.1.2 特性グラフ

今回試作したツールでは、以下の4つのグラフを生成する。

- トルク mNM * 電流 mA
- トルク mNm * 回転数 rpm
- トルク mNm * 効率 %

- トルク mNm * 出力 W

以降、各グラフについて述べる。

「トルク * 電流」グラフ

「トルク * 電流」グラフとは、横軸が「トルク mNm」、縦軸が「電流 mA」で生成するグラフのことである。

「トルク * 回転数」グラフ

「トルク * 電流」グラフとは、横軸が「トルク mNm」、縦軸が「回転数 rpm」で生成するグラフのことである。

「トルク * 効率」グラフ

「トルク * 電流」グラフとは、横軸が「トルク mNm」、縦軸が「効率 %」で生成するグラフのことである。

「トルク * 出力」グラフ

「トルク * 電流」グラフとは、横軸が「トルク mNm」、縦軸が「出力 W」で生成するグラフのことである。

3.2 ツールの実行

今回試作したツールの実行するためのコマンドは、「python characteristic.py 第1引数 第2引数 第3引数」である。

なお、このコマンドは、ツールの実行ファイルが存在するディレクトリで実行する必要がある。

第1引数には、入力とする csv ファイルのパスを含めたファイル名を指定する。

第2引数には、第1引数で指定した csv ファイルの中の、計算したいモータのモデルに含まれる、慣性部品のオブジェクト名が必要である。

第3引数には、第1引数で指定した csv ファイルの中の、計算したいモータのモデルに含まれ

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	time	inductor1.i	inertial1.p	inertial1.w	der(induc	der(inertial	der(inertial	der(emf.p	emf.phi	emf.tau	emf.v	emf.w	ground1.p	ground1.p	inductor1.i	inductor1.i	inertial1.a	inertial1.f	resistor1.L	resistor1.F
2	0	0	0	0	187500	0	0	0	0	0	0	0	0	0	1.5	1.5	0	0	0	2.58
3	1	0.580744	1.078986	2.157574	-0.00065	2.157574	2.156371	2.157574	1.078986	-0.00045	0.001681	2.157574	0	0	0.001681	-5.21E-09	2.156371	0	0.87014	2.58
4	2	0.580093	4.314343	4.312737	-0.00065	4.312737	2.153955	4.312737	4.314343	-0.00045	0.00336	4.312737	0	0	0.00336	-5.20E-09	2.153955	0	0.868191	2.58
5	3	0.579443	9.703655	6.465485	-0.00065	6.465485	2.151542	6.465485	9.703655	-0.00045	0.005037	6.465485	0	0	0.005037	-5.20E-09	2.151542	0	0.866246	2.58
6	4	0.578794	17.24451	8.615821	-0.00065	8.615821	2.149131	8.615821	17.24451	-0.00045	0.006712	8.615821	0	0	0.006712	-5.19E-09	2.149131	0	0.864306	2.58
7	5	0.578145	26.9345	10.76375	-0.00065	10.76375	2.146723	10.76375	26.9345	-0.00045	0.008385	10.76375	0	0	0.008385	-5.19E-09	2.146723	0	0.86237	2.58
8	6	0.577498	38.7712	12.90927	-0.00065	12.90927	2.144317	12.90927	38.7712	-0.00045	0.010056	12.90927	0	0	0.010056	-5.18E-09	2.144317	0	0.860439	2.58
9	7	0.57685	52.75223	15.05238	-0.00065	15.05238	2.141915	15.05238	52.75223	-0.00045	0.011726	15.05238	0	0	0.011726	-5.17E-09	2.141915	0	0.858512	2.58
10	8	0.576204	68.87517	17.1931	-0.00065	17.1931	2.139514	17.1931	68.87517	-0.00045	0.013393	17.1931	0	0	0.013393	-5.17E-09	2.139514	0	0.856589	2.58
11	9	0.575558	87.13763	19.33141	-0.00065	19.33141	2.137117	19.33141	87.13763	-0.00045	0.015059	19.33141	0	0	0.015059	-5.16E-09	2.137117	0	0.85467	2.58
12	10	0.574914	107.5372	21.46733	-0.00064	21.46733	2.134723	21.46733	107.5372	-0.00045	0.016723	21.46733	0	0	0.016723	-5.16E-09	2.134723	0	0.852756	2.58
13	11	0.574269	130.0715	23.60086	-0.00064	23.60086	2.132331	23.60086	130.0715	-0.00045	0.018385	23.60086	0	0	0.018385	-5.15E-09	2.132331	0	0.850846	2.58
14	12	0.573626	154.7382	25.73199	-0.00064	25.73199	2.129941	25.73199	154.7382	-0.00045	0.020045	25.73199	0	0	0.020045	-5.14E-09	2.129941	0	0.84894	2.58
15	13	0.572983	181.5347	27.86074	-0.00064	27.86074	2.127555	27.86074	181.5347	-0.00045	0.021704	27.86074	0	0	0.021704	-5.14E-09	2.127555	0	0.847039	2.58
16	14	0.572341	210.4589	29.9871	-0.00064	29.9871	2.125171	29.9871	210.4589	-0.00045	0.02336	29.9871	0	0	0.02336	-5.13E-09	2.125171	0	0.845142	2.58
17	15	0.5717	241.5082	32.11108	-0.00064	32.11108	2.122789	32.11108	241.5082	-0.00045	0.025015	32.11108	0	0	0.025015	-5.13E-09	2.122789	0	0.843249	2.58
18	16	0.571059	274.6803	34.23268	-0.00064	34.23268	2.120411	34.23268	274.6803	-0.00044	0.026667	34.23268	0	0	0.026667	-5.12E-09	2.120411	0	0.84136	2.58
19	17	0.570419	309.9728	36.35191	-0.00064	36.35191	2.118035	36.35191	309.9728	-0.00044	0.028318	36.35191	0	0	0.028318	-5.12E-09	2.118035	0	0.839476	2.58
20	18	0.56978	347.3833	38.46875	-0.00064	38.46875	2.115662	38.46875	347.3833	-0.00044	0.029967	38.46875	0	0	0.029967	-5.11E-09	2.115662	0	0.837596	2.58
21	19	0.569142	386.9095	40.58323	-0.00064	40.58323	2.113291	40.58323	386.9095	-0.00044	0.031614	40.58323	0	0	0.031614	-5.10E-09	2.113291	0	0.83572	2.58
22	20	0.568504	428.549	42.69534	-0.00064	42.69534	2.110923	42.69534	428.549	-0.00044	0.03326	42.69534	0	0	0.03326	-5.10E-09	2.110923	0	0.833848	2.58
23	21	0.567867	472.2995	44.80508	-0.00064	44.80508	2.108558	44.80508	472.2995	-0.00044	0.034903	44.80508	0	0	0.034903	-5.09E-09	2.108558	0	0.83198	2.58
24	22	0.567231	518.1584	46.91245	-0.00064	46.91245	2.106195	46.91245	518.1584	-0.00044	0.036545	46.91245	0	0	0.036545	-5.09E-09	2.106195	0	0.830117	2.58
25	23	0.566595	566.1236	49.01747	-0.00064	49.01747	2.103835	49.01747	566.1236	-0.00044	0.038185	49.01747	0	0	0.038185	-5.08E-09	2.103835	0	0.828257	2.58
26	24	0.56596	616.1926	51.12013	-0.00063	51.12013	2.101478	51.12013	616.1926	-0.00044	0.039823	51.12013	0	0	0.039823	-5.08E-09	2.101478	0	0.826402	2.58
27	25	0.565326	668.3631	53.22043	-0.00063	53.22043	2.099123	53.22043	668.3631	-0.00044	0.041459	53.22043	0	0	0.041459	-5.07E-09	2.099123	0	0.824551	2.58
28	26	0.564693	722.6327	55.31837	-0.00063	55.31837	2.096771	55.31837	722.6327	-0.00044	0.043093	55.31837	0	0	0.043093	-5.06E-09	2.096771	0	0.822705	2.58
29	27	0.56406	778.9991	57.41397	-0.00063	57.41397	2.094422	57.41397	778.9991	-0.00044	0.044725	57.41397	0	0	0.044725	-5.06E-09	2.094422	0	0.820862	2.58

図 3.1: 図 2.2 のシミュレーション結果の csv ファイルの一部

```
$ python characteristic.py DCmotor_res.csv inertia1 signalVoltage1
characteristicTable.pdf created
```

図 3.2: 実行コマンド例

る、電源部品のオブジェクト名が必要である。

第 2 引数、第 3 引数で慣性部品と電源部品のオブジェクト名が必要な理由は、モータ特性表を作成するために使用するデータを持つ部品だからである。

図 3.1 に示した csv ファイルのファイル名が、「DCmotor_res.csv」だった場合の実行コマンドを、図 3.2 に示す。

モータ特性表が作成できた場合は、図 3.2 に示すように、画面上に「characteristicTable.pdf created」と青色で表示する。

作成できなかった場合については、3.3 節で述べる。

図 3.2 のコマンドでツールを実行し、図 3.1 から作成したモータ特性表を、図 3.3 に示す。

モータ特性表

電圧 V	2.4
始動電流 mA	431.9945756634572
停動トルク mNm	0.9244683919197983
最大効率 %	99.89790026553919
定格トルク mNm	0.0007704063898538723
定格回転数 rpm	10700.575772223889
定格電流 mA	0.3600029859130245
定格出力 W	0.0008632878742652782
最大回転数 rpm	10700.575772223889

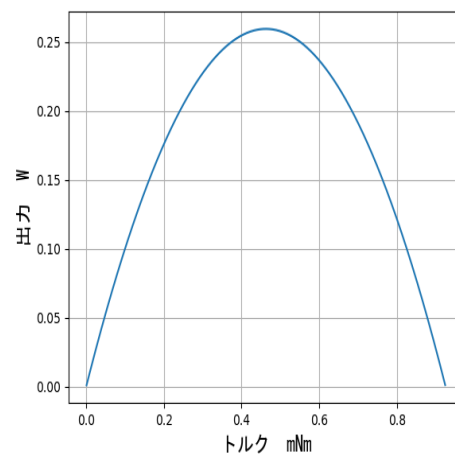
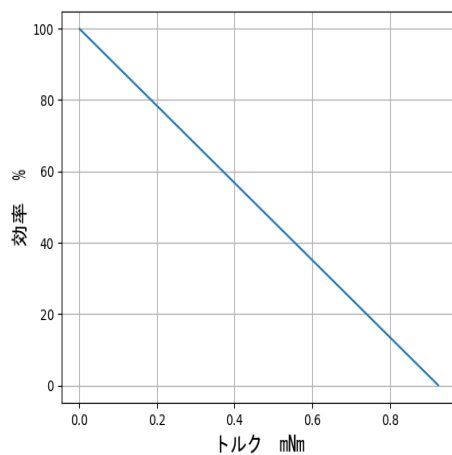
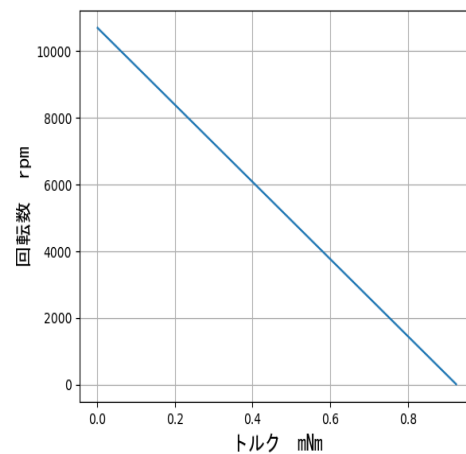
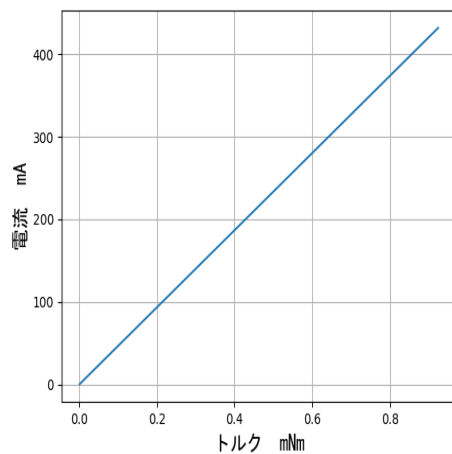


図 3.3: 図 3.1 の csv ファイルから作成したモータ特性表


```
$ python characteristic.py DCmoto_res.csv inertia1 signalVoltage1  
ERROR : 指定したファイル名が間違っています
```

図 3.4: 第 1 引数に誤りがあった場合のエラー文の例

```
$ python characteristic.py DCmotor_res.csv inertia1 signalvoltage1  
ERROR : 指定したオブジェクト名が間違っています
```

図 3.5: 第 2、第 3 引数に誤りがあった場合のエラー文の例

3.3 エラー表示

3.2 節で、試作したツールを実行するためのコマンドは、「python characteristic.py 第 1 引数 第 2 引数 第 3 引数」であると述べた。このコマンドを実行した際に、モータ特性表が作成できなかった場合、原因が 2 つ考えられ、それぞれの原因に即したエラー文を、画面上に表示する。1 つ目の原因は、第 1 引数で指定した csv ファイルのファイル名が間違っている場合が考えられ、エラー文として「ERROR : 指定したファイル名が間違っています」を、赤色で画面上に表示する。

例を、図 3.4 に示す。

2 つ目の原因は、第 2 引数と第 3 引数のどちらか一方、もしくはその両方に間違いがあった場合が考えられ、エラー文として「ERROR : 指定したオブジェクト名が違います」を、赤色で画面上に表示する。

例を、図 3.5 に示す。

第 4 章

実装

本章では、本研究で試作したモータ特性表自動生成ツールの実装について説明する。
特性表生成機能の処理の流れを以下に示す。

1. OpenModelica から出力された csv ファイルを読み込む
2. 特性表の各要素を算出するために必要なデータを、csv ファイルから取得する
3. 特性表の各要素を算出する
4. 特性表を生成する

以下、各処理について具体的に説明する。

4.1 csv ファイルの読み込み

Python で実装するため、Python の標準ライブラリの csv モジュールをインポートし、csv ファイルを読み込む。

4.2 特性表の各要素を算出するために必要なデータを取得

4.1 節で読み込んだ csv ファイルから、以下のデータを取得する。

- 時間

- 電流
- 電圧
- トルク
- 角速度

図 3.1 に示したように、OpenModelica から出力された csv ファイルの 1 行目には、各部品の変数名が記載されている。これを利用して、次の処理で必要なデータを取得する。

1.

取得したいデータを持つ変数名を探し、その変数名がある場所の添字を取得する。各データごとに用意した配列に、同じ添字の位置にある値を繰り返し処理で格納する方法でそれぞれの値を取得する。

4.3 特性表の各要素を算出

4.2 節で取得したデータを用いて、3.1 節で挙げた各要素の値を求める。

4.3.1 電圧

今回試作したツールでは、電源部品の電圧値を電圧とする。

4.3.2 始動電流

モータが起動した後は、逆起電力を発生させるため、モータ・コイル部分にかかる電圧が下がり、電流値も下がる。

したがって、今回試作したツールでは、電流値の中で一番大きい値を始動電流とする。

4.3.3 停動トルク

したがって、今回試作したツールでは、トルク値の中で一番大きい値を停動トルクとする。

4.3.4 最大効率

効率は以下の式で算出する。

$$\text{効率} = \frac{\text{出力}}{\text{入力}} * 100$$

$$\text{出力} = \text{角速度} * \text{トルク}$$

$$\text{入力} = \text{電圧} * \text{電力}$$

今回試作したツールでは、効率値の中で一番大きい値を最大効率とする。

4.3.5 定格トルク

したがって、トルク値の配列の中で、最大効率のある効率値の配列の添字と同じ位置にある値が定格トルクとなる。

4.3.6 定格回転数

回転数は以下の式で算出できる。

$$\text{回転数} = \frac{30 * \text{角速度}}{\pi}$$

したがって、一度繰り返し処理で角速度を回転数に変換し、回転数値の配列の中で、最大効率のある効率値の配列の添字と同じ位置にある値が定格回転数となる。

4.3.7 定格電流

したがって、電流値の配列の中で、定格トルクのあるトルク値の配列の添字と同じ位置にある値が定格電流となる。

4.3.8 定格出力

定格出力は以下の式で算出できる。

$$\text{定格出力} = \text{定格回転数} * \text{定格トルク} * \frac{2\pi}{60}$$

4.3.9 最大回転数

今回試作したツールでは、回転数地の配列の中から一番大きい値を

4.4 特性表を生成

4.3 章で求めた各値と、3.1 章で挙げた各要素を、電圧から順に”,”で区切りつつ特性表生成配列に格納する。そして特性表生成配列を用いて csv ファイルを作成する。

第 5 章

適用例

本章では、本研究で作成した

5.1 モータ単体のモデル

5.2 モータ単体の **Modelica** モデルをサブシステムとするモデル

第 6 章

考察

本論文では、モータ特性表自動生成ツールを試作した。

6.1 評価

6.1.1 評価方法

6.1.2 結果

本論文で試作したモータ特性表自動生成ツールは、

6.2 関連研究

関連研究について述べる。

6.3 ツールの問題点

以下に、今回作成したモータ特性表自動生成ツールの問題点を示す。

- 対応するモータのモデルは 1 種類しかない
モータは～種類に分けることができ、今回は 1 つにしか対応していない。対応できる数を増やす必要がある。

第 7 章

おわりに

以下に、今後の課題を示す。

謝辞

参考文献

- [1] 日本電産株式会社. 身の回りのモータ. <https://www.nidec.com/jp/technology/scenes/>. Accessed: 2020-1-22.
- [2] Tech Web motor. ブラシ付き dc モータの特性. https://micro.rohm.com/jp/techweb_motor/knowledge/basics/basics-03/209. Accessed: 2020-1-23.
- [3] Peter Fritzson. Modelica によるシステムシミュレーション入門. TechShare 社, 2015.
- [4] Peter Fritzson, Peter Aronsson, Adrian Pop, Hakan Lundvall, Kaj Nystrom, Levon Saldamli, David Broman, and Anders Sandholm. Openmodelica-a free open-source environment for system modeling, simulation, and teaching. In *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 1588–1595. IEEE, 2006.