

宮崎大学大学院

博士学位論文

電気回路の振る舞いの数理と
そのシンボリック計算について

2017年3月

農学工学総合研究科 物質・情報工学専攻

平田 拓也

指導教員: 穂高 一条教授

Contents

1	はじめに	3
	参考文献	6
2	予備知識	9
2.1	線形代数学	9
2.2	制御理論	9
2.2.1	状態方程式とその一般解	9
2.2.2	状態方程式とその周期解	10
2.2.3	状態方程式から伝達関数を算出する方法	12
2.3	表記上の注意	12
2.3.1	部品の電圧極性と電流の向き	12
2.4	グラフ理論	12
2.4.1	グラフによる電気回路の表現	12
2.4.2	グラフの数式表現	13
2.4.3	閉路行列	14
2.4.4	カットセット行列	16
2.4.5	各行列の関係式	19
2.4.6	状態方程式を計算するアルゴリズム	19
	参考文献	27
3	回路方程式から状態方程式へのシンボリック変換について	29
3.1	はじめに	29
3.2	提案手法	29
3.3	提案手法の適用例	32
3.4	提案手法を使用した回路の解析例—ワイヤレス給電の送電効率を最大にする数式表現	34
3.5	おわりに	36
	Appendix 3.A ワイヤレス給電の受電・送電コイルのモデル	37
	参考文献	39
4	回路の数式モデルの妥当性の検証方法について	41
4.1	はじめに	41
4.2	電気回路の法則と回路方程式	41
4.3	回路方程式における代数方程式の解の存在と一意性	44
4.4	モデリングの矛盾	45
4.5	おわりに	45
	Appendix 4.A 本章で省略された補題, 系, 証明	46
	参考文献	48

5	おわりに	49
A	Wasabi の使い方	53
A.1	Wasabi の初期設定	53
A.2	素子の電圧の極性と電流の向き of 仮定	54
A.3	Wasabi の UI の説明	54
A.4	状態方程式を生成する手順, 回路方程式の解を算出する手順	55
A.4.1	LCR 回路のモデリング	55
A.4.2	磁気結合素子がある回路	58

CHAPTER

1

はじめに

電気回路には、それを構成する素子や電源が含まれる。それらにはパラメータがあり、パラメータに特定の数値を与えた回路の振る舞い解析のことを、数値回路解析 (Numerical Circuit Analysis) と呼ぶ。

数値回路解析を行うコンピュータソフトウェアとして、SPICE [1] がある。1973 年に発表されてから、今日に至るまで様々な改良が施されながら、長く使われているソフトウェアである [2] [3]。

SPICE は、与えられた回路図から、数値計算によって、回路の動作を計算し、結果をグラフによって可視化する。扱うことのできる回路は広く、たとえば、トランジスタやダイオードをはじめとする半導体から構成される非線形な回路や、外部クロックによって駆動されるスイッチング回路のような時変な回路をも扱うことができる。この特徴のために、産業・教育・研究などの様々な分野で、回路設計支援や回路の動作解析の目的で使われている。

一方で、その回路のパラメータを記号として扱うことで、回路の振る舞いを解析を試みるものがあり、それをシンボリック回路解析 (Symbolic Circuit Analysis) と呼ぶ。シンボリック回路解析を実現するソフトウェアとしては、SAPWIN [4], SCAM [5] がある。これらのソフトウェアは、与えられた回路図から、回路の数式モデルとして伝達関数モデルを生成する。回路定数やそれ以外のパラメータを記号として扱った、シンボリックな伝達関数モデルを生成することができる。伝達関数モデルを生成できるために、周波数領域で、時不変回路の振る舞いを解析したり、設計するのに適している。数値回路解析と比較して、回路の振る舞いを支配するパラメータ、あるいはしないパラメータを正確に把握できる。

上であげたソフトウェアを支えるアルゴリズムは、その多くが今日で言うところのグラフ理論に根ざしている。グラフ理論は、その幅広い応用があり、特にコンピュータに関連した物が多く、例えば、最短経路問題 [6] がある。

電気回路においても、それは例外ではない。回路図から KVL 方程式や KCL 方程式を立式するシステムティックな方法や、回路の振る舞いを計算する上で効率的な手法がある。

例えば、AC 解析を行うときには、回路図からフェーザ表示を求めることでそれを行うことができるが、その際に、例えば、SPICE では、Modified Nodal Analysis 法 [7] が使われている。

シンボリックな回路解析を行う場合に、記号を含む回路方程式を解く作業が必要になり、これは一般に、数値のみの方程式を解くことより、計算コストの高い処理であり、それを行うには工夫が必要になる。伝達関数を生成する場合にもその工夫は必要となり、そのために、グラフ理論をベースに、回路図から伝達関数を生成する方法は、多く研究されている [8-17]。

一方で回路の振る舞いを数式表現する方法は、伝達関数だけではなく、状態方程式によるものがある。状態方程式は、時間領域で回路の振る舞いを表すことができ、しかもその振る舞いを表すのに必要最小限な方程式となっている。

回路図からシンボリックな状態方程式を算出する方法として、[18] [19] がある。この方法によれば、“適切な木”を回路図から探し、代数的な演算を行うことに帰着される。

本稿の最初の目的は、回路図から“適切な木”を見つけることの代わりに線形代数の演算を用いることにより、回路図からシンボリックな状態方程式を生成する方法を提案する。これは、本稿の第3章で紹介され、そのアルゴリズムは、Wasabi [20] としてソフトウェアに実装されている。

また、提案した手法を用いて、ワイヤレス給電回路の送電効率を最大化する問題について、シンボリックな回路パラメータを用いて数式表現を行った例を紹介する。一般に、ワイヤレス給電の送電効率や出力は、回路トポロジーや回路上の素子や電源のパラメータ(例えば、キャパシタンスや電源の周波数など)ごとに異なる。従って、いくつかワイヤレス給電を実現する回路を試作し、その数式モデルを用いて、あるパラメータについての送電効率や出力の最大化を行い、その上で、送電効率や出力を比較する必要がある。つまり、そのたびに数式モデルを製作する必要があるが、Wasabiによってその時間を短縮することができる。これはまた、DC/DC コンバータの振る舞いを解析する際にも同じことができる。それは、DC/DC コンバータは、スイッチングによって、回路トポロジーが変化し、従って、それに対応する数式モデルが必要になるからである。

次の表に、コンピュータ上の電気回路の設計支援 (CAD) ツールについて、それぞれのソフトウェアが扱える回路の種類 (解析可能な回路) やユーザーが得る結果 (出力) をもとに、いくつかを記した。

表 1.1: 電気回路の設計支援 (CAD) ツールの比較

	SPICE	SAPWIN	SCAM ²	Wasabi (提案するソフトウェア)
計算方法	数値計算	数値計算/ シンボリック計算	数値計算/ シンボリック計算	シンボリック計算
出力	グラフ	伝達関数モデル	伝達関数モデル	状態方程式モデル
解析可能な回路	非線形 時変	線形 時不変	線形 時不変	線形 時不変 ¹

今日上で挙げたソフトウェアをはじめとする CAD ツールを用いれば、コンピュータ上で回路の振る舞いをシミュレーションできるから、実際の現実での試作回数を減らすことができ、費用の削減や設計時間の短縮につなげることができる。

現実の回路の振る舞いのシミュレーションは、その回路を理想化したモデルに基づいて行われる。モデルを作るプロセスモデリングは、回路の解析や設計において大切なステップである。これは、現実の回路を、動作点を決定し、動作周波数を考慮しながら、回路素子の電気的特性を線形化し、回路図に落とし込む作業に他ならない。

¹Wasabi は、将来的に時変システムに対応できる。

²使用するには、MATLAB と Symbolic Math Toolbox が必要。

現実の回路の振る舞いは、それに対応する回路図から、キルヒホッフの電圧・電流則やオームの法則、キャパシタ・インダクタの電圧・電流特性を基に、電気回路方程式を構築し、その方程式の解で表現される。

SPICEは、業界標準の回路シミュレータであり、回路の振る舞いを数値計算によって求め、結果をグラフにプロットすることができる。このことは、電気回路の振る舞いは、存在して一意であることを仮定している。同じように、現実の回路の振る舞いを測定するときも、その仮定のもとに行う。

したがって、現実の回路に対応するモデルは、その振る舞いの存在と一意性を持つべきである。言い換えれば、現実の回路や対応するモデルにおいて、各素子の電圧・電流は、すべての時刻上で定義される一価関数であるべきである。

いくつかの先行研究では、この振る舞いの存在と一意性が失われる回路図を紹介している。例えば [18, 19, 21, 22] にあるように、回路図中のすべてのインダクタ電流やキャパシタ電圧が状態変数でない場合が、これにあたる。この問題は、グラフ理論における、全域木を探す問題に翻訳できるとされる [18, 19, 22]。

本稿の第4章では、グラフ理論の問題に置き換えること無く、直接的に、モデルが振る舞いの存在と一意性を持つかを判定する方法を提案する。それは、線形回路方程式の解が存在して一意であるかを判定する方法である。

また、判定方法を提案するに当たって、電気回路の振る舞いが存在して一意であることを数学的に定義し、回路方程式の各係数のみの情報から、計算によって判定できることを示した。これによって、すでにある回路シミュレータ SPICE やシンボリック回路数式モデラ SAPWIN/SCAM/Wasabi に与えた回路図の検証に、容易に用いることが可能となっている。

回路の振る舞いの存在と一意性が失われる例として、スイッチング回路を取り上げることにする。この例は、あるスイッチの状態では、振る舞いの存在と一意性が保たれるが、あるスイッチの状態では、それらが破られる。このことは、スイッチング回路のように、回路トポロジーが時々刻々と変化する場合に、考え得るすべてのトポロジーについて、検証すべきことを示唆している。

参考文献

- [1] L. W. Nagel and D. O. Pederson. SPICE (Simulation Program with Integrated Circuit Emphasis). *Memorandum No. UCB/ERL M382 Electronic Research Laboratory*, 1973.
- [2] L. W. Nagel. Spice2: A computer program to simulate semiconductor circuits. *Memorandum No. UCB/ERL M520 Electronics Research Laboratory*, 1975.
- [3] T. L. Quarles. Analysis of performance and convergence issues for circuit simulation. *Memorandum No. UCB/ERL M89/42 Electronics Research Laboratory*, 1989.
- [4] A. Luchetta, S. Manetti, and A. Reatti. SAPWIN-a symbolic simulator as a support in electrical engineering education. *IEEE Trans. Educ.*, 44(2), 2001
<http://cirrlab.det.unifi.it/Sapwin/>.
- [5] E. Cheever. Symbolic Circuit Analysis in MatLab (SCAM).
<http://www.swarthmore.edu/NatSci/echeeve1/Ref/mna/MNA6.html>, 2015.
- [6] Kevin Wayne Robert Sedgewick. *Algorithms*. Addison-Wesley, 2014.
- [7] A. E. Ruehli Chung-Wen Ho and P. A. Brennan. The modified nodal approach to network analysis. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS*, CAS-22(6), 1975.
- [8] Wai-Kai Chen. Topological analysis for active networks. *IEEE Transactions on Circuit Theory*, 12(1):85–91, Mar 1965.
- [9] WATARU MAYEDA. *GRAPH THEORY*. Wiley-Interscience, 1972.
- [10] S. J. Mason. Feedback theory-some properties of signal flow graphs. *Proceedings of the IRE*, 41(9):1144–1156, Sept 1953.
- [11] R. Mielke. A new signal flowgraph formulation of symbolic network functions. *IEEE Transactions on Circuits and Systems*, 25(6):334–340, Jun 1978.
- [12] P. Sannuti and N. Puri. Symbolic network analysis-an algebraic formulation. *IEEE Transactions on Circuits and Systems*, 27(8):679–687, Aug 1980.
- [13] J. Starzyk and E. Sliwa. Upward topological analysis of large circuits using directed graph representation. *IEEE Transactions on Circuits and Systems*, 31(4):410–414, Apr 1984.
- [14] J. Starzyk and A. Konczykowska. Flowgraph analysis of large electronic networks. *IEEE Transactions on Circuits and Systems*, 33(3):302–315, Mar 1986.
- [15] M. M. Hassoun and Pen-Min Lin. A hierarchical network approach to symbolic analysis of large-scale networks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 42(4):201–211, Apr 1995.
- [16] Xiang-Dong Tan and C. J. R. Shi. Hierarchical symbolic analysis of analog integrated circuits via determinant decision diagrams. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(4):401–412, Apr 2000.

- [17] G. Shi, W. Chen, and C. J. R. Shi. A graph reduction approach to symbolic circuit analysis. In *2007 Asia and South Pacific Design Automation Conference*, pages 197–202, Jan 2007.
- [18] E. S. Kuh and R. A. Rohrer. The state-variable approach to network analysis. *Proceedings of the IEEE*, 53(7):672–686, July 1965.
- [19] 齐藤正男, 篠崎寿夫訳. 回路理論—状態変数解析入門—(原本: *Ronald A. Rohrer, CIRCUIT THEORY: An Introduction to the State Variable Approach*). 学献社, 1973.
- [20] T. Hirata, Y. Yamamoto, K. Yamaguchi, and I. Hodaka. A software of generating a symbolic circuit model with computers for wireless power transmission system. *WSEAS Trans. Circuits Syst.*, 13:266–273, 2014
<http://wasabi.matrix.jp/wp/>.
- [21] T. Bashkow. The a matrix, new network description. *IRE Transactions on Circuit Theory*, 4(3):117–119, Sep 1957.
- [22] D. A. Calahan. *Computer-Aided Network Design, Revised Edition*. McGraw-Hill, 1972.

CHAPTER 2

予備知識

本稿を読むに当たって、必要となる知識を紹介する。

2.1 線形代数学

本稿の後半の章で下記の定理の考え方を利用している。

定理 2.1.1 (次元定理). V, W をそれぞれベクトル空間とし、 V を有限次元とする。このとき、線形写像 $F : V \rightarrow W$ について、次が成立する。

$$\text{rank } F + \dim(\text{Ker } F) = \dim V$$

定理 2.1.2. V をベクトル空間、 U, W を V の有限次元部分空間とする。このとき、 $U \cap W$ と $U + W$ は有限次元であり、次の等式が成立する。

$$\dim(U + W) = \dim U + \dim W - \dim(U \cap W)$$

定理 2.1.3. 代数方程式 $Ax = b$ が解を持つための必要十分条件は、次である。

$$\text{rank } A = \text{rank}[A \ b]$$

2.2 制御理論

回路の振る舞いを記述するための数式モデルとして、状態方程式を用いる。状態方程式を用いれば、必要最小限の方程式によって、時間に対する回路の振る舞いを記述できる。また、制御理論をはじめとするシステム論的なアプローチが可能になる。

2.2.1 状態方程式とその一般解

線形時不変システムは次のように書かれる。

$$\dot{x}(t) = Ax(t) + Bu(t) \quad : \quad A(n \times n), B(n \times m) \quad (2.2.1)$$

$$y(t) = Cx(t) + Du(t) \quad : \quad C(l \times n), D(l \times m) \quad (2.2.2)$$

式 (2.2.1) を状態方程式と呼び、 $x(t)$ を状態変数、 $u(t)$ を入力と呼ぶ。式 (2.2.2) は、出力方程式と呼ばれる。これら二つの式をまとめて状態空間と呼ぶ。

状態方程式の一般解 $x(t)$ は、行列指数関数 e^{At} を用いて、

$$x(t) = e^{At}x(t_0) + e^{At} \int_{t_0}^t e^{-Ap}Bu(p)dp \quad (2.2.3)$$

とかける。

電気回路の数式モデルとして、状態空間表現を用いるとき、入力 $u(t)$ は、回路中の電源となる。 $x(t)$ は、回路中のすべてのキャパシタの電圧・インダクタの電流であることが多いが、そうでない場合もあることに注意する。

2.2.2 状態方程式とその周期解

電源が周期関数であるとき、線形時不変な回路の定常状態を数式表現できる。その準備として、次の命題を証明する。

命題 2.2.1.

$$\psi(t-T) = Q\psi(t) \text{ ならば } (I-Q) \int_0^t \psi(p)dp = -Q \int_0^T \psi(p)dp + \int_0^T \psi(t-p)dp$$

Q は行列、 I は単位行列を表す。

証明. 次式の左辺は、置換積分法と積分区間を分割すれば、

$$\int_0^T \psi(t-p)dp = \int_{t-T}^0 \psi(p)dp + \int_0^t \psi(p)dp \quad (2.2.4)$$

となる。

ここで、右辺の第2項、置換積分法を適用して、 $\psi(t-T) = Q\psi(t)$ を用いると、

$$\int_0^t \psi(p)dp = Q \int_t^T \psi(p)dp$$

となる。

上式の右辺の積分区間を分割すれば、

$$Q \int_t^T \psi(p)dp = Q \int_0^T \psi(p)dp - Q \int_0^t \psi(p)dp$$

となる。

式(2.2.4)の右辺第2項に代入して整理すれば、命題の式を得る。 \square

準備ができたので、線形時不変な回路の定常状態を数式表現を導く。電源 $u(t)$ が周期 T を持つ関数だから、任意の t に対して、

$$u(t+T) = u(t)$$

が成り立つ。ここで、 $\psi(t) = e^{-At}Bu(t)$ とおいて、次式を計算する。

$$\psi(t-T) = e^{-A(t-T)}Bu(t-T) = e^{AT}e^{-At}Bu(t) = e^{AT}\psi(t)$$

となる。命題 2.2.3 において、 $Q = e^{AT}$, $\psi(t) = e^{-At}Bu(t)$ とおくと、

$$(I - e^{AT}) \int_0^t e^{-Ap}Bu(p)dp = -e^{AT} \int_0^T e^{-Ap}Bu(p)dp + \int_0^T e^{-A(t-p)}Bu(t-p)dp$$

となる。

$\det(I - e^{AT}) \neq 0$ のとき、すなわち、 $T > 0$ より、 $A \neq 0$ のとき、

$$\begin{aligned} \int_0^t e^{-Ap}Bu(p)dp &= -(I - e^{AT})^{-1}e^{AT} \int_0^T e^{-Ap}Bu(p)dp + \\ &\quad (I - e^{AT})^{-1} \int_0^T e^{-A(t-p)}Bu(t-p)dp \end{aligned}$$

(2.2.3) の第 2 項において、 $t_0 = 0$ として、上式を代入すれば、

$$\begin{aligned} x(t) &= e^{At} \left(x(0) - (I - e^{AT})^{-1}e^{AT} \int_0^T e^{-Ap}Bu(p)dp \right) + \\ &\quad (I - e^{AT})^{-1} \int_0^T e^{Ap}Bu(t-p)dp \end{aligned}$$

となる。とくに初期値 $x(0)$ が、

$$x(0) = (I - e^{AT})^{-1}e^{AT} \int_0^T e^{-Ap}Bu(p)dp$$

であれば、

$$x(t) = (I - e^{AT})^{-1} \int_0^T e^{Ap}Bu(t-p)dp$$

が、電源の関数と同じ周期 T をもつ関数となることがわかる。この式を周期解、または定常解と呼び、本稿では以降 $x_{ss}(t)$ と表現する。

この $x_{ss}(t)$ から、回路の定常状態や平均電力などを数式表現できる。実際にワイヤレス給電の回路の例を用いて、後ほど紹介する。

また、一般に、回路シミュレータを用いて、回路の定常状態を調べるとき、線形時不変な回路においても、定常状態応答に至るまでのすべてを計算するが、この $x_{ss}(t)$ を使えば、定常状態応答の 1 周期分だけをすぐに計算でき、計算時間や手間を削減できる。それは、例えば回路の平均電力を計算する場合に有効である。

上の議論をまとめると、周期 $T > 0$ として、 $u(t+T) = u(t)$, $A \neq 0$ のとき、線形時不変システムの周期解 $x_{ss}(t)$ は、

$$x_{ss}(t) = (I - e^{AT})^{-1} \int_0^T e^{Ap}Bu(t-p)dp \quad (2.2.5)$$

とかける。

2.2.2.1 電源が正弦波関数である場合の回路の周期解

線形時不変な回路で、電源が正弦波入力 $u(t) = a \sin \omega t$ であるときの、回路の定常状態における応答、すなわち、周期解を求めておく。

式(2.2.5)の積分項について、計算を進めれば、

$$\int_0^T e^{Ap} B u(t-p) dp = -a(\omega^2 I + A^2)^{-1} (I - e^{AT}) (\omega I \cos \omega t + A \sin \omega t) B$$

となる。これを、式(2.2.5)に代入すれば、

$$x_{ss}(t) = -a(\omega^2 I + A^2)^{-1} (\omega I \cos \omega t + A \sin \omega t) B \quad (2.2.6)$$

となる。

2.2.3 状態方程式から伝達関数を算出する方法

式(2.2.1)において、 $x(0) = 0$ とする。そのとき、電源から状態変数への伝達関数ベクトル $G(s)$ は、次の式で一意的に求められる。

$$G(s) = \frac{X(s)}{U(s)} = (sI - A)^{-1} B \quad G(s) : (n \times 1)$$

ただし、 $x(t)$ 、 $u(t)$ のラプラス変換をそれぞれ、 $X(s)$ 、 $U(s)$ と表した。

2.3 表記上の注意

特に断りがない限り次の表記を採用する。

2.3.1 部品の電圧極性と電流の向き

本稿における電気回路を構成する部品(素子、電源)は、一端子対であり、電圧 v ・電流 i に関する変数を持つ。その極性や向きを図のように仮定する。つまり、電流は、素子の高電位から低電位側に流れると仮定するのである。電気回路図上では、回路図の煩雑さを避けるために、電圧の極性か電流の向きのどちらか一方を記載する。

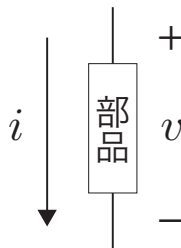


図 2.1: 部品の電圧極性と電流の向き

2.4 グラフ理論

2.4.1 グラフによる電気回路の表現

例えば、次の図の左側における回路をグラフで表現すると、右側のようになる。

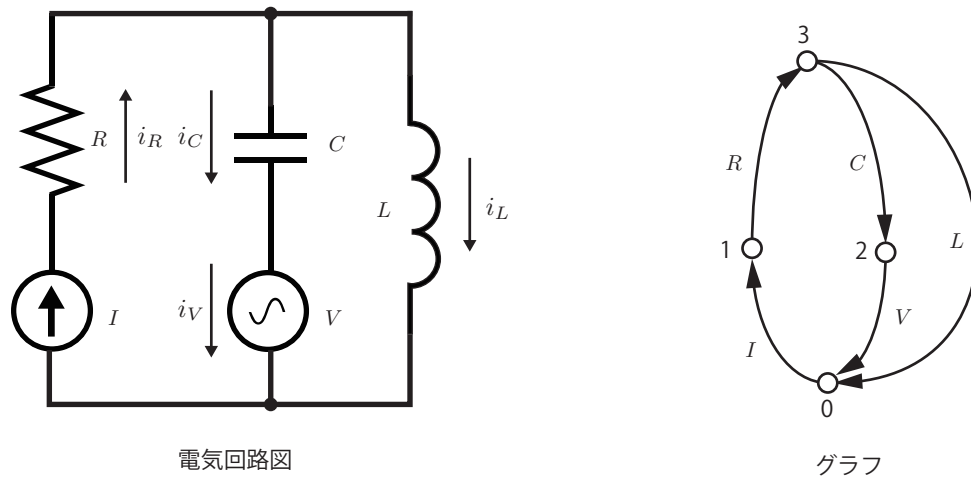


図 2.2: 電気回路図のグラフ表現

円が示すのは、等電位箇所であり、これをノード (node) と呼ぶ。線分の矢印が素子の電流の向きを表しており、この線分を辺や枝 (branch) と呼ぶ。グラフは、ノードと線分で表現されるものであり、線分に向きが着いているので、有向グラフ (directed graph, oriented graph) と呼ばれる。一般に電気回路網を表現するグラフは、有向グラフである。

任意の二つのノードの間に、一方のノードから出発して、枝を経由して他方のノードへたどり着くことができるとき、そのグラフを連結グラフ (connected graph) と呼ぶ。

部分グラフ (subgraph) G' とは、グラフ G のノードの集合と枝の集合の一部をもつ、グラフのことである。

閉路 (loop, circuit) とは、グラフ G の部分グラフ G' において、 G' が連結グラフであり、かつ、すべてのノードに2つの枝が接続されているとき、グラフ G の閉路と呼ぶ。閉路には、方向があり、閉じた枝に沿って決定される。

カットセット (cut-set) とは、グラフ G からある枝の集合を除去すると、 G のノード集合が二分される最小個の枝の集合のことである。カットセットには、方向があり、二分されたノード集合の一方から、他方へ向かってカットセットの向きが決定される。

木 (tree) T とは、グラフ G のすべてのノードを含み、閉路を持たないような部分グラフ T のことである。

補木 (cotree) とは、グラフ G から木 T のもつ枝をすべて除去した、残りの枝の集合を、木 T に対する補木と呼ぶ。補木を構成する各々の枝を、補木枝 (link) と呼ぶ。

自己ループ (self loop) とは、ひとつの枝に注目したときに、その枝のふたつの端点が同じノードに接続されている部分グラフ G' のことである。電気回路における部品のショートと同義である。

2.4.2 グラフの数式表現

2.4.2.1 接続行列

ショートがない連結グラフ G の接続関係を数的表現を考える。
行列 A の

行方向:ノード, 列方向:枝

と割り当て、次のルールで A の要素 a_{ij} を埋める。

$$a_{ij} = \begin{cases} 1: & \text{枝 } j \text{ がノード } i \text{ に接続し, 枝の向きが } i \text{ から外へ向かう時} \\ -1: & \text{枝 } j \text{ がノード } i \text{ に接続し, 枝の向きが } i \text{ から内へ向かう時} \\ 0: & \text{枝 } j \text{ がノード } i \text{ に接続していない時} \end{cases} \quad (2.4.1)$$

この行列 A を接続行列 (**Incidence Matrix**) と呼ぶ。

例

図 2.2 のグラフにおいて, その接続行列 A は,

$$A = \begin{array}{c} \text{ノード/枝} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \end{array} \begin{bmatrix} I & V & R & C & L \\ -1 & 1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 \end{bmatrix}$$

と表すことができる。

2.4.2.2 接続行列 A について

- すべての行を足すと 0 になる (列ベクトルの要素の総和は, 必ず 0 になる。任意の 1 行が他の行の線形結合でかける。)
- 任意の 1 行を除いた行列 A を既約 (基底) 接続行列と呼ぶ
任意の 1 行を除いていない行列 A を拡大接続行列と呼ぶこともある。
- 除いた任意の 1 行に該当するノードを基準ノード, 基準節点 (reference node) とよぶ。
- n 個のノードを有する連結グラフ G の (拡大) 接続行列 A について, $\text{rank } A = n - 1$ である。

2.4.2.3 接続行列を用いたキルヒホッフの電流則の記述

接続行列 A を用いてキルヒホッフの電流則を記述することができる。 \mathbf{I} を A の列に対応する素子の電流が並んだベクトルとすれば,

$$A\mathbf{I} = 0$$

と書ける。接続行列 A は, 既約接続行列であれば, 重複した方程式を含まないことになる。そこで, 一般には, 既約接続行列を用いて表記する。

2.4.3 閉路行列

ショートがない連結グラフ G の閉路を数的に表すことを考える。行列 B の

行方向:閉路, 列方向:枝

と割り当て、次のルールで行列 B の要素 b_{ij} を埋める。

$$b_{ij} = \begin{cases} 1: & \text{枝 } j \text{ が閉路 } i \text{ に含まれ, 枝の向きと閉路の向きが一致する時} \\ -1: & \text{枝 } j \text{ が閉路 } i \text{ に含まれ, 枝の向きと閉路の向きが一致しない時} \\ 0: & \text{枝 } j \text{ が閉路 } i \text{ に含まれない時} \end{cases} \quad (2.4.2)$$

この行列 B を閉路行列 (**circuit matrix**) と呼ぶ。

2.4.3.1 閉路行列の例

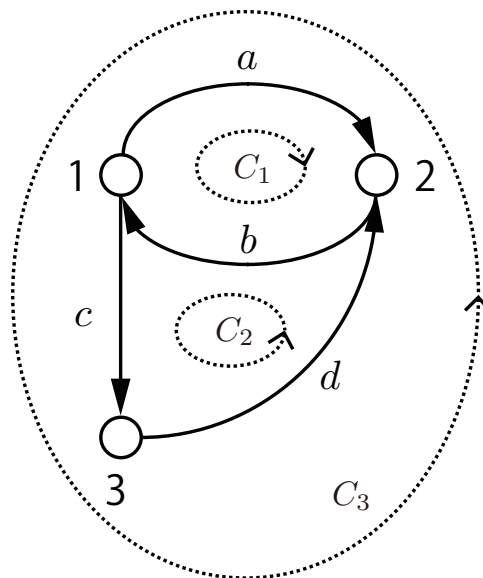


図 2.3: $C_1 = \{a, b\}$, $C_2 = \{b, c, d\}$, $C_3 = \{a, c, d\}$

このグラフの閉路行列 B は、

$$B = \begin{array}{c} \text{閉路/枝} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{bmatrix} a & b & c & d \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \end{bmatrix}$$

である。ここで、行ベクトルについて、 $C_1 + C_3 = C_2$ であることから、 C_2 は従属である。

接続行列のときと同じように、閉路行列も各々が線形独立な数的表現を行うことを考える。それには、グラフから木をひとつ見つければよい。ここで、便宜のために次のように言葉を定義しておく。

基本閉路 (fundamental circuit) とは、グラフの木を一つ選んだときに、一つのリンク (補木の枝) と木の枝から構成される閉路をいい、その向きは、含まれるリンクの方向と一致させるものとする。

また、**基本閉路行列 B_f** は、グラフ G の基本閉路に関して、閉路行列を構築したものである。

2.4.3.2 基本閉路行列の例

図 2.3 のグラフにおいて、枝 $\{a, c\}$ から構築される木を選ぶと、基本閉路行列 B_f は、

$$B_f = \begin{array}{c} \text{閉路/枝} \\ C_1 \\ C_3 \end{array} \begin{array}{cccc} b & d & a & c \\ \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{array} \right] \end{array}$$

と書くことができる。行列 B_f の木の列のブロック行列は、単位行列になっていることがわかる。基本閉路行列 B_f は、この例のように、一般に次の形にすることができる。

2.4.3.3 基本閉路行列 B_f の分解

ショートがない連結グラフ G (ノード総数: n , 枝総数: b) の基本閉路行列 $B_f (b-n+1 \times b)$ は

$$B_f = {}_{b-n+1}\{ \left[\begin{array}{c|c} \overbrace{E}^{\text{リンク: } b-n+1} & \overbrace{B_{f12}}^{\text{木: } n-1} \end{array} \right] \}$$

と分解でき、 E の列の集合は補木を表し、 B_{f12} の列の集合は木を表す。 E は単位行列。

2.4.3.4 基本閉路行列を用いたキルヒホッフの電圧則

基本閉路行列 B_f を用いてキルヒホッフの電圧則を記述することができる。 \mathbf{V} を B_f の列に対応する素子の電圧が並んだベクトルとすれば、

$$B_f \mathbf{V} = 0$$

と書ける。

2.4.4 カットセット行列

ショートがない連結グラフ G のカットセットを数的に表すことを考える。行列 C の

行方向: カットセット 列方向: すべての枝

と割り当て、次のルールで行列 C の要素 c_{ij} を次のルールで埋める。

$$c_{ij} = \begin{cases} 1: & \text{枝 } j \text{ がカットセット } i \text{ に含まれ、それらの向きが一致する時} \\ -1: & \text{枝 } j \text{ がカットセット } i \text{ に含まれ、それらの向きが反対の時} \\ 0: & \text{上記以外} \end{cases} \quad (2.4.3)$$

この行列 C をカットセット行列 (cut-set matrix) と呼ぶ。

接続行列や閉路行列の時と同じように、カットセット行列も線形独立な行列を作る方法がある。

2.4.4.1 基本カットセット行列

基本カットセット (fundamental cut-set) とは、グラフから一つの木を選び、木枝とリンクから成るカットセットをいい、その向きは、含まれる木枝の方向と一致させて定める。基本カットセットに対して、カットセット行列 C_f を構築する。この行列 C_f を基本カットセット行列と呼ぶ。

ショートがない連結グラフ G の枝の数を b 、ノード数を n_v とすると、基本カットセットは、 $n_v - 1$ だけ存在する。したがって、基本カットセット行列 C_f は、そのサイズが $n_v - 1 \times b$ となる。

例

図 2.2 において，次のように木を選ぶ。

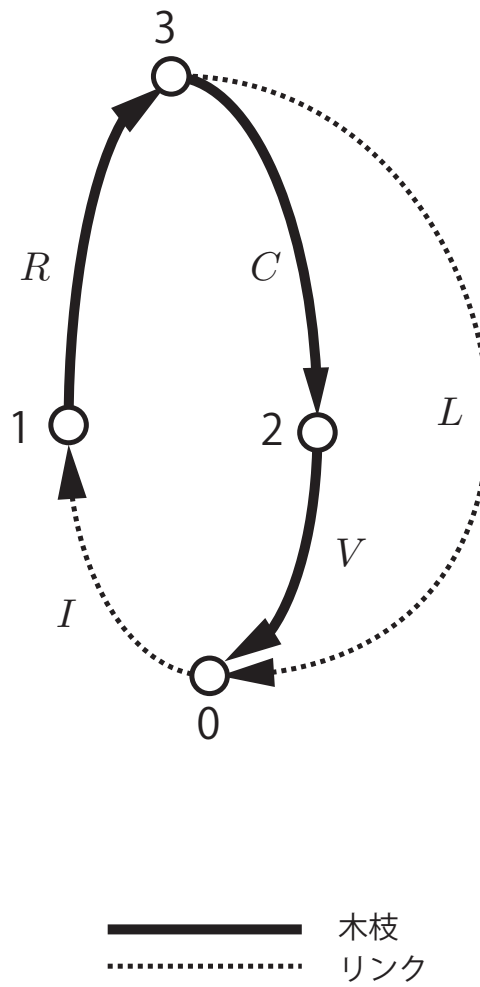


図 2.4: グラフの木

このときに，次のように基本カットセットが定まる。

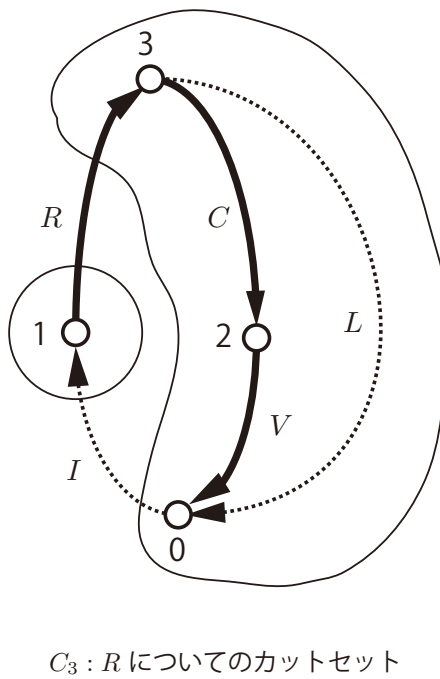
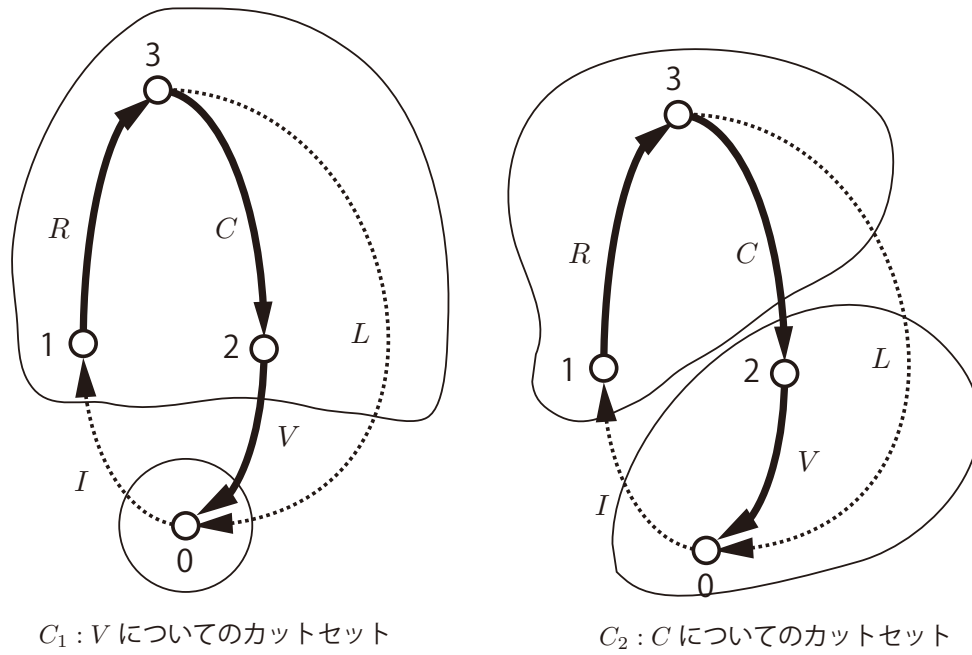


図 2.5: 基本カットセットによるグラフが分離される様子

然るに、カットセット行列 C_f は、

$$C_f = \begin{array}{c} \text{カットセット/枝} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{ccccc} V & C & R & L & I \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{array} \right] \end{array}$$

となる。

基本カットセット行列 C_f は一般に次の形に書くことができる。

2.4.4.2 基本カットセット行列 C_f の分解

ショートがない連結グラフ G (ノード総数: n , 枝総数: b) の基本カットセット行列 C_f ($(n-1) \times b$) は

$$C_f = \begin{matrix} \text{木:n-1} & \text{リンク:b-n+1} \\ \left[\begin{array}{cc} E & C_c \end{array} \right] \end{matrix}$$

と分解でき、 E の列の集合は木を表し、 C_c の列の集合は補木を表す。 E は単位行列。

2.4.4.3 基本カットセット行列によるキルヒホッフの電流則

基本カットセット行列 C_f の列に対応する木枝の電流が並んだベクトルを \mathbf{I} とすれば、キルヒホッフの電流則は、

$$C_f \mathbf{I} = 0$$

と書くことができる。上式は、基本カットセット行列 C_f の代わりに、カットセット行列 C でも成り立つが、 C_f を用いることで、電流則は一次独立で書かれる。

この式を幾何学的に解釈すれば、カットセットによって分離された二つのグラフ (回路) の流れ込む電流と流れ出す電流の総和が等しくなければならないことを指している。

2.4.5 各行列の関係式

既約接続行列から代数的な演算によって、基本閉路行列や基本カットセット行列する方法が知られている。ここでは、それらを紹介する。

ショートがない連結グラフ G の既約接続行列を A とする。このときに、 G の木 T を一つ見つけて、 A を次のように分解する。

$$A = \begin{matrix} \text{補木} & \text{木} \\ \left[\begin{array}{cc} A_{11} & A_{12} \end{array} \right] \end{matrix}$$

A_{11} は、補木を表す既約接続行列であり、 A_{12} は、木を表す既約接続行列である。また、 A_{12} は正則であることが知られている [5]。このときに、次が成り立つ [5]。

基本カットセット行列を C_f 、基本閉路行列を B_f とし、木 T とその補木に、既約接続行列 A の列の並びと同じになるように、分解する。

$$C_f = \begin{matrix} \text{木} & \text{補木} \\ \left[\begin{array}{cc} E & P \end{array} \right], B_f = \begin{matrix} \text{木} & \text{補木} \\ \left[\begin{array}{cc} -P^t & E \end{array} \right] \end{matrix} \end{matrix}$$

E は単位行列を表す。ただし、行列 P は、

$$P = A_{12}^{-1} A_{11}$$

である。

任意のショートがない連結グラフ G に対して、既約接続行列 A を構築すれば、上述の計算をすることで、KVL/KCL に対応した回路方程式を作ることができる。

2.4.6 状態方程式を計算するアルゴリズム

グラフ理論を回路網に適用することで、回路の数式モデルとして、状態方程式を導くことのできるアルゴリズムを紹介する [1-3]。

このアルゴリズムは、グラフから“適切な木”を見つけ、状態方程式を生成する。

また、状態変数を、回路中の個々のインダクタ電流とキャパシタ電圧とするのではなく、電荷と磁束としている。このようにするには、一般にいつもすべてのインダクタ電流とキャパシタ電圧が状態変数になるとは限らないからである。この例は、この節の後半に紹介する。

回路の条件

- 電圧源のみの閉路は存在しない
- 電流源のみのカットセットは存在しない

ステップ1 次を満たす木・補木を作る。

- 電圧源はすべて木の枝とする
- 電流源ははすべて補木の枝とする
- Cを木の枝に「できるだけ」割り当てる
- Lを補木の枝に「できるだけ」割り当てる
- 抵抗については、木でも補木の枝になっても良い

ステップ2 枝電圧ベクトル \mathbf{V} ・枝電流ベクトル \mathbf{I} を次のように書く。

$$\mathbf{V} = \begin{bmatrix} v_V \\ v_C \\ v_R \\ v_L \\ v_C \\ v_R \\ v_L \\ v_I \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} i_V \\ i_C \\ i_R \\ i_L \\ i_C \\ i_R \\ i_L \\ i_I \end{bmatrix}$$

V : 電圧源 (木)

C : C(木)

R : R(木), コンダクタンス

L : L(木)

C : C(補木)

R : R(補木)

L : L(補木)

I : 電流源 (補木)

回路方程式

(KVL/KCL)

基本カットセット行列 C_f と基本閉路行列 B_f を用いて,

$$C_f \mathbf{I} = 0 \text{ (KCL)}$$

$$B_f \mathbf{V} = 0 \text{ (KVL)}$$

(素子の電気的特性)

$$C \text{ についての電気的特性: } i_C = C_C \dot{v}_C, \quad i_C = C_C \dot{v}_C$$

$$R \text{ についての電気的特性: } v_R = R_R i_R, \quad i_R = R_R v_R$$

$$L \text{ についての電気的特性: } v_L = L_L \dot{i}_L, \quad v_L = L_L \dot{i}_L$$

$$C_f = \begin{bmatrix} \text{木 (VCRL)} & \text{補木 (CRLI)} \\ E & P \end{bmatrix}, \quad B_f = \begin{bmatrix} \text{木 (VCRL)} & \text{補木 (CRLI)} \\ -P^t & E \end{bmatrix}$$

E は単位行列を表す。

回路の既約接続行列を A として、次のように分解する。

$$A = \begin{bmatrix} \text{木 (VCRL)} & \text{補木 (CRLI)} \\ A_{12} & A_{11} \end{bmatrix}$$

ただし、行列 P は、 $P = A_{12}^{-1} A_{11}$ である。また、行列 P は、行が木の枝数、列がリンクの数に等しい。

ステップ3 P を次のように分解する。

$$P = \begin{matrix} \text{電圧源 (木枝) を含むカットセット} \\ \text{C (木枝) を含むカットセット} \\ \text{R (木枝) を含むカットセット} \\ \text{L (木枝) を含むカットセット} \end{matrix} \begin{bmatrix} P_{VC} & P_{VR} & P_{VL} & P_{VI} \\ P_{CC} & P_{CR} & P_{CL} & P_{CI} \\ P_{RC} & P_{RR} & P_{RL} & P_{RI} \\ P_{LC} & P_{LR} & P_{LL} & P_{LI} \end{bmatrix} \quad (2.4.4)$$

ただし、いつも $P_{RC} = 0$, $P_{LR} = 0$, $P_{LC} = 0$ となる。

ステップ4 状態変数の取り方を v_C や i_L のように素子の電圧・電量の単位ではなく、正味の電荷・磁束で考える。すなわち、

$$x = \begin{bmatrix} q \\ \Phi \end{bmatrix}, \quad u = \begin{bmatrix} v_V \\ i_I \end{bmatrix}$$

とする。ただし、

$$q = C_0 v_C + P_{CC} C_C P_{VC}^t v_V \quad (2.4.5)$$

$$\Phi = L_0 i_L + P_{LL}^t L_L P_{LI} i_I \quad (2.4.6)$$

である。

ここで、行列 A , B を、前の意味を忘れて、次のように改めて定義する。

$$A = \begin{bmatrix} A_1 C_0^{-1} & A_2 L_0^{-1} \\ A_3 C_0^{-1} & A_4 L_0^{-1} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 - A_1 C_0^{-1} P_{CC} C_C P_{VC}^t & B_2 - A_2 L_0^{-1} P_{LL}^t L_L P_{LI} \\ B_3 - A_3 C_0^{-1} P_{CC} C_C P_{VC}^t & B_4 - A_4 L_0^{-1} P_{LL}^t L_L P_{LI} \end{bmatrix}$$

$$\begin{aligned}
 A_1 &= -P_{CR}R_0^{-1}P_{CR}^t \\
 A_2 &= (P_{CR}R_0^{-1}P_{\mathbb{R}\mathbb{R}}^t\mathbb{R}_0^{-1}P_{\mathbb{R}\mathbb{L}} - P_{CL}) \\
 A_3 &= (P_{CL}^t - P_{\mathbb{R}\mathbb{L}}^t\mathbb{R}_0^{-1}P_{\mathbb{R}\mathbb{R}}R_R^{-1}P_{CR}^t) \\
 A_4 &= -P_{\mathbb{R}\mathbb{L}}^t\mathbb{R}_0^{-1}P_{\mathbb{R}\mathbb{L}} \\
 B_1 &= -P_{CR}R_0^{-1}P_{VR}^t \\
 B_2 &= (P_{CR}R_0^{-1}P_{\mathbb{R}\mathbb{R}}^t\mathbb{R}_0^{-1}P_{\mathbb{R}\mathbb{I}} - P_{CI}) \\
 B_3 &= (P_{VL}^t - P_{\mathbb{R}\mathbb{L}}^t\mathbb{R}_0^{-1}P_{\mathbb{R}\mathbb{R}}R_R^{-1}P_{VR}^t) \\
 B_4 &= -P_{\mathbb{R}\mathbb{L}}^t\mathbb{R}_0^{-1}P_{\mathbb{R}\mathbb{I}}
 \end{aligned}$$

$$\begin{aligned}
 C_0 &= C_C + P_{CC}C_C P_{CC}^t \\
 L_0 &= L_L + P_{LL}^t L_L P_{LL} \\
 R_0 &= R_R + P_{\mathbb{R}\mathbb{R}}^t \mathbb{R}_0^{-1} P_{\mathbb{R}\mathbb{R}} \\
 \mathbb{R}_0 &= \mathbb{R}_\mathbb{R} + P_{\mathbb{R}\mathbb{R}}R_R^{-1}P_{\mathbb{R}\mathbb{R}}^t
 \end{aligned}$$

として,

$$\dot{x} = Ax + Bu$$

2.4.6.1 アルゴリズムの適用例

上記のアルゴリズムを各回路に適用し、状態方程式を生成する過程を紹介する。

例 1

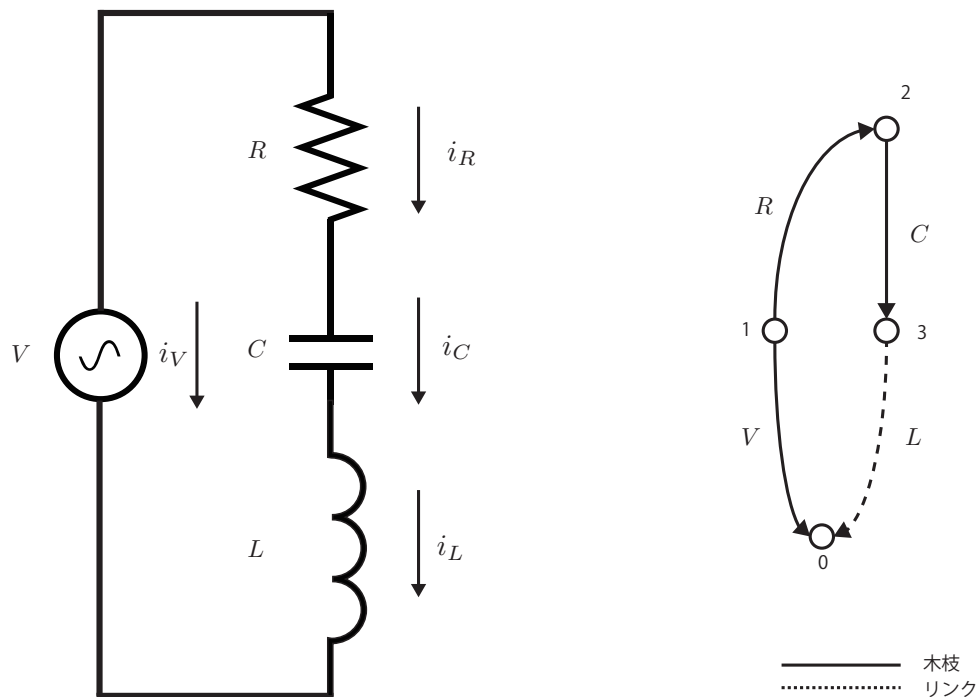


図 2.6: LCR 回路とその木

既約接続行列は,

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{cccc} & V & C & R & L \\ \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{array} \right] \end{array}$$

である。ひとつの木は, アルゴリズムの木を作るルールに沿って選ばれていて, 図 2.6 の右に示されている。既約接続行列 A を木と補木について分解すると,

$$A_{11} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} L \\ \left[\begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right] \end{array}, \quad A_{12} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{ccc} & V & C & R \\ \left[\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & -1 & 0 \end{array} \right] \end{array}$$

となる。これらの接続行列と (2.4.4) から,

$$P = A_{12}^{-1} A_{11} = \begin{array}{c} \\ \\ \\ \end{array} \begin{array}{c} L \\ \text{Vを含むカットセット} \\ \text{Cを含むカットセット} \\ \text{Rを含むカットセット} \end{array} \begin{array}{c} \\ \\ \\ \end{array} \begin{array}{c} L \\ \left[\begin{array}{c} 1 \\ -1 \\ -1 \end{array} \right] \end{array}$$

すなわち,

$$P_{VL} = 1, P_{CL} = -1, P_{RL} = 1$$

である。また,

$$C_c = C, L_L = L, R_R = \frac{1}{R}$$

である。

状態変数を次のようにとって, ステップ4の計算を進める。

$$x = \begin{bmatrix} C v_C \\ L i_L \end{bmatrix}, \quad u = v_V$$

であり,

$$A_1 = 0, A_2 = 1, A_3 = -1, A_4 = -R, B_1 = 0, B_3 = 1$$

である。ゆえに, 状態方程式は,

$$\dot{x} = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

となる。

状態変数をキャパシタの電圧 v_C ・ インダクタの電流 i_L , すなわち

$$x = \begin{bmatrix} v_C \\ i_L \end{bmatrix}$$

とすれば, 上式は,

$$\dot{x} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u$$

となる。

この例では, 回路中のすべてのキャパシタの電圧・インダクタ電流を状態変数として選ぶことができる。ところが一般には, そのように選ぶことができない例もある。その例を次に紹介する。

例 2

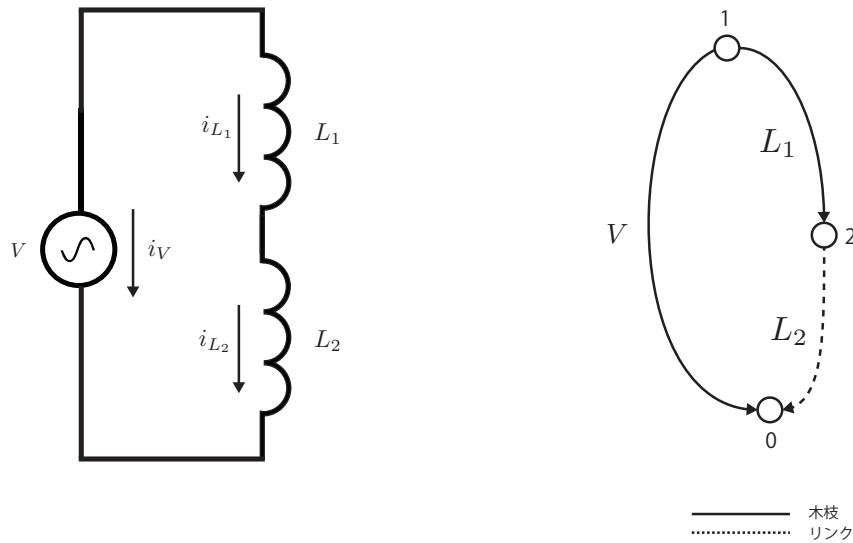


図 2.7: インダクタが直列の回路

既約接続行列は,

$$A = \begin{matrix} & V & L_1 & L_2 \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \end{matrix}$$

である。

回路から、ひとつの木を、ルールに沿って上図の右のように選び、既約接続行列 A を木と補木について分解すると,

$$A_{11} = \begin{matrix} & L_2 \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{matrix}, \quad A_{12} = \begin{matrix} & V & L_1 \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \end{matrix}$$

となる。これらの接続行列と (2.4.4) から,

$$P = A_{12}^{-1} A_{11} = \begin{matrix} & L_2 \\ \begin{matrix} V \text{ を含むカットセット} \\ L_1 \text{ を含むカットセット} \end{matrix} & \begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{matrix}$$

すなわち,

$$P_{LL} = -1, \quad P_{VL} = 1$$

である。また,

$$L_L = L_2, \quad L_{\mathbb{L}} = L_1$$

である。状態変数は, (2.4.6) から,

$$x = (L_1 + L_2)i_{L_2}, \quad u = v_V$$

である。

ステップ 4 の計算を進めて,

$$B_3 = 1$$

となるから、ゆえに、状態方程式は、

$$\dot{x} = u$$

となる。

状態変数をインダクタの電流、すなわち

$$x = i_{L_2}$$

とすれば、上式は、

$$\dot{x} = \frac{1}{L_1 + L_2} u$$

となる。これは、 i_{L_1} と i_{L_2} が同時に状態変数となり得ない例である。

このとき、基本カットセット行列 C_f を用いたキルヒホッフの電流則表現から

$$i_{L_1} - i_{L_2} = 0$$

とわかる。 i_{L_1} と i_{L_2} は、片方が与えられれば、片方はそれに拘束される。このような場合に、状態変数はすべてのインダクタ電流を取ることができない。グラフ理論の言い方をすれば、インダクタからなるカットセットが存在するときに、状態変数にすべてのインダクタ電流を取ることができない。

同様にキャパシタだけからなる閉路が存在するとき、状態変数にすべてのキャパシタ電圧を取ることができない(図 2.8)。

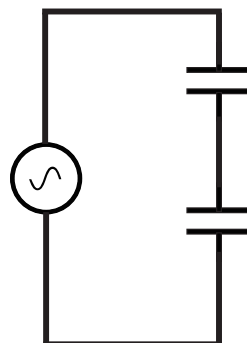


図 2.8: キャパシタと電圧源の閉路からなる回路

そのほかにも、いくつかの例を紹介する。

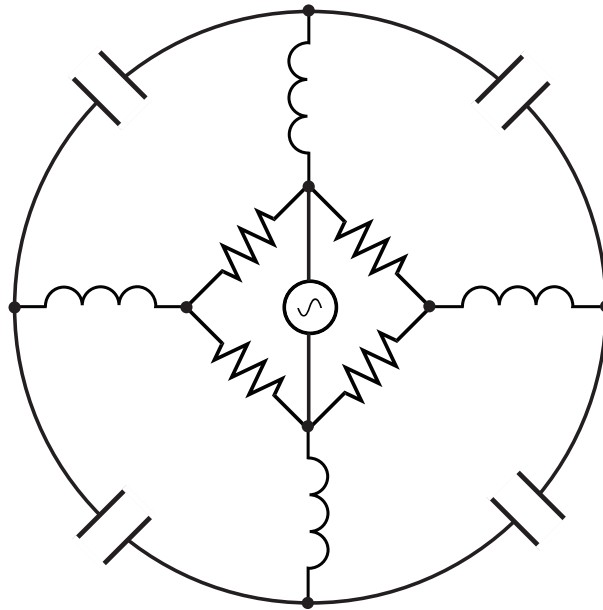


図 2.9: すべてのインダクタ電流・キャパシタ電圧が状態変数にならない回路

図 2.9 の回路は、外周にキャパシタからなる閉路が存在する。従って、すべてのキャパシタ電圧は、状態変数にならない。また、すべてのインダクタを取り除くと、回路が二つの部分に分かれる (インダクタからなるカットセットが存在する)。従って、すべてのインダクタ電流は状態変数になることはできない。

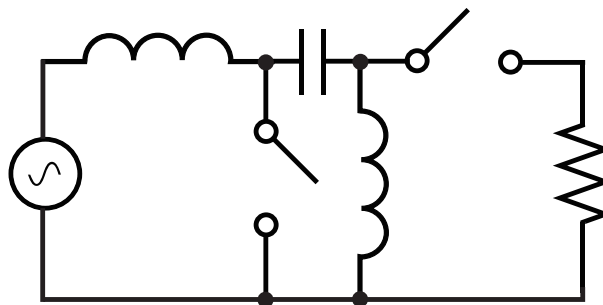


図 2.10: SEPIC コンバータ (DCM) のモデル

これは、SEPIC と呼ばれる DC/DC コンバータの例である。スイッチの部分は実際の回路では、MOSFET およびダイオードである。MOSFET が外部クロックによって、スイッチングされて、回路のトポロジーが変化する。

スイッチング DC/DC コンバータの解析は、ダイオードやトランジスタをスイッチとして扱い解析を行う方法があり、continuous conduction mode (CCM) と discontinuous conduction mode (DCM) [9] と呼ばれる。図 2.10 は、DCM と呼ばれる動作モードを示している。

このモードにおいて、すべてのインダクタ電流は状態変数になり得ない。一方で、CCM の動作においては、すべてのインダクタ電流が状態変数になる。すなわち、CCM と DCM では、状態方程式モデルの次元が異なる。

他の種類の DC/DC コンバータ、例えば、Boost、Ćuk コンバータなども、モデリングと動作モードによっては、同様なことがおこる。本稿の第 4 章で Boost コンバータを例に詳しく紹介する。

参考文献

- [1] 齊藤正男, 篠崎寿夫訳. 回路理論—状態変数解析入門—(原本: Ronald A. Rohrer, CIRCUIT THEORY: An Introduction to the State Variable Approach). 学献社, 1973 年.
- [2] E.S. Kuh, R.A. Rohrer, The state-variable approach to network analysis. Proceedings of the IEEE, 53(7):672-686, July 1965.
- [3] 齊藤正男, 回路解析論, コロナ社, 1974 年.
- [4] 小野田真穂樹, 現代回路理論, 昭晃堂, 1983 年.
- [5] WATARU MAYEDA. GRAPH THEORY. Wiley-Interscience, 1972.
- [6] 松坂 和夫. 線型代数入門. 岩波書店, 1980 年.
- [7] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*, Second Edition. Academic Press, 1985.
- [8] 小郷 寛, 美多 勉. システム制御理論入門. 実教出版株式会社, 1979 年.
- [9] R. W. Erickson and D. Maksimović. *Fundamentals of Power Electronics*, Second Edition. Kluwer, 2004.

CHAPTER

3

回路方程式から状態方程式へのシンボリック変換について

3.1 はじめに

この章では、回路を構成する素子の値や電源関数を記号として扱い、回路の時間的な振る舞いを表す、シンボリックな状態方程式を生成するアルゴリズムについて提案する。このアルゴリズムは、すべてのキャパシタ電圧とインダクタ電流を、状態変数として扱うことができる場合に限られることに注意する。より詳しくいえば、与えられた回路が、「プロパーな回路(第4章で述べる)」であれば、ここで紹介するアルゴリズムで、状態方程式を求めることができる。第2章の2.4.6節で述べたように、回路図から状態方程式を生成する従来の研究は、(上記の場合においても)“適切な木”を見つける必要があったが、提案するアルゴリズムでは、その代わりに、線形代数を用いた演算を行っていることが大きな違いである。

回路定数を文字として扱う状態方程式が得られることで、回路の振る舞いを回路定数を用いて数式表現でき、ある変数について、ある関数を最大化する問題について、その解を回路定数を用いて表現できる場合がある。これについて、ワイヤレス給電回路の効率を最大化する方法についてを例として、第3.4節で紹介する。

3.2 提案手法

与えられた回路の挙動を記述する回路方程式は、KVL(Kirchhoff's Voltage Law)とKCL(Kirchhoff's Current Law)、そして素子の電気的特性(たとえば、素子が抵抗ならば、オームの法則)から構成される。

今、回路が抵抗やインダクタ、キャパシタおよび電圧源から構成される状況を考える。このとき、回路方程式は、線形微分方程式と線形代数方程式から記述される。

回路方程式は、回路の状態を表す物理量—各部の電圧、電流—を含む。このうち、抵抗の電圧・電流、キャパシタ・電源の電流、インダクタ・ノードの電圧は冗長な変数である。なぜならば、これらは、独立な変数であるキャパシタの電圧、インダクタの電流および、電源電圧の線形結合で表すことができるからである。

本稿では、この冗長な変数を含む回路方程式を 冗長な回路方程式と呼ぶことにする。また、独立変数のみから記述された回路方程式を、冗長ではない回路方程式と呼ぶことにする。あとでわかるように、冗長ではない回路方程式から、式変形によって、状態方程式を導くことができる。この冗長な回路方程式から、状態方程式を得るためのコンピュータ

上のアルゴリズムを、次のように提案する。

冗長な回路方程式中の線形微分方程式は、次のように書くことができる。

$$z = F\dot{x} \quad (3.2.1)$$

ここに、 z はインダクタの電圧、キャパシタの電流を並べたベクトルであり、 x は、インダクタの電流、キャパシタの電圧を並べたベクトルである。行列 F は x に応じて決定される。また、冗長な回路方程式中の線形代数方程式から、抵抗の電圧・電流、電源の電流、ノードの電圧を消去すると、(3.2.2) のように書くことができる。

$$K_1 z + K_2 x + K_3 u = 0 \quad (3.2.2)$$

ここに、 z は、インダクタおよびトランスの電圧、キャパシタの電流を表し、 u は電源の電圧を表す。 K_1, K_2, K_3 はそれぞれの係数行列である。

また、行列 $Q(r \times s)$ を次のように定義する。

$$Q = [K_1 \ K_2 \ K_3] \quad (3.2.3)$$

ここで、 $\text{rank } Q = r$ ならば、(3.2.2) を次のように変形できる。

$$z = -(K_1^T K_1)^{-1} K_1^T (K_2 x + K_3 u) \quad (3.2.4)$$

また、 $\text{rank } Q \neq r$ の時、 Q から 1 次従属な行を取り除くことによって、 $\text{rank } Q = r$ の形へ Q を変形できる。

さらに、(3.2.1) を用いて、 $\det F \neq 0$ の時、(3.2.4) を下記のように書き直すことができる。

$$\dot{x} = -F^{-1} (K_1^T K_1)^{-1} K_1^T (K_2) x - F^{-1} (K_1^T K_1)^{-1} K_1^T (K_3) u \quad (3.2.5)$$

ここで、

$$A = -F^{-1} (K_1^T K_1)^{-1} K_1^T (K_2) \quad (3.2.6)$$

$$B = -F^{-1} (K_1^T K_1)^{-1} K_1^T (K_3) \quad (3.2.7)$$

とおけば、(3.2.5) は、

$$\dot{x} = Ax + Bu \quad (3.2.8)$$

となり、我々が求めたい、冗長ではない回路方程式が得られた。この方程式は、状態方程式とも呼ばれる。

図 3.1 に、提案手法をソフトウェア実装を考慮し、フローチャートに図示した。

一般に、文字の逆行列計算は、行列のサイズが大きくなると実時間以内終了することが難しくなる。従って、本稿では、冗長な回路方程式から抵抗の電圧・電流、電源の電流、ノードの電圧を消去する過程において、図 3.1 の 1~4 にあげる工夫をして、文字を含む形での逆行列計算を避けている。これにより、逆行列演算が必要になるのは、最後の行列 A, B を求める過程のみとなり、逆行列演算が少なくてすむような手法になっている。

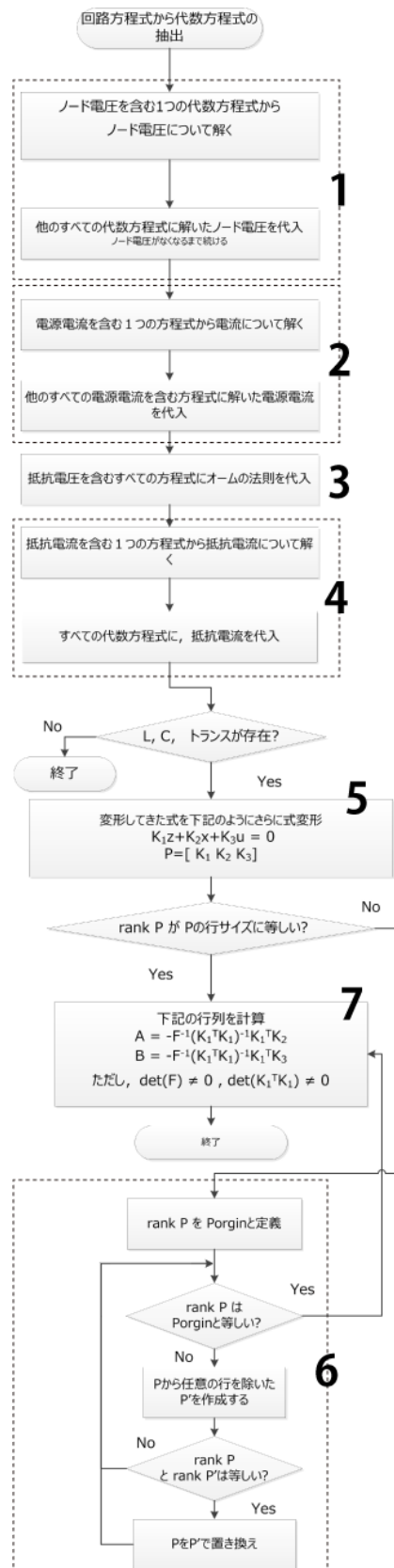


図 3.1: 回路方程式から状態方程式へのシンボリック変換

3.3 提案手法の適用例

図 3.1 に示したアルゴリズムを, 図 3.2 に示される実際の回路に適用した例を紹介する。

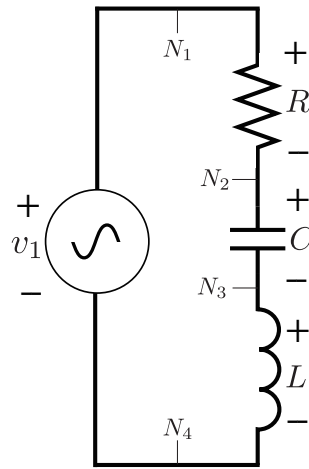


図 3.2: LCR 回路

この回路において, 完全な回路方程式は次のように書かれる。

KVL:

$$v_R = v_{N1} - v_{N2} \quad (3.3.1)$$

$$v_L = v_{N2} - v_{N3} \quad (3.3.2)$$

$$v_C = v_{N3} - v_{N4} \quad (3.3.3)$$

$$v_1 = v_{N1} - v_{N4} \quad (3.3.4)$$

KCL:

$$i_{v_1} - i_R = 0 \quad (3.3.5)$$

$$i_R - i_L = 0 \quad (3.3.6)$$

$$i_L - i_C = 0 \quad (3.3.7)$$

$$i_C - i_{v_1} = 0 \quad (3.3.8)$$

各素子の電気的特性:

$$v_R = Ri_R \quad (3.3.9)$$

$$z = F\dot{x} \quad (3.3.10)$$

ここで,

$$z = [v_L \ i_C]^T, \quad F = \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix}, \quad x = [i_L \ v_C]^T \quad (3.3.11)$$

である。

最初に 図 3.1 の 1 を (3.3.1) - (3.3.4) に適用する。

KVL:

$$v_1 = v_R + v_L + v_C \quad (3.3.12)$$

次に、図 3.1 の 2 を (3.3.5) - (3.3.8) に適用する。

KCL:

$$i_C - i_R = 0 \quad (3.3.13)$$

$$i_R - i_L = 0 \quad (3.3.14)$$

$$i_L - i_C = 0 \quad (3.3.15)$$

図 3.1 の 3 を (3.3.12) に適用する。

KVL:

$$v_1 = Ri_R + v_L + v_C \quad (3.3.16)$$

図 3.1 の 4 を (3.3.12) に適用する。

KCL:

$$i_C - i_L = 0 \quad (3.3.17)$$

$$i_L - i_C = 0 \quad (3.3.18)$$

KVL:

$$v_1 = Ri_C + v_L + v_C \quad (3.3.19)$$

図 3.1 の 5 を (3.3.17)-(3.3.19) に適用する。

$$P = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & -R & 0 & -1 & 0 \end{bmatrix} \quad (3.3.20)$$

$\text{rank } P = 3 \neq 2$ であるから、図 3.1 の 6 を行列 P に適用する。

$$P = \begin{bmatrix} 0 & -1 & 1 & 0 & 0 \\ -1 & -R & 0 & -1 & 0 \end{bmatrix} \quad (3.3.21)$$

最後に、行列 P と図 3.1 の 7 を用いて、行列 A , B を計算することで、この回路の状態方程式を導けた。

$$A = \begin{bmatrix} -\frac{R}{L} & \frac{-1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \quad (3.3.22)$$

ここで、

$$K_1 = \begin{bmatrix} 0 & -1 \\ -1 & -R \end{bmatrix}, K_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, K_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.3.23)$$

であり、 v_N はノードの電位を表す。

3.4 提案手法を使用した回路の解析例—ワイヤレス給電の送電効率を最大にする数式表現

ワイヤレス給電は、無線で電気エネルギーを送電するシステムの総称である。その主立った研究は、送電効率や給電の出力の向上を目指すものであり、これらの向上のためには、一般に、回路定数(たとえば、コイルの巻き数や形状)や回路トポロジー、および電源の周波数を適宜調整しなければならない。

本節では、図 3.3 のワイヤレス給電回路における、送電効率を最大にする電源周波数についての設計手順 [1] [2] を紹介し、その数式表現を導くことにする。

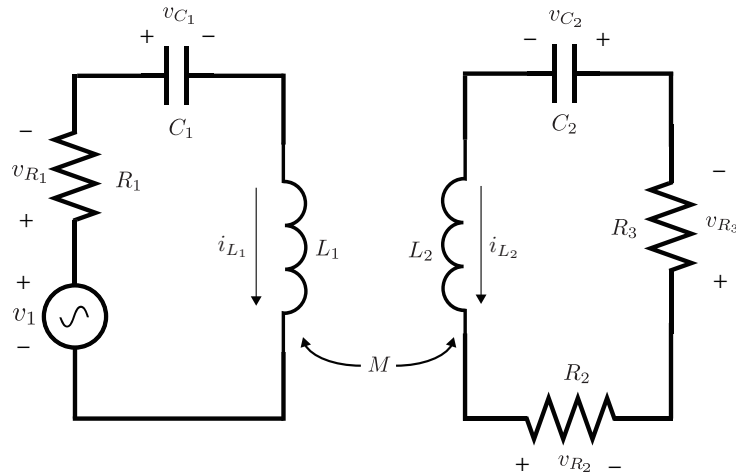


図 3.3: ワイヤレス給電回路

L_1, L_2 は、それぞれ送電・受電コイル [H] であり、 M はコイル L_1, L_2 の相互インダクタンス [H] を表し、 R_1, R_2 は、それぞれ一次側・二次側の寄生要素 [Ω] を、 R_3 は負荷抵抗 [Ω] を表す。電源としては、振幅 a [V]、角周波数 ω [rad/s] の正弦波電圧源 $v_1 = a \sin \omega t$ [V] を用いる (t は時間 [s] を表す)。

送電効率(あるいは、単に効率とよぶ)は、次のように定義する。

$$\eta = \frac{\bar{P}_2}{\bar{P}_1} \quad (3.4.1)$$

ここで、 \bar{P}_2 は二次側負荷 R_3 の平均消費電力 [W]、 \bar{P}_1 は電源の平均供給電力 [W] である。

効率を最大にする電源周波数を求めるために、この回路の状態方程式を計算する。これは、提案手法を実装したソフトウェア「Wasabi」によって、回路図をコンピュータ上で描くだけで、求めることができる。

$$\dot{x} = Ax + Bu$$

ただし、

$$A = \begin{bmatrix} 0 & 0 & \frac{1}{C_1} & 0 \\ 0 & 0 & 0 & \frac{1}{C_2} \\ \frac{-L_2}{(L_2L_1-M^2)} & \frac{M}{(L_2L_1-M^2)} & \frac{-R_1L_2}{(L_2L_1-M^2)} & \frac{(R_2+R_3)M}{(L_2L_1-M^2)} \\ \frac{M}{(L_2L_1-M^2)} & \frac{-L_1}{(L_2L_1-M^2)} & \frac{R_1M}{(L_2L_1-M^2)} & \frac{(-R_2-R_3)L_1}{(L_2L_1-M^2)} \end{bmatrix} \quad (3.4.2)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{L_2}{(L_2L_1-M^2)} \\ \frac{-M}{(L_2L_1-M^2)} \end{bmatrix} \quad (3.4.3)$$

$$x = [v_{C1} \quad v_{C2} \quad i_{L1} \quad i_{L2}]^T \quad (3.4.4)$$

$$u = v_1 \quad (3.4.5)$$

である。この行列 A , B から、平均電力 \bar{P}_1 , \bar{P}_2 を計算することができる。

はじめに、電源が正弦波関数であるから、この回路の定常状態の応答は、式 (2.2.6) で表すことができる。この式 x_{ss} を、便宜的に

$$x_{ss}(t) = \alpha \cos \omega t + \beta \sin \omega t$$

と書いておく。ただし、

$$\alpha = -a\omega(\omega^2 I + A^2)^{-1}B \quad (3.4.6)$$

$$\beta = -aA(\omega^2 I + A^2)^{-1}B \quad (3.4.7)$$

I は単位行列である。

平均電力の定義式は、周期 T , また、 $P(t)$, \bar{P} をそれぞれ瞬時、平均電力として

$$\bar{P} = \frac{1}{T} \int_0^T P(t) dt$$

である。したがって、

$$\bar{P}_1 = \frac{1}{T} \int_0^T v_1(t) i_{L1}(t) dt = \frac{1}{T} \int_0^T a \sin \omega t (\alpha_3 \cos \omega t + \beta_3 \sin \omega t) dt$$

$$\bar{P}_2 = \frac{1}{T} \int_0^T R_3 i_{L2}^2(t) dt = \frac{1}{T} \int_0^T R_3 (\alpha_4 \cos \omega t + \beta_4 \sin \omega t)^2 dt$$

となる。ただし、 α と β の添え字は、各々のベクトルの行方向の成分を表す。これを計算すれば、

$$\bar{P}_1 = \frac{\beta_3}{2} a \quad (3.4.8)$$

$$\bar{P}_2 = \frac{\alpha_4^2 + \beta_4^2}{2} R_3 \quad (3.4.9)$$

が得られる。

式 (3.4.8)-(3.4.9) は、さらに計算を進めることができるが、複雑で長くなるので省略する。

平均電力 \bar{P}_1 , \bar{P}_2 が計算できたので、効率の計算をさらに進めることができ、(3.4.1) は、

$$\eta = \frac{C_2^2 M^2 R_3 \omega^4}{C_2^2 M^2 (R_2 + R_3) \omega^4 + R_1 (1 - 2C_2 L_2 \omega^2 + C_2^2 \omega^2 (R_2^2 + 2R_2 R_3 + R_3^2 + L_2^2 \omega^2))} \quad (3.4.10)$$

と書ける。

この式をもとに、効率の最大を与える電源周波数が計算できる。それにあって、

$$\frac{\partial \eta}{\partial \omega} = 0$$

を満たす ω について、 $\omega > 0$ のもとに解くと、

$$\omega_{\text{opt}} = \sqrt{\frac{2}{C_2 L_2 - C_2^2 (R_2 + R_3)^2}} \quad (3.4.11)$$

となる。ただし、 $C_2 L_2 - C_2^2 (R_2 + R_3)^2 > 0$ 。このときの、効率を η_{max} とすれば、

$$\eta_{\text{max}} = \frac{4M^2 R_3}{(R_2 + R_3)(4M^2 + C_2 R_1 (R_2 + R_3)(4L_2 - C_2 (R_2 + R_3)^2))} \quad (3.4.12)$$

となる。この極値について調べると次の増減表から、

表 3.1: 増減表

ω	\cdots	ω_{opt}	\cdots
$\frac{\partial \eta}{\partial \omega}$	+	0	-
η	\nearrow	η_{max}	\searrow

η_{max} は極大値であることがわかる。つまり、この給電回路が極値を持つとき、効率の最大を与える電源周波数は、(3.4.11) であり、そのときの最大効率は、(3.4.12) となる。回路定数を文字として扱った状態方程式を用いれば、いわゆるシステム論的なアプローチが可能となり、考えている問題の解を文字式として定式化可能となる一例を紹介した。

さらに、ここから考察を進めることもできる。例えば、(3.4.10) と (3.4.12) から、効率および最大効率は C_1 の容量によらない (支配されない) こと、 $R_2 + R_3 \approx R_3$ がかつ、 C_2 の容量が小さければ小さいほど、その最大効率は 100% に近づく、ということがわかる。いずれも、文字式で現象を記述することができたことから、はじめて正確にわかるものである。

3.5 おわりに

提案手法によって、回路定数を文字として扱った状態方程式が生成することができれば、電源の角周波数について、ワイヤレス給電の効率を最大にする数式表現が、回路定数を文字として扱った文字式で表現できる例を紹介した。また、その文字式を用いて、回路の振る舞いを支配するパラメータの有無の把握や、期待通りの振る舞いを設計するためのパラメータに関する考察も可能になることも示した。

ワイヤレス給電は、回路のトポロジーによって、一般に、給電効率・電力が異なる。例えば、どの回路トポロジーがより高出力の給電を可能にするかは、実際に計算を進めてみなければわからない。また、スイッチング回路は、そのスイッチによって回路トポロジーが時々刻々と変化するために、必要となる数式モデルは一般に複数になる。

この章で紹介したように、システム論的なアプローチによって、これらの回路の振る舞いの解析・設計を、試みる時、トポロジーごとに数式モデル (状態方程式や伝達関数) が必要になるのである。そのような問題を考えるとき、提案手法を実装したソフトウェアは、多少なりとも助けになるであろう。

3.A ワイヤレス給電の受電・送電コイルのモデル

ワイヤレス給電のように、複数のコイル(インダクタ)があって、それらが互いに磁気のやりとりを行う場合の、コイルの電圧・電流特性のモデルを記す。

図3.4にあるとおり、1次側の送電コイルに対して、2次側に受電コイルが $n-1$ 個存在する場合を考える。

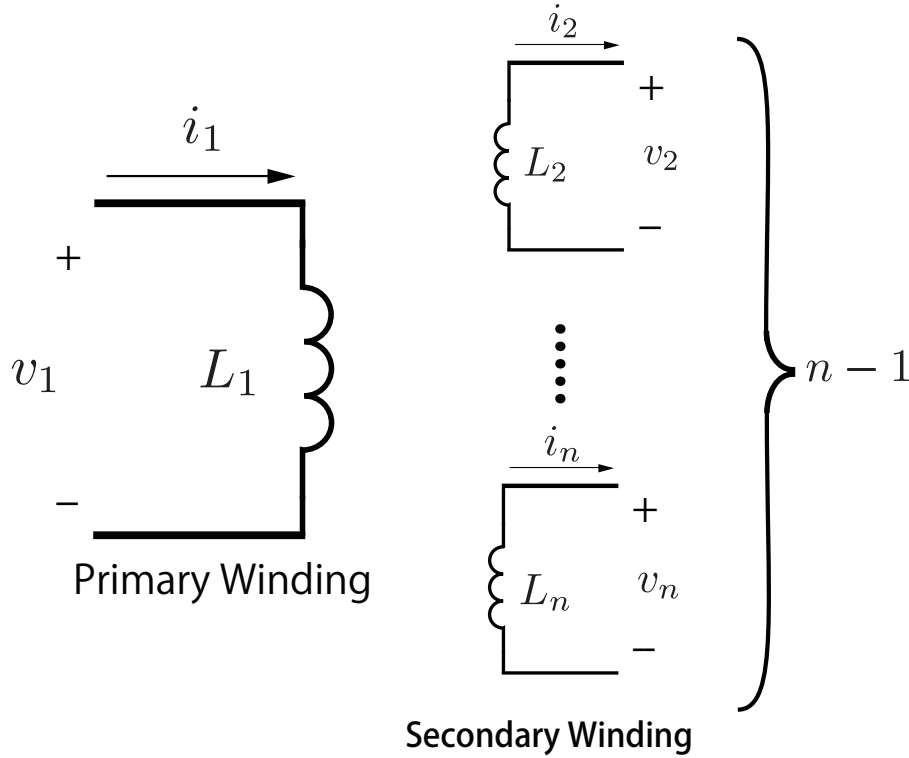


図 3.4: ワイヤレス給電のコイルの電圧・電流特性のモデル

このとき、1次側2次側のそれぞれの電圧・電流の関係式は、(3.A.1)のように書かれる。

$$v = \begin{bmatrix} L_1 & \pm M_{12} & \pm M_{13} & \dots & \pm M_{1n} \\ \pm M_{21} & L_2 & \pm M_{23} & \dots & \pm M_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \pm M_{n1} & \pm M_{n2} & \dots & \pm M_{nn-1} & L_n \end{bmatrix} i \quad (3.A.1)$$

ただし、 v, i はそれぞれ $v = [v_1 \ v_2 \ \dots \ v_n]^T$, $i = [i_1 \ i_2 \ \dots \ i_n]^T$ である。

相互インダクタンス M の前の符号は、1次側巻き線と2次側巻き線の巻き方向が同じ場合を正とする。

また、Neumann's formula より、相互インダクタンス M は、下記の式で表すことができる。

$$M_{ij} = \frac{\mu_0}{4\pi} \oint_{C_i} \oint_{C_j} \frac{dl_i \cdot dl_j}{r_{ij}} \quad (3.A.2)$$

ただし、 $1 \leq i \leq n, 1 \leq j \leq n$ で、 $i = j$ の時、 M_{ij} は自己インダクタンス L_i を表す。 C_i, C_j は、それぞれ、巻き線 i , 巻き線 j の周の経路を表し、 dl_i, dl_j は、それぞれ、巻き線 i , 巻き線 j の微小のベクトルを表し、 r_{ij} は、それぞれ、巻き線 i , 巻き線 j の距離を表す。

つまり、相互インダクタンス M は、巻き線 i, j 間の距離や巻き線の形状 (巻き数) などの要素で決定される。

また、相互インダクタンスの相反定理 [3] から、

$$M_{ij} = M_{ji}$$

である。

参考文献

- [1] Kazuya Yamaguchi, Yuta Yamamoto, Takuya Hirata, Eko Setiawan and Ichijo Hodaka. *Mathematical Expression of Optimal Frequencies for Wireless Power Transfer*, The 3rd International Conference on Computer Engineering and Mathematical Sciences (ICCEMS 2014), pp.826-827, 2014.
- [2] 山口和也, 高効率・高出力ワイヤレス給電技術へのシステム論的アプローチ, 2016年
- [3] ファインマン, レイトン, サンズ. ファインマン物理学, 電磁気学, 3巻. 岩波書店, 1986年.

CHAPTER

4

回路の数式モデルの妥当性の検証方法について

4.1 はじめに

現実の回路の振る舞いのシミュレーションは、その回路を理想化したモデルに基づいて行われることについては、第1章で述べた。そのモデルとは、回路図とそれと等価な回路方程式から導かれる伝達関数モデルや状態方程式モデルである。

一方で、回路図には、現実の回路にはない自由度があるために、電気回路の法則(キルヒホッフの法則、オームの法則、キャパシタ・インダクタの電圧・電流特性)に反する回路を描くことができてしまい、それが結果として、現実の回路の振る舞いを正しくシミュレーションできない状況を招くことがある。

この章では、回路図と等価である回路方程式をもとに、数学的に電気回路の法則に従う回路の条件が定義され、それをもとに、回路方程式がその法則と矛盾しないか、ひいては、現実の回路を正しくモデリングできているかどうかを調べる判別方法が提案される [1]。スイッチング回路のようにトポロジーが時々刻々と変化する複雑な回路の場合、あるトポロジーにおけるモデリングは、電気回路の法則に従うが、あるトポロジーは、その法則に反する例があることを、提案する判別法を用いながら紹介する。

4.2 電気回路の法則と回路方程式

与えられた回路図に対して、キルヒホッフの電圧則・電流則やオームの法則、そして、キャパシタやインダクタの電圧・電流に関する電氣的な特性によって、回路方程式が定義される。これらの法則や特性のことを電気回路の法則と呼ぶ。ここで、電気回路の法則における、すべての変数は、すべての時刻 t で定義される一価関数と仮定される。つまり、一般に回路方程式は、

$$Aw + By + Cz = 0 \quad (4.2.1)$$

$$H(x(t) - x(t_0)) = \int_{t_0}^t z_1(p) dp \quad (4.2.2)$$

と書かれ、ここで、 A, B, C, H は、電気回路の法則から構成される定数の係数行列であり、

$$x = \begin{bmatrix} i_L \\ v_C \end{bmatrix}, u = \begin{bmatrix} v_V \\ i_I \end{bmatrix}, w = \begin{bmatrix} u \\ x \end{bmatrix}, y: \text{ノード電位ベクトル}, z_1 = \begin{bmatrix} v_L \\ i_C \end{bmatrix}, z = \begin{bmatrix} i_V \\ v_I \\ v_R \\ i_R \\ z_1 \\ v_S \\ i_S \end{bmatrix} \quad (4.2.3)$$

また、 $v_V, v_I, v_C, v_L, v_R, v_S$ はそれぞれ、電圧源・電流源・キャパシタ・インダクタ・抵抗・スイッチの電圧であり、同様に $i_V, i_I, i_C, i_L, i_R, i_S$ はそれぞれ、電圧源・電流源・キャパシタ・インダクタ・抵抗・スイッチの電流を表す。 v_N は、ノードの電位である。また、 t, t_0 は時刻を表す。

本章では、すべての変数は、リーマン可積分と仮定する。つまり、式(4.2.2)中の \int は、リーマン積分を表す(詳しくは、例えば、[3]を参照)。

4.2.0.2 インダクタ・キャパシタの電圧・電流の特性

式(4.2.2)は、インダクタ・キャパシタの電圧・電流特性についての式であるが、それについて、重要な事項を補足する。まず、式(4.2.2)を、インダクタ・キャパシタごとにそれぞれ書きくれば、次の通りである。

$$L(i_L(t) - i_L(t_0)) = \int_{t_0}^t v_L(p) dp \quad (4.2.4)$$

$$C(v_C(t) - v_C(t_0)) = \int_{t_0}^t i_C(p) dp \quad (4.2.5)$$

L, C は、それぞれ、インダクタンス、キャパシタンスを表す。

ここで、積分はリーマン積分と仮定した。このため、 $v_L(t), i_C(t)$ はリーマン可積分関数でなければならない。よくスイッチング回路で見られるような、区分的連続な関数(波形)は、リーマン積分可能な関数である。また、リーマン積分の不定積分は、連続関数である。すなわち、少なくとも、 i_L, v_C は連続でなければならない。言い換えると、 i_L, v_C は、飛び飛びの値をとらない。

回路図と回路方程式の例

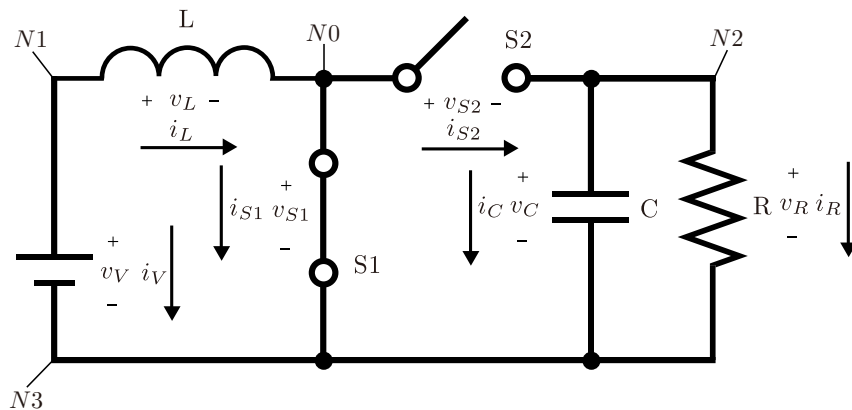


図 4.1: Boost converter - CCM ($S1$: ON, $S2$: OFF)

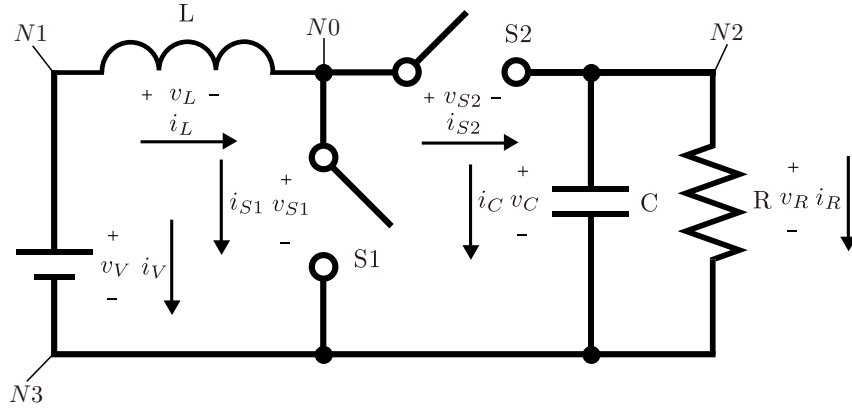
図 4.2: Boost converter - DCM ($S1$: OFF, $S2$: OFF)

図 4.1 と 4.2 は, Boost converter [4] と呼ばれる DC/DC コンバータの一つであり, それぞれ, continuous conduction mode (CCM) と discontinuous conduction mode (DCM) と呼ばれるモード時の回路図を表す。

図 4.1 における回路方程式を, (4.2.1) と (4.2.2) 表記で記述すれば,

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -R & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2.6)$$

$$H = \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix}, x = \begin{bmatrix} i_L \\ v_C \end{bmatrix}, z_1 = \begin{bmatrix} v_L \\ i_C \end{bmatrix}, w = \begin{bmatrix} v_V \\ i_L \\ v_C \end{bmatrix}, y = \begin{bmatrix} v_{N1} \\ v_{N3} \\ v_{N0} \\ v_{N2} \end{bmatrix}, z = \begin{bmatrix} i_V \\ v_R \\ i_R \\ v_L \\ i_C \\ v_{S1} \\ i_{S1} \\ v_{S2} \\ i_{S2} \end{bmatrix} \quad (4.2.7)$$

となる。

同様に，図 4.2 の回路方程式は，

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -R & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2.8)$$

であり，そのほかの表記は，(4.2.7)に従う。

図 4.2 について，回路方程式の代数方程式部分 (4.2.8) において，インダクタ電圧 v_L は，任意の v_V, i_L, v_C に対して，一意に決定しないことが，この回路図からわかる。回路方程式の代数方程式部分 (4.2.1) の定数の係数行列部分 A, B, C から，行列の Rank 計算によって，上述の状況を捉える，新しい定理を紹介する。

4.3 回路方程式における代数方程式の解の存在と一意性

ある回路における，任意の w に対して，回路方程式の代数方程式の解 (y, z) が存在して，かつ， z が一意であるとき，その回路をプロパー (proper) と呼ぶ。プロパーでない回路を，インプロパー (improper) と呼ぶ。

定理 4.3.1. ある回路がプロパーであるための必要十分条件は，その回路の回路方程式 (4.2.1) における，代数方程式の係数行列が

$$\text{rank}[A \ B \ C] = \text{rank}[B \ C] = \text{rank } B + m.$$

を満たすときである。ここで， m は，行列 C の列数を表す。

定理 4.3.1 の証明は，付録 4.A に記す。定理 4.3.1 にある Rank 計算は，汎用数式処理ソフトウェアで計算することができる。すなわち，回路がプロパーであるか，インプロパーであるかを，コンピュータを使用して確かめることができる。このことは，回路の規模が大きい場合や複雑な場合などのように，手計算や見た目から判断することが困難な場合において，コンピュータ支援が得られることを意味する。

プロパーな回路の例

Boost Converter (CCM) 例として，図 4.1 の回路において，定理 4.3.1 を使用して，プロパーな回路であるかどうかを調べよう。式 (4.2.6) から

$$\text{rank}[A \ B \ C] = 12, \text{rank}[B \ C] = 12, \text{rank } B = 3, m = 9$$

とわかる。定理 4.3.1 から，この回路はプロパーな回路である。

インプロパーな回路の例

Boost Converter (DCM) 図 4.2 の回路において、同様に、プロパーな回路であるかどうかを調べよう。式 (4.2.8) から、

$$\text{rank}[A \ B \ C] = 12, \text{rank}[B \ C] = 11, \text{rank} \ B = 3, m = 9$$

とわかる。定理 4.3.1 から、この回路はインプロパーな回路である。

ある回路がプロパーな回路であるとき、その回路の振る舞い(回路中の各部の電圧・電流)は、付録 4.A の系 4.A.1 によって与えられる。したがって、図 4.1 は、回路を、すべての i_L , v_C を状態変数として、状態方程式モデルで表すことができる。

ところが、図 4.2 の回路にあるようなインプロパーな回路は、 i_L を状態変数として、状態方程式を書くことができない。

このことについて、もう少し考察を進める。

4.4 モデリングの矛盾

図 4.1 に示される CCM から、図 4.2 に示される DCM に回路のトポロジーが遷移したとしよう。ここで、DCM では、 $i_L = 0$ に拘束されることに注意する。

回路の動作が、CCM から DCM に遷移する際、 $i_L \neq 0$ である場合、すなわち、 i_L の時間についての左極限值 (CCM から DCM へ近づけた) と右極限值 (CCM から DCM へ近づけた) が一致しないから、 i_L は連続関数ではなくなる。

ところが、これは、インダクタの電圧・電流特性に反する。もはやこの L はインダクタの性質を示さない。もし、 $i_L = 0$ である場合は、 i_L は連続関数であるから、インダクタの性質を保つ。

同様なことは、個々では紹介しないが、別の例を考えれば、キャパシタについても起こる得ることである。

このように、物理的なキャパシタおよびインダクタの性質と回路方程式が矛盾する可能性の原因の一つとして、モデリングにおいて、現実の回路を理想化を行いすぎた結果により発生した矛盾と考えられる。

4.5 おわりに

ある回路図について、電気回路の法則に矛盾しないかどうかの判別が、回路方程式の係数行列を調べることによって可能になった。この提案された判別法は、数値回路解析およびシンボリックな回路解析問わず利用でき、判別法によってモデルがプロパーな回路であると明らかになったとき、モデルが現実の回路を正しく模擬できていることを、電気回路の法則に矛盾しないという点で保証するものである。

4.A 本章で省略された補題, 系, 証明

系 4.A.1. ある回路がプロパーである時, その回路の回路方程式 (4.2.1) と (4.2.2) の解を, 一意に

$$z = K_1 x + K_2 u \quad (4.A.1)$$

$$x(t) = e^{F(t-t_0)} x(t_0) + e^{Ft} \int_{t_0}^t e^{-Fp} G u(p) dp \quad (4.A.2)$$

と表すことができる。

証明 4.A.1. 定理 4.3.1 から, z が x, u の線形結合で書かれること, すなわち (4.A.1) を得る。ここで, 行列 K は, 適当な係数行列を表す。これは, $z_1 = K_{11}x + K_{21}u$ を含む。行列 H は, ブロック対角であり, そのブロックの要素は, キャパシタンス C ・自己インダクタンス L ・相互インダクタンス M からなり, 正則である。

$F = H^{-1}K_{11}$, $G = H^{-1}K_{21}$ と置くと, 部分積分 [3] と行列指数関数から, 式 (4.A.2) を得る。□

式 (4.A.2) は, $x(t)$ が微分可能であるということを仮定せずに, 得ることができることを強調しておく。

補題 4.A.1. 行列 B, C が, 共通の行数を持つとする。任意の y について, 方程式 $By + Cz = 0$ が解 z を持つための必要十分条件は, $\text{rank}[B \ C] = \text{rank } C$ である。

証明 4.A.2.

(必要条件) $B = CZ$ を満たす, 行列 Z が存在する。また, $[B \ C] = [CZ \ C] = C[Z \ U_m]$, ここで, U_m は単位行列を表す。 $[Z \ U_m]$ が行フルランクであるから, ゆえに, $\text{rank}[B \ C] = \text{rank } C$ である。□

(十分条件) $\text{rank}[B \ C] = \text{rank } C$ と $\text{Im } [B \ C] \supset \text{Im } C$ から, $\text{Im } [B \ C] = \text{Im } C$ である。したがって, $\text{Im } B \subset \text{Im } [B \ C] = \text{Im } C$ である。つまり, y が任意に与えられたとき, $By + Cz = 0$ を満たす z が存在する。□

証明 4.A.3 (定理 4.3.1 の証明).

(必要条件) 仮定と補題 4.A.1 から $\text{rank}[A \ B \ C] = \text{rank}[B \ C]$ である。 $w = 0$ とおく。式

(4.2.1) の解空間 Q は, $Q = \text{Ker } [B \ C]$ である。 $\begin{bmatrix} y_1 \\ z_1 \end{bmatrix} \in Q$ とし, $y_2 \in \text{Ker } B$ とする。

$\begin{bmatrix} y_2 \\ 0 \end{bmatrix} \in Q$ から, $z_1 = 0$ である。したがって, $\dim(\text{Ker } [B \ C]) = \dim(\text{Ker } B)$ である。次元定理から,

$$\text{rank}[B \ C] = \text{rank } B + m$$

が導かれる。□

(十分条件) $Q(w) = \left\{ \begin{bmatrix} y \\ z \end{bmatrix} \mid Aw + By + Cz = 0 \right\}$ とおく。補題 4.A.1 から, 任意の w が与えられたとき, $Q(w) \neq \phi$ である。 $\text{rank}[B \ C] \leq \text{rank } B + \text{rank } C$ と一般に $\text{rank } C \leq m$ から, $\text{rank}[B \ C] = \text{rank } B + m$ は, $m = \text{rank}[B \ C] - \text{rank } B \leq \text{rank } C \leq m$ である。つまり,

$$\text{rank } C = m = \text{rank}[B \ C] - \text{rank } B. \quad (4.A.3)$$

一般に成立する等式 $\text{rank}[B \ C] = \text{rank } B + \text{rank } C - \dim(\text{Im } B \cap \text{Im } C)$ [5] と (4.A.3) から

$$\dim(\text{Im } B \cap \text{Im } C) = 0. \quad (4.A.4)$$

が示される。 $\begin{bmatrix} y_1 \\ z_1 \end{bmatrix}, \begin{bmatrix} y_2 \\ z_2 \end{bmatrix} \in Q(w)$ とする。すなわち, $By_1 + Cz_1 = By_2 + Cz_2$ が成り立つ。
(4.A.4) から, $C(z_1 - z_2) = 0$ である。式 (4.A.3) の最初の等式から, $z_1 = z_2$ となる。 \square

参考文献

- [1] Takuya Hirata, Eko Setiawan, Kazuya Yamaguchi, and Ichijo Hodaka. A Criterion on Existence and Uniqueness of Behavior in Electric Circuit. *International Journal of Electrical and Computer Engineering (IJECE)*, 6(4):1529-1533, August 2016.
- [2] D. A. Calahan, *Computer-Aided Network Design*, Revised Edition. McGraw-Hill, 1972.
- [3] K. Knopp. *Theory and Application of Infinite Series*. Blackie, 1951.
- [4] R. W. Erickson and D. Maksimović. *Fundamentals of Power Electronics*, Second Edition. Kluwer, 2004.
- [5] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*, Second Edition. Academic Press, 1985.

CHAPTER

5

おわりに

第3章では、与えられた回路の振る舞いの数式モデルとして、その回路のパラメータを記号とし、状態方程式を、コンピュータ上で生成するアルゴリズムについて述べた。それは、従来の方法と比較して、“適切な木”を見つけること無しに行われ、また、状態方程式が生成される過程に必要な逆行列演算が少なくなるようなアルゴリズムである。

また、提案されたアルゴリズムは、コンピュータソフトウェア Wasabi として実装され、業界標準の SPICE ネットリスト (回路図接続関係をテキストで表現したもの) との互換性や、生成された数式モデルがシームレスに汎用数式処理ソフトウェアにエクスポートできるようになっている。すなわち、コンピュータ上で、回路図を描くだけで、その回路図に対応する数式モデルが容易に生成される。

これにより、第3章の小節で示したような回路の振る舞いや、ある変数について、ある関数を最大化する問題に対する解について、回路パラメータによる数式表現の機会が、従来よりも比較的容易に考えられるようになった。また、一般に、DC/DC コンバータ回路の振る舞いを解析する際は、ひとつの回路で多数のトポロジーに切り替わって動作する回路の、トポロジーごとの数式モデルが必要になるが、3章で提案した手法により、コンピュータ上で回路図を描くだけで数式モデルが生成できることが利点である。

第4章では、現実の回路をモデリングする際、現実の回路がもつ振る舞いの存在と一意性が、モデリングの回路図やそれと等価な回路方程式に継承されているかを判定する方法について述べた。回路方程式の係数行列の Rank 演算のみで判定可能であり、Rank 演算は汎用数式処理ソフトウェアに備わっていることから、比較的容易に確認できる。判定法は回路の数値回路解析やシンボリック回路解析の双方に使うことができる。

また、回路の振る舞いをコンピュータ使って模擬する場合に、その回路が電気回路の法則に反していないか、すなわち、現実の回路を正しくモデリングできているかを保証するものである。回路方程式を解くことで、回路の振る舞いは求められるが、一般に、シンボリック計算では、数値計算で行う場合よりも多くの時間が必要である。

ここで問題なのは、もし、モデリングの結果である回路図あるいは、それと等価な回路方程式に、電気回路の法則に反するような矛盾があっても、解を求めるための計算が行われることである。これは、もちろん「意味のない計算」を行っていることになる。

判定法を使用して、モデリングの結果である回路図を調べておき、電気回路の法則に反する矛盾がないことを確認しておくことで、「意味のある計算」を行うことができ、計算時間を無駄にせずすむのである。

謝辞

本稿を執筆するにあたって多くの人によって支えられた。

指導教員である穂高 一条教授には、約6年近くお世話になった。研究についての相談に、いつも快くのっていただいた。ゼミなどをおして、制御や数学を教わり、その制御や数学を基礎として、様々なシステムの抱える問題を定式化する方法、それから、理論とシミュレーションだけにとどまらず、実験までも考慮した、問題解決のための方法を教わった。

また、淡野 公一教授、湯浅 高志教授、飯田 雅人教授、田村 宏樹教授には、ご多忙の中、時間を割いて、論文に目を通してくださり、適切なご指摘やアドバイスを頂いた。

同じ研究室の人々にも多く助けられた。特に、同じ博士課程に所属する山本 悠太氏、山口 和也氏、Eko Setiawan 氏には、研究に対する議論やご指摘を頂いた。また、研究の一部に携わる機会もあり、それによって自身の知見を広げることができた。

本稿で参考文献に挙げた本の著者の方々には、活字を通して、論文を読む際の基礎的な知識という支えにだけにとどまらず、特に、俯瞰図のように回路解析の手法の全体を理解する上で、大いに助けられた。

最後に、大学生活を通して経済面だけではなく、あらゆる面で支えてくれた家族や祖父がいる。

この場を借りて、お世話になった方々に感謝の意を表したい。

2016年11月8日
平田 拓也

APPENDIX

A

Wasabiの使い方

Wasabiはプロパーな回路を対象として、回路の状態方程式モデルを生成できる。Wasabiを使って状態方程式を生成するには、回路に対応するSPICE ネットリストが必要である。ここでは、LTSPICE(Linear Technology 社)を用いて、回路図を描き、SPICE ネットリストを生成し、それを Wasabi に読み込ませることで、状態方程式を生成する、その例を紹介する。

A.1 Wasabiの初期設定

Wasabi を動かすのに必要なもの

- PARI/GP 2.3.4 以降 (readline version でないバイナリ)¹
- Microsoft .Net Framework 4.5²

初期設定

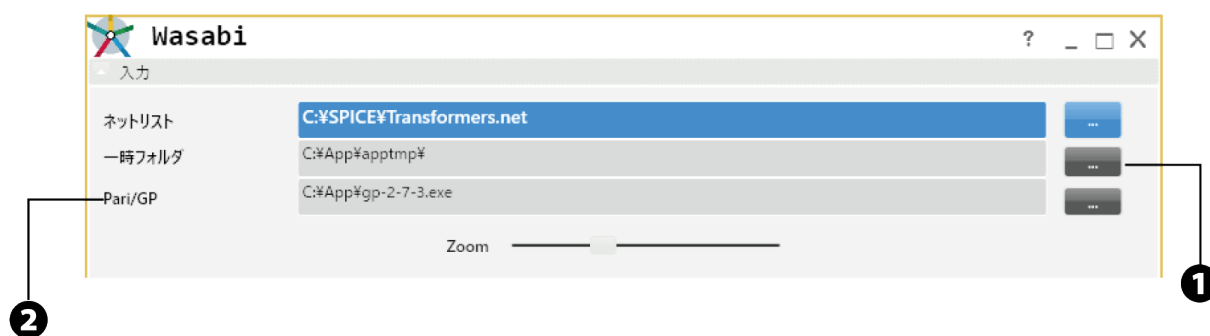


図 A.1: Wasabi の入力パネル—初期設定の仕方

日本語やスペースを用いてディレクトリ (フォルダ) を作成したり、ファイル名にそれらを使用しないことを推奨する。

1. 一時フォルダ—ここに、Wasabi が自由に使用してよいフォルダを指定する。計算の途中の結果などが入るので、新たに適切なフォルダを作ることを推奨する。

¹<http://pari.math.u-bordeaux.fr/>

²<https://www.microsoft.com/en-us/download/details.aspx?id=30653>

- 2. PARI/GP—ここに、ダウンロードしたPARI/GPのバイナリへのパスを指定する。
この操作は初回のみ必要な作業となる。

A.2 素子の電圧の極性と電流の向き of 仮定

SPICE ネットリストの素子についての記述は、次のようになる³。

素子名 ノード名1 ノード名2 素子の値

Wasabiは、ノード名1が高電圧側と仮定する。したがって、ノード1からノード2の方向が正として、電流の向きが仮定される。回路図を描いている段階で、LTSPICE上でそれを確認するのは、手順を要するが、そのために電流の向きを確認するための方法をWasabiで用意した。後ほど紹介する。

A.3 WasabiのUIの説明

LTSPICEなどで、状態方程式を生成したい回路図を描き、そのネットリストを出力し、下記の番号の手順に沿って、状態方程式を生成すれば良い。

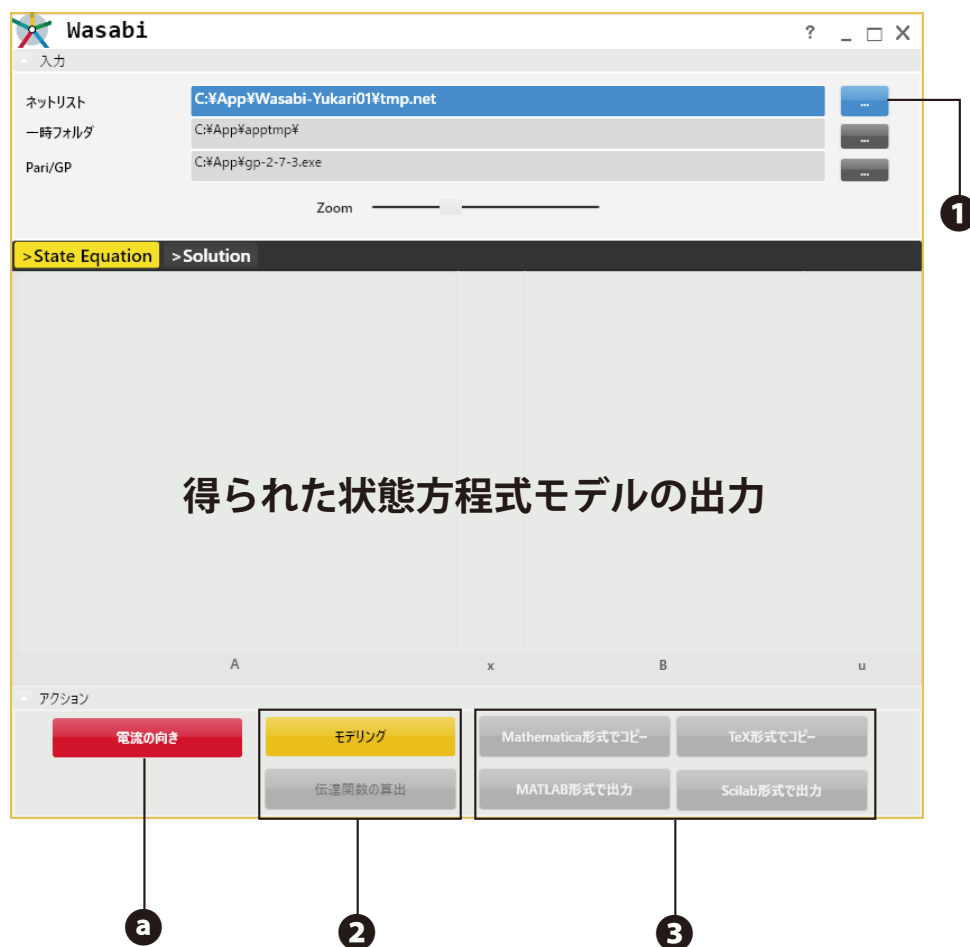


図 A.2: [>State Equation] タブ—状態方程式を算出する

³素子名には、電圧源や電流源など部品の役割を示す名前が先頭に与えられている。

- 1. 状態方程式を生成したい SPICE ネットリスト形式のファイルを読み込む
- 2. 状態方程式を生成する [モデリング] ボタン，伝達関数を生成する [伝達関数の算出] ボタン
- 3. 得られた状態方程式モデルを Mathematica/MATLAB/TeX/Scilab 形式へエクスポートする
 - a. 回路図でそれぞれ素子の電流の向きの仮定を確認する



図 A.3: [>Solution] タブで回路方程式の解を算出する

- 1. 回路の基準電位を指定し，[Solve] ボタンを押せば，回路方程式の解が出力される

A.4 状態方程式を生成する手順，回路方程式の解を算出する手順

A.4.1 LCR 回路のモデリング

LCR 回路の状態方程式を生成しよう。

1. LTSPICE で，次のように回路図を描く。

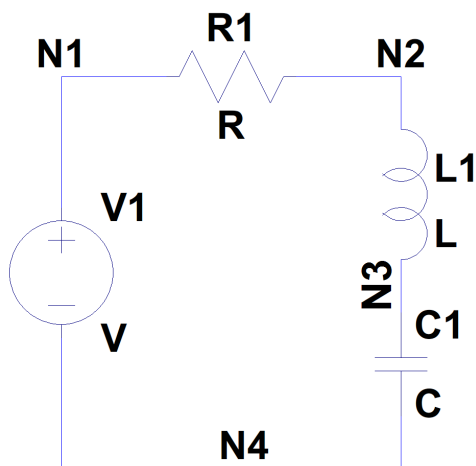


図 A.4: LTSPICE 上の LCR 回路の作成

N から始まるノード名を回路図上に与えておいた。

2. 描いた回路図を保存する。
3. 次の手順で、SPICE ネットリストを生成する。ファイル名と同名のネットリストができる。

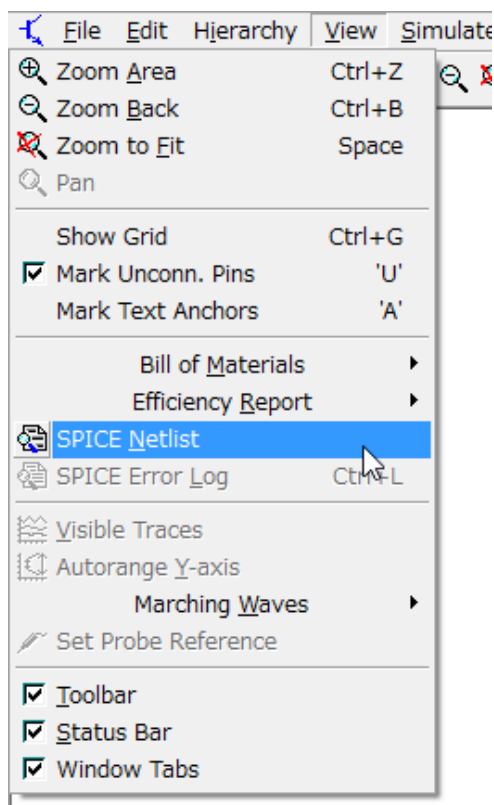


図 A.5: LTSPICE におけるネットリストの出力の仕方

生成されたネットリストの一部

```
V1 N1 N4 V
```


$$B = \begin{bmatrix} \frac{1}{L_1} \\ 0 \end{bmatrix}$$

$$x = [i_{L_1} \quad v_{C_1}]^T$$

$$u = [v_{V_1}]^T$$

また、[モデリング] ボタンを押した後に、[伝達関数の算出] ボタンを押すことで、電源から状態変数への伝達関数を求めることができる。

A.4.1.1 回路方程式の解

次にこの回路の方程式を解いてみる。[>Solution] タブをクリックして、基準電位となるノードを指定する。今回の場合は、N4を基準電位として選ぶ。その次に[Solve] ボタンを押すことで、回路方程式の解が得られる。

The screenshot shows the 'Solution' tab in the software. On the left, under 'Select GND Node', nodes N1, N4, N2, and N3 are listed. Node N4 is selected. The main table shows the following equations:

Variable	Equation
Solution =	_
i_V1	= -i_L1
v_R1	= i_L1*R1
i_R1	= i_L1
i_C1	= i_L1
v_L1	= -i_L1*R1 + (v_V1 - v_C1)
v_N1	= (v_V1 + v_N4)
v_N2	= -i_L1*R1 + (v_V1 + v_N4)
v_N3	= (v_C1 + v_N4)

Below the table is a 'Solve' button. At the bottom, there is an 'アクション' (Action) section with several buttons: '電流の向き' (Current direction), 'モデリング' (Modeling), 'Mathematica形式でコピー' (Copy in Mathematica format), 'TeX形式でコピー' (Copy in TeX format), '伝達関数の算出' (Calculate transfer function), 'MATLAB形式で出力' (Output in MATLAB format), and 'Scilab形式で出力' (Output in Scilab format).

図 A.8: LCR 回路の回路方程式の解

A.4.2 磁気結合素子がある回路

ワイヤレス給電回路の送電・受電コイルや変圧器が存在する場合の回路の状態方程式を生成する。

例として、次の回路を用いる。

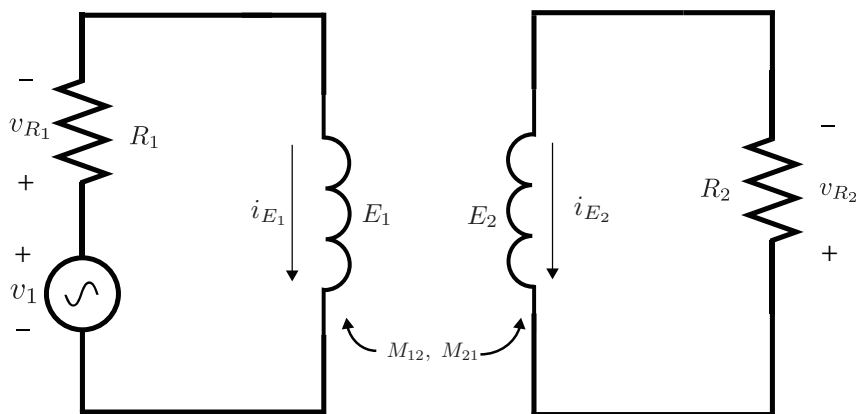


図 A.9: トランスが存在する回路

インダクタ E_1 と E_2 は磁気結合していて, M_{12} , M_{21} は相互インダクタンスを表す⁴。

1. 電圧制御型電圧源を用いて, 磁気結合がある素子を表現する⁵。

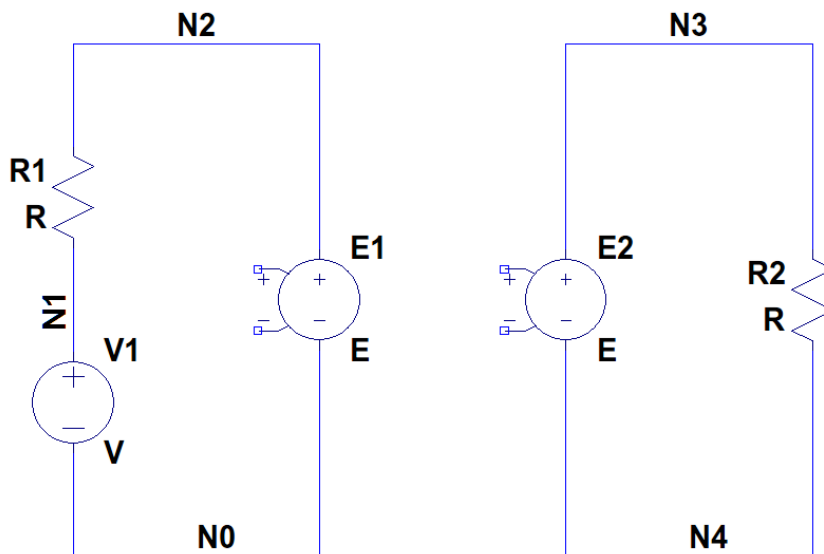


図 A.10: LTSPICE 上のトランス回路

電圧制御型電圧源のシンボル

⁴相互インダクタンスの相反定理から, $M_{12} = M_{21}$ である。

⁵LTSPICE では, $K L1 L2$ のように, SPICE directive を使って, インダクタの結合を指示し, トランスとして扱う。Wasabi はそのかわりに, 電圧制御型電圧源を用いているだけである。

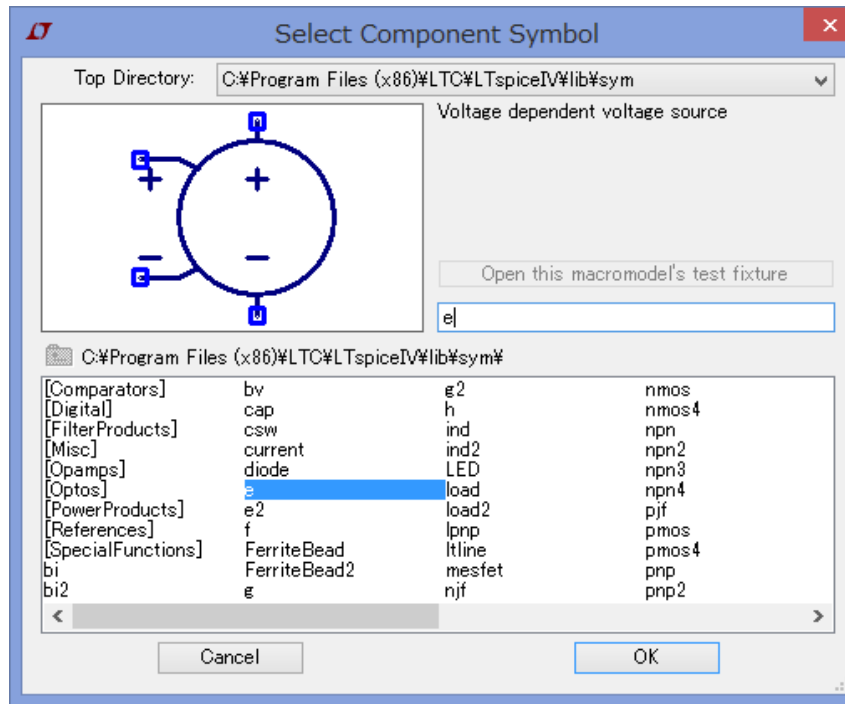


図 A.11: 電圧制御型電圧源の選択 (LTSPICE)

2. 電流の向きを確認して、モデリングボタンを押す。

>State Equation		>Solution	
$-R1 \cdot E2$	$R2 \cdot M12$	i_{E1}	$E2$
$(E2 \cdot E1 - M21 \cdot M12)$	$(E2 \cdot E1 - M21 \cdot M12)$		$(E2 \cdot E1 - M21 \cdot M12)$
$R1 \cdot M21$	$-R2 \cdot E1$	i_{E2}	$-M21$
$(E2 \cdot E1 - M21 \cdot M12)$	$(E2 \cdot E1 - M21 \cdot M12)$		$(E2 \cdot E1 - M21 \cdot M12)$
$A(2 \times 2)$		$x(2 \times 1)$	$b(2 \times 1)$
			$u(1 \times 1)$

図 A.12: 磁気結合した回路の状態方程式

A.4.2.1 回路方程式の解

次にこの回路の方程式を解いてみる。基準電位となるノードは、 $N0$, $N4$ を基準電位として選ぶことにする。

>State Equation >Solution

Select GND Node

N2
N0
 N3
N4
 N1

Solution	=	_
v_E1	=	-i_E1*R1 + v_V1
v_E2	=	-i_E2*R2
i_V1	=	-i_E1
v_R1	=	i_E1*R1
i_R1	=	i_E1
v_R2	=	i_E2*R2
i_R2	=	i_E2
v_N2	=	-i_E1*R1 + (v_V1 + v_N0)
v_N3	=	(-i_E2*R2 + v_N4)
v_N1	=	(v_V1 + v_N0)

Solve

図 A.13: 磁気結合した回路の回路方程式の解