

パワーエレクトロニクスを対象とした
リアルタイム・シミュレータに関する研究

崇城大学 応用情報学専攻 博士課程 城所 仁

目次

はじめに	5
1 章 リアルタイム・シミュレータ	8
1.1. フィードバック制御機器開発におけるリアルタイム・シミュレータの必要性	8
1.2. パワーエレクトロニクス分野におけるリアルタイム・シミュレータの現状	9
1.3. リアルタイム・シミュレータの構成	11
1.4. MBD と V 型開発手法	14
2 章 常微分方程式の数値解法	17
2.1. 計算ステップとリアルタイム性	17
2.2. 数値計算の安定性	18
2.3. 1 段階法	19
2.3.1. 前進 Euler 法	19
2.3.2. 前進 Euler 法の絶対安定領域	19
2.3.3. 後退 Euler 法	20
2.3.4. 後退 Euler 法の絶対安定領域	21
2.3.5. 台形積分法	22
2.3.6. 台形積分法の絶対安定領域	23
2.3.7. Runge-Kutta 法	24
2.3.8. Runge-Kutta 法の絶対安定領域	25
2.4. 線形多段階法	26
2.4.1. 線形多段階法の一般形式	27
2.4.2. 線形多段階法の絶対安定領域	28
2.4.3. Adams-Bashforth 法	28
2.4.4. Adams-Bashforth 法の絶対安定領域	29
2.4.5. Adams-Moulton 法	30
2.4.6. Adams-Moulton 法の絶対安定領域	30
2.4.7. 後退微分公式(BDF/Gear 法)	31
2.4.8. 後退微分方程式の絶対安定領域	32
2.5. 結論	33

3 章 回路方程式の構築手法.....	34
3.1. 回路方程式の構築手法の歴史.....	34
3.2. 回路方程式に使用する基本的な公式.....	35
3.3. 微分方程式の扱い.....	35
3.4. サンプル RLC 回路の設定.....	36
3.5. 節点解析法 (Nodal Analysis Method)	37
3.5.1. 基本的な考え方.....	37
3.5.2. 具体的な方程式の構築法 (スタンプ)	39
3.5.3. 基準電位の未知変数からの削除.....	39
3.5.4. 動的要素の扱い (キャパシタ)	40
3.5.5. 動的要素の扱い (インダクタ)	42
3.5.6. サンプル回路の定式化.....	43
3.6. 修正節点解析法 (Modified Nodal Analysis Method)	44
3.6.1. 基本的な考え方.....	45
3.6.2. 具体的な方程式の構築法-1 (等価電流源方式)	45
3.6.3. 具体的な方程式構築法-II (等価電圧源方式)	46
3.6.4. サンプル回路の定式化.....	47
3.7. スパースタブロー法 (Sparse Tableau Method)	48
3.7.1. 接続行列の導入.....	49
3.7.2. タブロー方程式の構築.....	50
3.7.3. サンプル回路の定式化.....	51
3.8. カットセット解析 (Cut Set Analysis Method)	52
3.8.1. カットセットとループの定義.....	52
3.8.2. 標準木.....	55
3.8.3. 状態変数でない余剰変数の削除.....	57
3.8.4. サンプル回路の定式化.....	58
4 章 スイッチング回路を計算する最適な計算手法.....	62
4.1. スイッチ素子の扱い.....	62
4.2. スイッチング時間の分解能の影響.....	62
4.3. スイッチ動作を伴う回路の代表的な計算手法	64
4.3.1. 固定ステップ法.....	64
4.3.2. 状態平均化法	65
4.3.3. 部分的可変ステップ法 (スイッチ駆動信号同期)	66
4.3.4. 部分的可変ステップ法 (スイッチ駆動信号非同期)	67
4.3.5. 状態変数の補間／外挿手法	68

4.4.	デッドバンド時間の扱い	70
4.5.	スイッチ非線形素子の扱い.....	71
4.6.	結論　一スイッチング回路の計算に最適な計算アルゴリズムー.....	72
5 章	陽解法による計算可能性と最適アルゴリズムによる実装結果.....	76
5.1.	比較対象とする実回路	76
5.1.1.	実験回路の作成.....	76
5.1.2.	導出された回路方程式.....	78
5.1.3.	回路定数の設計.....	80
5.1.4.	安定性解析	81
5.2.	汎用プロセッサによる実装.....	83
5.2.1.	Simulink モデル.....	84
5.2.2.	dSPACE 社 RTI ブロックによる I/O 実装.....	86
5.2.3.	実装上の問題	87
5.2.4.	実験結果（実回路との比較）	88
5.3.	FPGA による実装.....	91
5.3.1.	Xilinx 社 SystemGenerator を用いた回路モデルの実装.....	92
5.3.2.	dSPACE 社 RTI-FPGA ブロックを用いた I/O 実装.....	92
5.3.3.	モデルの構造	93
5.3.4.	実験結果（実回路との比較）	99
6 章	まとめ	104
6.1.	総括	104
6.2.	残された課題.....	105
6.3.	最期に	105
	謝辞.....	106
	参考文献	106
	添付資料	110

はじめに

本論文は、「パワーエレクトロニクスを対象とするリアルタイム・シミュレータ」を開発するために必要な理論と実装技術に関する研究を記したものである。

制御対象の挙動を検出し、所望の状態へ導くフィードバック制御は、様々な分野で社会を支える基幹技術であるが、これを開発するにおいては、制御対象を実験機材として必要とするという本質的な問題がある。しかしながら開発段階の不完全なコントローラを用いて制御を行う事は制御対象の破壊等のリスクを伴うため、制御対象の代替を行うリアルタイム・シミュレータを活用する事は、今日では常識とされている。

パワーエレクトロニクスの分野においても、組み込み用マイクロプロセッサの進歩により、近年ではソフトウェアによる複雑で高度なフィードバック制御が行われるようになり、その制御開発の効率化を図るために、リアルタイム・シミュレータの必要性が非常に高まってきている。また、制御対象として高電圧／大容量の電力回路を使用する場合は、制御開発時に実機を用いることは安全性の観点からも極めて困難であるため、リアルタイム・シミュレータは不可欠である。

しかし、パワーエレクトロニクス分野においては、下記に述べる制御周期の問題などにより、十分な性能を持つリアルタイム・シミュレータはいまだ実現されておらず、これを開発する事は、この分野の制御ソフトウェア開発を大きく前進させ、より複雑な制御を、機器の破壊や火災などリスクなしに、より効率的に行う事を可能とするものである。

リアルタイム・シミュレータとは、上記のようにフィードバック制御のコントローラの開発に用いられ、実制御対象を用いて開発や検証を行うことが困難な場合に制御対象を代替する装置であるが、今日では、制御対象を微分方程式で表し、デジタル計算機を用いてリアルタイムに方程式を計算する事で実現されている。リアルタイムとは、微分方程式を計算するに当たり、方程式が持つ時間軸と同一の時間軸で計算結果を出力する事である。リアルタイム・シミュレータを実現するうえでの最大の課題が、この高速な計算の実現にある。

パワーエレクトロニクスの分野では、回路シミュレータとしては、回路動作の検証を行う回路シミュレーションソフトウェアが一般的であるが、このような回路シミュレーションソフトウェアとリアルタイム・シミュレータを比べた場合、「リアルタイム・シミュレータは、コントローラに対して、制御対象の代替を行うものであり、制御対象、すなわち回路の動作の検証を行うものではない」という点において、根本的に目的を異とする装置である。従って、「リアルタイム・シミュレータ」を用いて、回路の損失や効率などを検討する事は適当ではない。

また、リアルタイム・シミュレータは、制御対象の代替をする装置であるから、制御装置と直接接続されて使用される。従って、リアルタイム・シミュレータは、制御装置と物理的な信号をやりとりする入出力 (I/O) 装置を有し、それゆえ計算出力はリアルタイムでなければならない。また、I/O 装置を持つ点においてもまた、回路シミュレーションソフトウェアとは異なるものである。

現在、航空宇宙や自動車等の機械系の分野におけるコントローラの開発においては、リアルタイム・シミュレータの活用は不可欠と言える状況であるが、パワーエレクトロニクス分野では、その主要な回路であるスイッチング回路の制御周波数が機械系に比べ非常に高いこと、および計算において高い時間分解能が必要なことより、これに対応できる高速な計算手法が確立されておらず、実用的なリアルタイム・シミュレータはいまだ実現されていない。

そこで、本論文では、スイッチ動作時刻を正確に計算する部分的可変ステップ法と名付けた計算手法を提案する事で時間分解能の向上を図ると同時に、連続系の状態方程式で記述された回路方程式を計算負荷が極めて少ない陽解法による数値積分で計算することにより、計算時間の大幅な短縮を実現した。陽解法は従来ほとんど使用されなかったが、リアルタイム・シミュレータが計算すべき回路の種類は限定されることに着目して、陽解法が適用可能であることを本論文で初めて明らかにした。

その結果、従来は不可能であった、汎用プロセッサを使用して一般的なスイッチング周波数で動作するスイッチング回路のリアルタイム・シミュレータを実現し、さらに、より高速な FPGA を使用することによって、これまで実現できなかった高周波のスイッチング周波数を持つリアルタイム・シミュレータを実現した。これにより、パワーエレクトロニクス分野においても、モデルベース開発法などの最新の開発技術を十分に活用することが可能になった。

以下に本論文の構成を記す。

第 1 章は序論であり、リアルタイム・シミュレータの必要性、それが生み出された歴史的経緯、パワーエレクトロニクス分野での現状、そしてその構成について述べた。

第 2 章においては、シミュレーション技術の基礎である、常微分方程式の数値解法について、計算式と数値安定性の条件についての検討を行った。

第 3 章においては、現在までに提案されてきた主要な回路定式化手法を検討し、同一のサンプル回路がどのような方程式で表されるかを示し、その得失を明らかにした。

第 4 章においては、スイッチング回路特有の問題についての考察を行い、第 2 章の数値解法、第 3 章の回路方程式の、どの組み合わせが、パワーエレクトロニクスを対象とした、リアルタイム・シミュレータを構築するため最適であることを明らかにした。

第 5 章においては、20kHz という標準的なスイッチング周波数において、標準的な回路

定数をもつ回路を実際に製作し、この条件下で第 4 章において提案された手法が適用可能であることを示し、またこの手法を実際に実装し、その効果を確認した。

実装は 1GHz のクロックで駆動される汎用プロセッサ (IBM PowerPC 750FX) と 100MHz で駆動される FPGA (Xilinx 社 Virtex-5) の 2 種類のハードウェアを用い、得失を検討した。

第 6 章において、全体を総括し、残された課題について言及した。

1章 リアルタイム・シミュレータ

1.1. フィードバック制御機器開発におけるリアルタイム・シミュレータの必要性

自動制御と言われるものは、制御対象に対し単純に入力を与えるだけの開ループ制御と、対象の挙動を検知し対象の挙動が所望の目標値に一致するように入力を与える、フィードバック制御としても知られる閉ループ制御に分類されるが、通常、自動制御と言えば、後者を指す。

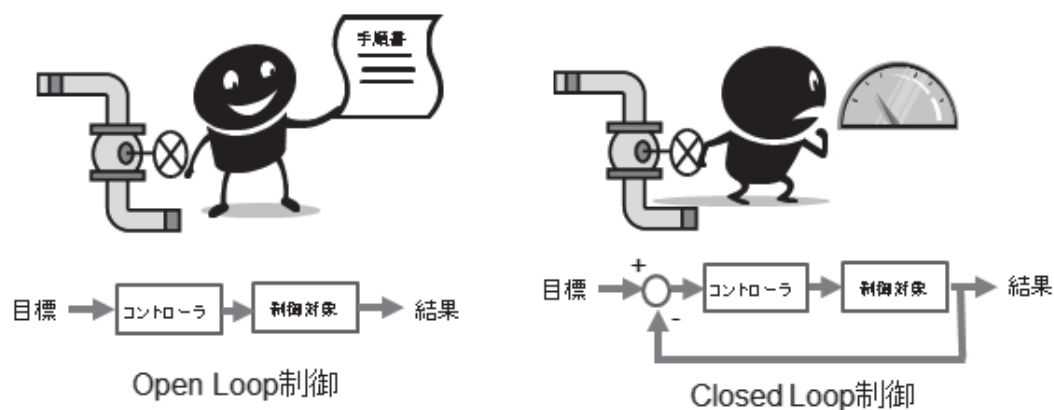


図 1.1 オープンループ制御とクローズドループ制御

フィードバック制御には条件によっては制御が発散するという重大な問題があり、自動制御における理論とは、この発散をいかにして防ぐか、すなわち安定な制御を実現するかという点に集約されるのであるが、さらにもう一つの問題として、制御装置の開発に際して、制御対象を必要とする、という問題がある。

制御装置は制御対象からフィードバックされる信号を入力とするため、制御対象がなければ入力値が得られない。これを仮のシステムで代用し、擬似的な信号を入力値とすることは可能であるが、制御対象の挙動は、制御装置の制御出力に依存するという因果関係がある。従って仮のシステムは、制御装置から受けた信号に対し、制御対象と同じ出力を適切なタイミングで出力する事が要求される。

このような装置は容易に得られるものではなく、制御理論が発展した 20 世紀初頭においても制御装置は、制御対象の“現物”を使用して開発する事が常識であった[1]。

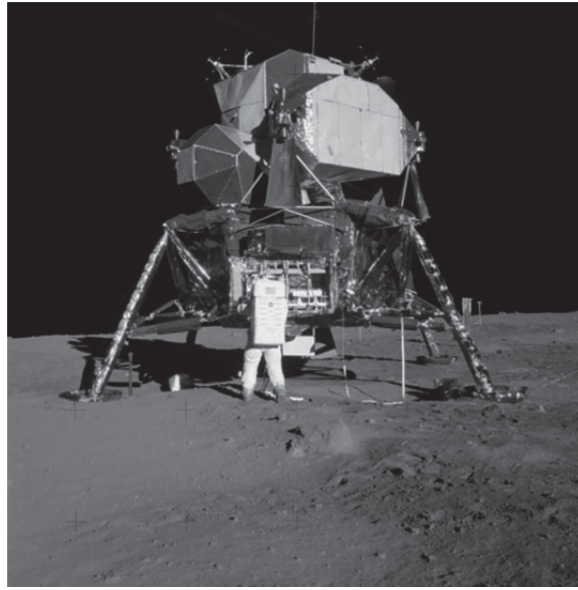


図 1.2 月着陸船（出典：NASA ホームページ）

しかし、電子技術、機械技術、および制御理論の発展に従い、制御装置が様々な分野に用いられるに従って、この問題が制御系の開発に対する、大きな障壁として顕在化してきた。典型的な例としては、第 2 次大戦後に開発された、敵戦闘機を自動追尾する飛翔体、すなわちミサイルや、宇宙ロケットの開発がある。

ミサイルなどにおいては、当時の技術では、実験を行っている間に制御装置の動作状況を詳細に観測する事が困難であり、さらに実験終了後に機材を無傷で回収することは期待できない。従って飛ばしてしまう前に、少なくともある程度は完成している必要がある。宇宙開発ではさらに問題は顕著で、宇宙や月面で使用する制御装置は、飛ばす前に完成している事はもちろん、失敗することが絶対に許されないにも関わらず、地球上での事前確認が全くできない。

このような課題を克服するために開発されたものが、リアルタイム・シミュレータである。

図 1.2 に示す月着陸船のオートパイロット装置の開発に使用されたシミュレータは、ビル 3 階のフロアを占有し、当時最新の IBM-7090 コンピュータ、アナログコンピュータ、および慣性力を再現する実物大モックアップから構成されていたという[2]。

1.2. パワーエレクトロニクス分野におけるリアルタイム・シミュレータの現状

以上のように今日では、航空機や自動車など、機械的な対象を制御するフィードバック制御を行う制御装置の開発において、リアルタイム・シミュレータは必要不可欠の存在である [3]。

しかしながら、インバータや DC/DC コンバータを初めとするパワーエレクトロニクス分野では様々なフィードバック制御系が用いられているにも関わらず、リアルタイム・シミュレータは、研究開発自体は活発に行われているものの、必要不可欠と言えるほどには普及はしていない。

その理由としては一つにはニーズの欠如がある。パワーエレクトロニクス分野では、基本的に半導体素子を ON/OFF の 2 状態のスイッチ素子として用い、交互に ON-OFF を繰り返して用いる。現実には半導体素子は理想スイッチではなく、スイッチを繰り返す周期が短ければ、単位時間あたりのスイッチ回数が増え、スイッチ損失が増大してしまうのであるが、一方インダクタやキャパシタなどの部品が小さくなりコストと小型化に貢献する。そのため、半導体素子のスイッチ特性が良くない年代においても msec 以下のスイッチ周期、素子の特性が向上した今日では μ sec オーダのスイッチ周期を使用しており、このような周期を制御周期とした場合、従来の組み込み制御用のプロセッサでは対応困難であり、制御にはアナログ回路が用いられていた。

アナログ回路での制御では、複雑な制御則を適用する事は困難であり、Type1、Type2、Type3 などと呼ばれる形の定まった位相遅れ進み制御が適用されていた[4]。このような制御系では、制御設計は周波数ゲイン/位相特性のみであり、調整が必要なパラメータ（素子定数）も少ないため、コントローラの開発ツールとしてリアルタイム・シミュレータを使用するほどのニーズは存在しえなかった。

しかし、近年では、高速、安価、かつ組み込み用途としての耐環境性に優れたデジタルデバイスが登場し、人間の可聴範囲を超える事より一応の目安とされる 20kHz のスイッチング周期におけるデジタル制御が可能となり、その結果、制御則が複雑化する傾向があり、それを実現する信頼性の高い制御ソフトウェアの開発のためのリアルタイム・シミュレータのニーズは日増しに増している。

一方、シーズ、すなわち技術的側面においても、パワーエレクトロニクスを構成するスイッチング回路を模擬するリアルタイム・シミュレータの実現は困難であった。

リアルタイム・シミュレータを実現する技術は、同様に電子回路を方程式として表し計算する、回路シミュレータの技術に基づくものではあるが、汎用回路シミュレータとして標準的な存在である SPICE[5]は、修正節点解析法を用い、非線形特性を反復計算で線形方程式に変換するため、扱える回路トポロジーの範囲も広く、非線形素子も良好に扱えるという特徴を持つが、その一方計算負荷が高く、リアルタイム計算は困難である。

送電系のシミュレータとして著名な EMTP[6]も、同じく節点解析法を用い微分方程式の離散化手法として台形積分法を標準としているため、やはり計算負荷が高く、さらに、スイッチング動作を行う回路では、数値的振動を起こすなどの問題があり[7]、やはりリアルタイム・シミュレーションに向かない。

以上のように、多く普及している“非”リアルタイム・シミュレーションソフトウェア

の理論とソフトウェア構造を直接利用して、リアルタイム・シミュレータを開発した場合には、数 100Hz 程度の低いスイッチング周波数に限定されるものにならざるを得ず[8][9]、スイッチング周波数を上げた場合は、並列プロセッサを使用するなど大規模なハードウェア[11][12]が必要となるが、そのため価格的にも高価となり、パワーエレクトロニクス分野におけるリアルタイム・シミュレータの普及を阻む要因となっていた。

しかし、近年では、FPGA(Field-Programmable Gate Array)等を利用して、比較的安価に 20kHz 程度のスイッチング周波数を実現する例もあり[13]-[15]、パワーエレクトロニクス分野でもリアルタイム・シミュレータの開発は活発になりつつあるが、まだ従来の”非”リアルタイム・シミュレーションソフトウェアの延長にあるアイデアを FPGA というハードウェアの計算能力で克服している傾向がある。

従って、リアルタイム・シミュレーション用に新しい理論やソフトウェアを開発すれば、より低コストなハードウェアで同等な性能が達成可能となる。あるいはこのような進歩したハードウェアを用いれば、より高い精度や高いスイッチング周波数を実現可能となることより、パワーエレクトロニクスの分野におけるリアルタイム・シミュレータの普及に貢献すると考えられる。

1.3. リアルタイム・シミュレータの構成

リアルタイム・シミュレータとは、実際の制御装置に対し、制御対象の代替を行うものであり、閉じた制御ループの中に、コントローラが存在する事が特徴である。そのため、HILS (Hardware In the Loop Simulator) とも呼ばれる。

これに対し、実際の制御装置ではなく、制御装置に内蔵される制御アルゴリズム、あるいは制御ソフトウェアに対し、制御対象を模擬した数学モデルを組み合わせ、制御アルゴリズムの妥当性を検証する、というシミュレーション手法もあり、この場合はオフライン・シミュレーションと呼ばれる。

オフライン・シミュレーションでは、純粋にソフトウェアレベルでのシミュレーションであるから、制御則の開発や、制御ソフトウェアの基本的な性能を確認する事には役立つが、コントローラに実装した最終状態での、I/O 性能やソフトウェア処理のタイミングの厳密な検証はできないため、両者の使い分けが重要であるが、最終状態での確認は必須であるから、リアルタイム・シミュレータの重要度は高い。

リアルタイム・シミュレータが“リアルタイム”である理由は、実際の制御装置を接続し、物理的な信号をやりとりする事による要求事項である。すなわち、物理的な時間軸を持つハードウェアとしての制御装置と、同じく物理的な時間軸を持つ入出力信号に接続されるため、同じ物理的時間軸を持つことが要求されている。

以上より、リアルタイム・シミュレータとは、

- ① コントローラが出力する信号を計測し
- ② 制御対象の挙動を、数学モデルを用いて所定の時間以内に計算し
- ③ 制御対象の挙動を検知するため設置されるセンサー信号に代わる信号を出力する。

装置である。

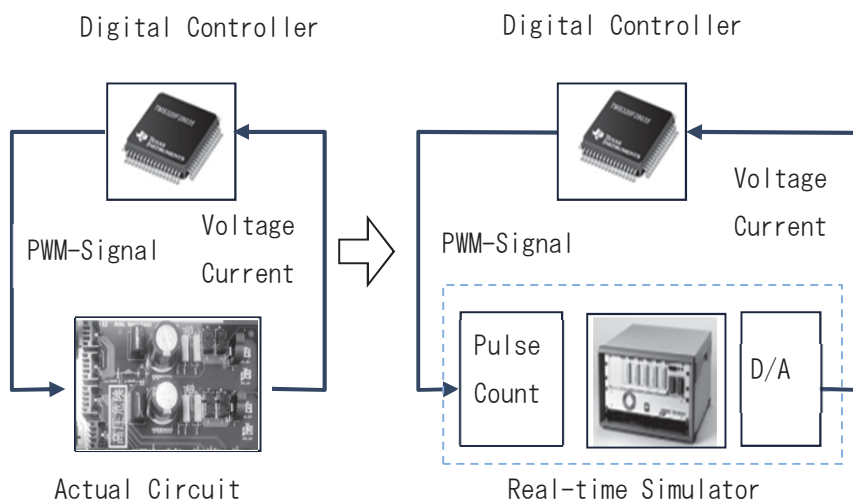


図 1.3 リアルタイム・シミュレータの構成

制御装置の持つ入出力信号は制御対象により様々なものがあるため、リアルタイム・シミュレータが持つ I/O 装置は、制御対象に応じて様々である。従って、多くの場合は、リアルタイム・シミュレータは模擬する対象に応じたカスタム仕様となる。

なおパワーエレクトロニクス分野においては、制御装置が出力する信号は一般的には複数のスイッチ素子駆動信号（多くは PWM 信号）であり、入力は電圧値、そして電圧に変換された電流値である。

図 1.3 にリアルタイム・シミュレータの使用例を示す。リアルタイム・シミュレータは I/O 装置を持つため、図 1.3 のように実機の代わりにコントローラを直接接続して使用することができる。

以上のように入出力装置は様々であるが、すべてのリアルタイム・シミュレータに共通しその根幹となるものは、制御対象の数学モデルと、これを計算する計算手段である。

初期のリアルタイム・シミュレータは、計算手段として、真空管を用いた計算回路、すなわち図 1.4 のようなアナログ計算機を使用していた[16]。アナログ計算機の計算速度は、物理現象そのものであることより、完全なリアルタイム性が得られるが、その一方、計算における精度を維持する事が困難で、また数式を入れ替えるに際し、回路構成を変更する必要がある、汎用性を持たせた場合、極めて複雑な装置となる。そのため、現在では計算手段として、デジタル計算機が用いられている。

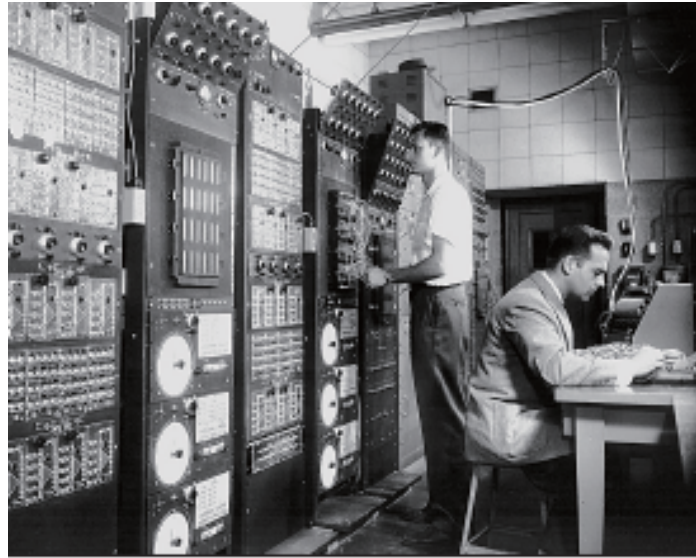


図 1.4 初期のアナログ計算機 (出典：参考文献[16])

デジタル計算機を用いた場合、信号値は量子化され、かつ時間的には離散化されていなければならない。離散化された値は、離散時間、すなわち計算ステップ時間を持つが、デジタル計算機におけるリアルタイム性とは、すなわちこの計算ステップ時間よりも短い時間において、所望の計算を完了する事であり、そのためには、

- I 最適な計算ステップ時間
- II 高速な計算手段
- III 効率的な計算手法
- IV 効率的な数学モデル (数式)

が必要である。

当然、計算ステップ時間、すなわち“締切り”が長ければ、他の要因の負担が減るわけであるから、Iの要素は極めて大きい。これは通常、接続するコントローラの制御則が適切に動作するというシミュレータ本来の目的による制約により、コントローラ制御周期の $1/5 \sim 1/20$ に設定される[17]。

IIの計算手段が高速であることは、絶対的な優位性であり、デジタル計算機を構成するデジタルプロセッサの高速化、あるいは複数のプロセッサを利用する並列化は、シミュレータ分野における一つの研究テーマであるが、本研究ではこれを対象としない。

IIIの計算手法の改善は極めて重要である。制御対象は連続系の微分方程式で表されるのが通例であるが、これを離散化する手法は数々あり、手法の選択により、計算時間は大きく変化する。

IVについては、そもそも”何を“計算するかは、最も重要なテーマであると言える。制御対象の数式表現は様々な手法があり、目的を達成する近似度を持ち、かつ最も計算を効率的に行える数学モデル(=数式)を導出する事が、計算の高速化に最も貢献し、リアル

タイム・シミュレータの性能を決定する。

本研究は、主にⅢとⅣについての改善を行うものである。

1.4. MBD と V 型開発手法

MBD とは Model-Based Design または Model-Based Development の略であり、日本語においては「モデルベース開発」と訳される。リアルタイム・シミュレータを実現するには、1.3 節で示したように、数学モデルの扱いが鍵であるが、The Math Works 社の製品であり、行列計算や常微分方程式ライブラリの集合体である MATLAB、およびブロック線図による記述を行う Simulink により、数学モデルを記述する事が、その扱いやすさより、今日ではデファクトスタンダードである。

この MATLAB/Simulink を用い、数学モデルを基盤とした (Model-Based) 制御ソフトウェア開発手法は、制御ソフトウェア開発の各段階において、様々な形で活用されているが、図 1.5 ような V 字型の図式で説明すると、これを理解しやすい[18]-[22]。

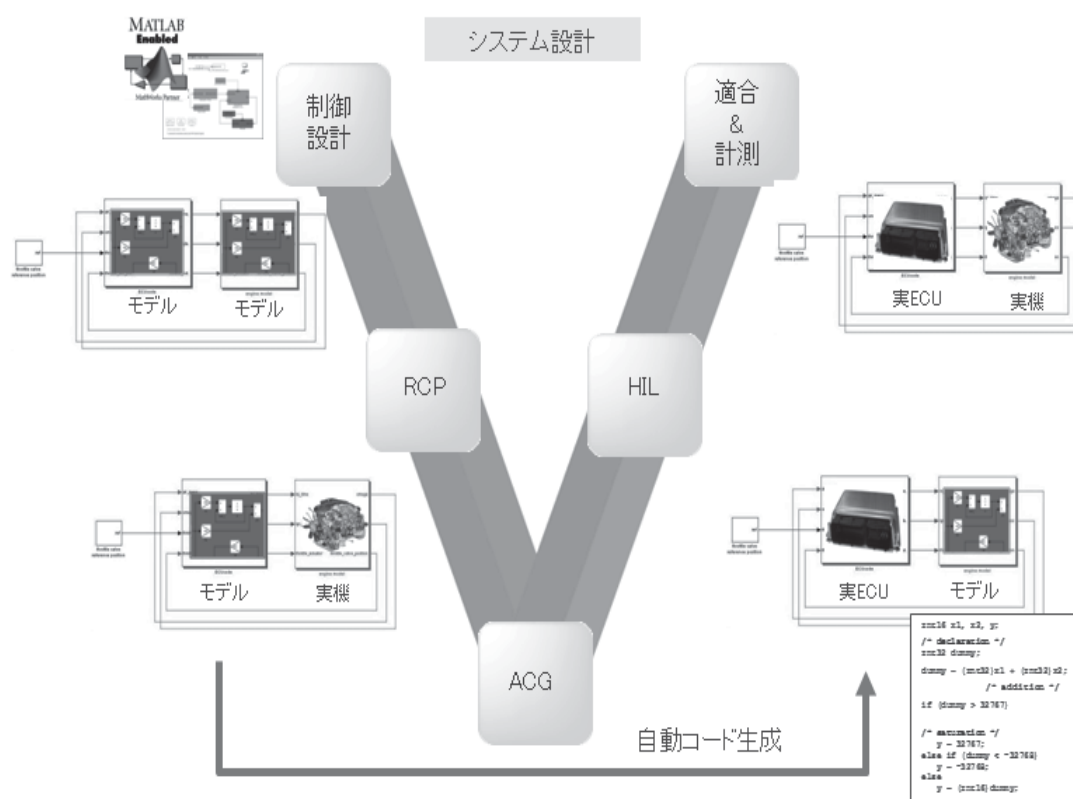


図 1.5 MBD が推奨する V 字型開発手法

図 1.5 の V 字型の開発手法は、まず、V 字の左上で制御対象とコントローラを数学モデルとして構築し、数値シミュレーションを行いコントローラの制御設計を検証する。これ

はオフライン・シミュレーションと呼ばれ、開発の早い段階において、制御対象を実験に用いることができない、あるいはそもそも制御対象が存在しない段階において、制御性能の妥当性を検証することが可能となる。

この時 Simulink を用いることで、コントローラ・モデルをブロック線図としてグラフィカルに表現することができるため、制御設計者には C プログラム等の知識は必要なく、制御ロジックの開発に専念できる。

次に、制御対象を数学モデルから実際の制御対象に置き換え、コントローラ・モデルにて直接制御対象を制御する RCP(Rapid Control Prototyping)と呼ばれる手法にて制御ロジックの検証を行う。

ここでは MATLAB/Simulink で設計され、ブロック線図として記述されたコントローラ・モデルを用いて直接制御対象を制御する。そのため、割り込み処理や固定小数点演算のようなプログラムに関する知識を持たない制御設計者であっても、容易に実機を用いた検証を行うことができる。実際にはブロック線図は The Math Works 社の Simulink Coder というツールによりブロック線図は C 言語に変換され、マイクロプロセッサによりプログラムとして実行されるのであるが、すべて自動的に行われるため、設計者がプログラムを意識することはなく、上記のようにコントローラ・モデルが直接制御対象を制御する、という感覚になる。

RCP に使用するハードウェアは基本的には、リアルタイム・シミュレータと同じ構成が用いられる。RCP とリアルタイム・シミュレータとの違いは、計算対象がコントローラであるか、制御対象であるかであるが、RCP の場合は自動車や航空機など移動する制御対象に搭載可能であることが必要であるため、形状は小型であり、バッテリー動作が可能である場合が多い。

次に、V 字の底辺にて、実際の装置に組み込むソフトウェアへ変換する。ここでは、ACG(Auto Code Generator)と呼ばれるソフトウェアにより Simulink 上のブロック線図を自動的に、実際に製品として使用可能なレベルの効率と信頼性を持つ C 言語プログラムに変換し、製品としてのコントローラに実装する。

ここで用いられる ACG ツールは RCP で使用した Simulink Coder に対し、量産レベルに対応できる、より高効率なコード生成が可能なものである。

Simulink にて記述された制御アルゴリズムを目標として、C 言語プログラムを一から手作業で作成しては、設計された制御アルゴリズムと制御ソフトウェアの一致性が疑わしくなり、RCP での実験結果が意味をなさなくなるため、この ACG の適用は MBD においては、極めて重要なプロセスである。

さらに V 字の右側へ進み、リアルタイム・シミュレータにてコントローラの機能を検証する。リアルタイム・シミュレータは、このような V 字型の開発プロセスが確立する以前に開発され、また単独で使用されることも多いが、フィードバック制御ソフトウェア開発において必要不可欠である以上、MBD においても重要な位置づけを占める。なお、MBD

においては、HILS（ヒルズ）ないし HIL（ヒル、もしくはエイチ・アイ・エル）との呼称が一般的である。

最後の V 字の右上では制御装置と制御対象を接続し、制御パラメータの最終調整が行われる。これまでは数学モデルを使用した開発であったが、数学モデルと実制御対象は必ずしも同一ではない。最終的な商品レベルのコントローラと、実際の制御対象の組み合わせにおける最終調整は必要不可欠である。

パワーエレクトロニクスの分野においても、このような MBD の手法を適用する事により、制御ソフトウェア開発の効率化が期待される。

しかしながら、現時点においては、制御周期の問題により、RCP の実行や HIL の構築が困難であるため、パワーエレクトロニクスの分野においては MBD の開発プロセスを十分に活用する事はできない。しかし本研究において、計算アルゴリズムの改善により、並列プロセッサなどの高価なハードウェアを用いることなく、20kHz のスイッチング周期に対応する HIL が構築できることが示されたことにより、MBD の導入が促進されるものと思われる。

2章 常微分方程式の数値解法

本章で対象にする微分方程式は常微分方程式(ODE: Ordinary Differential Equation)であり、式(2-1)で表される。

ここで、 \mathbf{X} はベクトルであり、高階の方程式はベクトルの次元を増やして表現する。

$$\frac{d}{dt}\mathbf{X} = f(\mathbf{X}, t) \quad (2-1)$$

これに対し、式(2-2)で表現される、陰的微分方程式は DAE (Differential Algebraic Equation) と呼称されるが[23]、この形になる回路方程式の定式化手法を本論文では対象としていないため、これに対する数値解法も、本論文では同じく対象外とする。

$$f(\dot{\mathbf{X}}, \mathbf{X}, t) = 0 \quad \dot{\mathbf{X}} = \frac{d}{dt}\mathbf{X} \quad (2-2)$$

2.1. 計算ステップとリアルタイム性

スイッチング回路用のリアルタイム・シミュレータとは、スイッチング回路を表現する回路方程式の過渡解析を実時間でを行い、回路方程式が意味する物理的な動作を再現する、シミュレーション装置である。

今日、リアルタイム・シミュレータはデジタル計算機により実現されるが、デジタル計算機における回路方程式の過渡解析、すなわち回路方程式の時間軸応答とは、所定の離散化時間、すなわち計算ステップにより離散化され、差分方程式に変換された微分方程式を、逐次的に計算していくことである。従って、これを実時間で行うとは、計算に必要な時間が、離散化に使用した計算ステップ時間より短いことを意味する。

そのために、計算ステップを、必要となる計算時間に合わせて延長するという逆転の発想もありうるが、リアルタイム・シミュレータがコントローラの制御ソフトウェアの開発を目的とする事より、これは成立しない。すなわち制御ソフトウェアに対して実制御対象に代替する信号を与えるためには、計算ステップは制御周期の $1/5 \sim 1/20$ である必要があり[17]、計算ステップ時間をこれより長く設定する事はできない。

さらに、計算ステップの延長は、数値計算の安定性という点からも制約を受ける。微分方程式の数値解法の多くは、計算ステップ時間が所定の条件を満たさない場合、計算が発散する[24]。従って、選択する解数値法に適した計算ステップを選定することもまた重要である。

SPICE などの汎用回路シミュレータにおいては、実時間性の制約が無い場合、計算ステップは主に計算精度により決定される。従って計算精度に応じて計算ステップを逐次変更する可変ステップの手法が、最も実用的な選択となる。可変ステップを採用した場合は、

ステップ幅が著しく短く、あるいは長くなる場合があるが、いかなるステップ幅において計算が不安定になることは避けるべきであるから、計算安定性に優れた台形積分法や後退微分公式 (BDF: Backward Difference Formula/Gear 法) などが用いられる傾向がある。

しかしながら、リアルタイム・シミュレータとしての観点では、上記のごとく、計算ステップの下限は計算速度、上限はコントローラの制御周期で規定されるため、汎用回路シミュレータとは異なる観点が必要である。

本章では一般的に知られている常微方程式の数値解法の計算精度と安定性を調べ、リアルタイム・シミュレータとして最適な計算手法を決定する。

2.2. 数値計算の安定性

上述のように、計算ステップは、計算の精度と安定性が考慮されなければならない。微分方程式の数値解法においては、対象となる微分方程式は必ずしも線形であることに限らないが、計算の安定性は、各計算の 1 ステップにおける問題であり、この 1 ステップにおいては線形と見なすことができるため、線形方程式を用いて議論される。

線形の常微分方程式をベクトル X と行列 A を用いて表すと、

$$\frac{d}{dt}X(t) = AX(t) \quad (2-3)$$

であるが、

これを固有値変換すると、

$$\frac{d}{dt}Y(t) = \Lambda Y(t) \quad (2-4)$$

となる。

ここで Λ は行列 A の固有値を対角要素として持つ対角行列であるから、式(2-4)は連立のスカラ方程式であり、式(2-5)となる。

$$\frac{d}{dt}y(t) = \lambda y(t) \quad (2-5)$$

なお、この式は Dahlquist のテスト方程式と呼ばれ[25]、常微分方程式の数値解法の特徴を比較する場合に、多く使用される。

この微分方程式は、 $\text{Re}(\lambda) < 0$ であれば任意の初期値において $x(\infty) = 0$ であるから、これを離散化し、数値積分した場合も、 $x_\infty = 0$ であることが期待される。

この条件は、任意の n において、

$$|x_{n+1}| \leq |x_n| \quad n=1,2,\dots \quad (2-6)$$

である事を意味するため、式(2-6)を満たす数値解法は、計算が発散しない。すなわち安定な計算が可能であると言える。($\text{Re}(\lambda) \geq 0$) であれば、テスト方程式自体が不安定、あるいは減衰のない振動であるから、この条件における数値安定性はあまり意味をもたない)

以後に各種の常微分方程式の計算精度、並びに計算安定性、すなわち式(2-6)を満たす条件を明らかにする。

2.3. 1 段階法

1 段階法とは、離散化されたベクトル \mathbf{X}_{n+1} を求めるに際し、 \mathbf{X}_n 、 \mathbf{X}_{n+1} およびこれらから計算された、 $f(\mathbf{X}_n, t_n)$ 、 $f(\mathbf{X}_{n+1}, t_{n+1})$ のみを使用する、いわば直前の過去以外の履歴は使用しない計算手法である。

2.3.1. 前進 Euler 法

前進 Euler 法は、最も単純であるが全ての解法の基本である。
以下にスカラー方程式で表すが、

$$\frac{d}{dt}x = f(x, t) \quad (2-7)$$

と表される常微分方程式において、計算ステップを h とすると、 $n > 0$ の整数 n により、 $t_n = n \times h$ であれば、

$$x_n = x(t_n) \quad (2-8)$$

と離散化変数 x_n が与えられたなら、

$$x_{n+1} = x_n + hf(x_n, t_n) \quad (2-9)$$

と計算するものが、前進 Euler 法である。

前進 Euler 法は x_{n+1} を計算するに当たり、 x_n 以外を必要としない。従って $n=0$ より順番に計算していけば、解を得ることができる。

このような計算手法を陽解法 (Explicit Method) と呼ぶ。

2.3.2. 前進 Euler 法の絶対安定領域

Dahlquist のテスト方程式に前進 Euler 法を適用すると、

$$x_{n+1} = x_n + h\lambda x_n = (1 + h\lambda)x_n \quad (2-10)$$

である。従って、

$$|(1 + h\lambda)| \leq 1 \quad (2-11)$$

であれば、式(2-6)を満たす事になる。

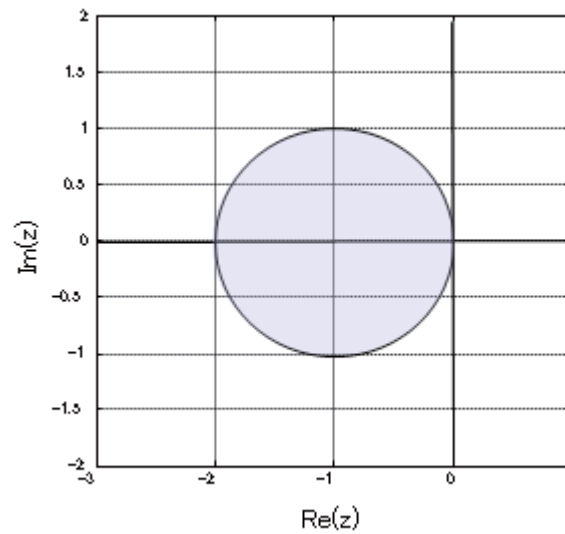


図 2.1 前進 Euler 法の安定領域

このとき、式(2-6)を満たす $z=h\lambda$ を複素平面上に表したものを絶対安定領域と呼ぶ。前進 Euler 法の場合は、式(2-11)であれば、式(2-6)を満たす。式(2-11)を満たす $z=h\lambda$ の領域を図 2.1 に示す。すなわち、計算対象の方程式が持つ固有値 λ に対し、 $z=h\lambda$ が図 2.1 の絶対安定領域となる h を選択しなければ、計算が発散してしまうのが前進 Euler 法である。

2.3.3. 後退 Euler 法

前進 Euler 法では、

$$\frac{d}{dt}x = f(x_n, t_n) \quad (2-12)$$

なる関係を用いているが、
これを

$$\frac{d}{dt}x = f(x_{n+1}, t_{n+1}) \quad (2-13)$$

としたものが、後退 Euler 法である。

従って、計算式は

$$x_{n+1} = x_n + hf(x_{n+1}, t_{n+1}) \quad (2-14)$$

となる。前進 Euler 法と後退 Euler 法の違いを図 2.2 に表す。

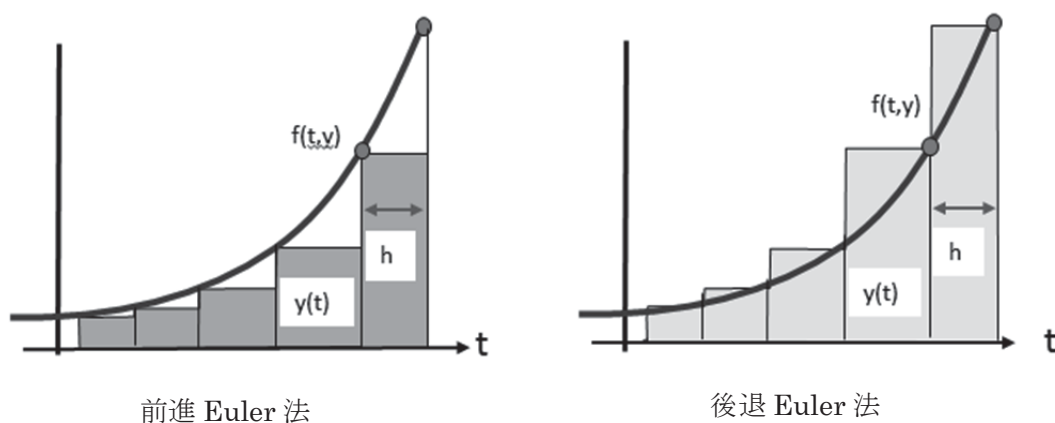


図 2.2 前進 Euler 法と後退 Euler 法の積分手法の考え方の違い

後退 Euler 法は、 x_{n+1} を計算するにあたり、 $f(x_{n+1}, t_{n+1})$ を必要とする。すなわち式の両辺に同じ変数が現れるため、計算に工夫が必要となる。

このような計算手法を陰解法 (Implicit Method) と呼ぶ。

2.3.4. 後退 Euler 法の絶対安定領域

後退 Euler 法にテスト方程式を適用すると、

$$\begin{aligned} x_{n+1} &= x_n + h\lambda x_{n+1} \\ (1 - h\lambda)x_{n+1} &= x_n \end{aligned} \quad (2-15)$$

以上より絶対安定領域は

$$\frac{1}{(1 - h\lambda)} \leq 1 \quad (2-16)$$

であり、図 2.3 となる。

後退 Euler 法では、安定な微分方程式 (λ の実部が負) はすべて安定に計算可能である。

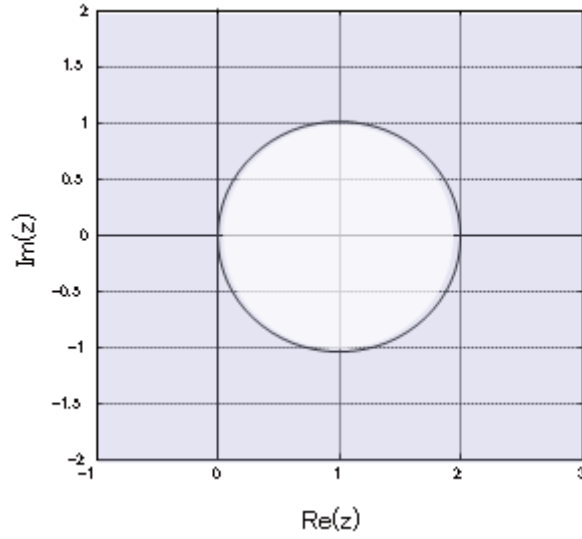


図 2.3 後退 Euler 法の安定領域

2.3.5. 台形積分法

図 2.2 を見ると、前進 Euler 法、後退 Euler 法ともに計算誤差が発生するが、両者を平均化すると、誤差が少なくなることが直感的に推測できる。これが台形積分法である。

理論的には、前進 Euler 法と後退 Euler 法は、それぞれ式(2-17)の x_{n+1} の x_n での Tyler 展開、および式(2-18)の x_n の x_{n+1} での Tyler 展開を、 h の 1 次の項で打ち切ったものであるから、 h^2 以降の項が誤差として残る。

$$x_{n+1} = x(t_{n+1}) = x(t_n + h) = x(t_n) + h \frac{d}{dt} x(t_n) + \frac{h^2}{2} \frac{d^2}{dt^2} x(t_n) + \frac{h^3}{3!} \frac{d^3}{dt^3} x(t_n) \dots \quad (2-17)$$

$$x_n = x(t_n) = x(t_{n+1} - h) = x(t_{n+1}) - h \frac{d}{dt} x(t_{n+1}) + \frac{h^2}{2} \frac{d^2}{dt^2} x(t_{n+1}) - \frac{h^3}{3!} \frac{d^3}{dt^3} x(t_{n+1}) \dots \quad (2-18)$$

$$x_{n+1} - x_n = x(t_n) - x(t_{n+1}) + h \frac{d}{dt} x(t_n) + h \frac{d}{dt} x(t_{n+1}) + \frac{h^3}{3!} \frac{d^3}{dt^3} x(t_n) + \frac{h^3}{3!} \frac{d^3}{dt^3} x(t_{n+1}) \dots \quad (2-19)$$

これに対し、式(2-17)から式(2-18)を引いて式(2-19)を作り、2 で割った場合、2 次の項が打ち消され、 h^3 以降の項が誤差として残る。このような場合、この計算手法は 2 次で適合している、あるいは 2 次の精度があるという。

ここで、式(2-19)より h^3 以上の項を打ち切り、

$$x_n = x(t_n) \quad (2-20)$$

$$\frac{d}{dt}x_n = f(x_n, t_n) \quad (2-21)$$

を適用して得られる式(2-22)は台形積分法と言われ、2次で適合した解法である。

$$x_{n+1} = x_n + \frac{h}{2} \{f(x_{n+1}, t_{n+1}) + f(x_n, t_n)\} \quad (2-22)$$

2.3.6. 台形積分法の絶対安定領域

同様に式(2-22)にテスト方程式適用すると、

$$x_{n+1} = x_n + \frac{h\lambda}{2}(x_{n+1} + x_n)$$

$$\left(1 - \frac{h\lambda}{2}\right)x_{n+1} = \left(1 + \frac{h\lambda}{2}\right)x_n \quad (2-23)$$

$$\left| \frac{(2 + h\lambda)}{(2 - h\lambda)} \right| \leq 1 \quad (2-24)$$

式(2-24)を表す領域は図 2.4 であるが、これは複素平面上の左半平面に一致する。すなわち、台形積分法の絶対安定領域は、連続系のテスト方程式の安定領域と一致する。

このように、数値解法の安定領域が元の連続系の方程式の安定領域、すなわち左半平面を含む性質を A 安定 (A-Stable) と呼んでいる[25]。後退 Euler 法は A 安定であるが、前進 Euler 法は A 安定ではない。

A 安定であれば、安定な連続系の方程式は、いかなる h においても安定に計算できるため、これは極めて重要な性質である。

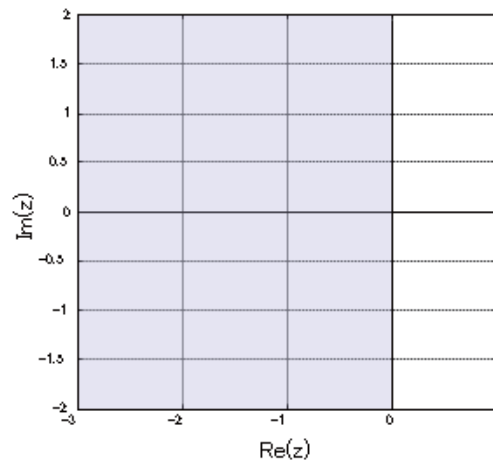


図 2.4 台形積分法の安定領域

2.3.7. Runge-Kutta 法

上述のように、前進／後退 Euler 法は、Tyler 展開を 1 次で打ち切った近似解であるから、打ち切る項を、より高次の項とすれば、計算精度の向上が期待できる。

しかし、1 階の微分値は、 $\frac{d}{dt}x(t) = f(x, t)$ と与えられているが、より高階の微分値は与えられていない。そこでこれを、やはり Tyler 展開を行い近似値として得る。

すなわち、任意の係数 a, b を持つ $f(x + bhf(x, t), t + ah)$ という式を導入し、これに 2 変数の Tyler 展開を 1 次まで行う。

$$f(x + bhf(x, t), t + ah) \approx f(x, t) + bhf(x, t) \frac{\partial}{\partial x} f(x, t) + ah \frac{\partial}{\partial t} f(x, t) \quad (2-25)$$

一方 $x(t)$ に対する 2 階の導関数は $\frac{d}{dt}x(t) = f(x, t)$ を用いて、

$$\frac{d^2}{dt^2}x(t) = \frac{d}{dt}f(x, t) = \frac{\partial}{\partial x} \frac{dx}{dt} f(x, t) + \frac{\partial}{\partial t} f(x, t) = f(x, t) \frac{\partial}{\partial x} f(x, t) + \frac{\partial}{\partial t} f(x, t) \quad (2-26)$$

と導くことができる。

式(2-25)と式(2-26)を比較すると、 $a=1$ 、 $b=1$ であれば、

$$h \frac{d^2}{dt^2}x(t) \approx f(x + hf(x, t), t + h) - f(x, t) \quad (2-27)$$

であるから、Tyler 展開の 1 次の精度で一致する事がわかる。

従って、式(2-27)を式(2-17)の 2 階微分項に代入して

$$\begin{aligned} x_{n+1} &= x_n + hf(x_n, t) + \frac{h}{2}(f(x_n + hf(x_n, t), t + h) - f(x_n, t)) \\ &= x_n + \frac{h}{2}(f(x_n, t) + f(x_n + hf(x_n, t), t + h)) \end{aligned} \quad (2-28)$$

が得られる。

式(2-27)の近似精度は h の 1 次であるが、さらに h が乗じて使用されるために、最終的には h の 2 次の精度が得られる。以上は改良 Euler 法、陽的台形法、あるいは Heun 法と呼ばれる計算手法である。

また、上記 a, b には、他にも解があり、修正 Euler として知られる陽的中点法などがある [26] (ただし、陽的台形法を修正 Euler 法と呼ぶ例 [27] もあるので、混乱を避けるうえで、中点法、陽的台形法と呼ぶ方が無難である)。

以上の手法は、2 次より高い項に対しても適用する事ができ、C. Runge.と M.W.Kutta により、計算に要する回数（段数）を s として、

$$X_i = x_{n-1} + h \sum_{j=1}^s a_{ij} f(X_j, t_{n-1} + c_j h) \quad 1 \leq i \leq s \quad (2-29)$$

$$x_n = x_{n-1} + h \sum_{j=1}^s b_j f(X_j, t_{n-1} + c_j h) \quad (2-30)$$

と、一般的な形に定式化されている。これが広義の Runge-Kutta 法である。
 なお、ここに表れる係数 a, b, c は Butcher Table として知られる、以下の記法により表現されることが多い。

表 2-1 Butcher Table

c_1	a_{11}	a_{12}	\cdots	a_{1s}
c_2	a_{21}	a_{22}	\cdots	a_{2s}
\vdots	\vdots			\vdots
c_s	a_{s1}	a_{s2}	\cdots	a_{ss}
	b_1	b_2	\cdots	b_s

通常は Runge-Kutta 法という呼称は、計算負荷に対して計算精度が最も優れる以下の Butcher Table で表される 4 次の公式を意味し、広義の Runge-Kutta 法と区別する必要がある場合は、これを古典的 Runge-Kutta 法と呼ぶ。

表 2-2 古典的 Runge-Kutta 法の Butcher Table

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	1/3	1/3	1/6

なお、 $a_{ij}=0, j \geq i$ 、の場合は陽解法となり、通常 Runge-Kutta 法と言えば陽解法を意味するが、上記にあてはまらない陰解法も Runge-Kutta 法に含まれる。近年では、陰的 Runge-Kutta 法が数値安定性の点において注目されている。

2.3.8. Runge-Kutta 法の絶対安定領域

Runge-Kutta 法の安定領域の導出は参考文献[23][25]などに譲るが、4 次以下の陽的

Runge-Kutta 法は次数を p とすると、

$$\left| 1 + h\lambda + \frac{(h\lambda)^2}{2} \cdots + \frac{(h\lambda)^p}{p!} \right| \leq 1 \quad (2-31)$$

であり、図 2.5 の領域となる（図 2.5 における RK-1、RK-2、RK-3、RK-4 はそれぞれ、1 次、2 次、3 次、4 次の Runge-Kutta 法の絶対安定領域の境界であり、安定領域は境界の内側である）。

Runge-Kutta 法は次数が増えると安定領域が増すという特徴があるが、A 安定にはならない。

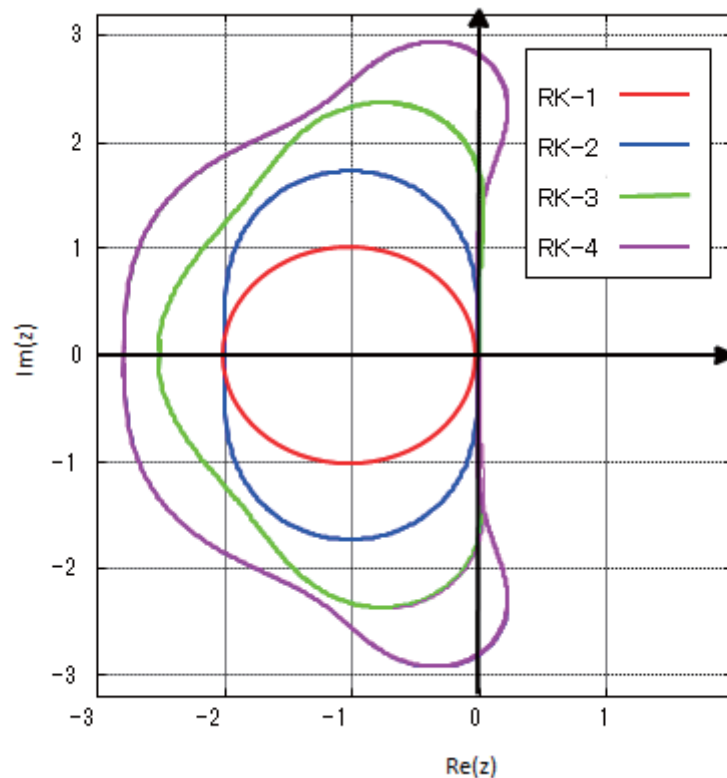


図 2.5 1~4 次の Runge-Kutta 法の安定領域

2.4. 線形多段階法

回路解析の分野では多段数値積分法、数値解析の分野では線形多段階法（Linear Multi Step Method）と呼ばれる解法は、台形積分法や Runge-Kutta 法が、現在と 1 ステップ前の計算結果だけを使用することに対し、数ステップ前の計算結果を利用する手法である。

2.4.1. 線形多段階法の一般形式

線形多段階法は以下の式で表される。

$$x_{n+1} = \sum_{i=0}^k a_i x_{n-i} + h \sum_{i=1}^k b_i f(x_{n-i}, t_{n-i}) \quad (2-32)$$

ここで、 $b_1=0$ であれば、式は陽解法であるが、 $b_1 \neq 0$ であれば、陰解法となる。

ワイエルシュトラウス (Weierstrass) の多項式近似定理によれば、有限区間の連続関数はべき級数の多項式関数で近似できることから、閉区間 $[t_{n-k}, t_{n+1}]$ における $x(t)$ は k 次の多項式で近似する事ができる。すなわち

$$x(t) = c_0 + c_1 t + c_2 t^2 \cdots + c_k t^k \quad (2-33)$$

である。

この多項式より多段階法の式を満足するように係数を決定する。すなわち時間軸の原点を

t_n とするならば、 $t_{n-i} = -ih$ であるから、 $p \leq k$ である p に対し、

$$x_{n-i} = c_0 + c_1(-ih) + c_2(-ih)^2 \cdots + c_k(-ih)^p = \sum_{m=0}^p c_m (-ih)^m \quad (2-34)$$

また、 $\dot{x} = f(x, t)$ より、

$$f(x_{n-i}, t_{n-i}) = c_1 + 2c_2(-ih) + 3c_3(-ih)^2 \cdots + pc_p(-ih)^{p-1} = \sum_{m=1}^p mc_m (-ih)^{m-1}$$

$$i = -1, 0, 1, \dots, n, \quad p = 2, 3, \dots, k \quad (2-35)$$

であり、

$$x_{n+1} = c_0 + c_1(h)t + c_2(h)^2 \cdots + c_k(h)^p = \sum_{m=0}^p c_m h^m \quad (2-36)$$

であるから、式(2-34)、式(2-35)、式(2-36)を、式(2-32)に代入すると、

$$\sum_{m=0}^p c_m h^m = \sum_{i=0}^k \sum_{m=0}^p [a_i c_m (-ih)^m] + h \sum_{i=-1}^k \sum_{m=1}^p [mb_i c_m (-ih)^{m-1}] \quad (2-37)$$

となり、詳細は参考文献[30]に譲るが、式(2-38)、式(2-39)が成立する。

$$\sum_{i=0}^k a_i = 1 \quad (2-38)$$

$$\sum_{i=0}^k (-i)^p a_i + p \sum_{i=-1}^k (-i)^{p-1} b_i = 1 \quad p = 1, 2, \dots, k \quad (2-39)$$

式(2-39)は $2k+3$ 個の未知数からなる $k+1$ 個の方程式から構成されているため、解は一意には定まらない。しかし $k+2$ 個の係数を適当に指定することで、残りの $k+1$ 個の係数を定めることができ、 $k+2$ 個の係数の指定の仕方により、線形多段階法は幾つかの手法に分類される。

2.4.2. 線形多段階法の絶対安定領域

$$x_{n+1} = \sum_{i=0}^k a_i x_{n-i} + h \sum_{i=1}^k b_i f(x_{n-i}, t_{n-i}) \quad (2-40)$$

で与えられる線形多段階法にテスト方程式を適用すると、

$$(1 - h\lambda b_{-1})x_{n+1} - \sum_{i=0}^k (a_i + h\lambda b_i)x_{n-i} = 0 \quad (2-41)$$

である、この式(2-41)の一般解は[30]に従うと、

$$x_n = c_1 z_1^n + c_2 z_2^n + \dots + c_{k+1} z_{k+1}^n \quad (2-42)$$

で与えられる。ただし、

$$\{z_1, z_2, \dots, z_{k+1}\} \quad (2-43)$$

は次の特性多項式の $k+1$ 個の根である。

$$(1 - h\lambda b_{-1})z^{k+1} - \sum_{i=0}^k (a_i + h\lambda b_i)z^{k-i} = 0 \quad (2-44)$$

従って、式(2-41)が安定であるとは、式(2-44)が有限であることであり、 z が単位円内、であることが必要である。

以上より、式(2-44)を書き換えて、以下の $h\lambda$ に関する式とし、 z の境界は単位円であるから、 $e^{j\theta}$ を代入する事で、絶対安定領域の境界が求まる。

$$h\lambda \sum_{i=1}^k b_i z^{k-i} = z^{k+1} - \sum_{i=0}^k a_i z^{k-i} \quad (2-45)$$

$$h\lambda = \frac{z^{k+1} - \sum_{i=0}^k a_i z^{k-i}}{\sum_{i=1}^k b_i z^{k-i}} \quad (2-46)$$

が絶対安定領域の境界となり、安定領域はこの内側となる。

2.4.3. Adams-Bashforth 法

式(2-32)において、 $a_0=1$ 、 $a_1=a_2\dots a_k=0$ 、 $b_{-1}=b_k=0$ とした、式(2-47)の形のものを Adams-Bashforth 法という

$$x_{n+1} = x_n + h \sum_{i=0}^{k-1} b_i f(x_{n-i}, t_{n-i}) \quad (2-47)$$

$k=1$ の場合は、前進 Euler 法であり、Adams-Bashforth 法は陽解法である。

Adams-Bashforth 法の係数 (bi) は以下に計算されている。

表 2-3 Adams-Bashforth 法の係数

次数	k	b ₀	b ₁	b ₂	b ₃	b ₄
1	1	1				
2	2	3/2	-1/2			
3	3	23/12	-16/12	5/12		
4	4	55/24	-59/24	37/24	-9/24	
5	5	1901/720	-2774/720	2616/720	-1274/720	251/720

2.4.4. Adams-Bashforth 法の絶対安定領域

Adams-Bashforth 法では $a_i=0$, $0<i$, $b_{-1}=b_k=0$, $a_0=1$ より、式(2-45)は、

$$h\lambda = \frac{z^{k+1} - z^k}{\sum_{i=0}^{k-1} b_i z^{k-i}} \quad (2-48)$$

であるが、これを[24][25]を参考に変換すると

$$h\lambda = \frac{e^{i\theta} - 1}{\sum_{j=0}^{k-1} \gamma_j (1 - e^{-i\theta})^j} \quad \gamma_0 = 1, \gamma_1 = \frac{1}{2}, \gamma_2 = \frac{5}{12}, \gamma_3 = \frac{3}{8}, \gamma_4 = \frac{251}{720}, \gamma_5 = \frac{95}{288} \quad (2-49)$$

となる。k=1~3 を図 2.6 に k=4~5 を図 2.7 に示す。

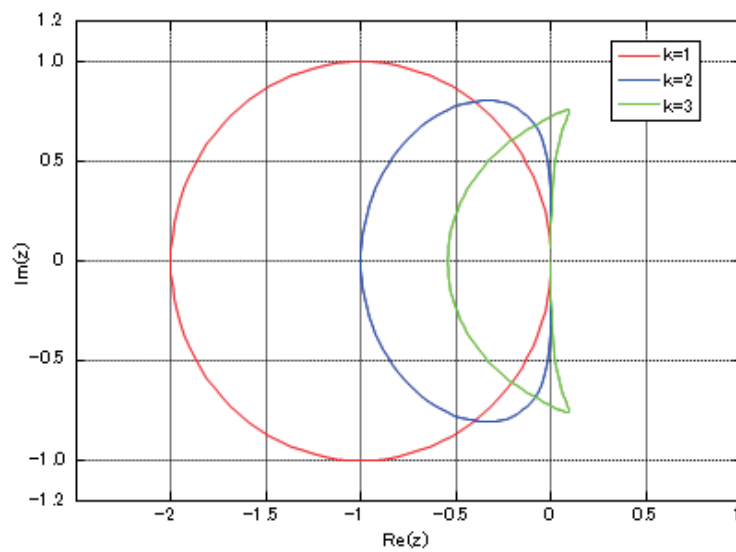


図 2.6 Adams-Bashforth の安定領域 (1~3 次)

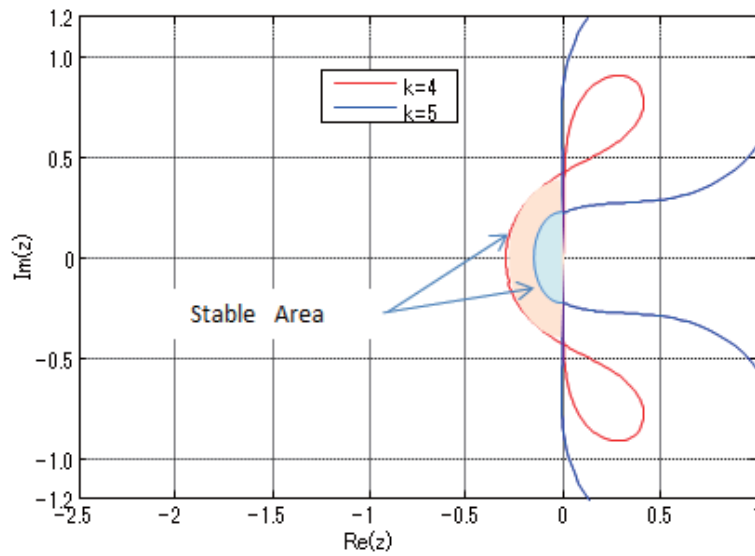


図 2.7 Adams-Basforth 法の安定領域 (4, 5 次)

2.4.5. Adams-Moulton 法

式(2-32)において $a_0=1$ 、 $a_1=a_2\ldots a_k=0$ 、 $b_{k-1}=b_k=0$ 、 $b_{k-1}\neq 0$ としたものを Adams-Moulton 法という。

なお、本手法において $k=1$ では後退 Euler 法、 $k=2$ では台形積分法と一致する。すなわち、Adams-Moulton 法は陰解法である。

Adams-Moulton 法の係数 (b_i) は以下に計算されている。

表 2-4 Adams-Moulton 法の係数

次数	k	b_{-1}	b_0	b_1	b_2	b_3
1	1	1				
2	1	1/2	1/2			
3	2	5/12	8/12	-1/12		
4	3	9/24	19/24	-5/24	1/24	
5	4	251/720	646/720	-264/720	106/720	-19/720

2.4.6. Adams-Moulton 法の絶対安定領域

Adams-Moulton 法では、 $a_0=1$ 、 $a_1=a_2\ldots a_k=0$ 、 $b_{k-1}=b_k=0$ 、 $b_{k-1}\neq 0$ であるから、式(2-45)は

$$h\lambda = \frac{z^{k+1} - z^k}{b_{-1}z^{k+1} + \sum_{i=0}^{k-2} b_i z^{k-i}} \quad (2-50)$$

同じく、[24][25]を参考に式(2-50)を変換すると、

$$h\lambda = \frac{1 - e^{-i\theta}}{\sum_{j=0}^k \gamma_j^* (1 - e^{-i\theta})^j} \quad \gamma_0^* = 1, \gamma_1^* = \frac{-1}{2}, \gamma_2^* = \frac{-1}{12}, \gamma_3^* = \frac{-1}{24}, \gamma_4^* = \frac{-19}{720}, \gamma_5^* = \frac{-3}{160} \quad (2-51)$$

であり、安定領域を図示すると図 2.8 になる。なお、k=2 は台形法であり、安定領域は左半平面であるから図 2.8 には記載していない。

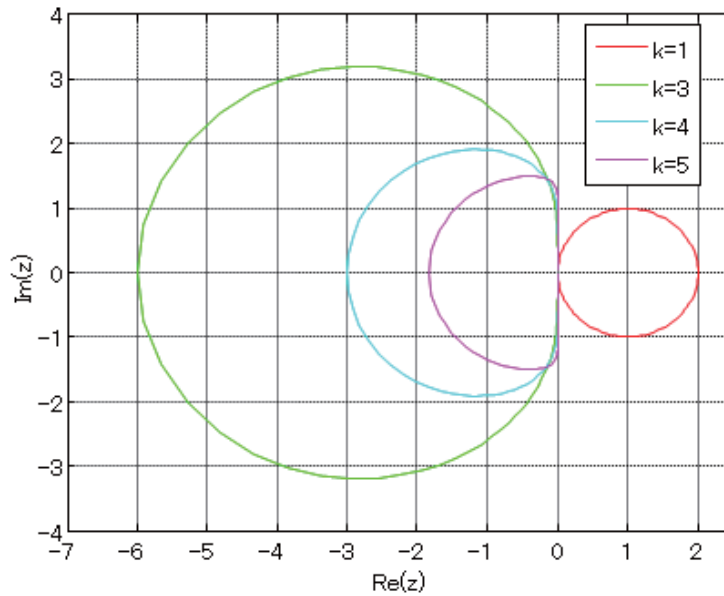


図 2.8 Adams-Moulton 法の安定領域

2.4.7. 後退微分公式(BDF/Gear 法)

式(2-32)において $a_k=0$ 、 $b_0=b_1 \dots b_k=0$ 、 $b_{-1} \neq 0$ とし、過去の値と現在の微分値のみを利用した手法は後退微分公式 (BDF(Backward Differentiation formula)) あるいは Gear 法として知られている。

$$x_{n+1} = \sum_{i=0}^{k-1} a_i x_{n-i} + h b_{-1} f(x_{n+1}, t_{n+1}) \quad (2-52)$$

本手法における k=1 は後退 Euler 法である。

本手法は、陰解法のなかでも特に計算安定性に優れているため、回路計算においては SPICE を始めとして採用事例が多い。

表 2-5 後退微分公式 (BDF/Gear 法) の係数

次数	k	b ₋₁	a ₀	a ₁	a ₂	a ₃	a ₃
1	1	1	1				
2	2	2/3	4/3	-1/3			
3	3	6/11	18/11	-9/11	2/11		
4	4	12/25	48/25	-36/25	16/25	-3/25	
5	5	60/137	300/137	-300/137	200/137	-75/137	12/137

2.4.8. 後退微分方程式の絶対安定領域

後退微分方程式の安定領域は次式で与えられる[25]。

$$h\lambda = \frac{z^{k+1} - \sum_{j=0}^{k-1} a_j z^{k-j}}{b_{-1} z^{k+1}} \quad (2-53) \quad h\lambda = \sum_{j=1}^k \frac{1}{j} (1 - e^{-i\theta}) \quad (2-54)$$

k=1~6 までを図 2.9 にプロットする。

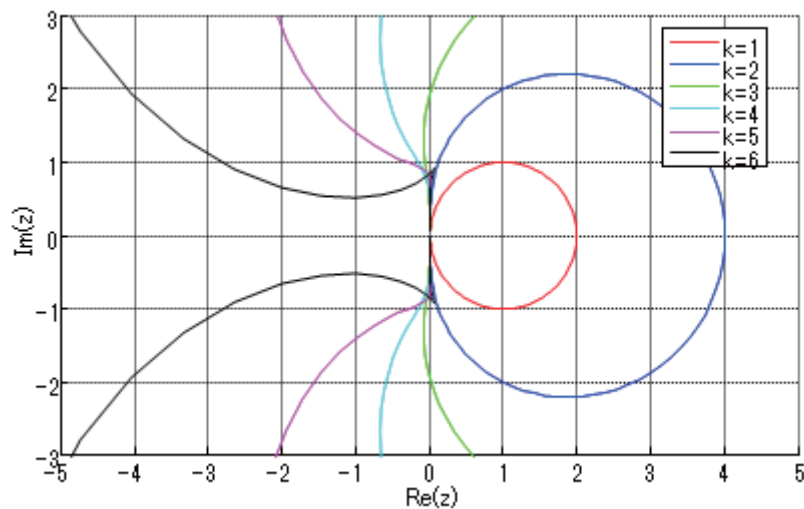


図 2.9 後退微分公式の安定領域

後退微分公式は k=2 までは A 安定であるが、k が増えるに従い左半平面にかなりの不安定領域が残る。後退微分公式を用いる第一の意義は、広い絶対安定領域にあるため、実用的には 5 次以下で用いられる例が多い。

2.5. 結論

以上、代表的な常微分方程式の数値解法を調べた。回路計算においては、EMTP が台形積分法、SPICE が台形積分法と後退 Euler を含む後退微分公式 (BDF/Gear 法) を採用しているが、これらは計算安定性を重視した結果である。しかし、計算安定性については、個々の回路によって方程式の持つ固有値は変わるため、一概に計算安定性に優れた手法が回路計算における絶対条件とは言えない。

パワーエレクトロニクス分野、すなわちスイッチング回路を計算するにあたり、最適な数値解法は、第 3 章で論じる回路方程式の構築手法ならびに第 4 章で論じるスイッチング回路特有の課題とあわせて考える必要があると思われるため、結論はそこで下すことにする。

3章 回路方程式の構築手法

現在までに、様々な回路方程式の構築手法が提案されている。本章では、代表的な手法について、その特徴を明らかにする。

3.1. 回路方程式の構築手法の歴史

電子回路のシミュレーション技術については、EMTP および PSCAD/EMTDC に代表される、電送系や高電圧回路を対象とした過渡解析シミュレータ系統と、SPICE を代用とする、LSI 設計を対象とした、過渡解析に加え、DC 解析、AC インピーダンス解析などを様々な解析を行う汎用シミュレータ系に大別できるが、不思議なことに、両者は共通の技術的要素をもちながら、直接の交流の形跡がない。すなわち、1969 年の EMTP の論文[32]は SPICE の源流とされ EMTP と同様に節点解析法を用いた IBM の ECAP (1965 年)の記述はなく、L.W.Nagel の高名な SPICE2 の博士論文[5]には、EMTP の作者であり研究を先行させていた H.W.Dommel の文献が含まれない。この点は興味深いものであるが、本章は回路方程式の定式化手法を扱うため、様々な定式化手法が試みられた SPICE 系統を中心に、その歴史を概観する。

回路用のシミュレータの歴史は、第 2 次大戦前に開発された TNA (Transient network analyzer) まで遡ることができる[33]。TNA は巨大なアナログ計算機であり、真空管により構成された加算、乗算、積分器により常微分方程式を解くことで、高電圧の電送系のモデルを計算する事が出来た。この装置がアナログ計算機であることは、解を実時間で解くことができる、リアルタイム・シミュレータであることを意味する。つまり、最も初期の回路シミュレーション装置は、オフライン・シミュレータではなく、リアルタイム・シミュレータであったことは興味深い。しかし、このようなアナログ計算機は、巨大で精密な装置であり、維持コストが高額であることより、デジタル計算機を用いたシミュレータに代替される事となる。

デジタルコンピュータによる回路解析の嚆矢は、状態変数法を用いた TAP (1962 年 IBM) あたりと考えられる。さらに同じ状態変数法の CIRCUS (1965 年 BOING) が続き、ASATAP (1973 年 IBM) ではタブロー法が試みられている。節点解析法は前述の ECAP (1965 年 IBM)、後に SPICE に改名される CANCER (1971 年 UCB) と続き、修正節点解析法を採用した SPICE2 (1975 年 UCB) へと続く[34][35]。SPICE2 はソースコードを公開したため、1980 年代にはこれを元にした多くの PC 用の商用回路シミュレータが輩出している。なお、日本で著名であった商用回路シミュレータの OrCAD も SPICE 系の一つであり OrCAD 社の設立は 1985 年である。

なお EMTP 系では EMTP の普及とユーザーサポートのために、EMTP Development

Coordination Group が 1982 年に設立されており、Manitoba HVDC Reserch Centor がグラフィカルな環境を持ち、UNIX 上で稼働する PSCAD/EMTDC をリリースしたのが 1990 年代初めとされている [33]。

3.2. 回路方程式に使用する基本的な公式

本章では、様々な回路方程式の構築手法を検討するが、いずれの方式においても以下の基本的な法則を用いて式の構築を行う。

I. キルヒホッフの電流法則 (Kirchoff's Current Raw: KCL)

任意の節点において、流れ込む向きを正（又は負）と統一するとき、流れ込む電流の総和は 0 となる。

II. キルヒホッフの電圧則 (Kirchoff's Voltage Raw: KVL)

任意の閉路において、電圧の向きを一方向に取ったとき、閉路に沿った各素子の電圧 V_i の総和は 0 となる。

III. 枝構成式 (Branch Consisitude Equation: BCE)

枝電圧と枝電流の関係を表す式

例: $V_b = R_b I_b$

$$C \frac{d}{dt} v_b = i_b$$

なお、本論文においては、KCL、KVL、BCE との略語を用い、それぞれキルヒホッフの電流則、キルヒホッフの電圧則、枝構成式を意味するものとする。

3.3. 微分方程式の扱い

キャパシタやインダクタなど、特性が微分方程式で表される素子より構成される回路の全体の方程式、すなわち回路方程式は、やはり微分方程式となる。これを過渡応答シミュレーション、すなわち回路の時間応答特性をデジタル計算機で計算するためには、微分方程式を離散化された差分方程式に変換する必要がある。この変換には、個々の素子単位で差分方程式を構成した後、回路方程式を差分方程式として構築する手法と、回路全体の方程式を連続系の常微分方程式として定式化し、常微分方程式の数値解法を用いて連続系の常微分方程式を計算する手法がある。

本来、これは微分方程式で表される素子の離散化手法の差異であるが、実際には回路方程式の構築手法と密接に関連しており、節点解析法、修正節点解析法、およびスパースタブロー法であれば、前者の離散化手法、カットセット解析などの状態方程式を用いる場合は、後者の離散化手法が組み合わせられ用いられるのが通例である。

3.4. サンプル RLC 回路の設定

この章では様々な回路定式化手法を検討するが、その比較を容易にするため、同一のサンプル回路に対する定式化を具体的に行う。回路図を図 3.1 に示す。

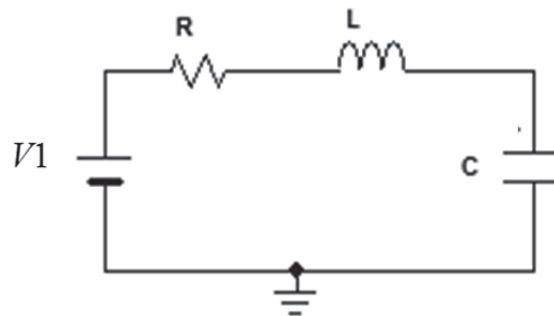


図 3.1 サンプル RLC 回路の回路図

なお、本回路において、以下のパラメータを使用し、V1 を 0~10V にステップ状に変化させたときのキャパシタ電圧は図 3.2 の応答となる。本来、すべての定式化手法で同様の結果となるべきであるが、実際には 0 秒直後の 1~2 ステップの値が異なっている。これは、離散化手法に応じた正しい初期値の設定に対する知見が足りず、すべて 0 としている事が理由であると考えられる。今後の研究で詳細を明らかにしたい。

$L=10\text{mH}$

$C=1\text{mF}$

$R=1\Omega$

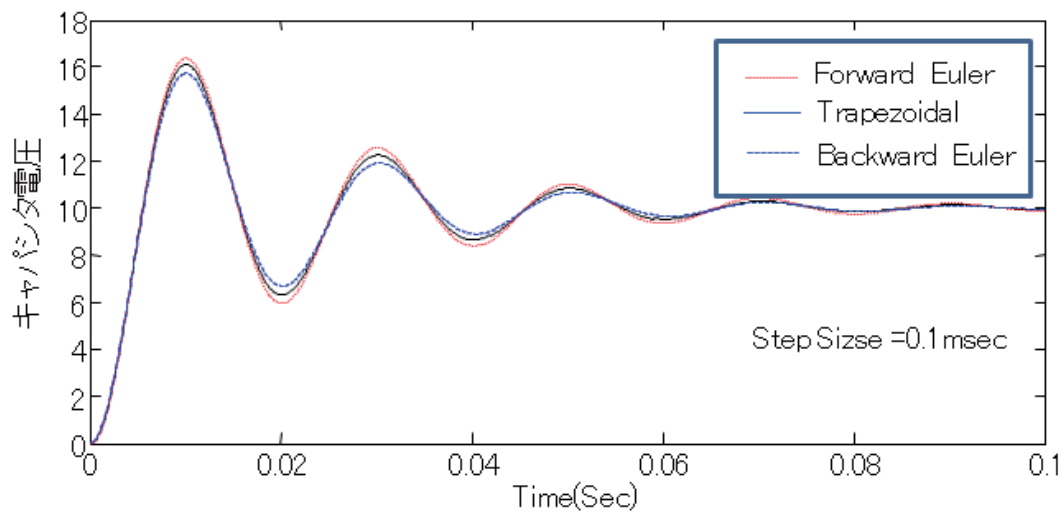


図 3.2 サンプル RLC 回路のステップ応答

3.5. 節点解析法 (Nodal Analysis Method)

3.5.1. 基本的な考え方

現在、最も一般的に知られている回路解析法が節点解析法である。初期の SPICE (SPICE-1 と呼ばれる CANCER)、および EMTP において用いられており、EMTP の作者である Dommel の手法とも呼ばれている[33]。

本手法は回路における素子の接続点である節点 (Node) に対して、KCL を適用して得られるものである[36]。

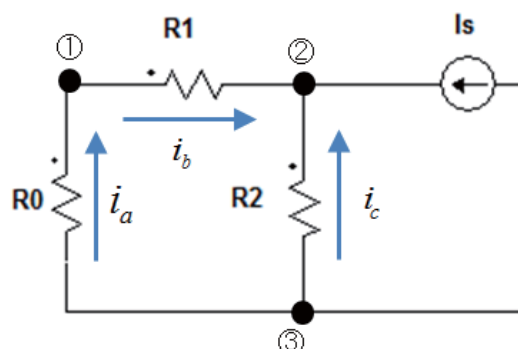


図 3.3 節点 (Node) と枝(Branch) の関係

図 3.3 のような抵抗と電流源からなる回路を考えた場合、節点(Node)は3個、枝(Branch)

は 4 本であるが、節点解析では図 3.3 の 3 個の節点すなわち、①、②および③について、KCL を適用し、節点電圧を未知数とした方程式を構築する。

各枝に流れる電流を i_a, i_b, i_c および定電流源の電流を I_s とすると、

$$\text{節点①} : i_a - i_b = 0 \quad (3-1)$$

$$\text{節点②} : i_b + i_c + I_s = 0 \quad (3-2)$$

$$\text{節点③} : -i_a - i_c = 0 \quad (3-3)$$

節点①の電圧を v_1 、②の電圧を v_2 、③の電圧を v_3 とし、BCE としてオームの法則を適用すると、

$$R_0 i_a = v_3 - v_1 \quad (3-4)$$

$$R_1 i_b = v_1 - v_2 \quad (3-5)$$

$$R_2 i_c = v_3 - v_2 \quad (3-6)$$

である。

式(3-1)~(3-3)を利用して i_a 、 i_b 、 i_c を消去すると、

$$\frac{1}{R_0}(v_3 - v_1) - \frac{1}{R_1}(v_1 - v_2) = 0 \quad (3-7)$$

$$\frac{1}{R_1}(v_1 - v_2) + \frac{1}{R_2}(v_3 - v_2) = -I_s \quad (3-8)$$

$$-\frac{1}{R_0}(v_3 - v_1) - \frac{1}{R_2}(v_3 - v_2) = 0 \quad (3-9)$$

であり、式(3-7)~(3-9)の連立方程式を行列式で表現すると、

$$\begin{bmatrix} \frac{1}{R_0} + \frac{1}{R_1} & -\frac{1}{R_1} & -\frac{1}{R_0} \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} \\ -\frac{1}{R_0} & -\frac{1}{R_2} & \frac{1}{R_0} + \frac{1}{R_2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ I_s \\ 0 \end{bmatrix} \quad (3-10)$$

となる。

このように節点解析では、未知数を電圧とし、抵抗値の逆数を係数とする連立方程式が得られるので、抵抗値はコンダクタンス値とした方が、表現しやすい。従って、通常、式(3-10)はコンダクタンス値を用いて、

$$\begin{bmatrix} G_0 + G_1 & -G_1 & -G_0 \\ -G_1 & G_1 + G_2 & -G_2 \\ -G_0 & -G_2 & G_0 + G_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ I_s \\ 0 \end{bmatrix} \quad (3-11)$$

とされる。

3.5.2. 具体的な方程式の構築法（スタンプ）

式(3-11)を見ると、コンダクタンスで構成される行列の要素に一定のルールがある事がわかる。

すなわち、行列の対角要素である a_{ii} は、節 i に接続されている枝のコンダクタンス要素の総和、それ以外の要素である a_{ij} ($i \neq j$) は、節 i と節 j を接続する枝のコンダクタンス要素の符号を負としたものである。

このようなルールに基づき、回路方程式を構築する手法は“スタンプ”と呼ばれている。回路を構成する個々の素子に対し、“スタンプ”をそれぞれ設定する事により、素子が接続される節点に対応する行列の要素に対し、スタンプを押すように、値を加算する事で、回路方程式を容易に構築することができる。

“スタンプ”の具体的な例を以下に示す。G ジーメンズ(S)のコンダクタンス（抵抗器）が節点 i と節点 j の間に接続された場合は、

	i 列	j 列	定数ベクトル
i 行	G	-G	0
j 行	-G	G	0

節点 j から節点 i に $I_s(A)$ で流れ込む定電流源は

	i 列	j 列	定数ベクトル
i 行	0	0	I_s
j 行	0	0	$-I_s$

である。

SPICE では、キャパシタ、インダクタ、トランス（カップルド・インダクタ）など、回路を構成する要素の “スタンプ” が設定されており、接続される節点に応じた位置に “スタンプする” 事で、回路方程式を容易に得ることができる。

3.5.3. 基準電位の未知変数からの削除

以上のように、回路方程式が構築されるが、実際には、式(3-11)の方程式はコンダクタン

ス要素で構成された左辺の行列が正則でないため解くことができない。これは物理的には、基準電位を定めなければ、節点電位は無数に存在しうることからも理解できる。

従って、いずれかの節点の電位を基準電位とし、基準電位に対応する行と列を削除した行列式を回路方程式として使用する事が普通である。式(3-11)で、節点③の電位を基準とし、0Vとした場合は、3行目と3列目を削除した、式(3-12)となる。

$$\begin{bmatrix} G_0 + G_1 & -G_1 \\ -G_1 & G_1 + G_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ I_s \end{bmatrix} \quad (3-12)$$

3.5.4. 動的要素の扱い（キャパシタ）

線形の理想キャパシタは枝の両端電圧（枝電圧）を v 、枝に流れる（枝電流）を i とすると、

$$C \frac{d}{dt} v = i \quad (3-13)$$

と表せる。

この式は常微分方程式であるから、第2章で示したすべての数値計算手法を用いて差分方程式に変換する事ができる。

ここで、離散化された枝電圧を v_n 、同じく枝電流を i_n とし、後退 Euler 法で式(3-13)を計算すると、

$$v_{n+1} = v_n + h \frac{d}{dt} v_{n+1} \quad (3-16)$$

$$C \frac{d}{dt} v_{n+1} = i_{n+1} \quad (3-17)$$

より、

$$v_{n+1} = v_n + \frac{h}{C} i_{n+1} \quad (3-18)$$

$$i_{n+1} = \frac{C}{h} v_{n+1} - \frac{C}{h} v_n \quad (3-19)$$

となる。

ここで、 i_{n+1} と v_{n+1} の関係は、同時刻の枝電圧と枝電流の関係であるから、抵抗要素（コンダクタ）により結び付けられると考えることができるため、仮想的な抵抗 G_{eqC} を導入する。

さらに、 i_{n+1} に対し、 v_n はすでに計算された既知の値であるから、 i_{n+1} とは無関係な独立

電流源として扱い、 I_{eqC} を定義する。

すなわち、

$$G_{eqC} = \frac{C}{h} \quad I_{eqC} = -\frac{C}{h} v_n \quad (3-20) \quad (3-21)$$

と置くと、

$$i_{n+1} = G_{eqC} v_{n+1} + I_{eqC} \quad (3-22)$$

と理解する事ができ、これは回路図で示すと、図 3.4 である。

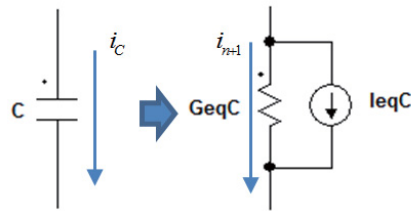


図 3.4 線形キャパシタの等価回路

従って、キャパシタの“スタンプ”は抵抗要素と定電流源の組み合わせで

	i 列	j 列	定数ベクトル
i 行	G_{eqC}	$-G_{eqC}$	$-I_{eqC}$
j 行	$-G_{eqC}$	G_{eqC}	I_{eqC}

とすることができる。

台形積分法を用いた場合は、

$$v_{n+1} = v_n + \frac{h}{2} \left(\frac{d}{dt} v_{n+1} + \frac{d}{dt} v_n \right) \quad (3-23)$$

より、

$$v_{n+1} = v_n + \frac{h}{2C} (i_{n+1} + i_n) \quad (3-24)$$

$$i_{n+1} = \frac{2C}{h} v_{n+1} - \frac{2C}{h} v_n - i_n \quad (3-25)$$

であり、

$$G_{eqC} = \frac{2C}{h} \quad (3-26)$$

$$I_{eqC} = -\frac{2C}{h} v_n - i_n \quad (3-27)$$

とすることができる。以上は、式の表現は異なるが、Dommel の論文[31]や[37]が詳しい。

参考までに、さらに BDF/Gear 法の 2 次の場合を記すと、

$$v_{n+1} = \frac{4}{3}v_n - \frac{1}{3}v_{n-1} + \frac{2h}{3} \frac{d}{dt}v_{n+1} \quad (3-26)$$

であるから、

$$v_{n+1} = \frac{4}{3}v_n - \frac{1}{3}v_{n-1} + \frac{2h}{3C}i_{n+1} \quad (3-27)$$

$$i_{n+1} = \frac{3C}{2h} \left(v_{n+1} - \frac{4}{3}v_n + \frac{1}{3}v_{n-1} \right) \quad (3-28)$$

あるので、

$$G_{eqC} = \frac{3C}{2h} \quad (3-29)$$

$$I_{eqC} = -\frac{2C}{h}v_n + \frac{1}{2}i_n \quad (3-30)$$

として扱うことができる。

なお、計算手法として前進 Euler 法を用いることも可能であるが、第 2 章に述べたように、前進 Euler 法は計算の安定度が低く、節点解析法を用いる場合は、もともと行列が左にある関係上、計算負荷が軽いというメリットもない。従って、節点解析法において離散化手法に前進 Euler 法が用いられることはない。

3.5.5. 動的要素の扱い（インダクタ）

線形の理想インダクタは、同じように両端電圧（枝電圧）を v 、流れる電流(枝電流)を i とすると、

$$L \frac{d}{dt}i = v \quad (3-31)$$

であるから、

後退 Euler 法では、

$$i_{n+1} = i_n + h \frac{d}{dt}i_{n+1} \quad (3-32)$$

より、

$$G_{eqL} = \frac{h}{L} \quad (3-33)$$

$$I_{eqL} = i_n \quad (3-34)$$

$$i_{n+1} = G_{eqL} v_{n+1} + I_{eqL} \quad (3-35)$$

と、同様に表せ、等価回路も図 3.5 となる。

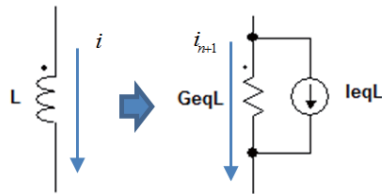


図 3.5 線形インダクタの等価回路

台形積分法の場合は、

$$G_{eqL} = \frac{h}{2L} \quad (3-36)$$

$$I_{eqL} = \frac{h}{2L} v_n + i_n \quad (3-37)$$

である。

以上のようにインダクタはキャパシタと同じく等価コンダクタと等価電流源で表すことができるので、同じように“スタンプ”を作成する事ができ、以下となる。

	i 列	j 列	定数ベクトル
i 行	GeqL	-GeqL	-leqL
j 行	-GeqL	GeqL	leqL

3.5.6. サンプル回路の定式化

3.4 節で導入したサンプル回路のキャパシタとインダクタは式(3-22)と式(3-35)で電流源とコンダクタに変換されるため、図 3.6 となる。

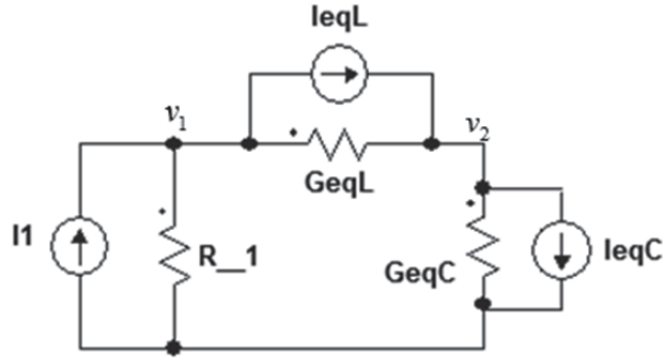


図 3.6 サンプル RLC 回路の節点解析法の場合の等価回路

$$\begin{bmatrix} \frac{1}{R} + GeqL & -GeqL \\ -GeqL & GeqL + GeqC \end{bmatrix} \begin{bmatrix} v_1(n+1) \\ v_2(n+1) \end{bmatrix} = \begin{bmatrix} Is - IeqL(n+1) \\ IeqL(n+1) - IeqC(n+1) \end{bmatrix} \quad (3-38)$$

キャパシタ、インダクタともに台形積分法で離散化した場合は、

$$GeqC = \frac{2C}{h} \quad IeqC(n+1) = -\frac{2C}{h} v_2(n) - I_c(n) \quad (3-39) \quad (3-40)$$

$$I_c(n) = \frac{2C}{h} (v_2(n) - v_2(n-1)) - I_c(n-1) \quad (3-41)$$

$$GeqL = \frac{h}{2L} \quad IeqL(n+1) = \frac{h}{2L} (v_1(n) - v_2(n)) + I_L(n) \quad (3-42) \quad (3-43)$$

$$I_L(n) = \frac{h}{2L} (v_1(n) - v_2(n) + v_1(n-1) - v_2(n-1)) + I_L(n-1) \quad (3-44)$$

以上の計算式を具体的に計算する MATLAB の M-Script を付録リスト 1 に示す。

3.6. 修正節点解析法 (Modified Nodal Analysis Method)

節点解析法は“スタンプ”を利用することで、容易に回路方程式が構築できるため、広く用いられてきた手法である。しかし、KCL のみを用いる定式化のため、

◇電圧源を直接取り扱えない

◇枝電流が直接計算されない⇒インダクタ電流が直接計算されない
という問題がある。

また、枝として所定のコンダクタンス値を持つとして定義した素子を一時的にショートさせる（コンダクタンス値=∞）などの処理が行いにくいという点でも問題がある。

そのような問題を解消するものとして考案されたものが、修正節点解析法である。これは SPICE2 に用いられ、博士論文[5]で公開されたことより、今日では SPICE の定式化手法として広く知られおり、SPICE を元にした回路シミュレータの解説書[36]や回路計算の教科書[38]などで詳細を知ることができる。

3.6.1. 基本的な考え方

節点解析法では未知変数として節点電圧のみを扱ったが、修正節点解析法では、これに加え電圧源を流れる電流、任意の枝電流、電源により制御される電流源を未知変数に加える。

その結果、回路法手式は

$$\begin{bmatrix} Y & B \\ C & D \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} I_s \\ V_s \end{bmatrix} \quad (3-45)$$

となる。

ここで、

v : 節点電圧変数ベクトル

i : 付加した枝電流変数ベクトル

Y : アドミタンス行列（節点解析法と同じ）

B : 付加した枝電流変数の KCL 方程式への組み込みのための係数（ $Yv + Bi = Is$ ）

Is : 独立電流源

Vs : 独立電圧源

である。

Y 行列の構成は、節点解析法と同様なコンダクタンス要素である。 B 行列は、KCL へ枝電流変数(=i)を組み込むため、対象となる節（行番号）に対し、枝電流変数が流れ出ていれば 1、流れ込んでいれば-1、接続が無ければ 0 とするが、実際には以下に説明するスタンプにより C 、 D 行列と共に構成される。

3.6.2. 具体的な方程式の構築法-1（等価電流源方式）

B, C, D 行列の構成に使用するスタンプは 2 種類ある。先ず、参考文献[38]による方法では、節点- i と節点- j に接続された枝の電流を変数とし、これが変数ベクトルの r 行目に配置される場合は、以下のスタンプを節点解析で用いた電圧変数に対するスタンプに追加して使用

する構成する。

抵抗素子の場合は、

	i 列	j 列	r 列	定数ベクトル
i 行	0	0	+1	0
j 行	0	0	-1	0
r 行	G	-G	-1	0

キャパシタンス素子の場合は、

	i 列	j 列	r 列	定数ベクトル
i 行	0	0	+1	0
j 行	0	0	-1	0
r 行	GeqC	-GeqC	-1	-IeqC

インダクタンス素子の場合は

	i 列	j 列	r 列	定数ベクトル
i 行	0	0	+1	0
j 行	0	0	-1	0
r 行	GeqL	-GeqL	-1	-IeqL

となる。

i、j の要素は、式(3-45)の Y 行列を構成し、r 列、r 行の要素が、B、C、D 行列となる。

なお、ここで用いる GeqC、GeqL、IeqC、および IeqL は節点解析法で用いた、等価コンダクタンスと等価電流源である。

3.6.3. 具体的な方程式構築法-II（等価電圧源方式）

これ対し、SPICE 系の解説書[36]で紹介されている手法では、インダクタを図 3.7 の等価抵抗+等価電圧源で表し、式(3-46)を用いる。

$$v_{n+1} = r_{eqL} i_{n+1} + V_{eqL} \quad (3-46)$$

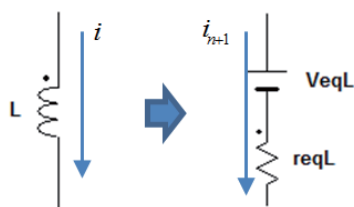


図 3.7 線形インダクタの等価回路

ここで用いるスタンプは以下である。

	i 列	j 列	r 列	定数ベクトル
i 行	0	0	1	0
j 行	0	0	-1	0
r 行	+1	-1	-reqL	Veql

ここでは、インダクタを表す差分方程式の電流と電圧の関係が逆になっているため、差分方程式の導出をあらためて行う必要があるが、キャパシタ、インダクタともに計算ステップである h が分母に表れるので、行列のバランスは良くなる。

以下に、後退 Euler 法と台形積分法における等価抵抗 $reqL$ と等価電圧源 $Veql$ を示す。

後退 Euler 法

$$r_{eqL} = \frac{L}{h} \quad (3-47) \quad V_{eqL} = -\frac{L}{h} i_n \quad (3-48)$$

台形積分法

$$r_{eqL} = \frac{2L}{h} \quad (3-49) \quad V_{eqL} = -\frac{2L}{h} i_n - v_n \quad (3-50)$$

3.6.4. サンプル回路の定式化

以下に SPICE 流のインダクタを電圧源とする手法でサンプル RLC 回路の方程式を導く。

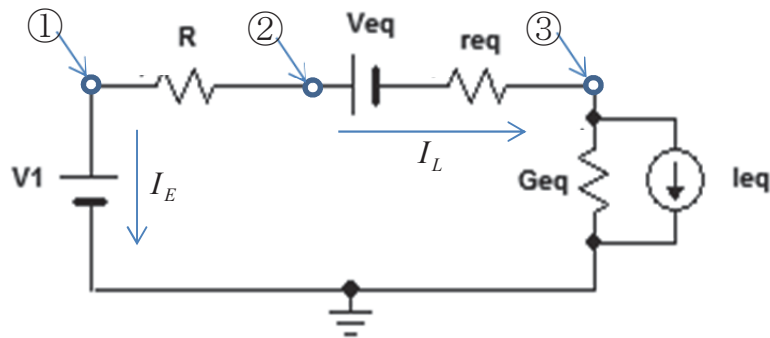


図 3.8 サンプル RLC 回路の節点解析法の場合の等価回路

図 3.8 の等価回路は節点の数が 3 か所に増え、各節点電圧に加え、電圧源およびインダクタに流れる電流を変数とすることにより、行列規模は節点解析法の 2×2 から 5×5 へ大きく拡大するが、電圧源やインダクタに流れる電流の値を直接算出することができる。

$$\begin{bmatrix} G & -G & 0 & 0 & 1 \\ -G & G & 0 & 1 & 0 \\ 0 & 0 & GeqC & -1 & 0 \\ 0 & 1 & -1 & -reqL & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1(n+1) \\ v_2(n+1) \\ v_3(n+1) \\ i_L(n+1) \\ i_E(n+1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -IeqC(n+1) \\ VeqL(n+1) \\ e_1 \end{bmatrix} \quad (3-51)$$

$$GeqC = \frac{2C}{h} \quad (3-52)$$

$$IeqC(n+1) = -\frac{2C}{h} v_2(n) - I_C(n) \quad (3-53)$$

$$I_C(n) = \frac{2C}{h} (v_2(n) - v_2(n-1)) - I_C(n-1) \quad (3-54a)$$

$$I_C(n) = GeqC v_2(n) + IeqC(n) \quad (3-54b)$$

$$reqL = \frac{2L}{h} \quad (3-55)$$

$$VeqL(n+1) = -\frac{2L}{h} i_L(n) - V_L(n) \quad (3-56)$$

$$V_L(n) = \frac{2L}{h} (i_L(n) - i_L(n-1)) - V_L(n-1) \quad (3-57a)$$

$$V_L(n) = VeqL i_L(n) - VeqL(n) \quad (3-57b)$$

以上を MATLAB の m-script で実装したものを付録リスト 2 に示す。

3.7. スパースタブロー法 (Sparse Tableau Method)

節点解析法では、KCL と BCE のみを用いることにより、3.6 節の修正節点法で指摘したような問題があるため、これに対応して開発された手法がスパースタブロー法である[39]。歴史的には、修正節点解析法より早く提案されている。

この手法では、規模は大きいが行列の要素に 0 を多く含むスパース性の強い行列が生成される。スパース行列は効率的な計算手法が様々に研究されているため、規模の大きさほどには計算負荷は高くない。従って、スパース行列の計算手法が鍵となるため、スーパータブロー法と呼ばれる。スパースタブロー法の後に修正節点解析法が開発され、SPICE

が採用したことより、今日では修正節点解析法が主流になっているが、すべての枝電圧、枝電流が直接算出されるなど、優れた特徴がある手法である。

3.7.1. 接続行列の導入

スパースタブロー法では、回路の接続情報として、有向グラフ理論に基づく、接続行列 (Incident Matrix) を利用する。

接続行列とは、回路を番号が振られた節と枝で表した後、

$$a_{ij} = \begin{cases} 1: \text{枝 } j \text{ が節点 } i \text{ から出る向きに接続されている} \\ -1: \text{枝 } j \text{ が節点 } i \text{ から入る向きに接続されている} \\ 0: \text{枝 } j \text{ は節点 } i \text{ に接続されていない} \end{cases}$$
 というルールで作成された行列である。

図 3.9 の回路であれば、

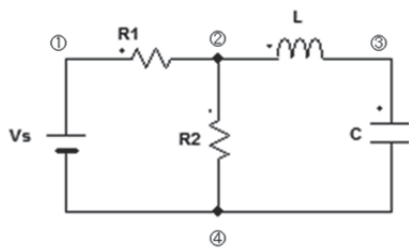
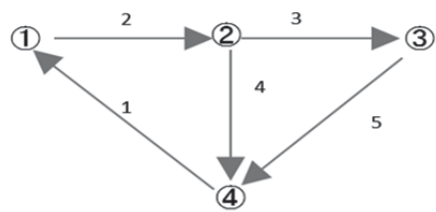


図 3.9 例題の回路



3.10 例題の回路の有向グラフ表現

図 3.10 のような有効グラフとして表すことができる。

$$D = \begin{matrix} & \overbrace{\begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix}}^{\text{枝の数}} \\ \left. \begin{matrix} \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \end{matrix} \right\} \begin{matrix} \text{節の数} \\ \text{節の数} \end{matrix} & \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 \end{bmatrix} \end{matrix}$$

図 3.11 接続行列表現

接続行列は図 3.11 である。

なお、節点解析の場合と同様に、基準点を設けないと解が不定となる関係上、基準となる節点を削除した接続行列を用いることが普通である。(図 3.11 の例であれば第 4 行を削除する)

3.7.2. タブロー方程式の構築

スパースタブロー法では、KCL、KVL、そして BCE のすべてを一つの行列式で記述する。

すなわち、接続行列 D 、枝電流ベクトル I_b 、枝電圧ベクトル V_b 、節点電圧 V_n であるならば、以下の式が成立する。

$$\text{KCL: } DI_b=0 \quad (3-58)$$

$$\text{KVL: } V_b=D^TV_n \quad (3-59)$$

ここで、それぞれの枝は、コンダクタンスで定義する枝 (G 枝) とレジスタンスで定義する枝 (R 枝) とに分類し、それぞれの枝について BCE を定義する。

$$I_b = \begin{bmatrix} I_{bG} \\ \cdots \\ I_{bR} \end{bmatrix} \quad V_b = \begin{bmatrix} V_{bG} \\ \cdots \\ V_{bR} \end{bmatrix} \quad (3-60)$$

$$\text{BCE (G 枝): } I_{bG} + G V_{bG} = I_s \quad (3-61)$$

$$\text{BCE (R 枝): } R I_{bR} + V_{bR} = V_s \quad (3-62)$$

G と R は、それぞれ対応する枝のコンダクタンス成分とレジスタンス成分を対角要素に持つ正方行列である。

$$G = \begin{bmatrix} -G_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & -G_n \end{bmatrix} \quad (3-63)$$

$$R = \begin{bmatrix} -R_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & -R_n \end{bmatrix} \quad (3-64)$$

以上をまとめるとスパースタブロー方程式は、

$$\begin{bmatrix} 0 & D & 0 \\ D^T & 0 & -1 \\ 0 & 1|R & G|1 \end{bmatrix} \begin{bmatrix} V_n \\ I_b \\ V_b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I_s | V_s \end{bmatrix} \quad (3-65)$$

と記述できる。ここで、 $1|R$ は、対応する I_b が G 枝の時は 1 、 R 枝の時は R 行列の対角成分を対角成分に持つ行列であり、 $G|1$ 、 $I_s|V_s$ も同様の意味である。

このタブロー方程式を解くことで、節点電圧だけでなく、すべての枝電流、枝電圧を求めることができ、また上記に示した節点解析法の欠点もすべて解消することができる。

スパースタブロー法の最大の問題点は解くべき逆行列の規模が大きくなることだが、式(3-65)が示すように行列のスパース性は高く、要素が 0 の場合は計算を必要としないため、近年研究が進んでいるスパース行列に対する効率的な解法が利用できる。

3.7.3. サンプル回路の定式化

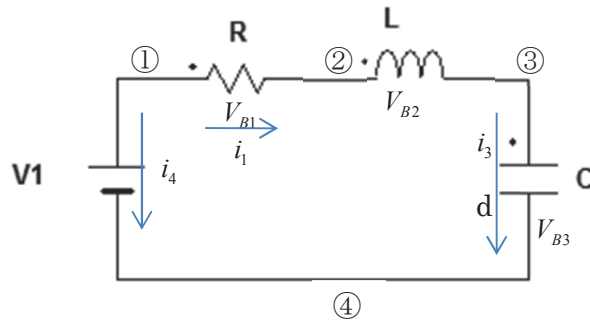


図 3.12 サンプル回路に対する枝電流／枝電圧の定義

3.4 節で導入したサンプル回路の場合、節点を①～④、枝電流を図のように、 $i_1 \sim i_4$ 、 $V_{b1} \sim V_{b3}$ および V_1 とすると、
接続行列 D' は

$$D' = \begin{matrix} \text{①} \\ \text{②} \\ \text{③} \\ \text{④} \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 \end{bmatrix} \quad (3-66)$$

であるが、節点解析法の場合と同じく④を GND 電位として削除し、以下を得る。

$$D = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad (3-67)$$

キャパシタがある枝をコンダクタンス枝とし、節点解析と同様に台形積分法により離散化された BCE を求めると、

$$\begin{aligned} i_{n+1} &= \frac{2C}{h} (v_{n+1} - v_n) - i_n \\ &= GeqCv_{n+1} + IeqC \end{aligned} \quad (3-68)$$

インダクタの枝をレジスタンス枝とし、SPICE タイプの修正節点解析と同様に台形積分法により離散化した BCE を求めると、

$$\begin{aligned} v_{n+1} &= \frac{2L}{h}(i_{n+1} - i_n) - v_n \\ &= reqL \cdot i_{n+1} + VeqL \end{aligned} \quad (3-69)$$

抵抗 R の枝は、そのままレジスタンス枝とする。

以上よりタブロー方程式を求めると、

$$\begin{bmatrix} 0 & \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} & 0 \\ \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} & 0 & -1 \\ 0 & \begin{bmatrix} -r & 0 & 0 & 0 \\ 0 & -reqL & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -GeqC & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ V_{B1} \\ V_{B2} \\ V_{B3} \\ V_{B4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ VeqL \\ leqC \\ V_s \end{bmatrix} \quad (3-70)$$

であり、11×11 の行列が得られる。次元は確かに大きいですが、スパース性が高いため、これを活用したアルゴリズムを用いることができれば、計算負荷はさほど高くはならない。スパースタブロー法を MATLAB の m-script で実装をしたものを付録リスト 3 に示す。

3.8. カットセット解析 (Cut Set Analysis Method)

カットセット解析は、これまでの節点電圧、枝電圧、枝電流などを未知数とする解析手法ではなく、回路のエネルギーの収支を定める状態変数を用いる回路解析である。

3.8.1. カットセットとループの定義

回路を n 個の節点(Node)と m 個の枝(Branch)による有効グラフであらわしたとき、n-1 個の枝により n 個の節点を結びつけることができる。これを木 (Tree) と呼ぶ。この時 m-n+1 個の木に含まれない枝が残るが、これらの枝の集合を補木 (CoTree) と呼ぶ[39]。

有効グラフの枝にハサミを入れ、有効グラフを 2 個に分割したとき、ハサミをいれた枝の集合がカットセットである。以下に参考文献[40]に従ってカットセットの概要を説明する。

基本カットセット (Fundamental Cut Set) :

図 3.13 は木を太線、補木を点線で表した有向グラフであるが、この有効グラフにおいて、木に含まれる枝を一つだけ含み、残りの補木からなるカットセットを基本カットセットと呼ぶ。図 3.13 の場合のカットセット①は {枝 1、枝 5} であり、この中で枝 1 が木に含まれる枝である。

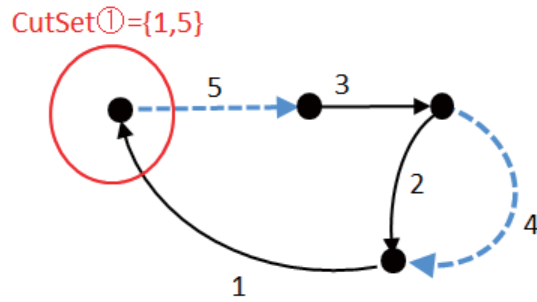


図 3.13 カットセットの例

基本カットセットは、以下のルールにより生成される行列 Q で表すことができる。

$$Q_{ij} = \begin{cases} 1: \text{枝 } j \text{ が基本カットセット } i \text{ に属し、方向が同じ} \\ -1: \text{枝 } j \text{ が基本カットセット } i \text{ に属し、方向が逆} \\ 0: \text{枝 } j \text{ が基本カットセット } i \text{ に属さない} \end{cases}$$

図 3.13 の有効グラフは 3 本の枝から構成される木を持つため、カットセットは 3 種類存在し、図 3.14 のように枝を選べば、カットセット行列 Q は以下である。

なお、式(3-71)では列番号を 1~6、行番号を①②③と注記しているが、それぞれ枝番号と、カットセット番号に対応する。

$$Q = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \end{matrix} \quad (3-71)$$

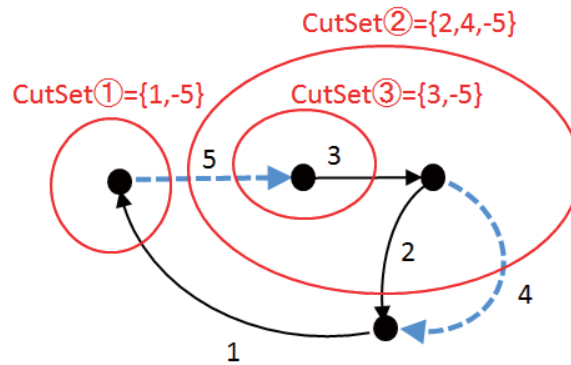


図 3.14 基本カットセット

基本ループ(Fundamental Loop) :

補木のなかから枝を 1 本選ぶと、選ばれた補木と木に含まれる枝からなるループができる。これを基本ループと呼ぶ

基本ループ行列 B は以下のルールにより生成される

$$b_{ij} = \begin{cases} 1 : \text{枝 } j \text{ が基本ループ } i \text{ に属し、方向が同じ} \\ -1 : \text{枝 } j \text{ が基本ループ } i \text{ に属し、方向が逆} \\ 0 : \text{枝 } j \text{ が基本ループ } i \text{ に属さない} \end{cases}$$

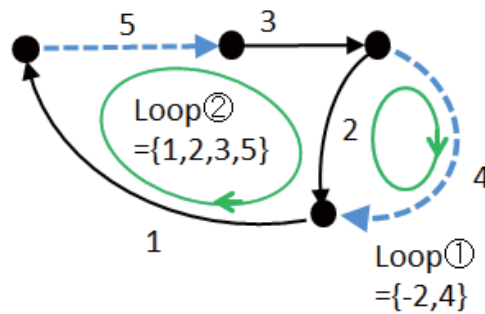


図 3.15 基本ループ

図 3.15 の基本ループは、補木が 2 本であるから、2 種類存在し、図 3.15 における基本ループ行列 B は、

$$B = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} \textcircled{1} \\ \textcircled{2} \end{matrix} & \begin{bmatrix} 0 & -1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \end{matrix} \quad (3-71)$$

である。

基本カットセットを選ぶ理由は、基本ループとの間に以下性質が成立する事にある。

$$Q = [I \ Q_l] \quad B = [B_l \ I] \quad (3-72)$$

$$Q_l = -B_l^T \quad (3-73)$$

KCL および KVL への対応：

KCL は節点に対する電流の総和で定義されるが、これはある領域に流入する電流の総和へ拡張することができる。すなわちカットセットに対しても KCL は同様に適用される。

また、基本ループに対しては、KVL は当然適用可能である。

従って、すべての枝の電圧、電流ベクトルを I 、 V として以下に定義すると、

$$I = [i_1, i_2, \dots, i_m]^T \quad V = [v_1, v_2, \dots, v_m]^T \quad (3-74)$$

$$\text{KCL : } Q I = 0 \quad (3-75)$$

$$\text{KVL : } B V = 0 \quad (3-76)$$

図 3.13 の場合では、

$$QI = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix} = \begin{bmatrix} i_1 - i_5 \\ i_2 + i_4 - i_5 \\ i_3 - i_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3-77)$$

$$BV = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \begin{bmatrix} -v_2 + v_4 \\ v_1 + v_2 + v_3 + v_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3-78)$$

である。

3.8.2. 標準木

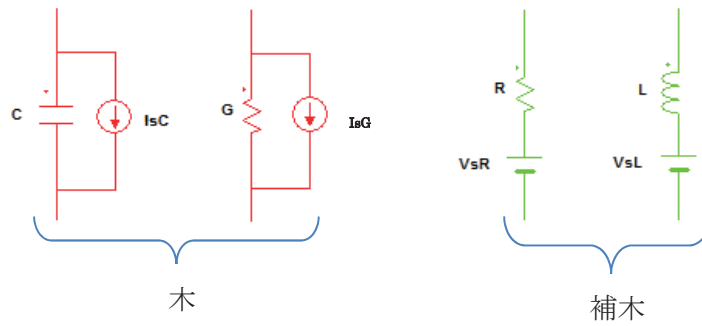
カットセット解析は標準木(Normal Tree)と呼ばれる、特定の回路要素のみで構成される基本木を用いることで機械的な方程式の導出が可能となる。

以下では、参考文献[41]の標準木の構成手法を紹介する。

この手法においては、

- ① 木を構成する枝は図 3.16 のようにキャパシタとコンダクタと独立電流源の組み合わせのみで構成されとする。
- ② 補木を構成する枝は図 3.16 のようにレジスタとインダクタと独立電圧源の組み合わせのみで構成されとする。

という制約をおくが、カットセット解析には様々な手法が提案されており、一般的には、より制約が緩い手法[42][43]が知られている。しかし、ここでは方程式の導出が簡略化され、説明が分かり易い本手法を説明する。



3.16 標準木の構成

以上の枝に対し、それぞれ以下の BCE が得られる。

$$\text{キャパシタ木: } i_C = C \frac{d}{dt} v_C + I_{sC} \quad (3-79a)$$

$$\text{コンダクタ木: } i_G = g v_G + I_{sG} \quad (3-79b)$$

$$\text{インダクタ補木: } v_L = L \frac{d}{dt} i_L + V_{sL} \quad (3-79c)$$

$$\text{レジスタ補木: } v_R = r i_R + V_{sR} \quad (3-79d)$$

以上の前提のもと、木に含まれる枝の枝電流を $[i_C \ i_G]^T$ 枝電圧を $[v_C \ v_G]^T$

補木に含まれる枝の枝電流を $[i_L \ i_R]^T$ 電圧を $[v_L \ v_R]^T$ とすると、

式(3-75)の KCL は

$$\begin{bmatrix} I & Q_l \end{bmatrix} \begin{bmatrix} i_C \\ i_G \\ i_L \\ i_R \end{bmatrix} = 0 \quad (3-80)$$

式(3-76)の KVL は

$$\begin{bmatrix} B_t & I \end{bmatrix} \begin{bmatrix} v_C \\ v_G \\ v_L \\ v_R \end{bmatrix} = 0 \quad (3-81)$$

と表すことができる。

3.8.3. 状態変数でない余剰変数の削除

式(3-80)、式(3-81)を整理すると

$$\begin{bmatrix} i_C \\ i_G \end{bmatrix} = -Q_l \begin{bmatrix} i_L \\ i_R \end{bmatrix} = -\begin{bmatrix} Q_{CL} & Q_{CR} \\ Q_{GL} & Q_{GR} \end{bmatrix} \begin{bmatrix} i_L \\ i_R \end{bmatrix} \quad (3-82)$$

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = -B_t \begin{bmatrix} v_C \\ v_G \end{bmatrix} = -\begin{bmatrix} B_{LC} & B_{LG} \\ B_{RC} & B_{RG} \end{bmatrix} \begin{bmatrix} v_C \\ v_G \end{bmatrix} \quad (3-83)$$

なお、式(3-73)より

$$B_{LC} = -Q_{CL}^T B_{LG} = -Q_{GL}^T B_{RC} = -Q_{CR}^T B_{RG} = -Q_{GR}^T \quad (3-84)$$

が成立する。

これに BCE である式(3-79a)～(3-79d)および、式(3-84)を用いて、 $i_C i_R i_G v_L v_R v_G$ を消去すると

$$\frac{d}{dt} \begin{bmatrix} v_C \\ i_L \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}^{-1} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}^{-1} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{bmatrix} \begin{bmatrix} I_{sC} \\ I_{sG} \\ V_{sL} \\ V_{sR} \end{bmatrix} \quad (3-85)$$

$$\begin{aligned} a_{11} &= Q_{CR} R_{eq}^{-1} B_{RC} & a_{12} &= -Q_{CL} - Q_{CR} R_{eq}^{-1} B_{RG} G^{-1} Q_{GL} \\ a_{22} &= B_{LG} G_{eq}^{-1} Q_{GL} & a_{21} &= -B_{LC} - B_{LG} G_{eq}^{-1} Q_{GR} R^{-1} B_{RC} \end{aligned} \quad (3-86)$$

$$\begin{aligned} b_{11} &= -1 & b_{12} &= -Q_{CR} R_{eq}^{-1} B_{RG} G^{-1} & b_{13} &= 0 & b_{14} &= Q_{CR} R_{eq}^{-1} \\ b_{21} &= 0 & b_{22} &= B_{LG} G_{eq}^{-1} & b_{23} &= -1 & b_{24} &= -B_{LG} G_{eq}^{-1} R^{-1} \end{aligned} \quad (3-87)$$

$$R_{eq} = R - B_{RG} G^{-1} Q_{GR} \quad G_{eq} = G - Q_{GR} R^{-1} B_{RG}$$

を得ることができる。なお、ここここで使用している G 、 R は、(3-79b)と(3-79d)で現れる g と r を対角要素に持つ行列である。

3.8.4. サンプル回路の定式化

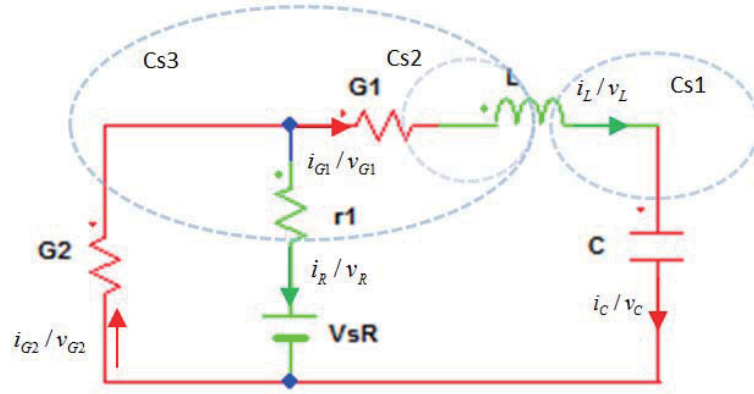


図 3.17 サンプル回路に対する標準木とカットセットの定義

3.4 節のサンプル回路に対し、方程式の形が分かり易いように、仮想的なコンダクタ $G2$ とレジスタ $R2$ を追加してあるが、 $G2=R2=0$ とすれば 3.4 節のサンプル回路と一致する。

上記の回路において図 3.17 における $Cs1, Cs2, Cs3$ をカットセットとするとカットセット行列は

$$[I \quad Q_l] = \begin{matrix} & C & G1 & G2 & L & R \\ \begin{matrix} Cs1 \\ Cs2 \\ Cs3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & -1 \end{bmatrix} \end{matrix} \quad (3-88)$$

である。

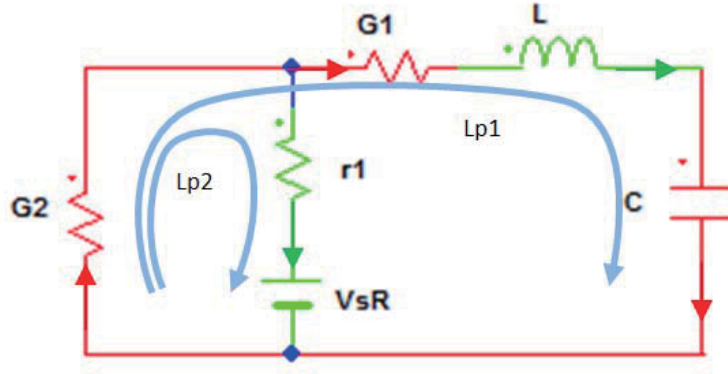


図 3.18 サンプル回路に対する標準ループの定義

図 3.18 のようにループを設定すると、ループ行列は

$$[B_l \quad I] = \begin{matrix} & C & G1 & G2 & L & R \\ \begin{matrix} Lp1 \\ Lp2 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix} \quad (3-89)$$

となる。

以上より i_c, v_L に関する式を導くと、

$$\begin{bmatrix} i_C \\ i_{G1} \\ i_{G2} \end{bmatrix} = - \begin{matrix} Q_{CL} & Q_{CR} \\ \begin{bmatrix} -1 & 0 \\ -1 & 0 \\ -1 & -1 \end{bmatrix} \\ Q_{GL} & Q_{GR} \end{matrix} \begin{bmatrix} i \\ i_{Vs} \end{bmatrix} \quad (3-90)$$

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{matrix} B_{LC} & B_{LG} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \\ B_{RC} & B_{RG} \end{matrix} \begin{bmatrix} v_C \\ v_{G1} \\ v_{G2} \end{bmatrix} \quad (3-91)$$

であり、

$$C = [C] \quad L = [L] \quad R = [r1] \quad I_{SR} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad V_{SR} = \begin{bmatrix} 0 \\ V1 \end{bmatrix} \quad G = \begin{bmatrix} G1 & 0 \\ 0 & G2 \end{bmatrix} \quad (3-92)$$

式(3-92)と式(3-86)を適用して

$$\begin{aligned}
 a_{11} &= 0 \\
 a_{12} &= 1 \\
 a_{21} &= -1 \\
 a_{22} &= B_{LG} G_{eq}^{-1} Q_{GL} \\
 &= [1 \quad 1] \begin{bmatrix} \frac{1}{G1} & 0 \\ 0 & \frac{r1}{r1G2+1} \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} \\
 &= \frac{-1}{G1}
 \end{aligned} \tag{3-93}$$

ここで、 $G2=r1=0$ 、 $G1=1/R$ を適用すると

$$a_{22} = -R$$

より、式(3-84)より

$$A = \begin{bmatrix} \frac{1}{C} & 0 \\ 0 & \frac{1}{L} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & -R \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \tag{3-94}$$

また、

$$I_{SR} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

より B 行列は 2×2 であるので、

$$\begin{aligned}
 b_{14} &= 0 \\
 b_{24} &= -[1 \quad 1] G_{eq}^{-1} \begin{bmatrix} 0 \\ -1 \end{bmatrix} R^{-1} \\
 &= -[1 \quad 1] \begin{bmatrix} \frac{1}{G1} & 0 \\ 0 & \frac{r_{sV}}{(r_{sV}G2+1)} \end{bmatrix} \begin{bmatrix} 0 \\ -\frac{1}{r_{sV}} \end{bmatrix} \\
 &= -[1 \quad 1] \begin{bmatrix} 0 \\ -1 \\ (r_{sV}G2+1) \end{bmatrix}
 \end{aligned} \tag{3-95}$$

であるが、 $G2=0$ より

$$b_{24}=1$$

よって

$$B = \begin{bmatrix} \frac{1}{C} & 0 \\ 0 & \frac{1}{L} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{-1}{L} & \frac{1}{L} \end{bmatrix}$$

(3-96)

であるが、

$$V_{sr} = \begin{bmatrix} 0 \\ V1 \end{bmatrix}$$

であるから、

得られる状態方程式は、

$$\frac{d}{dt} \begin{bmatrix} v_c \\ i_L \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} v_c \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} V1 \end{bmatrix} \quad (3-97)$$

となる。

参考までに、本回路は

$$L \frac{di}{dt} = v_L$$
$$C \frac{dv_c}{dt} = i_L$$

として、KVL を適用することで、容易に状態方程式を導くことができる。

$$\frac{d}{dt} \begin{bmatrix} v_c \\ i_L \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{1}{LG1} \end{bmatrix} \begin{bmatrix} v_c \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} V1 \end{bmatrix} \quad (3-98)$$

式(3-97)は式(3-98)と一致している。

4章 スイッチング回路を計算する最適な計算手法

パワーエレクトロニクスとは、半導体素子を主に導通と非導通の 2 状態のスイッチとして使用するものであるが[44]、このスイッチ動作を行う回路に対し回路方程式を構築し、計算し、動作を模擬するためには、いくつかの特有な課題が存在する。この章では、そのような特有の問題についての考察を行う。

4.1. スイッチ素子の扱い

スイッチ動作を行う回路に対し、先ず考慮しなければならないことは、スイッチ素子の扱い方である。

スイッチ素子を、その定義に従い ON=導通、OFF=非導通の 2 状態とすると、スイッチ動作に伴い、回路構成（トポロジー）が変更されことになるが、その場合は回路を定義する節点や枝が生成／消滅するため、未知変数もその都度生成／消滅するなど、扱いにくい点がある。

本論文では ON 状態では低い抵抗値、OFF 状態では高い抵抗値を持つ、可変の抵抗要素として扱うことを基本とする。

しかし、この場合は、ON 状態では実際のデバイスの特性にあわせて数 $m\Omega$ とすることができるが、OFF 状態では非導通であり、抵抗値で表すには難がある。これを数 $100M\Omega$ というような限りなく高い値にすれば、実デバイスに近くなるが、数 $m\Omega$ と数 $100M\Omega$ というような設定にすると、パラメータの幅が 10^{10} 以上ということになり、方程式が Stiff になりやすい。従って、Stiff 性を回避するため、回路構成を変更するという手法の検討も、今後の課題として必要である。

4.2. スイッチング時間の分解能の影響

スイッチング回路では、ON 動作の時間と OFF 動作の時間により、回路の動作が決定される。このスイッチ時間は通常はコントローラの制御出力値であり、シミュレータとしては計測された入力値である。従って、他のパラメータ値のように任意の精度が得られないため、ON/OFF 時間の計測に必要な精度を検討する。

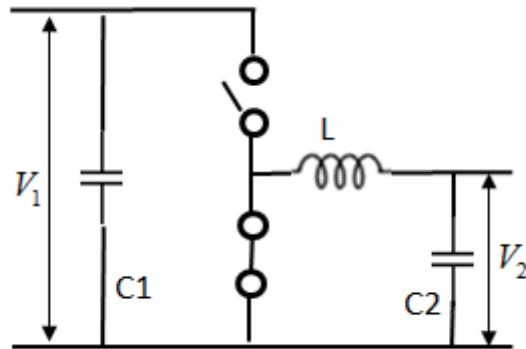


図 4.1 スイッチ動作の時間分解能の検討を行うサンプル回路

図 4.1 に示すハーフブリッジ回路において、ハイサイドとローサイドのスイッチを交互に ON とする構成であれば、ハイサイド側のスイッチの ON 時間を T_{ON} 、OFF 時間を T_{OFF} とし、内部損失等を見放した場合、降圧コンバータとして使用した場合に定常状態に達した時の降圧比は

$$\frac{T_{ON}}{T_{ON} + T_{OFF}} = D \quad (4-1)$$

とすると、

$$\frac{V_2}{V_1} = D \quad (4-2)$$

で表され、

昇圧コンバータとして使用した場合の昇圧比は、入出力が逆転する事より、

$$\frac{V_1}{V_2} = \frac{1}{D} \quad (4-3)$$

で表される。

D は Duty 比と呼ばれ、定義から $0 \leq D \leq 1$ である。

ここで、D に対し、 α % の乗法誤差が生じた場合、降圧コンバータの場合

$$\alpha D = \alpha \frac{V_2}{V_1} \quad (4-4)$$

であり、

昇圧コンバータの場合は、

$$\frac{1}{\alpha D} = \frac{V_1}{\alpha V_2} \quad (4-5)$$

であるため、 α に比例した誤差が昇降圧比、すなわち出力値に発生する[44]。

従って、 D の計測分解能は、必要とされる出力値の精度と式(4-4)、式(4-5)より決定されなければならない。

4.3. スイッチ動作を伴う回路の代表的な計算手法

スイッチ動作の存在は計算手法に対しても大きな影響を与えており、現在までに様々な手法が提案されている。以下にこれまでに提案されてきた代表的手法を検討する。

4.3.1. 固定ステップ法

最も容易に考えうるのは、適切な一定の計算ステップ幅による計算であり、回路以外の分野でのリアルタイム・シミュレータの方式としては最も一般的なものである。

計算ステップ幅を h とすると、図 4.2 のように所定の h の間隔で、状態変数を計算する。ハーフブリッジ回路を例にとり、ハイサイド、ローサイドが交互に ON する状況を考え、各々に対応する路方程式が

ローサイドスイッチ=ON

$$\frac{d}{dt}X(t) = A_1X(t) + B_1U(t) \quad (4-6)$$

ハイサイドスイッチ=ON

$$\frac{d}{dt}X(t) = A_2X(t) + B_2U(t) \quad (4-7)$$

であるとし、計算手法を前進 Euler 法であるとする、

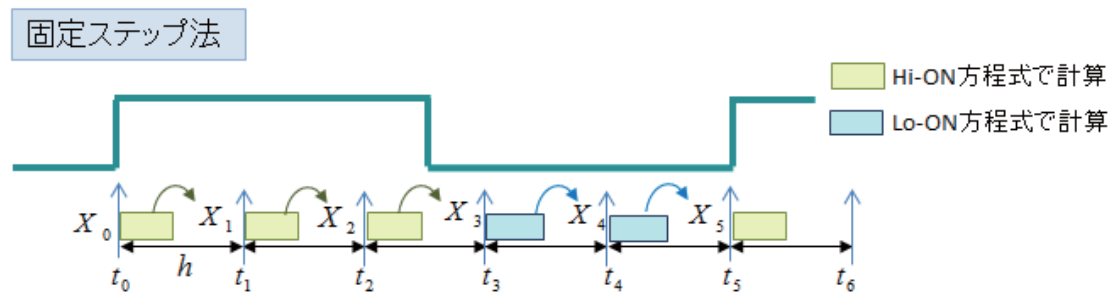


図 4.2 固定ステップ法の計算

すなわち、 t_0 を過ぎ、 X_0 と U_0 が確定したならば、 t_1 の時点までに、

$$X_1 = X_0 + h(A_1X_0 + B_1U_0) \quad (4-8)$$

式を計算し、出力値を t_1 のタイミングで物理的に出力する。

ここで、 t_2 と t_3 の間でスイッチング動作が発生するならば、スイッチ動作を検出できるのは時刻 t_3 であるから、 t_3 において、A、B 行列を A_1 、 B_1 から A_2 、 B_2 に変更し、

$$X_4 = X_3 + h(A_2 X_3 + B_2 U_3) \quad (4-9)$$

として計算する。

この時使用する X_3 は、すでに A_1 行列で計算されているため、これは時刻 t_3 においてスイッチ動作が発生したとして計算することと等価である。

すなわち、所定の計算ステップ h を用いる固定ステップ法では、スイッチ動作の発生時刻の分解能は計算ステップ幅と一致する。

4.2 節で示したように、Duty 比の時間分解能は重要であるが、固定ステップ法を導入した場合は、この時間分解能を高めるために、極めて細かい計算ステップ幅が必要となる。

4.3.2. 状態平均化法

スイッチング回路の理論解析では定番である状態平均化法であるが、リアルタイム・シミュレータの分野では、この手法は平均値 (Mean Value) モデルと呼ばれる。

この手法は式(4-10)、(4-11)のように状態方程式の A、B 行列をスイッチの ON 時間(T_{ON})、OFF 時間(T_{OFF})で平均を取り、単一の A、B 行列に変換し、これを用いて計算を行うものである。

$$A = \frac{T_{ON}}{T_{ON} + T_{OFF}} A_1 + \frac{T_{OFF}}{T_{ON} + T_{OFF}} A_2 \quad (4-10)$$

$$B = \frac{T_{ON}}{T_{ON} + T_{OFF}} B_1 + \frac{T_{OFF}}{T_{ON} + T_{OFF}} B_2 \quad (4-11)$$

固定ステップ法の例と同じく前進 Euler 法であれば、

$$X_{n+1} = X_n + (T_{ON} + T_{OFF})(AX_n + Bu_n) \quad (4-12)$$

となり、この計算をスイッチの ON/OFF 周期の間に 1 回だけ行う。

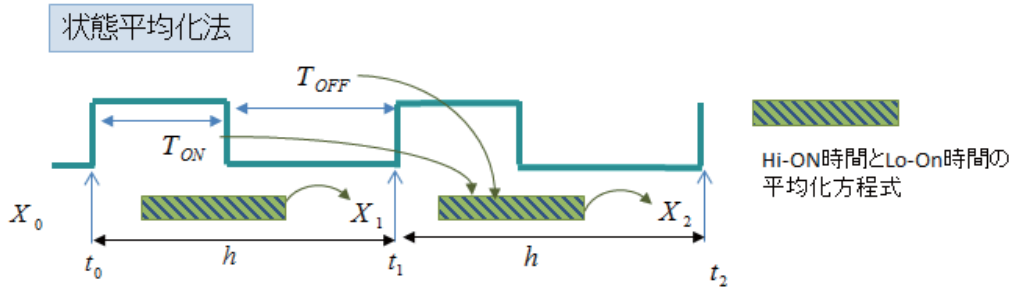


図 4.3 状態平均化法による計算

状態平均化法を用いた場合、通常はスイッチ素子を PWM (Pulse Width Modulation) 駆動する場合が対象であるが、

- ① 出力は、PWM 信号の一周期の平均であり、リップル電圧／リップル電流等は再現できない。
- ② 状態の平均を算出するために PWM 信号の一周期を計測し、Duty 比が決定された後に、状態方程式の平均を取るため、入力に対する出力が PWM 信号の 1 周期遅れる。
- ③ PWM 以外の PFM (Pulse Frequency Modulation) 等、スイッチの ON/OFF 周期が代わる信号に対しては、計算ステップ幅が変化するため、実装および安定性解析が困難になる。従って、PWM 以外の駆動方式への適用が困難である。

という問題があるが、計算ステップ幅＝スイッチ素子の ON/OFF 周期 (PWM であればキャリア周期) であり、他の手法に比べ極めて長い計算時間が取れることが特徴である。

4.3.3. 部分的可変ステップ法 (スイッチ駆動信号同期)

部分的可変ステップ法との名称は、通常の計算精度に応じて計算ステップ幅を変更する可変ステップ法と区別するため、著者が名づけたものであるが[45]、この手法は時刻 $t_2 + h_{hf}$ でスイッチ動作が起きたならば、 t_3 の時刻において、

$$X_{hf} = X_3 + h_{hf}(A_1 X_3 + B_1 U_3) \quad (4-13)$$

$$X_4 = X_{hf} + (h - h_{hf})(A_2 X_{hf} + B_2 U_3) \quad (4-14)$$

として X_4 を計算する手法である。

この手法においても、スイッチ動作を検出できるのは、 t_3 時点であるため、 $t_2 \sim t_3$ の間にスイッチ動作が発生しても、それを反映できるのは t_4 時点での X_4 となる。

従って、全体の計算を一ステップ遅らせる必要が生じるが、Duty 比の計算精度は、 h_{hf} の計測精度次第であり、基本計算ステップの h とは無関係となることが特徴である。

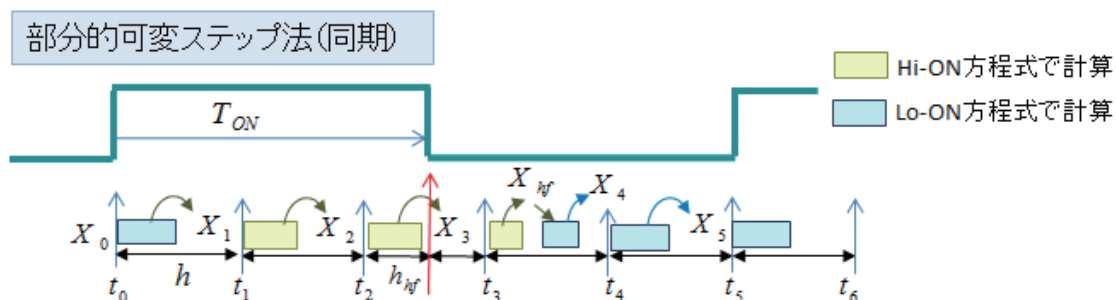


図 4.4 部分的可変ステップ法（同期）による計算

また、以上の手法は、スイッチング周波数に変化する PFM 駆動等にも適用可能であるが、スイッチ動作開始時刻 T_{ON} を計測において、使用とするスイッチング周波数範囲でオーバーフローしない、あるいは必要な分解能が使用するすべての駆動周波数で確保できる工夫が必要となる。

さらに、スイッチ駆動を停止した（Duty 比=0%/100%）場合に対応できないという問題がある。

4.3.4. 部分的可変ステップ法（スイッチ駆動信号非同期）

スイッチ駆動信号(PWM/PFM 信号)とは無関係の、所定の一定間隔で計算を行う手法で、上記同期手法では対応できないスイッチ駆動停止（Duty 比=0%/100%）に対応可能であり、PFM にも適用可である。

一方、スイッチ駆動信号と計算タイミングが一致しないことより、リップル電圧／電流の位相ずれ（計算遅れ）が一定にならないという問題が発生する。従って、この位相のずれが無視できるだけの小さな計算ステップ幅 h が実現できる事が望ましい。

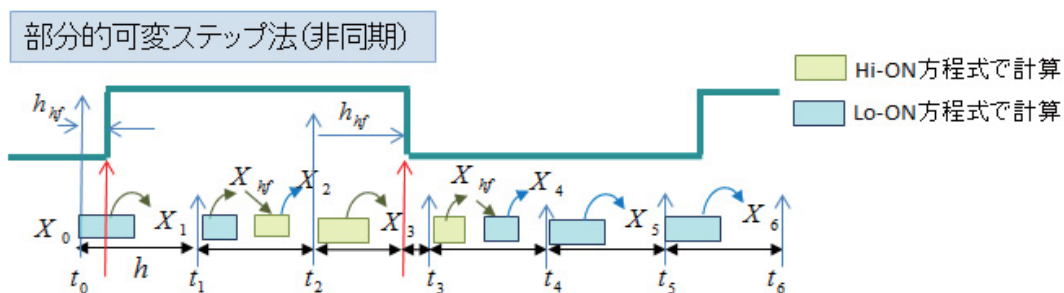


図 4.5 部分的可変ステップ法（非同期）による計算

スイッチ駆動信号とは非同期の所定の間隔（＝計算ステップ）で計算タスクを起動し、タスク起動と同時にタイマー起動を行う。各タスク起動時にスイッチ動作を検出したとき、例えば、時刻 t_1 においてスイッチ動作が発生し、スイッチ動作発生までの時間(= h_{hf})が検出されたなら、時刻 t_2 において、

$$X_{hf} = X_1 + h_{hf}(A_2X_1 + B_2U_1) \quad (4-15)$$

$$X_2 = X_{hf} + (h - h_{hf})(A_1X_{hf} + B_1U_1) \quad (4-16)$$

の計算を行う。

時刻 t_3 においてもスイッチ動作が発生しているため同様の計算を行うが、Hi→Lo と Lo→Hi の場合は、使用する A、B 行列の順番が逆となる。

本手法は、Duty 比=0 及び 100%においても使用することができる。すなわち、タスクが同期しなければならないスイッチ駆動信号のエッジを必要としない。Duty 比=0%ないし 100%の場合は、A,B 行列を変更せずに計算を継続するだけで良く、タイマー値の最大値はタスク起動間隔（＝ h ）であるため、キャリア周波数を変動させても、タイマーがオーバーフローする可能性は無い。

同期、非同期、いずれの部分的可変ステップ法もスイッチ動作の時間分解能が計算ステップ幅 h とは無関係であるため、精度の高い計算が可能となる。計算ステップ幅 h が大きくなる場合は、数値積分を行う際に次数の高い解法を選択する事で、計算精度を維持する事が可能である。

しかしその一方、固定ステップ法に比べて、

- ① 固定ステップ法の出力遅れが最大 1 ステップである事に対し、
最大 2 計算ステップ遅れとなる。
- ② スイッチ動作が発生したときの、単位時間あたりの計算量が 2 倍となる。
が欠点として挙げられる。

4.3.5. 状態変数の補間／外挿手法

部分的可変ステップ法は、常微分方程式の計算において、計算ステップ幅を任意に変更する事が容易であることが必要となる。しかし、回路方程式の定式化手法の主流である、修正節点解析法などにおいては、予め決まった計算ステップを用いて素子単位で離散化した差分方程式化を元に、回路方程式を構築しているため、スイッチ動作に伴い計算ステップ幅を変更した場合、回路方程式を再構築する必要が生じる。これはリアルタイム計算に対して大きな障害となる。

$$Y X = C \quad (4-17)$$

さらに、修正節点解析法は式(4-17)のように、未知数 X に対して、アドミタンス行列 Y が左側にあり、 X を得るには逆行列を解く必要がある。通常は、予め逆行列を計算しメモリに蓄えておき処理を高速化するが、回路方程式が再構築される場合、逆行列の計算も必要となり、逆行列の計算は、直接法では一般に $O(n^3)$ とされており、これをリアルタイムに行う事はかなり困難である。

このような問題に対し、図 4.6 のように状態変数の線形補間を 2 回行う方法、および、線形補間と外挿を組み合わせる方法が提案されている[47]-[50]。

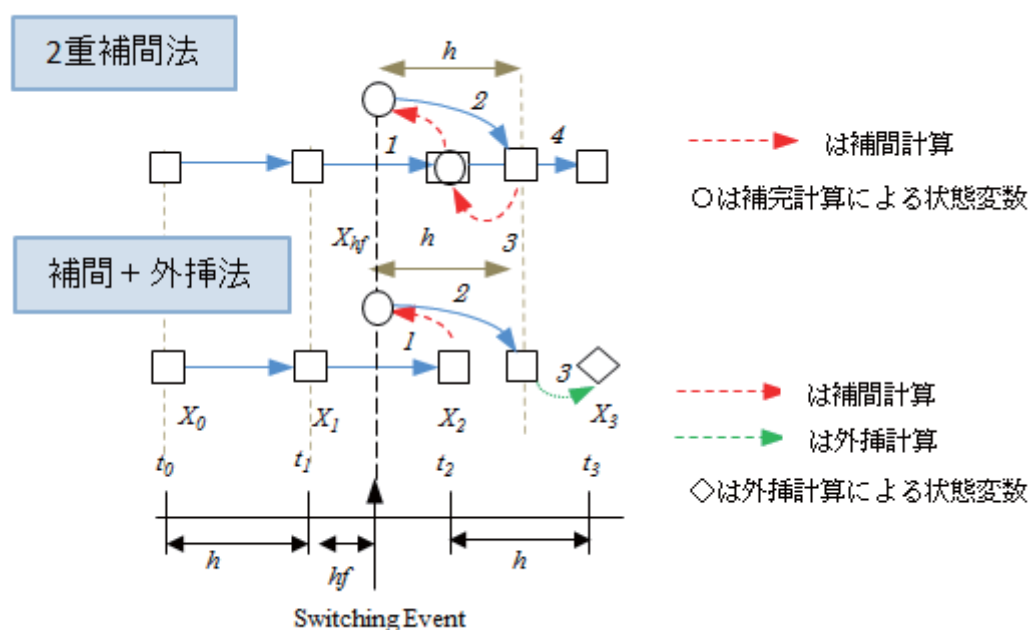


図 4.6 2重補間法と補間+外装法の計算

先ず 2 重補間法であるが、時刻 t_3 においてスイッチ動作を検出した場合、図 4.6 の”1”の計算によりスイッチ動作発生時点の状態 X_{hf} を線形補間により算出する。次に”2”の計算では、計算ステップ h を用いて通常の計算を行うが、このままでは、計算周期がずれてしまうので、”3”で再び線形補間を行い、 X_2 を算出し、通常計算に戻るというものである。

これに対し、補間+外挿法では、最後に X_2 を算出する負荷が大きいとして、外挿処理で直接 X_3 を求めるというもので、精度的には劣ると思われるが、2 重補間より高速である。

以上が概略であるが、いずれにせよ、線形補間であるから、計算誤差の点で疑問が残る。

すなわち、参照した論文においては具体的な方法は記載されていないが、

$$\text{スイッチ動作が起きなかった時の計算: } X_{2-1} = X_1 + hA_1X_1 \quad (4-18)$$

$$\text{スイッチ動作が起きた時の計算： } X_{2_2} = X_1 + hA_2 X_1 \quad (4-19)$$

これをスイッチ動作発生時点での h_{hf} を用いて平均化すると、

$$X_2 = \frac{1}{h} (h_{hf} X_{2_1} + (h - h_{hf}) X_{2_2}) \quad (4-20)$$

であるが、

この計算を上記”1”のステップで行うのであれば、

$$\begin{aligned} X_2 &= \frac{1}{h} (h_{hf} (X_1 + hA_1 X_1) + (h - h_{hf}) (X_1 + hA_2 X_1)) \\ &= X_1 + h_{hf} A_1 X_1 + (h - h_{hf}) (A_2 X_1) \end{aligned} \quad (4-19)$$

であり、物理的実態を正しく計算している式(4-16)に比べ、式(4-19)は X_{hf} で計算すべきところを X_1 で行っているため、計算精度に難があると言わざるを得ない。

4.4. デッドバンド時間の扱い

通常ブリッジ回路の駆動においては、スイッチ素子の **OFF** 動作の遅れによる短絡を防ぐため、ハイサイド、ローサイド、両スイッチが **OFF** 状態となる、デッドバンド時間を設けることが普通である。

従って、スイッチング回路の計算においても両スイッチ **OFF** のデッドバンドを考慮すべきであるが、これをそのまま方程式にすると、極めて **Stiff** な A 行列が生成される。

しかし、通常のブリッジ回路においては、図 4.7 のようにフライホイールダイオードを設けており、**FET** を用いた場合は、寄生ダイオードとして **FET** 内にダイオードが内蔵されている。これらダイオードの順方向電圧降下(V_f)が 0.7~1V 程度であるから、扱う電圧が 400V 程度と高い場合は、順方向電圧降下を無視することができる。

すなわち、図 4.7 のように電流の流れる向きを判断し、ハイサイド、もしくはローサイドを **ON** するという扱いにすることで、双方のスイッチを **OFF** とした場合に表れる **Stiff** な方程式を回避する事が可能である。

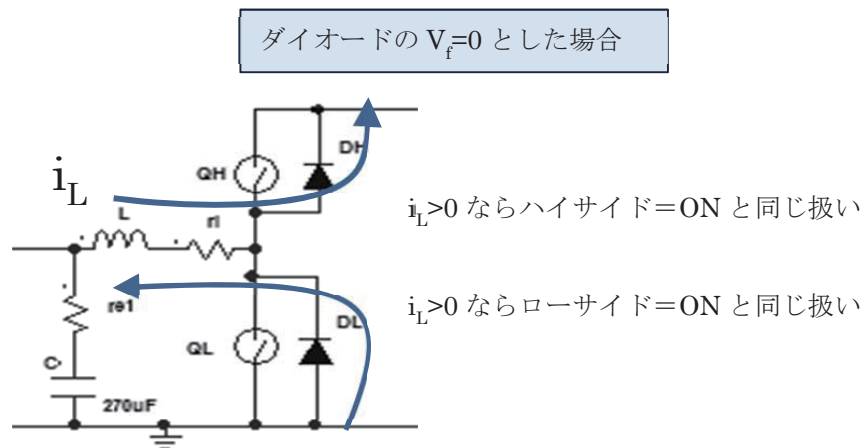


図 4.7 デッドバンド時間（両スイッチ=OFF）の扱い

一方、電圧が低く、順方向電圧降下が無視できない場合は、Stiff な方程式を解くための対策が必要となる。この点は今後の研究課題である。

4.5. スイッチ非線形素子の扱い

SPICE の特徴として、トランジスタやダイオードなどの非線形特性を持つ素子を扱うことができ、SPICE に対応した各種のデバイスモデルが提供されている事がある。

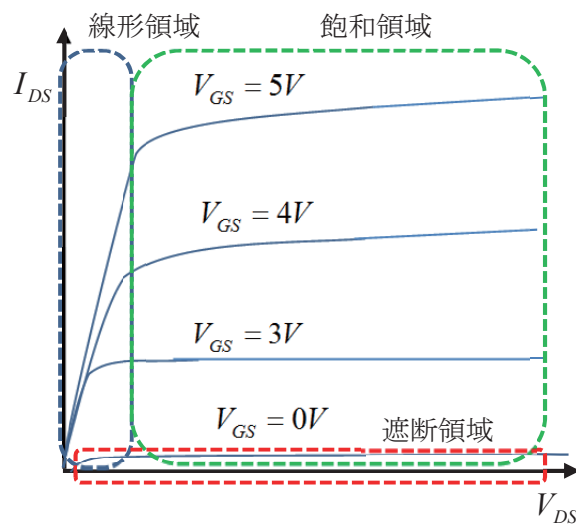


図 4.8 MOS-FET の出力特性

本来スイッチ素子は、図 4.8 に示す MOS-FET の例のように非線形特性を持つが、本研究では、4.1 節にあるように、スイッチ素子は線形／遮断領域で使うことを前提に ON/OFF で 2 値の抵抗値を持つ抵抗素子としている。これは、リアルタイム・シミュレータは制御

装置の開発に用いるもので、回路特性の解析に用いるものではないため、素子の非線形性を再現する事は過剰性能である、との考えによる。

上記、2 値の抵抗ではなく、スイッチ素子が線形領域から遮断領域に移行する間の過渡的な現象を考慮するならば、ハードスイッチングであれば、そこに損失が発生するから、通常は損失を熱設計的に、あるいはエネルギー収支的に許容できるところまで、スイッチング周波数を下げていくのが常識である。これは制御機器側から見れば、制御に影響がなくなるまでスイッチング周波数を下げる、と言い換えてもよい。

一方、ソフトスイッチの回路に対しても、スイッチ素子の浮遊容量と、これを打ち消す共振回路を、線形の回路方程式として扱えば十分である。

しかしこのような回路は、固有値が高くなるためリアルタイム計算が困難である。ソフトスイッチ対応のリアルタイム・シミュレータの実現には、さらなる技術開発が必要である。

4.6. 結論 —スイッチング回路の計算に最適な計算アルゴリズム—

以上 4 章で検討した、スイッチング回路固有の問題を解消するアルゴリズム、3 章において検討した、回路方程式の構築手法、第 2 章において検討した常微分方程式の組み合わせにおいて、どれが最適であるかを結論する。

まず、スイッチング回路の計算アルゴリズムとしては、部分的可変ステップ法が最適である。同期方式にするか非同期方式にするかは、対象とする制御手法、および実現手段に依存するが、Duty 比を 0%あるいは 100%するような制御を行う場合は、ゲート駆動信号に同期するエッジがないため、非同期にせざるを得ない。

固定ステップ法は、計算ステップ幅=Duty 比の時間分解能となり、4.2 節で検討したように、Duty 比の時間分解能は計算精度に直接関わるため、問題が多い。FPGA 等の高速なハードウェアを導入しても、同一ハードウェアで部分的可変ステップ法を適用すれば、より高精度になるため、解決とは言えない。

状態変数の補間／外挿の手法は、部分的可変ステップ法に比べ、計算精度が落ちる上に、スイッチ動作発生時に単位時間当たりの計算負荷が高まる点は同じなので、魅力がない。

状態平均化法は、簡易的なリアルタイム・シミュレータの構築には適するが、性能の向上要素に乏しい。

次に回路方程式の構築手法について検討する。部分的可変ステップ法を用いた場合、計算ステップ幅が、実稼働状態まで確定しない(計算ステップがある)、という問題が生じる。従って、所定の計算ステップで素子単体を離散化し、等価コンダクタや等価電流源として回路方程式を構築する、節点解析法、修正節点解析法、スパースタブロー法は、実時間での回路方程式の構築が必要となり、不利である。

また、これらによる回路の定式化では、行列が未知数の左側にあり、逆行列を解く必要がある。従来の回路用リアルタイム・シミュレータでは、この逆行列を実時間計算を行う前に予め計算しておき、実時間計算では、記憶された逆行列のみを用いるという手法が用いられているが、行列の要素に計算ステップ幅が組み込まれているため、部分的可変ステップ法では計算ステップ幅が確定する実時間計算状態で逆行列を解く必要がある。FPGAで実装した例はあるが[15]、本例では逆行列の解法に Gauss-Seidel 法を使用しており、2回の反復で計算を終了している。わずか2回で計算を収束させるためには行列が強く対角優位性である必要があるが、スイッチ素子のモデル化において所定の容量のキャパシタンスを並列に追加する事で対角優位を確保する、という手法の妥当性には疑問が残るところである。

従って、回路方程式としては、計算ステップ幅が回路方程式に含まれない、連続系の状態方程式が最も適しており、状態方程式で回路方程式を構築する手法としてよく知られた、カットセット解析を採用する。

最期に常微分方程式の数値解法的方式であるが、先ず線形多段解法の採用は問題がある。この手法は、過去のデータを使用するため、次数を大きくすると、計算の初めに使用する初期値もそれだけ必要になる。実用上は、計算の初めには、1段階法等を使い、計算が進んで、初期値の設定が可能になるところで多段解法に切り替える、あるいは低い次数（1次であれば1段階法と同じ）で始めて、その後に高い次数に切り替えるなどの手法が定番だが、スイッチング回路を対象と考えた場合、スイッチ動作直後のかなり重要な計算において上記のように1段階法とするか次数を減らすなどの対応となるので、線形多段解法として、“多段階”を活用しているとは言えない。

さらに、Adams系はBashforthもMultonも次数を上げると安定領域が減少する点が評価できない。

従って選択枝としては1段階法であるが、安定領域の観点からは1ないし2次の後退微分公式（BDF/Gear法）が最も有力であるが、A安定であれば十分であるから、精度に優れる台形積分法を採用すべきであろう。実際、SPICEを初めとした多くの回路シミュレータは台形積分法の採用が多く、後退微分公式がそれに続く。

しかし、台形積分法も後退微分公式も陰解法であり、計算が複雑になる点で問題である。陰解法を用いると、ニュートン法などの反復計算を行うか、逆行列の計算が必要となる。反復計算は計算終了時間が非確定なため、リアルタイム・シミュレータに適合する事は困難である。また、逆行列の計算は、直接法では $O(n^3)$ であり、Gauss-Seidel法の反復計算は、同じく計算終了時間が非確定である。従って、陽解法で1段階法であるRunge-Kutta法が第一の候補となる。

数値解法の安定性は、対象となる方程式の固有値と、計算ステップにより決まることは、

第 2 章で述べたが、これはつまり、対象となる方程式の固有値により、最適な計算手法が変わる事を意味する。

リアルタイム・シミュレータは、制御ソフトを開発するツールであり、回路を解析するものでない、という原点に立ち返るならば、リアルタイム・シミュレータが計算すべき回路のバリエーションはさほど多くないことに気付く。

第 5 章において具体的な値で述べるが、スイッチング周波数を 20kHz とした場合、経験的に言われる指針に基づいて設計された“標準的な回路定数”を用いた回路を表す方程式の固有値は極端な値とはならず、 $10\mu\text{sec}$ の計算ステップを使用した場合において、ありふれた陽的 Runge-Kutta 法の安定領域に十分に入る。一方、ガイドライン[21]等に表示されるリアルタイム・シミュレータの計算ステップは制御周期の $1/5 \sim 1/20$ であるから、20kHz であれば $2.5\mu\text{sec} \sim 10\mu\text{sec}$ であり、上記の安定性を確保できるとした $10\mu\text{sec}$ より小さい。 $10\mu\text{sec}$ で安定に計算できれば、それより小さい計算ステップでは、(特殊な例を除けば)同じく安定に計算できる。

つまり、20kHz を対象としたリアルタイム・シミュレータで必要とされる、制御周期 $50\mu\text{sec}$ の $1/5 \sim 1/20$ の計算ステップで、“標準的な回路定数を持つスイッチ回路”を計算した場合、陽解法で安定的に計算可能な計算ステップを見出せる可能性は非常に高いのである。

以上の結論は一種の盲点であり、これを指摘した文献はこれまでにない。

後は、使用可能な計算手段が、上記の $2.5\mu \sim 10\mu\text{sec}$ の計算ステップにおいて、どの程度の計算速度を実現できるかで、計算ステップを決めれば良いことになる。なお、第 5 章で示すように、 $10\mu\text{sec}$ 程度の計算ステップであれば、現在普通に入手可能な 1GHz クラスの汎用プロセッサで十分に実装可能である。

逆に、陰解法を採用した場合、計算の安定性はあるものの、計算負荷も高くなるため、陽解法では満たせた $2.5 \sim 10\mu\text{sec}$ が満たせなくなる可能性が高く、リアルタイム・シミュレータとして成立しない可能性がでてくる。

同様の考察は Runge-Kutta 法の次数の選定についてもあてはまる。一般に Runge-Kutta 法とは 4 次公式が知られているが、これは計算ステップ幅と精度のバランスが良く、最終的な計算時間が最も短いとされることによるが、上記のようにリアルタイム・シミュレータでは計算ステップ幅に上限が存在する。従って、計算ステップ幅を長くすることができず、次数を増やす動機がなくなる。現時点では経験則であるが、固有値の $1/20$ 以下の計算ステップであれば、1 次 (つまり前進 Euler 法)、それ以上でも 2 次の Runge-Kutta で十分な精度が得られている。

以上、結論は 1 次ないし 2 次の Runge-Kutta 法であるが、この結論には、“標準的な回路定数を用いた場合”という但し書きがあり、またダイオードを用いた昇圧／降圧回路で電流不連続モードが発生した場合や、ブリッジ回路において 4.4 節で検討したデッドバンド状態において V_f が無視できない場合などは、方程式の固有値が大きくなることが解ってお

り、上記の前提が当てはまらなくなる。

この点は、所謂 **Semi-Implicit**[51]として知られる、陰解法を近似的に陽解法に変換するなど処理を行えば対応可能と思われる。今後の研究課題とする。

よって結論は、

計算アルゴリズム	: 部分的可変ステップ法
回路方程式:	: カットセット解析による状態方程式
常微分方程式の解法	: 1 次ないし 2 次の Runge-Kutta 法

である。

以後、第 5 章において汎用プロセッサおよび **FPGA** による実装を行い、効果を検証する。

本回路では、インダクタ値を変化させることで、特性がどのように変化するかを観察するため L1 と L2 という 2 個のインダクタを直列に置き、L2 をバイパスするスイッチを設けている。また、出力キャパシタ C2 に並列にキャパシタ C7 を接続し容量を増加させるスイッチ、さらに ESR を増加させるための ESR 模擬抵抗を接続するスイッチを設けているが、現段階で機能させているのは、ESR 模擬抵抗接続スイッチのみである。

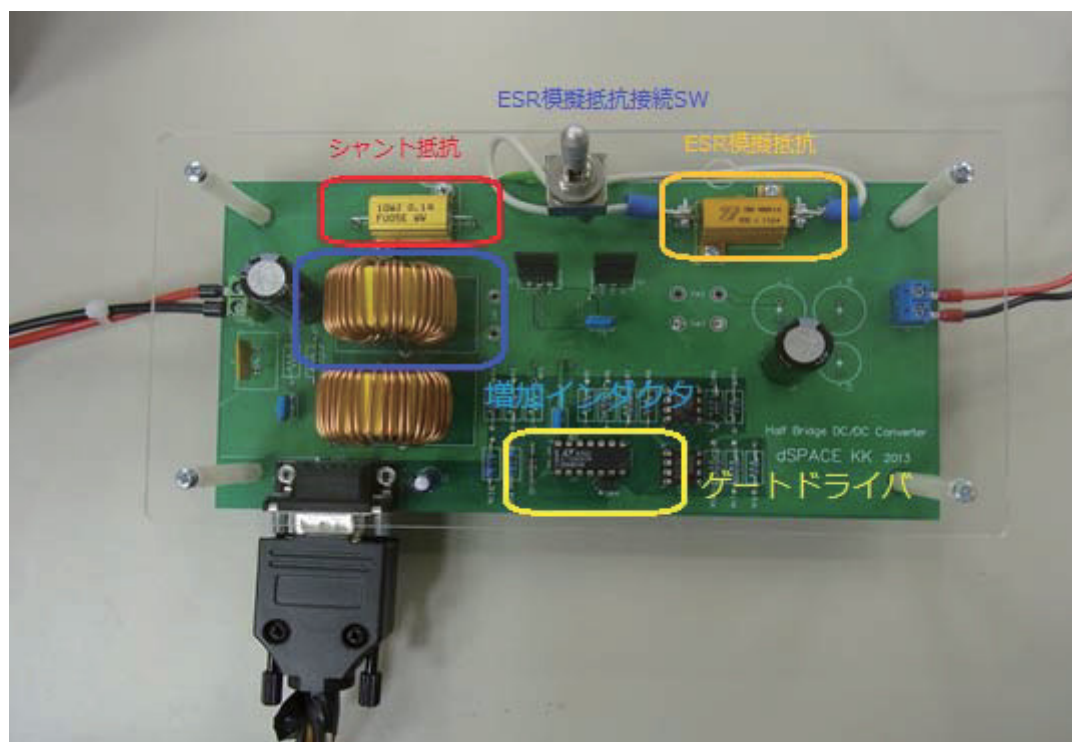


図 5.2 実験用回路の外観

FET は 60V 耐圧の 2SK2232 を選定し、ゲートドライバは LT1160 を使用した。9 ピンの DSUB コネクタに接続した TTL レベルの外部信号により、ハーフブリッジのスイッチ素子を独立に駆動する。このためデッドバンド時間は任意に設定可能であり、また駆動停止 (Duty=0%/100%) も可能であるが、LT1160 がブートストラップ方式のため、ローサイドパルス停止でのハイサイドスイッチの駆動はできない。

以上が、実際の回路構成であるが、使用していない機能を省いた等価回路を図 5.3 に示す。以後の回路方程式に使用する素子シンボル名は図 5.3 によるものである。

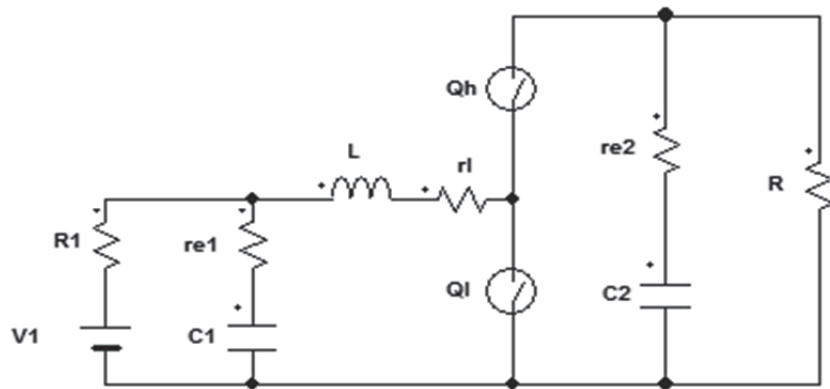


図 5.3 実験用回路の等価回路

5.1.2. 導出された回路方程式

3.1 節で用いた基本法則を用いて、KCL、KVL を用いて手計算で導出した状態方程式は以下である。U は入力ベクトル、X は状態変数ベクトルであり、ベクトルの要素は L 電流、C1 電圧、C2 電圧である。また、C、D 行列による出力値は R（負荷抵抗）の両端電圧である（実験においては L 電流も出力値として扱っているが、L 電流は状態変数から直接算出可能なため、C、D 行列より割愛している）。

$$U = V1 \quad (5-1)$$

$$\frac{d}{dt}X = AX + BU \quad (5-2)$$

$$A = \begin{bmatrix} \frac{-1}{L} \left(r_l + \frac{R_1 r_{e1}}{R_1 + r_{e1}} + \frac{r_{sl} r_{sh}}{R^S} + \frac{r_{sl}}{R^S} \frac{R_2 r_{e2}}{R_2 + r_{e2}} \right) & \frac{1}{L} \frac{R_1}{R_1 + r_{e1}} & \frac{-r_{sl}}{L R^S} \frac{R_2}{R_2 + r_{e2}} \\ \frac{-1}{C_1} \frac{R_1}{R_1 + r_{e1}} & \frac{-1}{C_1} \frac{1}{R_1 + r_{e1}} & 0 \\ \frac{r_{sl}}{C_2 R^S} \frac{R_2}{(R_2 + r_{e2})} & 0 & \frac{-1}{C_2} \frac{1}{(R_2 + r_{e2})} \end{bmatrix} \quad (5-3)$$

Q1 の抵抗抵抗値を r_{sl} 、Qh の抵抗値を r_{sh} とし、 $R^S = r_{sl} + r_{sh}$ とする。

$$B = \begin{bmatrix} \frac{1}{L} \frac{r_{e1}}{R_1 + r_{e1}} \\ \frac{1}{C_1} \frac{1}{R_1 + r_{e1}} \\ 0 \end{bmatrix} \quad (5-4)$$

$$C = \begin{bmatrix} \frac{r_{s1} r_{e2} R^C}{R_2 + r_{e2} (1 + R^C)} & 0 & \frac{R_2}{R_2 + r_{e2} (1 + R^C)} \end{bmatrix} \quad (5-5)$$

$$D = \begin{bmatrix} 0 & \frac{r_{e2}}{R_2 + r_{e2} (1 + R^C)} \end{bmatrix} \quad (5-6)$$

一方、回路を図 5.4 のような基本木と補木に分解する事で、カットセット解析を適用する事ができる。選択した基本木を図 5.4 に示す。この基本木に対する、カットセット行列 Q を式(5-7)に示す。これより、4.2 節の計算手順に従うと、A、B 行列を導くことができる。式(5-7)を用いて A、B 行列を導出する m-script を添付付録のリスト 4 に示す。

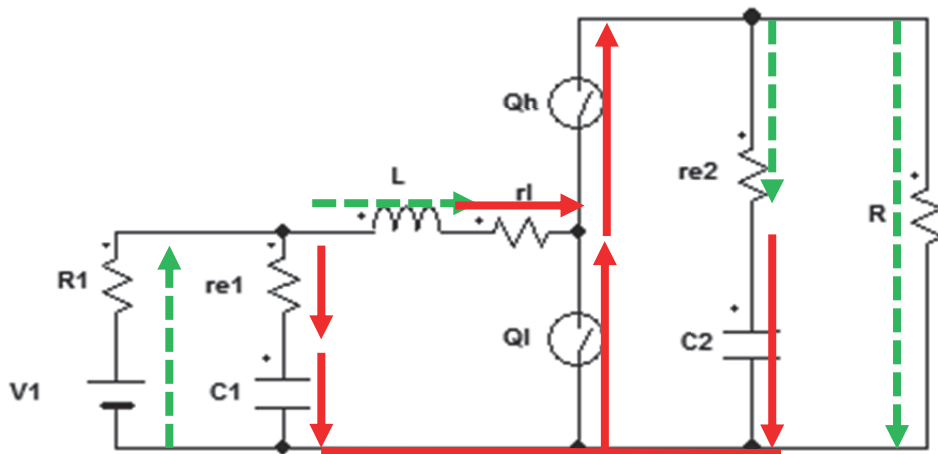


図 5.4 選択した標準木

$$Q = \begin{array}{c} \begin{array}{cccccc} C1 & C2 & Ql & Qh & re1 & rl \end{array} \\ \left[\begin{array}{cccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{array} \right] \begin{array}{c} L \\ rs \\ re2 \\ R \end{array} \end{array} \quad (5-7)$$

なお、現段階では、回路図よりカットセットを構成する木を選ぶ作業は手作業で選定しているが、これを自動的に算出する手法は参考文献[42]に紹介されている。また、実用的には A、B 行列だけでなく C、D 行列の算出が必要である。今後は、参考文献[42]を参考に、カットセットの選定の自動化および A、B、C、D 行列の自動生成を行う予定である。

5.1.3. 回路定数の設計

スイッチング周波数は 20kHz としたが、コンバータ回路の代表的な指針として、「インダクタ（図 5.3 の L）および出力キャパシタ（図 5.3 の C₂）で構成されるローパスフィルタのカットオフ周波数をスイッチング周波数の 1/50 とする」[52]、ものがある。

この指針に従うと、

$$\frac{1}{2\pi\sqrt{LC_2}} = 400 \quad (5-8)$$

であり、

$$LC_2 = \left(\frac{1}{2\pi 400} \right)^2 \cong 1.6 \times 10^{-7} \quad (5-9)$$

となる。

定数値 200 μH（ESR メータによる実測値は 180 μH）のインダクタを使用した、L をこの値とすると、式(5-9)より C₂は 800 μF が必要となるが、シミュレータの性能を確認するのが目的であるから、リップルが大きめに出るように 220 μF とした。

実際にはキャパシタの寿命を縮めるリップル電流の低減を重視し 1mH 程度のインダクタの使用例が多い事から考えると、この設定は応答性／インダクタサイズを重視した、少々特殊な設計である。

以下に使用した素子の定数（一部は実測値）を示す。

表 5-1 作成した実験回路の素子定数

素子名	定数
L	180 μ H (素子単体の実測値)
C ₁	220 μ F
C ₂	220 μ F
R ₁	100m Ω (電圧源の内部抵抗)
r _{e1}	60m Ω
r _{e2}	160m Ω (模擬 ESR 抵抗を含む設計値)
r _l	300m Ω (シャント抵抗値を含む)
r _{sh}	30m Ω (ON 時) / 1M Ω (OFF 時)
r _{sl}	30m Ω (ON 時) / 1M Ω (OFF 時)

5.1.4. 安定性解析

5.1.3 節で示した素子定数において負荷抵抗 R を変化させて固有値を調べた。

この回路定数であれば、h=10sec の計算ステップ幅において、4 次以下の Runge-Kutta 法で十分な安定度を得ることができることを図 5.5 に示す。赤文字は最も安定性に影響を与える最大固有値である。

表 5-2 作成した実験回路の固有値

Ql (ローサイド) =ON			Qh (ハイサイド) =ON		
R=			R=		
10 Ω	100 Ω	1k Ω	10 Ω	100 Ω	1k Ω
-2.80e4	-2.80e4	-2.80e4	-2.8e4	-2.8e4	-2.8e4
-2.42e3	-2.42e3	-2.42e3	-1.87e3	-1.68e3	-1.66e3
-447	-45	-4.5	$\pm 4.7e3i$	$\pm 4.8e3i$	$\pm 4.8e3i$

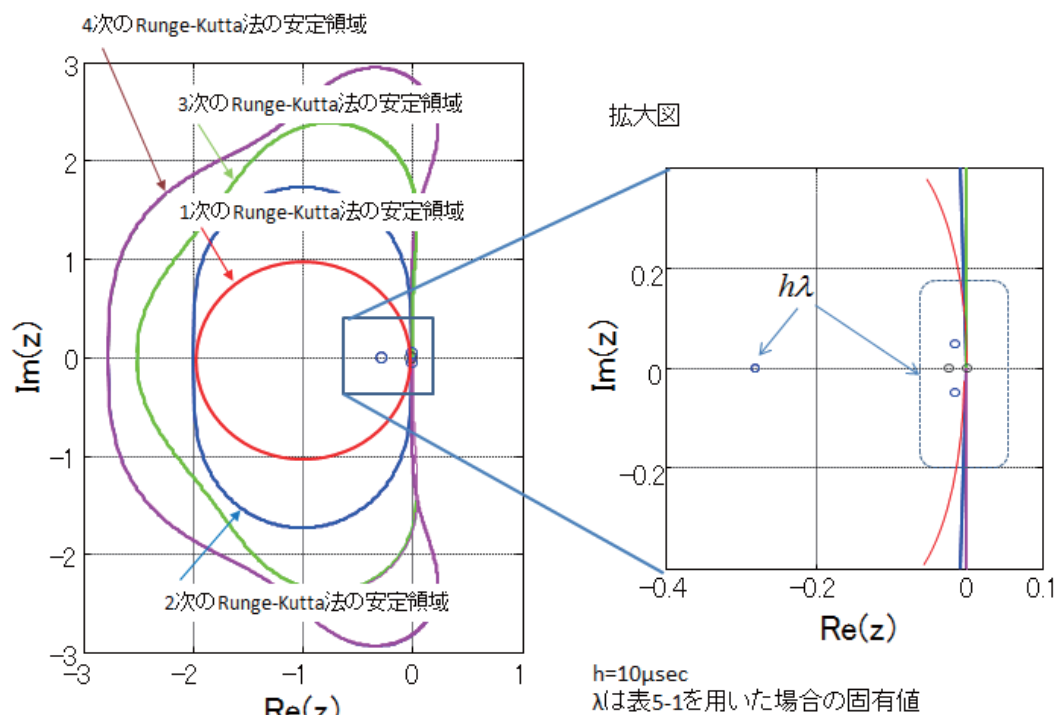


図 5.5 作成した回路定数を用いた固有値と安定領域 ($h=10\mu\text{sec}$)

表 5-1 に対し、素子の定数が、より小さい方が計算安定性的には厳しいため、参考として、より小さい定数での固有値を計算した。

表 5-3 スイッチング周波数 20kHz で限界と思われる回路定数

素子名	定数
L	$100\mu\text{H}$
C_1	$47\mu\text{F}$
C_2	$100\mu\text{F}$
R_1	$100\text{m}\Omega$ (電圧源の内部抵抗)
r_{e1}	$50\text{m}\Omega$
r_{e2}	$50\text{m}\Omega$
r_l	$10\text{m}\Omega$
r_{sh}	$10\text{m}\Omega$ (ON 時) / $10\text{M}\Omega$ (OFF 時)
r_{sl}	$10\text{m}\Omega$ (ON 時) / $10\text{M}\Omega$ (OFF 時)

表 5-3 の素子定数ではリップル電圧／電流がかなり大きく、実用的な回路設計とは言えないが、この値においても、固有値は表 5.4 であり、最大固有値の移動は少なく、 $h=10\mu\text{sec}$ にて Runge-Kutta 法の安定領域に、十分に入ることが図 5.6 によりわかる。

表 5-4 限界と思われる回路定数による固有値

Ql (ローサイド) =ON			Qh (ハイサイド) =ON		
R2=			R2=		
10Ω	100Ω	1kΩ	10Ω	100Ω	1kΩ
-3.3e4	-3.3e4	-3.3e4	-3.3e4	-3.3e4	-3.3e4
-2.1e3	-1.2e3	-1.2e3	-1.9e3	-913	-818
-1.2e3	-210	-21	$\pm 1.5e-4i$	$\pm 1.5e4i$	$\pm 1.5e4i$

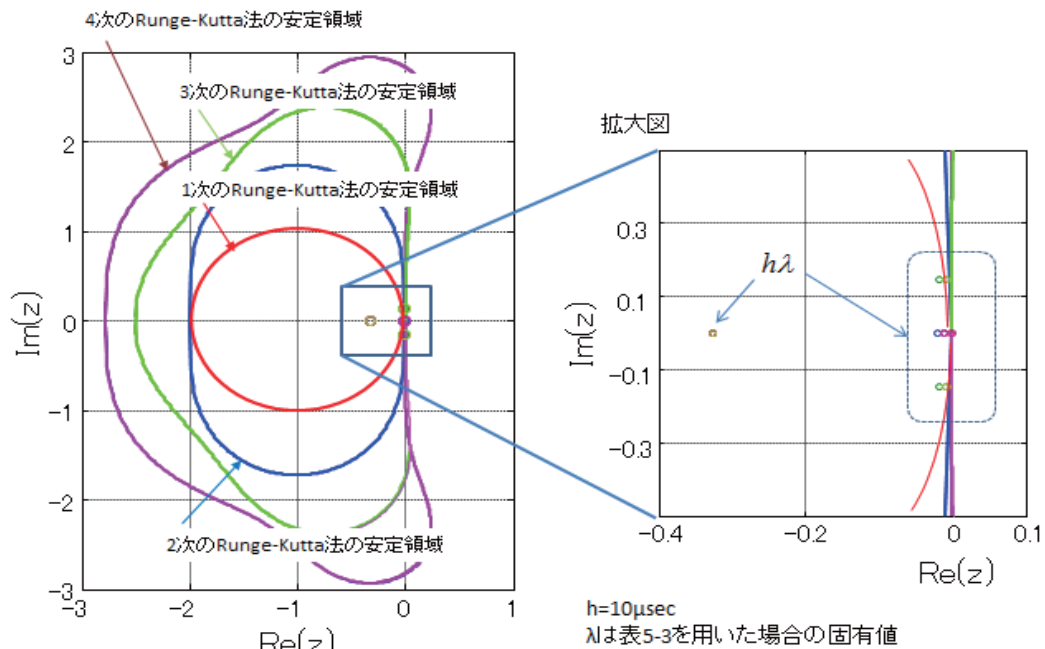


図 5.6 限界と思われる回路定数を用いた固有値と安定領域 ($h=10\mu\text{sec}$)

以上より、4.7 節で指摘した、実用的なコンバータ回路を計算対象とする場合は、陽解法の Runge-Kutta 法で十分に計算可能であることが確かめられた。

5.2. 汎用プロセッサによる実装

まず、基本的な性能を確認するために、汎用プロセッサを用いてリアルタイム・シミュ

レータを構成した。計算アルゴリズムの実装には MATALB/Simulink を用い、dSPACE 社の MBD ツールで主に RCP で用いられる DS1103 汎用コントローラボードと、DS1103 と通信を行い、アプリケーションプログラムのダウンロードと計算内容の確認モニタを行う、Control Desk Ver3.7 を使用した。

計算ステップ幅は、4.7 節での検討のように $2.5\mu\text{sec} \sim 10\mu\text{sec}$ の間で選ぶことが必要であるが、5.1.4 節での検討のように $10\mu\text{sec}$ 以下であれば、計算安定性は確保できる。DS1103 を用いた場合は $3.5\mu\text{sec}$ まで短縮することが可能であった。この値は 20kHz の約 $1/14$ の値であり、十分な計算速度と言える。なお、厳密には端数が存在するため、14 回の計算 ($3.5 \times 14 = 49\mu\text{sec}$) を行うと、PWM 信号の立ち上がり同期するまで、計算を停止するようにしている。なお、本章では、スイッチ駆動信号は PWM 信号に限定し、以後 PWM 信号と呼ぶことにする。

表 5-1 プロセッサ実装モデルの概要

使用ハードウェア	DS1103 (IBM PowerPC 750FX 1GHz)
回路方程式	状態変方程式 (カットセット解析)
常微分方程式の解法	2 次の Runge-Kutta 法 (Heun 法)
計算アルゴリズム	部分的可変ステップ法 (PWM 同期方式)
デッドバンド	対応不可
数値変数の型	倍精度浮動小数点 (IEEE754)
計算ステップ幅	$3.5\mu\text{sec}$

5.2.1. Simulink モデル

部分的可変ステップ法を Simulink で実装したブロック図を図 5.7 に示す。Do-While サブシステム内に StateFlow のブロックと 2 次の Runge-Kutta 法である Heun 法による状態変数計算ブロックが含まれている。StateFlow 処理の詳細は図 5.8 で示すが、PWM 信号の ON 時間 (StateFlow 内では変数 TsTime) が現在の計算時刻を超えた場合、計算ステップの“あまり”である h_{hf} を算出すると同時に、変数 WhileCond を TRUE に設定する。これにより上記の Do-While サブシステムが 2 回計算を行うという内容になっている。

このサブシステムは While サブシステムのため、
スイッチ動作が起きた時は、2回実行される

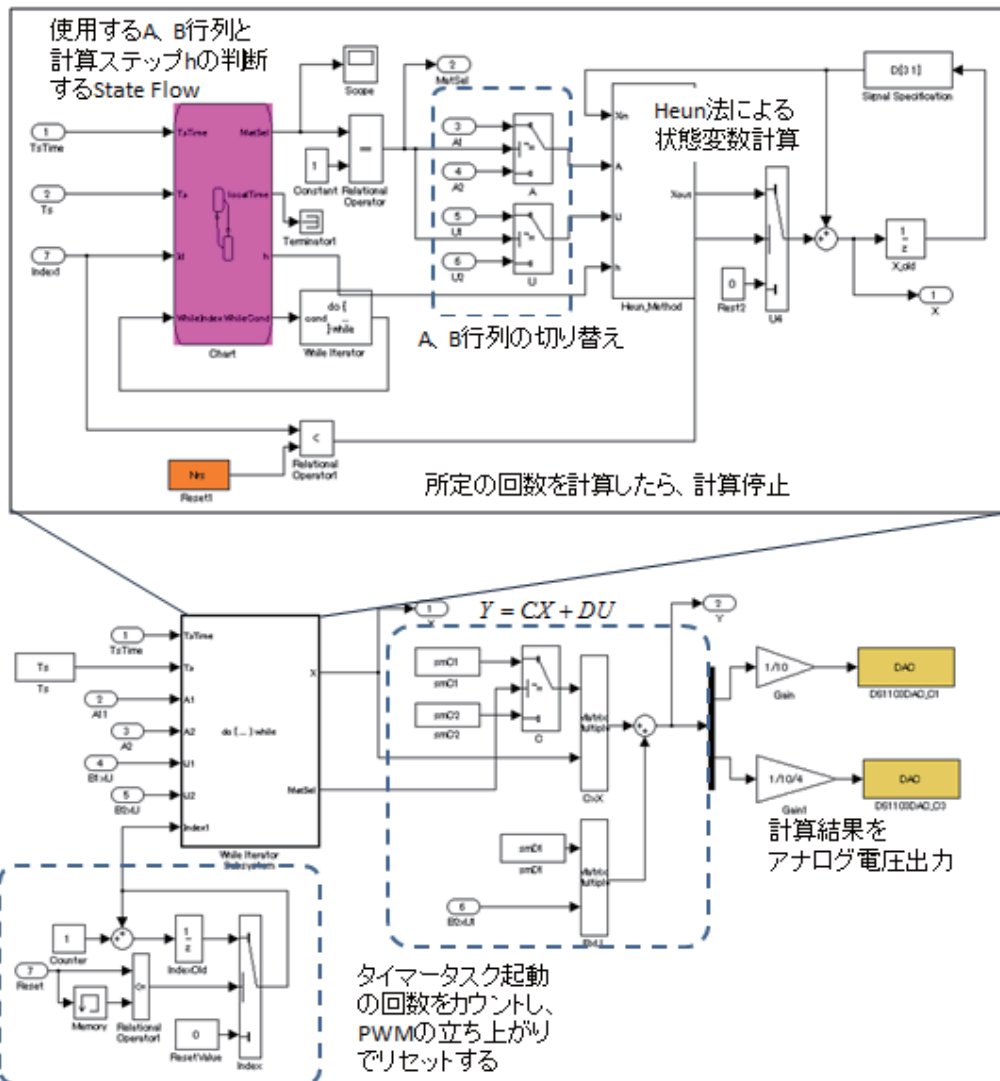


図 5.7 部分的可変ステップ法（同期）の Simulink モデルによる実装

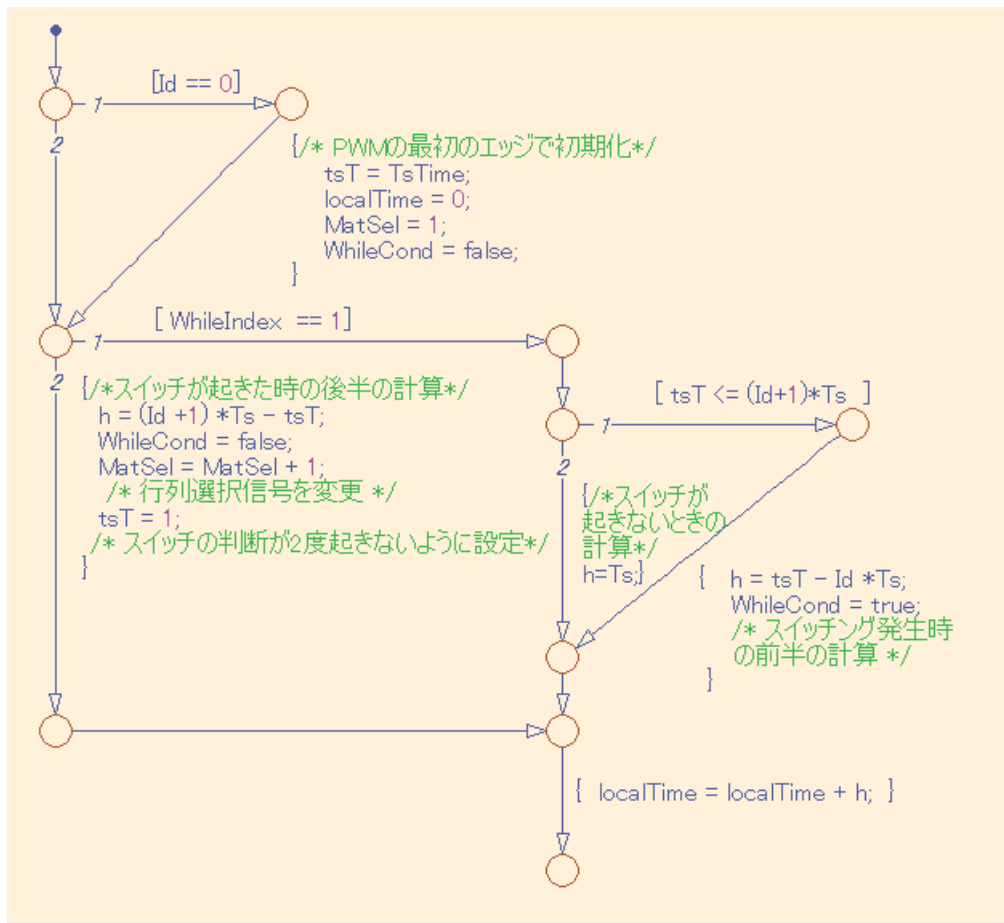


図 5.8 主要な処理を行う StateFlow の ControlFlow

5.2.2. dSPACE 社 RTI ブロックによる I/O 実装

リアルタイム・シミュレータは、模擬対象を計算するだけでなく I/O 機能が必要である。従来、このような I/O 機能を使用するためには複雑な I/O ドライバをアセンブラ等で記述する必要があり高度なスキルが要求されたが、MBD ツールを用いることで、回路方程式のみならず、I/O 機能も、Simulink のブロック線図内のブロックとして用いることができる。

図 5.9 は今回使用した、DS1103 に使用する RTI と呼ばれる I/O ブロックである。



図 5.9 リアルタイム・シミュレータの作成に用いた RTI ブロック

5.2.3. 実装上の問題

MBD では、開発を容易にする様々なツールが用意されているが、これはいわばレディメイドの既製服であり、特殊な要求には対応できない場合がある。

本シミュレータは図 5.10 のようにタスク構成を行っているが、DS1103 に用意されている汎用の OS では、一定間隔で起動する定時タスクと割り込みにより起動されるトリガータスクの選択が可能である。ところが、同期型の部分可変ステップ法では、PWM 信号に同期し、さらに一定間隔で処理を行う、定時タスクとトリガータスクの双方の性格を持つタスクを必要とし、DS1103 ではこれが実現できない。すなわち、図 5.10 では $3.5 \mu \text{sec}$ のタスクは PWM 信号に同期しない。

また部分的可変ステップ法は、スイッチ動作発生時刻を正確に計測する必要がある。DS1103 には様々な RTI ブロックが用意されており、図 5.9 の Duty 比と周波数を計測するブロックを使用した。このブロックは周波数を 1 周期計測した後に Duty 比が確定するものであり、Duty 比の計測が 1 周期遅れる。さらにこのブロックはハイサイドスイッチ駆動 PWM 信号の Duty 比を計測するだけであり、ハイサイド、ローサイド駆動に対してデッドバンド時間を設けても計測できない。

従って、プロセッサベースでの実装は、同期型の部分的可変ステップ法を目指したものの、Duty 比の計測のため処理が 1 周期遅れる（状態平均化法の欠点）、PWM 信号に同期できないためリプル電圧／電流の位相がずれる（非同期型の部分ステップ法の欠点）、Duty 比を計測しているため、スイッチ駆動停止（Duty 比=0%/100）に対応できない、など多くの欠点を持つものになっている。

これらを解消するためには、自前の OS、I/O システムに対する直接のコーディングが必

要であるが、それでは MBD ツールを使用し、実装を容易にするという目的に反するため、現状仕様としている。

しかしながら、原理確認モデルとしては、十分であると考えている。

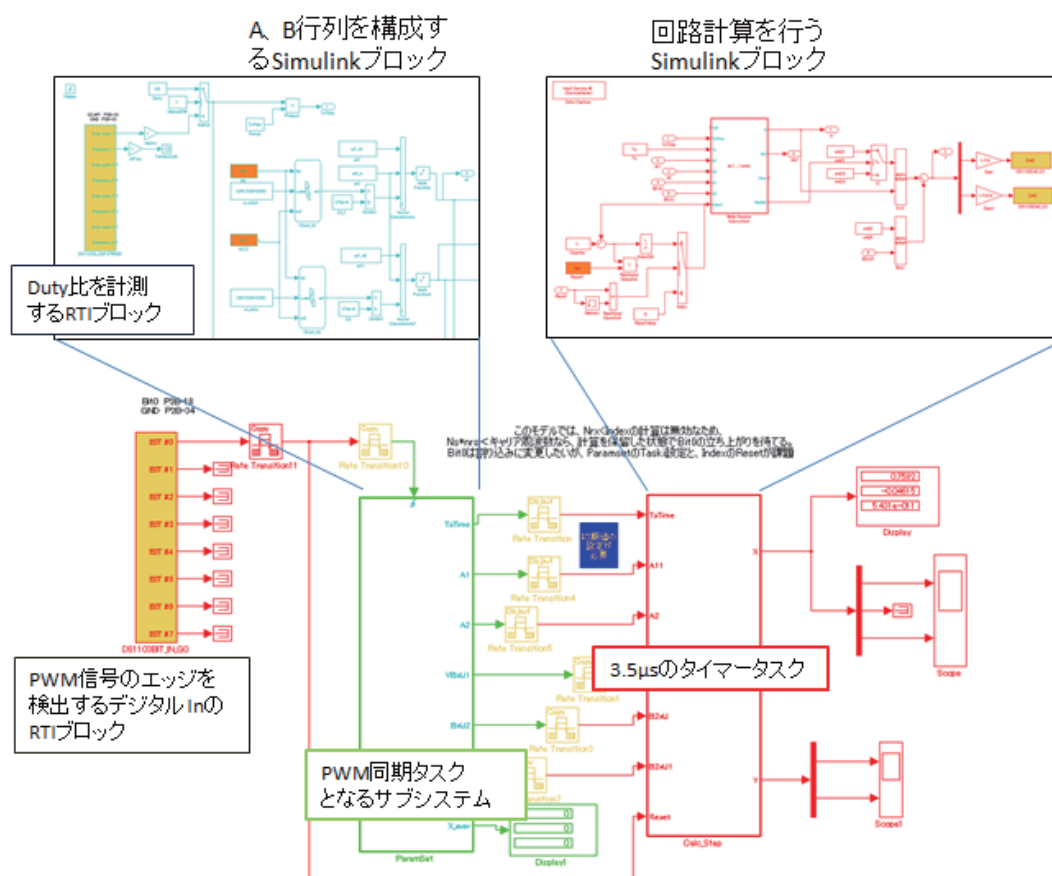


図 5.10 Simulink 実装におけるタスク配分

5.2.4. 実験結果（実回路との比較）

図 5.11 のようにファンクションジェネレータで発生させた PWM 信号をサンプル回路と DS1103 を使用したシミュレータに与え、双方の出力を比較した結果を示す。

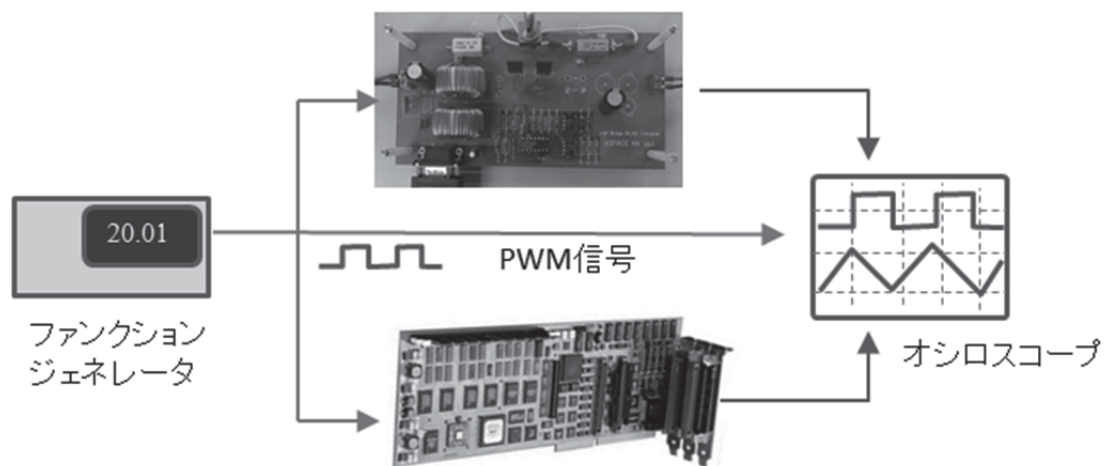


図 5.11 比較実験の構成

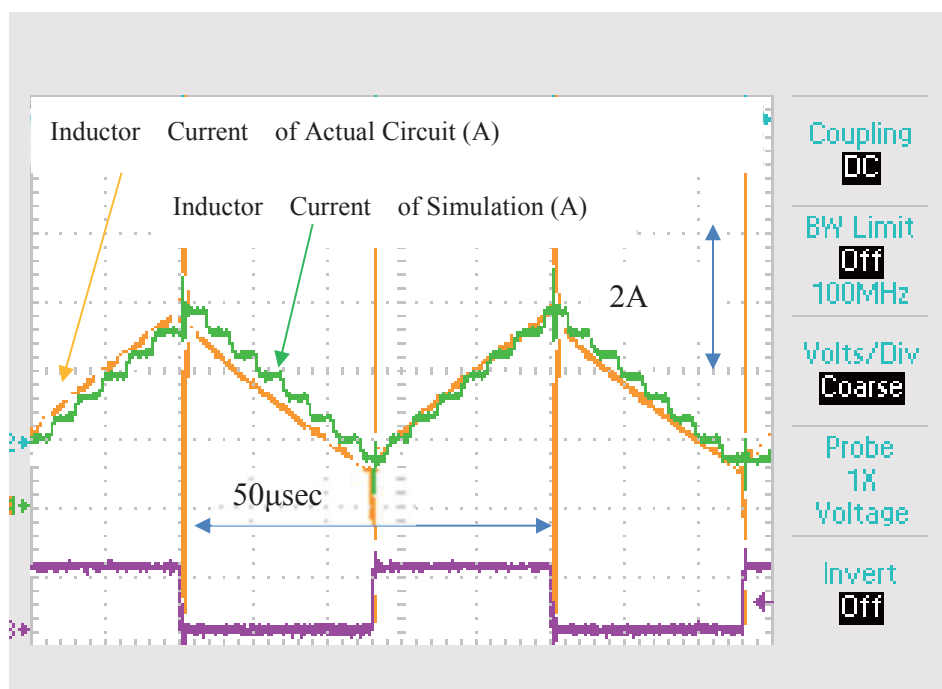


図 5.12 実回路とシミュレーションのインダクタ電流の比較

図 5.12 に実回路の計測結果およびリアルタイム・シミュレータのインダクタ電流の出力結果を示す。V1(電源電圧)=12.0V、Duty 比=0.5、 $R=56\Omega$ とし、スイッチング周波数は 20kHz である。実回路においてはスイッチング動作に伴うスパイクノイズが観測されるが、シミュレータ出力にノイズは無い。この実験によるインダクタ電流の傾きより、実回路の実装状態におけるインダクタ値として $270\mu\text{H}$ を推定し、以後シミュレータ側はこの値を用いている。

なお、実回路のインダク電流は 0.1Ω のシャント抵抗で計測しており、シミュレータの電流／電圧比率は 1:1 のため、オシロスコープ上のレンジは 50mv/dev と 500mV/dev での表示である。

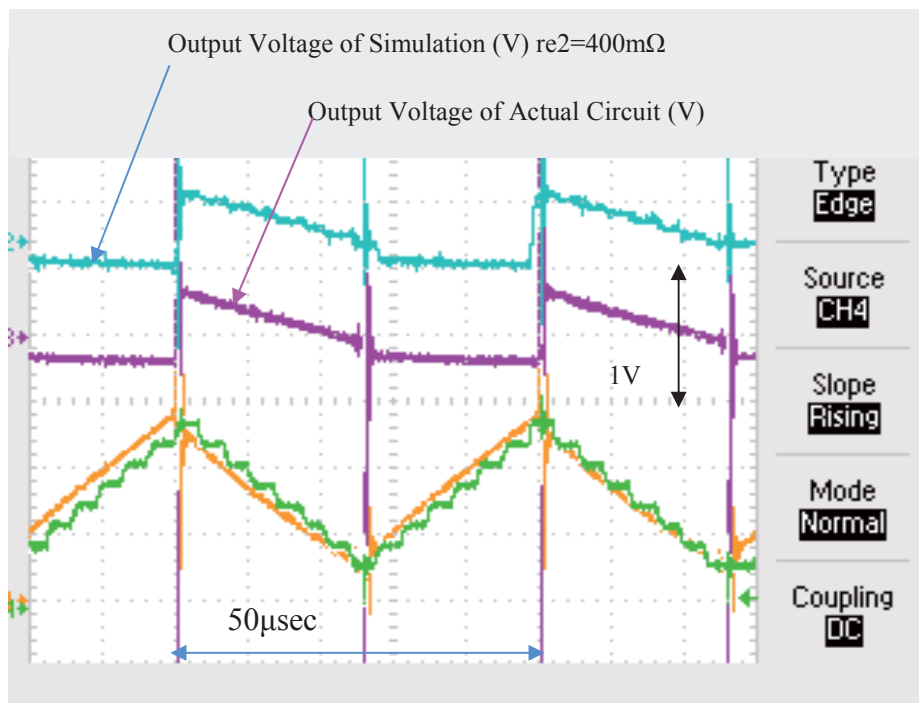


図 5.13 実回路とシミュレーションのリプル電圧の比較

図 5.13 に実回路とシミュレータの出力電圧(負荷抵抗 R の両端電圧)の比較を示す。リプル電圧を確認するため、オシロスコープは AC カップリングで使用している。ESR の設計値は ESR メータにより 10kHz の周波数で測定した C_2 コンデンサの単体実測値 ($60m\Omega$) と ESR 模擬抵抗 $100m\Omega$ の合計値である $160m\Omega$ であるが、実験結果では $400m\Omega$ を用いてシミュレーションした結果と合致した。実回路では ESR 模擬抵抗接続スイッチの接触抵抗で抵抗値が増加しているものと思われる。

なお、シミュレータ出力は D/A コンバータの出力範囲が $\pm 10V$ であるため、1/2.5 としている。従って、オシロスコープのレンジは、実回路は 500mV/dev であり、シミュレータは 200mV/dev として表示している。

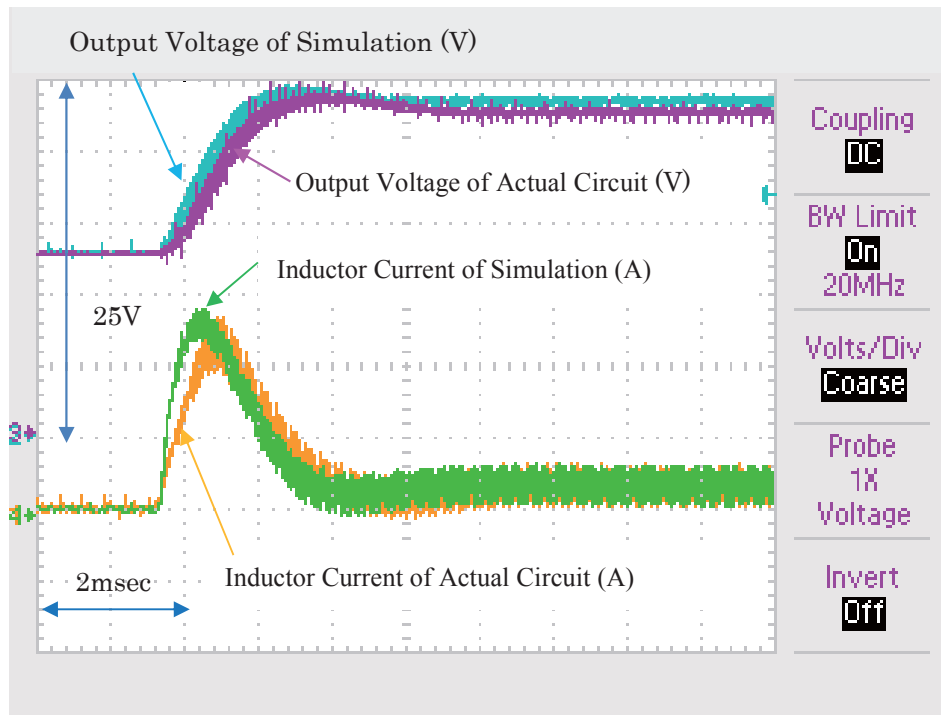


図 5.14 実回路とシミュレーションのステップ応答の比較

図 5.14 に Duty 比を 0.1→0.5 に変化させたときのステップ応答を示す。電流、電圧の立ち上がりはシミュレーションの方が若干早い、良好な結果とすることができる。

5.3. FPGA による実装

5.2 節での汎用プロセッサを用いた実装での課題は、汎用の RCP ツールを用い、dSPACE の汎用のリアルタイム・カーネルを用いた結果、PWM 信号にタスクが同期できない等の問題の他にも、タスクスイッチ遅延や D/A コンバータの I/O ドライバの遅延のため、 $3.5\mu\text{sec}$ 以下への計算ステップの短縮が困難であるという問題があった。

これを解決するために、dSPACE 社の FPGA ボードである、DS5203/SX95 ボードを用いて実装を行った。なお、FPGA とは、Field-Programmable Gate Array の略であり、製造者ではなく、使用者が回路構成を変更する事が可能な集積回路である。ロジック IC を組み合わせた回路をソフトウェアにより実現する事ができるが、一般的なプロセッサがソフトウェアによりアルゴリズム（計算手順）を定義する事に対し、FPGA では、ソフトウェアにより回路構成を定義する事になり、プログラムの考え方は大きく異なる。

使用したハードウェアと計算アルゴリズムを表 5-2 に記載する。

表 5-2 FPGA 実装モデルの概要

使用ハードウェア	DS5203 (Xilinx Vertex-5 100MHz) DS1005 (PowerPC 1GHz)
常微分方程式の解法	前進 Euler 法
回路方程式	状態変方程式 (カットセット解析)
計算アルゴリズム	部分的可変ステップ法 (PWM 非同期方式)
デッドバンド	対応可能
数値変数の型	32bit 固定小数点
計算ステップ幅	200nsec

DS5203 を使用することにより、FPGA ボードに実装された高速な A/D コンバータ、D/A コンバータ、デジタル I/O を使用する事ができる。

なお DS5203 はホストプロセッサボード (DS1005) と組み合わせられ、計算の設定の変更や計算結果の表示はプロセッサボードがホスト PC と通信して行う。この構成を利用して、計算に使用する入力値や A、B、C、D 行列の構築など、複雑であるが時間的な余裕がある処理はプロセッサボードに任せ、高速計算が必要な部分のみを FPGA に実装している。

5.3.1. Xilinx 社 SystemGenerator を用いた回路モデルの実装

FPGA に対するプログラムは、Simulink とよく似たブロック線図による記述が可能な Xilinx 社の System Generator を用いた。通常 FPGA はハードウェア記述言語 (HDL) でプログラムされるが、System Generator を用いることで、容易に FPGA へのプログラムを行う事ができる。System Generator で使用するブロックの一例を図に示す。

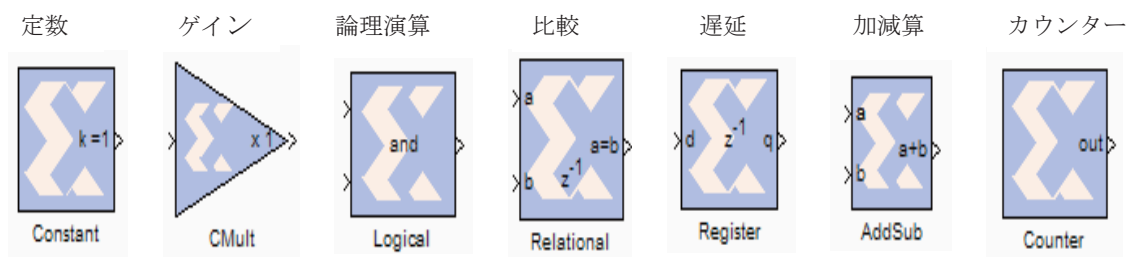


図 5.15 System Generator の代表的な演算ブロック

5.3.2. dSPACE 社 RTI-FPGA ブロックを用いた I/O 実装

本システムにおいては 2 種類の I/O が存在する。一つは、オンボードの A/D コンバータ、

D/A コンバータ、bit-I/O であり、もう一つはプロセッサボードとの通信をサポートするものである。両者は、5.2 節のプロセッサ実装の時と同じように、dSPACE 社の RTI-FPGA を用いることで、容易に Simulink および System Generator モデルに統合することができる。

図 5.16 に概略図を示す。水色の RTI ブロックが、プロセッサとの通信に使用する RTI であり、オレンジ色のブロックがオンボードの I/O による外部信号を扱う RTI ブロックである。

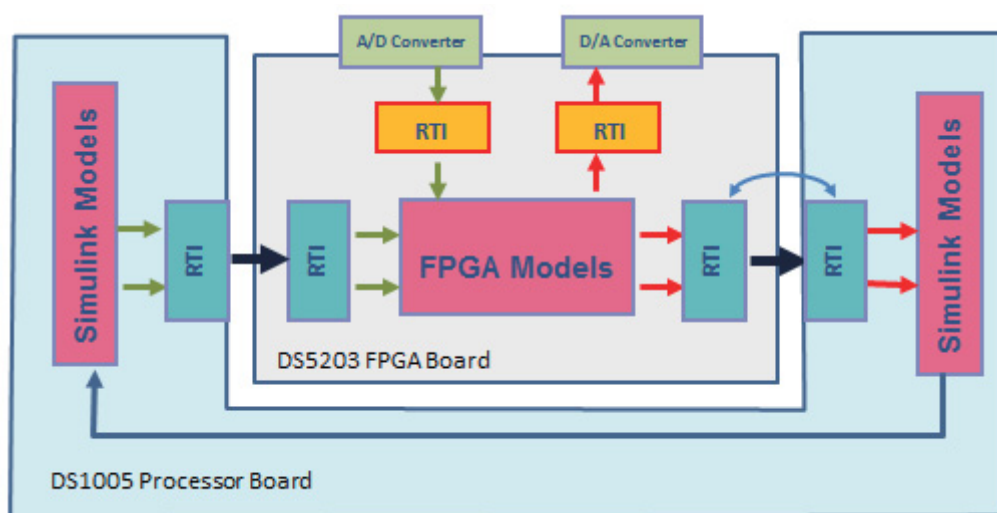


図 5.16 2 種類の RTI-FPGA

5.3.3. モデルの構造

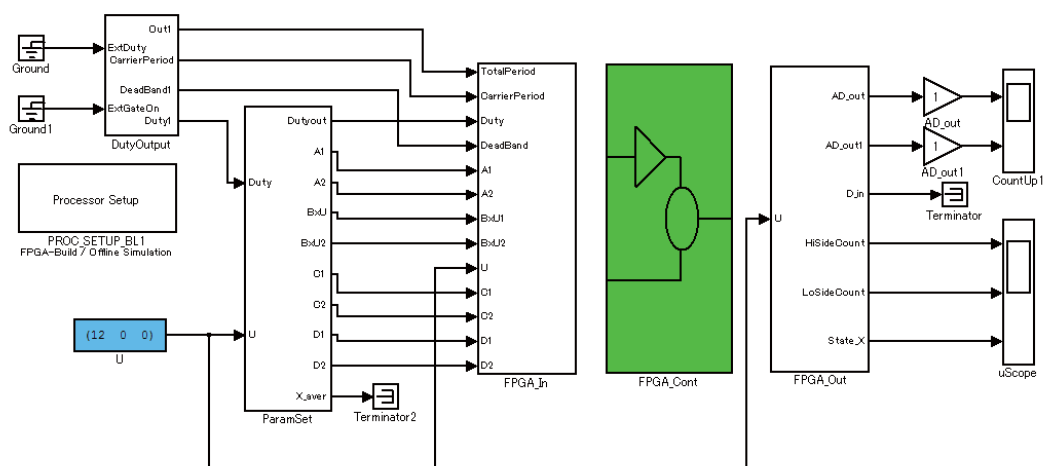


図 5.17 Simulink モデルの全体構造

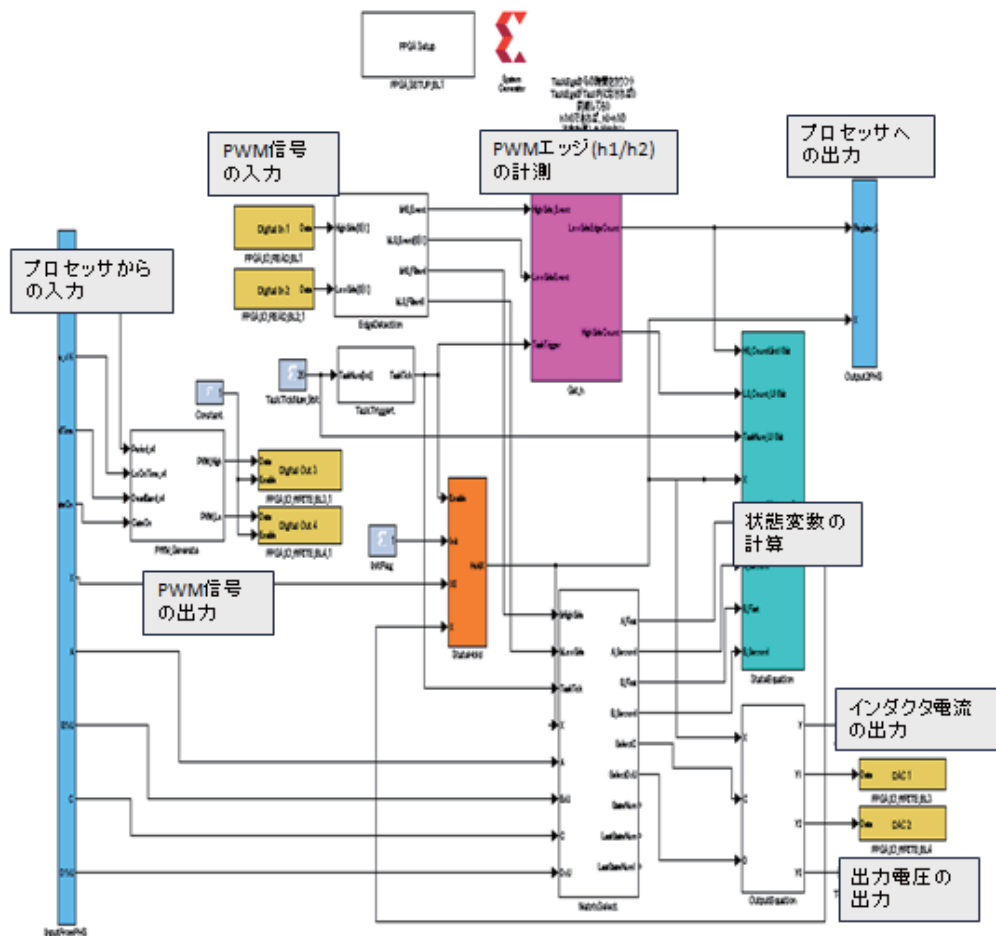


図 5.18 FPGA による非同期型部分的可変ステップ法の実装

Simulink モデルの全体構造を図 5.17 に示す。緑の部分が FPGA モデルであり、この部分が DS5203 で動作する。残りの Simulink モデルでは、A、B、C、D 行列の構築や、固定小数点データへの変換、FPGA への転送データの構築を行っている。FPGA モデルを図 5.18 に示す。黄色のブロックが外部 I/O を用いる RTI ブロックであり、水色のサブシステム内にプロセッサの通信用の RTI ブロックが含まれている。

本実装において、核となるのは状態変数の計算部分であるため、図 5.19 に計算のタイミングを示す。

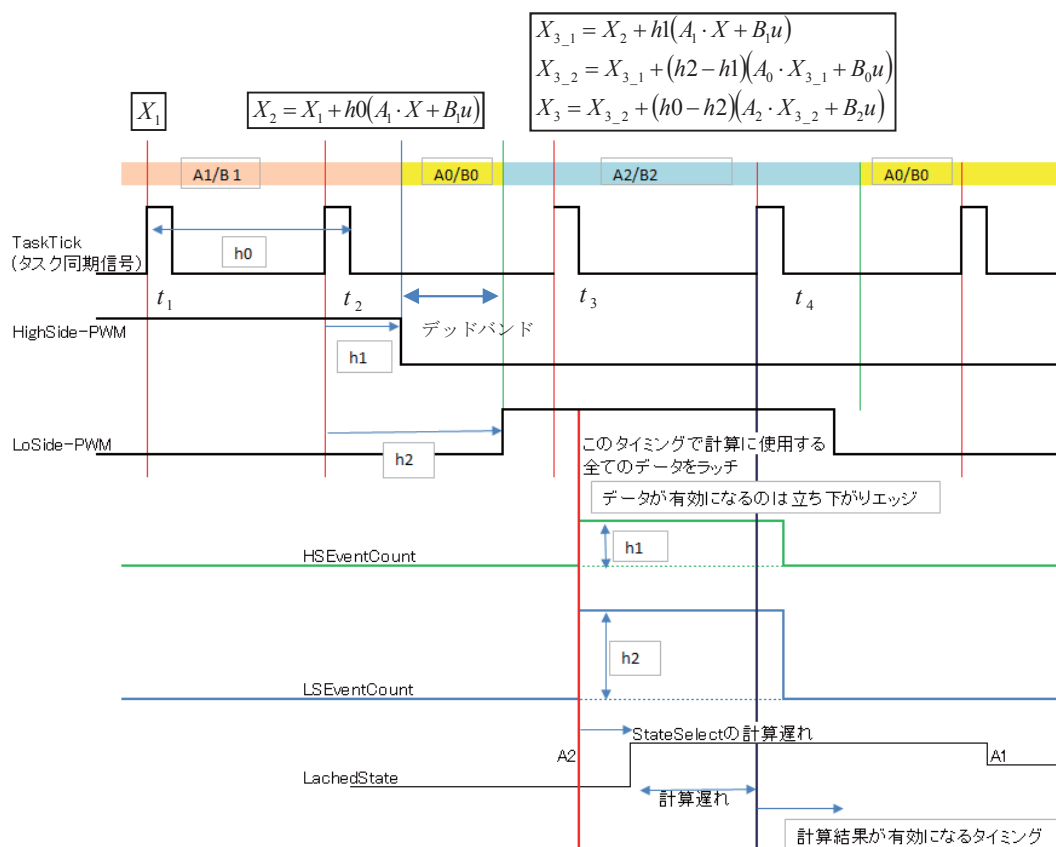


図 5.19 状態変数計算のタイミング図

対象とするコンバータ回路はハーフブリッジ型であるので、ハイサイド側とローサイド側の PWM 信号を入力とする。また、デッドバンドの処理も可能としている。計算は TaskTick と呼ぶ信号に同期して所定の間隔で行うが、FPGA の場合、実際にタスクが起動されるわけではなく、TaskTick はデータラッチのタイミングを決める Enable 信号である。

図 5.19 において最初の TaskTick(t_1)で X_1 が計算されるならば、次の X_2 は、その時点で有効である A、B 行列、すなわち、ハイサイド側のスイッチが ON した場合の状態方程式を用いて状態変数の計算が行われる。 $t_1 \sim t_2$ までにスイッチ動作は起きていないので、 X_2 は TaskTick の間隔 ($=h0$)で計算される。

$t_2 \sim t_3$ までの間、ハイサイド、ローサイドがそれぞれスイッチ動作を行うが、この時間、 $h1$ および $h2$ は後述するサブシステムにより HiEventCount 値、LoEventCount 値として FPGA のクロック数がカウントされており、それぞれ $h1$ 、 $h2$ として、 t_3 の TaskTick のタイミングでラッチされる。 t_3 のタイミングでは、この $h1$ 、 $h2$ を用いて、現在のローサイドが ON の場合の方程式 (A_2/B_2)、ひとつ前の状態であるハイサイド、ローサイドともに OFF の時の方程式 (A_0/B_0 ：実際には電流の方向より、 A_1/B_1 または A_2/B_2 を用いる)、ハイサイド ON の場合の方程式 (A_1/B_1) をそれぞれ、 $(h0-h2)$ 、 $(h2-h1)$ 、 $h1$ を計算ステップとして以下の計算を行う。なお、以上のように時間のパラメータである $h0$ 、 $h1$ 、 $h2$

は FPGA クロック数の整数値として与えられるため、A、B 行列の要素のパラメータの時間の単位は FPGA クロック値で予め正規化されている。

$$\begin{aligned} X_{3_1} &= X_2 + h1 (A_1 X_1 + B_1 u) \\ X_{3_2} &= X_{3_1} + (h2 - h1) (A_0 X_{3_1} + B_0 u) \\ X_3 &= X_{3_2} + (h0 - h2) (A_2 X_{3_2} + B_2 u) \end{aligned} \quad (5-10)$$

なお、スイッチング動作が発生しない場合は、 $h2=0$ 、 $h1=0$ とすることで、以上の式をすべての TaskTick において同様に実行する事ができ、モデルを簡略化している。
例えば X_2 の計算には、実際には以下の式(5-11)を実行している。

$$\begin{aligned} X_{2_1} &= X_1 + 0 (A_1 X_1 + B_1 u) \\ X_{2_2} &= X_{2_1} + (0 - 0) (A_0 X_{2_1} + B_0 u) \\ X_2 &= X_{2_2} + (h0 - 0) (A_2 X_{2_2} + B_2 u) \end{aligned} \quad (5-11)$$

図 5.20 に HiEventCount 値と、LoEventCount 値を計測する System Generator のブロック図と入出力のタイミングを示す。

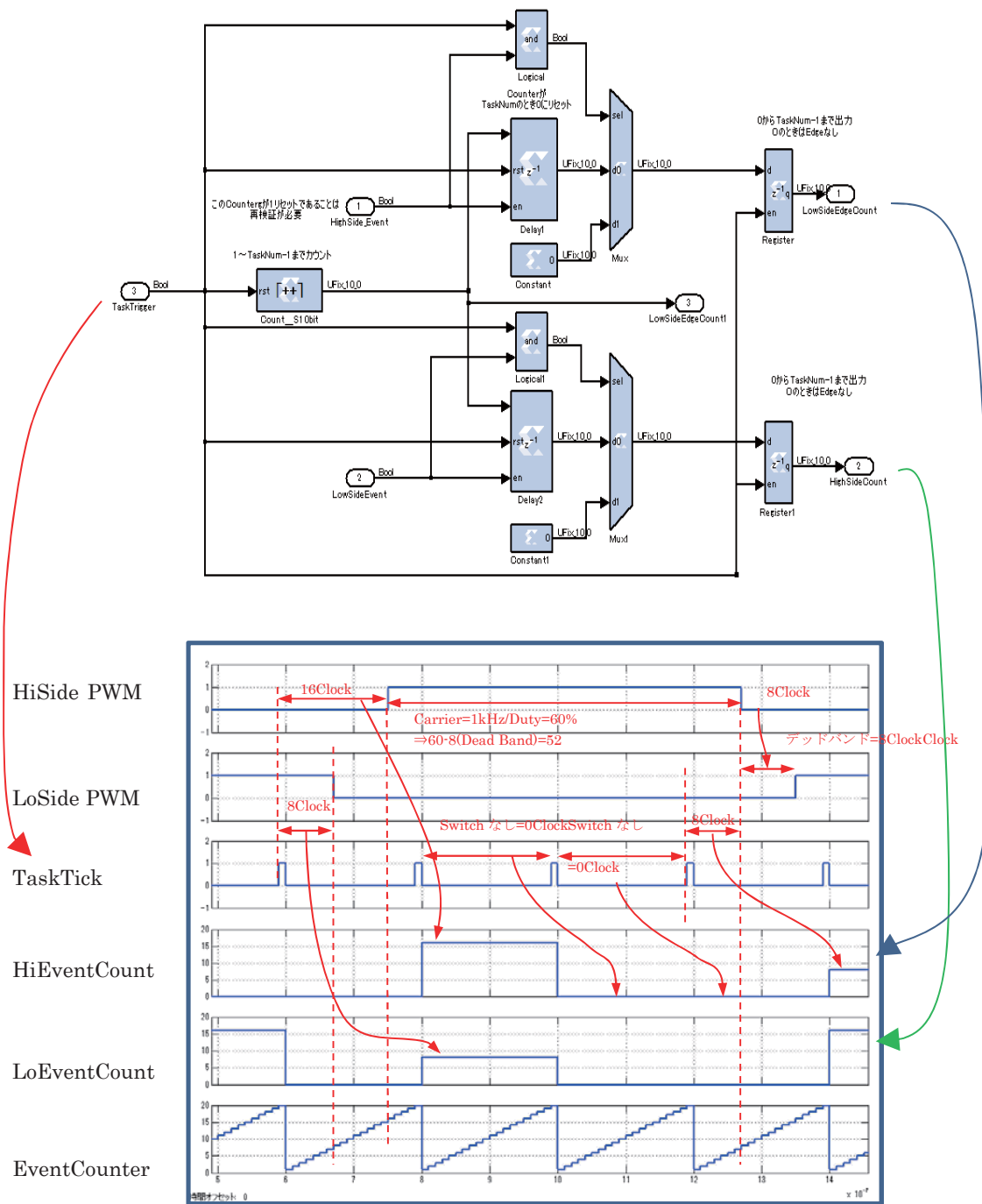


図 5.20 HiEventCount 値と LoEventCount 値を計測する FPGA モデル

なお、TaskTick とスイッチイベントが重なった時の処理はカウント値=0、つまりその時点では、スイッチイベントが発生していないとする処理を図 5.21 に示す。これは A、B、C、D 行列を切り替える処理に遅延が避けられないため、この遅延に合わせた結果である。

図 5.22 では TaskTick の直前にスイッチイベントが発生した場合の処理を示しており、

カウント値=19である。

FPGA プログラミングでは、このような微妙なタイミングに対する慎重な考察が重要となる。

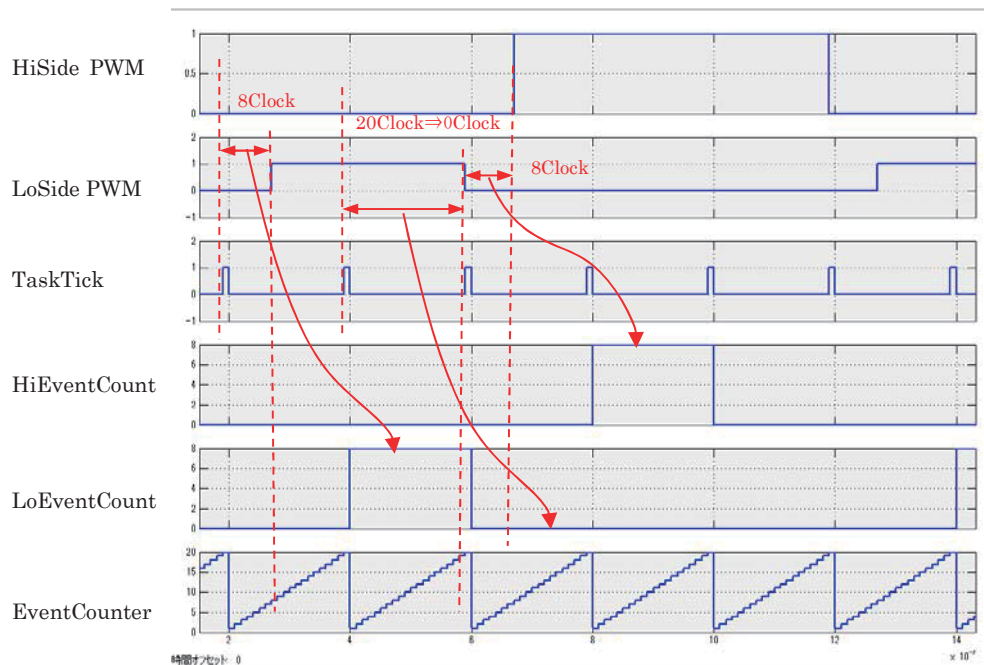


図 5.21 TaskTick とスイッチ動作が同時の場合の処理 (LoEventCount 値=0)

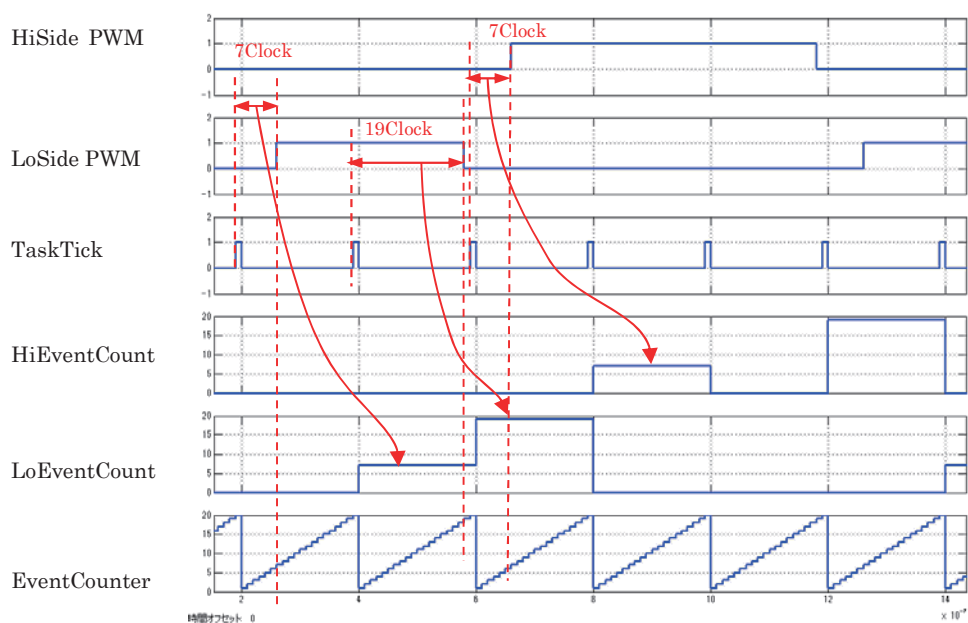


図 5.22 TaskTick の直前にスイッチ動作が起きた場合 (LoEventCount 値=19)

5.3.4. 実験結果（実回路との比較）

以下に実験結果を示す。図 5.23 に実験装置の外観を示す dSPACE 社のプロセッサボードである DS1005、FPGA ボードである DS5203 は左側にある筐体に収められており、計算結果と実回路の出力はホスト PC で観測するには速すぎるため、オシロスコープで直接観察する。なお FPGA 実装においては、PWM 信号も FPGA ボードで生成しているためファンクションジェネレータは使用していない。PWM 信号を FPGA で生成したことにより、ハイサイド、ローサイドの出力に設ける任意のデッドバンド時間設定が可能となったため、これを $1\mu\text{sec}$ としている（実際は 4.7 節で示したように、ボディダイオードが電流を流すため、ハイサイドの ON 時間が $1+1\mu\text{sec}$ 延長される事と等価である）。

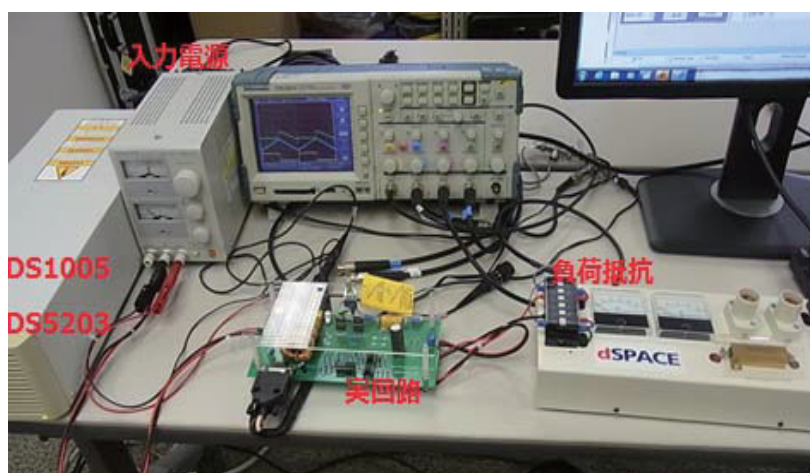


図 5.23 実験装置の外観

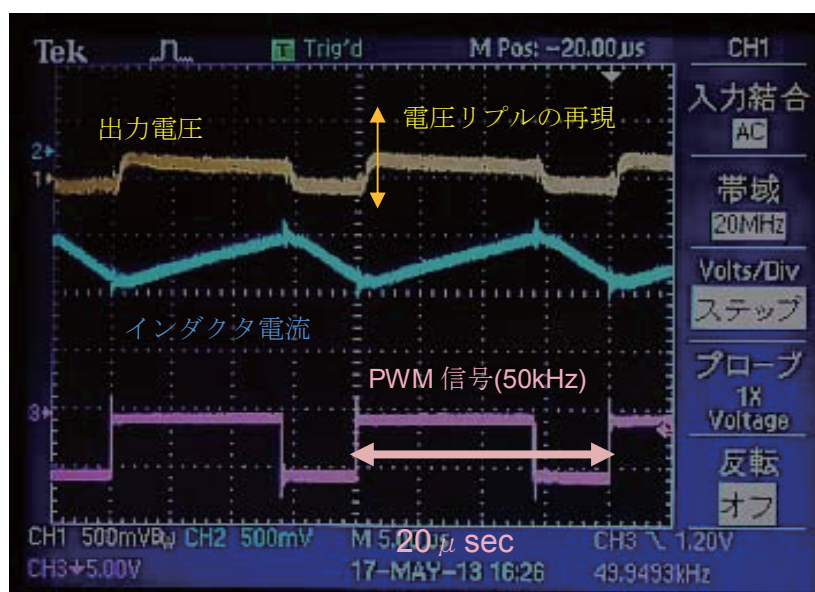


図 5.24 スイッチング周波数を 50kHz としたときのシミュレータ出力

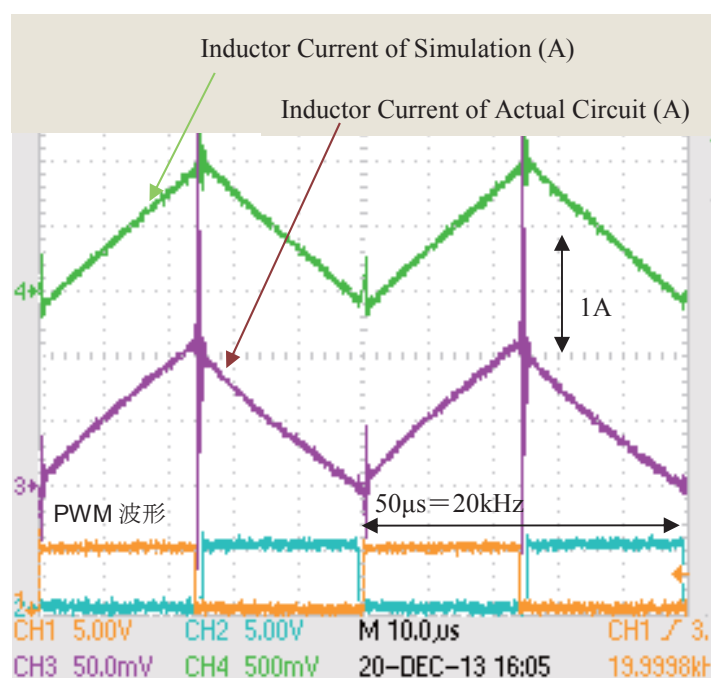


図 5.25 スイッチング周波数を 20kHz としたときの実回路とシミュレータ出力の比較
(Duty 比=0.5)

図 5.24、図 5.25 に実験結果を示す。プロセッサ実装にくらべ、波形は滑らかであり、計算遅延を観察する事はできず、良好な結果と言える。しかしながら、上記に示したように FPGA の実装は演算遅延を考慮した注意深い実装が必要であり、また System Generator モデルから、FPGA の回路構成を定義した RTL ファイルを生成するまでに 1 時間程度の時間を必要とするため、開発は非常に困難である。

なお、実回路の出力には、スイッチング動作に伴うかなり大きなスパイクノイズが乗っているが、シミュレータではそこまでは計算していない。シミュレータ出力にスパイクノイズが乗るのは、オシロスコープの内部回路の影響である。

図 5.26、図 5.27 に Duty 比を 0.3 及び 0.7 とした場合の結果を示す。プロセッサ実装の時と同じく $L=270\mu\text{H}$ とした値と一致するようであるが、図 5.27 のように Duty 比が大きくなり、流れる電流値の絶対値が大きくなると、実回路のインダクタ値は小さくなっているようである。定格 8A のインダクタであるが、非線形性の影響が現れたものと思われる。

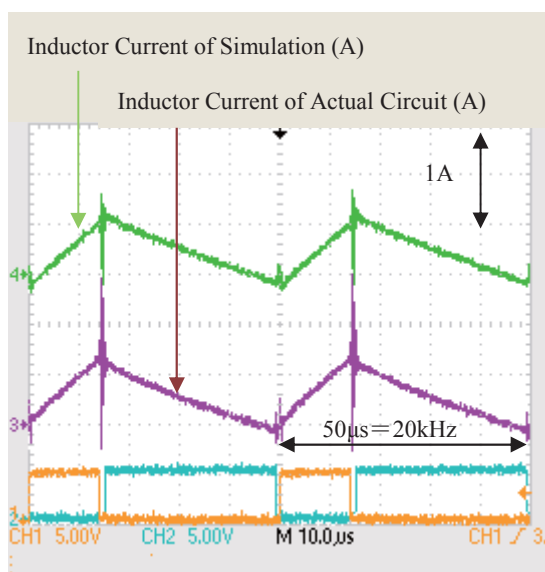


図 5.26 Duty 比=0.3 の場合

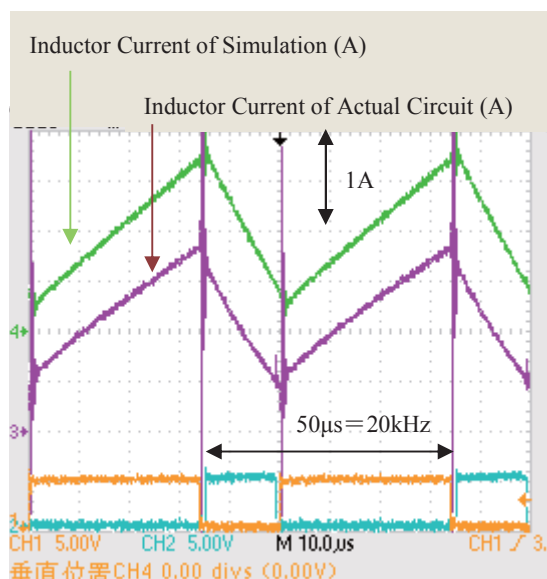


図 5.27 Duty 比=0.7 の場合

図 5.28 に実回路とシミュレータの出力電圧(負荷抵抗 R の両端電圧)の比較を示す。リップル電圧を確認するため、オシロスコープは AC カップリングで使用している。ESR の設計値は ESR メータにより 10kHz の周波数で測定した C2 コンデンサの単体実測値 (60mΩ) と ESR 模擬抵抗 100mΩ の合計値である 160mΩ であるが、実験結果では 300mΩ を用いてシミュレーションした結果と合致した。実回路は ESR 接続スイッチの接触抵抗で抵抗値が増加しているものと思われる。

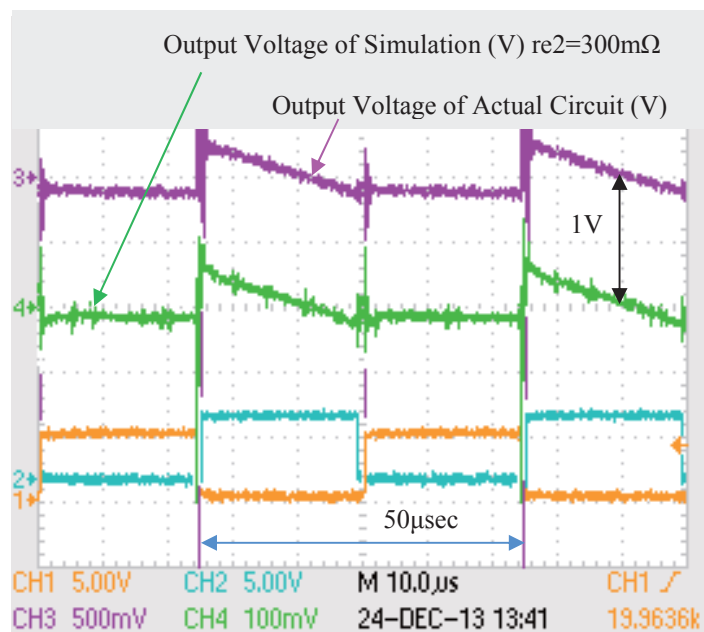


図 5.28 実回路とシミュレーションのリップル電圧の比較

FPGA 実装では非同期方式なため、PWM 駆動停止が可能であることより、図 5.29 に PWM 駆動停止状態(ハイサイド=OFF、ローサイド=OFF)から Duty 比=0.5 を出力した直後の出力電圧 (R2 電圧) のステップ応答を示す。R1=1Ω した場合のシミュレーション波形は立ち上がり直後においては一致しているが、実回路の波形は、その後に停滯するような波形となっている。これは、実回路は入力電源としてシリーズレギュレータ方式の安定化電源を使用しているため、安定化電源内の保護回路の影響と考えられる。

そこで、入力電圧源を鉛電池とした実験結果を図 5.30 に示す。この実験においては、R1=400 mΩ を使用した。電池内部抵抗および配線の接触抵抗が 400mΩ 程度であることが推定される。

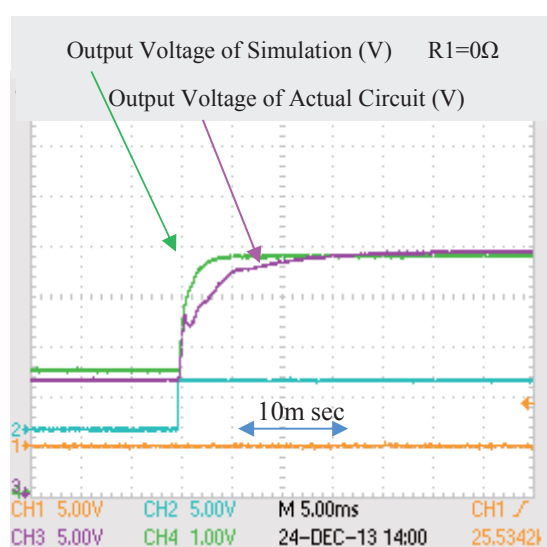


図 5.29 安定化電源を用いた
ステップ応答波形 (R1=1Ω)

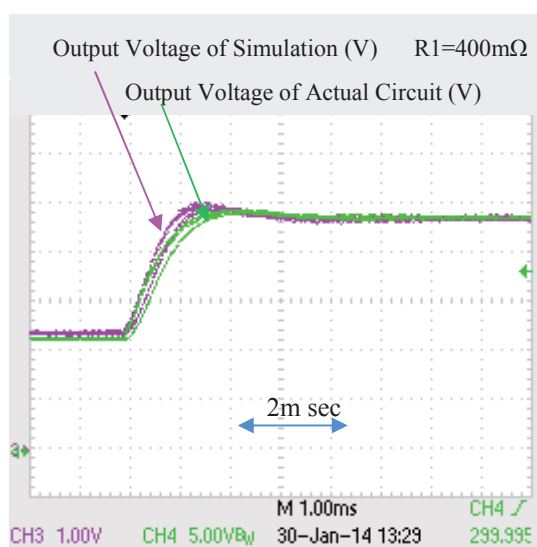


図 5.30 鉛電池を用いた
ステップ応答波形 (R1=400mΩ)

図 5.31 は Duty 比=0.5 から PWM 駆動停止としたときの結果である。この場合はアウトプットコンデンサ C2 と負荷抵抗 R の CR 回路による時定数 ($220\text{E-}6 \times 100 = 22\text{msec}$) で減衰する事になるが、PWM 駆動停止の場合は FET のボディダイオードにより入力電圧 (=12V) 以下には電圧が下がらない。

時定数の半分の時間 (11msec) では理論的に約 60%減衰するはずであるが、11msec 経過後の値を見ると 24V の 60%である 14.4V に近い値となっており、計算が妥当であることがわかる。

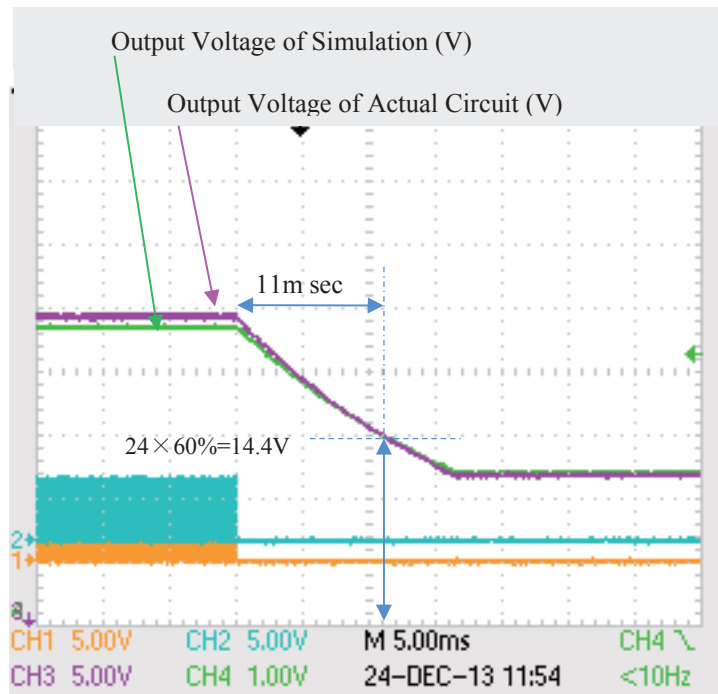


図 5.30 PWM-OFF 時の減衰応答波形 ($R1=0\Omega$)

6章 まとめ

6.1. 総括

本論文は、第 1 章において、フィードバック制御を行う制御ソフトウェアを開発するにあたり、リアルタイム・シミュレータの活用は不可欠である事を説明し、パワーエレクトロニクスの分野においても、リアルタイム・シミュレータの導入が期待されるものの、パワーエレクトロニクスの分野においては、制御周期が機械系に比べ短いことにより、制御ソフトウェアを使用するデジタル制御の普及が遅れているニーズサイド、および同様の理由によりリアルタイム・シミュレータの構築が困難であるシーズサイドの両方から、リアルタイム・シミュレータの導入が遅れていることを示した。

またリアルタイム・シミュレータ構築のための技術要素は、制御ソフトウェア開発の様々な分野で用いられており、それらは MBD(Model-Based Design/モデルベース開発)と呼ばれる開発手法として体系化されている。自動車や航空などの分野では MBD は制御ソフトウェア開発の効率化に大きく貢献しているが、パワーエレクトロニクスの分野においては、MBD の根幹をなすリアルタイム・シミュレータが上記の理由により開発が遅れており、このことが MBD の導入を妨げる要因となっている。

しかしながら、本研究が、計算アルゴリズムの改善により、並列プロセッサ等の高価なハードウェアを用いることなく、20kHz のスイッチング周期に対応するリアルタイム・シミュレータの構築が可能であることを明らかにした事より、パワーエレクトロニクスの分野においても、MBD の導入がなされていくものと期待される。

第 2 章においては、リアルタイム・シミュレータの実現手段としてデジタルコンピュータを使用した場合に必要不可欠である常微分方程式の数値解法について、代表的な手法を概観し、精度と数値安定性について検討を行った。

第 3 章においては、パワーエレクトロニクスにおいてシミュレータが模擬する対象、すなわちスイッチング回路を常微分方程式に変換する定式化手法について、代表的な、節点解析法、修正節点解析法、スパースタブロー法、カットセット解析による状態方程式法について検討を行い、同一のサンプル回路がどのような方程式に変換されるかを示し、各手法の得失を検討した。

第 4 章においては、スイッチング回路の計算において、スイッチ動作発生時刻の扱いが非常に重要であることを明らかにし、固定ステップ法、状態平均化法、部分的可変ステップ法、そして補間/外挿法を検討し、部分的可変ステップ法が最適であることを示した。

その結果、部分的可変ステップ法を採用するには、連続系の状態方程式を用いなければならない、状態方程式の数値解法については、“常識的なコンバータ回路”のシミュレーションを行うのであれば、1 ないし 2 次の Runge-Kutta 法において数値安定性を確保でき、従来からの回路計算には陰解法が必要との常識を覆し、陽解法による高速な計算が可能である

ことを明らかにした。

すなわち、これが本研究の結論である。

第 5 章においては、上記組み合わせを使用する事で、1GHz の PowerPC プロセッサを使用した場合において計算ステップ幅 $3.5 \mu \text{ sec}$ を達成できることを示した。この値は 20kHz のスイッチング周波数に対する最速制御周期である $50 \mu \text{ sec}$ に対して約 1/14 であり、リアルタイム・シミュレータとして満足できる値である。

また、FPGA で実装しうることを示し、その場合は、Duty 比の分解能を 0.1%(1/1000) とするならば、FPGA のクロック周波数の 1/1000 倍のスイッチング周波数 (100MHz の FPGA であれば 100kHz) に対応できることを示した。

6.2. 残された課題

部分的可変ステップ法を用い、状態方程式の計算を、陽的 Runge-Kutta 法で行う事により、精度の高い計算を高速で行う事が可能となり、スイッチ回路のリアルタイム・シミュレータに適用可能であることが本研究の結論であるが、20kHz のスイッチング周波数での“常識的なコンバータ回路”という但し書きが必要である。

この制約を外した場合、あるいは、ダイオードを使用したコンバータなどで電流不連続モードが発生した場合は、この結論は成立しない。

この点が、今後に解決すべき課題であるが、Semi-Implicit 法と呼ばれる、陰解法に対し近似式を適用し陽解法に変換する手法を適用する事が考えられる。この場合、計算式に近似式が含まれるため、計算精度が落ちることになる。すなわち、高周波の Stiff 極を低周波に移動させ、計算安定性を確保する事に起因する高周波応答の悪化が起きるが、本論文で強調しているようにリアルタイム・シミュレータは、制御ソフトウェアの開発ツールであり、回路解析用のツールではない。従って、制御ソフトウェアが対象としない、高周波の応答は無視してもよいと考える。この Semi-Implicit 法の適用を次のステップで行っていきたい。

また、現時点では状態方程式の構築では、手作業による標準木の選定を必要としているが、これを自動化するソフトウェアの開発も必要である。

6.3. 最期に

本研究の背景には、パワーエレクトロニクス分野への MBD 適用を図るために、その根幹である HIL、すなわちリアルタイム・シミュレータが必要とされている事がある。MBD においては、様々なツールベンダが、有り余る機能を持つ開発ツールを提供しており、これを利用すれば一夜にして制御ソフトウェアが完成する様な印象を与えているが、

しかし、モデルとは何か？

それは数学モデルである。すなわち本論文に記載された数々の数式こそがモデルであり、この理解なくしては、Model-Based ではないことを、最後に付け加えておきたい。

謝辞

本研究と論文の執筆にあたり、ご指導いただきました崇城大学エネルギーエレクトロニクス研究所の中原正俊教授、および FPGA プログラミング上の様々な難解な問題の解決と論文の校正に尽力いただいた dSPACE Japan(株)の皆様に深くお礼申し上げます。

参考文献

第 1 章

- [1] Stuart Bennett 著 古田勝久, 山北 昌毅 監訳 “制御工学の歴史” コロナ社 1998/08
- [2] The Math Work, “Fly Me to the Moon-Then and Now” MATLAB newsletter/summer 1999
http://www.mathworks.co.jp/company/newsletters/news_notes/sum99/lunar_module.html
- [3] スズキ(株) “Virtual Vehicle Test Drivers”, dSPACE Magazine3/2010
- [4] 日本テキサス・インスツルメンツ(株) “中級者のための電源設計セミナーテキスト”
2011 ver.2,” Sep. 2011.
- [5] W.Nagel, “SPICE2: A Computer Program to Simulate Semiconductor Circuits,” University of California, Berkeley Technical Report No. UCB/ERL M520 1975.
- [6] H.W. Dommel EMTP Theory Book.2nd edition.Vancouver,British Columbia:Microtran Power System Analysis Corporation,1992
- [7] J. R. Marti and J. Lin, “Numerical Oscillations in EMTP-Like Programs,” IEEE Trans. Power Syst., vol. 4, no. 2, pp. 739–747, May 1989
- [8] J. R. Marti and J. Lin, “Suppression of numerical oscillations in the EMTP,” IEEE Trans. Power Syst., vol. 4, no. 2, pp. 739–747, May 1989
- [9] M.Hamouda, F.Fnaiech,K.AL-Haddad.” DSP based real-time simulation of Dual-Bridge Matrix Converters,” Industrial Electronics, 2007,ISIE 2007,IEEE International Symposium.
- [10] T. Larsson, J. Hasler, P. Forsyth, and T. Maguire, “Voltage source converter modeled in RTDS—experiences and comparison with field results,” presented at the IPST, Lyon, France, Jun. 2007.
- [11] C. Dufour,V. Jalili-Marandi,, J.Bélanger, L.Snider “Power system simulation algorithms for

- parallel computer architectures,” Proceedings of the 2012 PES General Meeting San Diego, USA, July 22-26, 2012.
- [12] C.Dufour, T.O. Bachir, L. Grégoire, J.Bélanger “Solvers for Real-Time Simulation of Bipolar Thyristor-Based HVDC and 180-cell HVDC Modular Multilevel Converter for System,” Dynamics and Control of Switched Electronic Systems Advances in Industrial Control2012, pp 451-487
- [13] G.G.Parama,V.Dinavahi, ”Real-Time Digital Hardware Simulation of Power Electronics and Drives,” Power Delivery, IEEE Transactions on (Volume:22 , Issue: 2) 2007.
- [14] M. Matar, R. Iravani “FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients,” IEEE Trans. Power Deliver, vol. 25, No.2. Apr. 2010.
- [15] H.F. Blanchette, T. Ould-Bachir, J. P. David, “A State-Space Modeling Approach for the FPGA-Based Real-Time Simulation of High Switching Frequency Power Converters,” IEEE Trans. Industrial Electronics Vol. 59, No. 12, Dec 2012
- [16] “IEEE Control Systems Magazine June2005”
- [17] A. Gole, A. Keri, C. Kwankpa, E. Gunther, H. Dommel, I. Hassan,J. Marti, J. Martinez, K. Fehrle, L. Tang, M. McGranaghan, O. Nayak,P. Ribeiro, R. Iravani, and R. Lasseter, “Guidelines for modeling power electronics in electric power engineering applications,” IEEE Trans. Power Del., vol. 12, no. 1, pp. 505.514, Jan. 1997.
- [18] dSPACE Japan 監修 “モデルベース開発ーモデリング、プラント・モデル、コントロール・モデルー “ 日経 BP 社
- [19] 城所 仁 “モデルからプログラム・コードを作る” インターフェース 2000 年 10 月号 CQ 出版
- [20] M.Rappl,P.Braun,M.Beeck,C.Schroder,” Automotive Software Development: A Model Based Approach,” SAE 2002-01-0875
- [21] H.Kidokoro, M.Nakahara “Application of Model-Based Design in Development of Micro-Grid System,” Proceeding INTELEC2011 Amsterdam
- [22] H.Kidokoro, M.Nakahara “New Development Method of Control Software for Smart Energy Applications dealing with Multi Sample Rate Controller,” Proceeding ICRERA 2012 P99 November/2012 9 Nagasaki

第 2 章

- [23] U. M. アッシャー, L. R. ペツォルド 共著, 中村 眞理雄 監訳 嘉村友作・三ツ間 均 共訳 常微分方程式と微分方程式の数値解法 培風館, 2006
- [24] E. ハイラー, G. ヴァンナー 三井 斌友 監訳, 常微分方程式の数値解法 I, シュプリンガー・ジャパン, 2007

- [25] E. ハイラー, G. ヴァンナー 三井 斌友 監訳, 常微分方程式の数値解法 II, シュプリンガー・ジャパン, 2008.
- [26] 三井 斌友 著, “常微分方程式の数値解法”, 岩波書店, 2003.
- [27] 伊藤 正夫 藤野和建, “数値計算の常識”, 共立出版, 1985.
- [28] 三井 斌友 小藤 俊幸 斉藤 善弘, “微分方程式による計算科学入門”, 2004.
- [29] 篠原 能材, “数値解析の基礎”, 日新出版, 1978.
- [30] 牛田 明夫 田中 衛 著, “電子回路シミュレーション”, 現代非線形科学シリーズ 7: コロナ社, 2002.
- [31] 山本 昌志 「講義ノート：常微分方程式の数値計算法」 値計算法
http://www.yamamoto10.jp/yamamoto/lecture/2007/5E_comp_app/differential_equation/diff_eq.html/index.html

第3章

- [32] H.W.Dommel, “Digital computer solutions of electromagnetic transients in single and multiphase networks,” IEEE Trans. Power App.Syst., vol. PAS-88, no. 4, pp. 388.399, Apr. 1969.
- [33] N. Watson and J. Arrilaga, Power Systems Electromagnetic Transients Simulation, A. Johns and D. Warne, Eds. London, U.K.: IET, 2003, IET Power and Energy Series
- [34] namichin ブログ “SPICE による電子回路シミュレーション／SPICE の歴史”
<http://wave2.iobb.net/doc/spice/2.html>
- [35] 日本半導体歴史館 <http://www.shmj.or.jp/index.html>
 集積回路⇒「1973 年：SPICE の開発(米国 University of California, Berkeley)」
<http://www.shmj.or.jp/museum2010/exhibi726.h>
- [36] Stefan Jahn, “Qucs Technical Papers”
 Qucs-Project <http://qucs.sourceforge.net/docs/technical.pdf>
- [37] S.Y.R.Hui,S.Morrall, ”Generalised accosiateddiscrete circuit model for switching deviced,” IEE Proc.Sci.Meas,Technol.,Vol 141,No.1,January 1994.
- [38] 浅井 秀樹 渡邊 貴之 著, 電子回路シミュレーション技法, 科学技術出版 , 2002.
- [39] G.D>Hachtel,R.K.Brayton,F.G.Gustavson, “The Space Tableau Approach to Network Analysis and Design,”, IEEE Trans. Circuit Theory, vol. CT-18,No.1, pp.101-113, Jan.1971
- [40] 川上 博 “電気回路の状態方程式—系統的に求めるには—” 1998
 徳島大学 先端技術科学教育学部 システム創生工学専攻
 システム解析特 LectureNote
<http://cms.db.tokushima-u.ac.jp/DAV//lecture/125260/LectureNote/Circuit/CFormulation.pdf>
- [41] A. L. Shenkman, Transient Analysis of Electric Power Circuits Handbook, Springer, 2005/11/7.
- [42] L. O. C. P.-M. Lin, Computer-Aided Analysis of Electronic Circuits: Algorithms and

Computational Techniques, series in electrical & computer engineering: Prentice-Hall, 1975/June.

- [43] Venkatachari Rajagopalan, "Computer-Aided Analysis of Power Electronic Systems (Electrical and Computer Engineering)," MARCEL DEKKER, INC, New York, 1987.

第 4 章

- [44] 原田 耕介, 二宮 保, 顧 文建 共著 スイッチングコンバータの基礎 コロナ社, 1992.
- [45] 城所 仁 中原 正俊 "DC/DC コンバータのリアルタイム・シミュレーションを実現する陽解法に基づく高速計算手法" 電子情報通信学会論文誌 B Vol.J97-B, No.6, Jun. 2014.
- [46] 城所 仁 中原 正俊 "FPGA を用いたスイッチングコンバータのリアルタイム・シミュレーション" 電子情報通信学会 信学技報, vol. 113, no. 444, EE2013-55, pp. 37-42, 2014 年 2 月
- [47] Q. Do, B. DeKelper, L. Dessaint, and J. Soumagne, "An algorithm for accurate switching representation in fixed-step simulation of power electronics," IEEE Power Engineering Society Winter Meeting, vol. 1, pp. 762 –767, 2000.
- [48] K. Strunz, X. Lombard, O. Huet, J. Marti, L. Linares, and H. Dommel, "Real nodal-analysis-based solution techniques for simulations of Electromagnetic Transients in Power Electronic Systems," 13th PSCC, pp. 1047-1053, June28-July 2, 1999.
- [49] K. Lian and P. Lehn, "Real-time simulation of voltage source converters based on time average method," IEEE Trans. Power Syst., vol. 20, no. 1, pp.110 -118 2005
- [50] V. Dinavahi, M. Iravani, and R. Bonert, "Real time digital simulation of power electronic apparatus interfaced with digital controllers" IEEE transactions on Power Deliver, vol. 16, no. 4, October, 2001.
- [51] G. Rill. "A modified implicit EULER algorithm for solving vehicle dynamic equations ". Multibody System Dynamics, 15(2):1–24, 2006.

第 5 章

- [52] 日経エレクトロニクス "デジタル電源時代にこそ学びたい電源制御理論" テキスト NE アカデミー 2012
- [53] dSPACE gmbH "DS1103Features.pdf"
- [54] dSPACE gmbH "DS1103RTI.pdf"
- [55] dSPACE gmbH "RTIFPGAProgrammingGuide.pdf"

添付資料

- リスト 1 節点解析法でサンプル RLC 回路を計算する m-Script
 - リスト 2 修正節点解析法でサンプル RLC 回路を計算する m-Script
 - リスト 3 スパースタブロー法でサンプル RLC 回路を計算する m-Script
 - リスト 4 カットセット解析で実装したハーフブリッジ回路を計算する m-Script
 - リスト 5 Adams-Bashforth 法の安定領域をプロットする m-Script
 - リスト 6 Adams-Moulton 法の安定領域をプロットする m-Script
 - リスト 7 Backward Differentiation formula 法の安定領域をプロットする m-Script
- 参考図： 4 次の Runge-Kutta 法の安定領域を計算する Simulink モデル

```

%
%   リスト1
%
%-----
%   節点解析法でサンプルLRCL回路の方程式を構築する
%
%   +---R---n1---L---n2+      n1:Node-1
%   V       C      n2:Node-2
%   +-----+
%               城所  仁    2014/Jan
%-----

L=10E-3
C=1E-3

R=1;
V1=10;
Is=V1/R
h=1E-3;
Ts=h;
Cycle =100;
INTEGRAL='BE'

if strcmp(INTEGRAL,'TR')
    GL=h/L/2; %Trapezoidal
    GC=2*C/h;
else
    GL=h/L;
    GC=C/h;  %BackwordEuler
end

Y= [ (1/R+GL)  -GL;
    -GL   (GC+GL)]

Vout = zeros(2,Cycle);
Tout = zeros(1,Cycle);

v = [0 0]';
I = [Is 0]';

Icn = 0;
Iln = 0;
Vold = [0 0]';

Tout(1,1)=0;

for i=1:Cycle
    Vout(:,i)=v;
    Tout(1,i+1)=i*h;
    v=Y\I;

    if strcmp(INTEGRAL,'TR') % ----- Trapezoidal -----

        Icn = GC*(v(2) - Vold(2)) - Icn;
        Iln = GL*((v(1)-v(2)) + (Vold(1) - Vold(2)) ) +Iln;

        IeqC = - GC*v(2) - Icn;
        IeqL =  GL*(v(1)-v(2)) + Iln ;

    else % ----- BackwordEuler -----

        Icn = GC*(v(2) - Vold(2));
        Iln = GL*(v(1)-v(2)) +Iln;

        IeqC = - GC*v(2) ;
        IeqL =  Iln ;

    end % -----

    I(1) = Is -IeqL;
    I(2) = IeqL - IeqC ;
    Vold = v;
end

hold on
plot(Vout(1,:),'b' ); % Node-1 電圧
plot(Vout(2,:),'g' ); % Node-2 電圧
% 接点解析法では インダクタ電流は直接計算できない

```

```

%
%   リスト2
%
%-----
%   修正節点解析法でサンプルLRCL回路の方程式を構築する
%
%   +---R--n1--L--n2+      n1:Node-1
%   V      C      n2:Node-1
%   +-----+
%               城所  仁   2014/Jan
%-----

L=10E-3
C=1E-3

R=1;
G=1/R
V1=10;
Is=V1/R
h=1E-3;
Ts=h;
Cycle =100;
INTEGRAL='BE'

if strcmp(INTEGRAL,'TR')
    GL=h/L/2; %Trapezoidal
    GC=2*C/h;
else
    GL=h/L;
    GC=C/h; %BackwordEuler
end

%===== Modified Nodal Analysis  QUCS type =====

reqL =1/GL;

Y=[G -G  0  0  1;
   -G  G  0  1  0;
    0  0  GC -1  0;
    0  1 -1 -reqL 0;
    1  0  0  0  0];

I=[ 0 0 0 0  V1];

Vout2=zeros(5,Cycle);
Tout2=zeros(1,Cycle);
v  =[0 0 0 0 0]';
Vold=[0 0 0 0 0]';
Vln =0;
Icn =0;

for i=1:Cycle
    Vout2(:,i)=v;
    Tout2(1,i+1)=i*h;
    v=Y\I;

    if strcmp(INTEGRAL,'TR') % ---- Trapezoidal -----

        Icn = GC*(v(3) - Vold(3)) - Icn;
        Vln = reqL*(v(4) - Vold(4)) - Vln;

        IeqC = - GC*v(3) - Icn;
        IeqL =  GL*(v(1)-v(2)) + Iln ;
        VeqL = -reqL*v(4) - Vln;
    else % ----- BackwordEuler -----

        IeqC = - GC*v(3) ;
        VeqL = -reqL*v(4) ; % v(4)=IL
    end % -----

    I(3)=-IeqC;
    I(4)= VeqL;
    Vold= v;
end

hold on
plot(Vout2(2,:)+0.1,'g'); % Node-1 電圧
plot(Vout2(3,:)+0.1,'b'); % Node-2 電圧
plot(Vout2(4,:)+0.1,'k'); % インダクタ電流

%===== Modified Nodal Analysis  Asahara type =====

Y=[G -G  0  0  1;
   -G  G  0  1  0;
    0  0  GC (-1) 0;
    0  GL (-GL) (-1) 0;
   -1  0  0  0  0];

I=[ 0 0 0 0 -V1];

Vout3=zeros(5,Cycle);
Tout3=zeros(1,Cycle);
v  =[0 0 0 0 0]';
Vold=[0 0 0 0 0]';
Vln =0;
Icn =0;

for i=1:Cycle

```



```

Vout3(:,i)=v;
Tout3(1,i+1)=i*h;
v=Y\I;

if strcmp(INTEGRAL,'TR') % ----- Trapezoidal -----

    Icn = GC*(v(3) - Vold(3)) - Icn;

    IeqC = - GC*v(3) - Icn;
    IeqL = GL*(v(2)-v(3)) + v(4); % v(4)=iL(n)

else % ----- BackwordEuler -----

    IeqC = - GC*v(3);
    IeqL = v(4); % v(4); % v(4)=iL(n)
end % -----

I(3)= -IeqC;
I(4)= -IeqL;
Vold= v;
end

plot(Vout3(2,:)-0.1,'r','LineWidth',2); % Node-1電圧
plot(Vout3(3,:)-0.1,'m','LineWidth',2); % Node-2電圧
plot(Vout3(4,:)-0.1,'c','LineWidth',2); % インダクタ電流
hold off

```

```
%
%   リスト3
%
%-----
%   スパースタブロー法でサンプルLRCL回路の方程式を構築する
%
%   +---R---L---+
%   V       C
%   +-----+
%
%                               城所  仁   2014/Jan
%-----
```

```
L=10E-3
C=1E-3

R=1;
G=1/R
V1=10;
Is=V1/R
h=1E-3;
Ts=h;
Cycle =100;
INTEGRAL='BE'

if strcmp(INTEGRAL,'TR')
    GL=h/L/2; %Trapezoidal
    GC=2*C/h;
else
    GL=h/L;
    GC=C/h; %BackwordEuler
end
```

```
% ===== Sparse Tableau =====
reqL =1/GL;

D=[ 1  0  0  1;
    -1  1  0  0;
     0 -1  1  0]

Zr4 = zeros(4,4);
Zr3 = zeros(3,3);

R_consti = eye(4,4); % R/Gマトリックスの未使用エレメントは1
G_consti = eye(4,4); %

R_consti(1,1) = -R;    % Node-1の抵抗値
R_consti(2,2) = -reqL; % Node-2の抵抗値
G_consti(3,3) = -GC;   % Node-2のコンダクタンス値

R_consti(4,4) = 0;    % Node-4は電圧源
G_consti(4,4) = 1;    % Node-4は電圧源
```

```
Tbr=[zeros(3,3)  D      zeros(3,4);
     D'      zeros(4,4) -1*eye(4,4);
     zeros(4,3) R_consti  G_consti]
```

```
Cst=[0 0 0 ,0 0 0 0,0 0 0 V1]';
v = zeros(11,1);
Vout4=zeros(11,Cycle);
Tout4=zeros(1,Cycle);
```

```
Tou4t(1,1)=0;
```

```
for i=1:Cycle
    Vout4(:,i)=v;
    Tout4(1,i+1)=i*h;
    v=Tbr\Cst;
```

```
if strcmp(INTEGRAL,'TR') % ----- Trapezoidal -----

    IeqC = -GC*v(10) - v(6); % v(10)=VB3  v(6)=i3
    VeqL = -reqL*v(5) - v(9); % v(5)=i2  v(9)=VB2

else % ----- BackwordEuler -----

    IeqC = - GC*v(10); % v(10)=VB3
    VeqL = -reqL*v(5); % v(5)=i2
end % -----

Cst(9) = VeqL;
Cst(10)= IeqC;
Vold= v;
end
```

```
hold on
plot(Vout4(2,:)-0.1,'b', 'LineWidth',1);
plot(Vout4(5,:)-0.1,'m', 'LineWidth',1);
plot(Vout4(10,:)-0.1,'c', 'LineWidth',1);
hold off
```

```
%
%   リスト4
%
%-----
%
% カットセット解析で第5章のハーフブリッジ型DC/DCコンバータを計算する
%   城所  仁   2014/Jan
%-----
```

```
L  =180e-6
C1 =220e-6
C2 =220e-6
```

```
rs = 0.1
R  = 1e2
```

```
rl = 0.3
```

```
re1 = 0.06
re2 = 0.16
```

```
rs1 = 0.03 % Lo On
rsh = 1e6
```

```
rsh = 0.03 % Hi On
rs1 = 1e6
```

```
Cm=[C1 0; 0 C2];
```

```
Lm=[L];
```

```
Gm=[1/rs1 0 0 0 ; 0 1/rsh 0 0;...
    0 0 1/re1 0; 0 0 0 1/rl];
```

```
Rm=[rs 0 0; 0 re2 0; 0 0 R];
```

```
Vsm=[0 0 12]';
```

```
ntr_C =size(Cm,1);
ntr_G = size(Gm,1);
nln_L = size(Lm,1);
nln_R = size(Rm,1);
```

```
Qt=[ 1 -1 0 0; ...
     0 0 -1 0; ...
     1 0 -1 -1; ...
     0 0 -1 -1; ...
     1 -1 0 0; ...
     -1 0 0 0];
```

```
QCL =Qt(1:ntr_C, 1:nln_L);
QCR =Qt(1:ntr_C, nln_L+1:nln_L+nln_R);
QGL =Qt(ntr_C+1:ntr_C+ntr_G, 1:nln_L);
QGR =Qt(ntr_C+1:ntr_C+ntr_G, nln_L+1:nln_L+nln_R);
```

```
Bl=[ -1 0 -1 0 -1 1; ...
     1 0 0 0 1 0; ...
     0 1 1 1 0 0; ...
     0 0 1 1 0 0];
```

```
BLC =Bl(1:nln_L, 1:ntr_C);
BLG =Bl(1:nln_L, ntr_C+(1:ntr_G));
BRC =Bl(nln_L+(1:nln_R), 1:ntr_C);
BRG =Bl(nln_L+(1:nln_R), ntr_C+(1:ntr_G));
```

```
Ginv=inv(Gm);
Rinv=inv(Rm);
```

```
Req=Rm - BRG*Ginv*QGR
Geq=Gm - QGR*Rinv*BRG
```

```
ReqInv=inv(Req);
GeqInv=inv(Geq);
```

```
A11 = QCR*ReqInv*BRC
A12 =-QCL - QCR*ReqInv*BRG*Ginv*QGL
A22 = BLG*GeqInv*QGL
A21 =-BLC-BLG*GeqInv*QGR*Rinv*BRC
```

```
b11 = -eye(2,2);
b12 =-QCR*ReqInv*BRG*Ginv;
b13 = [0 0]'
b14 = QCR*ReqInv;
```

```
b21 = zeros(1,2)
b22 = BLG*GeqInv
b23 =-[0]
b24 =-BLG*GeqInv*QGR*Rinv;
```

```
B=[b11 b12 b13 b14; b21 b22 b23 b24]
```

```
M=[Cm zeros(2,1); zeros(1,2) Lm]
```

$A = M \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$

$M = \begin{bmatrix} Lm \text{ zeros}(1,2) & \text{zeros}(2,1) \text{ Cm} \end{bmatrix}$

$A = M \begin{bmatrix} A_{22} & A_{21} \\ A_{12} & A_{11} \end{bmatrix}$

```
% -----
%   リスト5
% -----
%   Adams-Bashforth法の安定領域をプロットする
%
%   城所 仁   2014/Jan
% -----
```

```
PlotNum=300;
th=[1:PlotNum]*2*pi/PlotNum
i=sqrt(-1);
```

```
zeta=exp(i*th);
zeta_i=1./zeta
dem= 1 - zeta_i;
```

```
p1 = (zeta -1);
p2 = (zeta -1)./(1+(dem.^1)/2);
p3 = (zeta -1)./(1+(dem.^1)/2 +(dem.^2)*5/12 );
p4 = (zeta -1)./(1+(dem.^1)/2 +(dem.^2)*5/12 + (dem.^3)*3/8 );
p5 = (zeta -1)./(1+(dem.^1)/2 +(dem.^2)*5/12 + (dem.^3)*3/8 + (dem.^4)*251/720);
```

hold on

```
plot(p1,'r');
plot(p2,'b');
plot(p3,'g');
plot(p4,'c');
plot(p5,'m');
```

```
% -----
%   リスト6
% -----
%   Adams-Moulton法の安定領域をプロットする
%
%   城所 仁   2014/Jan
% -----
```

```
PlotNum=300;
th=[1:PlotNum]*2*pi/PlotNum
i=sqrt(-1);
```

```
zeta=exp(i*th);
zeta_i=1./zeta
dem= 1 - zeta_i;
```

```
p1 = (1-zeta_i);
p2 = (1-zeta_i)./(1-(dem.^1)/2);
p3 = (1-zeta_i)./(1-(dem.^1)/2 -(dem.^2)/12 );
p4 = (1-zeta_i)./(1-(dem.^1)/2 -(dem.^2)/12 -(dem.^3)/24 );
p5 = (1-zeta_i)./(1-(dem.^1)/2 -(dem.^2)/12 -(dem.^3)/24 - (dem.^4)*19/720);
p6 = (1-zeta_i)./(1-(dem.^1)/2 -(dem.^2)/12 -(dem.^3)/24 - (dem.^4)*19/720 - (dem.^5)*3/160);
```

hold on

```
plot(p1,'r');
%plot(p2,'b');
plot(p3,'g');
plot(p4,'c');
plot(p5,'m');
plot(p6,'k');
```

```
% -----
%   リスト7
% -----
%   Backward Differentiation formula法の安定領域をプロットする
%
%   城所 仁   2014/Jan
% -----
```

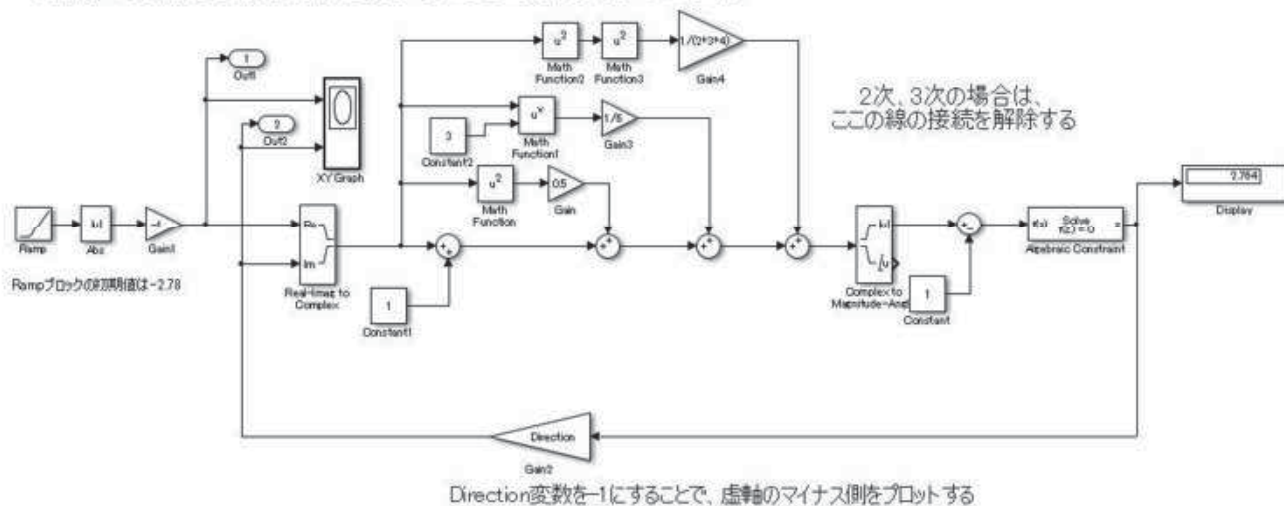
```
PlotNum=300;
th=[1:PlotNum]*2*pi/PlotNum
i=sqrt(-1);
```

```
p1 = (1-exp(-i*th));
p2 = (1-exp(-i*th)).^2;
p3 = (1-exp(-i*th)).^3;
p4 = (1-exp(-i*th)).^4;
p5 = (1-exp(-i*th)).^5;
p6 = (1-exp(-i*th)).^6;
```

hold on

```
plot(p1,'r');
plot(p1+p2/2,'b');
plot(p1+p2/2+p3/3,'g');
plot(p1+p2/2+p3/3+p4/4,'c');
plot(p1+p2/2+p3/3+p4/4+p5/5,'m');
plot(p1+p2/2+p3/3+p4/4+p5/5+p6/6,'k');
```

4次のRunge-Kutta法の安定領域を 実軸の左側から初めて折り返すところまでプロットするモデル



D4次のRunge-Kutta法の安定領域を 虚軸の0から初めて折り返すところまでプロットするモデル

虚軸の変数を変化させる

