令和 元年度 卒業論文

OpenModelica のシミュレーション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 2 月

目次

第 1 章

はじめに

近年ソフトウェア開発の効率化が求められており、設計段階や製品を試作する前に、製品の機能や性能を検証したいというニーズが高まっている [1]。このニーズに応えるために、モデルベースシステム開発手法がある [1]。モデルベースシステム開発手法とは、製品の設計を元にシミュレーションツールを用いて、シミュレーションを行いながら、設計品質の向上を図る開発手法である [2]。設計の品質が向上することで、不具合による後戻りを削減できるため、生産性の向上が期待できる [2]。

モデルベースシステム開発手法は、 組込みシステムの開発において特に重要である [3]。製品開発を行う際に、モデルベースシステム開発手法を適用することで、実際に製品を試作することなく、製品の様々な性能を事前に確認できるため、開発コストを削減できる [1]。また、実際に試作品を作る回数を減らすことができるため、製品開発の納期の短縮が期待できる。

モデルベースシステム開発手法を用いた組込みシステムの開発に、OpenModelica[4] が使われる。OpenModelica は、Modelica[**?**] コードのモデリング、シミュレーション、デバッグのための機能などを持つオープンソースプラットフォームである。OpenModelica が出力するシミュレーションの結果は、グラフや数値であり、csv ファイルに出力できる。しかし、この出力を用いて、性能を決定付ける特定の値を確認するためには、手間と時間がかかる。そこで、本研究では、性能を決定付ける特定の値を確認するためにかかる時間の削減を目的として、OpenModelica のシミュレーション結果を用いた特性表の自動生成ツールの試作を行う。特性表とは、製品の性能をまとめた表であり、特性表を用いることで、特定の値を容易に確認できる。なお、本研究では、シミュレーションの対象として、ブラシ付き DC モータ [参考文献] を対象とする。

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構成および手順について説明する。

第 4 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 5 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 6 章では、、本論文のまとめとの課題を述べる。

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 Modelica 言語

Modelica 言語とは、微分代数方程式を用いた、複合領域のマルチドメイン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 [?]。その言語仕様は、非営利団体の Modelica Association が策定している。Modelica Association では、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 Modelica 標準ライブラリ (Modelica Standard Library : MSL) をリリースしている [?]。

2.2 OpenModelica

OpenModelica とは、Open Source Modelica Consortium (OSMC) が開発している Modelica コードのモデリング、シミュレーション、デバッグのための機能などを持つオープンソースプラットフォームである [?]。OpenModelica では、グラフィカルにモデリングすることが可能であり、専用のグラフィカルモデルエディタが用意されている。グラフィカルモデルエディタの外観を、図 2.1 に示す。また、グラフィカルモデルエディタ上にモデルを配置する際、オブジェクト名の確認をポップアップウィンドウで行う。ポップアップウィンドウの例を、図 2.2 に示す。OpenModelica は、シミュレーション結果をグラフとして画面上に描画できる。また、シミュレーション結果は、以下の 3 つのファイル形式で保存できる。

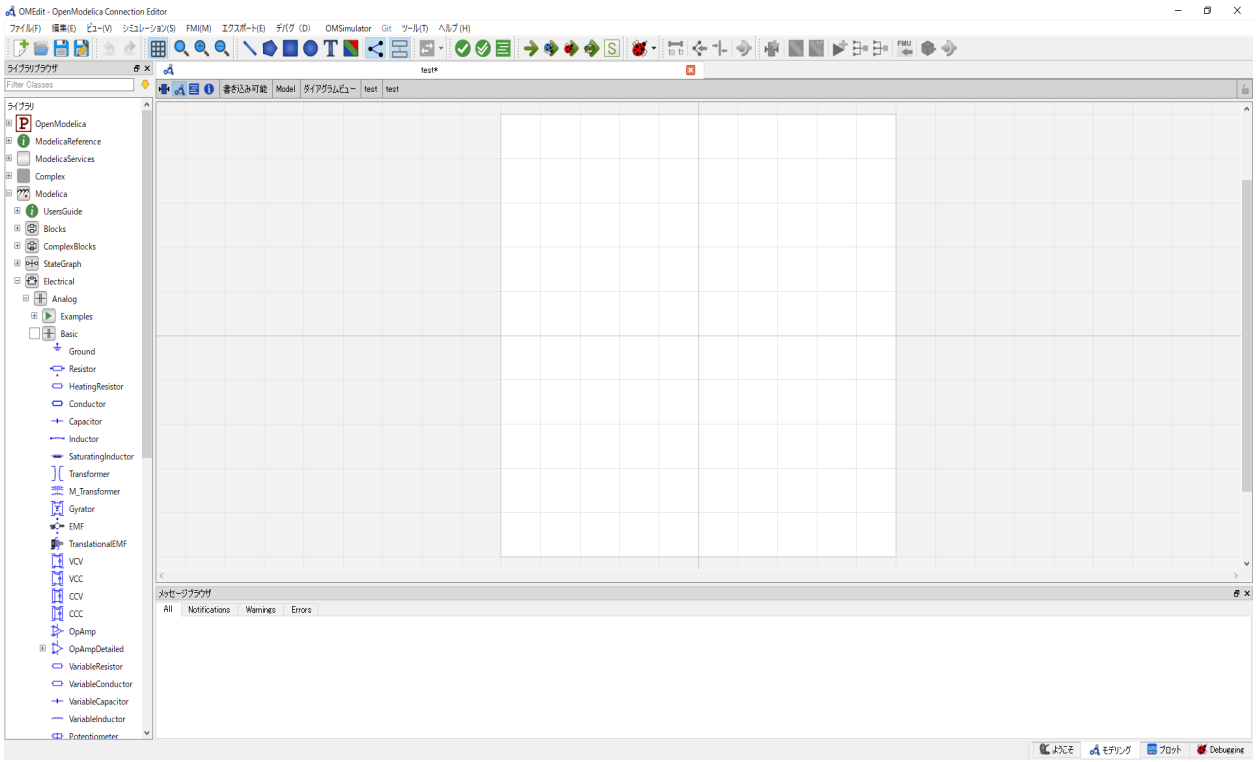

図 2.1: グラフィカルモデルエディタの外観

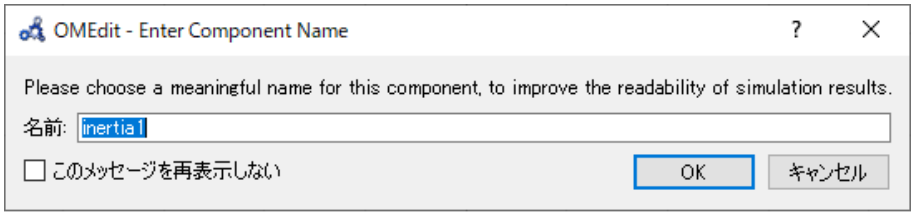

図 2.2: オブジェクト名の確認のポップアップウィンドウの例

- mat ファイル
- plt ファイル
- csv ファイル

今回試作するモータ特性表自動生成ツールでは、csv ファイルにのみ対応する。

OpenModelica から出力される csv ファイルの一部を、図 2.3 に示す。

図 2.3 に示す csv ファイルの 1 列目には、「0.0 秒」から始まる時刻を表すデータが必ず入る。そして 1 行目には、1 列目の「time」を除いて、オブジェクト名を含んだ変数名が必ず入る。また以降には、1 列目の「time」における各変数の値が 2 列目以降にそれぞれ入る。

```
"time","inductor1.i","inertial.phi","inertial.w","der(inductor1.i)","der(inertial.phi)","der(inertia
0,0,0,0,122448.9795918368,0,0,0,0,-0,0,0,0,0,9,9,0,0,0,1.78,0,9,0,-0,0,-0,0,0,0,0,-0,0,0,0,0,0,0,-0,
0.1,5.047915163691204,0.0707464037147575,1.415104150827973,-0.08264678851114857,1.415104150827973,14
0.2,5.039657246826229,0.2829450187532758,2.828477427971375,-0.08251158624533513,2.828477427971375,14
0.3,5.031412839149343,0.6363654143980045,4.239538557176244,-0.08237660493127441,4.239538557176244,14
0.4,5.023181918532681,1.130776271214502,5.648291325710747,-0.08224184440823901,5.648291325710747,14.
0.5,5.014964462926303,1.765947084414186,7.054739507505285,-0.08210730441070999,7.054739507505285,14.
0.6,5.006760450312931,2.541647690661693,8.458886870899837,-0.08197298469436279,8.458886870899837,14.
0.7,4.998569858707249,3.45764826095431,9.860737178760534,-0.08183888845401014,9.860737178760534,14.0(
0.8,4.990392666186916,4.513719624349956,11.26029418317943,-0.08170500376628731,11.26029418317943,13.
0.9,4.982228850797614,5.70963227572398,12.65756164171997,-0.08157134205288172,12.65756164171997,13.9
1,4.974078390690659,7.045157814256005,14.05254329386543,-0.08143789896712425,14.05254329386543,13.93
1.1,4.965941263932,8.520066946679792,15.44524289371061,-0.08130467414554903,15.44524289371061,13.915
1.2,4.957817448793185,10.1341325290293,16.83566416016098,-0.08117166726641792,16.83566416016098,13.8
1.3,4.94970692345335,11.88712645122738,18.22381082793964,-0.08103887803442675,18.22381082793964,13.8
1.4,4.941609666177463,13.77882150055004,19.60968661707778,-0.0809063060255005,19.60968661707778,13.8
1.5,4.933525655273357,15.80899091233752,20.99329524027073,-0.0807739509069499,20.99329524027073,13.8
1.6,4.925454869029084,17.97740771513312,22.37464041359953,-0.0806418123056796,22.37464041359953,13.8
1.7,4.917397285872307,20.28384639697651,23.75372582924972,-0.08050988983726544,23.75372582924972,13.
1.8,4.909352884122416,22.72808031399924,25.13055519793894,-0.08037818322415165,25.13055519793894,13.
1.9,4.901321642311751,25.30988504848918,26.50513219393718,-0.08024669201469649,26.50513219393718,13.
2,4.893303538801044,28.02903438879071,27.8774605208857,-0.08011541590755371,27.8774605208857,13.712(
2.1,4.885298552177078,30.88530448629863,29.24754384373682,-0.07998435457041164,29.24754384373682,13.
2.2,4.877306660941829,33.87847060588226,30.61538584195743,-0.07985350765434308,30.61538584195743,13.
2.3,4.869327843661874,37.00830868772714,31.98099018395787,-0.07972287479153942,31.98099018395787,13.
2.4,4.861362078936959,40.27459501875122,33.34436053247161,-0.07959245568140967,33.34436053247161,13.
2.5,4.853409345433533,43.67710658319652,34.70550053881518,-0.07946224987080158,34.70550053881518,13.
```

図 2.3: シミュレーション結果の csv ファイルの一部

OpenModelica から出力される csv ファイルのファイル名は、「(シミュレーションしたモデルの名前)_res.csv」で生成される。例えば、シミュレーションしたモデルの名前が「hoge」だった場合、csv ファイルのファイル名は「hoge_res.csv」となる。

2.3 ブラシ付き DC モータ

DC モータとは、磁場の中にあるコイルに電流を流す事で発生するローレンツ力を回転方向に利用することで回すモータである [**?**]。また、ブラシ付き DC モータとは、モータ内部に「ブラシ」と呼ばれる電極と、「コミュテータ」と呼ばれる整流子を設け、その二つを接触させ機械的に電流の切り替えを行い、回す DC モータである。ブラシ付き DC モータは、シンプルな構造で、制御が容易で汎用性が高く、模型用モータや自動車補機用モータなど、世界で一番多く使われているモータである [**?**]。

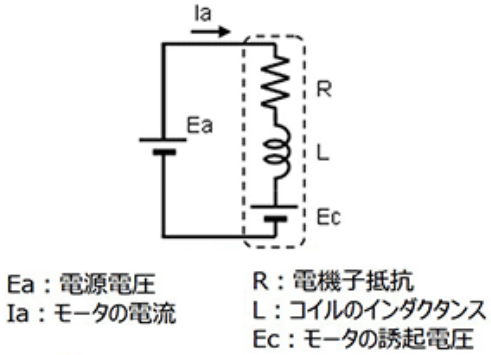

図 2.4: ブラシ付き DC モータの等価回路

2.4 対象とするモデル

試作するモータ特性表自動生成ツールでは、以下に示す 2 つの Modelica モデルのシミュレーション結果を対象とする。

- ブラシ付き DC モータの Modelica モデル
- ブラシ付き DC モータの Modelica モデルをサブシステムとするモデル

以下にて、それぞれについて説明する。

2.4.1 ブラシ付き DC モータの Modelica モデル

ブラシ付き DC モータの Modelica モデルとは、ブラシ付き DC モータの等価回路 [**?**] を Modelica 言語で表したモデルのことである。ブラシ付き DC モータの等価回路を、図 2.4 に示す。また、ブラシ付き DC モータの Modelica モデルの例を図 2.5 に示す。ブラシ付き DC モータの等価回路を Modelica 言語で表すためには、電源部品、抵抗部品、インダクタ部品、起電力部品、慣性部品、接地部品が必要である。

また、電源部品、抵抗部品、インダクタ部品、起電力部品、慣性部品には、それぞれ以下のパラメータを設定しなければならない。各部品で使用する MSL を表 2.1 に示す。

- 電源部品 ・・・ 電圧値 (V)
- 抵抗部品 ・・・ 抵抗値 (Ω)

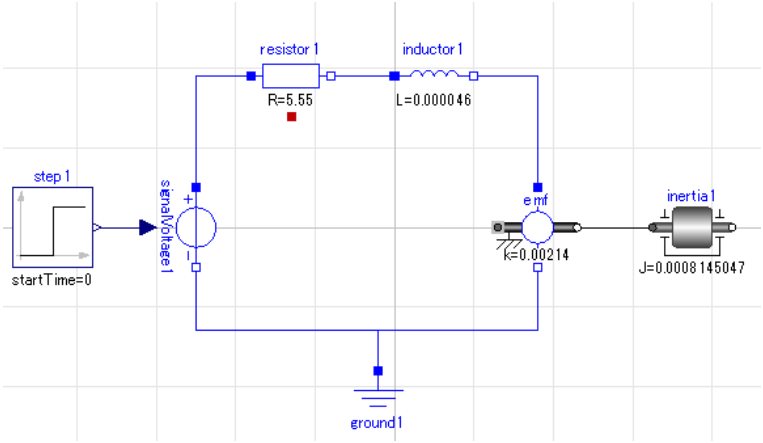

図 2.5: ブラシ付き DC モータの Modelica モデルの例

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

- インダクタ部品 ・・・ インダクタンス値 (H)
- 起電力部品 ・・・ トルク定数 ($N \cdot m/A$)
- 慣性部品 ・・・ 慣性モーメント ($kg \cdot m^2$)

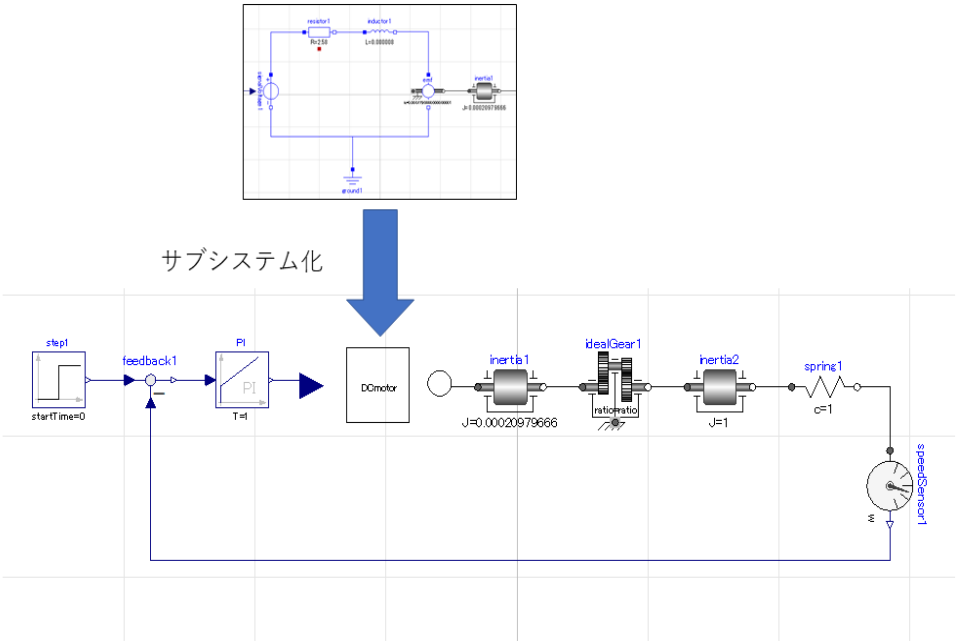

図 2.6: DC サーボモータのモデル

2.4.2 ブラシ付き DC モータの Modelica モデルをサブシステムとするモデル

ブラシ付き DC モータの Modelica モデルをサブシステムとするモデルとは、2.4.1 節で説明したブラシ付き DC モータの 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、ブラシ付き DC モータのサブシステムを用いた DC サーボモータのモデルを、図 2.6 に示す。

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 [**?**]。例えば、図 2.7 に示すモータ特性表がある。一般的に決まった形式はなく、企業によって掲載する要素は異なるため、10 社のブラシ付き DC モータのモータ特性表 [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] に含まれる要素を集計した。集計結果を表 2.2 に示す。表 2.2 で示した要素の中で、対象とするモデルの性質により、「無負荷電流」、「無負荷回転数」、「起動トルク」を除く 9 個の要素と 4 つのグラフを、今回自動生成

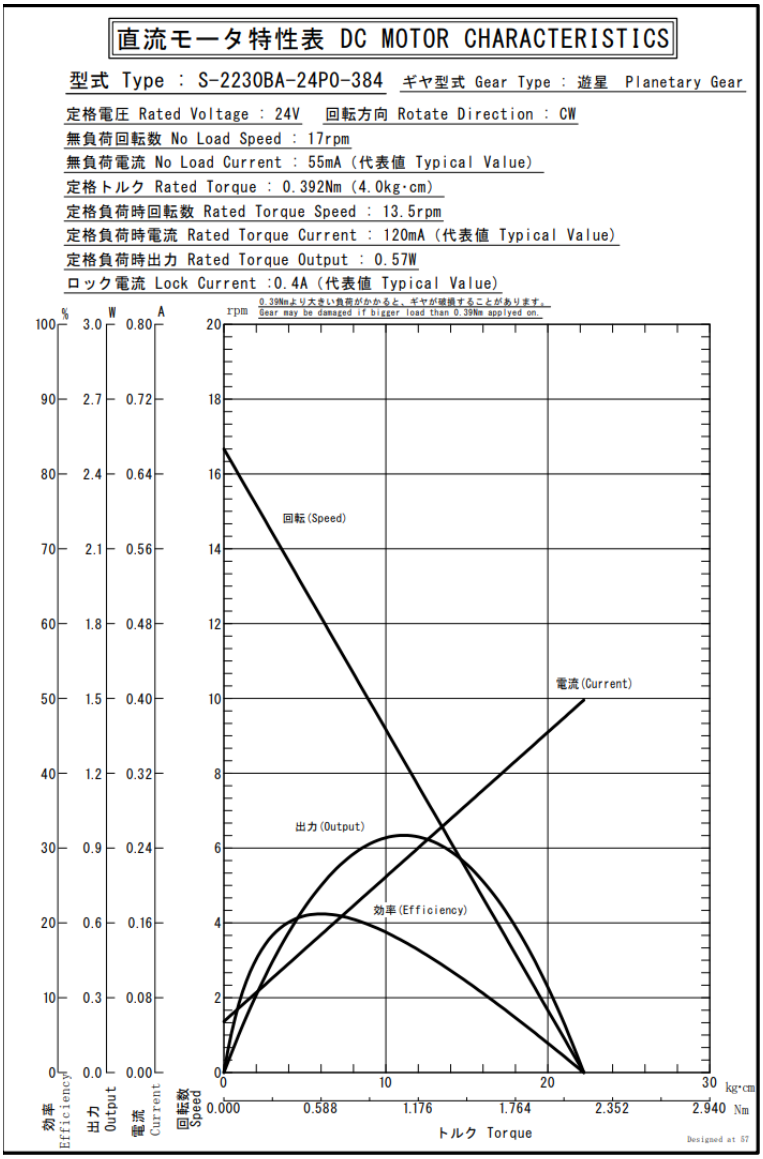

図 2.7: モータ特性表の例

するモータ特性表の要素とした。

以下に、自動生成するモータ特性表の構成を示す。

- モータ特性表
  - 特性表
  - 特性グラフ

以下にて、本研究で対象とするモータ特性表の、特性表と特性グラフの内容について述べる。

表 2.2: モータ特性表の要素の集計結果

要素名	出現回数	要素名	出現回数
定格トルク	10	定格出力	5
定格電流	9	「負荷トルク × 効率」グラフ	5
定格回転数	9	「負荷トルク × 出力」グラフ	5
無負荷電流	8	停動トルク	4
無負荷回転数	8	始動電流	3
「負荷トルク × 電流」グラフ	7	最大回転数	2
「負荷トルク × 回転数」グラフ	7	起動トルク	2
定格電圧	5	最大効率	1

2.5.1 特性表

特性表とは、以下の 9 つの要素で構成する。

定格電圧

定格電圧とは、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 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 である。

停動トルク

停動トルクとは、モータが停止する負荷トルクの値を表す。単位は、(mN · m) 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［%］で表したものである。最大効率とは、効率の最大値を表す。

単位は、% 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、(mN · m) 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 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 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 である。

最大回転数

最大回転数とは、回転数値の中で最大値を表す。

単位は、rpm である。

2.5.2 特性グラフ

特性グラフは、以下 4 つのグラフで構成する。

「負荷トルク × 電流」グラフ

「負荷トルク × 電流」グラフとは、横軸が「負荷トルク (mN · m)」、縦軸が「電流 (mA)」のグラフである。

このグラフでは、トルクに対する電流の変化量を表している。

「負荷トルク × 回転数」グラフ

「負荷トルク × 回転数」グラフとは、横軸が「負荷トルク (mN · m)」、縦軸が「回転数 (rpm)」のグラフである。

このグラフでは、トルクに対する回転数の変化量を表している。

「負荷トルク × 効率」グラフ

「負荷トルク × 効率」グラフとは、横軸が「負荷トルク (mN · m)」、縦軸が「効率 (%)」のグラフである。

このグラフでは、トルクに対する効率の変化量を表している。

「負荷トルク × 出力」グラフ

「負荷トルク × 出力」グラフとは、横軸が「負荷トルク (mN · m)」、縦軸が「出力 (W)」のグラフである。

このグラフでは、トルクに対する出力の変化量を表している。

2.6 Python

Python は、1991 年にオランダ人のグイド・ヴァンロッサムによって開発され、オープンソースで運営されている動的プログラミング言語である [**?**]。Python の用途は様々で、組込み開発や、Web アプリケーション、デスクトップアプリケーション、さらには人工知能開発、ビッグデータ解析などと多岐に渡る [**?**]。Python のプログラミング言語としての主な特徴は、少ないコードで簡潔にプログラムを書けること、専門的なライブラリが豊富にあることが挙げられる。

表 2.3: 使用する Python のライブラリ

ライブラリ	役割
csv	csv ファイルの操作
math	数学関数の使用
matplotlib	グラフ描画
numpy	配列の計算
decimal	指数表記の計算
reportlab	PDF ファイル作成
PIL	画像処理
pdf2image	PDF ファイルの画像化
sys	スクリプトの起動パラメータの取得
os	不要なファイルの削除

今回試作するモータ特性表自動生成ツールの開発言語には、Python を用いる。また、使用する Python のライブラリを、表 2.3 に示す。

第 3 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。

3.1 評価

3.1.1 評価方法

(TODO:下記のコマンドを書き換える！)

既存の手法と、本研究の手法で、作成 (生成) に要した時間の比較検証を行った。その結果を、表 3.1 に示す。

対象とした仕様書は、**??**節で用いた コード**??**である。仕様書を作成する時間を計測した。生成する仕様書としては、以下を基準とした。(TODO:なにをもって完成かを書く！)

1. on ポイント、off ポイント、in ポイント、out ポイントを出力 (記述) する
2. on ポイント、off ポイント、out ポイントには、着目条件式も出力 (記述) する
3. off ポイントには、着目変数も出力 (記述) する
4. 各ポイントには、期待出力と正常系であるかどうかも出力 (記述) する

検証に参加したメンバーは本研究室の大学院生 (TODO:X) 人と学部 4 年生 (TODO:Y) 人であり、普段からソースコードの読み書きを行い、基本的なプログラミングの知識を有している。仕様書の知識を持たない者も含まれるが、今回の検証に必要な文法は、事前に他の仕様書の例を用

いてレクチャーした。また、仕様書生成についても、事前に他の仕様書と仕様書の例を用いてレクチャーした。

人手による検証では、コード**??**を印刷した紙を渡し、仕様書を確認後、仕様書を書き始めてから、仕様書を記述し終えるのに要した時間を計測した。(TODO:なんか) が不正確な場合、間違いを指摘し、被験者が正しい仕様書を記述した時点で時間計測終了とした。また、制限時間を (TODO:Z) 分とし、制限時間を超えた場合、その場で時間計測終了とした。

本研究の手法による検証では、(TODO:計測はじめから終わりの条件と、使った PC の仕様を書く) コマンドライン上での命令操作で、本研究の手法による仕様書生成を行うのに要した時間を計測した。また、実験に用いたコンピュータは、OS:macOS 10.14.5、CPU:2.3GHz Intel Core i5、メモリ:16GB である。

(TODO:純粋な、実行時間を書く) なお、Java の System.nanoTime[**?**] メソッドを用いて、命令操作を省いた純粋な仕様書生成処理に本研究の手法が要した時間を計測した結果、(TODO:A) 秒であった。

人手による作成と比較した結果、平均で (TODO:B) 分程の時間短縮を確認できた。対象にした仕様書には、(TODO:なにか) 独特の文法等は含まれないため、(TODO:なにか) に対する慣れなどの影響は無視できるものと思われる。また、人手による仕様書生成の場合、ヒューマンエラーも見られた。(TODO:ヒューマンエラーがあったら、具体例を書く) 具体的には、off ポイントの記述時に、条件式の解釈を間違え、誤った期待出力を記述してしまった。(例：入力 (17、 20) の期待出力を “遊園地チケットは割引価格とならない。(妻の年齢 < 16)” と記述した。) 仕様書の規模が拡大すると、人手とコンピュータとの処理効率の差に加えて、ヒューマンエラーの有無などにより、仕様書生成に要する時間の差は更に拡大していくと思われる。以上から、本研究の手法は有用性が向上したと考える。

3.1.2 結果

本論文で試作したモータ特性表自動生成ツールは、

3.2 関連研究

関連研究について述べる。

表 3.1: コード**??**の仕様書作成に要した時間の比較

被験者	時間
被験者 A	8m 16s
被験者 B	10m 23s
被験者 C	30m(制限時間超過)
被験者 D	24m 04s

	時間
被験者 (平均)	18m 10s
BWDM	0m 15s

3.3 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 4 章

おわりに

以下に、今後の課題を示す。

謝辞

参考文献

[1] 平野　豊, “Modelica によるモデルベースシステム開発入門-Modelica と FMI の活用による実践的モデルベース開発”, TechShare 株式会社, 2017.

[2] “IPA 独立行政法人 情報処理推進機構:先進的な設計・検証技術の適用事例報告書　2016 年版 No.61 モデルベースによるシステム開発におけるプラントモデルの高精度化について”, https://www.ipa.go.jp/files/000057837.pdf, アクセス日:2020/02/07

[3] “IPA 独立行政法人 情報処理推進機構:組込みシステムの先端的モデルベース開発実態調査報告”, https://www.ipa.go.jp/sec/softwareengineering/reports/20120323.html, アクセス日:2020/02/07

[4] “Welcome to OpenModelica - OpenModelica”, https://openmodelica.org, アクセス日:2020/02/07