

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	OpenModelica	2
2.1.1	出力	2
2.1.2	グラフィカルモデリング	2
2.2	Modelica 言語	3
2.2.1	Modelica 標準ライブラリ (MSL)	3
2.3	モータ特性表	3
2.3.1	特性表	3
2.3.2	特性グラフ	3
2.4	ブラシ付き DC モータ	3
2.5	対応するモデル	3
2.5.1	モータ単体の Modelica モデル	4
2.5.2	モータ単体の Modelica モデルをサブシステムとするモデル	4
2.6	Python	5
3	機能	8
3.1	ツールの実行方法	8
3.1.1	ファイル操作	8
3.1.2	実行コマンド	8
3.2	ツールの機能	8
3.2.1	モータ特性表生成	8
4	実装	13
4.1	csv ファイルの読み込み	13
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	13
4.3	特性表の各要素を算出	14

4.3.1	電圧	15
4.3.2	始動電流	15
4.3.3	停動トルク	15
4.3.4	最大効率	15
4.3.5	定格トルク	15
4.3.6	定格回転数	15
4.3.7	定格電流	15
4.3.8	定格出力	15
4.4	特性表を生成	15
5	適用例	16
5.1	モータ単体のモデル	16
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	16
6	考察	17
6.1	評価	17
6.1.1	評価方法	17
6.1.2	結果	17
6.2	関連研究	17
6.3	ツールの問題点	17
7	おわりに	18
	謝辞	19
	参考文献	20

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 OpenModelica

Modelica 言語に対応した OSS である [2]。

2.1.1 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
- plt ファイル
- csv ファイル

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。

2.1.2 グラフィカルモデリング

サブセクションで書くなら何か他にも書く。

2.2 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。

2.2.1 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。

2.3 モータ特性表

モータを選ぶ際に、参考にする資料である。

今回作成する特性表の各要素の必要性はここで書く？

2.3.1 特性表

2.3.2 特性グラフ

2.4 ブラシ付き DC モータ

ブラシ付き DC モータだけかけばいいのか？モータ全体の話も必要か？

2.5 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

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

なお、今回はモータの中でもブラシ付き DC モータに対応する。

以降、上記のモデルについて具体的に説明する。

2.5.1 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。

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 [3] を Modelica 言語で表す際に、使用する部品 [4] 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。

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

モータ単体の Modelica モデルをサブシステム [4] とするモデルとは、2.5.1 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 Modelica コードを図 2.5 に、それぞれ示す。

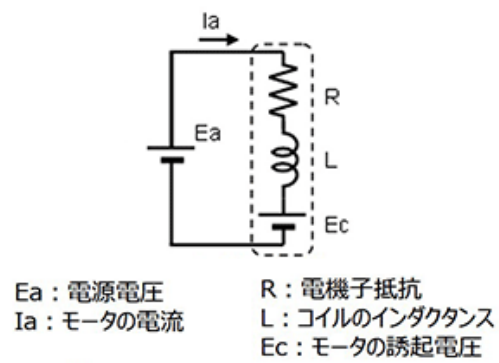


図 2.1: ブラシ付き DC モータの等価回路

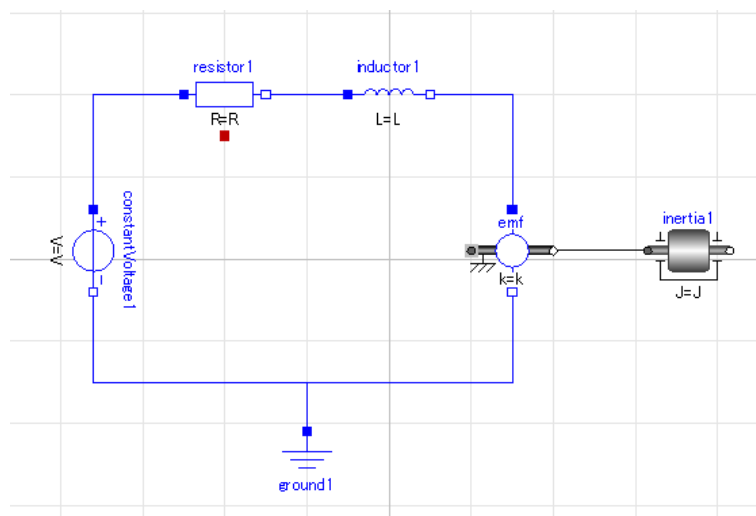


図 2.2: モータ単体の Modelica モデルの例

2.6 Python

Python の説明——今回は——を実装するために Python を使用した。


```

1 model DCmotor
2 Modelica.Electrical.Analog.Basic.Resistor resistor1(T = 283.15) annotation(
3   Placement(visible = true, transformation(origin = {-40, 54}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
4 Modelica.Electrical.Analog.Basic.Inductor inductor1 annotation(
5   Placement(visible = true, transformation(origin = {-12, 54}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
6 Modelica.Electrical.Analog.Basic.Ground ground1 annotation(
7   Placement(visible = true, transformation(origin = {-34, -54}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
8 Modelica.Mechanics.Rotational.Components.Inertia inertial(a(start = 0), phi(start = 0), w(start = 0)) annotation(
9   Placement(visible = true, transformation(origin = {25, 18}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
10 Modelica.Electrical.Analog.Basic.EMF emf(useSupport = false) annotation(
11   Placement(visible = true, transformation(origin = {-2, 18}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
12 Modelica.Electrical.Analog.Sources.ConstantVoltage constantVoltage1 annotation(
13   Placement(visible = true, transformation(origin = {-64, 8}, extent = {{-10, -10}, {10, 10}}, rotation = -90)));
14 equation
15 connect(constantVoltage1.n, ground1.p) annotation(
16   Line(points = {{-64, -2}, {-64, -2}, {-64, -30}, {-34, -30}, {-34, -44}, {-34, -44}, {-34, -44}}, color = {0, 0, 255}));
17 connect(constantVoltage1.n, emf.n) annotation(
18   Line(points = {{-64, -2}, {-64, -2}, {-64, -30}, {-2, -30}, {-2, 8}, {-2, 8}}, color = {0, 0, 255}));
19 connect(constantVoltage1.p, resistor1.p) annotation(
20   Line(points = {{-64, 18}, {-64, 18}, {-64, 54}, {-50, 54}, {-50, 54}}, color = {0, 0, 255}));
21 connect(resistor1.n, inductor1.p) annotation(
22   Line(points = {{-30, 54}, {-22, 54}}, color = {0, 0, 255}));
23 connect(inductor1.n, emf.p) annotation(
24   Line(points = {{-2, 54}, {-2, 28}}, color = {0, 0, 255}));
25 connect(emf.flange, inertial.flange_a) annotation(
26   Line(points = {{8, 18}, {15, 18}}));
27 annotation(
28   uses(Modelica(version = "3.2.3"));end DCmotor;

```

図 2.3: 図 2.2 の Modelica コード

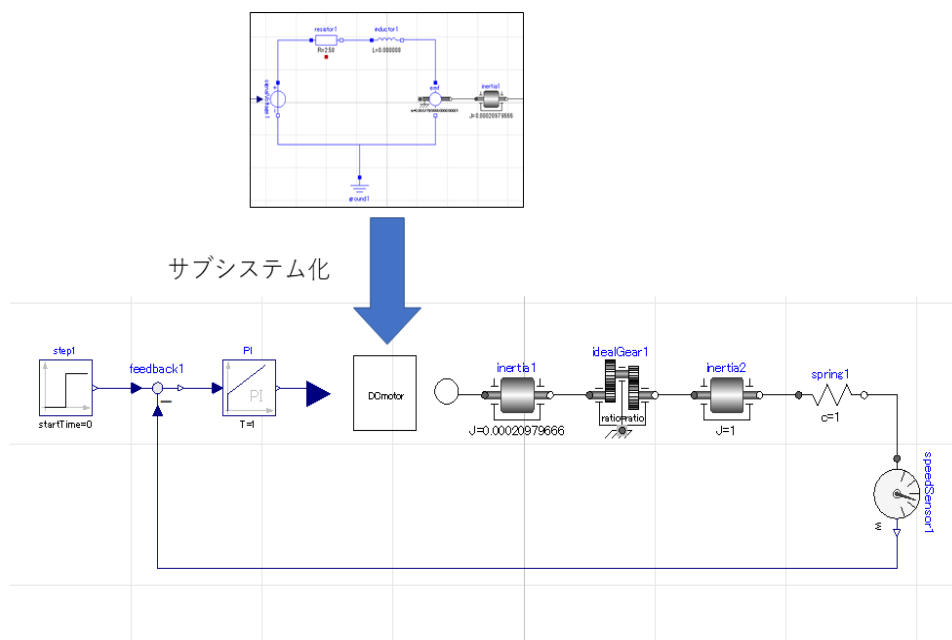


図 2.4: DC モーターサーボのモデル

```

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_iner 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

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時に 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 ツールの実行方法

3.1.1 ファイル操作

3.1.2 実行コマンド

3.2 ツールの機能

3.2.1 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、次の 9 個の要素を持つモータ特性表を生成する。

- 電圧 V
- 始動電流 mA

	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(inertia	der(inertia	der(emf.p	emf.phi	emf.tau	emf.v	emf.w	ground1.p	ground1.p	inductor1.i	inductor1.i	inertial1.a	inertial1.fl	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

- 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

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、図 3.1 から作成できるモータ特性表を図 3.2 に、それぞれ示す。

OpenModelica から出力される

電圧

シミュレーション時に印加した値を取る。

始動電流

始動電流とは、モータの起動時に流れる大きな電流のことである。モータが起動した後はモータ自体が発電機にもなり、逆起電力を発生するため、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、電流値の配列の中で一番大きい値を始動電流とする。

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となる。したがって、トルク値の配列の中で一番大きい値を停動トルクとする。

最大効率

効率は以下の式で算出することができる。

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

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

$$\text{入力} = \text{電圧} * \text{電力}$$

各配列を上記の式に当てはめ、繰り返し処理で効率値の配列を作成する。最大効率は効率値の配列の中で一番大きい値とする。

定格トルク

最大効率時のトルクを定格トルクという。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。

定格回転数

最大効率時の回転数を定格回転数という。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。

定格電流

定格電流とは、モータに定格トルクがかかっているときの電流値である。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。

定格出力

定格出力とは、定格動作点における出力の値である。定格出力は以下の式で算出できる。

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

モータ特性表

電圧 V	18
始動電流 mA	13429.71488396701
停動トルク mNm	218.9043526086622
最大効率 %	99.1239351002758
定格トルク mNm	1.928718019325694
定格回転数 rpm	10452.345530321025
定格電流 mA	118.3262588543371
定格出力 W	2.1111112203940428
最大回転数 rpm	10452.345530321025

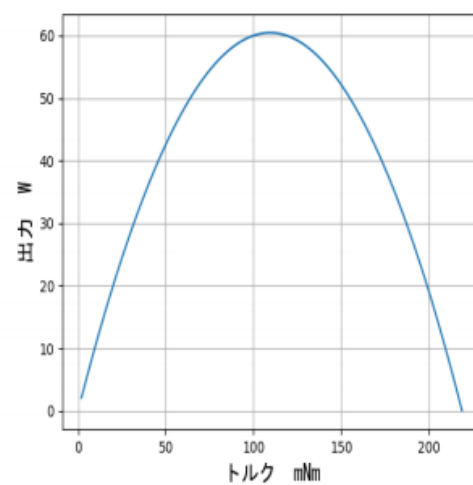
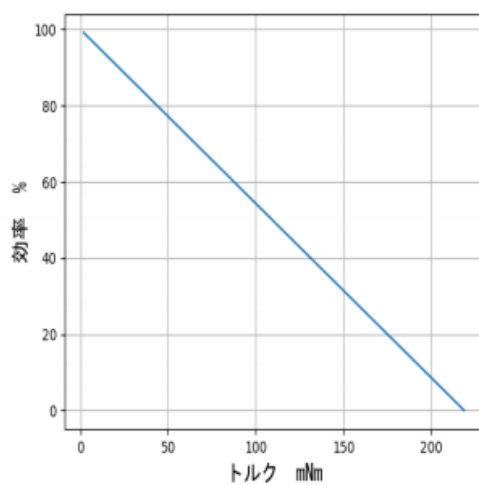
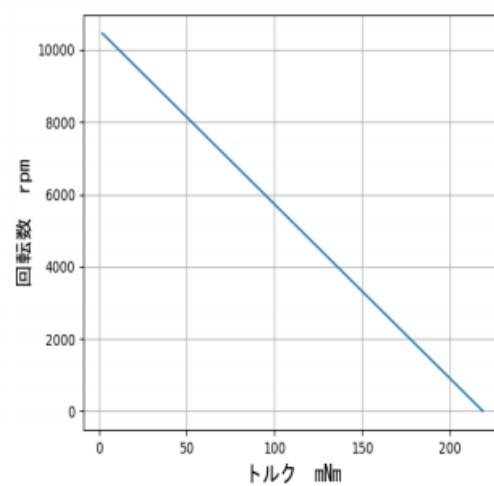
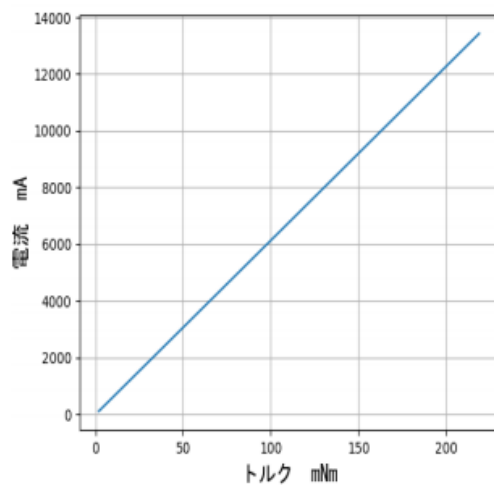


図 3.2: 図 3.1 の csv ファイルから作成したモータ特性表

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、??節で挙げた各要素の値を求める。

4.3.1 電圧

4.3.2 始動電流

4.3.3 停動トルク

4.3.4 最大効率

4.3.5 定格トルク

4.3.6 定格回転数

4.3.7 定格電流

4.3.8 定格出力

4.4 特性表を生成

4.3 章で求めた各値と、??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] 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.
- [3] 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.
- [4] Peter Fritzson. Modelica によるシステムシミュレーション入門. TechShare 社, 2015.