

Rechnergestützte, transiente Simulation getakteter, leistungselektronischer Schaltungen

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg
zur
Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von
Jens Martin Göttle
aus Mühlacker

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich–Alexander–Universität Erlangen–Nürnberg

Tag der mündlichen Prüfung: 7. Februar 2020

Vorsitzender des Promotionsorgans: Prof. Dr.-Ing. habil. Andreas Fröba

Gutachter: Prof. Dr.-Ing Thomas Dürbaum
Prof. Dr.-Ing Matthias Luther

Kurzfassung

Die Leistungselektronik zählt zu den sogenannten Schlüsseltechnologien. Sie findet überall dort Verwendung, wo elektrische Energie umgeformt werden muss. Sei es die Stromversorgung eines Prozessors, die Motorelektronik einer Waschmaschine, das Vorschaltgerät einer LED-Lampe, unzählige Steuergeräte im Automobil oder die Anbindung regenerativer Energiequellen an das Versorgungsnetz. Die Anwendungsbereiche sind äußerst vielfältig. Fortschritte im Gebiet der Leistungselektronik wirken sich dadurch direkt auf die Einsatzgebiete aus und erlauben noch effizientere, kleinere und kostengünstigere Gesamtlösungen.

In den letzten Jahren prägt ein hoher Anteil an Modellbildung und Simulation zunehmend die Entwicklung moderner Leistungselektronik. Der große wirtschaftliche Druck auf die Hersteller sowie die zunehmende Komplexität der Schaltungen stellen den Schaltungsentwickler vor Herausforderungen, die er nur durch eine simulationsbasierte Vorgehensweise beherrschen kann. So helfen Simulationsergebnisse die Funktionsweise und Zusammenhänge besser und schneller zu verstehen sowie die Schaltungsparameter optimal an die Entwicklungsziele anzupassen.

Ausgehend von einem Überblick über aktuelle Lösungsansätze zur Schaltungssimulation, beschäftigt sich die vorliegende Arbeit mit der Entwicklung eines neuen Programms zur transienten Simulation getakteter, leistungselektronischer Schaltungen. Die rechnerinterne Beschreibung geschieht, ausgehend von einer SPICE-Netzliste, mit Hilfe stückweise linearer Netzwerke im Zustandsraum. Durch eine Eingangsgrößenmodellierung, d. h. dem Ersetzen der unabhängigen Strom- und Spannungsquellen durch Ersatznetzwerke, gelingt die Reduktion auf ein homogenes Differentialgleichungssystem. Virtuelle Widerstände helfen unterbestimmte Netzwerke, wie sie im Zusammenhang mit idealen Schaltern auftreten können, zu beheben. Ebenfalls eine mögliche Folge der Verwendung idealer Schaltelemente sind inkonsistente Anfangswerte der dynamischen Schaltungselemente, die das Programm selbstständig mit Hilfe der Gesetze zur Ladungs- bzw. Flusserhaltung löst. Die dabei auftretenden Spannungs- und Stromimpulse werden mittels ihres Gewichts quantitativ erfasst und ermöglichen die automatische Auffindung des korrekten Zustands aller als stückweise linear modellierten, nichtlinearen Schaltungselemente. Maßgeblich für die benötigte Simulationsdauer ist unter anderem die Bestimmung des Zeitpunkts intern gesteuerter Unstetigkeiten, bspw. dem Ein- bzw. Ausschaltzeitpunkt idealer Dioden. Dieser wird mit Methoden der Intervallarithmetik abgeschätzt und durch iteratives Auswerten durch immer kleinere Zeitintervalle zuverlässig eingegrenzt. Eine Blockdiagonalisierung der Systemmatrix mit anschließender Eigenwertverschiebung liefert die für die Abschätzung

nötige, analytische Lösung der Matrixexponentialfunktion. Die Kombination all dieser Methoden erlaubt eine hochgradige Ausnutzung des Potenzials der stückweise linearen Modellierung.

Das im Rahmen dieser Arbeit entwickelte Simulationswerkzeug ermöglicht es dem Schaltungsentwickler, einzig auf Basis einer SPICE-Netzliste, zuverlässige und hochgenaue Ergebnisse mit geringem Rechenaufwand zu erhalten.

Abstract

Power electronics is a key enabling technology which can be found wherever electric power has to be controlled and converted. Its extremely wide application area ranges from power supplies for CPUs to motor electronics of washing machines to LED lamp ballasts to countless car control units and to grid integration of regenerative energy sources. Progress in power electronics, thus, has large implications on many other areas and enables more efficient, smaller and less expensive solutions.

The design process of modern power electronics is characterised by a large amount of modeling and simulation. A simulation-based approach helps the circuit designer to master the conflict between the demand for shorter time to market and an ever increasing circuit complexity. Simulation results allow to understand a circuit's basic operation and let the designer optimize parameter values to reach specified design constraints.

Based on an overview of the state of the art of circuit simulation this thesis develops a new computer simulation software aimed at switched power electronic circuits. Within computer memory, state-space matrices, derived from piecewise linear networks, represent the circuit, which itself is defined by a SPICE-netlist. Replacing the networks' independent sources by equivalent circuits allows a formulation as an homogeneous differential equation system. Underdetermined networks, which can occur with ideal switches, are fixed using virtual resistors. Networks with ideal switches may as well exhibit inconsistent initial conditions of energy storing circuit elements. The simulator uses the laws of charge and flux conservation to solve this issue. Accompanying impulses in voltage or current are quantified by their weight and allow the automatic state detection of all piecewise linear elements. During the transient analysis some circuit components, e. g. diodes, cause breakpoints controlled by internal quantities. The detection of these events exhibits a large impact on simulation time. The corresponding time instants are bound iteratively by increasingly narrow intervals employing methods from interval analysis. Block diagonalisation of the system matrix in combination with eigenvalue shifting enables the analytical expressions required for the upper and lower bounds of the matrix exponential function. The combination of all these methods provides access to the full potential of piecewise linearly modeled circuits.

The proposed simulation tool allows circuit designers to get reliable and highly accurate simulation results in short time on the basis of a SPICE-netlist without the need for any further user input.

Inhaltsverzeichnis

Kurzfassung	iii
Abstract	iv
1. Einleitung	1
1.1. Hinführung zum Thema	2
1.2. Aufbau der Arbeit	5
2. Grundlagen der rechnergestützten transienten Schaltungsanalyse	7
2.1. Eingabe	7
2.2. Verarbeitung	8
2.2.1. Formulierung des Gleichungssystems	8
2.2.2. Dynamische Schaltungselemente	15
2.2.3. Nichtlineare Schaltungselemente	26
2.2.4. Anfangswerte	37
2.2.5. Lösung des Gleichungssystems	41
2.2.6. Unstetigkeiten	42
2.3. Ausgabe	45
2.4. Zusammenfassung und Diskussion	46
2.4.1. Eingabe	48
2.4.2. Formulieren des Gleichungssystems	48
2.4.3. Dynamische Schaltungselemente und numerische Integration	49
2.4.4. Nichtlineare Schaltungselemente	50
2.4.5. Anfangswerte	50
2.4.6. Lösung des Gleichungssystems	50
2.4.7. Unstetigkeiten	51
2.4.8. Ausgabe und Aufbereitung der Simulationsergebnisse	51
3. Rechnergestützte Formulierung der Zustandsraumdarstellung	53
3.1. Anwendung der Kirchhoff'schen Regeln	55
3.1.1. Aufstellen der Knotengleichungen	56
3.1.2. Aufstellen der Maschengleichungen	57
3.2. Anwendung der Strom-/Spannungsbeziehungen	61
3.2.1. Widerstände	62
3.2.2. Übertrager	63

3.2.3.	Induktivitäten und Kapazitäten	64
3.2.4.	Ideale Schalter und Dioden	65
3.2.5.	Gesteuerte Quellen	67
3.2.6.	Unabhängige Spannungs- und Stromquellen	69
3.3.	Formulierung der Zustandsraumdarstellung	73
3.3.1.	Aufstellen der Zustandsdifferentialgleichung	74
3.3.2.	Aufstellen der Ausgangsgleichung	76
4.	Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung	79
4.1.	Lösen des DGL mit Hilfe der Matrixexponentialfunktion	79
4.1.1.	Reihenapproximation und “Scaling and squaring“	80
4.1.2.	Block-Diagonalisierung	81
4.1.3.	Anwendung der Lösungsverfahren im Rahmen der Arbeit	86
4.2.	Bestimmung konsistenter Anfangswerte	86
4.2.1.	Gesetz der Ladungs- und Flusserhaltung	87
4.2.2.	Formulierung der Schalttransformationgleichung	90
4.2.3.	Einteilung in abhängige und unabhängige Zustandsgrößen	93
4.2.4.	Impulse höherer Ordnung	95
4.2.5.	Zusammenfassung	97
4.3.	Bestimmung konsistenter Zustände nichtlinearer Schaltungselemente	99
4.3.1.	Einleitendes Beispiel	99
4.3.2.	Formulieren der Umschaltbedingungen als Ungleichungssystem	101
4.3.3.	Auswerten des Gesamtungleichungssystems	102
4.4.	Bestimmung der Zeitpunkte eines Netzwerkwechsels	103
4.4.1.	Schaltungselemente mit zeitabhängigen Parametern	103
4.4.2.	Schaltungselemente mit intern gesteuerten Parametern	104
4.4.3.	Nullstellensuche mit hochauflösender Abtastung	105
4.4.4.	Nullstellensuche mit Polynominterpolation	106
4.4.5.	Nullstellensuche mit Eigenwertversch. und Polynominterp.	120
4.5.	Verarbeitung und Ausgabe von Simulationsergebnissen	129
4.5.1.	Bestimmung des Mittelwerts	130
4.5.2.	Bestimmung des Effektivwerts	132
4.6.	Zusammenfassung des Ablaufs	134
5.	Verifizierung der Algorithmen und Lösungsstrategien	137
5.1.	Demonstrations-Schaltungen	137
5.1.1.	Inkonsistente Anfangswerte und relativer Fehler	137
5.1.2.	Schaltungen mit versteckten Bedingungen	140
5.1.3.	Impulsbehaftete Schaltvorgänge und PWL-Elemente	143
5.1.4.	Zuverlässige Detektion von Unstetigkeiten	145
5.1.5.	Unterbestimmtes Netzwerk	147

5.2. Getaktete, leistungselektronische Schaltungen	148
5.2.1. Spannungsverdopplung mit Ladungspumpe	149
5.2.2. DC-DC Schaltwandler mit Spannungsregelung	152
6. Zusammenfassung und Ausblick	157
A. Ergänzungen zu Kapitel 3	161
A.1. Übersicht der Netzlistensyntax: Schaltungselemente und Modelldefinitionen	161
A.1.1. Widerstände, Kapazitäten und Induktivitäten	161
A.1.2. Gleichspannungs- und Gleichstromquellen	162
A.1.3. Sinus-Quellen	162
A.1.4. Puls-Quellen	163
A.1.5. Übertrager	164
A.1.6. Gesteuerte Quellen	164
A.1.7. Schalter	165
A.1.8. Ideale Dioden	166
A.2. Übersicht der Netzlistensyntax: Befehle und Direktiven	167
A.2.1. Transiente Analyse und eingeschwungener Zustand	167
A.2.2. Zeitverläufe	167
A.2.3. Kennwerte	168
A.2.4. Anfangswerte	168
A.3. Formulieren der Zustandsdifferentialgleichung	169
A.3.1. Methode der virtuellen Widerstände	170
A.3.2. Reduzierung der Anzahl an Zustandsvariablen	174
B. Ergänzungen zu Kapitel 4	179
B.1. Effektivwertberechnung mit reduziertem Zustandsraumsystem	179
Symbole und Abkürzungen	181
Tabellenverzeichnis	187
Abbildungsverzeichnis	189
Literatur	195

1. Einleitung

Der ökologische Fußabdruck¹ eines Einwohners der Bundesrepublik Deutschland betrug im Jahre 2014 durchschnittlich etwa 5,05 Hektar². Insgesamt stehen auf dem Planeten Erde eine biologisch produktive Fläche von 11,3 Milliarden Hektar³ für die ökologischen Fußabdrücke aller Menschen zur Verfügung. Damit gibt es insgesamt genügend Fläche für 2,4 Milliarden Erdenbürger mit dem Lebensstil eines durchschnittlichen Deutschen. Ein Vergleich mit der Gesamtbevölkerungszahl der Erde, 7,67 Milliarden⁴, offenbart den momentanen und schon seit einigen Jahrzehnten anhaltenden übermäßigen Ressourcenverbrauch⁵. Da jedoch auf der einen Seite derjenige Anteil der Erdbevölkerung mit erhöhtem Verbrauch seinen Lebensstil aus Eigenantrieb sehr wahrscheinlich nicht um das notwendige Maß reduzieren wird und auf der anderen Seite viele derjenigen Menschen, die heute noch einen ökologisch ausgeglichenen Lebensstil (freiwillig oder gezwungenermaßen) führen, diesen in Zukunft nicht unbedingt beibehalten, müssen Maßnahmen ergriffen werden, die zur Reduzierung des ökologischen Fußabdrucks beitragen. Andernfalls wird in der Zukunft die Ressourcen aufgebraucht oder zumindest stark vermindert sein, was wiederum negative Auswirkungen auf Mensch und Natur mit sich bringen wird.

Der ökologische Fußabdruck der deutschen Bevölkerung setzt sich zum größten Teil (64 %) aus Kohlenstoff zusammen². Dieser Anteil am Fußabdruck wird benötigt, um das vom Menschen durch Verbrennung fossiler Ressourcen freigesetzte Kohlenstoffdioxid zu binden. Folglich könnte der Ressourcenverbrauch durch einen effizienteren Umgang mit fossilen

¹ Der sogenannte ökologische Fußabdruck beruht auf einem von M. Wackernagel und W. Rees entwickelten Konzept [1], welches den Ressourcenverbrauch eines Menschen (Energie, Nahrung, Kleidung, Entsorgung von Abfällen und Binden von Kohlendioxid) in einen äquivalenten Flächenverbrauch an biologisch produktivem Boden, dem ökologischen Fußabdruck, umrechnet. Das Konzept ist nicht frei von Kritik, da zum einen Faktoren, die Einfluss auf die Nachhaltigkeit besitzen, nicht adäquat berücksichtigt werden können und zum anderen Fehlentwicklungen in Bezug zur Nachhaltigkeit fälschlicherweise positiv bewertet werden [2]. Allerdings kann der ökologische Fußabdruck als eine notwendige, jedoch nicht hinreichende Mindestbedingung für Nachhaltigkeit aufgefasst werden (https://www.footprintnetwork.org/content/documents/Footprint_FAQs.pdf, aufgerufen am 20.12.2018).

² <https://data.footprintnetwork.org>, aufgerufen am 20.12.2018

³ Die angegebene Fläche stimmt nicht mit der gesamten Planetenoberfläche überein, da die biologische Produktivität des Bodens regional stark schwankt (vgl. Wüste und Ackerfläche). Der Wert entspricht der zur Verfügung stehenden Fläche mit weltweit durchschnittlicher biologischer Produktivität.

⁴ <https://www.dsw.org/>, aufgerufen am 20.12.2018

⁵ Der sich nicht ausschließlich auf die deutsche Bevölkerung beschränkt (s. <https://data.footprintnetwork.org>, aufgerufen am 20.12.2018).

1. Einleitung

Brennstoffen einerseits und durch einen zunehmenden Anteil an erneuerbaren Energieformen andererseits deutlich reduziert werden. Für beide Lösungsansätze kann und muss die Leistungselektronik einen wichtigen Beitrag liefern. Moderne leistungselektronische Schaltungen sind, bei geeigneter Auslegung und optimalem Design, in der Lage, elektrische Energie mit sehr hohen Wirkungsgraden (i. A. über 90 %) umzuformen. Dadurch wird nur ein geringer Anteil der Primärenergie innerhalb der Schaltung in Wärme umgesetzt, wodurch der Hauptanteil ausgangsseitig als elektrische Energie zur Verfügung steht. Darüber hinaus wären viele Quellen an erneuerbaren Energien ohne die Leistungselektronik nicht zuverlässig, effizient und kostengünstig nutzbar [3].

1.1. Hinführung zum Thema

Noch vor wenigen Jahrzehnten bestand der typische Entwicklungsprozess einer leistungselektronischen Schaltung aus einer iterativen Abfolge von händischer Rechnung auf Basis stark abstrahierte Schaltungsmodelle, dem anschließenden Aufbau und Vermessung von Prototypen und einer Anpassung der Schaltungsparameter mittels der Methode des „Versuch und Irrtum“ [4–6]. Diese Abfolge wurde so oft wiederholt, bis das Ergebnis den Vorgaben entsprach. Im Gegensatz zur Leistungselektronik stößt dieses Vorgehen bei der Entwicklung integrierter Schaltkreise sehr schnell an seine Grenzen. Die hohe Anzahl an (nichtlinearen) Schaltungselementen sowie die enormen Kosten zur Herstellung eines Prototyps machen die händische Schaltungsanalyse unmöglich und mehrmalige Iterationen zu kostspielig. Die Veröffentlichung des Schaltungssimulators SPICE war deshalb für die Entwicklung hochintegrierter Schaltkreise unverzichtbar [7, 8]. Allerdings sorgen Fortschritte in der Leistungshalbleitertechnik, den Materialwissenschaften und der Mikroelektronik auch in der Leistungselektronik für eine zunehmende Bedeutung der Modellbildung und Simulation. Stetig steigende Schaltfrequenzen, eine immer höhere Leistungsdichte und die damit einhergehende Notwendigkeit, auch parasitäre Schaltungselemente in das Simulationsmodell miteinzubeziehen sind nur einige der Gründe dafür. Zudem können gewisse Phänomene innerhalb der Schaltung nicht oder nur aufwendig messtechnisch erfasst werden und werden erst durch Simulationen zugänglich [5]. Daneben spielen auch wirtschaftliche Faktoren wie niedrigere Entwicklungskosten, höhere Entwicklungsqualität, eine kürzere „time-to-market“ und das Erreichen des „right first time“-Prinzips eine wichtige Rolle [5]. So ist die Entwicklung moderner, hocheffizienter Leistungselektronik geprägt von einem hohen Anteil an Modellbildung und Simulation. Exemplarisch zeigt dies die Entwicklung einer Stromversorgung nach [5, 9]. Danach gliedert sich deren typischer Entwicklungsprozess in mehrere Grundschritte:

1. Spezifikationen, Entwicklungsziele und evtl. technische Einschränkungen werden definiert.
2. Eine oder mehrere Schaltungskandidaten werden vorgeschlagen. Die Wahl beruht stark auf Erfahrung, Wissen und evtl. ersten Abschätzungen mittels stark verein-

fachter Modelle.

3. Der Leistungsteil wird mittels eines einfachen Modells beschrieben und eine transiente Simulation durchgeführt. Der Entwickler lernt dabei dessen Funktionsweise und Zusammenhänge kennen.
4. Die Schaltung wird einer entwurfsorientierten Analyse unterzogen. Dies beinhaltet die Herleitung von Zusammenhängen anhand derer zulässige Wertebereiche für Schaltungsparameter bestimmt werden können, sodass Spezifikationen und Entwurfsziele erfüllt werden. Das stark vereinfachte Modell des Leistungsteils muss evtl. an geeigneter Stelle erweitert werden, sodass Effekte, die die Parameterwahl stark beeinflussen, mitberücksichtigt werden können.
5. Der Leistungsteil wird nach den Entwurfszielen und unter Einhaltung der Spezifikationen mittels einer sehr hohen Anzahl an transienten Simulationsdurchläufen mit jeweils leicht veränderten Schaltungsparametern optimiert.
6. Das optimale Schaltungsdesign wird als Prototyp realisiert. Anschließende Messungen erlauben die Verifikation und, falls erforderlich, eine Anpassung des Schaltungsmodells.
7. Analysen für den schlimmstmöglichen Fall („worst-case“) oder andere Zuverlässigkeits- und Produktionsausbeuteanalysen werden durchgeführt. Da Schaltungsparameter aufgrund von Serienstreuung, Umgebungseinflüssen (z. B. der Temperatur) und Alterungsprozessen einer Schwankung unterliegen, muss geprüft werden, ob die gestellten Anforderungen unter allen Bedingungen eingehalten werden.
8. Die vorherigen Schritte müssen ggf. mehrmals durchlaufen werden (Iteration) bis alle Anforderungen erfüllt sind.

Der Zeitbedarf zum Durchlaufen dieser Grundschrirte sowie die Qualität des Endergebnisses wird zum einen beeinflusst von der Erfahrung, dem Wissen und den Fähigkeiten des Schaltungsentwicklers und zum anderen von einer geeigneten Modellierung samt Simulationswerkzeug. Dabei hat sich gezeigt, dass eine hierarchische Struktur, ausgehend von abstrakten, stark vereinfachten Modellen hin zu sehr detaillierten Modellen, dem Schaltungsentwickler den meisten Nutzen bringt. Die unterschiedlichen Modellstufen sind [9,10]:

1. Sogenannte gemittelte Modelle [9] des Leistungsteils erlauben eine noch schnellere transiente Simulation, als es mit den stark vereinfachten Modellen der Fall ist. Die Herleitung und Anwendung gemittelter Modelle ist jedoch abhängig von der Schaltungstopologie und eignet sich nur sehr eingeschränkt für eine automatisierte, rechnergestützte Simulation [11].
2. Mehr oder weniger stark vereinfachte Modelle, die das Verhalten der Halbleiterbauelemente im einfachsten Fall als Kurzschluss (leitender Zustand) bzw. Leerlauf (sperrender Zustand) beschreiben, können verwendet werden, um die grundlegende Funktionsweise des Leistungsteils zu analysieren. Der Schaltvorgang, d. h. der Übergang zwischen leitendem und sperrendem Zustand, kann und soll mit diesen

1. Einleitung

Modellen nicht exakt beschrieben werden. Trotz der starken Abstraktion eignen sich diese Modelle um wichtige Größen wie Effektiv-, Mittel- und Spitzenwerte der Spannungen und Ströme zu ermitteln. Ebenso erlauben sie die Vorhersage der dynamischen Eigenschaften im Falle von Lastwechseln und Sprünge der Eingangsgrößen.

3. Detaillierte, teils sehr komplexe Modelle, die das Verhalten der Halbleiterbauelemente so exakt wie möglich wiedergeben können, werden verwendet, um beispielsweise Schaltvorgänge zu analysieren. Von Interesse sind dabei Details während des Schaltvorgangs, die Ursachen für Schaltverluste und die Spitzenwerte der Spannungs- und Strombelastung. Hersteller von Leistungshalbleitern stellen oftmals solche Modelle ihrer Produkte kostenlos zur Verfügung. Aufwendiger, jedoch umso nützlicher, können eigene Modelle sein, deren Parameter mittels Messungen angepasst und verifiziert werden [12]. Eine isolierte Betrachtung der detaillierten Modelle, ohne die Hinzunahme der parasitären Schaltungselemente aufgrund von Bauelementgehäuse und Leiterbahnen, wäre allerdings wenig aussagekräftig, sodass auch der Rest des Schaltungsmodells einen hohen Detailgrad besitzen muss. Die transiente Simulation eines solchen Modells ist aufgrund der großen Variation an Zeitkonstanten – von wenigen Nanosekunden im Bereich der parasitären Schaltungselemente bis hin zu mehreren Sekunden im Falle einer großen Ausgangskapazität bei geringer Ausgangslast – mit einem hohen Zeitaufwand verbunden. Detailbetrachtungen werden daher typischerweise nur für Teilschaltungen angewendet und sind auf die Dauer von wenigen Schaltzyklen begrenzt.

Jede dieser Modellkategorien besitzt ihren primären Einsatzzweck, denn eine ausschließliche Anwendung der detaillierten, komplexen Modelle würde vor allem bei unbekanntem Schaltungen den Entwickler überfordern und die Funktionsweise sowie die Zusammenhänge wären schwer zu durchschauen [13, 14]. Andererseits würde eine ausschließliche Anwendung der gemittelten Modelle keine direkte Aussage über wichtige Kenngrößen, wie etwa den Effektiv- und Spitzenwert, zulassen. Was innerhalb einer Schaltperiode oder gar während eines Schaltvorgangs geschieht, bleibt bei diesen einfachen Modellen ohne weiterführende Maßnahmen verborgen. Der Nutzen der vereinfachten, jedoch nicht gemittelten Modelle liegt darin begründet, dass sie sowohl eine sehr schnelle Auswertung mittels spezieller Schaltungssimulatoren, den „switched-circuit simulators“ [14, 15] erlauben, gleichzeitig jedoch auch viel Information über die Kurvenverläufe der Ströme und Spannungen liefern. Möglich wird dies durch die Beschreibung als stückweise lineare Schaltung im Zustandsraum [16, 17]. An diesem Punkt setzt das Thema der vorliegenden Arbeit an, deren Ziel es ist, bereits veröffentlichte Lösungsansätze im Zusammenhang mit der schnellen und effizienten transienten Simulation getakteter, leistungselektronischer Schaltungen zusammenzutragen, zu bewerten und an geeigneter Stelle um eigene Ansätze zu erweitern. Ergebnis soll ein Entwurf für ein Simulationsprogramm sein, welches das Potenzial der vereinfachten, stückweise linearen Modellierung bestmöglich für eine noch effizientere und schnellere Simulation ausnutzt. Schließlich ermöglicht ein solches Simulationswerkzeug dem Schaltungsentwickler in kürzerer Zeit, bessere und kostengünstigere

Schaltungen zu realisieren.

1.2. Aufbau der Arbeit

Im anschließenden, zweiten Kapitel wird der transiente Simulationsprozess in seine Teilbereiche gegliedert und diese einzeln und anhand von Beispielen erläutert. Beginnend bei der Eingabe, über die Verarbeitung bis hin zur Ausgabe werden die Lösungsansätze des populären Schaltungssimulationsprogramms SPICE vorgestellt. Parallel dazu werden alternative Methoden aufgezeigt, die sich besonders zur Simulation leistungselektronischer Schaltungen eignen. Das Kapitel schließt mit einer Zusammenfassung und kurzen Diskussion.

Aufgrund der besonderen Bedeutung der „rechnergestützten Formulierung der Zustandsraumdarstellung“ im Rahmen dieser Arbeit ist diesem Teilbereich des Simulationsprozesses ein eigenes, drittes Kapitel gewidmet. Die Formulierung beginnt bei der Definition der zu simulierenden Schaltung als SPICE-Netzliste und endet mit einem Differentialgleichungssystem erster Ordnung in Matrizenschreibweise, der sogenannten Zustandsraumdarstellung. Die vorgestellten Lösungsansätze werden anhand von Beispielen erklärt und sind in ihrem Umfang auf die Anwendung zur Simulation leistungselektronischer Schaltungen reduziert.

Die in Kapitel 4 vorgestellten Ansätze zur Lösung spezieller Teilaufgaben der rechnergestützten Simulation im Zustandsraum stellen den eigentlichen Kern des Simulationsprogramms dar. Sie sind mit der nötigen Tiefe aber dennoch weitestgehend anschaulich mit Hilfe von Beispielen erläutert. Den Abschluss des Kapitels bildet eine Zusammenfassung des abstrahierten Programmablaufs anhand eines Flussdiagramms.

Der Hauptteil schließt mit einer Verifikation der vorgestellten Algorithmen und Lösungsstrategien in Kapitel 5. Eine zentrale Rolle spielen dabei spezielle Demonstrationsschaltungen, die bestimmte Problemfälle provozieren, welche ein Simulationsprogramm für getaktete, leistungselektronische Schaltungen lösen können sollte.

Die an den Hauptteil anknüpfenden Anhänge A und B behandeln einige zusätzliche Informationen im Zusammenhang mit den zugehörigen Kapiteln.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

Der Aufbau eines Programms zur Schaltungssimulation kann nach dem Eingabe-Verarbeitung-Ausgabe-Prinzip in drei Teilbereiche gegliedert werden [4]. Im ersten Teil, der Eingabe, übermittelt der Anwender die Schaltung und das gewünschte Analyseverfahren an den Rechner. Es folgt als Zweites der Verarbeitungsschritt, welcher stark vom gewählten Analyseverfahren abhängt. Die Ausgabe als letzter Teilbereich beinhaltet typischerweise eine grafische Ausgabe der Ergebnisse. Allerdings sollte auch eine Ausgabe der Datenwerte möglich sein, damit der Anwender diese in geeigneter Form speichern und für seine Zwecke weiterverwenden kann. Im Folgenden werden sämtliche Teilbereiche anhand des populären Schaltungssimulators SPICE erläutert und mit alternativen Ansätzen, die sich insbesondere für die transiente Simulation leistungselektronischer Schaltungen eignen, verglichen.

2.1. Eingabe

Die Eingabe einer Schaltung kann üblicherweise auf zwei Arten geschehen: mittels einer grafischen Benutzeroberfläche oder in Textform. Eine grafische Benutzeroberfläche erleichtert vor allem unerfahrenen Anwendern die Bedienung des Programms erheblich. Um dem Rechner mitzuteilen, aus welchen Elementen die Schaltung besteht und wie diese verschaltet sind, ist keine Syntax notwendig. Im Gegenteil, sofern die grafische Oberfläche gut durchdacht ist, kann das Programm intuitiv bedient werden. Zudem gestaltet sich die Eingabe der Schaltung in Form eines Schaltplans deutlich übersichtlicher als in Textform. Dadurch entstehen weniger Fehler bei der Eingabe bzw. auftretende Fehler können leichter gefunden werden. Der erfahrene Anwender wünscht allerdings auch die Möglichkeit zur Eingabe der Topologie (d. h. die Verschaltung der Elemente) sowie der Parameter in Textform. Zudem bietet diese Form der Eingabe eine einfache Möglichkeit der automatisierten Programmsteuerung. In der Praxis hat sich deshalb, neben einer grafischen Oberfläche, die Eingabe als sogenannte SPICE-Netzliste bewährt [18]. Diese listet sämtliche für die Schaltungssimulation notwendige Informationen in lesbarer Textform auf. Näheres zum Aufbau und der Syntax einer SPICE-Netzliste befindet sich in Anhang A.

2.2. Verarbeitung

Da diese Arbeit ausschließlich die transiente Schaltungsanalyse betrachtet, umfasst der Verarbeitungsschritt die folgenden Unterschritte:

- Formulierung des Gleichungssystems,
- Berücksichtigung dynamischer Schaltungselemente (bspw. einer Induktivität),
- Berücksichtigung nichtlinearer Schaltungselemente (bspw. einer Diode),
- Bestimmung von Anfangswerten,
- Lösung des Gleichungssystems und
- Berücksichtigung von Unstetigkeiten (bspw. aufgrund einer Pulsquelle).

Diese werden in den nun folgenden Unterabschnitten beschrieben.

2.2.1. Formulierung des Gleichungssystems

Eine Schaltungsanalyse kann mittels mehrerer Ansätze geschehen, die sich in Umfang und Aufwand teils erheblich unterscheiden. Die händische Analyse, also die Lösung mittels Papier und Stift, versucht üblicherweise das Netzwerk in einzelne kleine Teilbereiche zu entkoppeln¹. Diese können dann mittels bereits bekannter Zusammenhänge wie z. B. dem Spannungs- und Stromteiler schnell und einfach gelöst werden [19]. Die Gesamtlösung ergibt sich dann durch geschickte Zusammenfassung dieser Teillösungen. Eine solche händische Vorgehensweise führt zwar in der Regel zügig auf das gewünschte Ergebnis, stößt jedoch bei umfangreicheren Netzwerken schnell an ihre Grenzen. Ein weiterer Nachteil, welcher im Hinblick auf eine rechnergestützte Umsetzung schwerer wiegt, ist ihr stark auf Erfahrung basierender Ablauf. Deutlich besser eignen sich diejenigen Verfahren der Netzwerkanalyse, die sowohl auf kleine als auch auf umfangreiche Netzwerke angewendet werden können und die eine streng systematische Vorgehensweise aufweisen. Aus diesem Grund haben sich in der rechnergestützten Schaltungssimulation einige wenige Verfahren bewährt und durchgesetzt [4]:

- Tableauverfahren, engl. „Sparse Tableau Analysis“ (STA),
- Modifiziertes Knotenpotentialverfahren, engl. „Modified Nodal Analysis“ (MNA),
- Zustandsraumverfahren, engl. „State Space Analysis“ (SSA).

All diese Formulierungsverfahren vereint die Eigenschaft, dass sie die Problemstellung als Matrix formulieren. Das heißt, Ströme und Spannungen im Netzwerk werden als Elemente eines unbekanntem Spaltenvektors \mathbf{x} aufgefasst und ein Gleichungssystem in der Form $\mathbf{F}\mathbf{x} = \mathbf{b}$ aufgestellt. Dieser systematische und für die rechnergestützte Auswertung äußerst

¹ Netzwerke mit dynamischen Schaltungselementen werden üblicherweise zuvor in den Laplacebereich transformiert.

vorteilhafte Ansatz geht auf [20] zurück. Je nach Schaltung und Verfahren ergeben sich:

1. große Gleichungssysteme mit dünnbesetzter Matrix \mathbf{F} oder kleine Gleichungssysteme mit dichtbesetzter Matrix \mathbf{F} ,
2. eine Matrix, die im transienten Verlauf konstante Dimension und eine gleichbleibenden Anzahl an Unbekannten aufweist oder eine Matrix, die im transienten Verlauf eine wechselnden Anzahl an Unbekannten mitsamt wechselnder Matrixdimension aufweist,
3. lineare oder nichtlineare Gleichungssysteme,
4. Gleichungssysteme, die ausschließlich aus algebraischen Gleichungen, Differentialgleichungen oder einer Mischung aus beiden bestehen.

Die drei Formulierungsverfahren (STA, MNA und SSA) und deren Unterschiede werden im Folgenden erläutert.

Tableauverfahren (STA)

Die Tableaudarstellung bezeichnet eine Darstellung des Gleichungssystems in nicht-reduzierter Form. Das bedeutet, dass alle Spannungen und Ströme des Netzwerks als Unbekannte im Gleichungssystem auftreten [4]. Je nach Variante des Tableauverfahrens handelt es sich bei den Spannungen um die Zweigspannungen, um die Knotenpotentiale² oder um beides. Zusätzlich können noch elektrische Ladungen, magnetische Flüsse³ und beliebige andere Größen in den Spaltenvektor \mathbf{x} der Unbekannten mit aufgenommen werden [22]. Schaltungssimulatoren, die die Tableaudarstellung verwenden sind u. a. ASTAP [23] und JANAP [24]. Im Beispiel nach (2.1) setzt sich der Spaltenvektor der Unbekannten aus dem Spaltenvektor der Zweigströme \mathbf{i} , der Zweigspannungen \mathbf{v} und der Knotenpotentiale \mathbf{v}_k zusammen.

$$\begin{array}{l} \text{Knotengleichungen} \\ \text{Maschengleichungen} \\ \text{Strom-Spannungsbeziehungen} \end{array} \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & -\mathbf{A}^\top \\ \mathbf{K}_i & \mathbf{K}_v & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{i} \\ \mathbf{v} \\ \mathbf{v}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{q} \end{bmatrix} \quad (2.1)$$

Die Untermatrix \mathbf{A} in (2.1) beinhaltet ausschließlich Einträge mit 1, -1 und 0. Eine Auswertung der ersten Zeilengruppe in (2.1), $\mathbf{A}\mathbf{i} = \mathbf{0}$, liefert die Knotengleichungen für alle Knoten des Netzwerks, ausgenommen des Bezugsknotens. Die zweite Zeile in (2.1), $\mathbf{v} - \mathbf{A}^\top \mathbf{v}_k = \mathbf{0}$, liefert Maschengleichungen unter Einbeziehung der Zweigspannungen und Knotenpotentia-

² Knotenpotentiale sind Spannungen zwischen einem beliebigen Netzwerkknoten und einem festen Bezugsknoten.

³ Die Verwendung von Ladungen und Flüssen kann Vorteile im Hinblick auf die Simulation von Schaltungen mit nichtlinearen Kapazitäten und Induktivitäten aufweisen [4, 21].

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

le⁴. Die letzte Zeile, $\mathbf{K}_i \mathbf{i} + \mathbf{K}_v \mathbf{v} = \mathbf{q}$, beinhaltet sämtliche Strom-Spannungsbeziehungen aller Zweige, wobei unabhängige Quellen zusätzlich einen Beitrag im Vektor \mathbf{q} liefern⁵. Die Darstellung in unreduzierter Form ermöglicht eine besonders schnelle und einfache Formulierung des Gleichungssystems mit Hilfe von sogenannten Ausfüllmustern [4]. Dabei wird direkt beim Einlesen der Netzliste für jedes Netzwerkelement die entsprechenden Einträge in die Matrix bzw. den Quellenvektor \mathbf{q} geschrieben. Jeder Elementtyp besitzt dabei seine eigenen charakteristischen Einträge, das sogenannte Ausfüllmuster. Abb. 2.1 zeigt dieses Vorgehen beispielhaft für einen Widerstand innerhalb eines größeren Netzwerks und anhand des relevanten Teilbereichs im Gleichungssystem.

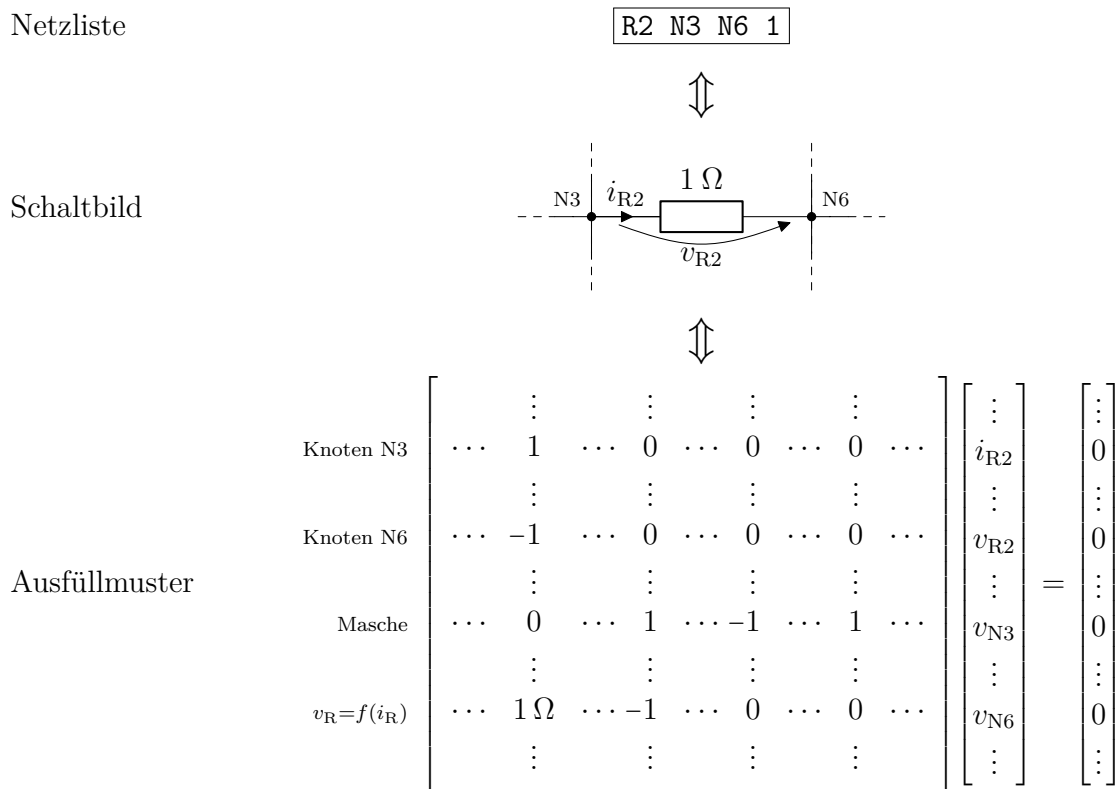


Abbildung 2.1.: Beispiel eines Netzlisteneintrags, Schaltbilds und Ausfüllmusters eines Widerstands in der Tableaudarstellung

⁴ Die Formulierung von Maschengleichungen erfolgt in der Regel unter Verwendung der Zweigspannungen. Werden jedoch zusätzlich auch die Knotenpotentiale (Spannungsdifferenz zwischen einem beliebigen Knoten und dem Bezugsknoten) miteinbezogen, so gestaltet sich das Aufstellen von linear unabhängigen Maschengleichungen einfacher. Allerdings muss das Netzwerk dann zwingend zusammenhängend sein.

⁵ Dynamische Netzwerkelemente, d. h. Induktivitäten und Kapazitäten, werden vor der Formulierung in äquivalente DC-Netzwerke, sogenannte „companion models“, umgewandelt (s. Kapitel 2.2.2). Auch nichtlineare Netzwerkelemente werden in linearisierte „companion models“ umgewandelt (s. Kapitel 2.2.3). Das resultierende äquivalente Netzwerk besteht ausschließlich aus Widerständen sowie unabhängigen und gesteuerten Quellen [4].

Die Matrix aus (2.1) weist, bedingt durch die nicht reduzierte Form, eine sehr geringe Dichte an nicht nullwertigen Einträgen auf. Eine solche Matrix wird deshalb auch als schwach- oder dünnbesetzt bezeichnet. Das Lösen solcher Gleichungssysteme kann erheblich beschleunigt werden, wenn auf triviale Rechenoperationen und unnötigen Speicherplatz im Zusammenhang mit den nullwertigen Einträgen verzichtet wird [4]. Für eine möglichst effiziente Schaltungssimulation ist bei Verwendung der Tableaudarstellung die Ausnutzung dieser Vorteile unverzichtbar. Sogenannte „sparse matrix solver“, also spezielle Lösungsverfahren für Gleichungssysteme mit dünnbesetzten Matrizen, bilden ein eigenes Teilgebiet der numerischen Mathematik. Für weiterführende Informationen sei auf [4] verwiesen.

Modifiziertes Knotenpotentialverfahren (MNA)

Das modifizierte Knotenpotentialverfahren kann als ein Versuch angesehen werden, das Gleichungssystem des Tableauverfahrens (STA) in seinem Umfang zu reduzieren ohne dabei die Vorteile der expliziten Formulierung aufgeben zu müssen [25]. Die prinzipielle Gestalt des zugehörigen Gleichungssystems zeigt (2.2).

$$\begin{array}{l} \text{Knotengleichungen} \\ \text{Strom-Spannungsbeziehungen} \end{array} \begin{bmatrix} \mathbf{Y} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{v}_k \\ \mathbf{i}_r \end{bmatrix} = \begin{bmatrix} \mathbf{j} \\ \mathbf{f} \end{bmatrix} \quad (2.2)$$

Im Gegensatz zum STA besteht der Spaltenvektor der Unbekannten nun nur noch aus den Knotenpotentialen \mathbf{v}_k und einigen ausgewählten Strömen \mathbf{i}_r . Genau diese Ströme stellen den Unterschied zum herkömmlichen, nicht-modifizierten Knotenpotentialverfahren dar [26, 27]. Durch Hinzunahme dieser Ströme ist es möglich auch Schaltungselemente, die keine Admittanzdarstellung besitzen, d. h. deren Strom nicht als Funktion der Knotenpotentiale bzw. deren zeitlichen Ableitungen darstellbar ist, zu berücksichtigen. Beispiele solcher Elemente sind Spannungsquellen und Induktivitäten. Zudem werden alle Steuerströme von stromgesteuerten Quellen wie auch alle als Ausgangsgröße gewünschten Ströme mit in den Spaltenvektor der Unbekannten \mathbf{i}_r aufgenommen [28].

Der rechtsseitige Spaltenvektor aus \mathbf{j} und \mathbf{f} beinhaltet die Werte sämtlicher Strom- und Spannungsquellen. Wie beim Tableauverfahren ist es auch im Fall des MNA möglich, das Gleichungssystem mittels Ausfüllmustern direkt beim Einlesen der Netzliste zu generieren [28]. Das Beispiel eines MNA-Ausfüllmusters für einen Widerstand⁵ zeigt Abb. 2.2.

Beim direkten Vergleich der beiden Matrizen der Tableaudarstellung und der modifizierten Knotenpotentialdarstellung fällt auf, dass die Matrix in (2.2) im Gegensatz zu jener aus (2.1) keine Nullmatrizen $\mathbf{0}$ als Untermatrizen aufweist. Dieser Umstand soll jedoch nicht darüber hinwegtäuschen, dass die Gesamtmatrix des MNA dünnbesetzt ist, im Allgemeinen allerdings nicht ganz so dünn wie die Matrix des Tableauverfahrens [4]. Welche der beiden Darstellungen zu einem schnelleren Ergebnis führt, hängt von vielen

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

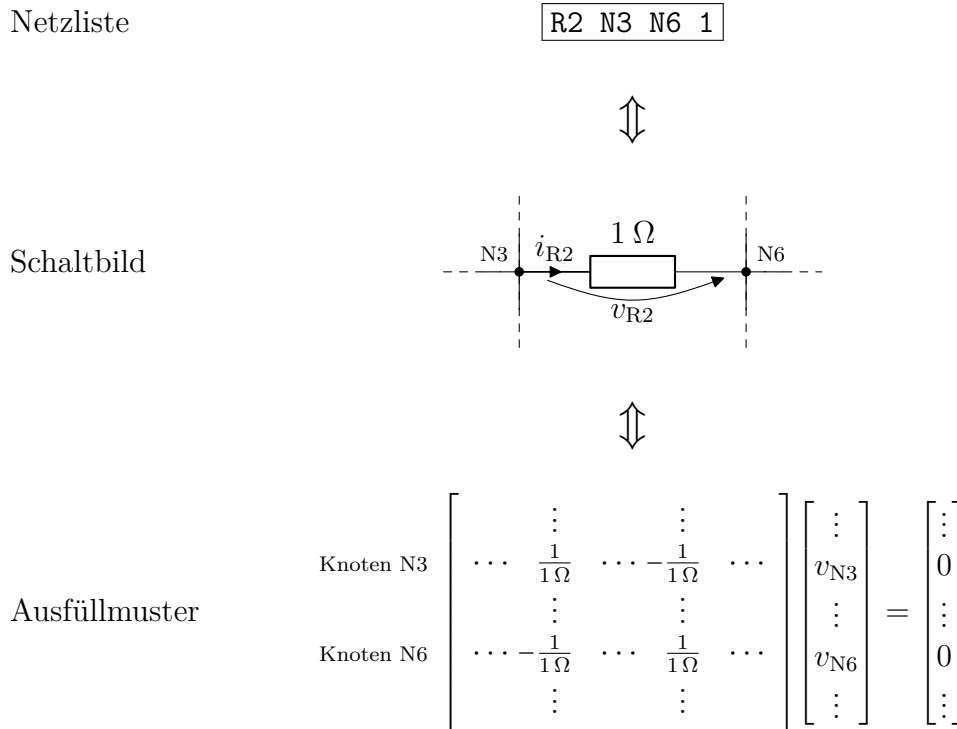


Abbildung 2.2.: Beispiel eines Netzlisteneintrags, Schaltbilds und Ausfüllmusters eines Widerstands in der modifizierten Knotenpotentialdarstellung

Umständen ab und ist nicht eindeutig zu beantworten [29]. Das populäre Schaltungssimulationsprogramm SPICE verwendet das MNA [29–34]. Neben den zahlreichen Simulationsprogrammen, die auf SPICE basieren, den sogenannten SPICE-Derivaten⁶, wird das MNA auch von folgenden Simulationsprogrammen in mehr oder weniger abgeänderter Form verwendet: SABER [36], PSIM [37], MULTISIM [38], PETS [39–41], CASPOC [42], GnuCap [43], SIMPLORER [44], KREAN [45], neuere EMTP-Derivate [46–48], QUCS [49], SWANN [50, 51] und viele weitere mehr [25, 52–54]. Das unmodifizierte, konventionelle Knotenpotentialverfahren wird in [55] sowie von sehr frühen SPICE-Versionen [30, 31] verwendet.

Neben dem MNA gibt es noch weitere Möglichkeiten, die Anzahl der Unbekannten im Gleichungssystem der STA zu reduzieren. Ein Beispiele dafür ist die Hybriddarstellung nach [4], die bspw. im Simulationsprogramm ECAP II [56] verwendet wird. Weiterhin besteht die Möglichkeit die MNA mit Hilfe der Zwei-Graphen-Methode aus [57] aufzustellen, was die Anzahl der Unbekannten gegenüber der Vorgehensweise in [28] in einigen Fällen nochmals verringern kann.

⁶ Ein Schaltungssimulator kann dann als SPICE-Derivat bezeichnet werden, wenn er eine globale Gesamtmatrix nach (2.2) aufweist, die sich im Laufe der transienten Simulation zwar in ihrem Inhalt, jedoch nicht in ihrer Form ändert [35].

Zustandsraumverfahren (SSA)

Das Zustandsraumverfahren (SSA) unterscheidet sich deutlich von den beiden vorhergehenden Formulierungsansätzen (MNA, STA). Das Gleichungssystem des SSAs besteht aus n expliziten, gewöhnlichen Differentialgleichungen erster Ordnung (engl. “Ordinary Differential Equation“ ODE):

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \quad (2.3)$$

Im Gleichungssystem der STA und der MNA sind hingegen algebraische und Differentialgleichungen vermischt, was auch als Differential-algebraisches Gleichungssystem, engl. “Differential Algebraic Equation“ (DAE), bezeichnet wird. Diese Unterscheidung spielt insofern eine wichtige Rolle, als dass die Lösung eines DAEs im Allgemeinen deutlich schwieriger ist, als die eines ODEs [58]. Näheres zur Lösung dieser Gleichungssysteme wird in einem der folgenden Abschnitte behandelt. Der Spaltenvektor der Unbekannten \mathbf{x} beinhaltet ausschließlich Spannungen über Kapazitäten bzw. Ströme durch Induktivitäten. Diese werden auch als Zustandsgrößen und ihre Momentanwerte als Zustand des Systems bezeichnet⁷. Sollen zusätzlich zu den im Vektor \mathbf{x} enthaltenen Netzwerkgrößen noch weitere Spannungen und Ströme des Netzwerks berechnet werden, so ist ein zusätzliches Gleichungssystem nötig:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (2.4)$$

Der Spaltenvektor dieser zusätzlichen Größen \mathbf{y} berechnet sich aus einer Linearkombination der Einträge in \mathbf{x} und \mathbf{u} .

Im Gegensatz zur STA und MNA ist das Formulieren der Matrizen \mathbf{A} , \mathbf{B} , \mathbf{C} und \mathbf{D} im Allgemeinen deutlich aufwendiger. Nach [60] lassen sich die Formulierungsmethoden in drei Kategorien einteilen:

1. rein topologische Methoden, die auf einen bestimmten Netzwerkbaum angewiesen sind, dafür aber eine explizite Formulierung erlauben [61, 62];
2. gemischt topologisch-algebraische Methoden, die auf einen bestimmten Netzwerkbaum angewiesen sind und explizite Zwischenergebnisse erlauben, jedoch keine explizite Formulierung der Zustandsraumdarstellung [63, 64];
3. rein algebraische Methoden, die auf keinen bestimmten Netzwerkbaum angewiesen sind dafür aber weder eine explizite Formulierung noch explizite Zwischenergebnisse erlauben⁸ [18, 70–73].

⁷ Eine für die Netzwerkanalyse geeignete Definition der Zustandsraumdarstellung stammt aus [59] und lautet: “Roughly speaking, the state of a system may be considered to be the **minimal** amount of information necessary at any time to characterize **completely** any possible future behavior of the system“.

⁸ Eine eigene Unterkategorie innerhalb der rein algebraische Methoden sind die Ansätze, die die Zustandsraumdarstellung mit Hilfe der MNA gewinnen [65–69].

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

Die Zustandsraumdarstellung von linear aktiven Netzwerken, d. h. Netzwerke, die lineare, gesteuerte Quellen enthalten, lässt sich allerdings im allgemeinen Fall nur algebraisch ermitteln [74]. Der Grund liegt darin, dass die Anzahl der Zustandsvariablen, also der Einträge im Vektor \mathbf{x} , nicht nur von der Verschaltung (Topologie) sondern auch von den Werten der Schaltungselemente abhängt. Alle Verfahren, die auf topologischen Methoden basieren (Kategorie 1 und 2), stellen daher gewisse Bedingungen an das Netzwerk, die für deren Anwendbarkeit erfüllt sein müssen.

Ein Beispiel für eine rein algebraische Methode stammt aus [18]. Den Ausgangspunkt bildet ein Gleichungssystem, das ähnlich der Tableaudarstellung alle Ströme und Zweigspannungen des Netzwerks als Unbekannte enthält.

$$\begin{array}{l} \text{Knotengleichungen} \\ \text{Maschengleichungen} \\ \text{Strom-Spannungsbeziehungen} \end{array} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \\ \mathbf{K}_i & \mathbf{K}_v \end{bmatrix} \begin{bmatrix} \mathbf{i} \\ \mathbf{v} \end{bmatrix} = \mathbf{0} \quad (2.5)$$

Die Formulierung des Gleichungssystems nach (2.5) kann, ähnlich wie in Abschnitt 2.2.1, mittels Ausfüllmustern geschehen. Im nachfolgenden Schritt werden die Ströme und Zweigspannungen und mit ihnen die Spalten innerhalb der Matrix nach folgendem Schema umsortiert:

- Spannungen und Ströme, die unter keine der folgenden Kategorien fallen (\mathbf{w});
- Spannungen und Ströme im Vektor \mathbf{y} ;
- Ströme durch Kapazitäten, Spannungen über Induktivitäten ($\dot{\mathbf{x}}$);
- Spannungen über Kapazitäten, Ströme durch Induktivitäten (\mathbf{x});
- Spannungen/Ströme von unabhängigen Spannungs-/Stromquellen (\mathbf{u}).

Das umsortierte Gleichungssystem wird anschließend mittels elementaren Zeilenumformungen [4] in reduzierte Zeilenstufenform gebracht:

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \dots & \dots \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & -\mathbf{C} & -\mathbf{D} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & -\mathbf{A} & -\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{y} \\ \dot{\mathbf{x}} \\ \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \mathbf{0}. \quad (2.6)$$

Die Gesamtmatrix in (2.6) enthält bereits die gesuchten Matrizen, \mathbf{A} , \mathbf{B} , \mathbf{C} und \mathbf{D} , der Zustandsraumdarstellung als Untermatrizen. Der hier dargestellte, vereinfachte Fall geht davon aus, dass sämtliche Spannungen über Kapazitäten und Ströme durch Induktivitäten auch gleichzeitig Zustandsgrößen sind. Ist dies nicht der Fall (bspw. bei zwei in Reihe geschalteten Induktivitäten), so müssen geeignete Zwischenschritte angewendet werden.

Auf deren Darstellung wird an dieser Stelle verzichtet und auf [18] verwiesen. Trotz der aufwendigeren Formulierung wird das SSA in mehreren Simulationsprogrammen angewendet, darunter PLECS [72], PECS [75, 76], SimPowerSystems [77], SIMPLIS [78], SWEAP [18] und einige weitere mehr [79–88].

2.2.2. Dynamische Schaltungselemente

Die beiden Hauptaufgaben eines Programms zur transienten Simulation leistungselektronischer Schaltungen sind der effiziente Umgang mit dynamischen⁹ sowie nichtlinearen Schaltungselementen. Während Nichtlinearitäten Gegenstand des nächsten Abschnitts sind, soll eine kurze Einführung in die rechnergestützte Lösung von Differentialgleichungen bzw. Differentialgleichungssystemen Thema dieses Unterabschnitts sein. Anhand von einfachen Beispielschaltungen sollen im Folgenden die Grundlagen der numerischen Lösung von Differentialgleichungen und deren Schwierigkeiten erläutert werden. Ausgangspunkt ist die Schaltung nach Abb. 2.3.

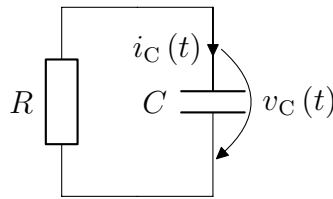


Abbildung 2.3.: Einfaches Beispielnetzwerk zur Demonstration numerischer Integration

Die zugehörige Differentialgleichung (2.7) besitzt mit λ einen charakteristischen Wert, der aufgrund seiner besonderen Bedeutung auch als Eigenwert bezeichnet wird.

$$\dot{v}_C = \lambda v_C \text{ mit } \lambda = -\frac{1}{RC} \quad (2.7)$$

Die Lösung dieser Differentialgleichung kann mit Hilfe der Exponentialfunktion direkt angegeben werden:

$$v_C(t) = v_{C,0} e^{\lambda t} \text{ mit } v_{C,0} = v_C(0). \quad (2.8)$$

Während die exakte Lösung nach (2.8) mit einem gegebenen Startwert $v_{C,0}$ die Berechnung der Kapazitätsspannung zu jedem beliebigen Zeitpunkt zulässt, versuchen numerische Lösungsverfahren die Lösungen zu diskreten Zeitpunkten anzunähern. Eine dieser numerischen Methoden ist das explizite Euler-Verfahren. Es entspricht einer Taylorreihenentwicklung um den Zeitpunkt $t = t_n$:

$$v_C(t) = v_C(t_n) + (t - t_n)\dot{v}_C(t_n) + \frac{1}{2!}(t - t_n)^2\ddot{v}_C(t_n) + \dots \quad (2.9)$$

⁹ Mit dynamischen Netzwerkelementen werden in der Literatur jene Elemente bezeichnet, deren Strom-Spannungsbeziehungen einen differentiellen Zusammenhang aufweisen. Diese Arbeit beschränkt sich dabei auf die beiden Elemente Induktivität und Kapazität.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

mit Abbruch nach dem zweiten Summanden. Mit einem fest vorgegebenen Zeitschritt der Länge h ergibt sich damit die Berechnungsvorschrift:

$$\tilde{v}_C(t_n + h) = \tilde{v}_C(t_n) + h\dot{v}_C(t_n) \quad (2.10)$$

Die Tilde über den Spannungen in (2.10) dient dazu, die auf numerischem Weg berechneten Werte von den exakten aus (2.8) unterscheiden zu können. Wird im Folgenden die Zeitabhängigkeit, wie bei zeitdiskreten Systemen üblich, über den Index ausgedrückt, so lässt sich das explizite Eulerverfahren als rekursive Folge notieren:

$$v_{C,n+1} = v_{C,n} + h\dot{v}_{C,n}. \quad (2.11)$$

Einsetzen von (2.7) in (2.11) führt schließlich auf:

$$v_{C,n+1} = v_{C,n} + h\lambda v_{C,n}. \quad (2.12)$$

Da das explizite Euler-Verfahren lediglich die ersten beiden Glieder der Taylorreihe verwendet, hängt die Abweichung der Näherung gegenüber der exakten Lösung ganz erheblich von der gewählten Schrittweite ab. Es kann nur in unmittelbarer Umgebung von einer guten Näherung ausgegangen werden. Ein Vergleich zwischen exakter und numerischer Lösung ist in Abb. 2.4 für unterschiedliche Schrittweiten dargestellt.

Ein Vergleich der vier gewählten Schrittweiten zeigt: Je kleiner die Schrittweite, desto besser die Näherungslösung. Mit $h \rightarrow 0$ geht die Näherungslösung in die exakte Lösung über. Allerdings wächst mit kleiner werdendem Zeitschritt zum einen die Anzahl der zu berechnenden Zeitpunkte für ein vorgegebenes Zeitintervall und zum anderen der Einfluss von Rundungsfehlern aufgrund der rechnergestützten Auswertung¹⁰. Die rechnergestützte Lösung bedarf demnach eines Kompromisses aus Rechenaufwand und Genauigkeit. Abb. 2.4 offenbart jedoch noch ein weiteres Problem des expliziten Euler-Verfahrens. Bei Überschreitung einer kritischen Schrittweite wächst die numerische Lösung über alle Maßen an, obwohl die zugrundeliegende, exakte Lösung gegen Null strebt. Ein solches Verhalten wird als numerisch instabil bezeichnet [4]. Die kritische Schrittweite, ab der beim expliziten Euler-Verfahren, für den Fall negativer Eigenwerte ($\lambda < 0$), instabiles Verhalten auftritt, liegt bei¹¹:

$$h_{\text{crit}} = \frac{2}{|\lambda|}. \quad (2.13)$$

Sie hängt also vom Eigenwert λ ab: Je größer der Betrag des Eigenwerts, desto kleiner muss die Schrittweite gewählt werden, um Stabilität zu garantieren. Allerdings weist mit wachsendem Betrag des Eigenwerts auch der Zeitverlauf der Kapazitätsspannung ein schnelleres Abklingen auf, sodass typischerweise nur ein kürzeres Zeitintervall simuliert

¹⁰ Jeder Rechenschritt ist aufgrund der rechnerinternen Zahlendarstellung mit Rundungsfehlern behaftet [4].

¹¹ Für eine allgemeine Einführung zur Stabilität numerischer Integrationsverfahren und zur Herleitung von (2.13) sei auf [4] verwiesen.

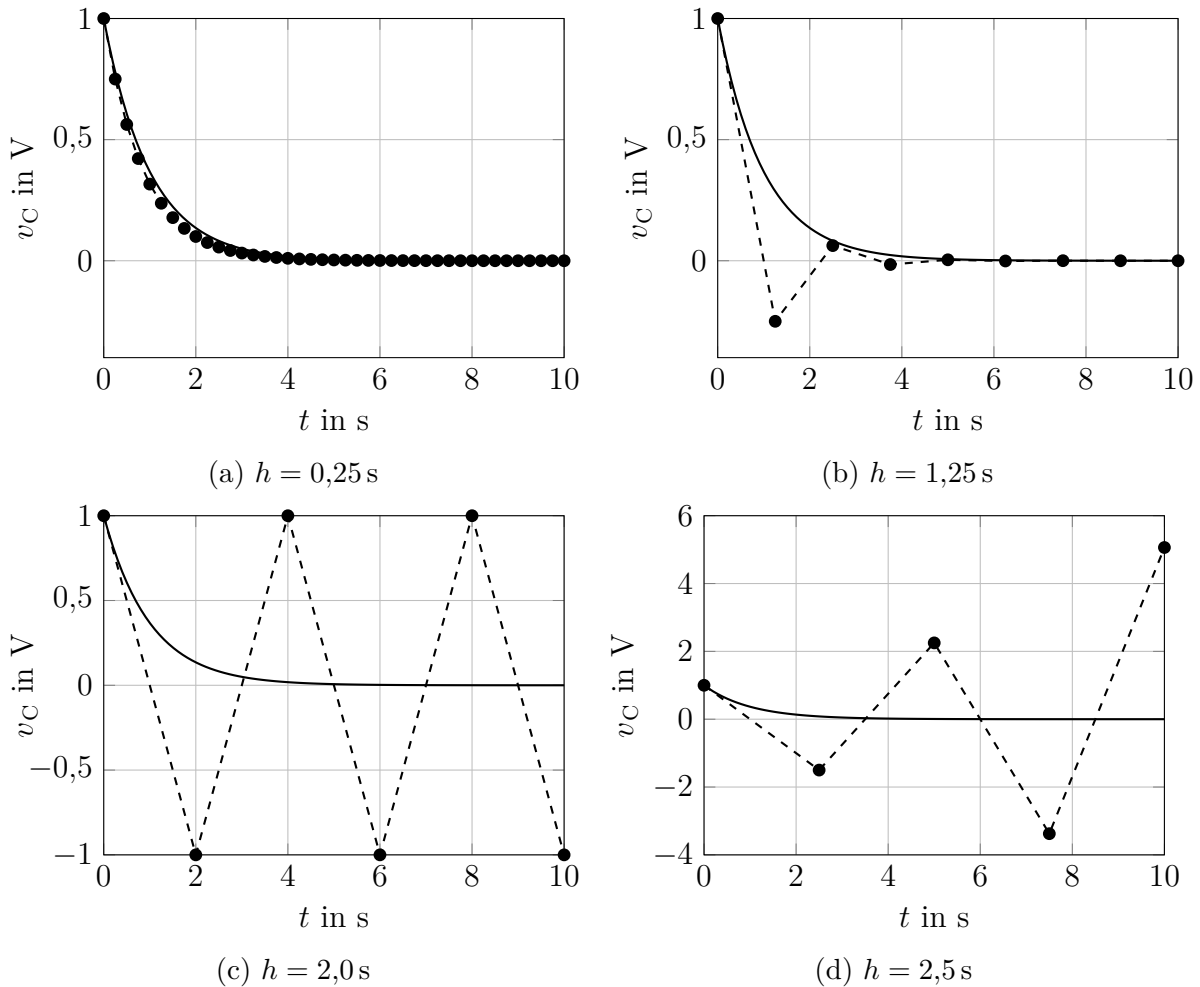


Abbildung 2.4.: Exakte Lösung (durchgezogen) und Lösung des expliziten Euler-Verfahrens (gestrichelt) für unterschiedliche Schrittweiten h mit $\lambda = -1\text{ s}^{-1}$ und $v_{C,0} = 1\text{ V}$, nach [89]

werden muss, bis die Lösung als näherungsweise stationär gilt. Der Rechenaufwand bleibt demnach im Fall der Schaltung nach Abb. 2.3 auch für große $|\lambda|$ in einem akzeptablen Rahmen¹². Die Problematik zeigt sich erst mit einer Schaltung, bei der sich die Lösung aus mehr als einer Exponentialfunktion zusammensetzt, wie etwa diejenige aus Abb. 2.5.

Mit der Annahme einer exponentiellen Stromquelle

$$i_g(t) = \hat{i}_g (1 - e^{\lambda_g t}) \quad (2.14)$$

¹² Vorausgesetzt, der Anwender oder das Programm passt das Simulationsintervall an den jeweiligen Eigenwert an.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

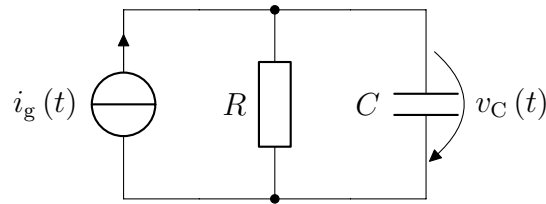


Abbildung 2.5.: Beispielnetzwerk mit $i_g(t) = \hat{i}_g(1 - e^{\lambda_g t})$, wobei $|\lambda_g|$ deutlich kleiner als $|\lambda|$ gewählt wird (vereinfacht aus [4] entnommen)

und dem Eigenwert λ aus (2.7) lautet die Lösung der Differentialgleichung¹³:

$$v_C(t) = c_1 + c_2 e^{\lambda_g t} + c_3 e^{\lambda t} \quad (2.15)$$

Für das folgende Beispiel soll nun die Exponentialfunktion des RC-Glieds mit einem Eigenwert von $\lambda = -1 \text{ s}^{-1}$ deutlich schneller abklingen als die der Quelle mit $\lambda_g = -0,1 \text{ s}^{-1}$. Der Zeitschritt soll mit $h = 0,25 \text{ s}$ zunächst klein gewählt werden. Um Rechenaufwand zu sparen, wird zum Zeitpunkt $t = 5 \text{ s}$ auf einen größeren Zeitschritt mit $h_g = 2,5 \text{ s}$ umgeschaltet, da dann die schnell abklingende Komponente des RC-Glieds nahezu vollständig abgeklungen ist. Das Ergebnis zeigt Abb. 2.6.

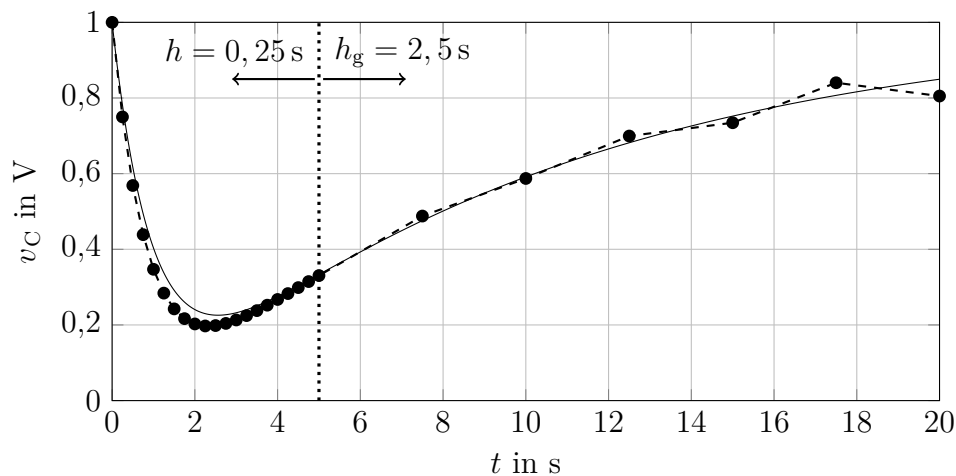


Abbildung 2.6.: Exakte (durchgezogene) und Näherungslösung (gestrichelt) des Netzwerks aus Abb. 2.5 mit $\lambda = -1 \text{ s}^{-1}$, $\lambda_g = -0,1 \text{ s}^{-1}$, $v_{C,0} = 1 \text{ V}$ und $R\hat{i}_g = 1 \text{ V}$, bei $t = 5 \text{ s}$ springt der Zeitschritt der Näherungslösung von $h = 0,25 \text{ s}$ auf $h_g = 2,5 \text{ s}$

Obwohl sich die exakte Lösung (durchgezogene Linie) zum Zeitpunkt $t = 5 \text{ s}$ tatsächlich deutlich langsamer ändert als zu Beginn, verursacht die Umschaltung auf den großen

¹³ Die konstanten Vorfaktoren in (2.15) sind: $c_1 = R\hat{i}_g$, $c_2 = R\hat{i}_g \frac{\lambda}{\lambda_g - \lambda}$ und $c_3 = R\hat{i}_g \frac{\lambda_g}{\lambda - \lambda_g} + v_{C,0}$

Zeitschritt ein Anwachsen der numerischen Lösung (Marker, bzw. gestrichelte Linie) über alle Maßen. Dies hat zur Folge, dass sich die Wahl des Zeitschritts beim expliziten Euler-Verfahren zu jeder Zeit am betragsmäßig größten Eigenwert orientieren muss, auch wenn dessen zugehörige Zeitfunktion längst abgeklungen ist¹⁴. Für Netzwerke mit mehreren, sehr weit auseinanderliegenden Eigenwerten, bedeutet dies eine inakzeptable Rechendauer, wenn die transiente Simulation so lange durchgeführt werden soll, bis die Gesamtlösung den näherungsweise stationären Zustand erreicht hat. Entsprechende Netzwerke sind in der Leistungselektronik keine Seltenheit [75]. Eine häufige Ursache sind parasitäre Schaltungskomponenten¹⁵, die typischerweise sehr weit auseinanderliegende Eigenwerte hervorrufen. Im Allgemeinen werden solche Systeme auch als steife Systeme bezeichnet. Numerische Integrationsverfahren werden, neben anderen Kriterien, dann als geeignet für steife Systeme erachtet („stiffly stable“), wenn sie unabhängig des negativen Realteils eines Eigenwerts Stabilität für beliebig große Schrittweiten garantieren können [4]. Nur dann ist es möglich, die Schrittweite, wie in Abb. 2.6, zu vergrößern, ohne dass ein instabiles Verhalten auftritt.

Eine Möglichkeit den kritischen Zeitschritt h_{crit} zu vergrößern besteht darin, zusätzliche Terme der Taylorreihe (2.9) zu berücksichtigen. In der Theorie führt dieser Ansatz zwar zum Ziel und bei genügend großer Anzahl an Termen ist jeder beliebige Zeitschritt denkbar¹⁶. In der Praxis neigt die rechnergestützte Auswertung der Taylorreihe zu Rundungsfehlern, sodass die Reihe zwar konvergiert, das Gesamtergebnis allerdings stark fehlerbehaftet ist [91]. Auch die sehr verbreiteten, expliziten Runge-Kutta-Verfahren besitzen ein vom betragsmäßig größten Eigenwert beschränktes Stabilitätsgebiet. In der Literatur finden sich deshalb andere, geeignetere Lösungsansätze:

- Implizite Integrationsverfahren
- Matrixexponentialfunktion
- „waveform relaxation“-Verfahren bzw. „multirate“-Verfahren [7, 92]
- numerische Laplace-Rücktransformation nach [50, 57, 93]
- Zustandsübergangsmatrix für MNA/STA nach [94]

Aufgrund der weiten Verbreitung und der universellen Anwendbarkeit sollen im Folgenden die impliziten Integrationsverfahren vorgestellt werden. Die Matrixexponentialfunktion ist ein weiterer interessanter Lösungsansatz, der vor allem im Bereich der Simulation leistungselektronischer Schaltungen häufiger zu finden ist und daher ebenfalls in einem folgenden Unterabschnitt vorgestellt werden soll. Für eine detaillierte Besprechung aller weiteren Lösungsansätze sei der Leser auf die angegebenen Literaturstellen verwiesen.

¹⁴ In [56, 90] wird dieses Problem als „time-constant barrier“ bezeichnet.

¹⁵ Eine mögliche Quelle parasitärer Schaltungskomponenten sind nicht-ideale Schaltermodelle mit sehr kleinen bzw. sehr großen Ersatzwiderständen im ein- bzw. ausgeschalteten Zustand.

¹⁶ Die Exponentialfunktion, mit der die exakte Lösung nach (2.8) bzw. (2.15) gewonnen wird, ist über eine Taylorreihe mit unendlich vielen Gliedern definiert.

Implizite Integrationsverfahren

Analog zum expliziten Eulerverfahren (2.11) existiert auch eine implizite Variante:

$$v_{C,n+1} = v_{C,n} + h\dot{v}_{C,n+1} \quad (2.16)$$

Der Unterschied zwischen beiden Varianten ist der Zeitpunkt zu dem die Ableitung ausgewertet wird: $t_{n+1} = (n+1)h$ statt $t_n = nh$. Einsetzen der ursprünglichen Differentialgleichung (2.7) in (2.16) führt auf:

$$v_{C,n+1} = v_{C,n} + h\lambda v_{C,n+1}. \quad (2.17)$$

Die gesuchte Kapazitätsspannung $v_{C,n+1}$ ist mit (2.17) allerdings nur noch implizit gegeben, daher auch die Bezeichnung des Verfahrens als implizit¹⁷.

Der größte Vorteil impliziter Verfahren und gleichzeitig Hauptgrund, weshalb die Mehrzahl der Simulationsprogramme diese verwendet, liegt in ihrer numerischen Stabilität. Das implizite Eulerverfahren ist für beliebig große Schrittweiten stabil [4], vorausgesetzt das System besitzt ausschließlich Eigenwerte mit negativem Realteil. Eine anschauliche Erklärung gibt Abb. 2.7. Am Beispiel der einfachen Differentialgleichung (2.7) zeigen die Diagramme links und rechts die Lösung des expliziten bzw. des impliziten Eulerverfahrens, wobei der Zeitschritt mit $h = \frac{2,1}{|\lambda|}$ so groß gewählt ist, dass die Lösung des expliziten Eulerverfahrens instabiles Verhalten aufweist. Die Lösung des impliziten Euler-Verfahrens zeigt allerdings stabiles Verhalten. Die Wahl des Zeitschritts des impliziten Verfahrens kann daher an der gewünschten Genauigkeit ausgerichtet werden und ist nicht aufgrund einer Instabilität begrenzt. Dank des in Abb. 2.7 zusätzlich eingetragenen Richtungsfelds¹⁸ können die Ursachen der Instabilität bzw. des stabilen Verhaltens anschaulich nachvollzogen werden. Während das explizite Eulerverfahren (in Abb. 2.7 links) eine Tangente zum jeweiligen momentanen Zeitpunkt (Index n) anlegt und die zukünftige Lösung (Index $n+1$) daraus extrapoliert, legt die implizite Variante (in Abb. 2.7 rechts) die Tangente im zukünftigen, noch unbekanntem Lösungspunkt (Index $n+1$) an. Diese Tangente muss zusätzlich durch den momentanen Lösungspunkt (Index n) verlaufen. Auf das Richtungsfeld bezogen heißt das, dass die Steigung der numerischen Lösung (in Abb. 2.7 gestrichelt) im expliziten Fall mit dem Pfeil des Richtungsfelds im linken Lösungspunkt übereinstimmt und im impliziten Fall mit dem Pfeil im rechten Lösungspunkt. Auf diese Weise stabilisiert sich das implizite Verfahren bei Lösungsfunktionen mit negativem Realteil im Eigenwert (positive Dämpfung) selbst.

¹⁷ Zur Lösung der impliziten Gleichung wenden Simulationsprogrammen häufig eine Fixpunktiteration [4] an. Der Startwert, den eine solche Iteration benötigt, stellt häufig eine explizite Integrationsmethode bereit. Eine solche Kombination aus explizitem Verfahren zur Bestimmung eines Startwerts und der Fixpunktiteration auf Basis eines impliziten Verfahrens wird auch als Prädiktor-Korrektor-Verfahren bezeichnet [4].

¹⁸ Das Richtungsfeld der Differentialgleichung (2.7) gibt für jeden Punkt in der v_C - t -Ebene die Steigung der Kondensatorspannung v_C wieder. Die Länge der Pfeile ist prinzipiell beliebig und beinhaltet keine Information.

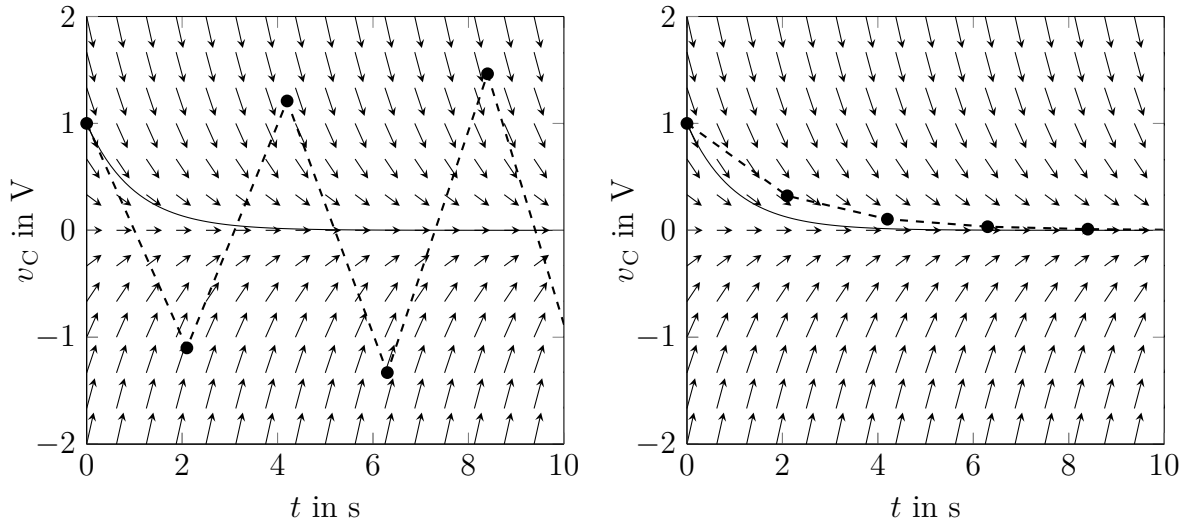


Abbildung 2.7.: Exakte Lösung (durchgezogen) und Lösungen des expliziten (links, gestrichelt) und impliziten (rechts, gestrichelt) Euler-Verfahrens der DGL (2.7) für eine Schrittweite $h = 2,1$ s und $\lambda = -1$ s⁻¹, $v_{C,0} = 1$ V; die Pfeile geben das Richtungsfeld der DGL wieder

Sowohl SPICE als auch viele weitere Simulationsprogramme, wie z.B. PETS [39], die auf der MNA oder STA basieren, implementieren ausnahmslos implizite Integrationsverfahren. Neben dem Eulerverfahren sind dies das Trapezverfahren

$$v_{C,n+1} = v_{C,n} + \frac{h}{2}(\dot{v}_{C,n+1} + \dot{v}_{C,n}) \quad (2.18)$$

und die sogenannten BDF-Verfahren [4]. Allerdings werden die Integrationsverfahren nicht direkt auf das Differentialgleichungssystem in Matrizenform angewendet. Seit den Anfängen von SPICE [30] bis heute [48] werden sogenannte „companion models“ zur Formulierung verwendet. Dabei wird die Strom-Spannungsbeziehung jedes einzelnen dynamischen Bauelements mittels des Integrationsverfahrens in eine zeitdiskrete Strom-Spannungsbeziehung gewandelt. Daraus ergibt sich wiederum ein Ersatzschaltbild, das „companion model“. Anhand einer Kapazität und dem impliziten Eulerverfahren soll ein solches Ersatzschaltbild exemplarisch hergeleitet werden. Die Strom-Spannungsbeziehung an einer Kapazität lautet:

$$i_C = C\dot{v}_C. \quad (2.19)$$

Ersetzen der Ableitung in (2.19) durch den Zusammenhang aus (2.16) führt auf¹⁹:

$$i_{C,n+1} = \frac{C}{h}v_{C,n+1} - \frac{C}{h}v_{C,n}. \quad (2.20)$$

¹⁹ Das implizite Eulerverfahren wird hier eigentlich nicht zur Integration sondern zur Näherung der Ableitung, also zur numerischen Differentiation, angewendet.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

Abb. 2.8 stellt den Zusammenhang aus (2.20) als elektrisches Netzwerk, dem „companion model“ einer Kapazität, dar. Die Stromquelle benötigt die Kapazitätsspannung $v_{C,n}$ aus dem vorhergehenden Zeitschritt. In ähnlicher Weise lässt sich auch ein „companion model“ für Induktivitäten herleiten [4].

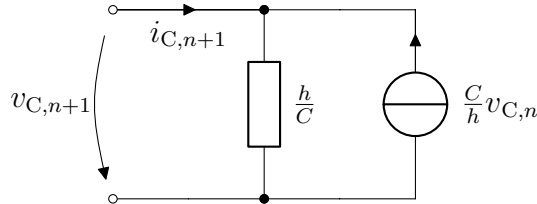


Abbildung 2.8.: Mit implizitem Eulerverfahren generiertes „companion model“ einer Kapazität

Werden alle dynamischen Netzwerkelemente durch ihr jeweiliges „companion model“ ersetzt, so resultiert aus dem gemischt differential-algebraischen Gleichungssystem DAE ein rein algebraisches. Solange der Zeitschritt h konstant bleibt und es sich um ein lineares, zeitinvariantes Netzwerk handelt, ändern sich zu jedem neuen Zeitpunkt ausschließlich die Werte der unabhängigen Quellen im „companion model“. Das restliche Netzwerk bleibt gleich. Das Beispiel in Abb. 2.8 zeigt, dass sich nur der Wert der Ersatzstromquelle nicht aber der des Ersatzwiderstands ändert. Für das Gleichungssystem des STA bzw. MNA bedeutet dies eine konstante Matrix, denn die unabhängigen Quellen befinden sich auf der rechten Seite des Gleichheitszeichens. Dieser Umstand kann geschickt ausgenutzt werden, sodass nicht das gesamte Gleichungssystem zu jedem Zeitpunkt von neuem gelöst werden muss [4].

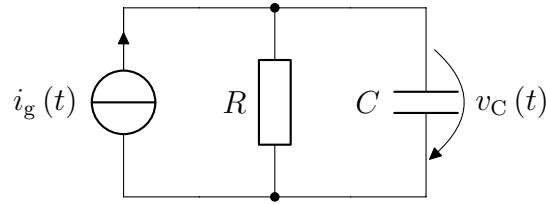
Einen anderen Ansatz verfolgen Programme wie SimPowerSystems [77] oder PLECS [73], die das SSA verwenden. In beiden Programmen wird die Lösung über die in MATLAB/-Simulink implementierten numerischen Integrationsverfahren, die sogenannten „ODE-solver“, bestimmt. Beide Handbücher [77, 95] empfehlen die Verfahren mit der Bezeichnung „ode23tb“ und „ode15s“ zu verwenden, die beide auf impliziten Integrationsverfahren basieren. Der Autor von SWEAP [18] hingegen empfiehlt ein explizites Verfahren (Runge-Kutta-Verfahren vierter Ordnung), welches allerdings, wie das explizite Euler-Verfahren, ein stark eingeschränktes Stabilitätsgebiet besitzt.

Matrixexponentialfunktion

Ausgangspunkt der folgenden Betrachtung ist erneut die Schaltung aus Abb. 2.5, die hier nochmals dargestellt sei.

Die Spannung über der Kapazität ergibt sich als Lösung der Differentialgleichung:

$$\dot{v}_C(t) = \lambda v_C(t) + \frac{1}{C} i_g(t). \quad (2.21)$$

Abbildung 2.9.: Netzwerk aus Abb. 2.5 mit beliebiger Quellfunktion $i_g(t)$

Das Verfahren der „Variation der Konstanten“ (s. [4]) sieht einen Lösungsansatz mit Hilfe der Exponentialfunktion und einer noch zu bestimmenden, zeitabhängigen Funktion $c(t)$ als Lösung der Differentialgleichung vor:

$$v_C(t) = e^{\lambda t} c(t). \quad (2.22)$$

Einsetzen dieses Ansatzes und dessen Ableitung in die ursprüngliche Differentialgleichung (2.21) liefert:

$$\dot{c}(t) = e^{-\lambda t} \frac{1}{C} i_g(t). \quad (2.23)$$

Ein Integrieren beider Seiten (2.23) liefert:

$$c(t) = \int_{t_0}^t e^{-\lambda \tau} \frac{1}{C} i_g(\tau) d\tau + c(t_0). \quad (2.24)$$

Zur Bestimmung der Konstante $c(t_0)$ wird (2.24) in den ursprünglichen Ansatz (2.22) eingesetzt und zum Zeitpunkt $t = t_0$ ausgewertet:

$$c(t_0) = e^{-\lambda t_0} v_C(t_0). \quad (2.25)$$

Einsetzen von (2.25) und (2.24) in (2.22) liefert schließlich die gesuchte Lösung:

$$v_C(t) = e^{\lambda(t-t_0)} v_C(t_0) + e^{\lambda t} \int_{t_0}^t e^{-\lambda \tau} \frac{1}{C} i_g(\tau) d\tau. \quad (2.26)$$

Während die Schaltung aus Abb. 2.9 nur ein einziges dynamisches Bauelement enthält (die Kapazität C), so bestehen typische leistungselektronische Schaltungen aus einer Vielzahl dynamischer Elemente. Eine Schaltungsanalyse führt daher auf mehrere, miteinander verkoppelte Differentialgleichungen, die sich mit Hilfe des SSA als explizites ODE in Matrizenform darstellen lassen:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t). \quad (2.27)$$

Analog zur skalaren Exponentialfunktion existiert auch eine Matrixexponentialfunktion,

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

die ähnlich definiert ist²⁰:

$$e^{\lambda t} = 1 + \lambda t + \frac{1}{2!}(\lambda t)^2 + \frac{1}{3!}(\lambda t)^3 + \dots \quad (2.28)$$

$$e^{\mathbf{A}t} = \mathbf{1} + \mathbf{A}t + \frac{1}{2!}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 + \dots \quad (2.29)$$

Mit Hilfe der Matrixexponentialfunktion (2.29) lässt sich die Lösung von (2.27) analog zum skalaren Fall angeben:

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + e^{\mathbf{A}t} \int_{t_0}^t e^{-\mathbf{A}\tau} \mathbf{B}\mathbf{u}(\tau) d\tau. \quad (2.30)$$

Ein in der Literatur [65, 78–80, 83, 88] weit verbreiteter Lösungsansatz für das Integral in (2.30) besteht in der stückweise konstanten Näherung der Eingangsgrößen $\mathbf{u}(t)$. Wie bei den numerischen Integrationsverfahren (explizites/implizites Eulerverfahren) wird die Lösung nur für einzelne Zeitpunkte bestimmt, d. h. zeitlich diskretisiert. Zwischen diesen Zeitpunkten werden die Eingangsgrößen als konstant angenommen²¹. Abb. 2.10 zeigt eine solche Näherung für eine beispielhafte Eingangsgröße.

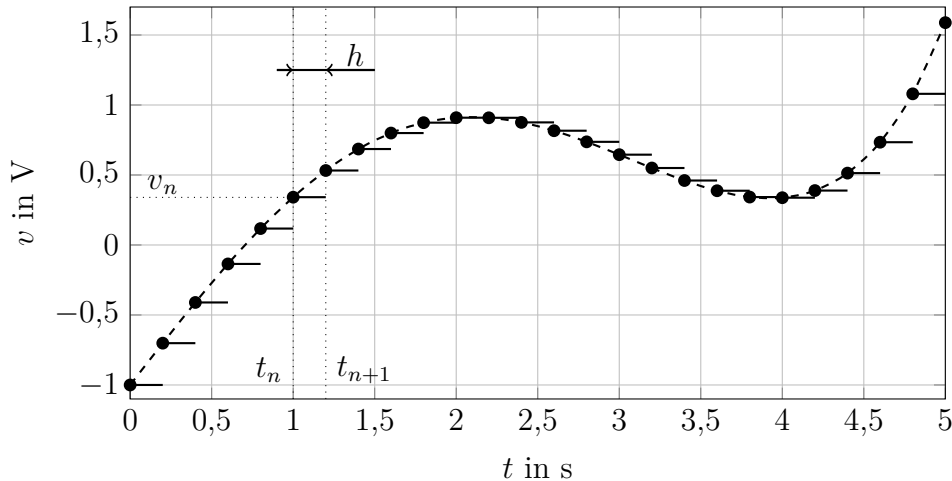


Abbildung 2.10.: Stückweise konstante Näherung (durchgezogen mit Punkten) einer beispielhaften Spannungsquelle (gestrichelt)

Mit der bereits aus den numerischen Integrationsverfahren bekannten Notation $\mathbf{u}_n = \mathbf{u}(t_n)$ lässt sich das Integral im Zeitintervall $t_n < t < t_{n+1}$ mit der Zeitdauer $h = t_{n+1} - t_n$ lösen²²:

²⁰ Die Taylorreihe dient hier nur zur Definition der Exponentialfunktion und ist in dieser Form zur rechnergestützten Auswertung nicht zu empfehlen [91].

²¹ Auch die bereits erwähnten numerischen Integrationsmethoden nähern die Eingangsgrößen durch einen stückweise konstanten Zeitverlauf an.

²² Eine ausführliche Herleitung von 2.31 kann [4] entnommen werden. Darin wird auch gezeigt, dass eine explizite Berechnung von \mathbf{A}^{-1} nicht nötig ist, bzw. die Inverse der Systemmatrix nicht notwendigerweise existieren muss.

$$\int_{t_n}^{t_{n+1}} e^{-\mathbf{A}\tau} \mathbf{B} \mathbf{u}_n d\tau = e^{-\mathbf{A}t_{n+1}} (e^{\mathbf{A}h} - \mathbf{1}) \mathbf{A}^{-1} \mathbf{B} \mathbf{u}_n. \quad (2.31)$$

Die Gesamtlösung (2.30) lässt sich mit dem Ergebnis aus (2.31) und der zeitdiskreten Notation mit \mathbf{x}_{n+1} als Näherungswert für $\mathbf{x}(t_{n+1})$ und \mathbf{x}_n als Näherung für $\mathbf{x}(t_n)$ übersichtlich darstellen:

$$\mathbf{x}_{n+1} = e^{\mathbf{A}h} \mathbf{x}_n + (e^{\mathbf{A}h} - \mathbf{1}) \mathbf{A}^{-1} \mathbf{B} \mathbf{u}_n. \quad (2.32)$$

Handelt es sich bei den Eingangsgrößen in \mathbf{u} um reine DC-Quellen, so gibt (2.32) die exakte Lösung an, unabhängig vom gewählten Zeitschritt. Andernfalls hängt die Genauigkeit der Näherung davon ab, wie gut die Eingangsgrößen über stückweise konstante Werte angenähert werden können. Es gilt: Je kleiner der Zeitschritt h , umso besser die Näherung. Allerdings steigt mit kleiner werdenden Zeitintervallen auch der Rechenaufwand und die Auswirkungen eventueller Rundungsfehler auf die Lösung [4]. In [81, 90, 96] hingegen wird der Zeitverlauf der Eingangsgrößen stückweise über Polynome angenähert, wodurch größere Zeitschritte bzw. eine höhere Genauigkeit erzielt werden können.

Ein alternativer Ansatz stellt die Eingangsgrößenmodellierung [97] dar, die sich, dank ihrer Systematik, gut für die rechnergestützte Umsetzung eignet. Die Methode unterscheidet sich insofern von den zuvor genannten, als dass die Problematik der näherungsweise Lösung des Integrals vollständig vermieden wird. Das Grundprinzip der Eingangsmodellierung soll anhand einer sinusförmigen Spannungsquelle $v_g(t) = \hat{v} \sin(\omega_0 t + \varphi)$ und dem Schaltbild nach Abb. 2.11 erläutert werden.

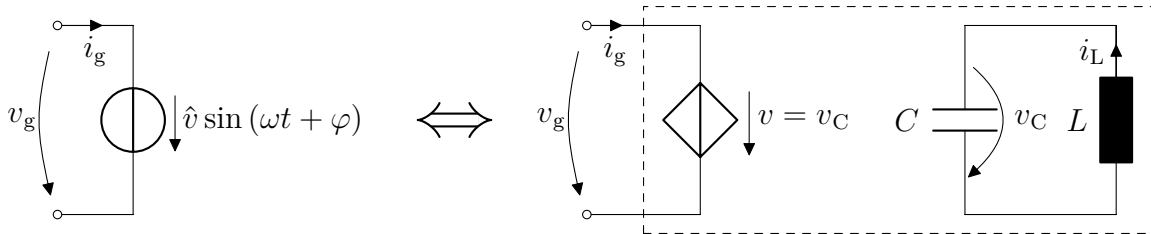


Abbildung 2.11.: Eingangsgrößenmodellierung einer sinusförmigen Spannungsquelle

Die Lösung der Spannung über der Kapazität in Abb. 2.11 lässt sich in Abhängigkeit der Anfangswerte von Kapazitätsspannung $v_{C,0}$ und Induktivitätsstrom $i_{L,0}$, sowie den charakteristischen Größen $\omega_0 = \frac{1}{\sqrt{LC}}$ und $Z_0 = \sqrt{\frac{L}{C}}$ angeben²³:

$$\begin{aligned} v_C &= v_{C,0} \cos(\omega_0 t) + i_{L,0} Z_0 \sin(\omega_0 t) \\ &= \sqrt{v_{C,0}^2 + i_{L,0}^2 Z_0^2} \sin\left(\omega_0 t + \arctan\left(\frac{v_{C,0}}{i_{L,0} Z_0}\right)\right) \end{aligned} \quad (2.33)$$

²³ Um Fallunterscheidungen aufgrund der Arkustangensfunktion zu vermeiden, wird an dieser Stelle davon ausgegangen, dass $i_{L,0} \geq 0$ gilt.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

Damit die gesteuerte Quelle exakt den gleichen Spannungsverlauf wie die unabhängige Quelle besitzt, müssen folgende Zusammenhänge gelten:

$$\hat{v} = \sqrt{v_{C,0}^2 + i_{L,0}^2 Z_0^2} \quad \text{und} \quad \varphi = \arctan\left(\frac{v_{C,0}}{i_{L,0} Z_0}\right) \quad \text{sowie} \quad \omega = \omega_0. \quad (2.34)$$

Eine Wahl der Anfangswerte, der Induktivität L sowie der Kapazität C , welche die Forderungen nach (2.34) erfüllen, führt dazu, dass die beiden Quellen in Abb. 2.11 äquivalent sind. Werden in einem Netzwerk sämtliche unabhängige Quellen durch ihre Eingangsgrößenmodelle ersetzt, so besitzt das Netzwerk keine externe Anregung mehr. Es wird allein durch die Anfangswerte der Eingangsgrößenmodelle angeregt. Der Eingangsgrößenvektor \mathbf{u} entfällt und damit auch das Integral in (2.30):

$$\tilde{\mathbf{x}}(t) = e^{\tilde{\mathbf{A}}(t-t_0)} \tilde{\mathbf{x}}(t_0). \quad (2.35)$$

Die Tilde in (2.35) dient der Unterscheidung zwischen der ursprünglichen Schaltung mit der Systemmatrix \mathbf{A} und dem Zustandsvektor \mathbf{x} und der um die Eingangsgrößenmodelle erweiterten Schaltung mit der Systemmatrix $\tilde{\mathbf{A}}$ und dem Zustandsvektor $\tilde{\mathbf{x}}$. Eine Auswertung zu äquidistant verteilten Zeitpunkten mit der Schrittweite h lässt sich analog zu (2.32) darstellen:

$$\tilde{\mathbf{x}}_{n+1} = e^{\tilde{\mathbf{A}}h} \tilde{\mathbf{x}}_n. \quad (2.36)$$

Die Eingangsgrößenmodellierung ist prinzipbedingt auf Polynom-, Sinus- bzw. Kosinusfunktionen, Exponentialfunktionen sowie deren stückweise Kombination beschränkt. Sie wird angewendet in PECS [75] sowie in [86, 87, 98]. Weiterführendes zur Eingangsgrößenmodellierung und der rechnergestützten Berechnung der Matrixexponentialfunktion wird in Kapitel 4.1 behandelt.

Wie schon bei den numerischen Integrationsverfahren ergibt sich ein Vorteil, wenn zu Beginn der Simulation ein fester Zeitschritt h gewählt wird. Die aufwendige Berechnung der Matrixexponentialfunktion $e^{\mathbf{A}h}$ muss dann nur ein einziges Mal zu Beginn der Simulation durchgeführt werden. So lange sich das Netzwerk nicht ändert, bleibt die Systemmatrix und damit die Lösung der Matrixexponentialfunktion erhalten. Voraussetzung dafür ist jedoch ein lineares, zeitinvariantes Netzwerk. Nahezu alle Veröffentlichungen, die zur transienten Schaltungssimulation das SSA anwenden, nutzen zur Lösung der transienten Simulation die Matrixexponentialfunktion mit einem oder einigen wenigen festen Zeitschritten [65, 75, 78–80, 83, 85, 88, 90, 99].

2.2.3. Nichtlineare Schaltungselemente

Die beiden Hauptaufgaben eines Programms zur transienten Simulation leistungselektronischer Schaltungen sind der effiziente Umgang mit dynamischen sowie nichtlinearen Schaltungselementen. Während der vorherige Abschnitt dynamische Elemente betrachtet

stellt dieser eine Auswahl an Ansätzen zur Lösung der Problematik durch nichtlineare Schaltungselemente vor. Die Schaltung nach Abb. 2.12 soll dazu als einführendes Beispiel dienen.

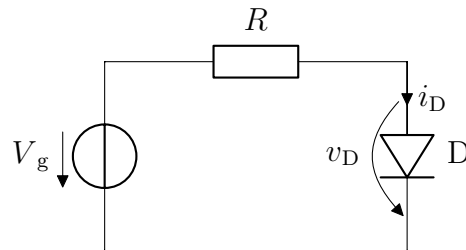


Abbildung 2.12.: Beispielnetzwerk mit nichtlinearem Schaltungselement

Die Diode in Abb. 2.12 wird durch eine vereinfachte Kennlinie mit der nichtlinearen Strom-/Spannungsbeziehung

$$i_D = i_0 \left(e^{\frac{v_D}{v_T}} - 1 \right) \quad (2.37)$$

modelliert [19]. Die Anwendung der Kirchhoff'schen Regeln sowie der Strom-Spannungsbeziehungen führt auf die nichtlineare Gleichung

$$0 = v_D + i_0 R \left(e^{\frac{v_D}{v_T}} - 1 \right) - v_g. \quad (2.38)$$

Eine Auflösung von (2.38) nach v_D lässt sich auf analytischem Wege nicht erreichen. Ein möglicher Lösungsansatz ist eine grafische Auswertung im v_D - i_D -Diagramm (s. Abb. 2.13). Der Schnittpunkt der Diodenkennlinie (2.37) mit der Quellenkennlinie $i_D = \frac{v_g - v_D}{R}$, d. h. der Serienschaltung aus Spannungsquelle und Widerstand, legt die Lösung fest.

Die grafische Lösungsfindung eignet sich nicht für die rechnergestützte Umsetzung. Ein anderer Lösungsansatz versucht sich der Lösung iterativ zu nähern, indem die nichtlineare Diodengleichung (2.37) in jedem Schritt durch die ersten beiden Glieder ihrer Taylorreihe ersetzt wird. Der Schnittpunkt dieser Tangenten mit der Quellenkennlinie dient als Entwicklungsstelle der Taylorreihe im folgenden Iterationsschritt. Das Verfahren ist für drei Iterationsschritte und dem Startwert $v_{D,0} = 0,53 \text{ V}$ in Abb. 2.14 dargestellt. Die Schnittpunkte der Tangenten mit der Quellkennlinie nähern sich mit jedem Schritt näher an die exakte Lösung an. Die Iteration kann abgebrochen werden, sobald eine gewünschte Genauigkeit erreicht ist.

Eine Alternative zur lokalen linearen Approximation ist die Nachbildung der nichtlinearen Kennlinie mittels stückweise linearer Verläufe. Abb. 2.15 zeigt das Beispiel einer solchen Approximation mittels dreier Segmente. Die näherungsweise Lösung ergibt sich durch den Schnittpunkt der Quellenkennlinie und dem entsprechenden Segment.

Die lokale lineare und die stückweise lineare Approximation stellen die beiden am häufigsten angewendeten Methoden der rechnergestützten nichtlinearen Schaltungssimulation

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

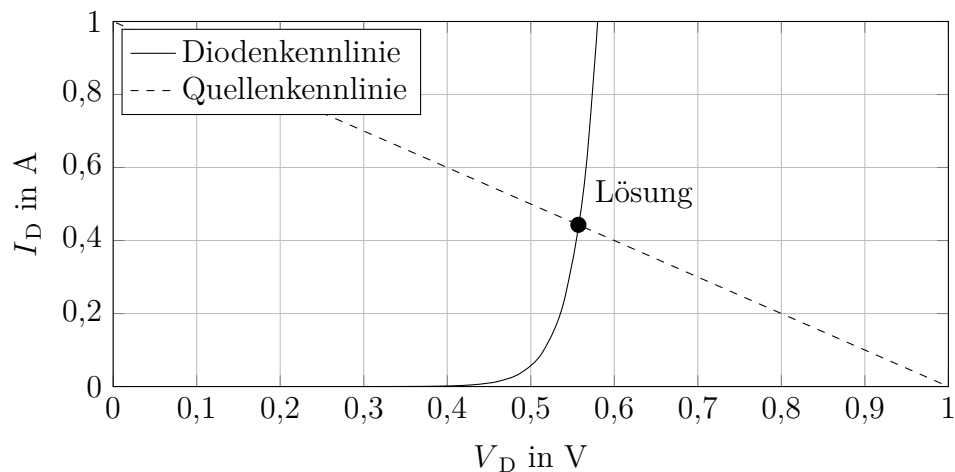


Abbildung 2.13.: v_D - i_D -Diagramm mit Kennlinien zur grafischen Bestimmung der Lösung mit $v_g = 1$ V, $R = 1 \Omega$, $i_0 = 1$ nA und $v_T = 28$ mV

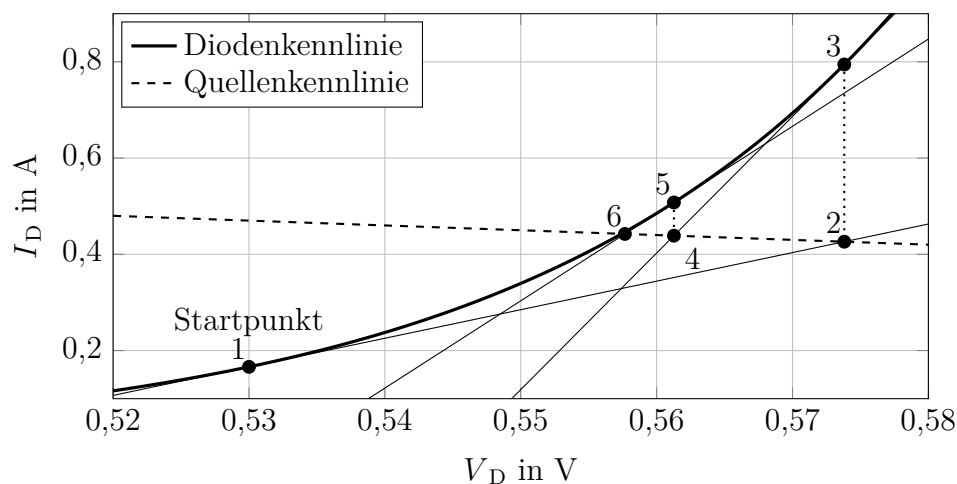


Abbildung 2.14.: Lokale lineare Approximation der Diodenkennlinie mit Tangenten (dünne, durchgehende Linien) und Zwischenergebnissen (nummerierte Marker)

dar [100]. Die folgenden Unterabschnitte geben daher eine etwas detailliertere Einsicht in deren konkrete Anwendung und Umsetzung.

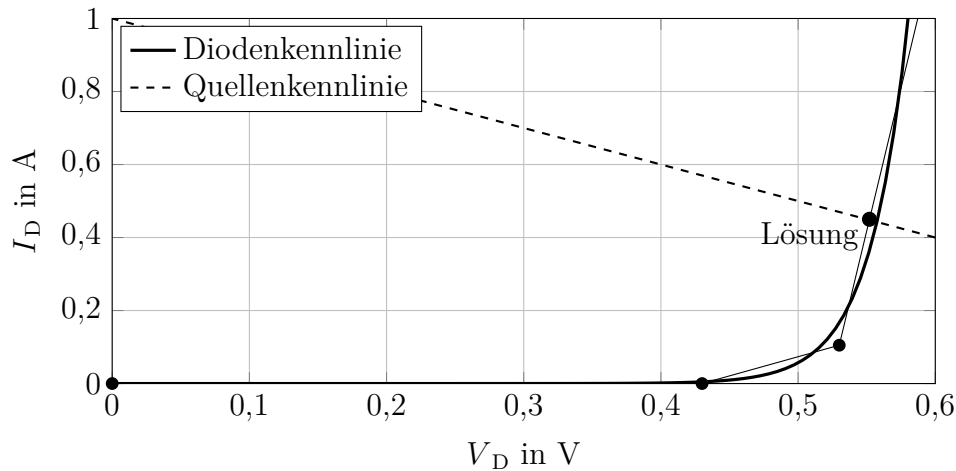


Abbildung 2.15.: Stückweise lineare Approximation der Diodenkennlinie mit drei Geradenstücken

Lokale lineare Approximation

Die Lösung einer nichtlinearen Gleichung mittels iterativer Auswertung der ersten beiden Taylorreihenglieder ist in der Literatur als das Newton-Raphson-Verfahren bekannt [4]. Dessen allgemeine Rekursionsvorschrift zur näherungsweise Lösung einer (nichtlinearen) Gleichung $f(x) = 0$ mit einer stetig differenzierbaren Funktion $f(x)$ lautet:

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)} \quad \text{mit} \quad f'(x_m) = \left. \frac{df(x)}{dx} \right|_{x=x_m}. \quad (2.39)$$

Im Vergleich zu anderen Lösungsverfahren benötigt das Newton-Raphson-Verfahren in jedem Iterationsschritt sowohl den Funktionswert selbst als auch den Wert der Ableitung. Vor allem Letzteres stellt in nicht wenigen Fällen ein Problem dar. Der eigentliche Nachteil des Verfahrens ist jedoch die Anfälligkeit für Konvergenzprobleme, wenn der Startwert x_0 nicht in „unmittelbarer Nähe“ der exakten Lösung liegt [32]. Dies liegt in der Tatsache begründet, dass lediglich die ersten beiden Glieder der Taylorreihe verwendet werden. Das Newton-Raphson-Verfahren wird daher meist in Kombination mit anderen (jedoch langsameren) Verfahren verwendet. Sobald das Zwischenergebnis durch die vorhergehenden Schritte im sogenannten Konvergenzradius liegt, kann mit Hilfe weniger Iterationen des Newton-Raphson-Verfahrens eine hohe Genauigkeit erzielt werden.

Auch das Schaltungssimulationsprogramm SPICE verwendet die lokale lineare Approximation, um aus nichtlinearen Netzwerkelementen lineare Gleichungssysteme zu generieren. Die rechnergestützte Umsetzung des Verfahrens geschieht mit Hilfe sogenannter „companion models“, das heißt Ersatznetzwerken. Wie schon im Fall der numerischen Integration (s. Kapitel 2.2.2) wird die Linearisierung vor der eigentlichen Formulierung des Gleichungssystems angewendet. Unter Anwendung der Rekursionsschreibweise lautet

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

die lokale lineare Approximation der Diodenkennlinie aus (2.37) um die Entwicklungsstelle $v_{D,m}$:

$$\begin{aligned} i_{D,m+1} &= i_0 \left(e^{\frac{v_{D,m}}{v_T}} - 1 \right) + \frac{i_0}{v_T} e^{\frac{v_{D,m}}{v_T}} (v_{D,m+1} - v_{D,m}) \\ &= \underbrace{\frac{i_0}{v_T} e^{\frac{v_{D,m}}{v_T}}}_{R_{\text{eq},m}^{-1}} v_{D,m+1} + \underbrace{i_0 \left(e^{\frac{v_{D,m}}{v_T}} \left(1 - \frac{v_{D,m}}{v_T} \right) - 1 \right)}_{i_{\text{eq},m}}. \end{aligned} \quad (2.40)$$

Daraus lässt sich das Ersatznetzwerk aus Abb. 2.16 ableiten. Im Gegensatz zur numerischen Integration (vgl. Abb. 2.8) steht der Zähindex m allerdings nicht für einen bestimmten Zeitpunkt, sondern für den Iterationsschritt des Newton-Raphson-Verfahrens.

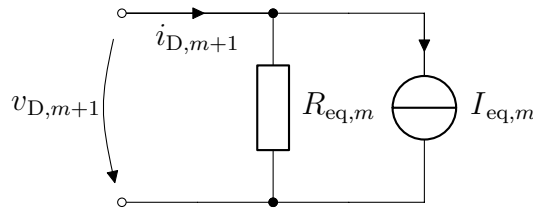


Abbildung 2.16.: Mit dem Newton-Raphson-Verfahren generiertes „companion model“ einer Diode

Die Anwendung der lokalen linearen Approximation auf jeden einzelnen nichtlinearen Netzwerkzweig ändert nichts an der zuvor erwähnten Konvergenzproblematik. Die kritische Bestimmung eines geeigneten Startwerts bleibt bestehen. Eine naheliegende Wahl ist die Lösung des vorherigen Zeitschritts als Startwert zu verwenden. Sollte das Newton-Raphson-Verfahren nicht konvergieren, so kann der Zeitschritt verkleinert werden bis entweder die neue Lösung gefunden ist oder die Iteration wegen Unterschreiten eines zuvor festgelegte minimalen Zeitschritts abgebrochen werden muss. Neben dem Newton-Raphson-Verfahren sind in SPICE noch einige zusätzliche Lösungsverfahren implementiert [7].

Bisher wurde lediglich die statische, nichtlineare Kennlinie einer Diode betrachtet. Ein genaueres Modell beinhaltet jedoch auch nichtlineare dynamische Elemente wie bspw. die spannungsabhängige Sperrschichtkapazität. Im Folgenden soll anhand des Beispiels einer nichtlinearen Kapazität gezeigt werden, wie SPICE und ähnliche Programme nichtlineare dynamische Elemente berücksichtigen. In einem ersten Schritt erlaubt die Anwendung numerischer Integrationsverfahren, die nichtlinearen, differentiellen Strom-Spannungsbeziehungen in nichtlineare Differenzgleichungen umzuwandeln. Die Strom-Spannungsbeziehungen der spannungsabhängigen Kapazität

$$i_C = C(v_C) \dot{v}_C \quad (2.41)$$

kann mit Hilfe der impliziten Eulermethode (s. Kapitel 2.2.2) durch

$$i_{C,n+1} = C(v_{C,n}) \frac{v_{C,n+1} - v_{C,n}}{h} \quad (2.42)$$

genähert werden. Das implizite Eulerverfahren verwendet im Zeitintervall $t_n < t \leq t_{n+1}$ demnach einen konstanten Kapazitätswert. Diese Näherung kann mitunter zu großen Fehlern führen [101]. Abhilfe schafft die Verwendung einer ladungsbasierten Beschreibung der nichtlinearen Kapazität²⁴ mit $q_C(v_C)$. Die Differentialgleichung

$$i_C = \dot{q}_C(v_C) \quad (2.43)$$

kann unter Zuhilfenahme des impliziten Eulerverfahrens mit der Differenzengleichung²⁵

$$i_{C,n+1} = \underbrace{\frac{q_C(v_{C,n+1})}{h}}_{i_{C,nl}} - \underbrace{\frac{q_C(v_{C,n})}{h}}_{i_C} \quad (2.44)$$

genähert werden. Der Strom in (2.44) kann in einen nichtlinearen Anteil $i_{C,nl}$ und einen konstanten Anteil i_C zerlegt werden. Auf den nichtlinearen Term $i_{C,nl}$ wird nun, wie zuvor bereits bei der nichtlinearen Kennlinie der Diode, eine lokale lineare Approximation angewendet²⁶:

$$i_{C,nl}^{(m+1)} = \frac{q_C(v_{C,n+1}^{(m)})}{h} + \frac{q'_C(v_{C,n+1}^{(m)})}{h} (v_{C,n+1}^{(m+1)} - v_{C,n+1}^{(m)}) \quad (2.45)$$

Einsetzen von (2.45) in (2.44) liefert schließlich den Strom-Spannungszusammenhang einer Kapazität für den Zeitpunkt t_{n+1} mit der lokalen linearen Approximation im Iterationsschritt $m + 1$:

$$i_{C,n+1}^{(m+1)} = \underbrace{\frac{q'_C(v_{C,n+1}^{(m)})}{h}}_{(R_{C,eq}^{(m)})^{-1}} v_{C,n+1}^{(m+1)} + \underbrace{\frac{q_C(v_{C,n+1}^{(m)})}{h} - \frac{q'_C(v_{C,n+1}^{(m)})}{h} v_{C,n+1}^{(m)} - \frac{q_C(v_{C,n})}{h}}_{i_{C,eq}^{(m)}}. \quad (2.46)$$

In jedem Iterationsschritt müssen die Ladung als auch deren Ableitung nach der Spannung (entspricht der Kapazität) für die Spannung $v_{C,n+1}^{(m)}$ aus der nichtlinearen Kennlinie bestimmt werden. Wie in (2.46) durch die geschweiften Klammern angedeutet, können der erste Term als Kehrwert eines Widerstand und der zweite als Konstantstromquelle interpretiert werden. Das zugehörige Ersatzschaltbild zeigt Abb. 2.17.

Aus einem anfänglich nichtlinearen Differentialgleichungssystem wird mit Hilfe numerischer Integrationsverfahren zunächst ein nicht-lineares algebraisches und mit der lokalen linearen Approximation schließlich ein rein algebraisches Gleichungssystem. SPICE vollzieht diese Umwandlung allerdings bereits auf Elementebene („companion model“) und

²⁴ Gleiches gilt auch für nichtlineare Induktivitäten mit einer flussbasierten Beschreibung.

²⁵ Das Eulerverfahren dient hier genau genommen dazu, die Ableitung durch eine Differenzengleichung anzunähern. Im Fall des impliziten Eulerverfahrens wird das Ergebnis der Differenzengleichung der Ableitung zum Zeitpunkt t_{n+1} zugeschrieben, beim expliziten hingegen der Ableitung zum Zeitpunkt t_n .

²⁶ Der hochgestellte Index m bzw. $m + 1$ bezeichnet den Iterationsschritt der lokalen linearen Approximation (Newton-Raphson-Verfahren).

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

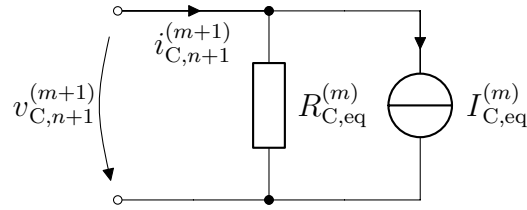


Abbildung 2.17.: Mit der lokalen linearen Approximation und dem impliziten Eulerverfahren generiertes „companion model“ einer nichtlinearen Kapazität

gelangt dadurch direkt auf das algebraische Gleichungssystem des modifizierten Knotenpotentialverfahrens. Eine nachfolgende, aufwendige Berechnung der für das Newton-Raphson-Verfahren benötigten Jacobi-Matrix [4, 101] entfällt somit.

Soll anstatt des in SPICE verwendeten MNA das SSA in Kombination mit der Matrixexponentialfunktion eingesetzt werden, so muss die lokale lineare Approximation auf Differentialgleichungen bzw. ein Differentialgleichungssystem angewendet werden²⁷. Die Theorie der sogenannten exponentiellen Integratoren [103] bedient sich dieser Herangehensweise. Ihre Bezeichnung verdanken sie der Tatsache, dass der lineare Anteil der Lösung über die Matrixexponentialfunktion ermittelt wird. Als Ausgangspunkt der folgenden Betrachtung dient ein homogenes²⁸, nichtlineares Differentialgleichungssystem der Form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)). \quad (2.47)$$

Der erste Schritt besteht darin, den linearen Anteil der Funktion f um die Entwicklungsstelle $\mathbf{x}_n = \mathbf{x}(t_n)$ herum zu extrahieren

$$\dot{\mathbf{x}}(t) = \mathbf{J}_n \mathbf{x}(t) + \mathbf{r}_n(\mathbf{x}(t)). \quad (2.48)$$

Die Matrix \mathbf{J}_n kann durch partielle Ableitungen gewonnen werden und wird dann als Jacobimatrix der Funktion f an der Stelle \mathbf{x}_n bezeichnet²⁹.

$$\mathbf{J}_n = \left. \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_n} \quad (2.49)$$

Die teils aufwendige Berechnung der Jacobi-Matrix kann auch durch eine Approximation derselben ersetzt werden. Der zweite Term in (2.48) $\mathbf{r}_n(\mathbf{x}(t))$ stellt den nichtlinearen Rest dar:

$$\mathbf{r}_n(\mathbf{x}(t)) = f(\mathbf{x}(t)) - \mathbf{J}_n \mathbf{x}(t). \quad (2.50)$$

²⁷ Einen alternativen Ansatz mittels Hilfsquellen zeigen [4, 75, 102]. Aufgrund einiger einschränkender Bedingungen an das Netzwerk soll dieser hier allerdings nicht näher vertieft werden.

²⁸ Die Theorie kann allerdings ohne weiteres auch auf inhomogene Differentialgleichungssysteme übertragen werden.

²⁹ In der Regel wird die Jacobimatrix in (2.48) mit der Differenz $[\mathbf{x}(t) - \mathbf{x}_n]$ multipliziert. Im vorliegenden Fall ist allerdings die Multiplikation aus \mathbf{J}_n und $-\mathbf{x}_n$ im nichtlinearen Restterm $\mathbf{r}_n(\mathbf{x}(t))$ enthalten.

Aus Kapitel 2.2.2 ist die exakte Lösung für (2.48) bereits bekannt:

$$\mathbf{x}(t) = e^{\mathbf{J}_n(t-t_n)}\mathbf{x}_n + e^{\mathbf{J}_nt} \int_{t_n}^t e^{-\mathbf{J}_n\tau} \mathbf{r}_n(\mathbf{x}(\tau)) d\tau. \quad (2.51)$$

Der lineare Anteil der Funktion f kann mit Hilfe der Matrixexponentialfunktion exakt integriert werden. Exponentielle Integratoren unterscheiden sich nun darin, wie mit der Lösung des nichtlinearen Integralteils verfahren wird. Einen guten Überblick geben [103] und [104]. Die einfachste Möglichkeit besteht darin, dass die nichtlineare Funktion innerhalb des Integrals durch einen konstanten Wert ersetzt wird. Die exponentielle Eulermethode verwendet dafür den konstanten Wert $\mathbf{r}_n(\mathbf{x}_n)$ zum Zeitpunkt t_n . Die Lösung mit $h = t_{n+1} - t_n$ ist aus Kapitel 2.2.2 bereits bekannt:

$$\mathbf{x}_{n+1} = e^{\mathbf{J}_nh}\mathbf{x}_n + (e^{\mathbf{J}_nh} - \mathbf{1})\mathbf{J}_n^{-1}\mathbf{r}_n(\mathbf{x}_n). \quad (2.52)$$

Einsetzen von (2.50) in (2.52) führt auf:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + (e^{\mathbf{J}_nh} - \mathbf{1})\mathbf{J}_n^{-1}f(\mathbf{x}_n). \quad (2.53)$$

Wie in [105] gezeigt, kann auch hier eine Vereinfachung ähnlich der Eingangsgrößenmodellierung (s. Kapitel 2.2.2) angewendet werden, wodurch sich die Rechnung auf die Auswertung eines einzigen Matrixexponentials beschränkt. Eine mögliche Anwendung exponentieller Integratoren im Bereich der Schaltungssimulation wird in [81] beschrieben.

Stückweise lineare Approximation

In Abb. 2.18 ist das Beispiel einer stückweise linearisierten Diodenkennlinie aus Abb. 2.15 nochmals mit unterschiedlicher Anzahl an Segmenten dargestellt.

Im Gegensatz zur lokalen Linearisierung repräsentiert ein Segment nicht zwangsweise die ersten beiden Glieder der Taylorreihe in einem jeweiligen Entwicklungspunkt. Die Einteilung in Abschnitte und die Parameter der linearen Approximation können nach beliebigen Kriterien (minimale Fehlerquadrate, kleinste maximale Abweichung, etc.) gewählt werden. Ihre Bestimmung kann in einem vorgelagerten Schritt geschehen und muss nicht, wie beim Newton-Raphson-Verfahren, in jedem Zeitschritt und in jeder Iteration neu berechnet werden. Die näherungsweise Lösung der nichtlinearen Gleichung ergibt sich durch Auffinden des korrekten Segments und Lösen der zugehörigen linearen Gleichung. Besitzt das stückweise linearisierte Netzwerk eine eindeutige Lösung, so ist die Konvergenz durch die begrenzte Anzahl an Segmenten sichergestellt [4, 106]. Der Nachteil ist jedoch direkt aus Abb. 2.18 ersichtlich: Die Genauigkeit, mit der die Kennlinie approximiert wird, ist stark Abhängig von der Anzahl der Segmente. Je höher die geforderte Genauigkeit, desto höher muss die Auflösung sein und desto größer ist der Speicherbedarf sowie der Aufwand zum Auffinden des korrekten Segments. Zudem muss im Verlauf der transienten

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

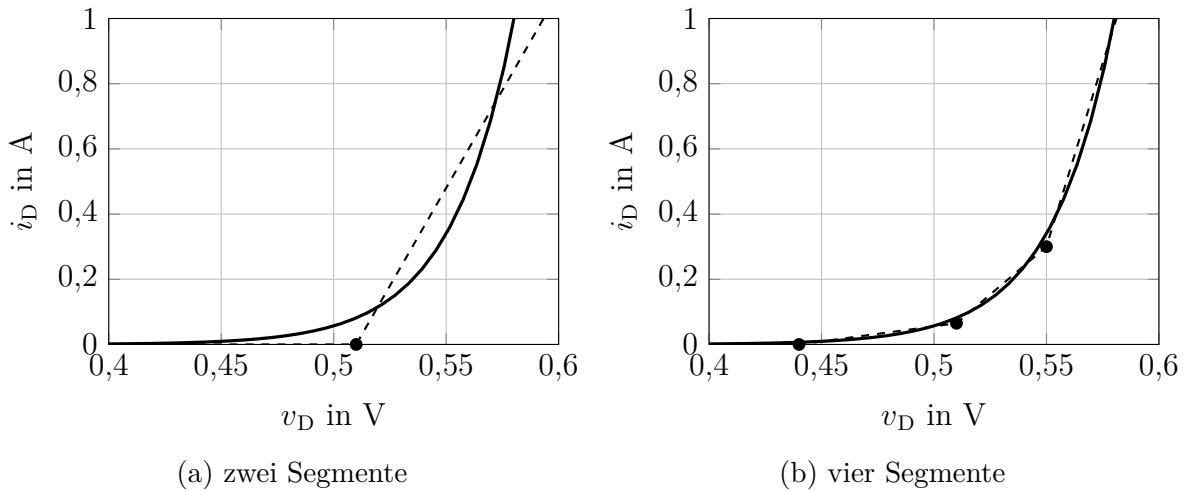


Abbildung 2.18.: Stückweise lineare Approximation (gestrichelt) der Diodenkennlinie (durchgezogen, Diodenparameter aus Abb. 2.13) mit unterschiedlicher Anzahl an Segmenten

Simulation häufiger das Gleichungssystem angepasst werden. Der Vorteil gegenüber der lokalen linearen Approximation schrumpft.

Stückweise lineare Approximation eignet sich besonders dann, wenn das Durchlaufen der exakten nichtlinearen Kennlinie für das Simulationsergebnis eine eher untergeordnete Rolle spielt. Im Bereich leistungselektronischer Schaltungen ist die stückweise lineare Approximation eine sehr häufig eingesetzte Methode, um die nichtlineare Kennlinie schaltender Halbleiterbauelemente anzunähern. Oftmals wird die Kennlinie mit Hilfe von nur zwei Segmenten approximiert: eines für den ein- und das andere für den ausgeschalteten Zustand. Dennoch unterscheiden sich die Simulationsergebnisse dieser stark vereinfachten Modellen nur unwesentlich von denen aufwendigerer, nichtlinearer Modelle. Nur dann, wenn genau diese minimalen Unterschiede interessieren, kann der zusätzliche Rechenaufwand, den Modelle mit einer höheren Anzahl an Segmenten verursachen, gerechtfertigt werden. Daher verwenden nahezu alle Schaltungssimulationsprogramme, die auf leistungselektronische Schaltungen spezialisiert sind, die stückweise lineare Approximation (PLECS [72, 73], PETS [106], SimPowerSystems [77] und einige weitere mehr [18, 45, 50, 107–109]).

Eine Schlüsselrolle innerhalb der stückweise linearen Approximation nimmt der Algorithmus zum Auffinden der korrekten Kombination ein. Eine Kombination³⁰ besteht aus einem Satz an Segmenten - eines pro linearisiertem Schaltungselement. Im Beispiel der Diodenkennlinie aus Abb. 2.18 mit einer eher geringen Anzahl an Segmenten ist die Bedeutung des Suchalgorithmus noch nicht offensichtlich. Wenn das Netzwerk jedoch mehrere (ihre Anzahl sei n) nichtlineare Bauelemente enthält, die jeweils alle durch m

³⁰ Die Kombination wird in der Literatur auch als Zustand des stückweise linearisierten Systems bezeichnet [100].

Segmente stückweise linearisiert sind, so ergeben sich insgesamt m^n Kombinationen. Dieser Zusammenhang zeigt, dass sich (selbst bei geringen m und n) durch das Potenzieren schnell eine Vielzahl an Möglichkeiten ergeben. Ein Suchalgorithmus, der nacheinander alle m^n Kombinationen überprüft, würde die Lösung (oder die Lösungen, falls es mehrere gibt) zwar zuverlässig finden, wäre aber sehr ineffizient. Eine Möglichkeit, den Suchprozess effizienter zu gestalten, liegt darin das Grundprinzip des Newton-Raphson-Verfahrens auf die stückweise lineare Approximation zu übertragen [4]. Anhand des Beispiels der in drei Segmente geteilten Diodenkennlinie (s. Abb. 2.19a) soll das Verfahren kurz vorgestellt werden.

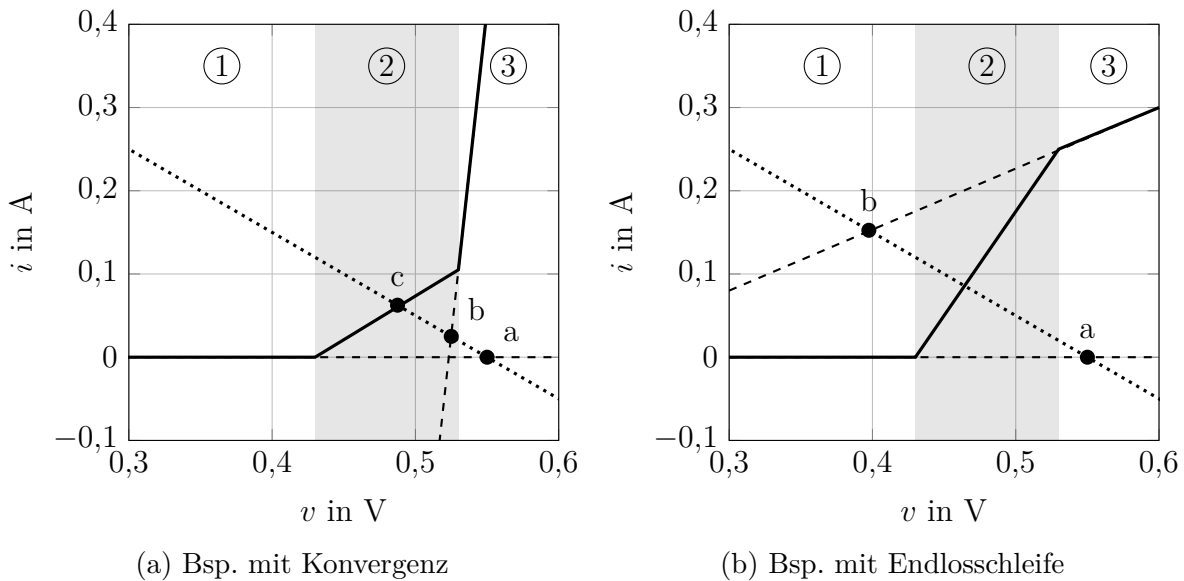


Abbildung 2.19.: Zwei Beispiele stückweiser linearisierter Kennlinien (durchgezogen) mit beispielhafter Quellenkennlinie (gepunktet) und Teillösungen (a,b) des Newton-Raphson-Verfahrens

Begonnen wird das Verfahren nun nicht mehr mit einem Startpunkt sondern mit einer Startkombination, d. h. einem Satz aus Segmenten mit je einem Segment eines jeden linearisierten Schaltungselements. Im Fall nach Abb. 2.19a, soll Segment ① als Ausgangspunkt dienen. Der Schnittpunkt „a“ der Quellenkennlinie und der mit Segment ① linearisierten Kennlinie kommt im Bereich von Segment ③ zu liegen. Die Lösung liegt also nicht im Bereich von Segment ① sondern wahrscheinlich im Bereich von Segment ③. Der nächste Durchgang des Newton-Verfahrens verwendet demnach Segment ③. Der neue Schnittpunkt „b“ befindet sich allerdings im Segment ②. Die Lösung liegt also nicht in Segment ③ sondern wahrscheinlich in Segment ②. Die Auswertung des letzten Durchgangs liefert schließlich einen Schnittpunkt „c“, der auch tatsächlich im zum Segment zugehörigen Bereich (in Abb. 2.19 grau hinterlegt) zu liegen kommt. Es handelt sich demnach nicht um eine Zwischenlösung sondern um eine gültige Lösung. Ob es allerdings noch weitere

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

Lösungen gibt, ist unklar. Für stückweise linearisierte Netzwerke mit mehreren Lösungen ist das Newton-Verfahren nicht geeignet und es müsste auf sogenannte kombinatorische Verfahren (s. [4]) zurückgegriffen werden.

Ähnliche Konvergenzprobleme, wie im Fall der lokalen linearen Approximation, können auch für stückweise linearisierte Kennlinien auftreten. Ein typischer Problemfall ist eine periodisch wiederkehrende Abfolge der gleichen Segmente („cyclic path“ [4]). Abb. 2.19b zeigt das Beispiel eines solchen Konvergenzproblems. Ausgehend von Segment ① liefert das Newton-Verfahren eine Lösung in Segment ③. Der Schnittpunkt aus Quellenkennlinie und der Linearisierung mit Segment ③ liegt allerdings wiederum im Bereich von Segment ①. Ein solcher „cyclic path“ lässt sich allerdings auf einfache Weise durch Aufzeichnung der bereits ausgewerteten Kombinationen detektieren. Sobald eine Kombination zum wiederholten Male auftritt, befindet sich das Verfahren in einer Endlosschleife. Der Algorithmus muss angehalten werden und mit einer neuen, noch nicht ausgewerteten Kombination wieder gestartet werden (im Fall aus Abb. 2.19b, wäre das Segment ②). Sind bereits alle der m^n Kombinationen geprüft, so besitzt das Netzwerk offensichtlich keine Lösung. Der Newton-Raphson-Algorithmus für stückweise linearisierte Schaltungselemente wird unter anderem in [50, 72, 83, 84, 107]³¹ verwendet.

Neben dem Newton-Raphson-Verfahren stellt das Katzenelson-Verfahren ein weiteres, iteratives Lösungsverfahren der stückweise linearen Approximation dar [4]. Es findet unter anderem im Schaltungssimulator PETS [40] sowie in [36, 110] Anwendung. Während der Newton-Algorithmus von einem Iterationsschritt zum nächsten unter Umständen mehrere Segmente überspringt (vgl. Abb. 2.19), so macht der Katzenelson-Algorithmus am Rand eines Segments halt³². Die neue Kombination ergibt sich dann durch den Segmentwechsel in die gegenüberliegende Seite der Segmentgrenze [4]. Das Verfahren arbeitet sich auf diese Weise systematisch voran, bis es schließlich eine gültige Kombination findet.

Für eine umfassende Einführung in Theorie, Modellierung und Lösung stückweise linearisierter Netzwerke sei auf [100] verwiesen³³.

³¹ Die Ausführungen in [107] erwecken den Eindruck, dass das sogenannte Kompensationstheorem [26] zum Auffinden einer gültigen Kombination eingesetzt wird. Allerdings wird das Theorem ausschließlich dazu verwendet, die Lösung für eine neue Kombination auf Basis der vorhergehenden Lösung zu finden. Die Entscheidung, welche Kombination als nächstes getestet wird, beruht auf dem Newton-Raphson-Verfahren.

³² Um die Zwischenlösung an einer Segmentgrenze zu erhalten, moduliert das Katzenelson-Verfahren die Netzwerkerregung, d. h. die Quellen im Netzwerk. Es zählt damit zu den sogenannten Homotopieverfahren.

³³ In [100] wird insbesondere der Lösungsansatz mit Hilfe des linearen, dynamischen Komplementärproblems behandelt.

2.2.4. Anfangswerte

Die transiente Simulation eines Netzwerks mit dynamischen Netzwerkelementen (Kapazitäten/Induktivitäten) stellt aus mathematischer Sicht ein sogenanntes Anfangswertproblem dar. Damit eine eindeutige Lösung bestimmt werden kann, ist es zwingend notwendig, dem Simulationsprogramm Anfangswerte mitzuteilen. In der Regel sind das die Kapazitätsspannungen und Induktivitätsströme zu Beginn der Simulation ($t = 0$). Der weitere Verlauf sämtlicher Spannungen und Ströme im Netzwerk hängt von diesen Anfangswerten ab. Viele Schaltungssimulatoren, wie etwa SPICE, akzeptieren beliebige, vom Anwender vorgegebene Anfangswerte. Häufig unterstützen sie auch eine Kombination aus vorgegebenen und selbstständig ermittelten Anfangswerten.

Ein weit verbreitetes Verfahren zur selbstständigen Ermittlung von Anfangswerten ist die sogenannte DC-Analyse³⁴ [7]. In einem zur transienten Simulation vorgelagerten Schritt werden alle Kapazitäten bzw. Induktivitäten, deren Anfangswerte durch den Anwender vorgegeben sind, durch Spannungsquellen bzw. Stromquellen ersetzt. Die noch verbleibenden Kapazitäten und Induktivitäten werden durch Leerläufe bzw. Kurzschlüsse ersetzt. Die Lösung dieses Ersatznetzwerks, d. h. der Arbeitspunkt, dient anschließend als Anfangswert für die transiente Simulation. Dieser Ansatz führt jedoch nicht immer zum gewünschten Ergebnis. Als Beispiel sei eine Reihenschaltung, bestehend aus einer Spannungsquelle und zwei Kapazitäten betrachtet. Wenn keine der Kapazitätsspannungen in ihrem Anfangswert vom Anwender spezifiziert wurde, ist das Potential des Verbindungsknotens zwischen beiden Kapazitäten beliebig. Das zugehörige Gleichungssystem ist unterbestimmt. Im umgekehrten Fall, wenn beide Kapazitätsspannung mit einem Anfangswert versehen sind, resultiert ein überbestimmtes Gleichungssystem. Es können nicht Anfangswerte für beide Kapazitäten vorgegeben werden, wenn die Spannung über beide Kapazitäten bereits durch die Spannungsquelle eingepreßt ist. Eine solche Kombination aus Anfangswerten, wird auch als inkonsistente Anfangswerte³⁵ bezeichnet. Konsistente Anfangswerte zu finden, kann sich jedoch noch deutlich schwieriger gestalten, als bislang beschrieben, wie das Beispiel aus Abb. 2.20 zeigt.

Die DC-Analyse ersetzt beide Kapazitäten durch Leerläufe und liefert die Anfangswerte $v_{C,1}(0) = 0 \text{ V}$ und $v_{C,2}(0) = 0 \text{ V}$. Allerdings ist nur der Anfangswert der ersten Kapazität, $v_{C,1}(0)$, korrekt. Beim Anfangswert der zweiten Kapazität, $v_{C,2}(0)$, versagt die DC-Analyse, denn der korrekte Wert lautet

$$v_{C,2}(0) = g_{i_{C,1}}(0) = g_{C_1} \left. \frac{dv_{C,1}}{dt} \right|_{t=0} = g_{C_1} \left. \frac{dv_g}{dt} \right|_{t=0} = 1 \text{ V}. \quad (2.54)$$

³⁴ SPICE wendet die DC-Analyse zur Bestimmung der Anfangswerte an, wenn das Attribut „uic“ nicht gesetzt ist [7].

³⁵ Die Definition des „inkonsistenten Anfangswerts“ unterscheidet sich je nach Literaturstelle. Eine Alternative ist die Definition über seine Auswirkung. Anfangswerte sind danach genau dann inkonsistent, wenn zu Beginn der transienten Simulation ($t = 0$) Ausgleichsvorgänge stattfinden, die unendlich hohe Strom- und/oder Spannungsimpulse (Dirac-Stöße) aufweisen.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

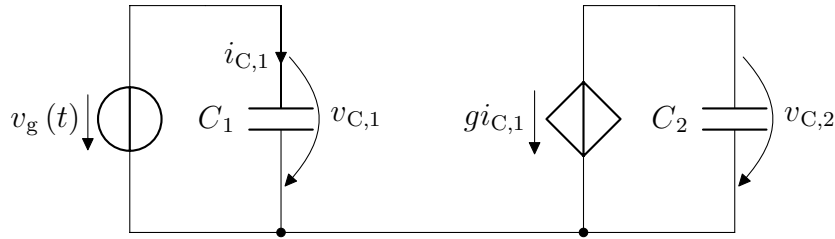


Abbildung 2.20.: Netzwerk mit stromgesteuerter Spannungsquelle und sogenannten versteckten Bedingungen ($v_g = 1 \text{ V} \sin(\omega t)$, $\omega = 1 \frac{\text{rad}}{\text{s}}$, $C_1 = 1 \text{ F}$, $C_2 = \text{beliebig}$, $g = 1 \Omega$)

Anfangswerte können demnach auch von Ableitungen anderer Netzwerkgrößen abhängen, den sogenannten versteckten Bedingungen. Die beschriebenen Schwierigkeiten im Zusammenhang mit der DC-Analyse wird in vielen Simulatoren, darunter auch SPICE, über zusätzliche eingeführte Hilfs widerstände [7] zu lösen versucht. Allerdings gelingt dies nur in einigen Fällen und löst nicht die Problematik der versteckten Bedingungen.

Ein weiterer, weit verbreiteter Ansatz zur Bestimmung konsistenter Anfangswerte ist die Verwendung des impliziten Eulerverfahrens³⁶ (s. Abschnitt 2.2.2). Alle Anfangswerte werden, wenn nicht anders vom Anwender vorgegeben, zu null gesetzt. Am Beispiel einer einfachen Reihenschaltung aus Spannungsquelle und Kapazität (s. Abb. 2.21) soll gezeigt werden, wie das implizite Eulerverfahren das Problem der inkonsistenten Anfangswerte löst.

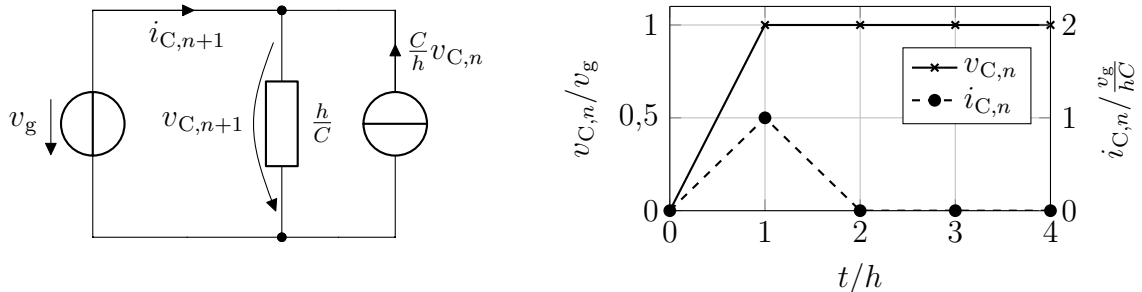


Abbildung 2.21.: Reihenschaltung aus Spannungsquelle mit $v_g \neq 0 \text{ V}$ und Ersatzschaltung einer Kapazität des „companion model“ des impliziten Eulerverfahrens mit inkonsistentem Anfangswert $v_{C,0} = 0 \text{ V}$ (links) sowie zugehöriger Strom- und Spannungsverlauf (rechts)

Unabhängig davon, welcher Startwert für $v_{C,0}$ gewählt wird, entspricht die Spannung über der Kapazität nach dem ersten Zeitschritt dem Wert der Spannungsquelle $v_{C,1} = v_g$. Auch kompliziertere Schaltungen, wie die aus zwei parallel verschalteten Kapazitäten mit unter-

³⁶ SPICE wendet das implizite Eulerverfahren zur Bestimmung der Anfangswerte an, wenn das Attribut „uic“ gesetzt ist [7].

schiedlichen Anfangswerten (s. Abb. 2.22), können mittels dem impliziten Eulerverfahren gelöst werden.

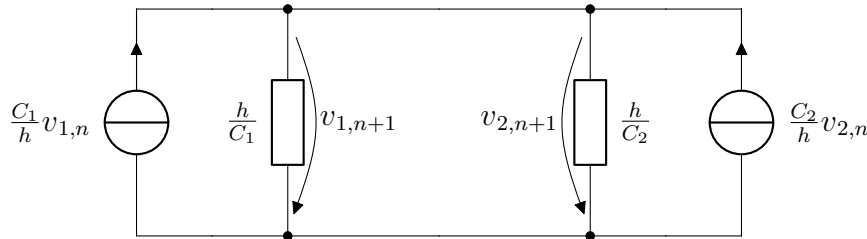


Abbildung 2.22.: Parallelschaltung zweier Kapazitäts-Ersatzschaltungen des impliziten Eulerverfahrens („companion model“) mit unterschiedlichen Anfangswerten $v_{1,0} \neq v_{2,0}$

Eine Analyse der Schaltung nach Abb. 2.22 liefert nach dem ersten Zeitschritt ($n = 0$ einsetzen)

$$v_{1,1} = v_{2,1} = \frac{\frac{h}{C_1} \frac{h}{C_2}}{\frac{h}{C_1} + \frac{h}{C_2}} \left[\frac{C_1}{h} v_{1,0} + \frac{C_2}{h} v_{2,0} \right] = \frac{C_1 v_{1,0} + C_2 v_{2,0}}{C_1 + C_2}. \quad (2.55)$$

Dieses Ergebnis ist nicht nur im Sinne der Kirchhoff'schen Maschenregel korrekt, sondern erfüllt auch das Gesetz der Ladungserhaltung. So entspricht die Gesamtladung auf beiden Kapazitäten zum Zeitpunkt $t = h$ der Summe der Einzelladungen zum Zeitpunkt $t = 0$. Das implizite Eulerverfahren ermöglicht zudem die Problematik der versteckten Bedingungen nach Abb. 2.20 zu lösen [111]. Diese vorteilhaften Eigenschaften des Eulerverfahrens nutzen viele der auf SPICE basierenden Simulatoren sowohl zur Bestimmung der Anfangswerte als auch für das Überschreiten sogenannter Unstetigkeiten (s. Kapitel 2.2.6) [112,113]. Sobald allerdings diese Unstetigkeiten überwunden sind, werden aufgrund der größeren Schrittweiten typischerweise Integrationsverfahren höherer Ordnung, wie etwa die Trapezregel, angewendet [7, 114].

Der Nachteil des Eulerverfahrens zum Auffinden konsistenter Anfangswerte liegt darin, dass die konsistenten Anfangswerte erst nach einer gewissen Anzahl an Zeitschritten zur Verfügung stehen [111]. Der eigentlich unendliche schnelle Umladevorgang wird verzögert und bedämpft [7]. Das bedeutet, dass sich die Lösung der regulären transienten Simulation und die Lösung des eigentlich auf den Zeitpunkt $t = 0$ begrenzten, unendlich schnell stattfindenden Ausgleichsvorgangs überlagern. Um diese Überlagerung rückgängig zu machen wird in [111] vorgeschlagen, die vorwärts gerichteten Zeitschritte durch die gleiche Anzahl an rückwärts gerichteten Schritten (mit negativem Zeitschritt $-h$) zu ergänzen. Das Ergebnis ist dann frei vom regulären transienten Zeitverlauf und entspricht den konsistenten Startwerten zum Zeitpunkt $t = 0^+$, also direkt nach dem impulsbehafteten Ausgleichsvorgang. Die transiente Simulation wird anschließend mit diesen ermittelten Anfangswerten neu gestartet. Mit Hilfe dieses Verfahrens lassen sich auch Anfangswerte

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

nach einem Schaltvorgang mit idealen Schaltern bestimmen [111]. Da ideale Schalter im eingeschalteten bzw. ausgeschalteten Zustand die Eigenschaften eines Kurzschlusses bzw. Leerlaufes besitzen, können innerhalb des unendlich schnell stattfindenden Schaltvorgangs impulsbehaftete Ausgleichsvorgänge stattfinden. Die Situation ist demnach vergleichbar mit dem Problem der inkonsistenten Anfangswerte. In [115] wird das Verfahren aus [111] durch eine symbolische Auswertung modifiziert, wodurch die Genauigkeit erhöht wird und auf die rückwärts gerichteten Schritte verzichtet werden kann.

Unter Verwendung der Zustandsraumdarstellung (s. Kapitel 2.2.1) kann auf das implizite Eulerverfahren gänzlich verzichtet werden. Auch die notwendige Trennung eines impulsiven und regulären Anteils ist nicht nötig, da die Integration nur vom Zeitpunkt $t = 0^-$ unmittelbar vor dem impulsbehafteten Ausgleichsvorgang bis zum Zeitpunkt $t = 0^+$ unmittelbar danach ausgeführt wird. Ein tatsächlicher Integrationsschritt von endlicher, nicht nullwertiger Dauer, wie im Fall des impliziten Eulerverfahrens, findet nicht statt. Ausschließlich impulsbehaftete Größen liefern einen Beitrag zum Integral. Um die Vorgehensweise besser nachvollziehen zu können, wird eine einfache Parallelschaltung zweier Kondensatoren mit unterschiedlichen Anfangswerten als Beispiel betrachtet. In einem ersten Schritt wird die widersprüchliche Situation der ungleichen Anfangswerte durch Einführung eines virtuellen, idealen Schalters³⁷ aufgehoben (s. Abb. 2.23).

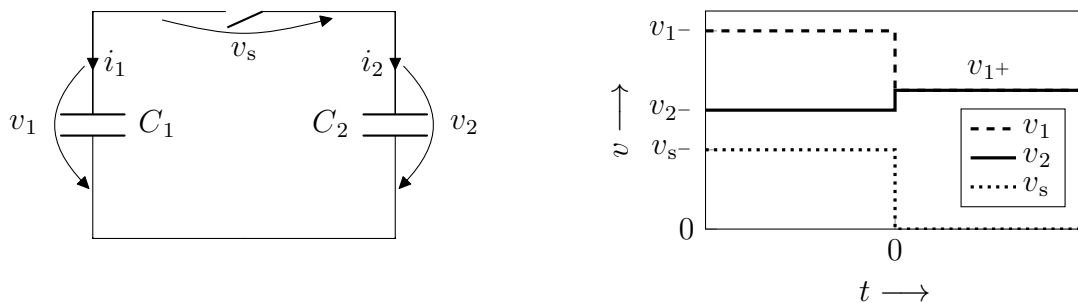


Abbildung 2.23.: Kurzschließen zweier Kapazitäten unterschiedlichen Anfangswerts mit idealem Schalter zum Zeitpunkt $t = 0$ (links) sowie zugehörigen Spannungsverläufen (rechts)

Ein Maschenumlauf ergibt die Schalterspannung zum Zeitpunkt vor dem impulsbehafteten Ausgleichsvorgang

$$v_{s,0^-} = v_{1,0^-} - v_{2,0^-}. \quad (2.56)$$

Um die eigentliche Ausgangssituation wiederherzustellen, wird der Schalter im nächsten Schritt geschlossen. Seine Spannung im eingeschalteten Zustand beträgt

$$v_{s,0^+} = 0. \quad (2.57)$$

³⁷ Der Ansatz mit Hilfe virtueller, idealer Schalter geht zurück auf [116].

Der Schalter kann durch eine Spannungsquelle mit sprungförmigem Zeitverlauf³⁸ ersetzt werden (vgl. Abb. 2.23), ohne dass sich an der Lösung etwas ändert.

$$v_s(t) = [v_{1,0^-} - v_{2,0^-}][1 - \sigma(t)]. \quad (2.58)$$

Unter Anwendung der Knoten- und Maschenregeln lässt sich die Ableitung der Spannung über der Kapazität C_1 in Abhängigkeit der Ableitung der Schalterspannungsquelle angeben

$$\dot{v}_1 = \frac{C_2}{C_1 + C_2} \dot{v}_s. \quad (2.59)$$

Obwohl die Sprungfunktion nicht im klassischen Sinne differenzierbar ist, so kann mit Hilfe der Theorie der Distributionen [117] eine Ableitung angegeben werden:

$$\dot{\sigma}(t) = \delta(t). \quad (2.60)$$

Integrieren beider Seiten in (2.59) vom Zeitpunkt unmittelbar vor dem Sprung bis unmittelbar nach dem Sprung liefert schließlich einen konsistenten Anfangswert

$$\begin{aligned} v_{1,0^+} &= \frac{C_2}{C_1 + C_2} \int_{t_0^-}^{t_0^+} \dot{v}_s(\tau) d\tau + v_{1,0^-} \\ &= \frac{C_2}{C_1 + C_2} \int_{t_0^-}^{t_0^+} [v_{2,0^-} - v_{1,0^-}] \delta(\tau) d\tau + v_{1,0^-} \\ &= \frac{C_1 v_{1,0^-} + C_2 v_{2,0^-}}{C_1 + C_2}. \end{aligned} \quad (2.61)$$

Der konsistente Anfangswert der zweiten Kapazität $v_{2,0^+}$ ist aufgrund der Parallelschaltung mit $v_{1,0^+}$ identisch.

Der hier zur Bestimmung von konsistenten Anfangswerten eingeführte virtuelle Schalter dient lediglich als Hilfskonstrukt. Besitzt eine Schaltungen hingegen tatsächlich ideale Schalter, so kann es, wie im Fall des virtuellen Schalters, ebenfalls zu unendlich schnellen Ausgleichsvorgängen kommen. Der hier vorgestellte Ansatz kann auch im Zusammenhang mit tatsächlich vorhandenen idealen Schaltern zur Bestimmung der neuen Anfangswerte angewendet werden [84].

2.2.5. Lösung des Gleichungssystems

Das Lösen linearer algebraischer Gleichungssysteme stellt, neben deren Formulierung, in vielen transienten Schaltungssimulatoren die rechenintensivste Aufgabe innerhalb des Verarbeitungsschritts dar [33, 113]. Dank numerischer Integrationsverfahren (s. Kapitel 2.2.2) und der lokalen linearen Approximation (s. Kapitel 2.2.3) lässt sich das Problem der

³⁸ Die Sprungfunktion $\sigma(t)$ ist definiert als $\sigma(t) = 0$ für $t < 0$ und $\sigma(t) = 1$ für $0 \leq t$ [117].

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

transienten nichtlinearen Schaltungssimulation auf wiederholtes Lösen linearer algebraischer Gleichungssysteme reduzieren. Wird zur Formulierung des Gleichungssystems die modifizierte Knotenpotentialanalyse oder das Tableauverfahren verwendet, so besitzt die zum Gleichungssystem zugehörige Matrix typischerweise eine sehr geringe Dichte an nicht nullwertigen Einträgen³⁹. Ein effizienter Umgang mit solch dünnbesetzten Matrizen kann die benötigte Gesamtrechendauer erheblich beeinflussen. So besitzen Lösungsalgorithmen für vollbesetzte Matrizen typischerweise einen Zeitaufwand, der proportional zur dritten Potenz der Matrizenordnung⁴⁰ n ist. Spezielle Verfahren für dünnbesetzte Matrizen erreichen hingegen einen Zeitaufwand der proportional zu $n^{1,1}$ bis $n^{1,5}$ ist [33]. Zusätzlich bieten diese Verfahren einen erheblichen Vorteil bezüglich des Speicherbedarfs, da auf das Speichern nullwertiger Einträge verzichtet werden kann. Im weiteren Verlauf dieser Arbeit besitzt das Lösen von Gleichungssystemen mit großen, dünnbesetzten Matrizen jedoch keine Relevanz. Daher sei an dieser Stelle auf weiterführender Literatur verwiesen. Eine gute Einführung gibt [4].

2.2.6. Unstetigkeiten

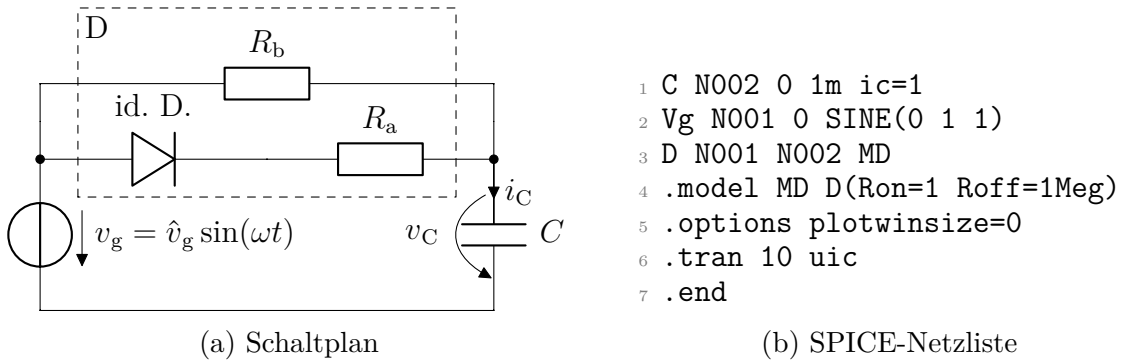
Unstetigkeiten in Strom- und Spannungsverläufen dynamischer Netzwerke stellen für transiente Schaltungssimulatoren eine besondere Herausforderung dar. Leistungselektronische Schaltungen beinhalten eine Vielzahl an möglichen Ursachen für unstetige Zeitverläufe: Sei es das Öffnen und Schließen eines Schalters, eine getaktete Quelle oder ein Schaltungselement mit stückweise linearisierter Kennlinie (s. Kapitel 2.2.3). Ein numerisches Integrationsverfahren muss bei der Wahl des Zeitschritts die Zeitpunkte solcher Unstetigkeiten berücksichtigen. Andernfalls kann dies zu groben Fehlern im Simulationsergebnis führen [118,119]. Viele Simulationsprogramme, darunter auch SPICE, zwingen das Integrationsverfahren durch Einführung sogenannter „Breakpoints“ den Zeitschritt so zu wählen, dass die Unstetigkeit exakt zum Zeitpunkt ihres Auftretens erfasst werden kann. Dadurch wird sichergestellt, dass Ereignisse, die Unstetigkeiten hervorrufen, detektiert und mit höherer zeitlicher Auflösung dargestellt werden [7]. Die Zeitpunkte dieser „Breakpoints“ lassen sich leicht aus den Zeitverläufen unabhängiger Strom- und Spannungsquellen bestimmen. Typischerweise gibt der Anwender selbst die Zeitpunkte dieser Unstetigkeiten im Quellverlauf zu Beginn der Simulation vor⁴¹. Umgekehrt gibt es jedoch auch Ereignisse, deren Zeitpunkte nicht derart einfach bestimmt werden können. Ein Beispiel dafür ist das Leiten bzw. Sperren einer mittels zwei oder mehr Segmenten stückweise linearisierten Diode. Hier geben die internen Größen (Diodenspannung und -strom) vor, wann die Leit- bzw. Sperrbedingung erfüllt ist. Zu welchem Zeitpunkt diese Ereignisse auftreten ist im

³⁹ Für sehr große Schaltungen liegt die Dichte der nicht nullwertigen Einträge der MNA-Matrix typischerweise im ppm Bereich [113]. Doch selbst für kleine Schaltungen mit einigen wenigen Knoten beträgt die Dichte typischerweise weniger als 50 % [4].

⁴⁰ Die Ordnung einer quadratischen Matrix bezeichnet die Anzahl der Zeilen bzw. Spalten.

⁴¹ Der Startzeitpunkt einer transienten Simulation $t = 0$ stellt ein Sonderfall innerhalb der Unstetigkeiten dar. Er wird gesondert in Kapitel 2.2.4 behandelt.

Allgemeinen nicht bekannt. Eine wichtige Rolle spielt dabei die Wahl des Zeitschritts. Ist er zu groß gewählt, so können Ereignisse, die nicht über „Breakpoints“ erfasst sind, leicht übersehen werden. Die Schaltung eines Spitzenwertdetektors nach Abb. 2.24a soll als anschauliches Beispiel dienen.



(a) Schaltplan

(b) SPICE-Netzliste

Abbildung 2.24.: Spitzenwertdetektor an Spannungsquelle („id. D.“ steht für ideale Diode, $R_{\text{off}} = R_b$, $R_{\text{on}} = \frac{R_a R_b}{R_a + R_b}$)

Sobald die Quellspannung v_g die Kapazitätsspannung v_C überschreitet, beginnt die Diode zu leiten. Im umgekehrten Fall wird die Diode in Sperrichtung betrieben und die Kapazität entlädt sich über den Widerstand R_b . Eine transiente Simulation der Netzliste aus Abb. 2.24b in LTspice IV⁴² (Version 4.21i) liefert das in Abb. 2.25 dargestellte Ergebnis.

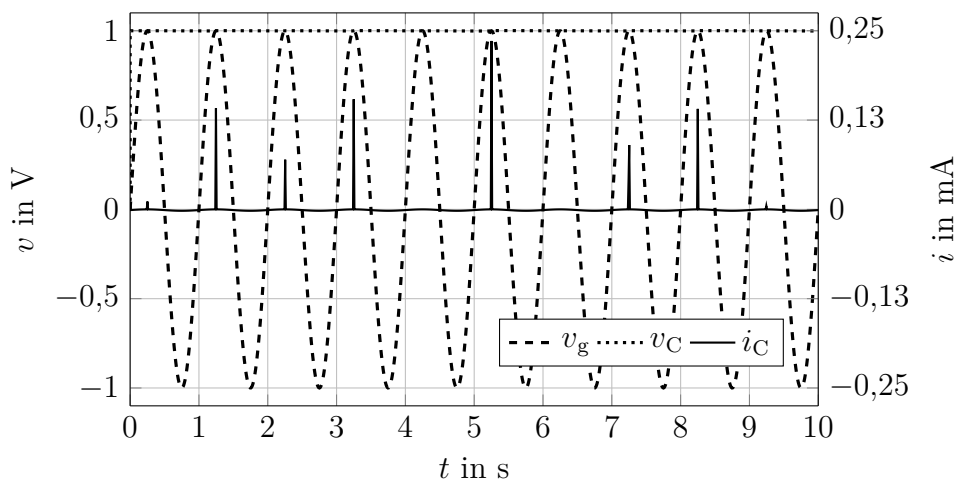


Abbildung 2.25.: LTspice IV (Version 4.21i) Simulationsergebnis der Netzliste aus Abb. 2.24b

Anhand des Kapazitätsstroms i_C ist gut erkennbar, dass die Kapazität, entgegen der

⁴² Wenn nichts anderes angegeben ist, so wird das Programm stets mit den Simulationsparametern (Toleranzen, Integrationsverfahren,...) im Auslieferungszustand betrieben.

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

Erwartung, nicht periodisch um den Zeitpunkt des positiven Scheitelwerts der Sinus-schwingung geladen wird. Zum Zeitpunkt $t = 4,25$ s fehlt die Leitphase der Diode gänzlich. Die Ursache liegt darin, dass SPICE keine „Breakpoints“ an den Scheitelpunkten der Schwingung setzt. Dadurch kann es, wie in Abb. 2.25 geschehen, zum Überspringen von Unstetigkeiten kommen. Abb. 2.26 stellt den entsprechenden Ausschnitt der transienten Simulation nochmals vergrößert dar. SPICE berechnet die Zeitverläufe ausschließlich zu den diskreten Zeitpunkten, die in Abb. 2.26 mit einem Marker hervorgehoben sind.

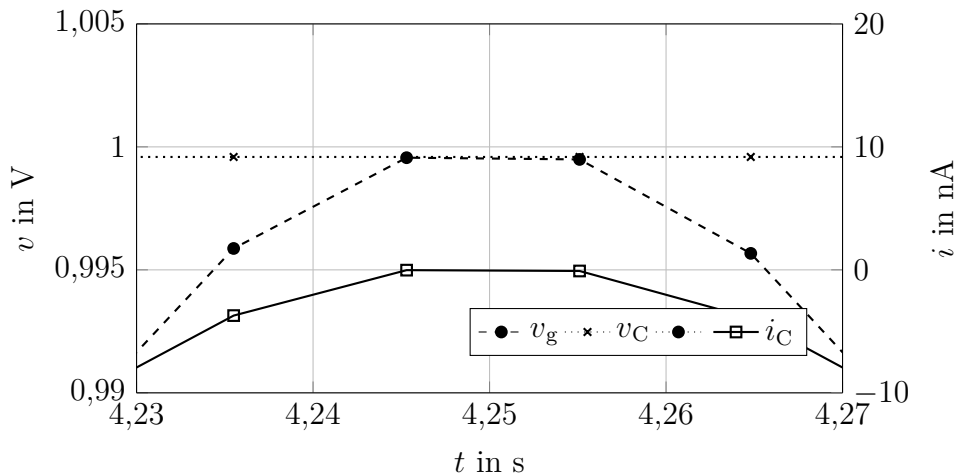


Abbildung 2.26.: Ausschnitt aus dem Simulationsergebnis in Abb. 2.25

Die korrekte Erfassung von Unstetigkeiten ist in vielen Simulationsprogrammen, darunter auch SPICE, nur unzureichend realisiert [7]. So liegt es häufig in der Zuständigkeit des Anwenders, die Simulationsparameter so anzupassen, dass ein korrektes Ergebnis entsteht. Im vorliegenden Fall des einfachen Spitzenwertdetektors kann der erfahrene Anwender ein falsches Simulationsergebnis noch leicht identifizieren. Umfangreichere Schaltungen stellen hingegen deutlich höhere Ansprüche an die Erfahrung und Vorkenntnis des Anwenders.

Im Gegensatz zu Simulationsprogrammen wie SPICE, die keine zuverlässige Detektion aller Unstetigkeiten garantieren können, wird in [120, 121] ein Verfahren vorgestellt, das eine sichere Detektion zulässt. Statt lediglich die Momentanwerte eines jeden Zeitschritts auszuwerten, wie in SPICE üblich, berücksichtigt das Verfahren den dazwischenliegenden Zeitverlauf, indem es die Strom- und Spannungsverläufe als Polynom über der Zeit beschreibt⁴³. Wird der Zeitschritt nur klein genug aufgelöst oder die Polynomordnung groß genug gewählt, so kann das Polynom den exakten Zeitverlauf im Rahmen der Maschinengenauigkeit⁴⁴ wiedergeben [4]. Die notwendigen Polynomkoeffizienten können auf sehr

⁴³ Das in SPICE häufig verwendete Trapezverfahren basiert darauf, dass die Lösung durch ein Polynom zweiten Grades angenähert werden kann [4]. Allerdings verzichtet SPICE darauf, den Extremwert dieses Polynoms explizit zu bestimmen, um so eine Unstetigkeit besser erkennen zu können.

⁴⁴ Die Maschinengenauigkeit ist ein Maß für den Rundungsfehler, der aufgrund der begrenzten, rechnerinterne Zahlendarstellung auftritt.

unterschiedliche Weise gewonnen werden, ein Beispiel gibt [122]. Dank des Polynoms ist nun nicht mehr nur der Strom- bzw. Spannungswert zu diskreten Zeitpunkten sondern auch dazwischen bekannt. Verlässt der Wertebereich des Polynoms innerhalb eines Zeitschritts die Segmentgrenzen eines als stückweise linear definierten Schaltungselements, so kann der Zeitpunkt des Übergangs zwischen Segmenten durch eine Nullstellensuche des Polynoms ermittelt werden. Um einen möglichst großen Zeitschritt überbrücken zu können, werden nach dem Verfahren in [120,121] allerdings Polynome mit Ordnungen im Bereich von acht verwendet. Eine Auswertung der Nullstellen ist demnach nur mit Hilfe numerischer Verfahren möglich. Um zu überprüfen, ob innerhalb eines Zeitschritts eine Unstetigkeit auftreten kann, reicht es zunächst aus, den Wertebereich des Polynoms im vorgegebenen Zeitintervall abzuschätzen. Dabei sollte es sich jedoch um eine konservative Abschätzung handeln, d. h. der exakte Wertebereich wird stets eingeschlossen, andernfalls wäre das Verfahren nicht zuverlässig. In [120,121] wird zur Abschätzung des Wertebereichs sowie der Nullstellensuche die Intervallarithmetik [117] angewendet. Ein Nachteil dieses zwar zuverlässigen jedoch aufwendigen Verfahrens ist der erhöhte Rechenzeitbedarf [123]. Andere Ansätze, wie die in [53,78] vorgestellten, verwenden zwar ebenfalls eine polynomiale Näherung der Zeitverläufe, nutzen diese allerdings nicht für eine zuverlässige Detektion von Unstetigkeiten.

2.3. Ausgabe

Der letzte Teilbereich eines Programms zur rechnergestützten Schaltungsanalyse ist die Ausgabe der Simulationsergebnisse. Die beiden gebräuchlichsten Ausgabeformate sind die grafische Ausgabe, bspw. in Form eines Strom/Spannungs-Zeit-Diagramms, sowie die textbasierte Ausgabe in Form eines geeigneten Datenformats, bspw. einer Matrix. Einige Simulationsprogramme, wie etwa LTspice, bieten zusätzlich eine verlustbehaftete Datenkompression an. Angesichts der enormen Datenmenge, die während einer transienten Simulation entstehen kann, erscheint dies sinnvoll. Alternative Simulationsprogramme, die eine Formulierung in Zustandsraumdarstellung und die stückweise lineare Approximation anwenden, erlauben ein wesentlich effizienteres Speichern der Simulationsergebnisse. So ist es bei stückweise linearen Netzwerken ausreichend, die Zeitpunkte eines Segmentwechsels („Breakpoints“) mitsamt dem zugehörigen Zustandsvektor sowie die Matrizen⁴⁵ der Zustandsraumdarstellung abzuspeichern [78]. Anhand dieser stark reduzierten Datenmenge lässt sich das Simulationsergebnis in einem „post-processing“ Schritt mit jeder beliebigen zeitlichen Auflösung schnell und effizient wiederherstellen. Auch eine Berechnung charakteristischer Kennwerte wie Mittel- und Effektivwert ist anhand dieser Daten mit sehr hoher Genauigkeit möglich [124].

⁴⁵ Sich wiederholende Matrizen müssen selbstverständlich nur ein einziges Mal abgespeichert werden.

2.4. Zusammenfassung und Diskussion

Die vorhergehenden Abschnitte geben eine Übersicht über die Grundlagen der rechnergestützten transienten Schaltungsanalyse. Aufgrund der Popularität des Schaltungssimulators SPICE ist dessen Arbeitsweise in den einzelnen Teilkapiteln stets berücksichtigt und in seinen Grundzügen erläutert. Das Flussdiagramm in Abb. 2.27 zeigt zusammenfassend den Ablauf einer nichtlinearen, transienten Schaltungsanalyse in SPICE.

Neben der Arbeitsweise von SPICE bzw. SPICE-ähnlichen Simulatoren werden zudem Ansätze präsentiert, die sich insbesondere für leistungselektronische Schaltungen eignen. Der direkte Vergleich zeigt, dass mit alternativen Ansätze in vielen der Teilbereichen Vorteile im Hinblick auf Rechenzeit, Genauigkeit und/oder Zuverlässigkeit der Simulationsergebnisse ggü. SPICE erzielt werden können. Ziel dieser Arbeit soll es sein, die für leistungselektronische Schaltungen erfolgversprechendsten Ansätze zu einem Simulationsprogramm zu vereinigen⁴⁶ und, falls erforderlich, durch eigene Ansätze zu ergänzen. Im Folgenden werden die Teilbereiche der Schaltungsanalyse nochmals aufgezählt und die zuvor präsentierten Ansätze diskutiert.

⁴⁶ Wie in vielen anderen Bereichen gilt auch hier: Um ein optimales Gesamtergebnis zu erzielen, muss die Auswahl der erfolgversprechendsten Ansätze in gegenseitiger Abwägung getroffen werden [29].

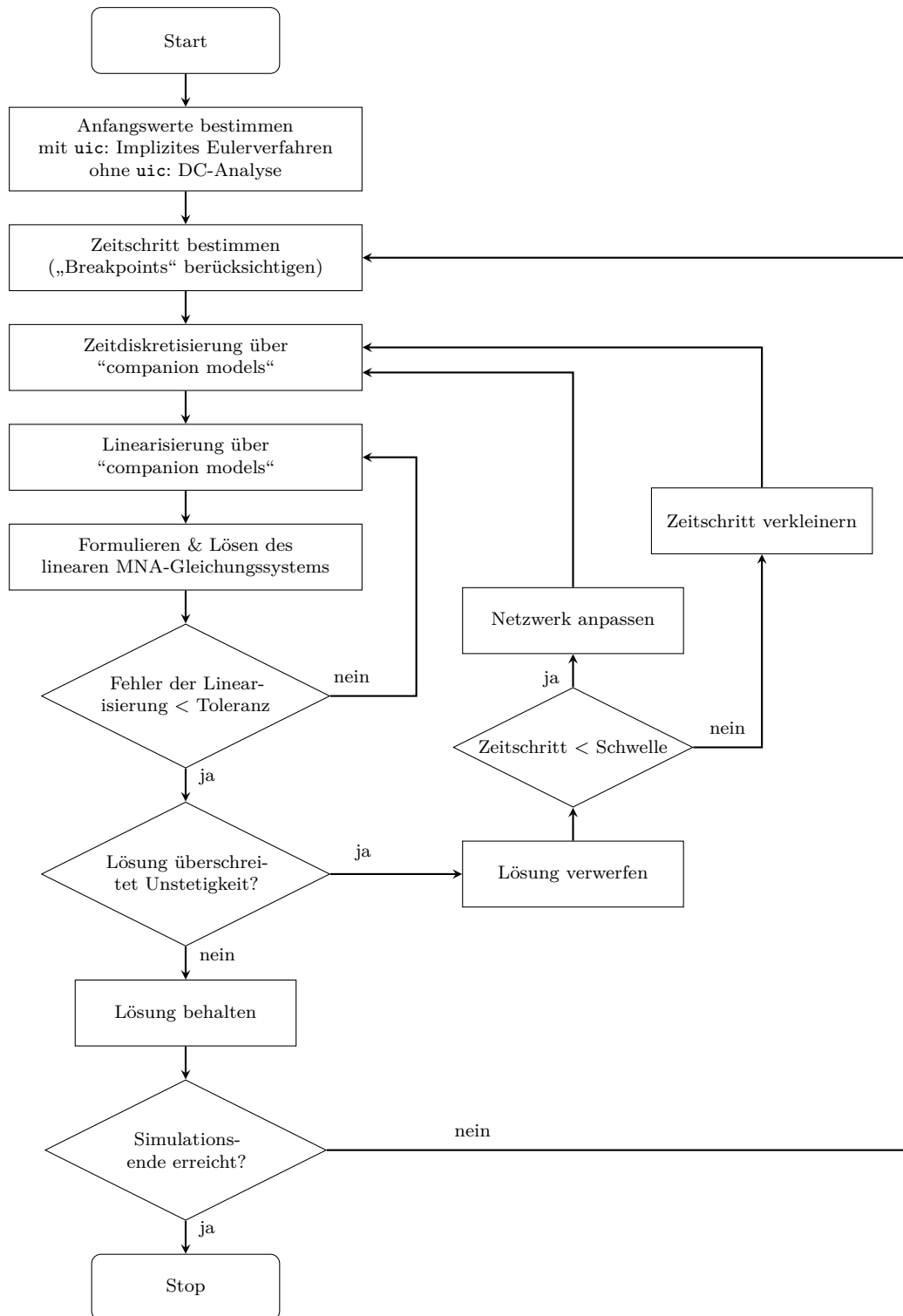


Abbildung 2.27.: Transiente Schaltungsanalyse (ohne Ein- und Ausgabe und ohne Berücksichtigung eventueller Konvergenzprobleme) in SPICE

2.4.1. Eingabe

In der Praxis haben sich beide Formen der Eingabe, also die grafische als auch die textbasierte Eingabe (bspw. über eine SPICE-Netzliste), bewährt. Beide besitzen ihre Vor- und Nachteile. Das Simulationsprogramm, welches im Rahmen der vorliegenden Arbeit erstellt wird, verzichtet auf eine eigene grafische Benutzeroberfläche und akzeptiert stattdessen SPICE-Netzlisten. Bei Bedarf können solche Netzlisten jedoch mit einem der vielen frei erhältlichen Simulationsprogramme⁴⁷ in grafischer Weise erstellt werden.

2.4.2. Formulieren des Gleichungssystems

Bereits in den Anfangsjahren der rechnergestützten Schaltungssimulation löste die Frage nach der geeignetsten Formulierung des Gleichungssystems heftige Debatten aus [29]. Viele der Kernpunkte dieses Streits verlieren jedoch an Bedeutung, wenn das Hauptanwendungsgebiet des Schaltungssimulators auf getaktete, leistungselektronische Schaltungen beschränkt wird. Der Umfang solcher Schaltungen reicht in den wenigsten Fällen über eine niedrige zweistellige Anzahl an Schaltungselementen hinaus [76]. Die Grundlage des damaligen Streits waren jedoch Schaltungen mit einer Elementanzahl im vierstelligen Bereich und darüber hinaus (sogenannte „kilo-branch circuits“) [56]. Dies wiederum hat zur Folge, dass viele der damaligen Kritikpunkte an der Formulierung im Zustandsraum (SSA) ihre Berechtigung verlieren. Nachteile wie:

- Aufwendige und problematische Reduzierung vom DAE zum ODE [39, 57, 75, 125];
- Keine dünnbesetzte Matrix und dadurch keine Vorteile bei der Anwendung von „Sparse-Matrix“-Lösern [21, 57, 58, 125–127];
- Kann nur in Sonderfällen explizit, also mittels topologischer Methoden, aufgestellt werden [60].

sind hauptsächlich nur dann von Bedeutung, wenn es sich um umfangreiche Schaltungen mit einer Vielzahl an dynamischen Bauelementen (z. B. parasitärer Kapazitäten [21]) handelt. In anderen Fällen sind entsprechende Behauptungen nicht genügend begründet, nicht ausreichend differenziert oder basieren auf falschen Annahmen. So ist in den angegebenen Literaturstellen zu lesen:

1. Die SSA benötige eine Anwendung der Graphentheorie und einen sogenannten Normalbaum [57, 125];
2. Keine eindeutige und teilweise wechselhafte Zuordnung von Netzwerkgrößen zu Zustandsvariablen [39];
3. Vorteile des SSA seien nicht mehr relevant, da es mittlerweile Lösungsverfahren für differential-algebraische Gleichungssysteme gibt [57, 125];

⁴⁷ Im Rahmen der Arbeit wird das Programm LTspice von Linear Technology, welches unter www.linear.com bezogen werden kann, verwendet.

4. Das SSA verursache ein Auftreten der zeitlichen Ableitung von Quellenfunktionen [57, 125].

Dem kann entgegnet werden:

1. Die Zustandsraumdarstellung kann auch indirekt über die Darstellung mit Hilfe des MNA und STA gewonnen werden (s. Kapitel 2.2.1). Nur direkte, explizite Formulierungsmethoden stellen Einschränkungen an das Netzwerk.
2. Die Zuordnung von Netzwerkgrößen zu Zustandsvariablen ist ein internes Detail des Simulationsprogramms. Es handelt sich hier also eher um eine Herausforderung bezüglich des Softwaredesigns. Für den Anwender eines Simulationsprogramms spielt dies allerdings keine Rolle. Seine Prioritäten liegen eher auf Rechendauer, Zuverlässigkeit und die Genauigkeit der Lösung.
3. Die effiziente Lösung steifer Systeme benötigt ein implizites Integrationsverfahren mit variabler Schrittweite. Deren Anwendung ist jedoch keineswegs trivial und bedarf einiger Vorsicht, da durch ungeeignete Schrittwahl und Unstetigkeiten Instabilität auftreten kann [75].
4. Zeitliche Ableitungen von Quellenfunktionen sind keine Folge des SSA. Sie sind eine immanente Eigenschaft der Schaltung.

Grundsätzlich spricht demnach nichts gegen eine Anwendung der Formulierung im Zustandsraum. Allerdings soll die in dieser Arbeit verwendete Formulierungsmethode die kleinst möglichen Anforderungen an das Netzwerk stellen, wodurch die expliziten, topologischen Methoden (s. Abschnitt 2.2.1) ausgeschlossen werden [60].

2.4.3. Dynamische Schaltungselemente und numerische Integration

Der große Vorteil einer Formulierung im Zustandsraum besteht in der deutlich einfacheren und effizienteren Lösung eines gewöhnlichen Differentialgleichungssystems (ODE) im Vergleich zu einem Differential-algebraischen Gleichungssystem (DAE) [58, 76]. Die Formulierung als DAE besitzt hingegen Vorteile im Bereich der Entwicklung integrierter Schaltkreise [21]. Die dort auftretende enorm hohe Anzahl an dynamischen (nichtlinearen) Schaltungselementen ist keine gute Voraussetzung zur Formulierung als ODE. Eine Formulierung im Zustandsraum hingegen ermöglicht die Anwendung eines schnellen, expliziten Integrationsverfahrens wie der Matrixexponentialfunktion [21, 41, 75]. Es ist schwierig Unterschiede in Bezug auf Genauigkeit und Geschwindigkeit zwischen impliziten Integrationsverfahren und der Matrixexponentialfunktion zu quantifizieren, da diese stark von der Schaltung abhängig sind. Dennoch zeigt sich, dass Verfahren auf Basis der Matrixexponentialfunktion (bei gleicher Genauigkeit) um den Faktor 10-100 weniger Rechenzeit als andere numerische Integrationsverfahren aufweisen [75, 91, 96, 99]. Der Vorteil vergrößert sich, je höher die Ansprüche an die Genauigkeit [25, 91] sind. Zudem erlauben Methoden wie die Eigenwertzerlegung einen deutlichen tieferen Einblick in das Systemverhalten [90],

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

was letztlich zur Entwicklung effizienterer Lösungsansätze beitragen kann.

2.4.4. Nichtlineare Schaltungselemente

Das vorhergehenden Teilkapitel verdeutlicht, dass die numerische Integration eines stückweise linearen Systems auf Basis der Matrixexponentialfunktion äußerst effizient ist. Daher soll im Rahmen dieser Arbeit die stückweise lineare Approximation der lokalen Linearisierung vorgezogen werden. Die Anwendung der in Kapitel 2.2.3 erwähnten, exponentiellen Integratoren besitzt gegenüber einer stückweise linearen Approximation Vorteile, wenn es sich um eine Nichtlinearität handelt, die nur schlecht über lineare Segmente genähert werden kann und ein hohes Maß an Genauigkeit gefordert ist. Je höher die geforderte Genauigkeit, desto mehr Segmente sind notwendig, um die nichtlineare Kennlinie zu diskretisieren. Die Folge davon sind häufige Segmentwechsel mit einer Vielzahl an notwendigen Formulierungen des sich ändernden Gleichungssystems. Der gewählte Ansatz aus SSA, Matrixexponentialfunktion und stückweise linearer Approximation verliert dadurch zunehmend an Vorteil. Die vorliegende Arbeit setzt jedoch mit Blick auf die Analyse leistungselektronischer Schaltungen voraus, dass es für die Gesamtgenauigkeit ausreichend ist, sämtliche nichtlinearen Schaltungselemente mit einer geringen Anzahl an Segmenten zu linearisieren.

2.4.5. Anfangswerte

Je nach Schaltung kann die Bestimmung konsistenter Anfangswerte eine anspruchsvolle Aufgabe darstellen. Das in SPICE und vielen anderen Schaltungssimulatoren eingesetzte implizite Eulerverfahren bietet theoretisch die Möglichkeit, konsistente Anfangswerte auch in schwierigen Fällen zu bestimmen. Allerdings müssen bei der rechnergestützten Umsetzung einige kritische Punkte beachtet werden [111, 115]. Entgegen der Darstellung in [39, 111] stellt demnach die Berücksichtigung inkonsistenter Anfangswerte, die per Definition impulsbehaftete Ausgleichsvorgänge hervorrufen, auch für eine Formulierung mit dem modifizierten Knotenpotentialverfahren oder dem Tableauverfahren eine Herausforderung dar. In Zustandsraumdarstellung hingegen können impulsbehaftete Vorgänge mit Hilfe virtueller Hilfsschalter und der Modellierung des Ein- und Ausschaltvorgangs dieser Schalter als Sprungspannungs- bzw. Sprungstromquelle effizient und genau erfasst werden [84, 116]. Enthält das Netzwerk allerdings stückweise linearisierte Kapazitäten/Induktivitäten, so muss das Verfahren modifiziert werden. Näheres dazu befindet sich in Abschnitt 4.2.

2.4.6. Lösung des Gleichungssystems

Aufgrund der bereits in den vorhergehenden Unterkapiteln getroffenen Entscheidung der Formulierung in Zustandsraumdarstellung und Verwendung numerischer Integration auf Basis der Matrixexponentialfunktion spielt die effiziente Lösung eines dünnbesetzten, algebraischen Gleichungssystems in dieser Arbeit keine Rolle. Diejenigen algebraischen Umformungen, die im Zusammenhang mit der Formulierung im Zustandsraum nötig sind, besitzen typischerweise keinen signifikanten Einfluss auf die benötigte Gesamtrechendauer [4].

2.4.7. Unstetigkeiten

Ein berechtigter Kritikpunkt in Bezug auf den Einsatz des SSA zur transienten Simulation leistungselektronischer Schaltungen ist die Häufigkeit mit der Unstetigkeiten auftreten. Jede dieser Unstetigkeiten macht eine Neuformulierung des sich ändernden Gleichungssystems nötig, was im Falle des SSA aufwendiger als mit dem MNA und STA ist [52]. Allerdings können die einmal generierten Matrizen zur späteren Verwendung abgespeichert werden und müssen auf diese Weise nur ein einziges Mal innerhalb der transienten Simulation formuliert werden [18, 72]. Für Schaltungen mit einer Vielzahl an stückweise linearisierten Elementen sollten die Matrizen aller Kombinationen keinesfalls zu Beginn der Simulation bestimmt werden (wie in [39] suggeriert) sondern auf Anforderung („on demand“). Zudem hängt der Rechenaufwand der Neuformulierungen im Verhältnis zum Gesamtrechenaufwand stark von der jeweiligen Schaltung ab. An dieser Stelle eine generelle Aussage treffen zu wollen, ist demnach schwierig. Die vorliegende Arbeit geht von einer insgesamt eher geringen Anzahl an Segmenten aus, wie in der Leistungselektronik typischerweise üblich.

Das im Rahmen dieser Arbeit erstellte Simulationsprogramm erhebt den Anspruch, Unstetigkeiten effizient und zuverlässig zu detektieren. Ein Vorgehen, das auf alleinige Auswertung der Ströme und Spannungen zu diskreten Lösungszeitpunkten basiert (wie etwa in SPICE üblich), scheidet daher aus. Allerdings weisen die in Kapitel 2.2.6 vorgestellten, zuverlässigen Vorgehensweisen wiederum den Nachteil einer erhöhten Rechenzeit auf. Es ist daher ein Anliegen dieser Arbeit, diesen Nachteil zu umgehen und ein alternatives Verfahren zu entwickeln (s. Abschnitt 4.4).

2.4.8. Ausgabe und Aufbereitung der Simulationsergebnisse

Auch bei Ausgabe und Aufbereitung der Simulationsergebnisse hat die Formulierung mit Hilfe des SSA Vorteile. Die Zustandsraumdarstellung mitsamt den notwendigen Anfangswerten stellen per Definition das absolute Mindestmaß an Information bereit, aus dem sämtliche anderen Netzwerkgrößen zu jedem beliebigen Zeitpunkt effizient und genau

2. Grundlagen der rechnergestützten transienten Schaltungsanalyse

gewonnen werden können [59]. In vielen Fällen sind jedoch nicht die Strom- und Spannungsverläufe von Interesse, sondern Kennwerte wie Effektiv-, Mittel- oder Spitzenwert. Ein Abspeichern tausender von Zeitpunkten, um anschließend einige wenige Kennwerte daraus zu berechnen, ist daher eher ineffizient. Mit Hilfe des SSA können diese Kennwerte mit minimalem Speicheraufwand und hoher Genauigkeit berechnet werden. Eine Datenkompression ist daher nicht erforderlich.

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Die rechnergestützte Formulierung der Zustandsraumdarstellung eines elektrischen Netzwerks kann in drei Teilaufgaben gegliedert werden. Im ersten Teil werden die linear unabhängigen Maschen- und Knotengleichungen, die sich aus den Kirchhoff'schen Regeln ergeben, aufgestellt. Anschließend werden die Strom-/Spannungsbeziehungen der Bauelemente als weitere Gleichungen im bereits bestehenden Gleichungssystem berücksichtigt. Die letzte Teilaufgabe besteht aus der eigentlichen Formulierung des Differentialgleichungssystems¹ in Normalform. Jeder Teilschritt wird anhand eines Beispiels aus dem Bereich der getakteten, leistungselektronischen Schaltungen veranschaulicht. Die in diesem Kapitel beschriebene Vorgehensweise orientiert sich stark an [4, 18]. Sie erhebt keinen Anspruch auf höchste Speichereffizienz oder kürzeste Rechenzeit. Wird jedoch der Gesamtbedarf an Speicherplatz und Rechenzeit über den gesamten Simulationsvorgang, also von Eingabe bis Ausgabe, berücksichtigt, so relativiert sich der Aufwand zur Formulierung des Gleichungssystems. Er kann in den meisten Fällen sogar vernachlässigt werden².

Ausgangspunkt der rechnergestützten Schaltungsanalyse ist die Beschreibung der Schaltung als sogenannte SPICE-Netzliste. Darin werden sämtliche Schaltungselemente und deren Verschaltung (in der Literatur auch als Topologie bezeichnet) in lesbarer Textform definiert. Der Programmausdruck 3.1 dient als Beispiel und beschreibt die in Abb. 3.1 dargestellte Schaltung eines Hochsetz-Tiefsetzstellers, der auch als Buck-Boost-Wandler bezeichnet wird [9]. Die eigentlichen Informationen über das Netzwerk befinden sich in den Zeilen zwei bis acht. Jedes Netzwerkelement besitzt eine eigene Zeile. Das erste Zeichen jeder Zeile gibt Auskunft über den Elementtyp ($V \triangleq$ Spannungsquelle, $R \triangleq$ Widerstand, etc.). Eine Übersicht der unterstützten Elementtypen befindet sich in Anhang A.1. Alle weiteren Zeichen bis zum ersten Leerzeichen sind frei wählbar und dienen lediglich zur Unterscheidung mehrerer Elemente des gleichen Typs. Die Zeichenkette nach dem ersten Satz aus einem oder mehreren Leerzeichen gibt Auskunft, an welchem Knoten der positive Pol des Elements angebunden ist. Folglich gibt der Eintrag nach dem zweiten Satz an

¹ Im Falle, dass das Netzwerk Schalter und/oder Dioden enthält, entsteht für jede Schaltkombination ein eigenes Differentialgleichungssystem.

² “[...] the time required for finding T , \mathbf{D} , and \mathbf{B} [Baum, fundamentale Trennmengenmatrix und fundamentale Maschenmatrix, Anmerkung des Verfassers] is insignificant [...] as compared to the time required in the other segments of computer simulation programs, such as numerical integration, piecewise-linear iteration, etc.” ([4], S.154)

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Listing 3.1: Netzliste des Buck-Boost-Wandlers

```
1 * Buck-Boost-Wandler
2 Vg N001 0 10
3 S N001 N002 N004 0 Switch
4 L N002 0 100u
5 C N003 0 1m
6 R N003 0 10
7 Vp N004 0 PULSE(0 1 0 1n 1n 5u 10u)
8 D N003 N002 Diode
9 .model Switch SW(Ron=1m Roff=1Meg Vt=0.5)
10 .model Diode D(Ron=1m Roff=1Meg)
11 .tran 100m
12 .end
```

Leerzeichen den Knoten des negativen Pols an³. Besitzt das Element mehr als zwei Pole (wie etwa der spannungsgesteuerte Schalter S, der insgesamt vier Pole besitzt), so folgen nun die weiteren Knotennamen. Der jeweils letzte Eintrag einer Zeile beschreibt den Wert des jeweiligen Elements. Während bei rein passiven Elementen (bspw. R, L und C) ein einfacher numerischer Wert (*hier*: Widerstand in Ohm, Induktivität in Henry, Kapazität in Farad) genügt, müssen bei komplexeren Elementen, wie etwa einer Pulsquellen, sämtliche Quellenparameter (Amplitude, Frequenz, Offset etc.) übergeben werden. Bei einigen Elementen (wie etwa dem Schalter S) werden die zusätzlichen Parameter nicht in der gleichen Zeile übermittelt, sondern es muss ein Modellname angegeben werden (*hier*: **Switch**). Dieses Modell muss in einer anderen Zeile innerhalb der gleichen Netzliste definiert sein. Im Fall des Schalters geschieht dies in Zeile neun mit der „.model“-Anweisung. Die Reihenfolge der Netzlisteneinträge ist unerheblich, den Abschluss bildet stets die „.end“-Anweisung. Eine Übersicht über die unterstützten Anweisungen ist in Kapitel A.2 zu finden.

Beim vorliegenden Beispiel des Gleichspannungswandlers ist eine händische Aufstellung der Netzliste noch mit vertretbarem Aufwand möglich. Eine ausführliche Erklärung der dazu notwendigen Netzlisten-Syntax gibt [128]. Sollte die Schaltung allerdings umfangreicher werden, so ist es ratsam die Netzliste mittels einer grafischen Benutzeroberfläche automatisiert erstellen zu lassen. Ein geeignetes Programm ist bspw. LTspice IV von Linear Technology⁴.

³ Es wird ausschließlich das Verbraucherzählpfeilsystem [19] angewendet, auch bei Quellen. Die Zweigspannung und der Zweigstrom zeigen in Richtung vom positiven zum negativen Pol.

⁴ Das Programm einschließlich der Nutzungsbedingungen kann unter www.linear.com bezogen werden.

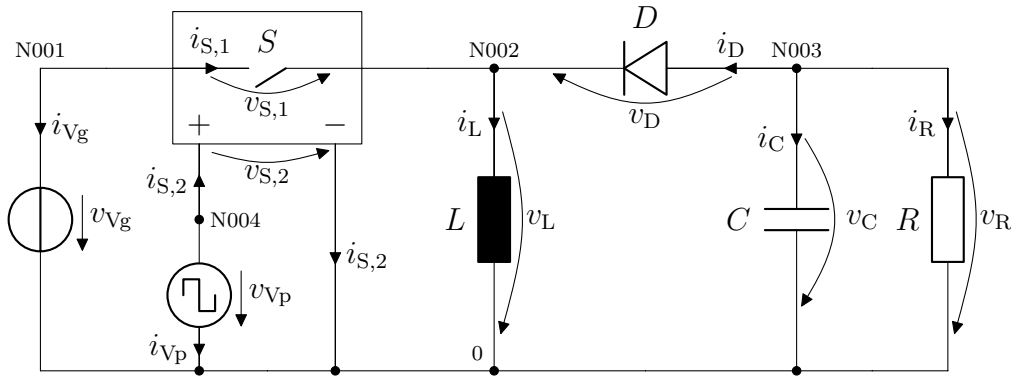


Abbildung 3.1.: Schaltplan des Buck-Boost-Wandlers mit Schalteransteuerung, bestehend aus Quelle V_p und Steuereingang $S,2$

3.1. Anwendung der Kirchhoff'schen Regeln

Die Kirchhoff'schen Regeln [19] besagen, dass

1. sich alle Ströme in einen Knoten und
2. alle Spannungen entlang einer Masche zu null addieren.

In einem zusammenhängenden, elektrischen Netzwerk mit z Zweigen⁵ und k Knoten⁶ ergibt die erste der beiden Regeln genau $k - 1$ linear unabhängige Gleichungen (Knotengleichungen), welche die Zusammenhänge zwischen den z Zweigströmen erfassen. Die zweite Regel liefert insgesamt $z - k + 1$ weitere, linear unabhängige Gleichungen (Maschengleichungen), die die Zusammenhänge zwischen den z Zweigspannungen beschreiben. Um nun alle z Zweigströme und z Zweigspannungen bestimmen zu können, fehlen allerdings noch z linear unabhängigen Gleichungen. Diese ergeben sich durch die Strom-/Spannungsbeziehungen der Netzwerkelemente bzw. direkt aus den Quellgrößen⁷.

⁵ Als Zweig wird hier ein aus einem Element bestehender Zweipol bezeichnet. Schaltungselemente mit mehr als zwei Polen (bspw. Schalter) müssen in mehrere Zweipole aufgeteilt werden. Zudem werden ausschließlich Netzwerke mit konzentrierten Elementen betrachtet, d. h. räumlich bedingte Laufzeiteffekte innerhalb eines Elements werden vernachlässigt. Wenn nicht anders angegeben, richten sich die verwendeten Begriffe der Graphentheorie nach der Norm [129].

⁶ Als Knoten werden hier die Verbindungsstellen zwischen zwei oder mehr Zweigen bezeichnet.

⁷ Dies gilt nur für lösbar Netzwerke. Unlösbar ist ein Netzwerk beispielsweise dann, wenn es Maschen besitzt, die ausschließlich aus unabhängigen Spannungsquellen bestehen. Besitzt das Netzwerk gesteuerte Quellen, so kann dies zu sogenannten versteckten Bedingungen ([4], S. 334) führen, die das Netzwerk unlösbar machen. Für weiterführende Informationen zur Thematik der Lösbarkeit sei auf [74] verwiesen.

3.1.1. Aufstellen der Knotengleichungen

Im folgenden Schritt werden die Informationen der Netzliste in ein für die weitere Verarbeitung geeignetes Format gebracht. Dafür eignet sich die sogenannte Inzidenzmatrix [4, 26]. Jedem der k Knoten wird eine Zeile und jedem der z Zweige eine Spalte der Matrix zugeordnet. Die Reihenfolge ist zunächst beliebig. Der positive und negative Pol eines jeden Zweigs werden durch einen Matrizeneintrag von 1 bzw. -1 gekennzeichnet. Alle anderen Einträge der Matrix werden zu null gesetzt. Eine zur Netzliste in Listing 3.1 passende Inzidenzmatrix zeigt (3.1). Die Zeilen und Spalten sind zusätzlich mit den Knoten- bzw. Zweignamen der Netzliste versehen. Die Spaltenreihenfolge ist hier so gewählt, dass sie mit der Reihenfolge der Netzlisteneinträge übereinstimmt. Die vier Knoten des Schalters S sind in zwei Paare, den eigentlichen Schalterzweig S,1 und den Zweig des Steuereingangs S,2, aufgeteilt. Da die SPICE-Netzliste bereits die prinzipielle Struktur einer Inzidenzmatrix besitzt, fällt deren rechnergestützte Formulierung besonders einfach aus.

$$\begin{array}{c}
 \text{N001} \\
 0 \\
 \text{N002} \\
 \text{N003} \\
 \text{N004}
 \end{array}
 \begin{array}{c}
 \text{Vg} \quad \text{S,1} \quad \text{S,2} \quad \text{L} \quad \text{C} \quad \text{R} \quad \text{Vp} \quad \text{D} \\
 \left[\begin{array}{cccccccc}
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 \\
 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0
 \end{array} \right] = \mathbf{A}_a \quad (3.1)
 \end{array}$$

Die Inzidenzmatrix \mathbf{A}_a beinhaltet alle notwendigen Informationen über die Topologie des elektrischen Netzwerks, d. h. dessen Zusammensetzung über Zweige und Knoten. Mit ihrer Hilfe sollte es demnach möglich sein, die gesuchten Knoten- und Maschengleichungen aufzustellen. Jeder Spalte der Inzidenzmatrix kann ihr zugehöriger Zweigstrom zugeordnet werden. Eine Formulierung als Matrix-Vektor-Produkt mit dem Spaltenvektor aller Zweigströme zeigt, dass sich damit direkt, ohne weitere Umformungen, die Kirchhoff'schen Knotenregeln für alle Knoten des Netzwerks ergeben.

$$\begin{array}{c}
 \text{N001} \\
 0 \\
 \text{N002} \\
 \text{N003} \\
 \text{N004}
 \end{array}
 \begin{array}{c}
 \text{Vg} \quad \text{S,1} \quad \text{S,2} \quad \text{L} \quad \text{C} \quad \text{R} \quad \text{Vp} \quad \text{D} \\
 \left[\begin{array}{cccccccc}
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 \\
 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0
 \end{array} \right]
 \begin{array}{c}
 i_{\text{Vg}} \\
 i_{\text{S,1}} \\
 i_{\text{S,2}} \\
 i_{\text{L}} \\
 i_{\text{C}} \\
 i_{\text{R}} \\
 i_{\text{Vp}} \\
 i_{\text{D}}
 \end{array} = \mathbf{0} \quad (3.2)
 \end{array}$$

Die Inzidenzmatrix in (3.2) wird nun mittels elementarer Zeilenumformungen in Zeilenstufenform gebracht⁸. Da bei einer Matrix in Zeilenstufenform sämtliche Einträge unterhalb

⁸ Näheres zu den notwendigen Umformungen und den Eigenschaften der Zeilenstufenform kann [4] entnommen werden.

3.1. Anwendung der Kirchhoff'schen Regeln

der Hauptdiagonalen den Wert Null besitzen kann aus Gründen der Übersichtlichkeit auf diese Einträge verzichtet werden. In (3.3) und im weiteren Verlauf stehen die fehlende Einträge unterhalb der Hauptdiagonalen für Nulleinträge.

$$\begin{array}{cccccccc} & v_g & s_{,1} & s_{,2} & L & C & R & v_p & D & \begin{bmatrix} i_{v_g} \\ i_{s,1} \\ i_{s,2} \\ i_L \\ i_C \\ i_R \\ i_{v_p} \\ i_D \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\ & & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ & & & 0 & 1 & 1 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & = & \mathbf{0} \end{array} \quad (3.3)$$

Der zuvor erwähnte Satz der Netzwerkanalyse, welcher besagt, dass zu jedem zusammenhängenden Netzwerk mit k Knoten genau $k - 1$ linear unabhängige Knotengleichungen existieren, kann durch (3.3) bestätigt werden. Daher kann die letzte Zeile aus (3.3) verworfen werden.

3.1.2. Aufstellen der Maschengleichungen

Wie das vorhergehende Unterkapitel zeigt, stellt das automatisierte Formulieren der Knotengleichungen dank des matrizenähnlichen Aufbaus der Netzliste einen eher geringen Aufwand dar. Die nun folgende Vorgehensweise zielt darauf ab, in ähnlich einfacher Weise, die noch fehlenden Maschengleichungen aufzustellen. Ausgangspunkt bildet wiederum die Inzidenzmatrix. Diese beinhaltet nicht nur die Knotengleichungen, wie im vorhergehenden Unterkapitel gezeigt, sondern ebenfalls die Zusammenhänge zwischen den Knotenpotentialen und den Zweigspannungen. Das Matrix-Vektor-Produkt der transponierten Inzidenzmatrix \mathbf{A}_a^T (gekennzeichnet durch das hochgestellte T) mit dem Spaltenvektor der Knotenpotentiale \mathbf{v}_k liefert die Zweigspannungen \mathbf{v} .

$$\mathbf{A}_a^T \mathbf{v}_k = \mathbf{v} \quad (3.4)$$

Einsetzen der Inzidenzmatrix aus (3.1) sowie der Knotenpotentiale in der gleichen Reihenfolge wie die Zeilennamen aus (3.1) und der Zweigspannungen mit der gleichen Reihenfolge wie die Spaltennamen aus (3.1) in (3.4) liefert

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{N001} \\ \varphi_0 \\ \varphi_{N002} \\ \varphi_{N003} \\ \varphi_{N004} \end{bmatrix} = \begin{bmatrix} v_{v_g} \\ v_{s,1} \\ v_{s,2} \\ v_L \\ v_C \\ v_R \\ v_{v_p} \\ v_D \end{bmatrix}. \quad (3.5)$$

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Die Richtigkeit des Gleichungssystems in (3.5) kann mittels des Schaltplans in Abb. 3.1 leicht nachvollzogen werden. Damit aus diesen Zusammenhängen zwischen Knotenpotentialen und Zweigspannungen allerdings Maschengleichungen entstehen, müssen einige Umformungen durchgeführt werden. Zunächst werden die Ausdrücke auf beiden Seiten des Gleichheitszeichens in (3.4) transponiert:

$$\begin{aligned} (\mathbf{A}_a^\top \mathbf{v}_k)^\top &= \mathbf{v}^\top \\ \mathbf{v}_k^\top \mathbf{A}_a &= \mathbf{v}^\top. \end{aligned} \quad (3.6)$$

Anschließend wird die Inzidenzmatrix in reduzierte Zeilenstufenform gebracht. Die dazu nötigen elementaren Zeilenumformungen können durch eine linksseitige Multiplikation mit einer quadratischen, invertierbaren Matrix \mathbf{L} beschrieben werden [4]:

$$\mathbf{A}_{\text{ref}} = \mathbf{L} \mathbf{A}_a. \quad (3.7)$$

Übertragen auf das Beispiel des Buck-Boost-Wandlers lauten die Matrizen aus (3.7):

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.8)$$

Um (3.7) in (3.6) einsetzen zu können, ist zunächst eine Erweiterung von (3.6) nötig

$$\mathbf{v}_k^\top \mathbf{L}^{-1} \mathbf{L} \mathbf{A}_a = \mathbf{v}^\top. \quad (3.9)$$

Substituieren von (3.7) in (3.9) liefert

$$\mathbf{v}_k^\top \mathbf{L}^{-1} \mathbf{A}_{\text{ref}} = \mathbf{v}^\top. \quad (3.10)$$

Die Multiplikation des Zeilenvektors der Knotenpotentiale \mathbf{v}_k^\top und der Inversen der Matrix \mathbf{L} ergibt wiederum ein Zeilenvektor, der im Fall des Buck-Boost-Konverters die folgende Gestalt besitzt:

$$\mathbf{v}_k^\top \mathbf{L}^{-1} = \begin{bmatrix} \varphi_{\text{N001}} \\ \varphi_0 \\ \varphi_{\text{N002}} \\ \varphi_{\text{N003}} \\ \varphi_{\text{N004}} \end{bmatrix}^\top \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ -1 & 0 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \varphi_{\text{N001}} - \varphi_0 \\ \varphi_{\text{N001}} - \varphi_{\text{N002}} \\ \varphi_{\text{N004}} - \varphi_0 \\ \varphi_{\text{N003}} - \varphi_0 \\ \varphi_{\text{N001}} \end{bmatrix}^\top \quad (3.11)$$

Bislang ist noch nicht offensichtlich, wie mit Hilfe der Inzidenzmatrix die Maschengleichungen gewonnen werden können. Die bisherigen Gleichungssysteme beinhalten stets die

3.1. Anwendung der Kirchhoff'schen Regeln

Knotenpotentiale, die jedoch in den Kirchhoff'schen Maschengleichungen nicht vorkommen. Mit Hilfe der Darstellung in (3.10) können die Linearkombinationen der Knotenpotentiale aus (3.11) jedoch direkt Zweigspannungen zugeordnet werden. Eine erneute Transponierung der Ausdrücke auf beiden Seiten des Gleichheitszeichens in (3.10) macht diesen Umstand besser erkennbar.

$$\begin{aligned} \left(\mathbf{v}_k^\top \mathbf{L}^{-1} \mathbf{A}_{\text{rref}} \right)^\top &= \left(\mathbf{v}^\top \right)^\top \\ \mathbf{A}_{\text{rref}}^\top \left(\mathbf{v}_k^\top \mathbf{L}^{-1} \right)^\top &= \mathbf{v} \end{aligned} \quad (3.12)$$

Übertragen auf das Beispiel des Buck-Boost-Wandlers lautet (3.12)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{\text{N001}} - \varphi_0 \\ \varphi_{\text{N001}} - \varphi_{\text{N002}} \\ \varphi_{\text{N004}} - \varphi_0 \\ \varphi_{\text{N003}} - \varphi_0 \\ \varphi_{\text{N001}} \end{bmatrix} = \begin{bmatrix} v_{\text{Vg}} \\ v_{\text{S,1}} \\ v_{\text{S,2}} \\ v_{\text{L}} \\ v_{\text{C}} \\ v_{\text{R}} \\ v_{\text{Vp}} \\ v_{\text{D}} \end{bmatrix}. \quad (3.13)$$

Eine Auswertung des Matrix-Vektor-Produkts der ersten drei Zeilen sowie der fünften Zeile in (3.13) liefert die gesuchten Zusammenhänge zwischen den Linearkombinationen der Knotenpotentiale aus (3.11) und den Zweigspannungen. Die linksseitige Multiplikation der Ausdrücke auf beiden Seiten der Gleichung in (3.13) mit der Elementarmatrix

$$\mathbf{L}_{s45} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

vertauscht die vierte und fünften Zeilen, sodass die ersten vier Zeilen und Spalten der transponierten und in reduzierte Zeilenstufenform gebrachte Inzidenzmatrix nun die Ein-

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

heitsmatrix bilden⁹

$$\mathbf{L}_{s45} \mathbf{A}_{\text{rref}}^T (\mathbf{v}_k^T \mathbf{L}^{-1})^T = \mathbf{L}_{s45} \mathbf{v} \quad (3.15a)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{N001} - \varphi_0 \\ \varphi_{N001} - \varphi_{N002} \\ \varphi_{N004} - \varphi_0 \\ \varphi_{N003} - \varphi_0 \\ \varphi_{N001} \end{bmatrix} = \begin{bmatrix} v_{Vg} \\ v_{S,1} \\ v_{S,2} \\ v_C \\ v_L \\ v_R \\ v_{Vp} \\ v_D \end{bmatrix} \quad (3.15b)$$

Dass die letzte Spalte der Matrix in (3.15b) nullwertig ist und damit auch der letzte Eintrag im Spaltenvektor der Linearkombinationen nicht weiter benötigt wird, zeigt sich bereits in (3.13). Sie werden deshalb im Folgenden verworfen. Die Auswertung der ersten vier Zeilen des resultierenden Gleichungssystems liefert den gesuchten direkten Zusammenhang zwischen den Knotenpotentialdifferenzen und einer Auswahl an Zweigspannungen

$$\begin{bmatrix} \varphi_{N001} - \varphi_0 \\ \varphi_{N001} - \varphi_{N002} \\ \varphi_{N004} - \varphi_0 \\ \varphi_{N003} - \varphi_0 \end{bmatrix} = \begin{bmatrix} v_{Vg} \\ v_{S,1} \\ v_{S,2} \\ v_C \end{bmatrix}. \quad (3.16)$$

Die Auswertung der restlichen vier Zeilen des Gleichungssystems, unter Berücksichtigung des Zusammenhangs aus (3.16), liefert schließlich die gesuchten, linear unabhängigen Maschengleichungen des Netzwerks:

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{Vg} \\ v_{S,1} \\ v_{S,2} \\ v_C \end{bmatrix} = \begin{bmatrix} v_L \\ v_R \\ v_{Vp} \\ v_D \end{bmatrix}. \quad (3.17)$$

Gleichung (3.17) kann auch als

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{Vg} \\ v_{S,1} \\ v_{S,2} \\ v_C \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_L \\ v_R \\ v_{Vp} \\ v_D \end{bmatrix} = \mathbf{0} \quad (3.18)$$

⁹ In der Netzwerktheorie bedeutet dies, dass die zur Einheitsmatrix zugehörigen Zweige, hier also die Zweige Vg, S,1, S,2 und C, einen Baum des Netzwerks bilden [4].

3.2. Anwendung der Strom-/Spannungsbeziehungen

formuliert werden. Damit schließlich die bekannte Beziehung, wie sie in der Literatur [4] zu finden ist, entsteht, wird auf beiden Seiten der Gleichung mit minus eins multipliziert:

$$-\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{Vg} \\ v_{S,1} \\ v_{S,2} \\ v_C \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_L \\ v_R \\ v_{Vp} \\ v_D \end{bmatrix} = \mathbf{0}. \quad (3.19)$$

In (3.19) können nun beide Matrizen und beide Spaltenvektoren zu jeweils einer Matrix und einem Vektor vereinigt werden

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{Vg} \\ v_{S,1} \\ v_{S,2} \\ v_C \\ v_L \\ v_R \\ v_{Vp} \\ v_D \end{bmatrix} = \mathbf{0}. \quad (3.20)$$

Anhand der exemplarischen Inzidenzmatrix des Buck-Boost-Wandlers leitet dieser Unterabschnitt die Vorgehensweise zur automatisierten Formulierung linear unabhängiger Maschengleichungen her. Zusammenfassend und allgemein formuliert sind dazu die folgenden Schritte nötig:

1. Inzidenzmatrix \mathbf{A}_a in reduzierte Zeilenstufenform bringen
2. ggf. nullwertige Zeilen der reduzierten Matrix löschen
3. ggf. Spalten vertauschen, damit auf der linken Seite die Einheitsmatrix entsteht
4. die resultierende Matrix besitzt nun die Form $\mathbf{D} = [\mathbf{1} \mid \mathbf{D}_1]$
5. vertauschen der Einträge im Spaltenvektor der Zweigspannungen \mathbf{v} identisch zum Spaltentausch
6. das Gleichungssystem der Maschengleichungen lautet $[-\mathbf{D}_1^T \mid \mathbf{1}] \mathbf{v} = \mathbf{B} \mathbf{v} = \mathbf{0}$

3.2. Anwendung der Strom-/Spannungsbeziehungen

Im Allgemeinen sind zur Bestimmung sämtlicher Ströme und Spannungen eines zusammenhängenden Netzwerks mit z Zweigen insgesamt $2z$ linear unabhängige Gleichungen notwendig. Die Kirchhoff'schen Regeln aus dem vorhergehenden Kapitel tragen dazu die

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Hälfte der Gleichungen bei.

$$\begin{array}{l}
 k-1 \text{ Knotengleichungen} \\
 z-k+1 \text{ Maschengleichungen}
 \end{array}
 \begin{array}{cc}
 z \text{ Spalten} & z \text{ Spalten} \\
 \left[\begin{array}{cc}
 \mathbf{D} & \mathbf{0} \\
 \mathbf{0} & \mathbf{B}
 \end{array} \right]
 \begin{bmatrix}
 \mathbf{i} \\
 \mathbf{v}
 \end{bmatrix}
 = \mathbf{0}
 \end{array}
 \quad (3.21)$$

Die andere Hälfte ergibt sich aus den Strom-/Spannungsbeziehungen der Netzwerkelemente bzw. direkt aus den Quellgrößen. Aus Platzgründen wird im Folgenden auf das vollständige Gleichungssystem nach (3.21) verzichtet und nur die Matrix notiert. Die Spalten sind mit den zugehörigen Strömen bzw. Spannungen betitelt.

3.2.1. Widerstände

Ohmsche Widerstände (s. Abb. 3.2) lassen sich sehr einfach in das bestehende Gleichungssystem einfügen. Sie liefern einen direkten, algebraischen Zusammenhang zwischen Strom und Spannung eines Zweigs.

$$Ri_R - v_R = 0 \quad (3.22)$$

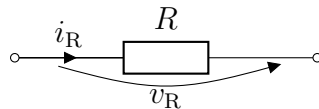


Abbildung 3.2.: Schaltbild des Widerstands mit Zählpfeilen

Das Beispiel des Buck-Boost-Wandlers aus Abb. 3.1 beinhaltet lediglich einen Widerstand. Entsprechend wird die Matrix aus (3.21) nun um eine zusätzliche Zeile erweitert.

$$\begin{array}{cccccccccccccccccccc}
 i_{Vg} & i_{S,1} & i_{S,2} & i_C & i_L & i_R & i_{Vp} & i_D & v_{Vg} & v_{S,1} & v_{S,2} & v_C & v_L & v_R & v_{Vp} & v_D \\
 \left[\begin{array}{cccccccc|cccccccc}
 & & & \mathbf{D} & & & & & & & & & & & & \\
 & & & \mathbf{0} & & & & & & & & & & & & \\
 0 & 0 & 0 & 0 & 0 & R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0
 \end{array} \right]
 \end{array}
 \quad (3.23)$$

Jeder Widerstand im Netzwerk liefert eine zusätzliche Zeile im Gleichungssystem.

3.2.2. Übertrager

Ideale Übertrager beschreiben das Verhalten einer idealen magnetischen Kopplung. Das ideale Verhalten zeigt sich zum einen darin, dass keine Streuinduktivitäten vorhanden sind, und zum anderen, dass die Selbstinduktivitäten gegen unendlich streben [19]. Eigenschaften realer Übertrager wie Streu- und Hauptinduktivität können sehr einfach mit Hilfe von idealen Übertragern und zusätzlichen Induktivitäten implementiert werden [19, 130]. Ideale Übertrager lassen sich sehr einfach in das bestehende Gleichungssystem einfügen. Sie liefern direkte, algebraische Zusammenhänge für Ströme und Spannungen. Im Gegensatz zu Widerständen handelt es sich dabei aber nicht um ein und denselben Zweig. Ein Übertrager mit insgesamt n Wicklungen und den Windungszahlen N_i liefert $n - 1$ Gleichungen der Form¹⁰:

$$\begin{aligned} v_1 - \frac{N_1}{N_2}v_2 &= 0 \\ v_1 - \frac{N_1}{N_3}v_3 &= 0 \\ &\vdots \\ v_1 - \frac{N_1}{N_n}v_n &= 0. \end{aligned} \tag{3.24}$$

Der n -Wicklungsübertrager (s. Abb. 3.3) besteht aus n Netzwerkzweigen und benötigt daher insgesamt n Gleichungen, damit das Netzwerk lösbar ist. Die noch fehlende Gleichung beschreibt den Zusammenhang zwischen den Strömen:

$$i_1 + \frac{N_2}{N_1}i_2 + \frac{N_3}{N_1}i_3 + \dots + \frac{N_n}{N_1}i_n = 0. \tag{3.25}$$

Die Matrix in (3.26) zeigt exemplarische Ausschnitte der zusätzlichen Zeilen im Gleichungssystem aufgrund eines idealen n -Wicklungsübertragers. Ein Beispiel anhand des Buck-Boost-Wandlers kann mangels eines Übertragers in der Schaltung nicht gegeben werden.

$$\begin{matrix} i_1 & i_2 & i_3 & \dots & i_n & v_1 & v_2 & v_3 & \dots & v_n \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & \dots & 0 & 1 & -\frac{N_1}{N_2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & -\frac{N_1}{N_3} & \dots & 0 \\ & & & & & \vdots & & & \ddots & \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & -\frac{N_1}{N_n} \\ 1 & \frac{N_2}{N_1} & \frac{N_3}{N_1} & \dots & \frac{N_n}{N_1} & 0 & 0 & 0 & \dots & 0 \end{array} \right] \end{matrix} \tag{3.26}$$

¹⁰ Die benötigten Windungszahlenverhältnisse lassen sich aus den in der Netzliste gegebenen Selbstinduktivitäten ermitteln. Es gilt: $\frac{N_n}{N_m} = \sqrt{\frac{L_{nn}}{L_{mm}}}$.

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

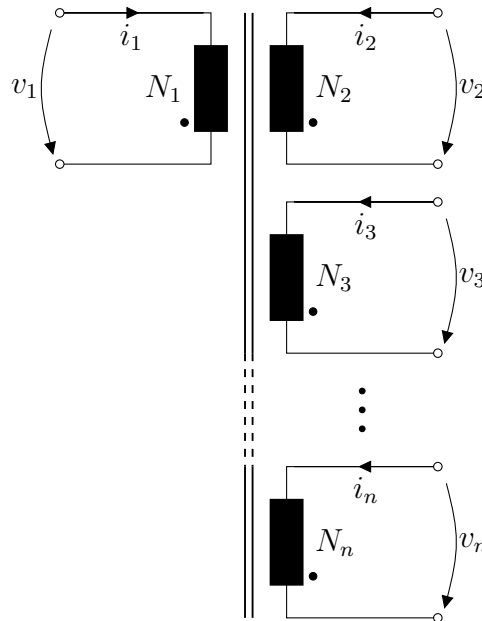


Abbildung 3.3.: Schaltbild des idealen n -Wicklungsübertragers

3.2.3. Induktivitäten und Kapazitäten

Im Gegensatz zu den Netzwerkelementen aus den vorherigen Kapiteln liefern reaktive Netzwerkelemente (s. Abb. 3.4) Strom-/Spannungsbeziehungen, welche die zeitliche Ableitung beinhalten.

$$v_L = L \frac{di_L}{dt} = L \dot{i}_L \quad (3.27)$$

$$i_C = C \frac{dv_C}{dt} = C \dot{v}_C \quad (3.28)$$

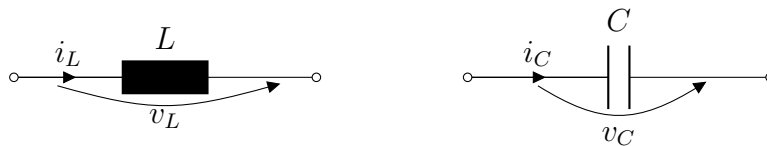


Abbildung 3.4.: Schaltbild von Induktivität (links) und Kapazität (rechts) mit Zählpfeilen

Eine Einbindung erscheint nicht möglich, denn das Gleichungssystem nach (3.21) beinhaltet keine Ableitungen der Ströme und Spannungen. Stattdessen können die Beziehungen an Induktivität und Kapazität dazu verwendet werden, die Unbekannten im Gleichungssystem umzudefinieren [4]. Die Spannungen an Induktivitäten sowie die Ströme durch Kapazitäten werden fortan als die zeitliche Ableitung des Induktivitätsstroms bzw. der

3.2. Anwendung der Strom-/Spannungsbeziehungen

Kapazitätsspannung interpretiert¹¹. Um (3.27) und (3.28) zu erfüllen, müssen zudem sämtliche Einträge der Matrix, die in der zur Induktivitätsspannung bzw. zum Kapazitätsstrom zugehörigen Spalte stehen, mit dem Induktivitätswert L bzw. dem Kapazitätswert C multipliziert werden. Am Beispiel des Buck-Boost-Wandlers ergibt sich nun die Matrix aus (3.29). Die geänderten Variablen und Spalten sind zusätzlich mit einem Kreis hervorgehoben.

$$\begin{array}{cccccccc|ccccccc}
 i_{Vg} & i_{S,1} & i_{S,2} & \dot{v}_C & i_L & i_R & i_{Vp} & i_D & v_{Vg} & v_{S,1} & v_{S,2} & v_C & \dot{i}_L & v_R & v_{Vp} & v_D \\
 \left[\begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & C & 0 & 1 & 0 & 1 \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 0 & 0 & 0 & 0 & 0 & R & 0 & 0 \\
 \end{array} \right] \begin{array}{ccccccc}
 -1 & 1 & 0 & 0 & L & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0
 \end{array} \quad (3.29)
 \end{array}$$

Die Matrix aus (3.29) erhält durch die Berücksichtigung der Zusammenhänge aus (3.27) bzw. (3.28) keine neuen Zeilen. Das Gleichungssystem wird demnach nicht ohne weiteres lösbar sein. Allerdings erlaubt der neu hinzugekommene Zusammenhang zwischen den Spalten \dot{v}_C und v_C bzw. \dot{i}_L und i_L das Aufstellen von (gekoppelten) Differentialgleichungen, die mittels geeigneter Methoden gelöst werden können. Eine systematische Vorgehensweise zur Extraktion der Differentialgleichungen ist Gegenstand von Kapitel 3.3.1.

3.2.4. Ideale Schalter und Dioden

Spannungsgesteuerte Schalter (s. Abb. 3.5) werden in SPICE-Netzlisten mit zwei Netzwerkwegen beschrieben. Der erste Zweig (Index 1) liegt am Ort des eigentlichen Schalters. Am anderen Zweig, dem Steuereingang (Index 2), wird die Steuerspannung erfasst. Zusätzlich zu diesen beiden Zweigen beinhaltet die Netzliste auch ein Schaltermodell (Zeile 9 in Listing 3.1), welches Parameter wie die Schaltschwelle der Steuerspannung (Vt) und Widerstand des Schalters im ein- und ausgeschalteten Zustand (R_{on} , R_{off}) definiert. Ideale Schalter ($R_{on}=0$ bzw. $R_{off}=\infty$) sind in SPICE nicht zulässig. Da der widerstandsbehaftete, nicht-ideale Schalter immer aus einem idealen Schalter mit zusätzlichen externen Widerständen modelliert werden kann, ist es an dieser Stelle ausreichend nur den speziellen Fall des idealen Schalters zu betrachten. Im eingeschalteten Zustand gilt für die Schalterspannung $v_{S,1} = 0$, im ausgeschalteten Zustand hingegen gilt für den Schalterstrom $i_{S,1} = 0$.

¹¹ Statt Spannung bzw. Strom können auch die elektrische Ladung q im Kondensator bzw. der magnetischer Fluss ϕ in der Spule verwendet werden. Die zugehörigen Gleichungen sind $v_L = \dot{\phi}$ und $i_C = \dot{q}$ sowie $i_L = \phi L^{-1}$ und $v_C = qC^{-1}$.

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Es liegt daher nahe, den Schalter, je nach Zustand, als nullwertige Spannungs- bzw. Stromquelle zu modellieren. Die Entscheidung, welcher der beiden Zustände ein idealer Schalter annimmt, gibt die Steuerspannung vor. Liegt diese oberhalb der Schwellenspannung ist der Schalter eingeschaltet, liegt sie darunter ist er ausgeschaltet. Die Steuerspannung $v_{S,2}$ liegt über dem zweiten Schalterzweig an. Dieser dient also lediglich der Spannungserfassung und weist einen unendlich hohen Innenwiderstand auf. Auch er wird daher als ideale Stromquelle mit Wert null im Gleichungssystem erfasst. Zusammenfassend gilt:

$$\text{leitender Schalter} \quad v_{S,1} = 0 \quad (3.30)$$

$$\text{sperrender Schalter} \quad i_{S,1} = 0 \quad (3.31)$$

$$\text{Steuereingang} \quad i_{S,2} = 0. \quad (3.32)$$

Im Falle stromgesteuerter Schalter wird der Steuerstrom über eine nullwertige Spannungsquelle am Steuereingang erfasst und statt (3.32) gilt dann: $v_{S,2} = 0$.

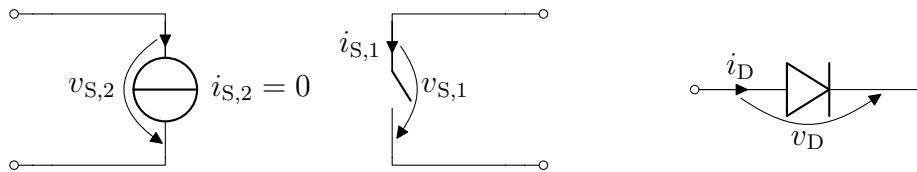


Abbildung 3.5.: Schaltbild von Schalter (links) und Diode (rechts) mit Zählpfeilen

Ideale Dioden können ebenfalls als ideale Schalter modelliert werden. Die Entscheidung, ob leitend oder sperrend, ergibt sich hier allerdings durch die Diodenspannung bzw. den Diodenstrom selbst. Ein zweiter Zweig als Steuereingang ist hier nicht nötig. Befinden sich Dioden im leitenden Zustand, so muss der Diodenstrom überwacht werden. Sobald dieser zu null wird, wechselt der Schaltzustand und die Diode wird durch eine ideale Stromquelle mit dem Wert null ersetzt. Ein erneuter Wechsel vom sperrenden in den leitenden Zustand erfolgt, sobald die Diodenspannung zu null wird.

$$\text{leitende Diode} \quad v_D = 0 \quad (3.33)$$

$$\text{sperrende Diode} \quad i_D = 0 \quad (3.34)$$

Spalten, die den Größen aus (3.30) bis (3.34) zugeordnet sind, werden an den rechten Rand der Gesamtmatrix verschoben. Diese Spalten bilden zusammen die Untermatrix der nullwertigen Hilfsquellengrößen. Die Gesamtmatrix am Beispiel des Buck-Boost-Wandlers

3.2. Anwendung der Strom-/Spannungsbeziehungen

für die Schaltkombination aus leitendem Schalter und sperrender Diode zeigt (3.35).

$$\begin{array}{cccccccccccccccc|cccc}
 i_{Vg} & i_{S,1} & \dot{i}_C & i_L & i_R & i_{Vp} & v_{Vg} & v_{S,2} & v_C & \dot{i}_L & v_R & v_{Vp} & v_D & i_{S,2} & i_D & v_{S,1} \\
 \left[\begin{array}{cccccccccccc}
 1 & 0 & 0 & 1 & 0 & 0 & & & & & & & & & & \\
 0 & 1 & 0 & -1 & 0 & 0 & & & & & & & & & & \\
 0 & 0 & 0 & 0 & 0 & 1 & & & & \mathbf{0} & & & & & & \\
 0 & 0 & C & 0 & 1 & 0 & & & & & & & & & & \\
 & & & & & & -1 & 0 & 0 & L & 0 & 0 & 0 & & & \\
 & & & & & & 0 & 0 & -1 & 0 & 1 & 0 & 0 & & & \\
 & & \mathbf{0} & & & & 0 & -1 & 0 & 0 & 0 & 1 & 0 & & \mathbf{0} & \\
 & & & & & & 1 & 0 & -1 & 0 & 0 & 0 & 1 & & & \\
 0 & 0 & 0 & 0 & R & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & & &
 \end{array} \right] \begin{array}{ccc}
 0 & -1 & \\
 0 & 1 & \mathbf{0} \\
 1 & 0 & \\
 0 & 1 & \\
 & & 1 \\
 & & 0 \\
 & \mathbf{0} & 0 \\
 & & -1 \\
 0 & 0 & 0
 \end{array} \quad (3.35)
 \end{array}$$

Die Matrix aus (3.35) erhält durch die Berücksichtigung der Zusammenhänge aus (3.30) bis (3.34) zwar keine neuen Zeilen, doch dafür ist die Anzahl der Unbekannten im System je betroffenem Zweig um eins geringer. Die Spannungen über leitenden Schaltern bzw. der Strom durch sperrende Schalter ist nullwertig und damit bekannt.

Um die Matrix für eine andere als der hier gezeigten Schaltkombination zu erhalten, müssen die entsprechenden Spalten der Matrix aus (3.35) verschoben werden [18, 131]. Wechselt der Schaltzustand bspw. in sperrenden Schalter und leitende Diode, so müssen in (3.35) die Spalten $i_{S,1}$, $i_{S,2}$ und v_D nach rechts und deren komplementäre Größen nach links verschoben werden¹². Insgesamt gibt es 2^n unterschiedliche Schaltkombinationen bzw. Matrizen, wenn n die Anzahl der im Netzwerk vorhandenen Schaltern und Dioden ist.

3.2.5. Gesteuerte Quellen

Die vier Grundtypen an linearen, gesteuerten Quellen können, je nach Steuergröße, in spannungs- und stromgesteuerte Quellen unterteilt werden. Neben dem Begriff der gesteuerten Quellen ist in der Literatur auch häufig die Bezeichnung als abhängige Quellen zu finden. Spannungsgesteuerte Quellen (E, G) beziehen ihre Ausgangsgröße auf eine beliebige Spannung im Netzwerk. Im einfachsten Fall setzt sich diese Spannung aus einer einzelnen Zweigspannung zusammen. Im allgemeinen Fall handelt es sich jedoch um eine beliebige Spannung zwischen zwei Netzwerkknoten, die sich aus mehreren Zweigspannungen zusammensetzen kann. Die direkte Einbindung in das bestehende Gleichungssystem erschwert sich dadurch erheblich. Eine mögliche Lösung besteht darin, den Eingang spannungsgesteuerter Quellen mit Hilfe einer Hilfsstromquelle mit dem Wert null zu modellieren. Abb. 3.6 zeigt das Schaltbild einer spannungsgesteuerten Stromquelle G mit Hilfsstromquelle.

¹² Auch hier gilt: Ist nichts anderes erwähnt, so werden zusammen mit den Spalten der Matrix auch die zugehörigen Elemente im Spaltenvektor des Gleichungssystems vertauscht.

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

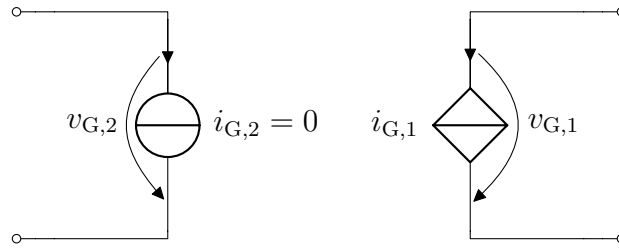


Abbildung 3.6.: Spannungsgesteuerte Stromquelle (rechts) mit Hilfsstromquelle (links)

Die beiden zur gesteuerten Quelle des Typs G zugehörigen Gleichungen lauten:

$$i_{G,1} - gv_{G,2} = 0 \quad (3.36)$$

$$i_{G,2} = 0. \quad (3.37)$$

Die zum Quellenstrom $i_{G,2}$ zugehörige Spalte wird in die gleiche Untermatrix wie die Hilfsquellen der Steuereingänge der Schalter verschoben (vgl. (3.35)). Die zur gesteuerten Quelle vom Typ G zugehörige Zeile zeigt (3.38). Mangels einer gesteuerten Quelle entfällt das konkrete Beispiel anhand des Buck-Boost-Wandlers.

$$\begin{array}{cccccccc|cccc} \dots & i_{G,1} & \dots & v_{G,1} & \dots & v_{G,2} & \dots & \dots & i_{G,2} & \dots & \dots \\ \mathbf{0} & 1 & \mathbf{0} & 0 & \mathbf{0} & -g & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \end{array} \quad (3.38)$$

Mit stromgesteuerten Quellen (F, H) verhält es sich prinzipiell ähnlich, wobei nun eine Spannungsquelle mit dem Wert null als Hilfsquelle am Eingang dient. Abb. 3.7 zeigt das Schaltbild einer stromgesteuerten Spannungsquelle H mit Hilfsspannungsquelle. Die zur

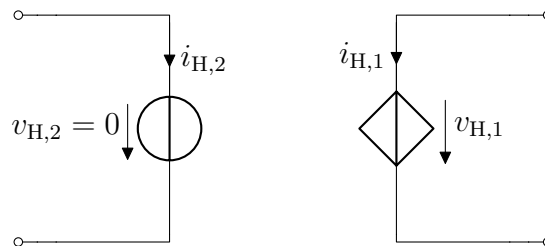


Abbildung 3.7.: Stromgesteuerte Spannungsquelle (rechts) mit Hilfsspannungsquelle (links)

gesteuerten Quelle vom Typ H zugehörige Zeile zeigt (3.39).

$$\begin{array}{cccccccc|cccc} \dots & v_{H,1} & \dots & i_{H,1} & \dots & i_{H,2} & \dots & \dots & v_{H,2} & \dots & \dots \\ \mathbf{0} & 1 & \mathbf{0} & 0 & \mathbf{0} & -g & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \end{array} \quad (3.39)$$

3.2.6. Unabhängige Spannungs- und Stromquellen

Die beiden Schaltungselemente in Abb. 3.8 zeigen eine Spannungs- und eine Stromquelle.

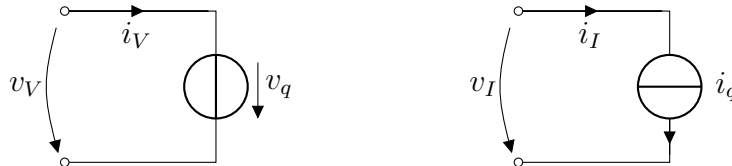


Abbildung 3.8.: Schaltbild von Spannungsquelle (links) und Stromquelle (rechts) mit Zählpfeilen

Die im Rahmen dieser Arbeit angewandte Eingangsgrößenmodellierung [97] sieht vor, sämtliche Quellen im Netzwerk durch Ersatznetzwerke zu beschreiben. Die notwendige Entkopplung, d. h. Rückwirkungsfreiheit des restlichen Netzwerks auf das Ersatznetzwerk der Quelle, wird durch gesteuerte Quellen garantiert. Es ist zu beachten, dass die Quellenmodellierung, wie sie in dieser Arbeit eingesetzt wird, die Zeitverläufe der ursprünglichen Quellen in keiner Weise vereinfacht, wie es in vielen anderen Simulationsprogrammen der Fall ist. Die Güte der Modellierung hängt ebenso wenig von der Dauer des Integrationszeitschritts ab. Die ursprünglichen Zeitverläufe werden vom jeweiligen Quellenmodell exakt wiedergegeben. Die in den folgenden Unterabschnitten vorgestellten Quellenmodelle beschränken sich auf die am häufigsten verwendeten Quellentypen in leistungselektronischen Schaltungen: Gleichspannungs- bzw. Gleichstromquellen, Pulsquellen und Sinus-Quellen.

Unabhängige Gleichspannungs- und Gleichstromquellen

Eine Gleichspannungsquelle kann mittels einer Kapazität und einer spannungsgesteuerten Spannungsquelle¹³ modelliert werden (vgl. Abb. 3.9).

Damit beide Spannungen äquivalent sind, muss die Verstärkung zu

$$e_g = \frac{v_{DC}}{v_C(0)} \quad (3.40)$$

gewählt. Der Anfangswert wird mit

$$v_C(0) = 1 \text{ V} \quad (3.41)$$

vorgegeben. Der Kapazitätswert C_{DC} besitzt keine Auswirkung auf den Zeitverlauf und kann beliebig gewählt werden. Gegenüber der ursprünglichen, unabhängigen Quelle erhöht das Quellenmodell die Anzahl der Zweige im Netzwerk um die Anzahl der zusätzlichen

¹³ Es wären auch andere Modelle einer Gleichspannungsquelle denkbar. So liefert eine Induktivität am Steuereingang einer stromgesteuerten Spannungsquelle ebenso eine Gleichspannung. Deren Wert hängt dann von der Quellenverstärkung und vom Anfangswert des Induktivitätsstroms ab.

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

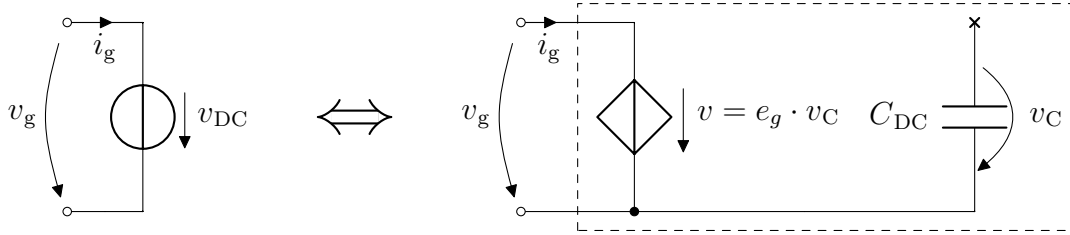


Abbildung 3.9.: Eingangsgroessenmodellierung einer Gleichspannungsquelle, auf einen zusätzlichen Eingangszweig mit nullwertiger Stromquelle (vgl. Abschnitt 3.2.5) kann hier und in den folgenden Quellenmodellen verzichtet werden

Zweige im Quellenmodell. Im Beispiel der Quelle aus Abb. 3.9 besitzt das Netzwerk aufgrund des Quellenmodells einen zusätzlichen Kapazitätszweig. Das Quellenmodell kann mit Hilfe der bereits beschriebenen Elemente einer gesteuerten Quelle aus Unterabschnitt 3.2.5 und einer Kapazität aus Unterabschnitt 3.2.3 im Gleichungssystem erfasst werden. Eine Gleichstromquelle lässt sich in gleicher Weise wie eine Gleichspannungsquelle modellieren, mit dem einzigen Unterschied, dass statt der spannungsgesteuerten Spannungsquelle eine spannungsgesteuerte Stromquelle verwendet werden muss.

Die um das Quellenmodell der Gleichspannungsquelle erweiterte Matrix des Buck-Boost-Wandlers lautet:

$$\begin{array}{cccccccccccccccccccc}
 i_{Vg} & i_{S,1} & \dot{v}_C & i_L & i_R & i_{VP} & \dot{v}_{Cdc} & v_{Vg} & v_{S,2} & v_C & \dot{i}_L & v_R & v_{VP} & v_D & v_{Cdc} & i_{S,2} & i_D & v_{S,1} \\
 \left[\begin{array}{cccccccccccccccc}
 1 & 0 & 0 & 1 & 0 & 0 & 0 & & & & & & & & & 0 & -1 & \\
 0 & 1 & 0 & -1 & 0 & 0 & 0 & & & & & & & & & 0 & 1 & \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & & \mathbf{0} & & & & & 1 & 0 & \mathbf{0} \\
 0 & 0 & C & 0 & 1 & 0 & 0 & & & & & & & & & 0 & 1 & \\
 & & & & & & & -1 & 0 & 0 & L & 0 & 0 & 0 & 0 & & & 1 \\
 & & & & & & & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & & & 0 \\
 & & & & \mathbf{0} & & & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & & \mathbf{0} & 0 \\
 & & & & & & & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & & & -1 \\
 0 & 0 & 0 & 0 & R & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & C_{DC} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -e_g & 0 & 0 & 0
 \end{array} \right]
 \end{array}
 \tag{3.42}$$

Hinzugekommen sind die beiden Spalten \dot{v}_{Cdc} und v_{Cdc} der Hilfskapazität sowie zwei Zeilen. Die vorletzte Zeile in (3.42) spiegelt die Knotengleichung an der Hilfskapazität wider und die letzte Zeile gibt den Zusammenhang der gesteuerten Hilfsquelle des Quellenmodells an.

Unabhängige Pulsspannungs- und Pulsstromquellen

Pulsspannungsquellen werden im Rahmen dieser Arbeit durch eine Spannungsquelle modelliert, die einen rampenförmigen, d. h. linear ansteigenden oder abfallenden Verlauf besitzt. Dabei wird die Steigung der Rampe dem aktuellen Zustand der Pulsquelle angepasst. Im Zustand der ansteigenden Flanke besitzt die Spannung eine positive Steigung, im Zustand des oberen sowie unteren Maximal- bzw. Minimalwerts eine Steigung von null und im Zustand der abfallenden Flanke eine negative Steigung. Das zugehörige Quellenmodell zeigt Abb. 3.10.

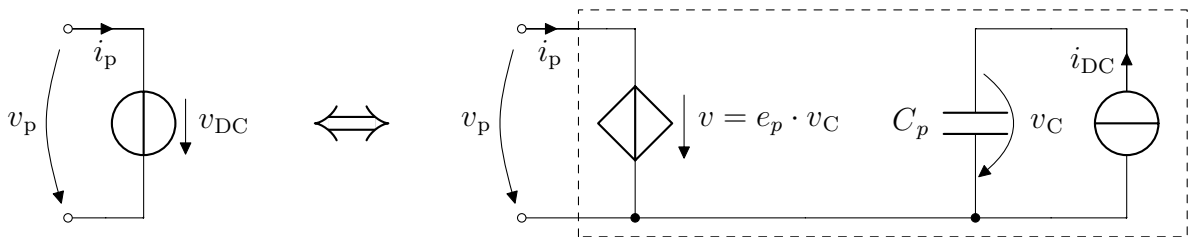


Abbildung 3.10.: Eingangsgrößenmodellierung einer Pulsspannungsquelle, die Verstärkung der spannungsgesteuerten Quelle beträgt eins

Die Steigung der Quellspannung ist

$$\dot{v}_p = e_p \dot{v}_C = \frac{e_p}{C_p} i_{DC} \quad (3.43)$$

und damit abhängig von der Verstärkung e_p , der Kapazität C_p und dem Strom der Gleichstromquelle i_{DC} . Wird die Kapazität zu $C_p = 1 \text{ F}$ und die Verstärkung zu $e_p = 1$ gewählt, so kann die Steigung direkt über den Wert der Stromquelle vorgegeben werden. Alle Schaltungselemente des Quellenmodells aus Abb. 3.10 können mit den bereits behandelten Verfahren aus den Abschnitten für gesteuerte Quelle (Unterabschnitt 3.2.5), Kapazitäten (Unterabschnitt 3.2.3) und Gleichstromquellen (Unterabschnitt 3.2.6) im Gleichungssystem erfasst werden.

Zur Einsparung unnötiger Spalten der Gesamtmatrix wird zur Beschreibung der Stromquelle in Abb. 3.10 die bereits vorhandene Kapazitätsspannung der DC-Quelle verwendet:

$$i_{DC} = g_p v_{Cdc} \quad (3.44)$$

Da die Spannung v_{Cdc} nach (3.41) bereits mit 1 V vorgegeben ist, kann der Strom und damit schließlich die Steigung \dot{v}_p direkt über die Transkonduktanz g_p gewählt werden. Die

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

um das Modell der Pulsspannungsquelle erweiterte Gesamtmatrix lautet:

$$\begin{array}{cccccccccccccccccccc|cccc}
 i_{Vg} & i_{S,1} & \dot{v}_C & i_L & i_R & i_{Vp} & \dot{v}_{Cdc} & \dot{v}_{Cp} & v_{Vg} & v_{S,2} & v_C & \dot{i}_L & v_R & v_{Vp} & v_D & v_{Cdc} & v_{Cp} & i_{S,2} & i_D & v_{S,1} \\
 \hline
 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & & & & & & & & & & 0 & -1 & \\
 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & & & & & & & & & & 0 & 1 & \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & & & & \mathbf{0} & & & & & & 1 & 0 & \mathbf{0} \\
 0 & 0 & C & 0 & 1 & 0 & 0 & 0 & & & & & & & & & & 0 & 1 & \\
 & & & & & & & & -1 & 0 & 0 & L & 0 & 0 & 0 & 0 & & & & 1 \\
 & & & & & & & & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & & & & 0 \\
 & & & & \mathbf{0} & & & & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & & & \mathbf{0} & 0 \\
 & & & & & & & & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & & & & -1 \\
 0 & 0 & 0 & 0 & R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & C_{DC} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -e_g & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_p & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -e_p & 0 & 0 & 0
 \end{array} \tag{3.45}$$

Hinzugekommen sind die beiden Spalten \dot{v}_{Cp} und v_{Cp} der Hilfskapazität sowie zwei Zeilen. Die vorletzte Zeile in (3.45) spiegelt die Knotengleichung an der Hilfskapazität wider und die letzte Zeile gibt den Zusammenhang der gesteuerten Hilfsquelle des Quellenmodells an.

Unabhängige Sinusspannungs- und Sinusstromquellen

Sinusspannungsquellen werden im Rahmen dieser Arbeit mit einem RLC-Schwingkreis und einer spannungsgesteuerten Spannungsquelle modelliert (vgl. Abb. 3.11).

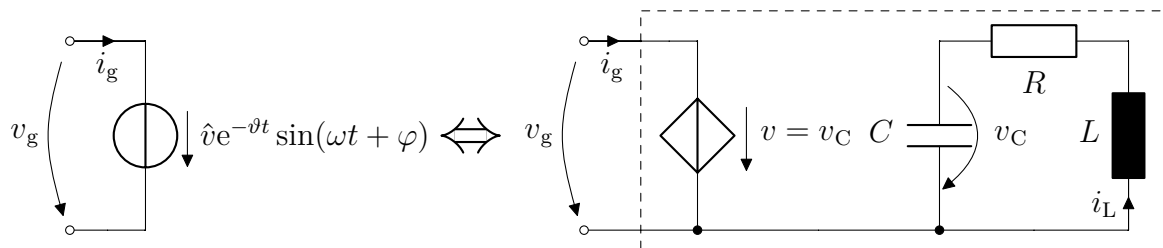


Abbildung 3.11.: Eingangsgößenmodellierung einer Sinusspannungsquelle, die Verstärkung der spannungsgesteuerten Quelle beträgt eins

Damit beide Spannungen, also die der Quelle und die des Quellenmodells übereinstimmen,

3.3. Formulierung der Zustandsraumdarstellung

müssen die Werte der Schaltungselemente zu

$$C = \frac{1}{\sqrt{\omega^2 + \vartheta^2}} \frac{1}{\Omega} \quad (3.46a)$$

$$L = \frac{1}{\sqrt{\omega^2 + \vartheta^2}} 1 \Omega \quad (3.46b)$$

$$R = 2\vartheta L \quad (3.46c)$$

gewählt werden. Zusätzlich müssen die Anfangswerte der Kapazitätsspannung und des Induktivitätsstroms zu

$$v_C(0) = \hat{v} \sin(\varphi) \quad (3.47a)$$

$$i_L(0) = -\frac{\hat{v}}{1\Omega} \sin\left(\varphi - \operatorname{atan}\left(\frac{\omega}{\vartheta}\right)\right) \quad (3.47b)$$

gewählt werden.

3.3. Formulierung der Zustandsraumdarstellung

Nachdem das Gleichungssystem mit den Knotengleichungen, den Maschengleichungen, den Strom-/Spannungsbeziehungen sowie den Quellenmodellen nun alle notwendigen Informationen enthält, besteht der folgende Schritt im Aufstellen der Zustandsraumdarstellung in ihrer Normalform:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (3.48)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (3.49)$$

Die Differentialgleichung (3.48), auch Zustandsdifferentialgleichung genannt, beschreibt das dynamische Systemverhalten der Zustandsvariablen (Einträge im Vektor \mathbf{x}) sowie den Einfluss der Eingangsvariablen (Einträge im Vektor \mathbf{u}). Um eine eindeutige Lösung dieser Differentialgleichung angeben zu können, sind zusätzlich Anfangswerte der Zustandsvariablen erforderlich. Aufgrund der Eingangsgrößenmodellierung entfällt die Matrix \mathbf{B} und der Vektor der Eingangsvariablen \mathbf{u} , d. h. es gilt: $\mathbf{B} = \mathbf{u} = \mathbf{0}$.

Die Ausgangsgleichung (3.49) beschreibt den Einfluss der Zustands- und Eingangsvariablen auf die Ausgangsvariablen (Einträge im Vektor \mathbf{y}). Aufgrund der Eingangsgrößenmodellierung entfällt die Matrix \mathbf{D} und der Vektor der Eingangsvariablen \mathbf{u} , d. h. es gilt: $\mathbf{D} = \mathbf{0}$.

Getaktete, leistungselektronische Schaltungen besitzen in den meisten Fällen nicht nur eine einzige Zustandsraumdarstellung. Je nach Schaltzustand der im Netzwerk vorhandenen Schalter und Dioden¹⁴ ergeben sich unterschiedliche Systemmatrizen $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ und in

¹⁴ Nicht nur die Zustände von Schaltern und Dioden führen auf unterschiedliche Matrizen, sondern auch die Zustände anderer Komponenten, wenn diese als abschnittsweise linear („piecewise linear“) modelliert werden.

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

einigen Fällen auch verschiedene Zustandsvektoren $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ mit wechselnder Anzahl an Zustandsvariablen [83].

3.3.1. Aufstellen der Zustandsdifferentialgleichung

Um aus dem Gleichungssystem in der Form (3.45) die Darstellung nach (3.48) zu erhalten, bedarf es zunächst einer Neuordnung der Spalten. Angelehnt an [71, 131], lautet die Spaltenreihenfolge von links nach rechts:

1. Spannungen und Ströme der Widerstände, gesteuerter Quellen und Übertrager
2. Ströme/Spannungen nullwertiger Hilfsspannungs-/Hilfsstromquellen
3. Ableitungen aller Kondensatorspannungen und Induktivitätsströme
4. alle Kondensatorspannungen und Induktivitätsströme¹⁵
5. Spannungen/Ströme nullwertiger Hilfsspannungs-/Hilfsstromquellen

Gleichung (3.50) zeigt das Ergebnis der Neusortierung am Beispiel des Buck-Boost-Wandlers für einen geschlossenen Schalter und eine sperrende Diode. Die Spaltensortierung nach Punkt 5 wurde bereits im Unterkapitel 3.2.4 erledigt. Für alle anderen Schaltkombinationen müssen die Spalten neu angeordnet werden.

$$\begin{array}{c}
 \begin{array}{cccccccccccccccccccc}
 i_R & v_R & i_{Vg} & i_{Vp} & v_{Vg} & v_{Vp} & v_{S,2} & v_D & i_{S,1} & \dot{i}_C & \dot{i}_L & \dot{i}_{Cdc} & \dot{i}_{Cp} & v_C & i_L & v_{Cdc} & v_{Cp} & i_{S,2} & i_D & v_{S,1}
 \end{array} \\
 \left[\begin{array}{cccccccccccccccccccc}
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\
 R & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{DC} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -e_g & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_p & 0 & 0 & -g_p & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -e_p & 0 & 0 & 0
 \end{array} \right]
 \end{array}
 \tag{3.50}$$

¹⁵ Die Reihenfolge der Kapazitäten und Induktivitäten muss in 3. und 4. gleich sein.

3.3. Formulierung der Zustandsraumdarstellung

Die Matrix aus (3.50) lässt sich mittels elementarer Zeilenumformungen in reduzierte Zeilenstufenform bringen.

$$\begin{array}{c}
 \begin{array}{cccccccccccc}
 i_R & v_R & i_{Vg} & i_{Vp} & v_{Vg} & v_{Vp} & v_{S,2} & v_D & i_{S,1} & \dot{v}_C & \dot{i}_L & \dot{v}_{Cdc} & \dot{v}_{Cp} & v_C & i_L & v_{Cdc} & v_{Cp} & i_{S,2} & i_D & v_{S,1}
 \end{array} \\
 \left[\begin{array}{cccccccccccc|cccc|cccc}
 1 & & & & & & & & & & & & & -\frac{1}{R} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & & & & & & & & & & & & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & 1 & & & & & & & & & & & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
 & & & 1 & & & & & & & & & & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & & & & 1 & & & & & & & & & 0 & 0 & -e_g & 0 & 0 & 0 & 0 \\
 & & & & & 1 & & & & & & & & 0 & 0 & 0 & -e_p & 0 & 0 & 0 \\
 & & & & & & 1 & & & & & & & 0 & 0 & 0 & -e_p & 0 & 0 & 0 \\
 & & & & & & & 1 & & & & & & -1 & 0 & e_g & 0 & 0 & 0 & -1 \\
 & & & & & & & & 1 & & & & & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
 & & & & & & & & & 1 & & & & \frac{1}{RC} & 0 & 0 & 0 & 0 & \frac{1}{C} & 0 \\
 & & & & & & & & & & 1 & & & 0 & 0 & -\frac{e_g}{L} & 0 & 0 & 0 & \frac{1}{L} \\
 & & & & & & & & & & & 1 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & 1 & 0 & 0 & -\frac{g_p}{C_p} & 0 & 0 & 0 & 0
 \end{array} \right]
 \end{array}
 \tag{3.51}$$

Das Differentialgleichungssystem kann für diesen Fall nun direkt aus den letzten vier Zeilen in (3.51) in Normalform abgelesen werden.

$$\mathbf{1} \begin{bmatrix} \dot{v}_C \\ \dot{i}_L \\ \dot{v}_{Cdc} \\ \dot{v}_{Cp} \end{bmatrix} + \begin{bmatrix} \frac{1}{RC} & 0 & 0 & 0 \\ 0 & 0 & -\frac{e_g}{L} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{g_p}{C_p} & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \\ v_{Cdc} \\ v_{Cp} \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{C} & 0 \\ 0 & 0 & \frac{1}{L} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{S,2} \\ i_D \\ v_{S,1} \end{bmatrix} = \mathbf{0} \tag{3.52}$$

Die Spalten der Schaltergrößen ($i_{S,2}$, i_D und $v_{S,1}$) können vernachlässigt werden, da sie mit den nullwertigen Hilfsquellgrößen multipliziert werden. Ein Vergleich von (3.52) mit (3.48) ergibt¹⁶:

$$\mathbf{x} = \begin{bmatrix} v_C \\ i_L \\ v_{Cdc} \\ v_{Cp} \end{bmatrix}, \mathbf{A} = - \begin{bmatrix} \frac{1}{RC} & 0 & 0 & 0 \\ 0 & 0 & -\frac{e_g}{L} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{g_p}{C_p} & 0 \end{bmatrix}, \mathbf{B} = \mathbf{0} \text{ und } \mathbf{u} = \mathbf{0}. \tag{3.53}$$

Für den Fall des Buck-Boost-Wandlers und den vorliegenden Schaltzuständen ist es demnach möglich die Systemmatrix \mathbf{A} (bzw. $-\mathbf{A}$) direkt aus dem Gleichungssystem abzulesen. Allerdings ist dies eher als Ausnahme anzusehen. Schon eine Änderung der Schaltkombination in sperrenden Schalter und sperrende Diode oder leitender Schalter und leitende Diode führt auf ein Gleichungssystem, bei dem die Matrix nicht mehr direkt abgelesen werden kann. Da diese Schaltzustände in leistungselektronischen Schaltungen keine

¹⁶ Die Vorzeichen der Matrizen ergeben sich aufgrund der Unterschiede in der Position bezüglich des Gleichheitszeichens zwischen (3.52) und (3.48).

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Seltenheit sind, muss ein Simulationsprogramm mit solchen Fällen umgehen können¹⁷. In Anhang A.3 sind deshalb Lösungsvorschläge beschrieben, die eine Formulierung der Zustandsraumdarstellung dennoch ermöglichen.

3.3.2. Aufstellen der Ausgangsgleichung

Eine Möglichkeit der rechnergestützten Formulierung von Ausgangsgleichungen ist die Verwendung von Hilfsquellen [4]. Im übertragenen Sinne werden dabei die Ausgangsgrößen mittels idealem Spannungs- bzw. Strommessgerät ermittelt. Das ideale Spannungsmessgerät besitzt einen unendlich hohen Innenwiderstand und kann daher durch eine Stromquelle mit dem Wert null beschrieben werden. Das ideale Strommessgerät hingegen besitzt einen Innenwiderstand von null und kann daher durch eine Spannungsquelle mit dem Wert null modelliert werden. Die gesuchte Ausgangsgröße ergibt sich dann als Spannung über der Hilfsstromquelle bzw. Strom durch die Hilfsspannungsquelle. Jede dieser Hilfsquellen stellt einen weiteren Zweig im Netzwerk dar, wodurch sich das Gleichungssystem zunächst vergrößert. Da jedoch der Quellenwert null beträgt, können diejenigen Spalten ignoriert werden, die mit dem Quellenwert multipliziert werden. Die betreffenden Spalten werden innerhalb der Matrix zwischen den Zustandsgrößen und den „echten“ Quellgrößen eingeordnet. Sie stehen damit in der gleichen Untermatrix wie die Spannungen über geschlossene Schalter und Ströme durch geöffnete Schalter (vgl. Kapitel 3.2.4). Beim Formulieren der Zustandsdifferentialgleichung fällt diese Untermatrix der Hilfsquellen daher raus (vgl. Kapitel 3.3.1). Unabhängig von der Anzahl der Ausgangsgrößen bleibt der Rechenaufwand zur Lösung des Differentialgleichungssystems demnach gleich.

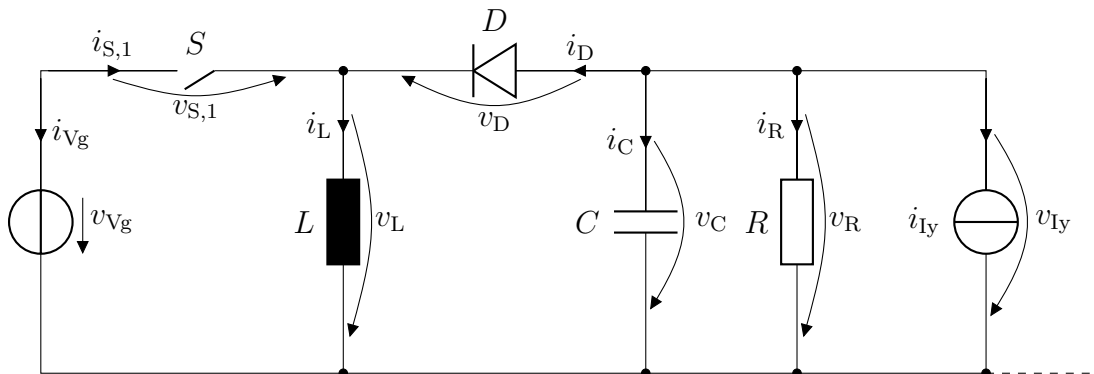


Abbildung 3.12.: Schaltplan des Buck-Boost-Wandlers mit Hilfsquelle I_y

Das Aufstellen einer Ausgangsgleichung soll nun anhand des Buck-Boost-Wandler-Beispiels verdeutlicht werden. Als Ausgangsgröße wird die Spannung über dem Widerstand ge-

¹⁷ Ein Grund für die genannte Problematik ist die Verwendung idealer Schalter. Eine einfache Abhilfe wäre ideale Schalter als unzulässig zu deklarieren, was bei der Mehrzahl an Simulationssoftware tatsächlich der Fall ist.

3.3. Formulierung der Zustandsraumdarstellung

wählt. Die um die notwendige Hilfsstromquelle erweiterte Schaltung zeigt Abb. 3.12. Aus Platzgründen wird auf die Darstellung der Pulsquelle v_p und des Schaltereingangs $S, 2$ verzichtet (vgl. Abb. 3.1). Ein Auszug aus der Gesamtmatrix des Gleichungssystems mit den für die Aufstellung der Ausgangsgleichung relevanten Zeilen und Spalten zeigt (3.54). Die zur Hilfsquelle zugehörigen Spalten v_{Iy} und i_{Iy} sind bereits an der richtigen Stelle einsortiert und müssen nicht mehr verschoben werden.

$$\left[\begin{array}{cccc|cccc|cccc|cccc} \dots & v_{Iy} & \dot{v}_C & \dot{i}_L & \dot{v}_{Cdc} & \dot{v}_{Cp} & v_C & i_L & v_{Cdc} & v_{Cp} & i_{S,2} & i_D & v_{S,1} & i_{Iy} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & & & & & & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 1 & & & & & \frac{1}{RC} & 0 & 0 & 0 & 0 & \frac{1}{C} & 0 & \frac{1}{C} \\ & & 1 & & & & 0 & 0 & -\frac{e_g}{L} & 0 & 0 & 0 & \frac{1}{L} & 0 \\ & & & 1 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 1 & & 0 & 0 & -\frac{g_p}{C_p} & 0 & 0 & 0 & 0 & 0 \end{array} \right] \quad (3.54)$$

Die Matrix befindet sich in reduzierter Zeilenstufenform. Die dafür notwendigen elementaren Zeilenumformungen sind in einem vorgelagerten Zwischenschritt bereits durchgeführt worden. Die Ausgangsgleichung kann aus (3.54) demnach direkt ausgelesen werden kann. Die betreffende Zeile lautet:

$$v_{Iy} + \begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \\ v_{Cdc} \\ v_{Cp} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{S,2} \\ i_D \\ v_{S,1} \\ i_{Iy} \end{bmatrix} = \mathbf{0}. \quad (3.55)$$

Die Spalten der Hilfsquellen und Schaltergrößen ($i_{S,2}$, i_D , $v_{S,1}$ und i_{Iy}) können vernachlässigt werden, da sie mit den nullwertigen Quellgrößen multipliziert werden. Ein Vergleich von (3.55) mit (3.49) ergibt¹⁸:

$$\mathbf{y} = v_{Iy}, \quad \mathbf{C} = -\begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \mathbf{0}. \quad (3.56)$$

Welche der Netzwerkgrößen als Ausgangsgröße gewünscht wird, ist dem Anwender in der Regel schon vor der Simulation bekannt und kann beim Formulieren des Gleichungssystems mitberücksichtigt werden. Das Programm fügt dann durch Interpretieren des `.save`-Befehls (s. Anhang A.2.2) in der Netzliste selbstständig die nötigen Hilfsquellen ins Netzwerk ein. Falls jedoch zusätzlich noch einige Größen nach der eigentlichen Berechnung benötigt werden, muss nicht die gesamte Simulation neu gestartet werden. Sämtliche Auswertungen können auch im Anschluss an die Berechnung durchgeführt werden. Da es sich bei den zusätzlichen Ausgangsgrößen um nullwertige Hilfsquellen handelt (Leerläufe parallel bzw. Kurzschlüsse in Reihe), gibt es effektiv keine Rückwirkung auf das Netzwerk.

¹⁸ Die Vorzeichen der Matrizen ergeben sich aufgrund der Unterschiede in der Position bezüglich des Gleichheitszeichens zwischen (3.55) und (3.49).

3. Rechnergestützte Formulierung der Zustandsraumdarstellung

Ein Hinzufügen bzw. Streichen dieser Hilfsquellen ändert nichts an der berechneten Lösung. Weitere Möglichkeiten der nachträglichen Verarbeitung von Simulationsergebnissen, wie etwa der Berechnung von Mittel- und Effektivwerten, werden in Kapitel 4.5 beschrieben.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Die rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung wird im Rahmen dieser Arbeit in sechs Teilaufgaben gegliedert:

- die Formulierung der Zustandsraumdarstellung auf Grundlage einer Netzliste,
- die Lösung des Differentialgleichungssystems,
- die Bestimmung konsistenter Anfangswerte,
- die Bestimmung konsistenter Zustände aller als stückweise linear definierten Schaltungselemente,
- die Bestimmung von Zeitpunkten eines Zustandswechsels der als stückweise linear definierten Schaltungselemente und
- die Verarbeitung und Ausgabe der Simulationsergebnisse.

Während die Formulierung der Zustandsraumdarstellung Gegenstand des vorhergehenden Kapitels ist, liegt der Fokus dieses Kapitels auf den weiteren Aufgabengebieten. Viele der Teilaufgaben und deren Lösungsansätze werden bereits in Kapitel 2 kurz vorgestellt. Die folgenden Abschnitte beschäftigen sich nun mit den mathematischen Details der Lösungsstrategien und weniger um deren softwaretechnische Umsetzung. Wo möglich, werden die Lösungen in Matrixschreibweise formuliert. Dadurch erleichtert sich die Umsetzung in Programmcode, denn die meisten Skript- und Programmiersprachen stellen bereits passende Datenstrukturen für Matrizen samt einer Bibliothek mit den zugehörigen Operationen bereit.

4.1. Lösen des Differentialgleichungssystems mit Hilfe der Matrixexponentialfunktion

In Kapitel 3 wird gezeigt, wie aus der Schaltungsbeschreibung in Form einer Netzliste ein lineares, zeitinvariantes und homogenes Differentialgleichungssystem der Form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \quad (4.1)$$

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

entsteht. Besteht das Netzwerk aus einem oder mehreren Schaltungselemente mit stückweise linearem Verhalten, so besitzt jedes der linearen Teilnetzwerke eine eigene Darstellung nach (4.1). Die transiente Schaltungsanalyse erfordert nun die Lösung dieses Differentialgleichungssystems für einen vorgegebenen Anfangswertevektor $\mathbf{x}(t_0)$. Ein möglicher Lösungsansatz besteht in der Verwendung der Matrixexponentialfunktion (vgl. Unterabschnitt 2.2.2):

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)} \mathbf{x}(t_0). \quad (4.2)$$

Die Schwierigkeit dieses Ansatzes liegt in der rechnergestützten Auswertung der Matrixexponentialfunktion. Die Entwicklung eines in jeder Hinsicht¹ zufriedenstellenden Algorithmus stellt eine besondere, nur schwer lösbare Aufgabe dar [91]. Dennoch besitzen die meisten Funktions-Bibliotheken numerischer Berechnungsprogramme bereits eine Routine zur Berechnung der Matrixexponentialfunktion. Viele der darin hinterlegten Algorithmen können [132] entnommen werden. Im Folgenden sollen zwei für diese Arbeit relevante Lösungsansätze vorgestellt werden, die nach [91] zu den wenigen empfehlenswerten gezählt werden können.

4.1.1. Reihenapproximation und “Scaling and squaring”

Aus Abschnitt 2.2.2 ist bereits bekannt, dass neben der skalaren Exponentialfunktion auch die Matrixexponentialfunktion über eine Taylorreihe definiert ist:

$$e^{\mathbf{A}t} = \mathbf{1} + \mathbf{A}t + \frac{1}{2!}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 + \dots \quad (4.3)$$

Die Reihe besitzt einen unbegrenzten Konvergenzradius, d. h. sie konvergiert für beliebige Argumente aus $\mathbb{C}^{n \times n}$ der Matrixexponentialfunktion [133]. Eine rechnergestützte Auswertung in endlicher Zeit erlaubt jedoch nur eine begrenzte Anzahl an Termen der Taylorreihe, wodurch ein Abbruchfehler entsteht. Zusätzlich haften dem Ergebnis Rundungsfehler an, da sämtliche Berechnungen in Gleitkommaarithmetik geschehen. Diese Rundungsfehler können dazu führen, dass das Ergebnis der Berechnung sehr ungenau wird, wie folgendes Beispiel in Anlehnung an [91] zeigt. Um die Problematik der Rundungsfehler zu veranschaulichen, ist es ausreichend, die Taylorreihe der skalaren Exponentialfunktion zu betrachten. Als beispielhaftes Argument sei -17 eingesetzt:

$$e^{-17} = \dots + \frac{(-17)^{16}}{16!} + \frac{(-17)^{17}}{17!} + \dots \quad (4.4)$$

Die exemplarischen Reihenglieder aus (4.4) besitzen unterschiedliche Vorzeichen jedoch den gleichen Betrag:

$$\frac{(-17)^{17}}{17!} = \frac{(-17)^{16} \cdot (-17)}{16! \cdot 17} = -\frac{(-17)^{16}}{16!}. \quad (4.5)$$

¹ Die Bewertungskriterien in [91] sind: Allgemeingültigkeit, Zuverlässigkeit, Stabilitätsverhalten, Genauigkeit, Effizienz, Speicherbedarf, Benutzerfreundlichkeit und Komplexität.

4.1. Lösen des DGL mit Hilfe der Matrixexponentialfunktion

Eine naive Auswertung der beiden Summanden im IEEE-754-Standard mit „double“-Genauigkeit verursacht jeweils einen relativen Fehler im Bereich von 10^{-16} , was im Bereich der sogenannten Maschinengenauigkeit liegt. Die anschließende Addition der beiden Zwischenergebnisse liefert aufgrund dieser Rundungsfehler nicht mehr exakt null, sondern einen kleinen Rest. Dieser Rest ist zwar im Verhältnis zum Betrag der Einzelglieder gering, auf das Gesamtergebnis bezogen kann dieser jedoch deutlich ins Gewicht fallen. Im vorliegenden Beispiel verursacht er eine relative Abweichung vom exakten Ergebnis von mehr als einem Prozent. Diese Problematik wird in der Numerik auch als Auslöschung bezeichnet (engl. „cancellation“) [134]. Sie ist nicht auf die Taylorreihe beschränkt und kann beispielsweise auch im Zusammenhang mit der weit verbreiteten Padé-Approximation² auftreten. Allerdings lässt sich das Problem der Auslöschung im Fall der Exponentialfunktion leicht umgehen. Unter Ausnutzung der Identität

$$e^{\mathbf{A}t} = \left(e^{\frac{\mathbf{A}t}{s}} \right)^s \quad (4.6)$$

kann das Argument der Exponentialfunktion mit $1/s$ soweit skaliert werden, dass die Auswirkungen der Rundungsfehler bei der Auswertung der Reihenapproximation in Grenzen gehalten werden können. Typischerweise entspricht s einer Zweierpotenz und wird häufig so gewählt, dass $\|\mathbf{A}\|t/s \leq 1$ gilt [91]. Das anschließende Potenzieren, das einem mehrmaligen Quadrieren (engl. „squaring“) gleichkommt, wenn s eine Zweierpotenz darstellt, macht die Skalierung (engl. „scaling“) rückgängig und liefert das gesuchte Ergebnis³. Algorithmen zur Berechnung der Matrixexponentialfunktion, die auf Basis einer Fehlerabschätzung selbstständig optimale Parameter für die Kombination aus Taylorreihenapproximation und dem „scaling and squaring“ finden, werden in [105, 135] vorgestellt. Eine Anwendung in der transienten Schaltungssimulation erfährt die Taylorreihenapproximation in [65, 78, 80] wohingegen in [75, 76] die Padé-Approximation eingesetzt wird.

4.1.2. Block-Diagonalisierung

Eine weitere nach [91] empfehlenswerte Methode zur Berechnung der Matrixexponentialfunktion, ist der Weg über eine Block-Diagonalisierung. Da die zugrundeliegende Matrix in den wenigsten Fällen in (Block-)Diagonalform vorliegt, sind Ähnlichkeitstransformationen nötig. Die Vorteile ergeben sich hauptsächlich dadurch, dass sich das Gesamtergebnis aus Teillösungen zusammensetzt, welche auf einfach zu behandelnde Sonderfälle reduziert werden können, darunter etwa die Bestimmung des Matrixexponentials nilpotenter Matrizen.

² Die MATLAB-Funktion `expm()` verwendet den Ansatz bestehend aus der Kombination des „Scaling & Squaring“ mit einer Padé-Approximation [91], sodass eine Auslöschung vermieden wird.

³ An dieser Stelle sollte erwähnt werden, dass es in einigen Fällen auch beim Quadrieren zu einer Auslöschung kommen kann [91].

Matrixexponential einer Blockdiagonalmatrix

Die Matrixexponentialfunktion einer Diagonalmatrix

$$\mathbf{A}_d = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix} \quad (4.7)$$

lässt sich sehr einfach über die skalare Exponentialfunktion ihrer Diagonalelemente berechnen:

$$e^{\mathbf{A}_d} = \begin{bmatrix} e^{a_{11}} & & & \\ & e^{a_{22}} & & \\ & & \ddots & \\ & & & e^{a_{nn}} \end{bmatrix}. \quad (4.8)$$

Doch nicht nur Diagonal-, sondern ebenfalls Blockdiagonalmatrizen

$$\mathbf{A}_{bd} = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n \end{bmatrix}, \quad (4.9)$$

deren Diagonaleinträge selbst wiederum Matrizen sind, welche auch als Blöcke bezeichnet werden, besitzen diese Eigenschaft:

$$e^{\mathbf{A}_{bd}} = \begin{bmatrix} e^{\mathbf{A}_1} & & \\ & \ddots & \\ & & e^{\mathbf{A}_n} \end{bmatrix}. \quad (4.10)$$

Matrixexponential einer nilpotenten Matrix

Neben den Diagonalmatrizen sind sogenannte nilpotente Matrizen eine weitere Matrizenklasse, die vorteilhafte Eigenschaften in Bezug auf die Berechnung der Matrixexponentialfunktion besitzt. Als nilpotent wird eine Matrix genau dann bezeichnet, sobald eine ihrer Potenzen die Nullmatrix ergibt:

$$\mathbf{A}_{nil}^m = \mathbf{0}. \quad (4.11)$$

Die Taylorreihenapproximation der Matrixexponentialfunktion dieser nilpotenten Matrix liefert eine endliche Summe:

$$e^{\mathbf{A}_{nil}} = \sum_{n=0}^{m-1} \frac{\mathbf{A}_{nil}^n}{n!}. \quad (4.12)$$

Die Reihe kann nach dem $m - 1$ -ten Glied abgebrochen werden, da alle weiteren Summanden aus Nullmatrizen bestehen und daher keinen Beitrag liefern.

Jordansche Normalform

Die Systemmatrix einer (leistungs-)elektronischen Schaltung ist typischerweise weder nilpotent, noch weist sie Diagonalform auf. Allerdings lässt sich jede Systemmatrix über eine Ähnlichkeitstransformation

$$\mathbf{A} = \mathbf{Q}\mathbf{J}\mathbf{Q}^{-1}. \quad (4.13)$$

in die Jordansche Normalform

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & & \\ & \ddots & \\ & & \mathbf{J}_n \end{bmatrix} \quad (4.14)$$

mit den auf der Hauptdiagonale liegenden Jordanblöcken

$$\mathbf{J}_j = \begin{bmatrix} \lambda_j & 1 & & \\ & \lambda_j & 1 & \\ & & \ddots & \ddots \\ & & & \lambda_j & 1 \\ & & & & \lambda_j \end{bmatrix} \quad (4.15)$$

bringen [117]. Die Matrixexponentialfunktion der Jordanschen Normalform lässt sich über die Exponentialfunktion der Jordanblöcke bestimmen [91]:

$$e^{\mathbf{J}} = \begin{bmatrix} e^{\mathbf{J}_1} & & \\ & \ddots & \\ & & e^{\mathbf{J}_n} \end{bmatrix}. \quad (4.16)$$

Die Exponentialfunktion eines Jordanblocks wiederum kann mittels einer Zerlegung in eine Diagonal- und eine nilpotente Matrix mit Hilfe der anfangs behandelten Ausnahmefälle bestimmt werden:

$$\mathbf{J}_j = \lambda_j \mathbf{1} + \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}. \quad (4.17)$$

Die Matrixexponentialfunktion eines einzelnen Jordanblocks der Größe m kann daher über

$$e^{\mathbf{J}_j} = e^{\lambda_j} \sum_{n=0}^{m-1} \frac{1}{n!} \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}^n \quad (4.18)$$

bestimmt werden. In [86] wird die Jordansche-Normalform und in [98] die reine Diagonalform zur transienten Schaltungssimulation eingesetzt. Allerdings ist die zugrunde liegende

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Ähnlichkeitstransformation (4.13) äußerst anfällig für Rundungsfehler. Da sich diese bei einer rechnergestützten Auswertung in Gleitkommaarithmetik nicht vermeiden lassen, wird von der Jordanschen-Normalform in numerischen Anwendungen abgeraten [91, 133].

Schursche Normalform

Numerisch stabil hingegen ist die sogenannte Schur-Zerlegung [91]

$$\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^{-1}, \quad (4.19)$$

welche die Systemmatrix zwar nicht in Blockdiagonalform, jedoch in eine obere Dreiecksmatrix⁴

$$\mathbf{T} = \begin{bmatrix} \lambda_1 & * & \cdots & * \\ & \lambda_2 & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{bmatrix} \quad (4.20)$$

transformiert. Die Einträge entlang der Hauptdiagonalen der oberen Dreiecksmatrix \mathbf{T} entsprechen dabei den Eigenwerten der Matrix \mathbf{A} . Zur Berechnung der Matrixexponentialfunktion einer oberen Dreiecksmatrix werden in [91] zwei Möglichkeiten genannt. Eine davon ist der rekursive Algorithmus nach Parlett⁵ [136]. Die andere Methode wandelt die obere Dreiecksmatrix durch eine weitere Transformation in eine Blockdiagonalmatrix um [133]:

$$\mathbf{A} = \mathbf{S}\mathbf{B}\mathbf{S}^{-1} \text{ mit } \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & & \\ & \ddots & \\ & & \mathbf{B}_n \end{bmatrix}. \quad (4.21)$$

Trotz ihrer ähnlichen Gestalt besteht ein wesentlicher Unterschied zwischen der Matrix der Jordanschen Normalform (4.14) und der blockdiagonalen Matrix (4.21). Während in der Jordanschen Normalform die Eigenwerte λ_j innerhalb eines Blocks identisch sind, können diejenigen innerhalb eines Blocks \mathbf{B}_j der Schur-Zerlegung voneinander abweichen. Das Ziel der Blockdiagonalisierung ist jedoch, die Blöcke so klein wie möglich zu halten, sodass im Idealfall die gleichen Blockgrößen wie in der Jordanschen Normalform vorliegen. Die Separation von sehr nahe beieinanderliegenden Eigenwerten in jeweils eigene Blöcke ist allerdings nicht immer sinnvoll, wie das folgende Beispiel nach [91] zeigt. Gegeben sei die Systemmatrix

$$\mathbf{A} = \begin{bmatrix} \lambda_1 & c \\ 0 & \lambda_2 \end{bmatrix} \quad (4.22)$$

⁴ Das Symbol * innerhalb Matrizen steht als Platzhalter für ein beliebiges Element.

⁵ In [136] wird gezeigt, dass die Parlett-Rekursion und die Blockdiagonalisierung mathematisch die gleichen Operationen anwenden, jedoch in unterschiedlicher Reihenfolge. Ein wesentlicher Unterschied ist allerdings, dass die Parlett-Rekursion, im Gegensatz zur Blockdiagonalisierung, nicht unabhängig eines etwaigen skalaren Faktors (bspw. der Zeit t) durchführbar ist.

4.1. Lösen des DGL mit Hilfe der Matrixexponentialfunktion

mit den zwei Eigenwerten λ_1 und λ_2 . Diese kann mittels der Ähnlichkeitstransformation

$$\mathbf{A} = \mathbf{SBS}^{-1} \text{ mit } \mathbf{S} = \begin{bmatrix} 1 & -\frac{c}{\lambda_1 - \lambda_2} \\ 0 & 1 \end{bmatrix} \text{ und } \mathbf{B} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (4.23)$$

diagonalisiert werden. Anhand des Elements $-c/(\lambda_1 - \lambda_2)$ der Transformationsmatrix \mathbf{S} aus (4.23) wird ersichtlich, dass deren Kondition umso schlechter wird, je näher die Eigenwerte beieinander liegen⁶. Ist die Matrix in oberer Dreiecksform (4.22) das Ergebnis einer vorhergehenden Schur-Transformation in Gleitkommaarithmetik, so können Eigenwerte durch Rundungsfehler lediglich sehr dicht beieinander liegen, selbst wenn die Systemmatrix ursprünglich identische Eigenwerte aufweist [91]. Ein Separieren dieser dicht beieinanderliegende Eigenwerte in unterschiedliche Blöcke provoziert folglich eine schlecht konditionierte Transformationsmatrix. Die Eigenwerte sollten in einem solchen Fall nicht in getrennte Blöcke verschoben werden, sondern als Matrixexponentialfunktion des gesamten Blocks

$$\mathbf{B}_j = \begin{bmatrix} \lambda_{j1} & * & \cdots & * \\ & \lambda_{j2} & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_{jn} \end{bmatrix} \quad (4.24)$$

ausgewertet werden. Mit dem Mittelwert der Eigenwerte

$$\bar{\lambda}_j = \frac{1}{n} \sum_{m=1}^n \lambda_{jm} \quad (4.25)$$

kann die Matrix in zwei Matrizen zerlegt werden

$$\mathbf{B}_j = \bar{\lambda}_j \mathbf{1} + \underbrace{\begin{bmatrix} \lambda_{j1} - \bar{\lambda}_j & * & \cdots & * \\ & \lambda_{j2} - \bar{\lambda}_j & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_{jn} - \bar{\lambda}_j \end{bmatrix}}_{\mathbf{M}_j}. \quad (4.26)$$

Dieser Vorgang wird auch als Eigenwertverschiebung bezeichnet [137]. Die Matrixexponentialfunktion dieser Summe zweier Matrizen lässt sich über die skalare Exponentialfunktion und eine Taylorreihe berechnen (vgl. (4.18)):

$$e^{\mathbf{B}_j} = e^{\bar{\lambda}_j} \sum_{n=0}^{\infty} \frac{\mathbf{M}_j^n}{n!}. \quad (4.27)$$

⁶ Nach [136] hängt die relative Genauigkeit des Gesamtergebnisses nicht von c ab, sondern ausschließlich vom Betrag des Abstands zwischen den Eigenwerten $\lambda_1 - \lambda_2$. Daher ist es ratsam, die Blockdiagonalisierung unter Berücksichtigung eines maximalen Zeitschritts h durchzuführen. Das bedeutet, dass die Systemmatrix mit h skaliert wird und die eigentliche Zeit t in eine ebenfalls skalierte Zeit $t_h = t/h$ gewandelt wird (vgl. Abschnitt 4.4.4).

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Aufgrund der innerhalb eines Blocks dicht beieinanderliegenden Eigenwerte ist die Matrix \mathbf{M}_j nahezu nilpotent [91] und die Taylorreihe konvergiert entsprechend schnell. Unter Berücksichtigung des Vorfaktors t lautet das Ergebnis entsprechend

$$e^{\mathbf{B}_j t} = e^{\bar{\lambda}_j t} \sum_{n=0}^{\infty} \frac{t^n \mathbf{M}_j^n}{n!}. \quad (4.28)$$

Der in dieser Arbeit verwendete Algorithmus zur Blockdiagonalisierung stammt aus [138] und ist Teil der MATLAB-Funktion `bdschur()`. Dieser benötigt zwar anfänglich mehr Rechenaufwand als alternative Berechnungsmethoden, liefert anschließend allerdings ein Ergebnis unabhängig⁷ des Zeitfaktors t .

4.1.3. Anwendung der Lösungsverfahren im Rahmen der Arbeit

Die in diesem Abschnitt vorgestellten Lösungsverfahren der Reihenapproximation und der Block-Diagonalisierung besitzen im Rahmen dieser Arbeit einen unterschiedlichen Einsatzzweck. Während ersterer zur Bestimmung des Matrixexponentials für fest vorgegebene Zeitschritte, also fest vorgegebene Werte des Zeitfaktors t Anwendung findet, dient die Block-Diagonalisierung zum Auffinden einer Lösungsfunktion in Abhängigkeit des Zeitfaktors t . Dieser funktionale Zusammenhang wird verwendet, um die Zeitpunkte von Zustandswechseln der als stückweise linear definierten Schaltungselemente zu detektieren (s. Abschnitt 4.4).

4.2. Bestimmung konsistenter Anfangswerte

Die Problematik der konsistenten Anfangswerte wird bereits in Abschnitt 2.2.4 des Grundlagenkapitels beschrieben. Darin zeigt sich, dass deren Bestimmung ein durchaus aufwendiger Vorgang sein kann. Im gleichen Abschnitt werden auch mögliche Lösungsansätze gezeigt. Diese unterliegen allerdings, wie die meisten der veröffentlichten Methoden, erheblichen Einschränkungen. So können viele nicht auf Netzwerke mit als stückweise linear definierten dynamischen Schaltungselementen (PWL-Kapazitäten/-Induktivitäten) angewendet werden. Zudem müssen häufig parasitäre Elemente in die Schaltung eingeführt werden, um impulsbehaftete Umladevorgänge zu unterdrücken. Darüber hinaus sind nur wenige der Lösungsmethoden für Netzwerke mit gesteuerten Quellen (engl. „active networks“) geeignet. Auch der in Abschnitt 2.2.4 vorgestellte Ansatz mit virtuellem Hilfsschalter ist in dieser Form nur auf Netzwerke mit linearen Kapazitäten und Induktivitäten anwendbar wie das folgende Beispiel zeigt.

⁷ An dieser Stelle sei angemerkt, dass eine optimale Zuordnung von Eigenwerten in gemeinsame Blöcke nicht unabhängig von einem eventuellen Vorfaktor ist, wie das Beispiel auf S. 13 in [136] zeigt. Dieser Umstand wird in dieser Arbeit berücksichtigt, da die Blockdiagonalisierung stets auf eine mit dem aktuellen Zeitschritt skalierte Systemmatrix angewendet wird (s. Abschnitt 4.4).

4.2. Bestimmung konsistenter Anfangswerte

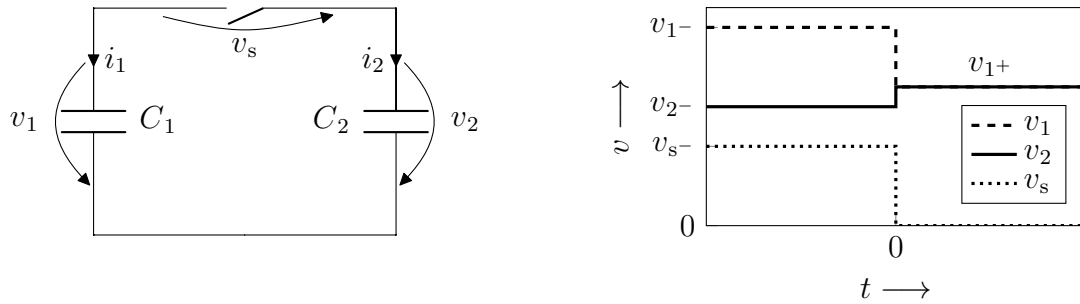


Abbildung 4.1.: Kurzschließen zweier Kapazitäten unterschiedlichen Anfangswerts mit idealem Schalter zum Zeitpunkt $t = 0$ (links) sowie zugehörige Spannungsverläufe (rechts)

Die Kapazitätsspannung aus Abb. 4.1 unmittelbar nach dem Schaltvorgang

$$v_{1+} = \frac{C_1 v_{1-} + C_2 v_{2-}}{C_1 + C_2} \quad (4.29)$$

hängt sowohl von den Endwerten des vorausgegangenen Intervalls, v_{1-} und v_{2-} , als auch von den Kapazitäten C_1 und C_2 ab. Es wird vorausgesetzt, dass es sich dabei um lineare Kapazitäten mit konstantem Kapazitätswert handelt. Sind im Gegensatz dazu als stückweise linear definierte Kapazitäten am Umladevorgang beteiligt, so können sich deren Kapazitätswerte während des Schaltvorgangs ändern. Der Anfangswert lässt sich dann nicht mehr mittels (4.29) bestimmen.

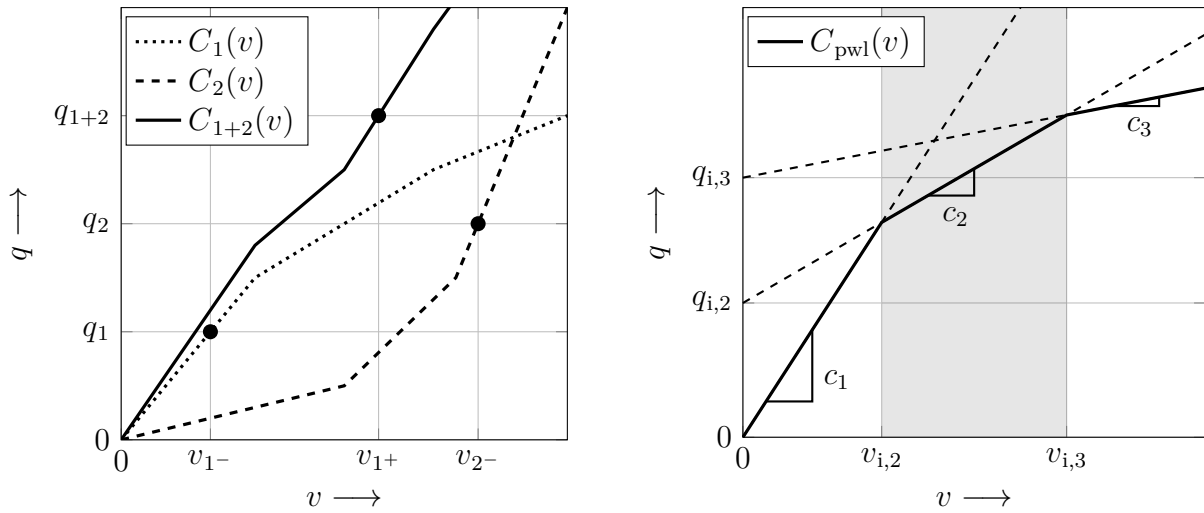
Eine mögliche Lösung bestünde nun darin, den Spannungssprung der Spannung über dem Schalter v_s in mehrere sukzessive Teilsprünge aufzuteilen. Die Sprunghöhe der einzelnen Durchgänge muss jeweils so angepasst werden, dass die Kapazitätsspannungen nach einem Teilsprung exakt auf den Grenzen der als stückweise linear definierten Kapazitätsverläufe zu liegen kommt. Die Anfangswerte der Kapazitätsspannungen entsprechen dann den Endwerten des vorausgegangenen Intervalls. Dieser eher aufwendige und umständliche Ansatz soll an dieser Stelle jedoch nicht weiter verfolgt werden. Stattdessen wird eine alternative Lösung erarbeitet, die auf dem Gesetz der Ladungs- bzw. Flusserhaltung beruht.

4.2.1. Gesetz der Ladungs- und Flusserhaltung

Das Gesetz der Ladungserhaltung fordert, dass die Gesamtladung beider Kapazitäten in Abb. 4.1 vor und nach dem unendlich schnellen Umladevorgang die gleiche ist.

Um die Situation zu verdeutlichen, werden die beispielhaften, nichtlinearen Kapazitätsverläufe für C_1 und C_2 aus Abb. 4.2a angenommen. Die beiden stückweise linearen Kapazitäten C_1 und C_2 ergeben in Parallelschaltung den stückweise linearen Verlauf der Ersatzkapazität C_{1+2} . Die Summe der beiden Einzelladungen $q_{1+2} = q_1 + q_2$ muss

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung



(a) Zwei q - v -Verläufe und deren Summe

(b) Ein exemplarischer q - v -Verlauf

Abbildung 4.2.: Beispiele für stückweise linearisierte q - v -Verläufe

laut dem Ladungserhaltungsgesetz vor und nach der Parallelschaltung identisch sein. Die gesuchte Spannung $v_{1+} = v_{2+}$ lässt sich damit direkt aus dem Diagramm anhand der Kennlinie C_{1+2} ablesen.

Eine solche grafische Auswertung kommt allerdings für die rechnergestützte Umsetzung nicht in Frage. Ein dafür besser geeigneter Weg soll anhand der Beispielschaltung aus Abb. 4.1 aufgezeigt werden. Deren Schaltungsanalyse liefert eine Maschengleichung für den Zeitpunkt unmittelbar nach dem Schaltvorgang

$$v_{1+} = v_{2+} \quad (4.30)$$

und eine Knotengleichung

$$i_1 = -i_2, \quad (4.31)$$

welche während des gesamten Schaltvorgangs Gültigkeit besitzt. Anhand der Maschengleichung (4.30) lässt sich erkennen, dass die Kapazitätsspannungen nach dem Schaltvorgang voneinander abhängig sind und daher eine von beiden nicht mehr länger zu den Zustandsgrößen gezählt werden kann. Im Beispiel soll nun die Spannung v_2 ihren Status als Zustandsgröße verlieren. Dies hat zur Folge, dass der Satz an nötigen Gleichungen um einen weiteren Zusammenhang ergänzt werden muss ([4], S. 354 bzw. [139], S. 82). Dazu wird für den Strom i_2 ein impulsbehaftete Ansatz [73, 108] gewählt:

$$i_2 = (q_{2+} - q_{2-})\delta(t). \quad (4.32)$$

Die Gewichtung des Dirac-Impulses ist die Differenz aus der Ladungsmenge der Kapazität C_2 unmittelbar nach dem Schaltvorgang q_{2+} bzw. davor q_{2-} . Ein Einsetzen von (4.32) in (4.31), das Ersetzen des Stroms i_1 durch die zeitliche Ableitung der Ladung und ein

4.2. Bestimmung konsistenter Anfangswerte

Integrieren vom Zeitpunkt unmittelbar vor bis unmittelbar nach dem Schaltvorgang führt auf

$$\begin{aligned}
 i_1 &= -i_2 \\
 \int_{t_0^-}^{t_0^+} i_1 \, dt &= - \int_{t_0^-}^{t_0^+} i_2 \, dt \\
 \int_{t_0^-}^{t_0^+} \dot{q}_1 \, dt &= - \int_{t_0^-}^{t_0^+} (q_{2^+} - q_{2^-}) \delta(t) \, dt.
 \end{aligned} \tag{4.33}$$

Die Lösung der Integrale auf beiden Seiten liefert

$$q_{1^+} - q_{1^-} = q_{2^-} - q_{2^+}. \tag{4.34}$$

Das Resultat in (4.34) entspricht dem Gesetz der Ladungserhaltung. Doch (4.34) kann lediglich eine Aussage über die Gesamtladung ($q_{1^+} + q_{2^+}$) zum Zeitpunkt t_{0^+} in Abhängigkeit der Ladungen zum Zeitpunkt unmittelbar vor dem Schaltvorgang t_{0^-} liefern. Die Aufteilung der Ladung auf die beiden Kapazitäten nach dem Schaltvorgang ist zunächst unbekannt, kann jedoch mit Hilfe der Maschengleichung (4.30) gewonnen werden. Dazu wird der stückweise lineare q - v -Verlauf einer nichtlinearen Kapazität C_{pwl} über Geradengleichungen der Form

$$q(v)|_{v_{i,n} \leq v \leq v_{i,n+1}} = c_n v + q_{i,n} \tag{4.35}$$

beschrieben (vgl. Abb. 4.2b). Einsetzen von (4.35) in (4.30) liefert

$$\frac{q_{1^+} - q_{1,i,n}}{c_{1,n}} = \frac{q_{2^+} - q_{2,i,m}}{c_{2,m}}. \tag{4.36}$$

Zusammen mit (4.34) ermöglicht (4.36) die Bestimmung der Ladungen q_{1^+} bzw. q_{2^+} . Da jedoch die Segmente des stückweise linearen q - v -Verlaufs, in denen die Lösung zu liegen kommt, nicht vor der Berechnung bekannt sind, müssen diese durch eine übergeordnete Iterationsschleife⁸ bestimmt werden. Anschließend kann mit Hilfe von (4.35) aus der Lösung die zugehörige Spannung bestimmt und somit ein konsistenter Anfangswert angegeben werden. Obwohl das Vorgehen hier anhand von nichtlinearen Kapazitäten gezeigt wird, kann es auch auf Induktivitäten übertragen werden. Statt einer Ladungserhaltung gilt dann die Erhaltung des verketteten Flusses.

⁸ Diese übergeordnete Iterationsschleife ist Bestandteil des noch folgenden Abschnitts 4.3.

4.2.2. Formulierung der Schalttransformationsgleichung

Anhand des Gesamtgleichungssystems^{9,10} eines beliebigen Netzwerks

$$\begin{bmatrix} \mathbf{1} & * & * & * \\ & \mathbf{1} & * & * \\ & & \mathbf{M}_i^{(0)} & -\mathbf{A}_i^{(0)} & -\mathbf{B}_i^{(0)} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \tilde{\mathbf{u}} \\ \dot{\mathbf{q}}^{(0)} \\ \mathbf{x}^{(0)} \\ \mathbf{u} \end{bmatrix} = \mathbf{0} \quad (4.37)$$

soll nun gezeigt werden, wie das Gesetz der Ladungs- und Flusserhaltung aus dem vorherigen Unterabschnitt auf ein umfangreiches Netzwerk mit mehreren stückweise linear definierten Kapazitäten und Induktivitäten angewendet werden kann. Es wird angenommen, dass die Gesamtmatrix, wie in (4.37), bereits in reduzierter Zeilenstufenform vorliegt. Der Formulierungsprozess besteht aus mehreren Iterationen innerhalb derer jeweils nur die letzte Zeile berücksichtigt wird:

$$[0 \ 0 \ \dots \ 0] \dot{\mathbf{q}}^{(0)} - [a_1 \ a_2 \ \dots \ a_n] \mathbf{x}^{(0)} - [b_1 \ b_2 \ \dots \ b_n] \mathbf{u} = 0. \quad (4.38)$$

Dabei wird davon ausgegangen, dass sämtliche Elemente m_j innerhalb der letzten Zeile der Untermatrix $\mathbf{M}_i^{(0)}$ null sind und mindestens eines der Elemente a_j von null verschieden ist¹¹. Andernfalls können die folgenden Schritte übersprungen werden und mit einer Matrix der Form (4.41) weiter verfahren werden¹². Die Elemente b_j sind hier ohne Bedeutung, da sie mit dem nullwertigen Spaltenvektor \mathbf{u} multipliziert werden. Im nächsten Schritt wird das erste nicht nullwertige Element a_j ermittelt. Die zugehörige Größe innerhalb des Vektors $\mathbf{x}^{(0)}$ ist von nun an eine abhängige Zustandsgröße. Deren Spalte wird vom Vektor $\mathbf{x}^{(0)}$ innerhalb der Matrix nach links in einen neu eingeführten Vektor $\tilde{\mathbf{x}}^{(1)}$ verschoben. Zudem muss auch die Spalte der komplementären Größe aus dem Vektor $\dot{\mathbf{q}}^{(0)}$ innerhalb der

⁹ Zur Wiederholung: Der Spaltenvektor \mathbf{w} enthält sämtliche Ströme und Spannungen über Widerständen und gesteuerten Quellen. Der Spaltenvektor $\tilde{\mathbf{u}}$ enthält sämtliche Ströme bzw. Spannungen nullwertiger (Hilfs-)Spannungsquellen bzw. Stromquellen. Der Spaltenvektor $\dot{\mathbf{q}}^{(0)}$ enthält sämtliche Ströme bzw. Spannungen der Kapazitäten bzw. Induktivitäten. Der Spaltenvektor $\mathbf{x}^{(0)}$ enthält sämtliche Spannungen bzw. Ströme der Kapazitäten bzw. Induktivitäten. Der Spaltenvektor \mathbf{u} enthält sämtliche Spannungen bzw. Ströme nullwertiger (Hilfs-)Spannungsquellen bzw. Stromquellen. Der hochgestellte Zählindex in Klammern ist nötig, da sich die jeweiligen Matrizen und Vektoren mit jedem Durchgang des Formulierungsprozesses ändern.

¹⁰ Die Matrizen in (4.37) besitzen den Index i um zu kennzeichnen, dass sie im Allgemeinen nicht identisch mit den Matrizen der Zustandsraumdarstellung \mathbf{A} , \mathbf{B} und \mathbf{M} sind.

¹¹ Bereits im Anhang A.3.2 wurde das Vorgehen für einen identischen Fall beschrieben. Im Gegensatz zur Formulierung der Zustandsraumdarstellung, wie im Anhang gezeigt, muss bei der Formulierung der Schalttransformationsgleichung jedoch zwischen abhängigen und unabhängigen Zustandsgrößen unterschieden werden. Für die Formulierung der Zustandsraumdarstellung ist diese Unterscheidung jedoch unnötig [4]

¹² Es wird davon ausgegangen, dass es sich um ein lösbares Netzwerk mit einer eindeutigen Lösung handelt. Andernfalls wird die Simulation mit einer Fehlermeldung abgebrochen.

4.2. Bestimmung konsistenter Anfangswerte

Matrix nach links in einen neu eingeführten Vektor $\tilde{\mathbf{q}}^{(1)}$ verschoben werden. Zeitgleich mit dem Verschieben der Spalten bekommt das Gleichungssystem eine zusätzliche Gleichung mit der Zeile

$$\mathbf{1} \tilde{\mathbf{q}}^{(1)} - \mathbf{1} \tilde{\boldsymbol{\delta}}^{(1)} = \mathbf{0}. \quad (4.39)$$

Der darin neu eingeführte Spaltenvektor $\tilde{\boldsymbol{\delta}}^{(1)}$ beinhaltet den gewichteten Delta-Impuls der komplementären Größe während des Schaltvorgangs. Im Beispiel der beiden Kapazitäten aus Abschnitt 4.2.1 entspricht (4.39) der Gleichung (4.32). Vor dem nächsten Durchgang muss das neue Gleichungssystem in reduzierte Zeilenstufenform gebracht werden:

$$\begin{bmatrix} \mathbf{1} & & & & & & \\ & \mathbf{1} & & & * & * & * \\ & & \mathbf{1} & & * & * & * \\ & & & \mathbf{1} & * & * & * \\ & & & & \mathbf{M}_i^{(1)} & -\mathbf{A}_i^{(1)} & -\mathbf{B}_i^{(1)} & \tilde{\mathbf{B}}_i^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \tilde{\mathbf{u}} \\ \tilde{\mathbf{q}}^{(1)} \\ \tilde{\mathbf{x}}^{(1)} \\ \dot{\tilde{\mathbf{q}}}^{(1)} \\ \mathbf{x}^{(1)} \\ \mathbf{u} \\ \tilde{\boldsymbol{\delta}}^{(1)} \end{bmatrix} = \mathbf{0}. \quad (4.40)$$

Solange nach der j -ten Iteration die letzte Zeile innerhalb der Untermatrix $\mathbf{M}_i^{(j)}$ ausschließlich nullwertige Elemente aufweist, müssen die gleichen Schritte wie zuvor ausgeführt werden. Jeder dieser Schritte beginnt mit der Auswahl einer abhängigen Zustandsgröße und endet mit einem erweiterten Gleichungssystem. Dabei ist zu beachten, dass die Spaltenvektoren $\tilde{\mathbf{q}}^{(j)}$, $\tilde{\mathbf{x}}^{(j)}$ und $\tilde{\boldsymbol{\delta}}^{(j)}$ bereits Einträge aus den $(j - 1)$ früheren Iterationen besitzen. Die Reihenfolge der abhängigen Zustandsgrößen, deren komplementären Größen sowie der zugehörigen gewichteten Delta-Impulse muss in jedem dieser Vektoren übereinstimmen. Nachdem alle abhängigen Zustandsgrößen ermittelt und verschoben wurden, besitzt das Gleichungssystem schließlich die folgende Gestalt¹³

$$\begin{bmatrix} \mathbf{1} & & & & & & \\ & \mathbf{1} & & & * & * & * \\ & & \mathbf{1} & & * & * & * \\ & & & \mathbf{1} & \mathbf{E} & * & * \\ & & & & \mathbf{1} & -\mathbf{A}_i & -\mathbf{B}_i & \tilde{\mathbf{B}}_i \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \tilde{\mathbf{u}} \\ \tilde{\mathbf{q}} \\ \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{q}}} \\ \mathbf{x} \\ \mathbf{u} \\ \tilde{\boldsymbol{\delta}} \end{bmatrix} = \mathbf{0}. \quad (4.41)$$

¹³ Es ist zu beachten, dass die Matrix \mathbf{E} in (4.41) nicht für die Einheitsmatrix steht. Diese ist im Rahmen dieser Arbeit durch das Symbol $\mathbf{1}$ gekennzeichnet.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Das Gesetz der Ladungs- bzw. Flusserhaltung lässt sich über den Integraloperator, angewendet auf die letzte Zeilengruppe in (4.41), aufstellen und lautet:

$$\int_{t_0^-}^{t_0^+} \dot{\mathbf{q}} dt = \int_{t_0^-}^{t_0^+} \mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} - \tilde{\mathbf{B}}_i \tilde{\boldsymbol{\delta}} dt. \quad (4.42)$$

Das Integral über eine unendlich kurze Zeitdauer von t_0^- bis t_0^+ liefert nur für impulsbehaftete Anteile ein Ergebnis ungleich null. Mit

$$\tilde{\boldsymbol{\delta}} = (\tilde{\mathbf{q}}_+ - \tilde{\mathbf{q}}_-) \delta(t) \quad (4.43)$$

lautet die Lösung der Integrale aus (4.42)

$$\mathbf{q}_+ - \mathbf{q}_- = \tilde{\mathbf{B}}_i (\tilde{\mathbf{q}}_- - \tilde{\mathbf{q}}_+) \quad (4.44)$$

Diese beinhaltet die Spaltenvektoren \mathbf{q}_+ bzw. \mathbf{q}_- , welche sich aus sämtlichen Ladungsmengen bzw. Flüssen aller Kapazitäten und Induktivitäten, deren Spannungen bzw. Ströme zu den Zustandsgrößen zählen, zusammensetzen. Der Index + bzw. - steht für den Zeitpunkt nach bzw. vor dem Schaltvorgang. Ebenso enthalten die Spaltenvektoren $\tilde{\mathbf{q}}_+$ bzw. $\tilde{\mathbf{q}}_-$ sämtliche Ladungen bzw. Flüsse aller Kapazitäten und Induktivitäten, deren Spannungen bzw. Ströme nicht zu den Zustandsgrößen zählen.

Die zweitletzte Zeilengruppe in (4.41)

$$\tilde{\mathbf{x}} = -\mathbf{E} \mathbf{x} \quad (4.45)$$

beinhaltet die Trennmengen- bzw. Maschengleichungen der Zustandsgrößen. Die Anteile der Spaltenvektoren \mathbf{u} und $\tilde{\boldsymbol{\delta}}^{(j)}$ können in (4.45) vernachlässigt werden, da sie keinen Beitrag liefern.

Der Zusammenhang zwischen Zustandsgrößen und den zugehörigen Ladungsmengen bzw. Flüssen lässt sich, analog zu (4.35), mit Hilfe der Geradengleichungen der stückweise linear definierten Verläufe ermitteln¹⁴:

$$\mathbf{q}_{\pm} = \mathbf{C}_{\pm} \mathbf{x}_{\pm} + \mathbf{q}_{i,\pm} \quad \text{bzw.} \quad \tilde{\mathbf{q}}_{\pm} = \tilde{\mathbf{C}}_{\pm} \tilde{\mathbf{x}}_{\pm} + \tilde{\mathbf{q}}_{i,\pm}. \quad (4.46)$$

Die Matrizen \mathbf{C}_+ bzw. $\tilde{\mathbf{C}}_+$ sind Diagonalmatrizen mit den differentiellen Kapazitäten bzw. Induktivitäten, gültig zum Zeitpunkt t_0^+ . Analog dazu bezeichnen \mathbf{C}_- bzw. $\tilde{\mathbf{C}}_-$ Diagonalmatrizen mit den zum Zeitpunkt t_0^- gültigen differentiellen Kapazitäten bzw. Induktivitäten. In gleicher Weise verhält es sich mit den Spaltenvektoren $\mathbf{q}_{i,+}$ bzw. $\tilde{\mathbf{q}}_{i,+}$, die die y-Achsenabschnitte der Geradengleichung des jeweiligen q - v -Verlaufs zum Zeitpunkt t_0^+ beinhalten. Einsetzen von (4.46) in (4.44) liefert:

$$\begin{bmatrix} \mathbf{1} & \tilde{\mathbf{B}}_i \end{bmatrix} \begin{bmatrix} \mathbf{C}_+ & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}_+ \end{bmatrix} \begin{bmatrix} \mathbf{x}_+ \\ \tilde{\mathbf{x}}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \tilde{\mathbf{B}}_i \end{bmatrix} \left(\begin{bmatrix} \mathbf{C}_- & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}_- \end{bmatrix} \begin{bmatrix} \mathbf{x}_- \\ \tilde{\mathbf{x}}_- \end{bmatrix} + \begin{bmatrix} \mathbf{q}_{i,-} \\ \tilde{\mathbf{q}}_{i,-} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{i,+} \\ \tilde{\mathbf{q}}_{i,+} \end{bmatrix} \right). \quad (4.47)$$

¹⁴ Die Indizes + bzw. - in (4.46) entsprechen den Indizes n bzw. m in (4.36).

Einsetzen von (4.45) als zusätzliche Zeilen in (4.47) führt auf:

$$\begin{bmatrix} \mathbf{C}_+ & \tilde{\mathbf{B}}_i \tilde{\mathbf{C}}_+ \\ \mathbf{E} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_+ \\ \tilde{\mathbf{x}}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \tilde{\mathbf{B}}_i \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \left(\begin{bmatrix} \mathbf{C}_- & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}_- \end{bmatrix} \begin{bmatrix} \mathbf{x}_- \\ \tilde{\mathbf{x}}_- \end{bmatrix} + \begin{bmatrix} \mathbf{q}_{i,-} \\ \tilde{\mathbf{q}}_{i,-} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{i,+} \\ \tilde{\mathbf{q}}_{i,+} \end{bmatrix} \right). \quad (4.48)$$

Die Lösung dieses Gleichungssystems

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_+ \\ \tilde{\mathbf{x}}_+ \end{bmatrix} &= \mathbf{F} \begin{bmatrix} \mathbf{x}_- \\ \tilde{\mathbf{x}}_- \end{bmatrix} - \mathbf{G} \left(\begin{bmatrix} \mathbf{q}_{i,+} \\ \tilde{\mathbf{q}}_{i,+} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{i,-} \\ \tilde{\mathbf{q}}_{i,-} \end{bmatrix} \right) \\ &= \mathbf{F} \begin{bmatrix} \mathbf{x}_- \\ \tilde{\mathbf{x}}_- \end{bmatrix} - \mathbf{G} \begin{bmatrix} \Delta \mathbf{q}_i \\ \Delta \tilde{\mathbf{q}}_i \end{bmatrix} \end{aligned} \quad (4.49)$$

liefert die gesuchten konsistenten Anfangswerte \mathbf{x}_+ in Abhängigkeit der abhängigen $\tilde{\mathbf{x}}_-$ und unabhängigen Zustandsgrößen \mathbf{x}_- unmittelbar vor dem Schaltvorgang. Die Lösung (4.49) wird auch als Schalttransformationsgleichung bezeichnet [140]. Abschließend fehlt noch eine Überprüfung, ob die Lösung für jedes der nichtlinearen, als stückweise linear definierten Schaltungselemente innerhalb der Bereichsgrenzen ihrer Segmente liegen. Der Vorgang des Prüfens und eventuellen Anpassens ist Bestandteil des noch folgenden Abschnitts 4.3.

4.2.3. Einteilung in abhängige und unabhängige Zustandsgrößen

Im Beispiel der beiden parallel geschalteten Kapazitäten aus Abb. 4.1 ist es unerheblich, ob die Kapazitätsspannung v_1 oder v_2 als die abhängige Zustandsgröße deklariert wird. In beiden Fällen ergibt sich das gleiche Ergebnis. Allerdings gibt es auch Schaltungen, wie diejenige aus Abb. 4.3, bei denen die Wahl der abhängigen bzw. unabhängigen Zustandsgrößen eingeschränkt ist.

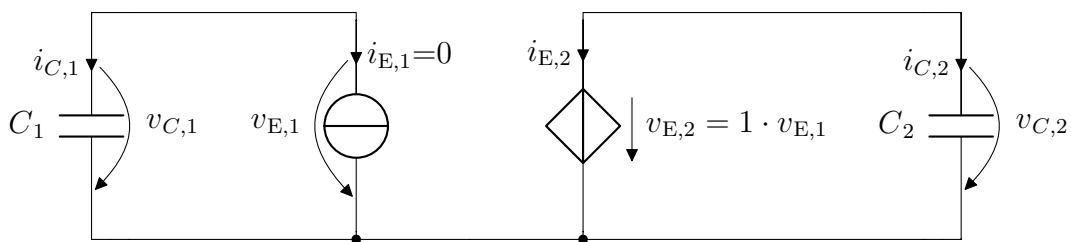


Abbildung 4.3.: Schaltung mit einer eindeutigen Zuordnung der unabhängigen und abhängigen Zustandsgröße

Anhand der Verschaltung in Abb. 4.3 wird offensichtlich, dass die Kapazitätsspannung $v_{C,2}$ am Ausgang der gesteuerten Quelle $v_{E,2}$ abhängig von der Kapazitätsspannung $v_{C,1}$ am Eingang $v_{E,1}$ der Quelle ist und nicht umgekehrt. Eine solche gerichtete Abhängigkeit zeigt sich jedoch in der zugehörigen und in reduzierte Zeilenstufenform gebrachte Matrix

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

des Gleichungssystems nicht direkt, wie (4.50) zeigt¹⁵:

$$\begin{array}{c}
 \mathbf{w}^\top \quad \mathbf{\tilde{u}}^\top \quad \mathbf{\dot{q}}^\top \quad \mathbf{x}^\top \quad \mathbf{u}^\top \\
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 v_{E,2} & i_{E,2} & v_{E,1} & i_{C,1} & i_{C,2} & v_{C,1} & v_{C,2} & i_{E,1} \\
 \hline
 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & 0 & -1 & 0 \\
 & & & 1 & 0 & 0 & 0 & 1 \\
 & & & & 0 & 1 & -1 & 0 \\
 \hline
 \end{array}
 \end{array} \cdot \quad (4.50)$$

Aus der letzten Zeile in (4.50) ist nicht ersichtlich, welche der beiden Kapazitätsspannungen nun zu den abhängigen Zustandsgrößen gezählt werden darf und welche nicht. Wird nun probeweise die Kapazitätsspannung $v_{C,1}$ als die abhängige Zustandsgröße erklärt und das Gleichungssystem um

$$i_{C,1} = (q_{1+} - q_{1-})\delta(t) = \tilde{\delta}_1 \quad (4.51)$$

ergänzt, lautet die Matrix des neuen Gleichungssystem in reduzierter Zeilenstufenform

$$\begin{array}{c}
 \mathbf{w}^\top \quad \mathbf{\tilde{u}}^\top \quad \mathbf{\tilde{q}}^\top \quad \mathbf{\tilde{x}}^\top \quad \mathbf{\dot{q}}^\top \quad \mathbf{x}^\top \quad \mathbf{u}^\top \quad \mathbf{\tilde{\delta}}_1^\top \\
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 v_{E,2} & i_{E,2} & v_{E,1} & i_{C,1} & v_{C,1} & i_{C,2} & v_{C,2} & i_{E,1} & \tilde{\delta}_1 \\
 \hline
 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 & & & 1 & 0 & 0 & 0 & 0 & -1 \\
 & & & & 1 & 0 & -1 & 0 & 0 \\
 & & & & & 0 & 0 & 1 & 1 \\
 \hline
 \end{array}
 \end{array} \cdot \quad (4.52)$$

Anhand der letzten Zeile in (4.52) zeigt sich die Auswirkung der falsch gewählten abhängigen Zustandsvariablen: Es entsteht ein inkonsistentes Netzwerk. Der unabhängigen Quellgröße $i_{E,1}$ wird ein Impuls aufgezungen, was unweigerlich eine Inkonsistenz hervorruft. Wird stattdessen die Spannung $v_{C,2}$ als abhängige Zustandsgröße gewählt sowie das Gleichungssystem um die Zeile

$$i_{C,2} = (q_{2+} - q_{2-})\delta(t) = \tilde{\delta}_2 \quad (4.53)$$

¹⁵ Die Kapazitäten C_1 und C_2 treten in der Matrix aus (4.50) nicht auf, da hier der Vektor $\dot{\mathbf{q}}$ statt $\dot{\mathbf{x}}$ verwendet wird.

4.2. Bestimmung konsistenter Anfangswerte

ergänzt, folgt daraus die in reduzierte Zeilenstufenform gebrachte Matrix

$$\begin{array}{c}
 \begin{array}{cc} \mathbf{w}^\top & \mathbf{\tilde{u}}^\top \end{array} \\
 \begin{array}{cc} v_{E,2} & i_{E,2} \end{array} \\
 \begin{array}{cc} v_{E,1} & i_{C,2} \end{array} \\
 \begin{array}{cc} v_{C,2} & i_{C,1} \end{array} \\
 \begin{array}{cc} v_{C,1} & i_{E,1} \end{array} \\
 \begin{array}{c} \mathbf{\tilde{\delta}}_1^\top \\ \tilde{\delta}_1 \end{array}
 \end{array}
 \left[\begin{array}{ccccccccc}
 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 & & & 1 & 0 & 0 & 0 & 0 & -1 \\
 & & & & 1 & 0 & -1 & 0 & 0 \\
 & & & & & 1 & 0 & 1 & 0
 \end{array} \right]. \quad (4.54)$$

Die letzte Zeile in (4.54) zeigt: Das Netzwerk bleibt konsistent.

Welche der Elemente innerhalb der letzten Zeile als abhängige Zustandsvariablen deklariert werden, kann demnach durch die Methode des „Versuch und Irrtum“ gefunden werden. Wenn sich nach einem Durchgang in der Matrix des Gleichungssystems eine Inkonsistenz zeigt, so muss in den vorherigen Zustand zurückgesprungen werden und die nächste Zustandsgröße als abhängig getestet werden. Dieser Vorgang wird solange wiederholt, bis eine abhängige Zustandsgröße gefunden ist. Falls die Suche scheitert, wird sie, zusammen mit einer entsprechenden Fehlermeldung an den Nutzer, abgebrochen.

4.2.4. Impulse höherer Ordnung

Weist ein Netzwerk mehrere abhängige Zustandsgrößen auf, so kann es unter Umständen zu Dirac-Impulsen höherer Ordnung kommen. Die Schaltung aus Abb. 4.4 stellt ein solches Beispiel dar.

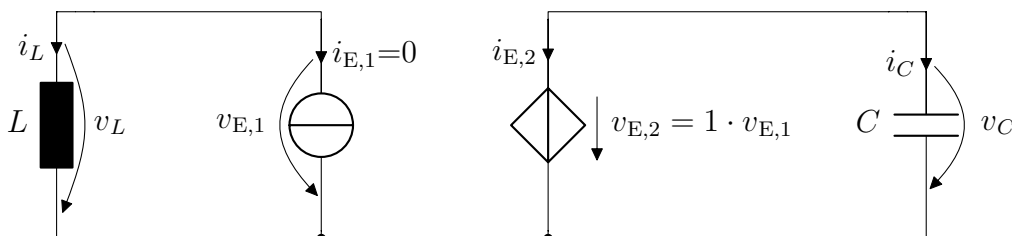


Abbildung 4.4.: Schaltung mit zwei abhängigen Zustandsgrößen sowie Dirac-Impulsen erster und zweiter Ordnung

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Die anfängliche Matrix des Gleichungssystems lautet in reduzierter Zeilenstufenform:

$$\begin{array}{c}
 \mathbf{w}^\top \quad \mathbf{\tilde{u}}^\top \quad \mathbf{\dot{q}}^\top \quad \mathbf{x}^\top \quad \mathbf{u}^\top \\
 \begin{array}{cc|cc|cc|c}
 v_{E,2} & i_{E,2} & v_{E,1} & i_C & v_L & v_C & i_L & i_{E,1} \\
 \hline
 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & -1 & 0 & 0 \\
 & & & 0 & 1 & -1 & 0 & 0 \\
 & & & & 0 & 0 & 1 & 1
 \end{array}
 \end{array} \quad (4.55)$$

Anhand der letzten Zeile in (4.55) lässt sich zunächst die Größe i_L eindeutig als abhängige Zustandsvariable identifizieren. Die bereits beschriebene Spaltenverschiebung und Ergänzung um die Gleichung¹⁶

$$v_L = (\lambda_{L+} - \lambda_{L-})\delta(t) = \tilde{\delta}_L \quad (4.56)$$

führen auf

$$\begin{array}{c}
 \mathbf{w}^\top \quad \mathbf{\tilde{u}}^\top \quad \mathbf{\tilde{q}}^\top \quad \mathbf{\tilde{x}}^\top \quad \mathbf{\dot{q}}^\top \quad \mathbf{x}^\top \quad \mathbf{u}^\top \quad \mathbf{\tilde{\delta}}^\top \\
 \begin{array}{cc|cc|cc|cc|c}
 v_{E,2} & i_{E,2} & v_{E,1} & v_L & i_L & i_C & v_C & i_{E,1} & \tilde{\delta} \\
 \hline
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\
 & & & 1 & 0 & 0 & 0 & 0 & -1 \\
 & & & & 1 & 0 & 0 & 1 & 0 \\
 & & & & & 0 & 1 & 0 & -1
 \end{array}
 \end{array} \quad (4.57)$$

Anhand der letzten Zeile in (4.57) lässt sich die Größe v_C wiederum eindeutig als abhängige Zustandsgröße identifizieren. Zudem wird aus der Zeile ersichtlich, dass diese Zustandsgröße bereits einen Dirac-Impuls aufweist. Die zusätzliche Gleichung muss demnach¹⁷

$$i_C = (q_{C+} - q_{C-})\delta(t) + C(\lambda_{L+} - \lambda_{L-})\dot{\delta}(t) = \tilde{\delta}_C + \tilde{\delta}_L^{(2)} \quad (4.58)$$

lauten. Mit dieser zusätzlichen Gleichung und der Spaltenverschiebung ergibt sich schließ-

¹⁶ In (4.56) wird statt der bisher verwendeten Variablen q die Variable λ verwendet, da es sich hier um den verketteten Fluss handelt.

¹⁷ Die Ordnung n der jeweiligen gewichteten Delta-Impulse wird mit einem hochgestellten Index in Klammern $\tilde{\delta}^{(n)}$ gekennzeichnet.

lich die gesuchte Matrix

$$\begin{array}{c}
 \begin{array}{cc|cc|cc|cc|cc|c}
 & \mathbf{w}^\top & & \tilde{\mathbf{u}}^\top & & \tilde{\mathbf{q}}^\top & & \tilde{\mathbf{x}}^\top & & \mathbf{u}^\top & & \tilde{\boldsymbol{\delta}}^{(1)\top} & & \tilde{\boldsymbol{\delta}}^{(2)\top} \\
 & v_{E,2} & i_{E,2} & v_{E,1} & i_C & v_L & v_C & i_L & i_{E,1} & \tilde{\delta}_C^{(1)} & \tilde{\delta}_L^{(1)} & & \tilde{\delta}_L^{(2)} & \\
 \hline
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & & \\
 & & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & & \\
 & & & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & & \\
 & & & & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & & \\
 & & & & & 1 & 0 & 0 & 0 & 0 & -1 & 0 & & \\
 & & & & & & 1 & 0 & 0 & 0 & -1 & 0 & & \\
 & & & & & & & 1 & 1 & 0 & 0 & 0 & & \\
 \hline
 \end{array}
 \end{array}
 \quad (4.59)$$

Auf die Berechnung konsistenter Anfangswerte haben Impulse höherer Ordnung jedoch keinen Einfluss, da sie aufgrund der zeitunabhängigen Matrix $\tilde{\mathbf{B}}$ keinen Anteil an der Lösung des Integrals (4.42) liefern [141].

4.2.5. Zusammenfassung

Die Bestimmung konsistenter Startwerte zum Zeitpunkt $t = 0^+$ unmittelbar nach dem Schaltvorgang geschieht im Rahmen dieser Arbeit mit einer Schalttransformationsgleichung der Form

$$\begin{bmatrix} \mathbf{x}_+ \\ \tilde{\mathbf{x}}_+ \end{bmatrix} = \mathbf{F} \begin{bmatrix} \mathbf{x}_- \\ \tilde{\mathbf{x}}_- \end{bmatrix} - \mathbf{G} \left(\begin{bmatrix} \mathbf{q}_{i,+} \\ \tilde{\mathbf{q}}_{i,+} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{i,-} \\ \tilde{\mathbf{q}}_{i,-} \end{bmatrix} \right). \quad (4.60)$$

Die neuen Startwerte sind demnach abhängig von den Werten aller Kapazitätsspannungen und Induktivitätsströme zum Zeitpunkt $t = 0^-$ unmittelbar vor dem Schaltvorgang sowie den „y-Achsenabschnitten“ der q - v -Verläufe bzw. der λ - i -Verläufe der nichtlinearen, als stückweise linear definierten (PWL) Kapazitäten und Induktivitäten. Das Flussdiagramm in Abb. 4.5 fasst nochmals die wichtigsten Schritte und deren Abfolge zusammen, die nötig sind, um aus der Matrix des Gesamtgleichungssystems nach (4.37) die Matrix des Gleichungssystems mit der reduzierten Anzahl an Zustandsvariablen (4.41) zu generieren. Aus den Untermatrizen dieses Gleichungssystems kann wiederum die Schalttransformationsgleichung (4.48) aufgestellt und nach dem Vektor der neuen Anfangswerte $[\mathbf{x}_+ \tilde{\mathbf{x}}_+]^\top$ aufgelöst werden.

Die Schlüsselkomponente des Verfahrens nach Abb. 4.5 ist ein sogenannter Stack, welcher auch als Stapelspeicher bezeichnet wird und dem „last-in-first-out“-Prinzip folgt. Ein Iterationsdurchgang der äußeren Schleife beginnt zunächst mit der Prüfung der Untermatrix \mathbf{M}_i auf Singularität. Fällt dieser Test positiv aus, d. h. die letzte Zeile besteht ausschließlich aus Nullen, so wird jeder Kandidat x_j , dessen zugehöriger Eintrag a_j in der letzten Zeile der Untermatrix \mathbf{A}_i nicht nullwertig ist, sowie die momentane Gesamtmatrix

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

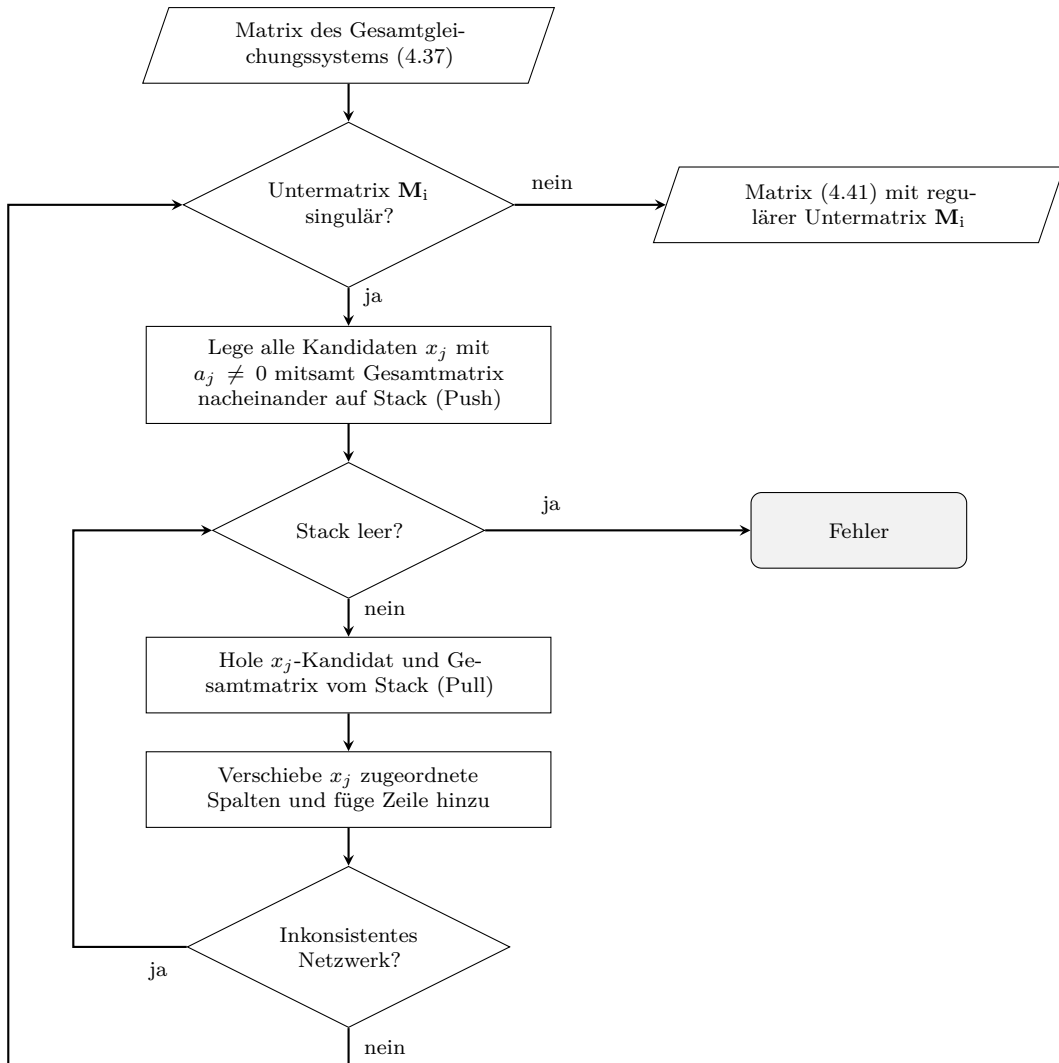


Abbildung 4.5.: Prozess um aus der Matrix des Gesamtgleichungssystems nach (4.37) die Matrix des Gleichungssystems mit der reduzierten Anzahl an Zustandsvariablen (4.41) zu generieren

als Kandidat/Matrix-Paar auf den Stack gelegt. Eine zweite, innere Schleife holt nun solange Kandidat/Matrix-Pärchen vom Stack und führt die Schritte aus Abschnitt 4.2.2 durch, bis eines gefunden wurde, welches auf ein konsistentes Netzwerk führt. Kann am Ende des Durchgangs kein weiteres Pärchen vom Stack abgerufen werden, so bricht das Programm mit einer Fehlermeldung ab. Andernfalls, wenn also ein Kandidat x_j erfolgreich als abhängige Zustandsgröße identifiziert ist, beginnt die äußere Schleife mit einem neuen Durchgang. Sobald der Test auf singuläre Untermatrix M_i negativ ausfällt, ist der Gesamtprozess beendet und es kann die Schalttransformationsgleichung (4.48) aufgestellt werden.

4.3. Bestimmung konsistenter Zustände nichtlinearer Schaltungselemente

Die Einführung impulsbehafteter Spaltenvektoren $\tilde{\delta}^{(n)}$ in die Matrix des Gesamtgleichungssystems diente in Abschnitt 4.2 der Bestimmung konsistenter Startwerte. Darüber hinaus lässt sich anhand des erweiterten Gleichungssystems erkennen, welche Netzwerkgrößen impulsbehaftet sind und welche nicht [73]. Lineare Schaltungselemente verhalten sich auch unter Dirac-Impulsen immer noch linear, das heißt sie ändern ihre Eigenschaft nicht. Nichtlineare Bauelemente, modelliert durch stückweise lineare Kennlinien, ändern jedoch möglicherweise ihr Verhalten unter Impulsbelastung. Eine ideale Diode sperrt einen Stromimpuls in negative Richtung, einen in positive Richtung jedoch nicht. Dank der zusätzlich eingeführten impulsbehafteten Spalten kann der Diodenstrom sehr einfach auf eine solche Verletzungen überprüft werden, wie das folgende Beispiel zeigt.

4.3.1. Einleitendes Beispiel

Eine mögliche Vorgehensweise zum Auffinden konsistenter Zustände soll anhand der Reihenschaltung einer Kapazität und einer idealen Diode (s. Abb. 4.6) erarbeitet werden.

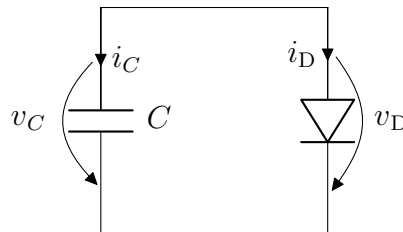


Abbildung 4.6.: Ist die Kapazität zu Beginn negativ geladen ($v_{C,0} < 0$) sperrt die Diode, ist sie positiv geladen ($v_{C,0} > 0$) tritt ein Dirac-Impuls auf. Ist sie vollständig entladen ($v_{C,0} = 0$), so sind beide Zustände der Diode (leitend/sperrend) erlaubt.

Anhand des Schaltbilds wird ersichtlich: Ist die Kapazität zu Beginn positiv geladen, d. h. $v_{C,0} > 0$, so wird die Diode in Leitrichtung betrieben und entlädt die Kapazität mit einem Stromimpuls. Ist sie dagegen negativ geladen, d. h. $v_{C,0} < 0$, so wird die Diode in Sperrrichtung betrieben und die Kapazitätsspannung behält ihren Anfangswert. Mit der Annahme, die Diode sperre, nimmt die in reduzierte Zeilenstufenform gebrachte Matrix

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

des Gleichungssystems die folgende Gestalt an:

$$\begin{array}{cccc}
 \tilde{\mathbf{u}}^\top & \dot{\mathbf{q}}^\top & \mathbf{x}^\top & \mathbf{u}^\top \\
 v_D & i_C & v_C & i_D \\
 \left[\begin{array}{c|c|c|c}
 1 & 0 & -1 & 0 \\
 0 & 1 & 0 & 1
 \end{array} \right] \cdot
 \end{array} \quad (4.61)$$

Ob die Annahme einer sperrenden Diode zutrifft, zeigt eine Auswertung der Diodenspannung. Diese kann aus der ersten Zeile in (4.61) direkt abgelesen werden:

$$v_D = v_C. \quad (4.62)$$

Zum Zeitpunkt $t = 0$ entspricht die Diodenspannung damit dem Anfangswert der Kapazität $v_D = v_{C,0}$. Jetzt zeigt sich, was bereits aus dem Schaltbild entnommen werden konnte: Ist der Anfangswert kleiner oder gleich null, so befindet sich die Diode im Sperrbetrieb und die anfänglich getroffene Annahme ist korrekt. Ein positiver Anfangswert verletzt jedoch die Sperrbedingung der idealen Diode. Die Annahme ist falsch und das Gleichungssystem muss auf den Zustand der leitenden Diode geändert werden:

$$\begin{array}{cccc}
 \tilde{\mathbf{u}}^\top & \dot{\mathbf{q}}^\top & \mathbf{x}^\top & \mathbf{u}^\top \\
 i_D & i_C & v_C & v_D \\
 \left[\begin{array}{c|c|c|c}
 1 & 1 & 0 & 0 \\
 0 & 0 & 1 & -1
 \end{array} \right] \cdot
 \end{array} \quad (4.63)$$

Die letzte Zeile aus (4.63) zeigt, dass die Kapazitätsspannung zu den abhängigen Zustandsgrößen gezählt werden muss. Zusätzlich zu den Spaltenverschiebungen muss das Gleichungssystem um die Zeile

$$i_C = (q_{C^+} - q_{C^-})\delta(t) = \tilde{\delta}_C \quad (4.64)$$

ergänzt werden. Die erweiterte, impulsbehaftete Netzwerkmatrix in reduzierter Zeilenstufenform lautet:

$$\begin{array}{ccccc}
 \tilde{\mathbf{u}}^\top & \tilde{\dot{\mathbf{q}}}^\top & \tilde{\mathbf{x}}^\top & \mathbf{u}^\top & \tilde{\delta}^\top \\
 i_D & i_C & v_C & v_D & \tilde{\delta}_C \\
 \left[\begin{array}{c|c|c|c|c}
 1 & 0 & 0 & 0 & 1 \\
 0 & 1 & 0 & 0 & -1 \\
 0 & 0 & 1 & -1 & 0
 \end{array} \right] \cdot
 \end{array} \quad (4.65)$$

4.3. Bestimmung konsistenter Zustände nichtlinearer Schaltungselemente

Wenn die Diode leitet, so muss deren Strom überwacht werden. Ist dieser negativ, so wird die Diode sperren. Der Diodenstrom kann aus der ersten Zeile in (4.65) direkt abgelesen werden:

$$i_D = -\tilde{\delta}_C = (q_{C-} - q_{C+})\delta(t). \quad (4.66)$$

Die Anwendung des Zusammenhangs zwischen Ladung, Kapazität und Spannung liefert:

$$i_D = C(v_{C-} - v_{C+})\delta(t). \quad (4.67)$$

Die Spannung der Kapazität zum Zeitpunkt t_{0+} kann direkt aus der letzten Zeile in (4.65) zu null abgelesen werden (da der Vektor \mathbf{u}^T nullwertig ist). Damit ergibt sich schließlich für den Diodenstrom

$$i_D = C v_{C-} \delta(t) = C v_{C,0} \delta(t). \quad (4.68)$$

Wie bereits aus dem Schaltbild ersichtlich, entlädt sich die Kapazität mit einem positiven Stromimpuls über die ideale Diode, wenn der Anfangswert der Kapazität $v_{C,0}$ positiv ist. Ist dieser jedoch negativ, so verletzt der negative Entladeimpuls die Leitbedingung der idealen Diode und sie sperrt.

Zusammenfassend gilt: Unabhängig davon, mit welchem Zustand bzw. Segment der als stückweise linear definierten Diodenkennlinie begonnen wird, lässt sich anhand des impulsbehafteten Gleichungssystems und der Leit- und Sperrbedingungen ein gültiger Endzustand finden.

4.3.2. Formulieren der Umschaltbedingungen als Ungleichungssystem

Der erste Schritt vom konkreten Beispiel der Schaltung aus Abb. 4.6 zu einer allgemeinen Vorgehensweise liegt in einer Formulierung der Umschaltbedingungen aller als stückweise definierten nichtlinearen Schaltungselemente¹⁸. Die in dieser Arbeit verwendete Vorgehensweise orientiert sich an [142]. Jedes nichtlineare Schaltungselement j besitzt für jeden Zustand n (d. h. für jedes seiner Segmente einer stückweise linearen Kennlinie) ein Ungleichungssystem der Form¹⁹:

$$\mathbf{L}_{j,n} \begin{bmatrix} \mathbf{v}_j \\ \mathbf{i}_j \end{bmatrix} \geq \mathbf{r}_{j,n}. \quad (4.69)$$

Im Fall einer idealen Diode lautet das Ungleichungssystem für den sperrenden Zustand

$$\underbrace{\begin{bmatrix} -1 & 0 \end{bmatrix}}_{\mathbf{L}_{j,1}} \begin{bmatrix} v_D \\ i_D \end{bmatrix} \geq \underbrace{0}_{\mathbf{r}_{j,1}} \quad (4.70)$$

¹⁸ Dies umfasst tatsächlich alle als stückweise definierten nichtlinearen Schaltungselemente, also sowohl PWL-Kapazitäten und -Induktivitäten als auch ideale Schalter und Dioden und alle sonstigen PWL-Schaltungselemente.

¹⁹ Die Verwendung von Vektoren für \mathbf{v}_j bzw. \mathbf{i}_j in (4.69) deutet darauf hin, dass es auch PWL-Elemente gibt, die aus mehr als einem Zweig bestehen, bspw. gesteuerte Schalter.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

und für den leitenden Zustand

$$\underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\mathbf{L}_{j,2}} \underbrace{\begin{bmatrix} v_D \\ i_D \end{bmatrix}}_{\mathbf{r}_{j,2}} \geq 0. \quad (4.71)$$

Im Falle eines PWL-Elements mit mehr als zwei Segmenten besitzt die Matrix $\mathbf{L}_{j,n}$ mehr als eine Zeile, denn mittlere Segmente benötigen sowohl eine obere als auch eine untere Schranke. Sobald eine der Ungleichungen verletzt wird, springt das Element in einen neuen, zunächst unbekanntem Zustand. Zum Auffinden des neuen, korrekten Zustands kann beispielsweise das Newton-Raphson-Verfahren aus Abschnitt 2.2.3 verwendet werden. Alternativ kann auch ein benachbarter Zustand ausgewählt werden. Im ungünstigsten Fall müssen alle Zustände des Elements getestet werden, bevor der korrekte gefunden wird. Allerdings lässt sich die Suche beschleunigen, wenn die erfolgreichen Zustandsübergänge abgespeichert und bei der nächsten Suche priorisiert werden.

4.3.3. Auswerten des Gesamtungleichungssystems

Eine Zusammenfassung der Ungleichungen aller nichtlinearen Elemente liefert das Gesamtungleichungssystem

$$\mathbf{L}_n \begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix} \geq \mathbf{r}_n \quad (4.72)$$

für eine Zustandskombination n . Der Spaltenvektor der Spannungen und Ströme aller Elemente kann über eine Ausgangs- bzw. eine Durchgangsmatrix²⁰, \mathbf{C}_{pwl} bzw. $\tilde{\mathbf{D}}$, auf die Zustandsgrößen \mathbf{x}_+ bzw. die impulsbehafteten Quellengrößen $\tilde{\boldsymbol{\delta}}$ bezogen werden:

$$\mathbf{L}_n (\mathbf{C}_{\text{pwl}} \mathbf{x}_+ + \tilde{\mathbf{D}} \tilde{\boldsymbol{\delta}}) \geq \mathbf{r}_n \quad (4.73)$$

mit dem impulsbehafteten Quellenvektor

$$\tilde{\boldsymbol{\delta}} = (\tilde{\mathbf{C}}_+ \tilde{\mathbf{x}}_+ - \tilde{\mathbf{C}}_- \tilde{\mathbf{x}}_- + \Delta \tilde{\mathbf{q}}_i) \delta(t). \quad (4.74)$$

Zum Zeitpunkt²¹ $t = 0$ dominiert der impulsbehaftete Anteil $\tilde{\mathbf{D}} \tilde{\boldsymbol{\delta}}$ die Ungleichung (4.73), da ein Dirac-Impuls mit seiner unendlichen Höhe sämtliche Beiträge der Zustandsgrößen $\mathbf{C}_{\text{pwl}} \tilde{\mathbf{x}}_+$, unabhängig von deren Betrag und Vorzeichen, überwiegt [108]. Wird eine oder mehrere Ungleichungen verletzt, so werden die Zustände der Schaltungselemente, die dieser Ungleichung zugeordnet sind, entsprechend angepasst. Innerhalb einer übergeordneten Iterationsschleife wird die Zustandskombination so lange angepasst, bis entweder (4.73)

²⁰ Die Ausgangs- und Durchgangsmatrix werden aus den entsprechenden Zeilen (zweite Zeilengruppe) und Spalten (sechste und achte Spaltengruppe) der Gesamtmatrix (4.41) extrahiert.

²¹ Die Ungleichung wird ohne den impulsbehafteten Anteil auch am Ende eines Teilintervalls benötigt, was Bestandteil des Abschnitts 4.4 ist.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

erfüllt ist oder alle Möglichkeiten getestet sind und die Simulation abgebrochen werden muss (vgl. Abschnitt 4.6). Die Suche nach der korrekten Zustandskombination lässt sich beschleunigen, wenn erfolgreiche Übergänge abgespeichert und bei der nächsten Suche priorisiert werden.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

Die Zeitpunkte, an denen ein Segmentwechsel eines als stückweise linear definierten Schaltungselements stattfindet, sind für einige Schaltungselemente bereits vor Simulationsbeginn bekannt. Ein Beispiel dafür sind Pulsquellen, die eine vorher festgelegte Periodendauer, Einschaltzeit sowie Anstiegs- und Abfallzeit besitzen. Daneben gibt es jedoch auch Schaltungselemente, die ihren Zustand, d. h. das gültige Segment ihrer stückweise linearen Kennlinie, nicht in Abhängigkeit der Zeit, sondern abhängig von Spannungen und/oder Strömen des Netzwerks definieren.

4.4.1. Schaltungselemente mit zeitabhängigen Parametern

Schaltungselemente mit zeitabhängigen Parametern wechseln ihre Eigenschaften in Abhängigkeit der Simulationszeit. Ein häufig auftretendes, zeitabhängiges Schaltungselement ist die Pulsquelle. Diese besitzt mehrere Parameter $t_{\text{lim},n}$, die eine maximale Verweildauer innerhalb eines Zustands angeben. Sobald die Ungleichung

$$t_n \leq t_{\text{lim},n} \quad (4.75)$$

verletzt wird, springt das Element in den zeitlich darauffolgenden, bekannten Zustand. Die Zeit t_n gibt dabei die relative Zeitdauer seit Beginn eines jeweiligen Zustands wieder. Das Beispiel einer Pulsquelle mit der SPICE-Netzlisten-Syntax²²

```
V0 N1 N2 PULSE(5 -1 1u 0 0 2u 4u 2),
```

besitzt den in Abb. 4.7 dargestellten Zeitverlauf. Daneben zeigt die Abbildung noch die zugehörigen Zustandsnummern²³, hervorgehoben durch Kreise, die relative Zeitvariable t_n sowie einen Vektor $\mathbf{t}_{\text{lim},n}$ mit zwei Verweildauern. Läuft die obere der beiden Verweildauern ab, so wechselt die Quelle in den periodisch nächstgelegenen Zustand mit der Zustandsabfolge ① ① ② ③ ④ ① ② ③ ④ ① Läuft hingegen die untere, übergeordnete Verweildauer ab, so wechselt die Quelle in Zustand ⑤, für welchen eine unbegrenzte Verweildauer besteht. Im Beispiel der Pulsquelle besagt die Definition der Netzliste, dass

²² Eine Beschreibung dieser Syntax befindet sich in Abschnitt A.1 im Anhang.

²³ Die Zustände ① und ③ entsprechen der ansteigenden bzw. abfallenden Flanke, die im Beispiel jedoch unendlich steil und deshalb nicht sichtbar sind.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

die beiden Verweildauern mit maximaler und minimaler Spannung jeweils $2 \mu\text{s}$ andauern, die Quelle für $1 \mu\text{s}$ im initialen Zustand ① verharrt und nach insgesamt $2 \cdot 4 \mu\text{s} + 1 \mu\text{s} = 9 \mu\text{s}$ in den Zustand ⑤ eintritt, in dem keine Zeitbegrenzung mehr vorliegt. Nach jedem Ablauf einer Verweildauer müssen die noch andauernden Zeitbegrenzungen innerhalb des Vektors $\mathbf{t}_{\text{lim},n}$, die noch nicht abgelaufen sind, für das neue Intervall um die gerade abgelaufene Zeit gekürzt werden. Besitzt die Pulsquelle keine Pulsanzahl, so entfällt die zweite Zeile in $\mathbf{t}_{\text{lim},n}$.

4.4.2. Schaltungselemente mit intern gesteuerten Parametern

Schaltungselemente mit intern gesteuerten Parametern definieren ihren Zustand, d. h. das gültige Segment ihrer stückweise linearen Kennlinie, nicht in Abhängigkeit der Zeit, sondern abhängig von Spannungen und/oder Strömen des Netzwerks.

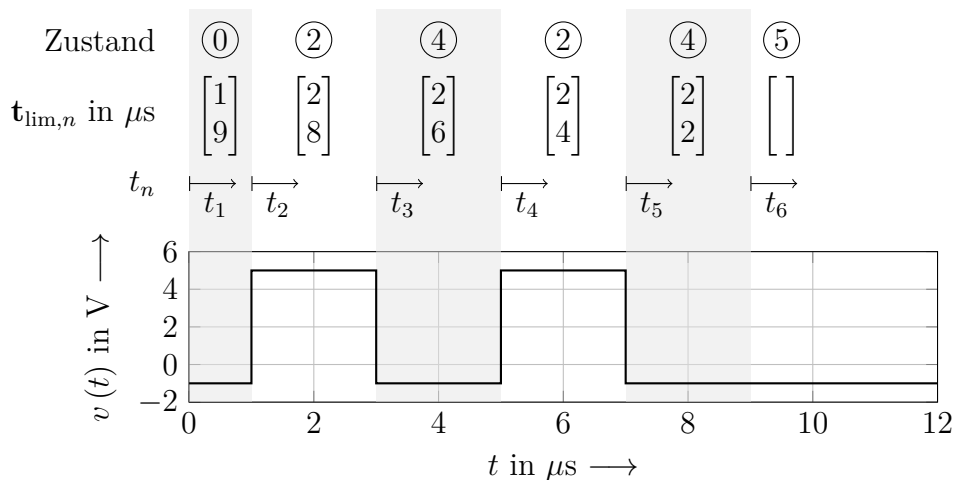


Abbildung 4.7.: Zeitverlauf einer Pulsspannungsquelle mit $1 \mu\text{s}$ Verzögerung und Begrenzung auf zwei Perioden

Ein Beispiel dafür sind ideale Dioden deren Leit- und Sperrphase von Strom (welcher stets größer oder gleich null sein muss) bzw. Spannung (welche stets kleiner oder gleich null sein muss) abhängig sind. Die Umschaltzeitpunkte zwischen diesen Phasen sind gekennzeichnet durch ein Verletzen der Ungleichungen

$$\mathbf{LC}_{\text{pwl}}\mathbf{x}(t) \geq \mathbf{r} \quad (4.76)$$

der als stückweise linear definierten Schaltungselemente²⁴. Dabei handelt es sich um die gleichen Ungleichungen wie in (4.73), allerdings ohne den impulsbehafteten Anteil²⁵. Auf

²⁴ In (4.76) und im Folgenden wird auf eine explizite Unterscheidung zwischen einer „lokalen“ Zeit t_n und der absoluten Zeit t verzichtet.

²⁵ Für eine bessere Übersichtlichkeit wird hier auf den Index „+“ aus (4.73) verzichtet.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

den zustandskennzeichnenden Index n aus (4.73) wird im Folgenden ebenfalls verzichtet. Gesucht wird nun der früheste Zeitpunkt t_{\min} , an dem mindestens eines der Schaltungselemente die Bereichsgrenzen seines momentanen Zustands verlässt. Mit dem Zeilenvektor zeitabhängiger Funktionen

$$\mathbf{f}(t) = [f_1(t) \quad f_2(t) \quad \cdots \quad f_k(t)]^T = \mathbf{LC}_{\text{pwl}} e^{\mathbf{A}t} \mathbf{x}_0 - \mathbf{r} \quad (4.77)$$

ergibt sich der gesuchte Zeitpunkt t_{\min} zu

$$t_{\min} = \min_{1 \leq j \leq k} \left\{ t_j \mid f_j(t) \stackrel{!}{=} 0, t \geq 0 \right\}. \quad (4.78)$$

Das vorliegende Problem nach (4.78) entspricht demnach der Nullstellensuche an univariaten, reellen Funktionen, wobei hier ausschließlich der früheste Zeitpunkt von Interesse ist.

4.4.3. Nullstellensuche mit hochauflösender Abtastung

Transiente Schaltungssimulatoren, die auf impliziten, numerischen Integrationsverfahren basieren (vgl. Abschnitt 2.2.2), wie etwa SPICE, orientieren sich bei der Wahl ihres Zeitschritts an einer vom Anwender vorgegebenen relativen und absoluten Genauigkeit [101]. Der Zeitschritt ist nicht, wie bei expliziten Integrationsverfahren, aufgrund numerischer Instabilität begrenzt. Lässt sich der Zeitverlauf gut mit Hilfe eines Polynoms zweiten Grades beschreiben, sind mit dem häufig verwendeten Trapezverfahren große Schrittweiten möglich [4]. Dies hat allerdings zur Folge, dass Nulldurchgänge nicht immer zuverlässig entdeckt werden, wie das Beispiel des Spitzenwertgleichrichters aus Unterabschnitt 2.2.6 zeigt. Auch Schaltungssimulatoren, die statt eines impliziten Integrationsverfahrens die Matrixexponentialfunktion verwenden, müssen sich dieser Problematik stellen. Obwohl die Matrixexponentialfunktion prinzipiell enorme Schrittweiten unter Beibehaltung hoher Genauigkeit erlaubt, müssen dennoch eventuelle Segmentwechsel nichtlinearer Schaltungselemente zuverlässig erkannt werden. Daher setzen die meisten dieser Schaltungssimulatoren eine hochauflösende Abtastung des Zeitverlaufs ein. Ein effizienter und genauer Algorithmus zur Berechnung äquidistanter Abtastwerte zeigt bspw. [105]. Sobald die Abtastwerte einen Vorzeichenwechsel aufweisen, muss - aufgrund des Zwischenwertsatzes [117] - im Zeitintervall zwischen dem aktuellen und dem vorherigen Abtastzeitpunkt mindestens eine Nullstelle liegen. Doch im Gegensatz zu der in SPICE eingesetzten Fehlerabschätzung orientiert sich die Wahl des Zeitschritts in den meisten Fällen an der kleinsten Zeitkonstanten im System [78, 83] oder einer vom Benutzer vorgegebenen Schrittweite [75]. Damit unterliegt dieser Ansatz jedoch der bereits im Zusammenhang mit expliziten Integrationsverfahren erwähnten „time-constant barrier“ [90] (vgl. Abschnitt 2.2.2). Je kleiner also die Zeitkonstante, desto enger müssen die Abtastzeitpunkte gewählt werden. Einige Ansätze versuchen diese Problematik dadurch zu umgehen, dass extrem kleine Zeitkonstanten, wie beispielsweise diejenigen, die durch parasitäre Komponenten hervorgerufen werden, bei

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

der Wahl des Zeitschritts nicht berücksichtigt werden [86, 87]. Zusätzlich können Zeitkonstanten, die im Zusammenhang mit exponentiell abklingenden Lösungsanteilen stehen, mit fortschreitender Zeitdauer vernachlässigt werden [143].

Mittels der hochauflösenden Abtastung ist es möglich, eine oder mehrere Nullstellen innerhalb eines Intervalls einzugrenzen. Allerdings ist der verwendete Zeitschritt und damit die Intervallweite für die meisten Anforderungen zu grob, sodass es einer weiteren, feineren zeitlichen Auflösung des Nulldurchgangs bedarf. Eine effiziente Möglichkeit bietet die Bisektion mit Abspeichern der Transitionsmatrix [144] für festgelegte Zeitintervalle [78, 99, 145]. Das kleinste dieser Zeitintervalle wird dabei gerade so groß gewählt, dass der Zeitverlauf innerhalb dieses Intervalls mit Hilfe eines quadratischen oder linearen Polynoms bis auf Maschinengenauigkeit approximiert werden kann [78]. Das Newton-Raphson-Verfahren kann als Ergänzung zur Bisektion dienen [75, 76, 79, 80, 146]. Aufgrund seiner Konvergenzproblematik (vgl. Unterabschnitt 2.2.3) kann und sollte es jedoch erst bei kleineren Abtastintervallen zur Annäherung an die Nullstelle bis auf Maschinengenauigkeit eingesetzt werden [147]. Allerdings ist auch dann nicht garantiert, dass die gefundene Nullstelle tatsächlich die früheste ist.

Auch wenn die hochauflösende Abtastung ein effizientes und genaues Verfahren darstellt, eine zuverlässige Detektion von Nulldurchgängen ist damit nicht gesichert [143].

4.4.4. Nullstellensuche mit Polynominterpolation

Eine Alternative zur hochauflösenden Abtastung stellt die Interpolation des Zeitverlaufs über ein Polynom dar. Zusätzlich zu den diskreten Abtastzeitpunkten beinhaltet das Interpolationspolynom auch Informationen über den Funktionsverlauf dazwischen. Einen ausreichend hohen Polynomgrad vorausgesetzt, können damit auch deutlich größere Zeitschritte abgedeckt werden, als dies mit „hochauflösender Abtastung“ typischerweise möglich ist. Der Vorteil dieses Ansatzes hängt jedoch entscheidend von der Interpolationsmethode und des Nullstellensuchalgorithmus ab.

Innerhalb der zahlreichen Strategien zur Nullstellensuche von Polynomen höheren Grades ($m > 3$) erscheinen diejenigen, die ein „Branch-and-Bound“-Prinzip²⁶ anwenden, als besonders vorteilhaft [121, 148–155]. Das „Branching“, also die Zerlegung in Teilintervalle, findet bereits bei der Reihenapproximation des Matrixexponentials (vgl. Unterabschnitt 4.4.4) statt. Das „Bounding“, also die Eingrenzung des Wertebereichs mittels geeigneter Schranken, ermöglicht eine Aussage, ob im jeweiligen Teilintervall eine Nullstelle auftreten kann. Der Vorteil besteht darin, dass geeignete Schranken oder Begrenzungsfunktionen deutlich schneller und einfacher ausgewertet werden können als die abzuschätzende Funktion (*hier*: ein Polynom hohen Grades) selbst. Im Folgenden werden obere Begrenzungsfunktionen bzw. der Zeilenvektor oberer Begrenzungsfunktionen durch einen

²⁶ Das „Branch-and-Bound“-Prinzip wird häufiger in globalen Optimierungsmethoden angewendet und findet nur vereinzelt Anwendung in Algorithmen zur Nullstellensuche.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

Überstrich

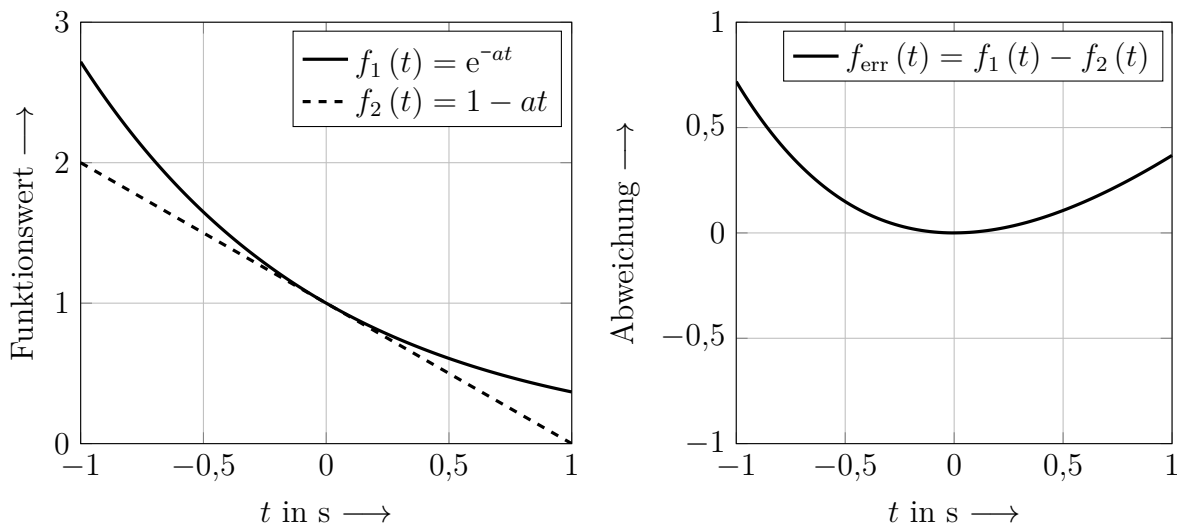
$$\bar{\mathbf{f}}(t) \geq \mathbf{f}(t) \quad (4.79)$$

und untere Begrenzungsfunktionen bzw. der Zeilenvektor unterer Begrenzungsfunktionen durch einen Unterstrich

$$\underline{\mathbf{f}}(t) \leq \mathbf{f}(t) \quad (4.80)$$

gekennzeichnet. Teilintervalle, die positiv auf eine mögliche Nullstelle getestet sind, werden in immer feinere Subintervalle zerlegt und wiederum mittels neu ermittelten, engeren Schranken auf Nullstellen geprüft.

Sowohl das „Branching“ als auch das „Bounding“ besitzen großen Einfluss auf die Konvergenzgeschwindigkeit und den Rechenaufwand des Verfahrens. Eine zu feine Zerlegung in Teilintervalle ermöglicht zwar typischerweise eine einfachere und engere Abschätzung der Funktion, allerdings müssen im ungünstigsten Fall auch sehr viele dieser Teilintervalle ausgewertet werden²⁷. Auf der anderen Seite ermöglichen möglichst enge Schranken eine bessere Lokalisierung der Nullstelle und damit ein gezielteres „Branching“, allerdings auf Kosten des Rechenaufwands, den eine solche genaue Begrenzungsfunktion typischerweise mit sich bringt.



(a) Verlauf der Funktion und Näherung (b) Abweichung von Funktion und Näherung

Abbildung 4.8.: Approximation der Funktion e^{-at} mit $a = 1 \text{ s}^{-1}$ über die ersten beiden Glieder ihrer Taylorreihe

Ein Kompromiss aus beidem stellt die am einfachsten auszuwertende Begrenzungsfunktion dar, die gerade noch ein gezieltes „Branching“ und damit eine gute Konvergenzgeschwindigkeit ermöglicht. Es erscheint daher naheliegend, die Taylorreihe nach dem linearen Term

²⁷ Die hochauflösende Abtastung aus Unterabschnitt 4.4.3 kann als ein Extremfall mit sehr feinem „Branching“ und einfachem bis vernachlässigbarem „Bounding“ angesehen werden.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

abzubrechen und für die Ermittlung der Schranke die Restglieder über einen konstanten Fehlerterm zusammenzufassen. Der Nachteil dieses Ansatzes lässt sich gut anhand des Beispiels aus Abb. 4.8a verdeutlichen: Selbst dann, wenn der Fehler (s. Abb. 4.8b), den die vernachlässigten Restglieder verursachen, exakt bestimmt werden könnte, approximieren die ersten Glieder einer Taylorreihe die zugrunde liegende Funktion nur in unmittelbarer Umgebung der Entwicklungsstelle (*hier*: $t = 0$) zufriedenstellend. Gleichzeitig wird aus Abb. 4.8a auch deutlich, dass die Funktion im dargestellten Intervall durch Verschieben des Taylorpolynoms und geringfügiger Anpassung der Steigung mit geringerer maximaler Abweichung genähert werden könnte. Die Bestimmung dieses Minimax-Polynoms, d. h. dem Polynom mit der geringsten maximalen Abweichung, ist im Allgemeinen jedoch sehr aufwendig [156]. Einen Kompromiss aus Bestimmungsaufwand und möglichst geringer maximaler Abweichung stellen die sogenannten Tschebyscheff-Polynome dar.

Tschebyscheff-Polynome

Aus Unterabschnitt 4.1.1 ist bereits bekannt, dass die Matrixexponentialfunktion als Taylorreihe

$$e^{\mathbf{A}t} = \mathbf{1} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \frac{1}{3!}\mathbf{A}^3t^3 + \dots \quad (4.81)$$

beschrieben werden kann. Allerdings approximiert die Taylorreihe bei vorgegebener Anzahl an Gliedern die zugrunde liegende Funktion nur in unmittelbarer Umgebung der Entwicklungsstelle (die in Abb. 4.8 bei $t = 0$ liegt) zufriedenstellend. Eine deutlich bessere Interpolation erlauben die Tschebyscheff-Polynome. Diese stellen einen Kompromiss zwischen Approximationsgüte (besser als Taylor, jedoch schlechter als das Minimax-Polynom) und Berechnungsaufwand (höher als bei Taylor, jedoch geringer als beim Minimax-Polynom) dar. Die (verschobenen) Tschebyscheff-Polynome²⁸ lauten

$$T_0^*(x) = 1 \quad (4.82a)$$

$$T_1^*(x) = 2x - 1 \quad (4.82b)$$

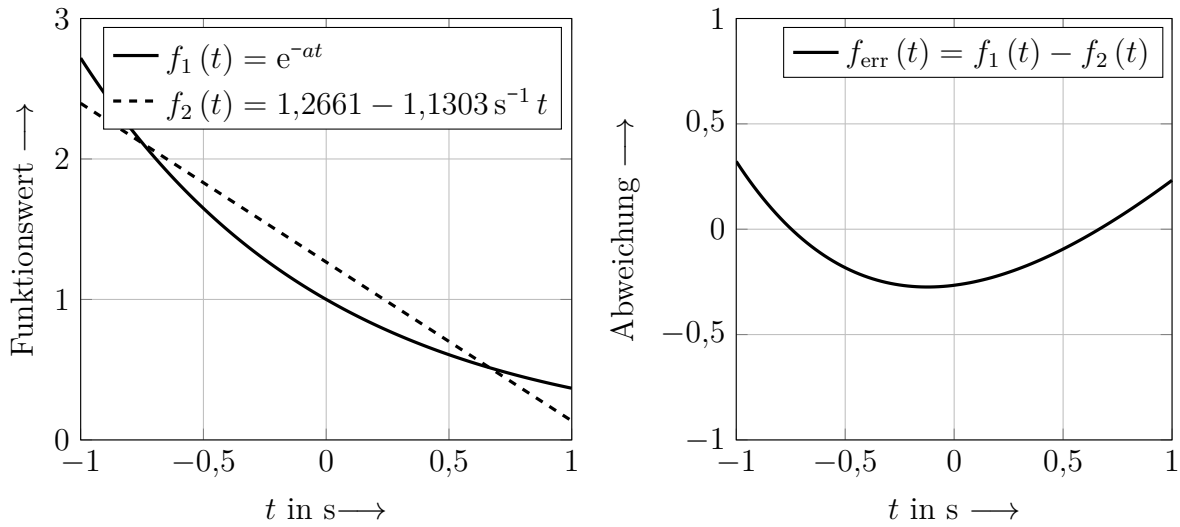
⋮

$$T_{n+1}^*(x) = 2(2x - 1)T_n^*(x) - T_{n-1}^*(x). \quad (4.82c)$$

Die Zeitverläufe und zugehörigen Abweichungen aus Abb. 4.8 bzw. Abb. 4.9 ermöglichen einen direkten Vergleich zwischen einer Taylor-Approximation ersten Grads und einer Tschebyscheff-Interpolation ersten Grads. Für die gleiche zu interpolierende Funktion und das identische Intervall besitzt die Tschebyscheff-Interpolation bei gleicher Polynomordnung eine deutlich geringere maximale Abweichung. Dies gilt nicht nur für die hier

²⁸ Wenn in dieser Arbeit von Tschebyscheff-Polynomen gesprochen wird, sind, falls nicht anders vermerkt, immer verschobene Tschebyscheff-Polynome erster Art gemeint. Der hochgestellte Stern „*“ in (4.82) kennzeichnet die Verschiebung der Tschebyscheff-Polynome vom Intervall zwischen -1 und 1 nach $0 \leq x \leq 1$.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels



(a) Verlauf der Funktion und Näherung

(b) Abweichung von Funktion und Näherung

Abbildung 4.9.: Approximation der Funktion e^{-at} mit $a = 1 s^{-1}$ über die ersten beiden Glieder ihrer Tschebyscheffentwicklung

verwendete, beispielhafte Funktion, sondern ist eine generelle Eigenschaft der Entwicklung in Tschebyscheff-Polynomen [96, 156].

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Tschebyscheff-Entwicklung der Matrixexponentialfunktion

Die Interpolation der Matrixexponentialfunktion mittels verschobener Tschebyscheff-Polynome erfordert zunächst eine Skalierung der Systemmatrix

$$\mathbf{A}_h = \mathbf{A}h \quad (4.83)$$

und der Zeitvariablen

$$t_h = \frac{t}{h}, \quad (4.84)$$

um das Zeitintervall zwischen $t = 0$ und einem maximalen Zeitschritt $t = h$ in den für die verschobenen Tschebyscheff-Polynome relevanten Wertebereich $0 \leq t_h \leq 1$ abzubilden. Die Matrixexponentialfunktion lässt sich dann über eine Reihenentwicklung in Tschebyscheff-Polynomen²⁹

$$e^{\mathbf{A}t} = e^{\mathbf{A}_h t_h} = \frac{1}{2} \hat{\mathbf{E}}_0 T_0^*(t_h) + \sum_{n=1}^{\infty} \hat{\mathbf{E}}_n T_n^*(t_h) = \sum_{n=0}^{\infty} \hat{\mathbf{E}}_n T_n^*(t_h) \quad (4.85)$$

für den Zeitbereich $0 \leq t \leq h$ darstellen [137]. Eine Auswertung der Summe aus (4.85) in Gleitkommarithmetik führt dazu, dass es ab einer gewissen Ordnung m keinen Unterschied mehr macht, ob die Koeffizienten von höherem Grad berücksichtigt werden oder nicht. Die Reihe kann dann nach dem m -ten Glied abgebrochen werden und die Matrixexponentialfunktion über

$$e^{\mathbf{A}_h t_h} \approx \sum_{n=0}^m \hat{\mathbf{E}}_n T_n^*(t_h) \quad (4.86)$$

approximiert werden. Die Koeffizientenmatrizen $\hat{\mathbf{E}}_n$ in (4.86) können auf verschiedenen Wegen gewonnen werden [96, 137]. Das lineare Matrix-Gleichungssystem

$$\begin{bmatrix} \frac{1}{2} \mathbf{1} & -\mathbf{1} & \mathbf{1} & -\mathbf{1} & \cdots \\ -\mathbf{A}_h & 4\mathbf{1} & \mathbf{A}_h & & \\ & -\mathbf{A}_h & 8\mathbf{1} & \ddots & \\ & & \ddots & \ddots & \mathbf{A}_h \\ & & & -\mathbf{A}_h & 4m\mathbf{1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{E}}_0 \\ \hat{\mathbf{E}}_1 \\ \vdots \\ \vdots \\ \hat{\mathbf{E}}_m \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (4.87)$$

bietet dazu eine einfache Möglichkeit [137]. Als Anhaltspunkt zur Bestimmung der notwendigen Anzahl an Gliedern m und der Intervalldauer h können die gleichen Werte wie bei der Taylorreihenentwicklung aus [105] dienen.

²⁹ Im Zusammenhang mit der Tschebyscheff-Entwicklung wird häufig das um einen hochgestellten Apostroph ergänzte Summenzeichen $\sum_{n=0}^{\prime\infty}$ verwendet, um anzuzeigen, dass das erste Element der Summe mit ein halb multipliziert wird.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

Bis zu einem Polynomgrad kleiner vier ($m < 4$) können Nullstellen einer Polynominterpolation direkt ausgewertet werden [156]. Eine möglichst genaue Interpolation mit ausschließlich den ersten vier Tschebyscheffpolynomen erlaubt allerdings nur einen vergleichsweise kleinen Zeitschritt h [157, 158]. Nullstellen von Polynomen höheren Grads können allerdings nur noch mit Hilfe numerischer Verfahren gefunden werden [156]. Ein Nachteil dabei ist jedoch der hohe Aufwand dieser Algorithmen, wenn man bedenkt, dass lediglich die erste Nullstelle des Polynoms von Interesse ist. Besonders schlecht wird das Aufwand-zu-Nutzen-Verhältnis, wenn das zu untersuchende Intervall keine Nullstelle aufweist. Algorithmen, die das sogenannte „Branch-and-Bound“-Prinzip anwenden, erscheinen daher vorteilhaft. Bevor ein zu untersuchendes Intervall einer aufwendigeren Untersuchung unterzogen wird, entscheidet eine Abschätzung des Wertebereichs („Bounding“), ob innerhalb des Intervalls eine Nullstelle vorhanden sein kann. Dazu sind allerdings möglichst enge und dennoch einfach auszuwertende Abschätzungen des Wertebereichs, sogenannte Schranken, nötig.

Abschätzen des Wertebereichs eines Polynoms („Bounding“)

Die Entwicklung in Tschebyscheff-Polynomen besitzt, neben vielen anderen vorteilhaften Eigenschaften (s. [156, 159]), die Möglichkeit den maximalen Abbruchfehler mit geringer Überabschätzung zu bestimmen. Möglich wird dies, da für den Wertebereich eines jeden Tschebyscheff-Polynoms $|T_n^*(x)| \leq 1$ im Intervall $0 \leq x \leq 1$ gilt. Eine konservative Abschätzung des Abbruchfehlers kann also durch Aufsummieren der Beträge der vernachlässigten Koeffizienten geschehen. Wird demnach die Tschebyscheff-Reihenentwicklung der Matrixexponentialfunktion mit den nach (4.87) berechneten Koeffizienten nach dem zweiten Glied abgebrochen, dann ist eine konservative Abschätzung des Betrags des Abbruchfehlers durch

$$\mathbf{R}_2 = \sum_{n=2}^m |\hat{\mathbf{E}}_n| \quad (4.88)$$

gegeben³⁰. Die obere und untere Schranke einer linearen Approximation lässt sich daher mit

$$e^{\mathbf{A}_h t_h} \leq \frac{1}{2} \hat{\mathbf{E}}_0 \mathbf{T}_0^* + \hat{\mathbf{E}}_1 \mathbf{T}_1^*(t_h) + \mathbf{R}_2 \quad (4.89a)$$

$$e^{\mathbf{A}_h t_h} \geq \frac{1}{2} \hat{\mathbf{E}}_0 \mathbf{T}_0^* + \hat{\mathbf{E}}_1 \mathbf{T}_1^*(t_h) - \mathbf{R}_2 \quad (4.89b)$$

³⁰ Die Betragsstriche in (4.88) und an jeder anderen Stelle in dieser Arbeit entsprechen der elementweisen Betragsbildung.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

angeben³¹. Einsetzen der Schrankenfunktionen (4.89) in (4.77) liefert die für das „Bounding“ benötigten oberen und unteren Grenzen

$$\bar{\mathbf{f}}(t_h) = \mathbf{LC}_{\text{pwl}} \left(\frac{1}{2} \hat{\mathbf{E}}_0 \mathbf{x}_0 \mathbf{T}_0^* + \hat{\mathbf{E}}_1 \mathbf{x}_0 \mathbf{T}_1^*(t_h) \right) + \sum_{n=2}^m \left| \mathbf{LC}_{\text{pwl}} \hat{\mathbf{E}}_n \mathbf{x}_0 \right| - \mathbf{r} \quad (4.90a)$$

$$\underline{\mathbf{f}}(t_h) = \mathbf{LC}_{\text{pwl}} \left(\frac{1}{2} \hat{\mathbf{E}}_0 \mathbf{x}_0 \mathbf{T}_0^* + \hat{\mathbf{E}}_1 \mathbf{x}_0 \mathbf{T}_1^*(t_h) \right) - \sum_{n=2}^m \left| \mathbf{LC}_{\text{pwl}} \hat{\mathbf{E}}_n \mathbf{x}_0 \right| - \mathbf{r}. \quad (4.90b)$$

Die Auswertung dieser Grenzen kann auf Kosten der Überabschätzung beschleunigt werden, wenn die Multiplikation mit dem Vektor der Anfangswerte erst nach der Summation stattfindet³²:

$$\bar{\mathbf{f}}(t_h) = \mathbf{LC}_{\text{pwl}} \left(\frac{1}{2} \hat{\mathbf{E}}_0 \mathbf{x}_0 \mathbf{T}_0^* + \hat{\mathbf{E}}_1 \mathbf{x}_0 \mathbf{T}_1^*(t_h) \right) + \left(\sum_{n=2}^m \left| \mathbf{LC}_{\text{pwl}} \hat{\mathbf{E}}_n \right| \right) |\mathbf{x}_0| - \mathbf{r} \quad (4.91a)$$

$$\underline{\mathbf{f}}(t_h) = \mathbf{LC}_{\text{pwl}} \left(\frac{1}{2} \hat{\mathbf{E}}_0 \mathbf{x}_0 \mathbf{T}_0^* + \hat{\mathbf{E}}_1 \mathbf{x}_0 \mathbf{T}_1^*(t_h) \right) - \left(\sum_{n=2}^m \left| \mathbf{LC}_{\text{pwl}} \hat{\mathbf{E}}_n \right| \right) |\mathbf{x}_0| - \mathbf{r} \quad (4.91b)$$

Durch diesen Schritt müssen zur Auswertung der oberen bzw. unteren Grenze insgesamt nur drei statt $m + 1$ Matrix-Vektor-Multiplikationen durchgeführt werden³³. Während der transienten Schaltungssimulation tritt typischerweise eine wiederholte Auswertung der oberen bzw. unteren Schrankenfunktion für unterschiedliche Anfangswertvektoren auf. Daher ist es vorteilhaft die Schrankenfunktionen in der Form

$$\bar{\mathbf{f}}(t_h, \mathbf{x}_0, |\mathbf{x}_0|) = \mathbf{V} \mathbf{x}_0 + \mathbf{U} \mathbf{x}_0 t_h + \mathbf{V}_{\text{abs}} |\mathbf{x}_0| - \mathbf{r} \quad (4.92a)$$

$$\underline{\mathbf{f}}(t_h, \mathbf{x}_0, |\mathbf{x}_0|) = \mathbf{V} \mathbf{x}_0 + \mathbf{U} \mathbf{x}_0 t_h - \mathbf{V}_{\text{abs}} |\mathbf{x}_0| - \mathbf{r} \quad (4.92b)$$

aufzustellen. Sind einer oder beide Funktionswerte einer unteren Schrankenfunktion \underline{f}_j zum Zeitpunkt $t_h = 0$ bzw. $t_h = 1$ negativ, so kann es innerhalb des Intervalls $0 \leq t_h \leq 1$ zu einer oder mehrerer Nullstellen der abgeschätzten Funktion kommen³⁴. In diesem Fall muss das Intervall weiter untersucht werden und es wird in kleinere Teilintervalle zerlegt („Branching“). Im gegenteiligen Fall, d. h. keiner der beiden Funktionswerte aller unteren Schrankenfunktionen ist negativ, kann keine Nullstelle auftreten und das betrachtete Intervall wird verworfen.

Besitzt eine der oberen Schrankenfunktionen \bar{f}_j im betrachteten Intervall eine Nullstelle, so findet dort mit Sicherheit auch ein Zustandswechsel der zugehörigen Schaltungskomponente statt. Auch diese Nullstelle kann dazu verwendet werden, das neue Suchintervall zu verkleinern.

³¹ Auch hier gilt: Sowohl „ \leq “ in (4.89a) als auch „ \geq “ in (4.89a) sind elementweise anzuwenden.

³² An dieser Stelle sei darauf hingewiesen, dass die elementweise Betragsfunktion submultiplikativ ist, d. h. sie erfüllt die Bedingung $|\mathbf{A}\mathbf{x}| \leq |\mathbf{A}||\mathbf{x}|$.

³³ Dies gilt nur unter der Annahme, dass alle Matrizenprodukte in (4.91), die unabhängig vom Vektor der Anfangswerte \mathbf{x}_0 sind, bereits berechnet und abgespeichert sind.

³⁴ Durch die Formulierung der PWL-Ungleichungen nach (4.72) ist sichergestellt, dass der Startwert stets positiv ist, d. h. es gilt $\bar{\mathbf{f}}(0, \mathbf{x}_0, |\mathbf{x}_0|) > 0$.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

Sobald die Intervallbreite h eine zuvor festgelegte Grenze unterschreitet oder das gesamte Intervall auf Nullstellenfreiheit geprüft wurde, bricht der Suchalgorithmus ab. Neben den beiden Fällen einer garantierten Nullstelle und der Nullstellenfreiheit kann es vorkommen, dass zwar die untere Schrankenfunktion eine mögliche Nullstelle anzeigt, die obere jedoch keinen Nulldurchgang aufweist und damit eine Nullstelle nicht garantiert werden kann. In besonders schwierigen Fällen hilft dann auch keine weitere Zerteilung des Intervalls, da die Genauigkeit der Berechnungen in Gleitkommaarithmetik bereits an ihre Grenzen stoßen. Diese Fälle müssen kein Grund sein, die Simulation abzubrechen, jedoch sollten sie dem Anwender mitgeteilt werden.

Bestimmung des nächsten Suchintervalls („Branching“)

Die einfachste Form des „Branching“ besteht in der Bisektion. Bisektion bedeutet, dass das ursprüngliche Suchintervall, nachdem es positiv auf eine mögliche Nullstelle getestet wurde, in genau zwei Teilintervalle aufgeteilt wird, welche dann wiederum einem „Bounding“ unterzogen werden. Bisektion garantiert zwar auf der einen Seite ein zuverlässiges Auffinden der ersten Nullstelle, allerdings auf Kosten der Konvergenzgeschwindigkeit, d. h. der Geschwindigkeit mit der die Breite des Suchintervalls innerhalb eines Iterationsschritt verringert werden kann. Die Intervallweiten einer Bisektion nehmen mit jedem Iterationsschritt linear ab. Das Newton-Raphson-Verfahren³⁵ besitzt hingegen eine zwar lokal begrenzte, jedoch quadratische Konvergenzordnung [156]. Im Folgenden soll daher eine Kombination aus beiden Verfahren auf die vorliegende Problemstellung angewendet werden.

Die Wahl linearer Schrankenfunktionen aus dem vorhergehenden Abschnitt wurde bereits in Voraussicht eines möglichst effizienten „Branching“ gewählt. Besitzt eine oder beide der Schrankenfunktionen im betrachteten Intervall Nulldurchgänge, so können diese sehr einfach und schnell ausgewertet und das Ergebnis zur Eingrenzung des neuen Suchintervalls verwendet werden. Dies entspricht dem Prinzip des Newton-Raphson-Verfahrens. Besitzen die Schrankenfunktionen allerdings keine Nulldurchgänge so wird die Bisektion angewendet. Dabei wird bei sequentieller Abarbeitung stets das linke (zeitlich frühere) Intervall zuerst ausgewertet.

Effiziente Wahl der Zeitschritte und Intervallweiten

Anstatt die Zeitpunkte des vorhergehenden „Branching“-Schritts direkt zu übernehmen ist es effizienter, diese an einem Zeitraster auszurichten. Wird das Gesamtintervall der Länge h in 2^N mit $N \in \mathbb{N}$ gleichlange Teilintervalle zerlegt, so sind zur Berechnung des Zustandsvektors für jeden beliebigen Rasterzeitpunkt maximal N Matrix-Vektor-Produkte nötig [78].

³⁵ Das herkömmliche Newton-Raphson-Verfahren ohne Über- bzw. Unterabschätzung scheidet hier allerdings aus, da es keine Konvergenz in die früheste (erste) Nullstelle sicherstellt.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Die notwendigen Transitionsmatrizen der N Zeitschritte der Länge $h/2^n$ mit $n \in 1 \dots N$ werden bei erstmaligem Bedarf berechnet und anschließend zum wiederholten Gebrauch abgespeichert. Auch die Matrizen der Schrankenfunktionen müssen für jeden der N Zeitschritte berechnet und abgespeichert werden. Das hier beschriebene Vorgehen nutzt die Tatsache aus, dass die transiente Simulation getakteter, leistungselektronischer Schaltungen mit Hilfe der stückweise linearen Approximation, eine sich ständig wiederholende Abfolge einiger (vergleichsweise weniger) Systemzustände ist. Im Gegensatz dazu führt die lokale lineare Approximation (wie bspw. in SPICE) zu einer unüberschaubaren Vielzahl an Systemzuständen, für die eine Abspeicherung unzweckmäßig wäre.

Ein einfaches Beispiel

Gegeben sei die Funktion

$$f(t) = -\sin(\omega t) - t + 1,5 \quad \text{mit} \quad \omega = \frac{2\pi}{0,75} \frac{\text{rad}}{\text{s}} \quad (4.93)$$

deren erste Nullstelle gesucht wird. Sie ist die Lösung des Differentialgleichungssystems mit der Systemmatrix

$$\mathbf{A} = \begin{bmatrix} 0 & \omega & 0 & 0 \\ -\omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (4.94)$$

der Ausgangsmatrix

$$\mathbf{C}_{\text{pwl}} = \begin{bmatrix} -0,5 & 0,5 & -1 & 2,5 \end{bmatrix}, \quad (4.95)$$

der PWL-Ungleichungsmatrix

$$\mathbf{L} = [1] \quad (4.96)$$

und dem Vektor der Anfangswerte

$$\mathbf{x}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T. \quad (4.97)$$

Eine Tschebyscheffentwicklung mit den ersten 24 verschobenen Tschebyscheffpolynomen reicht aus, um (4.93) im Intervall $0 \leq t \leq 1$ s bis auf Maschinengenauigkeit zu approximieren³⁶. Die zugehörige obere und untere Schrankenfunktion nach (4.90) lautet

$$\bar{f}(t) = 1,3073 - 1,2696 \text{ s}^{-1} t + 2,2508 \quad (4.98a)$$

$$\underline{f}(t) = 1,3073 - 1,2696 \text{ s}^{-1} t - 2,2508 \quad (4.98b)$$

³⁶ Die genaue Anzahl der notwendigen Polynome stammt aus einer Auswertung mit der MATLAB-Bibliothek „chebfun“, erhältlich unter www.chebfun.org.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

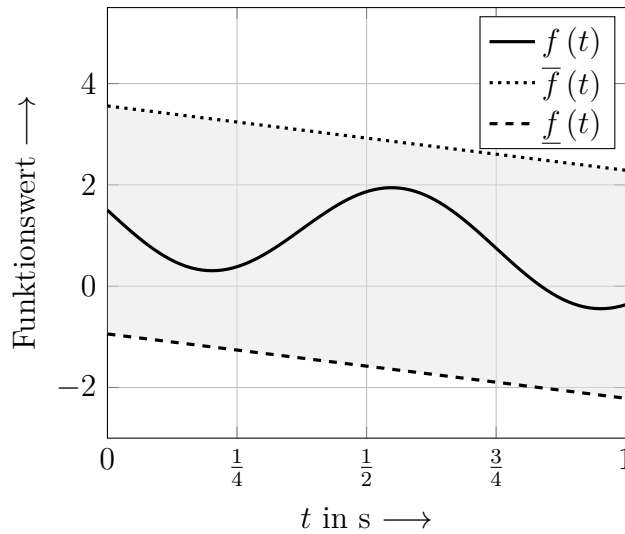


Abbildung 4.10.: Beispielfunktion (4.93) mit oberer und unterer Schranke (4.98a) bzw. (4.98b)

Eine Auswertung der unteren Schranke (4.98b) (s. Abb. 4.10) zeigt, dass im betrachteten Zeitintervall $0 \leq t \leq 1$ s eine Nullstelle auftreten kann, jedoch nicht muss, da die obere Schranke (4.98a) keinen Nulldurchgang aufweist. Das Intervall muss demnach in weiteren Iterationen genauer untersucht werden. Da weder die untere noch die obere Schranke einen Nulldurchgang besitzen, ist ein gezieltes „Branching“ nicht möglich, sodass hier auf Bisektion ausgewichen wird. Eine Teilung des Intervalls bei $t_b = \frac{1}{2}$ s führt zu zwei neuen Intervallen. Da ausschließlich die erste (früheste) Nullstelle von Interesse ist, wird stets das früheste Intervall zuerst ausgewertet. Die Schranken³⁷ des neuen Teilintervalls $t_a \leq t \leq t_b$ mit $t_a = 0$ lauten

$$\bar{f}_{2,a}(t_{2,a}) = 0,7841 + 0,6377t_{2,a} + 0,9845 \quad (4.99a)$$

$$\underline{f}_{2,a}(t_{2,a}) = 0,7841 + 0,6377t_{2,a} - 0,9845. \quad (4.99b)$$

Die beiden durch ein Komma getrennten Indizes in (4.99) geben zum einen Aufschluss über die Intervallbreite und zum anderen über die zeitliche Verschiebung:

$$t_{x,y} = \frac{1}{h_x}(t - t_y) \quad (4.100)$$

$$h_x = \frac{1}{x} \text{ s.} \quad (4.101)$$

³⁷ Aufgrund der kleineren Intervallbreite reichen nun weniger als die anfänglichen 24 Polynomglieder aus. Mit zunehmend geringerer Intervallbreite sind im Allgemeinen auch zunehmend weniger Glieder der Tschebyscheffentwicklung nötig.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

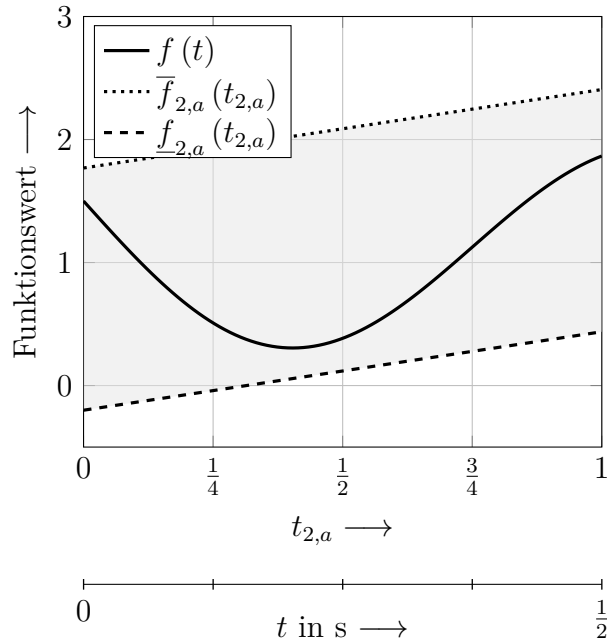


Abbildung 4.11.: Obere und untere Schranke (4.99a) bzw. (4.99b) mit skaliertem Zeitachse $t_{2,a}$ sowie die ursprüngliche Funktion $f(t)$ auf der Zeitachse t

Eine Auswertung der unteren Schranke (4.99b) in Abb. 4.11 zeigt, dass im betrachteten Zeitintervall eine Nullstelle auftreten kann, jedoch nicht muss, da die obere Schranke keinen Nulldurchgang aufweist. Im Gegensatz zum vorherigen Intervall besitzt die untere Schranke nun jedoch einen Nulldurchgang. Das neue zu untersuchende Intervall würde daher zwischen $t_a = 0$ und dem Nulldurchgang bei $t \approx 0,1571$ s zu liegen kommen. Wie jedoch in Abschnitt 4.4.4 erläutert, ist es aus Effizienzgründen vorteilhaft, die Intervallweiten auf die Werte $\frac{1}{2^N}$ mit $N \in \mathbb{N}$ zu beschränken. Die nächst größere Intervallweite liegt bei $h_4 = \frac{1}{4}$ s, womit das neue Intervall zwischen $t_a \leq t \leq t_c$ mit $t_c = 0,25$ s liegt. Die zugehörige obere und untere Schranke sind

$$\bar{f}_{4,a}(t_{4,a}) = 1,3106 - 1,1601t_{4,a} + 0,2609 \quad (4.102a)$$

$$\underline{f}_{4,a}(t_{4,a}) = 1,3106 - 1,1601t_{4,a} - 0,2609. \quad (4.102b)$$

Eine Auswertung der unteren Schranke (4.102a) in Abb. 4.12 zeigt, dass im betrachteten Zeitintervall eine Nullstelle nur im Bereich $0,2262 \text{ s} \leq t \leq 0,25 \text{ s}$ auftreten kann. Dieser wurde jedoch bereits im vorherigen Schritt ausgeschlossen (was in Abb. 4.12 durch ein gefülltes Rechteck auf der t -Achse gekennzeichnet ist).

Die bisherigen Auswertungen haben ergeben, dass im Intervall $0 \leq t \leq 0,5$ s keine Nullstelle auftreten kann. Als nächstes muss demnach das Intervall $0,5 \leq t \leq 1$ s untersucht werden.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

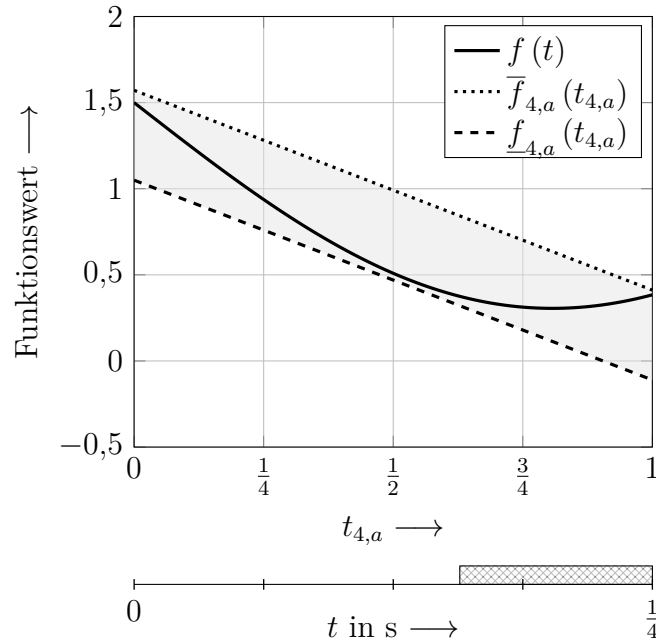


Abbildung 4.12.: Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $0 \leq t \leq 0,25$ s. Das gefüllte Rechteck markierte den Bereich, in dem zuvor ein Nulldurchgang ausgeschlossen werden konnte.

Die zugehörige obere und untere Schranke sind

$$\bar{f}_{2,b}(t_{2,b}) = 2,1377 - 2,7755t_{2,b} + 0,7217 \quad (4.103a)$$

$$\underline{f}_{2,b}(t_{2,b}) = 2,1377 - 2,7755t_{2,b} - 0,7217. \quad (4.103b)$$

Eine Auswertung der unteren Schranke (4.103b) in Abb. 4.13 zeigt, dass im betrachteten Zeitintervall eine Nullstelle nur im Bereich $0,7550 \text{ s} \leq t \leq 1 \text{ s}$ auftreten kann. Mit Berücksichtigung der zulässigen diskreten Intervallbreiten ergibt sich daher das neue zu untersuchende Intervall $t_d \leq t \leq 1 \text{ s}$ mit $t_d = 0,75 \text{ s}$. Die zugehörige obere und untere Schranke sind

$$\bar{f}_{4,d}(t_{4,d}) = 0,5606 - 1,1601t_{4,d} + 0,2609 \quad (4.104a)$$

$$\underline{f}_{4,d}(t_{4,d}) = 0,5606 - 1,1601t_{4,d} - 0,2609. \quad (4.104b)$$

Eine Auswertung der oberen und unteren Schranke in Abb. 4.14 zeigt, dass die Funktion eine oder mehrere Nullstellen im Bereich $0,8146 \leq t \leq 0,927$ besitzt. Eine effiziente Wahl des Rasterzeitpunkts und der Intervallweite führt auf das neue Intervall $t_e \leq t \leq t_e + \frac{1}{8}$ mit $t_e = \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}$. Die zugehörige obere und untere Schranke sind

$$\bar{f}_{8,e}(t_{8,e}) = 0,1328 - 0,6309t_{8,e} + 0,0703 \quad (4.105a)$$

$$\underline{f}_{8,e}(t_{8,e}) = 0,1328 - 0,6309t_{8,e} - 0,0703. \quad (4.105b)$$

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

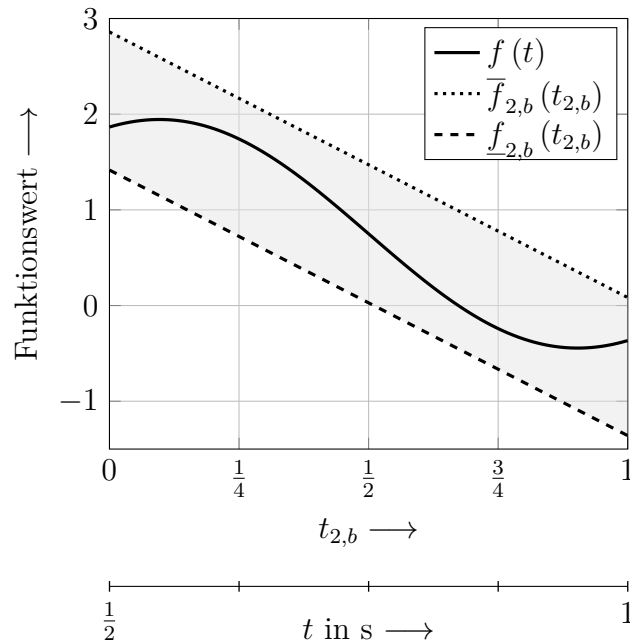


Abbildung 4.13.: Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $0,5\text{ s} \leq t \leq 1\text{ s}$

In allen folgenden Iterationsschritten verläuft die Suche nach dem gleichen Schema wie bereits gezeigt. Daher sei an dieser Stelle auf eine graphische Darstellung verzichtet und stattdessen die Ergebnisse einer jeden Iteration (einschließlich der bereits gezeigten) in Tab. 4.1 zusammengefasst. Die Suche wird nach der elften Iteration abgebrochen, da das Intervall danach kleiner ist, als es mit einer „double“-Genauigkeit darstellbar wäre.

Anhand der Ergebnisse aus Tab. 4.1 lässt sich gut die Geschwindigkeit, mit der sich die Intervallweite mit jeder Iteration verringert, abschätzen. Die Intervallweiten verringern sich ab der vierten Iteration im Mittel um eine Potenz von 1,69. Aufgrund der diskreten Rasterzeitpunkte und Intervalllängen (vgl. Abschnitt 4.4.4) liegt die Konvergenzgeschwindigkeit zwar nicht ganz bei der quadratischen Ordnung eines Newton-Raphson-Verfahrens, jedoch deutlich über der linearen Konvergenz einer Bisektion.

Im Vergleich zur hochauflösenden Abtastung aus Unterabschnitt 4.4.3 ist die Nullstellensuche mit Polynominterpolation und Schranken zuverlässiger. Allerdings unterliegt auch sie der „time-constant barrier“, denn die kleinste Zeitkonstante im System bestimmt die Wahl der maximalen Schrittweite bzw. die notwendige Gliederanzahl der Polynominterpolation.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

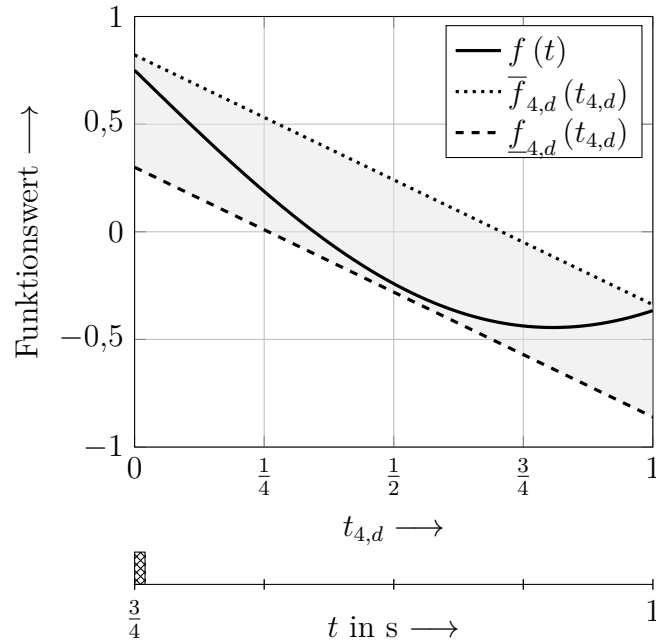


Abbildung 4.14.: Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $0,75 \text{ s} \leq t \leq 1 \text{ s}$. Das gefüllte Rechteck markiert den Bereich, in dem zuvor ein Nulldurchgang ausgeschlossen werden konnte.

Tabelle 4.1.: Ergebnisse aller Iterationen der Nullstellensuche für die exemplarische Funktion (4.93)

Iteration	Intervall in s	Weite in s
1	$0 \leq t \leq 1$	1
2	$0 \leq t \leq \frac{1}{2}$	$\frac{1}{2}$
3	$0 \leq t \leq \frac{1}{4}$	$\frac{1}{4}$
4	$\frac{1}{2} \leq t \leq 1$	$\frac{1}{2}$
5	$\frac{3}{4} \leq t \leq 1$	$\frac{1}{4}$
6	$\frac{13}{16} \leq t \leq \frac{15}{16}$	$\frac{1}{8}$
7	$\frac{13}{16} \leq t \leq \frac{14}{16}$	$\frac{1}{16}$
8	$\frac{53}{64} \leq t \leq \frac{54}{64}$	$\frac{1}{64}$
9	$\frac{1713}{2^{11}} \leq t \leq \frac{1714}{2^{11}}$	$\frac{1}{2^{11}}$
10	$\frac{438611}{2^{19}} \leq t \leq \frac{438612}{2^{19}}$	$\frac{1}{2^{19}}$
11	$\frac{114979377536}{2^{37}} \leq t \leq \frac{114979377537}{2^{37}}$	$\frac{1}{2^{37}}$

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

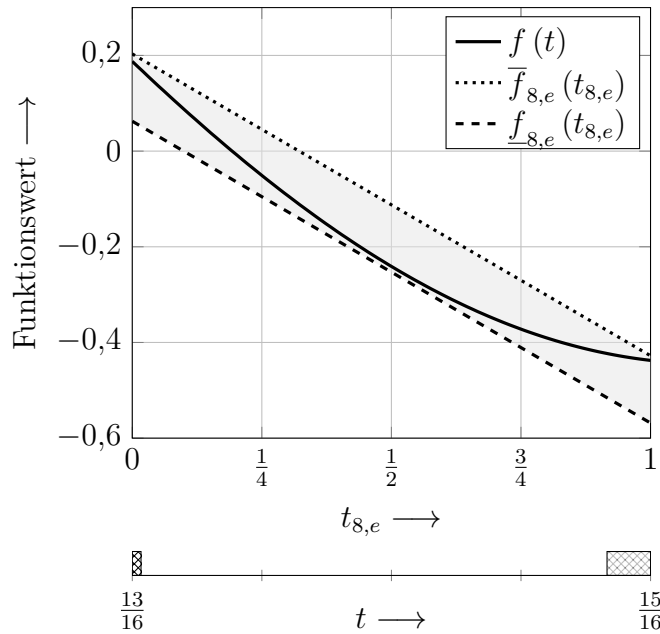


Abbildung 4.15.: Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $\frac{13}{16} \leq t \leq \frac{15}{16}$. Die gefüllten Rechtecke markieren den Bereich, in dem zuvor ein Nulldurchgang ausgeschlossen werden konnte.

4.4.5. Nullstellensuche mit Eigenwertverschiebung und Polynominterpolation

Ein Nachteil der im vorherigen Abschnitt gezeigten Nullstellensuche ist die Abhängigkeit der maximalen Schrittweite h von der kleinsten Zeitkonstanten im System („time-constant barrier“). Schaltungen mit sowohl langsamen als auch schnell oszillierenden Lösungsanteilen, beispielsweise hervorgerufen durch parasitäre Schwingkreise, können dadurch eine enorme Anzahl an notwendigen Zeitschritten aufweisen³⁸. Als Beispiel sei hier nochmals eine Funktion ähnlich der aus Abschnitt 4.4.4 herangezogen

$$f_{\text{osc}}(t) = -e^{\lambda t} \sin(\omega t) - 2s^{-1}t + 1,5. \quad (4.106)$$

Im Gegensatz zu (4.93) wird nun die Kreisfrequenz der Oszillation mit $\omega = 2\pi 10^3 \frac{\text{rad}}{\text{s}}$ deutlich größer als zuvor gewählt und zudem mit $\lambda = -16 \text{ s}^{-1}$ bedämpft. Der hochfrequente, oszillierende Anteil sorgt nun dafür, dass die grafische Auswertung des Zeitverlaufs in Abb. 4.16 nur noch als gefüllte schwarze Fläche erkennbar ist. Die einzelnen Perioden der Schwingung liegen so dicht beieinander, dass sie nicht mehr zu unterscheiden sind.

Eine Nullstellensuche mittels der in Abschnitt 4.4.3 oder Abschnitt 4.4.4 erläuterten Verfahren würde eine enorm hohe Anzahl an Suchintervallen benötigen. Angenommen

³⁸ Differentialgleichungssysteme dieser Art werden auch als oszillatorische Differentialgleichungssysteme bezeichnet [58, 160].

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

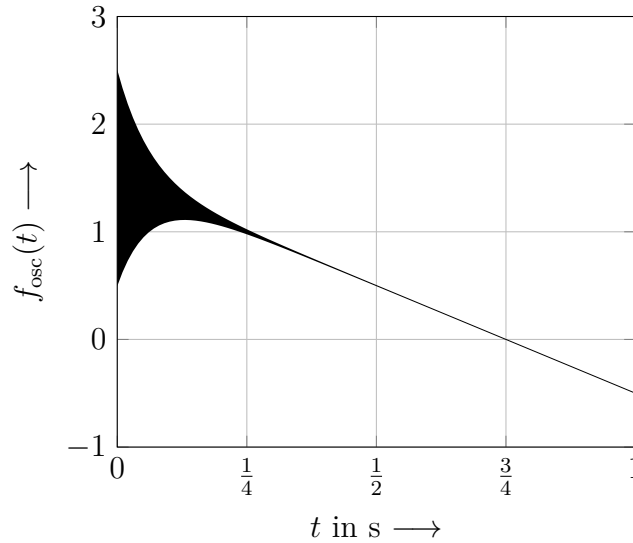


Abbildung 4.16.: Lösung (4.106) eines exemplarischen, gedämpft oszillatorischen Differentialgleichungssystems (4.106) mit deutlich kleinerer Oszillationsperiodendauer $T_{\text{osc}} = 10^{-3}\text{s}$ als dargestellter Intervallweite

der maximale Zeitschritt würde für beide Verfahren bei der halben Periodendauer der Oszillation liegen, d. h. $h = 0,5\text{ ms}$, so wären etwa 1500 Zeitschritte nötig, bis die Nullstelle erreicht wird. Ein Blick auf die zugrundeliegende Funktion (4.106) genügt jedoch, um festzustellen, dass wenn der oszillierende Anteil (unter Vernachlässigung der Dämpfung) nach unten mit -1 abgeschätzt wird, ein Nulldurchgang frühestens ab dem Zeitpunkt

$$0 = -1 - 2\text{s}^{-1}t + 1,5 \Leftrightarrow t = 0,25\text{ s} \quad (4.107)$$

eintreten kann. Durch diese einfache und grobe Abschätzung könnten 500 Zeitschritte eingespart werden. Das Ziel der im Folgenden vorgestellten Methode ist es demnach, das „Bounding“, also die Abschätzung der Lösung durch eine obere und untere Schranke, nicht direkt auf die Tschebyscheffentwicklung der Matrixexponentialfunktion anzuwenden. Stattdessen wird in einem vorgelagerten Schritt die Systemmatrix blockdiagonalisiert (vgl. Abschnitt 4.1.2). Dank der Darstellung in Blockdiagonalform besitzt die Methode Zugriff auf die Eigenwerte der Systemmatrix und kann dadurch die Lösungsfunktionen in der Form, wie sie in (4.106) vorliegen, ermitteln. Das „Bounding“ kann dadurch wesentlich effizienter gestaltet werden, da bekannte Lösungsfunktionen direkt und nicht über den Umweg einer Tschebyscheffentwicklung abgeschätzt werden können³⁹.

³⁹ Es gibt selbstverständlich noch andere Lösungsstrategien, als die hier vorgestellte. Eine Alternative ist die vollständige Unterdrückung „hochfrequenter“ Oszillationen [86,87]. In einem solchen Fall sollte das Simulationsprogramm den Anwender jedoch darüber informieren, dass Lösungsanteile vernachlässigt wurden.

Zusammensetzung der Gesamtlösung aus Blocklösungen

Die Matrixexponentialfunktion der Systemmatrix kann dank der Blockdiagonalisierung aus Abschnitt 4.1.2 auf die Matrixexponentialfunktion einer Blockdiagonalmatrix mit anschließender Rücktransformation reduziert werden:

$$e^{\mathbf{A}_h t_h} = \mathbf{S} e^{\mathbf{B} t_h} \mathbf{S}^{-1} \text{ mit } \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{B}_n \end{bmatrix}. \quad (4.108)$$

Um den Beitrag eines einzelnen Blocks zum Gesamtergebnis darzustellen, werden die Transformationsmatrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \cdots & \mathbf{S}_n \end{bmatrix} \quad (4.109)$$

und ihre Inverse

$$\mathbf{S}^{-1} = \begin{bmatrix} \tilde{\mathbf{S}}_1 & \tilde{\mathbf{S}}_2 & \cdots & \tilde{\mathbf{S}}_n \end{bmatrix}^T \quad (4.110)$$

in Spaltenvektoren (bzw. Spaltenvektorguppen) und Zeilenvektoren (bzw. Zeilenvektorguppen) partitioniert. Damit lässt sich das gesamte Matrixexponential als Summe der Matrixexponentialfunktionen jedes einzelnen Blocks darstellen:

$$e^{\mathbf{A}_h t_h} = \sum_{j=1}^n \mathbf{S}_j e^{\mathbf{B}_j t_h} \tilde{\mathbf{S}}_j. \quad (4.111)$$

Die beiden folgenden Unterabschnitte erläutern, getrennt nach reeller und imaginärer Eigenwertverschiebung, wie die Lösungen der einzelnen Blöcke abgeschätzt und zu einer Abschätzung der Gesamtlösung addiert werden können.

Eigenwertverschiebung um einen rein reellen Anteil

Mithilfe der Eigenwertverschiebung um den Mittelwert aller Eigenwerte eines Blocks $\bar{\lambda}_j$ (vgl. (4.26))

$$e^{\mathbf{B}_j t_h} = e^{(\bar{\lambda}_j \mathbf{1} + \mathbf{M}_j) t_h} = e^{\bar{\lambda}_j t_h} e^{\mathbf{M}_j t_h} \quad (4.112)$$

lässt sich die Exponentialfunktion in das Produkt zweier Exponentialfunktionen zerlegen. Die Imaginärteile werden in diesem Abschnitt nicht weiter berücksichtigt. Eine Eigenwertverschiebung, die sowohl Real- als auch Imaginärteile berücksichtigt, folgt im anschließenden Abschnitt. Die skalare Exponentialfunktion des Mittelwerts aller reellen Eigenwerte $\bar{\lambda}_j$ kann im Intervall $0 \leq t_h \leq 1$ über

$$e^{\bar{\lambda}_j t_h} \leq \frac{1 + e^{\bar{\lambda}_j}}{2} + \frac{|1 - e^{\bar{\lambda}_j}|}{2} \quad (4.113a)$$

$$e^{\bar{\lambda}_j t_h} \geq \frac{1 + e^{\bar{\lambda}_j}}{2} - \frac{|1 - e^{\bar{\lambda}_j}|}{2} \quad (4.113b)$$

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

abgeschätzt werden. Der Wertebereich der Matrixexponentialfunktion von \mathbf{M}_j lässt sich über eine Tschebyscheffentwicklung mit Resttermabschätzung (vgl. (4.89))

$$e^{\mathbf{M}_j t_h} \leq \frac{1}{2} \hat{\mathbf{E}}_{j,0} \mathbf{T}_0^*(t_h) + \hat{\mathbf{E}}_{j,1} \mathbf{T}_1^*(t_h) + \sum_{n=2}^m |\hat{\mathbf{E}}_{j,n}| \quad (4.114a)$$

$$e^{\mathbf{M}_j t_h} \geq \frac{1}{2} \hat{\mathbf{E}}_{j,0} \mathbf{T}_0^*(t_h) + \hat{\mathbf{E}}_{j,1} \mathbf{T}_1^*(t_h) - \sum_{n=2}^m |\hat{\mathbf{E}}_{j,n}| \quad (4.114b)$$

einschränken. Die weitere Vorgehensweise kann mit Hilfe der Arithmetik sogenannter Tschebyscheff-Modelle [161] beschrieben werden. Ein Tschebyscheff-Modell setzt sich aus einem Polynomanteil $P(t_h)$ und einem (konstanten) Intervallanteil Δ zusammen:

$$(P(t_h), \Delta) = P(t_h) + \Delta. \quad (4.115)$$

Die Abschätzung der skalaren Exponentialfunktion nach (4.113) kann demnach durch das Tschebyscheff-Modell

$$(P_1(t_h), \Delta_1) = \frac{1 + e^{\bar{\lambda}_j}}{2} + \left[-\frac{|1 - e^{\bar{\lambda}_j}|}{2}, \frac{|1 - e^{\bar{\lambda}_j}|}{2} \right] \quad (4.116)$$

repräsentiert werden. Ebenso lässt sich die Abschätzung des Matrixexponentials (4.114) als Tschebyscheff-Modell

$$(P_2(t_h), \Delta_2) = \frac{1}{2} \hat{\mathbf{E}}_{j,0} \mathbf{T}_0^*(t_h) + \hat{\mathbf{E}}_{j,1} \mathbf{T}_1^*(t_h) + \left[-\sum_{n=2}^m |\hat{\mathbf{E}}_{j,n}|, \sum_{n=2}^m |\hat{\mathbf{E}}_{j,n}| \right] \quad (4.117)$$

darstellen. Die Gesamtabschätzung nach (4.112) erfordert eine Multiplikation der beiden Exponentialfunktionen. Die Vorschrift zur Multiplikation zweier Tschebyscheff-Modelle lautet (in Anlehnung an [161])

$$(P_1(t_h), \Delta_1)(P_2(t_h), \Delta_2) = P_1(t_h)P_2(t_h) + B(P_2(t_h))\Delta_1 + B(P_1(t_h))\Delta_2 + \Delta_1\Delta_2 \quad (4.118)$$

und erfordert daher unter anderem die Abschätzung⁴⁰ des Wertebereichs der beiden Polynomanteile $B(P_1(t_h))$ und $B(P_2(t_h))$. Da der Polynomanteil der skalaren Exponentialfunktion lediglich aus einer Konstanten besteht, ist dessen Abschätzung trivial:

$$B(P_1(t_h)) = \left[\frac{1 + e^{\bar{\lambda}_j}}{2}, \frac{1 + e^{\bar{\lambda}_j}}{2} \right]. \quad (4.119)$$

⁴⁰ Die Abschätzung des Wertebereichs eines Polynoms $P(t_h)$ wird in Anlehnung an [161] mittels $B(P(t_h))$ ausgedrückt.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Die Abschätzung des Polynomanteils der Matrixexponentialfunktion ist dank der Entwicklung in Tschebyscheff-Polynomen ebenfalls leicht zu bestimmen:

$$B(P_2(t_h)) = \frac{1}{2}\hat{\mathbf{E}}_{j,0} + \left[-|\hat{\mathbf{E}}_{j,1}|, |\hat{\mathbf{E}}_{j,1}|\right]. \quad (4.120)$$

Einsetzen der Tschebyscheff-Modelle (4.116) und (4.117) sowie der Polynomabschätzungen (4.119) und (4.120) in (4.118) liefert unter Anwendung der Intervallarithmetik [162] schließlich die Abschätzung der Matrixexponentialfunktion des gesamten Blocks im Intervall $0 \leq t_h \leq 1$:

$$\begin{aligned} e^{\mathbf{B}_j t_h} &\leq \frac{1 + e^{\bar{\lambda}_j}}{2} \left(\frac{1}{2} \hat{\mathbf{E}}_{j,0} - \hat{\mathbf{E}}_{j,1} + 2\hat{\mathbf{E}}_{j,1} t_h \right) + \\ &\quad \frac{|1 - e^{\bar{\lambda}_j}|}{2} \left(\frac{1}{2} |\hat{\mathbf{E}}_{j,0}| + |\hat{\mathbf{E}}_{j,1}| \right) + \\ &\quad \left(\frac{1 + e^{\bar{\lambda}_j}}{2} + \frac{|1 - e^{\bar{\lambda}_j}|}{2} \right) \sum_{n=2}^m |\hat{\mathbf{E}}_{j,n}| \end{aligned} \quad (4.121a)$$

$$\begin{aligned} e^{\mathbf{B}_j t_h} &\geq \frac{1 + e^{\bar{\lambda}_j}}{2} \left(\frac{1}{2} \hat{\mathbf{E}}_{j,0} - \hat{\mathbf{E}}_{j,1} + 2\hat{\mathbf{E}}_{j,1} t_h \right) - \\ &\quad \frac{|1 - e^{\bar{\lambda}_j}|}{2} \left(\frac{1}{2} |\hat{\mathbf{E}}_{j,0}| + |\hat{\mathbf{E}}_{j,1}| \right) - \\ &\quad \left(\frac{1 + e^{\bar{\lambda}_j}}{2} + \frac{|1 - e^{\bar{\lambda}_j}|}{2} \right) \sum_{n=2}^m |\hat{\mathbf{E}}_{j,n}|. \end{aligned} \quad (4.121b)$$

Zusätzlich sind in (4.121) die beiden Tschebyscheff-Polynome durch ihre Definition aus (4.82) ersetzt. Mit (4.121) sind nun eine obere und untere Schranke der Matrixexponentialfunktion eines einzelnen Blocks gegeben. Ein Einsetzen in (4.111) liefert mit

$$\mathbf{V}_j = \frac{1 + e^{\bar{\lambda}_j}}{2} \mathbf{LC}_{\text{pwl}} \mathbf{S}_j \left(\frac{1}{2} \hat{\mathbf{E}}_{j,0} - \hat{\mathbf{E}}_{j,1} \right) \tilde{\mathbf{S}}_j \quad (4.122)$$

$$\mathbf{U}_j = \frac{1 + e^{\bar{\lambda}_j}}{2} \mathbf{LC}_{\text{pwl}} \mathbf{S}_j 2\hat{\mathbf{E}}_{j,1} \tilde{\mathbf{S}}_j \quad (4.123)$$

$$\begin{aligned} \mathbf{V}_{\text{abs},j} &= \frac{|1 - e^{\bar{\lambda}_j}|}{2} \left(\frac{1}{2} |\mathbf{LC}_{\text{pwl}} \mathbf{S}_j \hat{\mathbf{E}}_{j,0} \tilde{\mathbf{S}}_j| + |\mathbf{LC}_{\text{pwl}} \mathbf{S}_j \hat{\mathbf{E}}_{j,1} \tilde{\mathbf{S}}_j| \right) + \\ &\quad \left(\frac{1 + e^{\bar{\lambda}_j}}{2} + \frac{|1 - e^{\bar{\lambda}_j}|}{2} \right) \sum_{n=2}^m |\mathbf{LC}_{\text{pwl}} \mathbf{S}_j \hat{\mathbf{E}}_{j,n} \tilde{\mathbf{S}}_j| \end{aligned} \quad (4.124)$$

den Beitrag eines einzelnen Blocks zur Gesamtabschätzung. Die Summe der Beiträge aller Blöcke liefert schließlich die Gesamtmatrizen der oberen und unteren Schrankenfunktionen

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

(vgl. (4.92))

$$\mathbf{V} = \sum_{j=1}^n \mathbf{V}_j \quad (4.125)$$

$$\mathbf{U} = \sum_{j=1}^n \mathbf{U}_j \quad (4.126)$$

$$\mathbf{V}_{\text{abs}} = \sum_{j=1}^n \mathbf{V}_{\text{abs},j} \quad (4.127)$$

Multiplizieren dieser Matrizen mit \mathbf{x}_0 , $\mathbf{x}_0 t$ bzw. $|\mathbf{x}_0|$ führt schließlich auf die gesuchten Schrankenfunktionen (4.92).

Eigenwertverschiebung um einen reellen und komplexen Anteil

Die Eigenwertverschiebung mittels des Mittelwerts aller Eigenwerte aus dem vorhergehenden Abschnitt verschiebt prinzipbedingt nur die Realteile der Eigenwerte eines Blocks⁴¹. Mit Hilfe einer anderen, etwas aufwändigeren Ähnlichkeitstransformation ist es allerdings auch möglich, eine Verschiebung um einen reellen und komplexen Eigenwertanteil unter Beibehaltung rein reeller Arithmetik durchzuführen [137]. Das Ergebnis dieser Verschiebung besitzt die allgemeine Form

$$e^{\mathbf{B}_j t_h} = \mathbf{R}_j \mathbf{P}_j e^{\mathbf{C}_j t_h} e^{(\hat{\mathbf{B}}_j - \mathbf{C}_j) t_h} \mathbf{P}_j^{-1} \mathbf{R}_j^{-1}. \quad (4.128)$$

Mittels der Matrizen \mathbf{R}_j und \mathbf{P}_j kann ein Diagonalblock

$$\mathbf{B}_j = \begin{bmatrix} \mathbf{S}_{11} & \cdots & \mathbf{S}_{1n} \\ & \ddots & \vdots \\ & & \mathbf{S}_{nn} \end{bmatrix} \text{ mit } \mathbf{S}_{ii} = \begin{bmatrix} a_{ii} & c_{ii} \\ -d_{ii} & a_{ii} \end{bmatrix}, \quad c_{ii} > 0 \text{ und } d_{ii} > 0 \quad (4.129)$$

in die Gestalt

$$\hat{\mathbf{B}}_j = \begin{bmatrix} \hat{\mathbf{S}}_{11} & \cdots & \hat{\mathbf{S}}_{1n} \\ & \ddots & \vdots \\ & & \hat{\mathbf{S}}_{nn} \end{bmatrix} \text{ mit } \hat{\mathbf{S}}_{ij} = \begin{bmatrix} a_{ij} & b_{ij} \\ -b_{ij} & a_{ij} \end{bmatrix} \text{ und } b_{ij} > 0 \quad (4.130)$$

transformiert werden. Die Matrixexponentialfunktion der Verschiebungsmatrix⁴²

$$\mathbf{C}_j = \begin{bmatrix} \mathbf{C}_j^* & & \\ & \ddots & \\ & & \mathbf{C}_j^* \end{bmatrix} \text{ mit } \mathbf{C}_j^* = \begin{bmatrix} a_j & b_j \\ -b_j & a_j \end{bmatrix} \text{ und } b_j > 0 \quad (4.131)$$

⁴¹ Dies gilt nur unter der Annahme der rein reellen Schur-Zerlegung.

⁴² Die Elemente a_j bzw. b_j in (4.131) entsprechen dem Mittelwert aller a_{ij} bzw. b_{ij} aus (4.130).

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

kann durch

$$e^{\mathbf{C}_j t_h} = e^{a_j t_h} (\cos(b_j t_h) \mathbf{1} + \sin(b_j t_h) \mathbf{Q}_j) \quad (4.132)$$

mit

$$\mathbf{Q}_j = \begin{bmatrix} \mathbf{Q}_j^* & & \\ & \ddots & \\ & & \mathbf{Q}_j^* \end{bmatrix} \text{ und } \mathbf{Q}_j^* = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (4.133)$$

direkt angegeben werden. Die Lösung des Matrixexponentialis der verschobenen, transformierten Blockmatrix wird wiederum über eine Tschebyscheffentwicklung approximiert

$$e^{(\hat{\mathbf{B}}_j - \mathbf{C}_j) t_h} \approx \sum_{n=0}^m \hat{\mathbf{E}}_{j,n} \mathbf{T}_n^*(t_h). \quad (4.134)$$

Die Multiplikation beider Lösungen (4.132) und (4.134) liefert

$$e^{\mathbf{C}_j t_h} e^{(\hat{\mathbf{B}}_j - \mathbf{C}_j) t_h} = e^{a_j t_h} (\cos(b_j t_h) \mathbf{1} + \sin(b_j t_h) \mathbf{Q}_j) \sum_{n=0}^m \hat{\mathbf{E}}_{j,n} \mathbf{T}_n^*(t_h). \quad (4.135)$$

Unter Ausnutzung der Identität

$$a \cos(x) + b \sin(x) = \sqrt{a^2 + b^2} \cos(x - \text{atan}(b/a)) \quad (4.136)$$

lässt sich der Wertebereich von (4.135) durch⁴³

$$e^{\mathbf{C}_j t_h} e^{(\hat{\mathbf{B}}_j - \mathbf{C}_j) t_h} \leq \left(\frac{1 + e^{a_j}}{2} + \frac{|1 - e^{a_j}|}{2} \right) \sqrt[3]{ \left(\sum_{n=0}^m |\hat{\mathbf{E}}_{j,n}| \right)^{\circ 2} + \left(\mathbf{Q}_j \sum_{n=0}^m |\hat{\mathbf{E}}_{j,n}| \right)^{\circ 2} } \quad (4.137a)$$

$$e^{\mathbf{C}_j t_h} e^{(\hat{\mathbf{B}}_j - \mathbf{C}_j) t_h} \geq - \left(\frac{1 + e^{a_j}}{2} + \frac{|1 - e^{a_j}|}{2} \right) \sqrt[3]{ \left(\sum_{n=0}^m |\hat{\mathbf{E}}_{j,n}| \right)^{\circ 2} + \left(\mathbf{Q}_j \sum_{n=0}^m |\hat{\mathbf{E}}_{j,n}| \right)^{\circ 2} }. \quad (4.137b)$$

abschätzen. Die Berücksichtigung der Transformationsmatrizen nach (4.128), der Matrizen aus (4.111) und schließlich der Matrizen der PWL-Ungleichung aus (4.77) führt schließlich auf

$$\mathbf{V}_j = \mathbf{0} \quad (4.138)$$

$$\mathbf{U}_j = \mathbf{0} \quad (4.139)$$

$$\mathbf{V}_{\text{abs},j} = \sqrt[3]{ \left(\mathbf{H}_j \sum_{n=0}^m |\hat{\mathbf{E}}_{j,n}| \tilde{\mathbf{H}}_j \right)^{\circ 2} + \left(\mathbf{H}_j \mathbf{Q}_j \sum_{n=0}^m |\hat{\mathbf{E}}_{j,n}| \tilde{\mathbf{H}}_j \right)^{\circ 2} } \quad (4.140)$$

⁴³ Der Ausdruck $\sqrt[3]{\mathbf{A}}$ berechnet die elementweise Quadratwurzel der Matrix \mathbf{A} . Der Ausdruck $\mathbf{A}^{\circ 2}$ berechnet das elementweise Quadrat der Matrix \mathbf{A} . Da quadratische Matrizen sowohl eine Quadratwurzel als auch ein Quadrat besitzen, ist hier eine Unterscheidung notwendig.

4.4. Bestimmung der Zeitpunkte eines Netzwerkwechsels

mit

$$\mathbf{H}_j = \left(\frac{1 + e^{a_j}}{2} + \frac{|1 - e^{a_j}|}{2} \right) \mathbf{L} \mathbf{C}_{\text{pwl}} \mathbf{S}_j \mathbf{R}_j \mathbf{P}_j \quad \text{und} \quad \tilde{\mathbf{H}}_j = \mathbf{P}_j^{-1} \mathbf{R}_j^{-1} \tilde{\mathbf{S}}_j. \quad (4.141)$$

Zusammenfassend lässt sich festhalten, dass durch eine Eigenwertverschiebung die Konvergenz der Tschebyscheffentwicklung erheblich beschleunigt werden kann und sich somit der maximale Zeitschritt bei gleichem Polynomgrad entsprechend vergrößert. Die Limitierung ergibt sich nun nicht mehr durch die Beträge der Eigenwerte selbst, sondern hängt einzig von den Abständen der Eigenwerte innerhalb eines Diagonalblocks ab [137]. Je näher die Eigenwerte beieinander liegen, umso größer sind die möglichen Zeitschritte (bei gleichem Polynomgrad). Im Grenzfall, d. h. bei identischen Eigenwerten, gibt es keine Begrenzung des maximalen Zeitschritts.

Auswirkung der Eigenwertverschiebung anhand eines Beispiels

Die Auswirkungen der Eigenwertverschiebung mit sowohl reellem als auch imaginärem Anteil soll anhand des Zustandsraumsystems mit der Systemmatrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \quad (4.142)$$

mit den Blöcken

$$\mathbf{A}_1 = \begin{bmatrix} \lambda & \omega \\ -\omega & \lambda \end{bmatrix} \quad \text{mit} \quad \omega = 2\pi 10^3 \frac{\text{rad}}{\text{s}} \quad \text{und} \quad \lambda = -16 \text{ s}^{-1} \quad (4.143)$$

und

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (4.144)$$

gezeigt werden. Mit dem Vektor der Anfangswerte

$$\mathbf{x}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T \quad (4.145)$$

und der Ausgangsmatrix

$$\mathbf{C}_{\text{pwl}} = \begin{bmatrix} -0,5 & 0,5 & -2 & 3,5 \end{bmatrix} \quad (4.146)$$

und der PWL-Ungleichungsmatrix

$$\mathbf{L} = [1] \quad (4.147)$$

besitzt die Lösung dieses Systems den gleichen Zeitverlauf wie (4.106). Die Systemmatrix (4.142) besitzt bereits die nötige blockdiagonale Form, um ohne Ähnlichkeitstrans-

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

formation eine Eigenwertverschiebung durchführen zu können. Der erste der beiden Blöcke (4.143) liefert nach (4.138)-(4.140) die Matrizen

$$\mathbf{V}_1 = \mathbf{0} \quad (4.148)$$

$$\mathbf{U}_1 = \mathbf{0} \quad (4.149)$$

$$\mathbf{V}_{\text{abs},1} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}. \quad (4.150)$$

Der zweite Block (4.144) liefert nach (4.122)-(4.124) die Matrizen

$$\mathbf{V}_2 = \begin{bmatrix} 0 & 0 & -2 & 3,5 \end{bmatrix} \quad (4.151)$$

$$\mathbf{U}_2 = \begin{bmatrix} 0 & 0 & 0 & -2 \end{bmatrix} \quad (4.152)$$

$$\mathbf{V}_{\text{abs},2} = \mathbf{0}. \quad (4.153)$$

Summierung der beiden Teillösungen nach (4.125)-(4.127) und einsetzen in (4.92) liefert für das Intervall $0 \leq t \leq 1$ s die beiden Schrankenfunktionen

$$\bar{f}(t) = 1,5 - 2,0 \text{ s}^{-1} t + \sqrt{2} \quad (4.154)$$

$$\underline{f}(t) = 1,5 - 2,0 \text{ s}^{-1} t - \sqrt{2}. \quad (4.155)$$

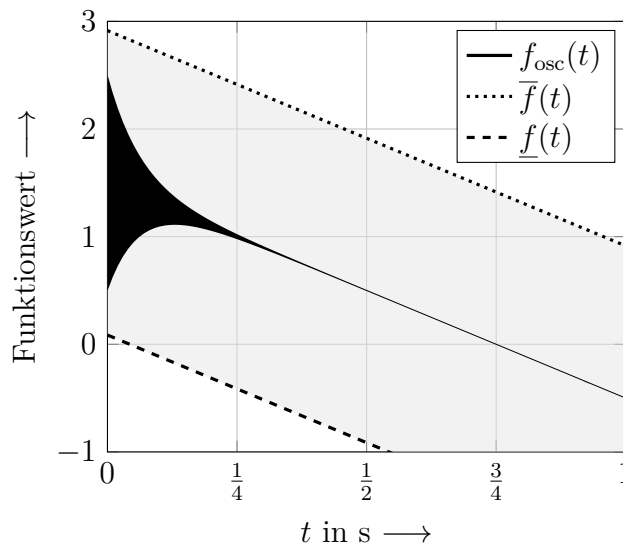


Abbildung 4.17.: Beispielfunktion (4.106) mit oberer und unterer Schranke (4.154) bzw. (4.155)

Eine grafische Auswertung der Lösung des Differentialgleichungssystems (4.106) als auch der beiden Schranken zeigt Abb. 4.17. Um das Intervall des Nulldurchgangs $1/2 \leq t \leq 1$ s zu erreichen, in dem annähernd eine quadratische Konvergenz erreicht wird, muss zunächst vier mal hintereinander die Bisektion angewendet werden. Anschließend liefert das

4.5. Verarbeitung und Ausgabe von Simulationsergebnissen

Verfahren nach der insgesamt 14. Iteration das gesuchte Intervall mit eingeschlossener Nullstelle. Auf weitere Durchgänge wird verzichtet, da die Intervallgrenzen bei Darstellung in „double“-Genauigkeit im darauffolgenden Schritt bereits aufeinander zu liegen kommen. Die Ergebnisse der einzelnen Iterationen sind in Tab. 4.2 zusammengefasst.

Tabelle 4.2.: Ergebnisse aller Iterationen der Nullstellensuche für die exemplarische Funktion (4.93)

Iteration	Intervall in s	Weite in s
1	$0 \leq t \leq 1$	1
2	$0 \leq t \leq \frac{1}{2}$	$\frac{1}{2}$
3	$0 \leq t \leq \frac{1}{4}$	$\frac{1}{4}$
4	$0 \leq t \leq \frac{1}{8}$	$\frac{1}{8}$
5	$0 \leq t \leq \frac{1}{16}$	$\frac{1}{16}$
6	$\frac{1}{32} \leq t \leq \frac{1}{16}$	$\frac{1}{32}$
7	$\frac{1}{16} \leq t \leq \frac{1}{8}$	$\frac{1}{16}$
8	$\frac{1}{8} \leq t \leq \frac{1}{4}$	$\frac{1}{8}$
9	$\frac{1}{4} \leq t \leq \frac{1}{2}$	$\frac{1}{4}$
10	$\frac{1}{2} \leq t \leq 1$	$\frac{1}{2}$
11	$\frac{767}{2^{10}} \leq t \leq \frac{769}{2^{10}}$	$\frac{1}{2^9}$
12	$\frac{49151}{2^{16}} \leq t \leq \frac{49153}{2^{16}}$	$\frac{1}{2^{15}}$
13	$\frac{50331647}{2^{26}} \leq t \leq \frac{50331649}{2^{26}}$	$\frac{1}{2^{25}}$
14	$\frac{52776558133247}{2^{46}} \leq t \leq \frac{52776558133249}{2^{46}}$	$\frac{1}{2^{45}}$

4.5. Verarbeitung und Ausgabe von Simulationsergebnissen

Es gibt mehrere Möglichkeiten das Ergebnis einer transienten Schaltungsanalyse in Zustandsraumdarstellung zu speichern und für eine nachfolgende Weiterverarbeitung zugänglich zu machen. Simulationsprogramme, die auf SPICE basieren, speichern üblicherweise die Ergebnisse sämtlicher Zeitschritte ab. Teilweise bieten sie dem Anwender die Möglichkeit, diese mitunter riesige Datenmenge durch verlustbehaftete Datenkompression zu reduzieren⁴⁴. Das im Rahmen dieser Arbeit entwickelte Programm verzichtet auf das Abspeichern sämtlicher Zwischenlösungen. Stattdessen werden die Anfangswerte \mathbf{x}_0 und die

⁴⁴ s. Hilfe zu LTspice IV, Version 4.21i, Abschnitt „Control Panel“, Unterabschnitt „Compression“

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

Zeitdauern eines jeden Zeitintervalls sowie die zum Intervall zugehörige Zustandsraumdarstellung abgespeichert. Um auch Strom- und Spannungsimpulse berücksichtigen zu können, werden darüber hinaus die Matrizen und Vektoren der Schalttransformationsgleichung für jeden impulsbehafteten Schaltvorgang abgespeichert⁴⁵.

Eine grafische Darstellung der Simulationsergebnisse erfordert typischerweise die nachgelagerte Berechnung von Zwischenwerten, denn die jeweiligen Anfangswerte eines jeden Zeitintervalls \mathbf{x}_0 geben keinen Aufschluss über den Zeitverlauf innerhalb des Intervalls.

Neben der grafischen Repräsentation stellen Kennwerte der Zeitverläufe wie Effektivwert, Mittelwert oder Spitzenwerte eine wichtige Information für den Anwender dar. Sie dienen zur Überprüfung der Beanspruchung bzw. der anfallenden Verluste von Schaltungselementen. Diese Kennwerte können, wie bei der grafischen Darstellung, über genügend fein gewählte Zeitschritte mit Hilfe numerischer Näherungsverfahren (z. B. der Trapezregel) bestimmt werden. Dies widerspricht jedoch dem in dieser Arbeit verwendeten Ansatz, welcher versucht, auf eine hochauflösende Abtastung der Kurvenverläufe zu verzichten. Durch die Einführung eines erweiterten Zustandsvektors ist es möglich auch ohne zusätzliche Abtastung den Mittelwert wie auch den Effektivwert zu bestimmen, wie die folgenden Unterabschnitte zeigen.

4.5.1. Bestimmung des Mittelwerts

Der Mittelwert der Zustandsgrößen $\bar{\mathbf{x}}$ im Zeitintervall von t_0 bis $t_0 + T$ ist

$$\bar{\mathbf{x}} = \frac{1}{T} \int_{t_0}^{t_0+T} \mathbf{x}(t) dt. \quad (4.156)$$

Das Einsetzen der Lösung aus (4.2) liefert

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{T} \int_{t_0}^{t_0+T} e^{\mathbf{A}t} \mathbf{x}(0) dt \\ &= \frac{1}{T} (e^{\mathbf{A}T} - \mathbf{1}) \mathbf{A}^{-1} \mathbf{x}(t_0). \end{aligned} \quad (4.157)$$

Eine analytische Mittelwertberechnung ohne hochauflösende Abtastung der Kurvenform ist mit Hilfe (4.157) zwar möglich, setzt allerdings eine invertierbare Systemmatrix voraus. Da sich eine singuläre bzw. nahezu singuläre Systemmatrix im Allgemeinen jedoch nicht vermeiden lässt, soll ein alternativer Lösungsweg aus [163, 164] aufgezeigt werden. Dieser bedient sich, wie auch schon die Eingangsgrößenmodellierung aus Unterabschnitt 2.2.2, zusätzlicher Hilfs-Zustandsgrößen. Eine mögliche Umsetzung dieses Vorgehens lässt sich mit Hilfe des Schaltbildes in Abb. 4.18 darstellen.

⁴⁵ Die Gewichte der Impulse sind in den Vektoren der Schalttransformationsgleichung enthalten.

4.5. Verarbeitung und Ausgabe von Simulationsergebnissen

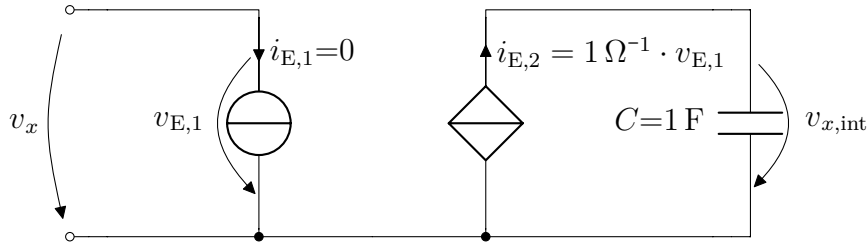


Abbildung 4.18.: Integrieren einer beliebigen Spannung v_x mittels spannungsgesteuerter Stromquelle und einer Hilfs-Zustandsgröße $v_{x,int}$

Die Kapazitätsspannung $v_{x,int}$ entspricht dem Integral der Spannung v_x . Mit Hilfe der entkoppelnden Eigenschaft der gesteuerten Quelle besitzt die zusätzliche Kapazität keine Rückwirkung auf das restliche Netzwerk. Das gleiche ist auch für beliebige Ströme im Netzwerk umsetzbar, allerdings muss dann der Eingangszweig einer stromgesteuerten Stromquelle in Reihe zum zu integrierenden Strom geschaltet werden.

Eine Formulierung direkt in Zustandsraumdarstellung statt dem Umweg über Hilfsschaltungen macht gesteuerte Quellen überflüssig und führt auf das gleiche Ergebnis. Eine Rückwirkungsfreiheit wird innerhalb der Systemmatrix durch nullwertige Elemente an der entsprechenden Stelle erreicht. Das Differentialgleichungssystem der Zustandsraumdarstellung wird dabei zur Integration der Zustandsgrößen um einen Vektor \mathbf{x}_{int} mit Hilfszustandsgrößen erweitert

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{int} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_{avg}} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{int} \end{bmatrix}. \quad (4.158)$$

Die Lösung des DGLs zum Zeitpunkt $t_0 + T$, gewichtet mit dem Kehrwert der Intervalldauer

$$\bar{\mathbf{x}} = \frac{1}{T} \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix} e^{\mathbf{A}_{avg} T} \begin{bmatrix} \mathbf{x}(t_0) \\ \mathbf{0} \end{bmatrix} \quad (4.159)$$

liefert schließlich den Mittelwert an Zustandsgrößen über dem Zeitintervall von t_0 bis $t_0 + T$. Dank der linearen Berechnungsvorschrift (4.156) können Mittelwerte der Ausgangsgrößen ebenso bestimmt werden:

$$\bar{\mathbf{y}} = \frac{1}{T} \begin{bmatrix} \mathbf{0} & \mathbf{C} \end{bmatrix} e^{\mathbf{A}_{avg} T} \begin{bmatrix} \mathbf{x}(t_0) \\ \mathbf{0} \end{bmatrix}. \quad (4.160)$$

Überspannt das zu betrachtende Zeitintervall von t_0 bis $t_0 + T$ mehrere Zeitintervalle mit unterschiedlichen Zustandsraumdarstellungen, so muss für jedes dieser Teilintervalle getrennt der Mittelwert berechnet werden und die Teillösungen zum Gesamtmittelwert mittels gewichteter Addition zusammengesetzt werden.

4.5.2. Bestimmung des Effektivwerts

Der Effektivwert aller Zustandsgrößen im Zeitintervall von t_0 bis $t_0 + T$ ist

$$\mathbf{x}_{\text{RMS}} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} [x_1^2(t) \quad x_2^2(t) \quad \cdots \quad x_n^2(t)]^T dt}. \quad (4.161)$$

Ein möglicher Lösungsweg zur Berechnung dieser Effektivwerte wird in [124], allerdings ohne Herleitung, vorgestellt. Dieser bedient sich, wie auch schon die Methode zur Mittelwertberechnung aus [164], neuer, künstlicher Zustandsgrößen. Als Ausgangspunkt der noch fehlenden Herleitung dient die Lösung einer linearen, homogenen Differentialgleichung höherer Ordnung, welche sich aus einer Linearkombination aus Basisfunktionen⁴⁶ der Form

$$t^n e^{\lambda t} \text{ mit } \lambda \in \mathbb{C} \quad (4.162)$$

zusammensetzt. Die Berechnung des Effektivwerts verlangt nun ein Quadrieren dieser Linearkombinationen aus Basisfunktionen. Das Ausmultiplizieren dieses Quadrats der Linearkombinationen liefert sowohl Produkte der Basisfunktionen untereinander

$$t^n e^{\lambda_1 t} \cdot t^m e^{\lambda_2 t} = t^{n+m} e^{(\lambda_1 + \lambda_2)t} \quad (4.163)$$

als auch die Quadrate der Basisfunktionen

$$(t^n e^{\lambda t})^2 = t^{2n} e^{2\lambda t}. \quad (4.164)$$

Anhand der Ergebnisse in (4.163) bzw. (4.164) lässt sich feststellen, dass auch diese wiederum die Form einer Basisfunktion (4.162) besitzen. Das bedeutet, dass das Quadrat der Lösung einer linearen, homogenen Differentialgleichung beliebiger Ordnung selbst wiederum durch eine lineare, homogene Differentialgleichung beschrieben werden kann. Dieser Umstand wird in der Methode nach [124] ausgenutzt, um die Effektivwerte direkt in Zustandsraumdarstellung zu berechnen.

Aus (4.161) ist bereits ersichtlich, dass der neue Zustandsvektor $\tilde{\mathbf{x}}$ in jedem Fall die Quadrate der ursprünglichen Zustände enthalten muss. Die Ableitung dieser neuen Zustände

$$\dot{x}_i^2 = 2x_i \dot{x}_i = 2a_{i1}x_i x_1 + 2a_{i2}x_i x_2 + \dots + 2a_{in}x_i x_n \quad (4.165)$$

zeigt jedoch, dass auch „Mischprodukte“ $x_i x_j$ aus zwei ursprünglichen Zustandsgrößen benötigt werden⁴⁷. Daher müssen sämtliche Produkt-Permutationen in den neuen Zustandsvektor aufgenommen werden. Um die Herleitung zu vereinfachen, wird zunächst

⁴⁶ Diese Basisfunktionen sind der Grund, weshalb die Methode der Eingangsgrößenmodellierung (s. Abschnitt 2.2.2) auf Quellen mit Exponential-, Sinus-/Kosinus- sowie Polynomfunktionen beschränkt ist.

⁴⁷ Die Vorfaktoren a_{i1} bis a_{in} aus (4.165) entsprechen den Elementen der i -ten Zeile der Systemmatrix \mathbf{A} .

4.5. Verarbeitung und Ausgabe von Simulationsergebnissen

angenommen, dass eine Permutation mit Wiederholung benötigt wird, d. h. es wird sowohl den neuen Zustand $x_i x_j$ als auch $x_j x_i$ geben, obwohl beide identisch sind. Für die Ableitungen der „Mischprodukte“

$$\begin{aligned} (\dot{x}_i x_j) &= \dot{x}_i x_j + x_i \dot{x}_j \\ &= a_{i1} x_j x_1 + a_{i2} x_j x_2 + \dots + a_{in} x_j x_n \\ &\quad + a_{j1} x_i x_1 + a_{j2} x_i x_2 + \dots + a_{jn} x_i x_n \end{aligned} \quad (4.166)$$

bedarf es keiner zusätzlichen Zustände. Der neue Zustandsvektor lautet folglich

$$\tilde{\mathbf{x}} = \begin{bmatrix} x_1 x_1 & x_1 x_2 & \cdots & x_1 x_n & x_2 x_1 & x_2 x_2 & \cdots & x_2 x_n & \cdots & x_n x_1 & x_n x_2 & \cdots & x_n x_n \end{bmatrix}^T. \quad (4.167)$$

Dieser kann mit Hilfe des Kronecker-Produkts \otimes

$$\tilde{\mathbf{x}} = \mathbf{x} \otimes \mathbf{x} \quad (4.168)$$

aus dem ursprünglichen Zustandsvektor gewonnen werden [165]. Die Lösung des neuen Zustandsvektors lässt sich mit Hilfe der Lösung aus (4.2) und unter Zuhilfenahme einiger Identitäten aus [165] als

$$\begin{aligned} \tilde{\mathbf{x}} &= e^{\mathbf{A}t} \mathbf{x}_0 \otimes e^{\mathbf{A}t} \mathbf{x}_0 \\ &= (e^{\mathbf{A}t} \otimes e^{\mathbf{A}t}) (\mathbf{x}_0 \otimes \mathbf{x}_0) \\ &= e^{(\mathbf{A} \oplus \mathbf{A})t} \tilde{\mathbf{x}}_0 \end{aligned} \quad (4.169)$$

darstellen. Daraus folgt direkt die Systemmatrix des neuen, erweiterten Systems, die sich mit Hilfe der ursprünglichen Systemmatrix und der Kronecker-Summe bzw. des Kronecker-Produkts zu

$$\tilde{\mathbf{A}} = \mathbf{A} \oplus \mathbf{A} = \mathbf{A} \otimes \mathbf{1} + \mathbf{1} \otimes \mathbf{A} \quad (4.170)$$

bestimmen lässt.

Zur Berechnung des Effektivwerts werden zusätzlich zum neuen Zustandsvektor noch die Hilfs-Zustände zur Bildung der Mittelwerte (s. Unterabschnitt 4.5.1) benötigt:

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}_{\text{aux}} \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_{\text{RMS}}} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}_{\text{aux}} \end{bmatrix}. \quad (4.171)$$

Mittels des neuen Ausgangsvektors

$$\tilde{\mathbf{y}} = \mathbf{y} \otimes \mathbf{y} \quad (4.172)$$

und dem Ansatz

$$\begin{aligned} \tilde{\mathbf{y}} &= \mathbf{C} e^{\mathbf{A}t} \mathbf{x}_0 \otimes \mathbf{C} e^{\mathbf{A}t} \mathbf{x}_0 \\ &= (\mathbf{C} \otimes \mathbf{C}) (e^{\mathbf{A}t} \otimes e^{\mathbf{A}t}) (\mathbf{x}_0 \otimes \mathbf{x}_0) \end{aligned} \quad (4.173)$$

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

lässt sich in gleicher Weise wie in (4.169) eine neue Ausgangsmatrix

$$\tilde{\mathbf{C}}_{\text{full}} = \mathbf{C} \otimes \mathbf{C} \quad (4.174)$$

bestimmen. Allerdings beinhaltet der neue Ausgangsvektor neben den Quadraten der ursprünglichen Ausgangsgrößen auch deren Mischprodukte. Um unnötigen Rechenaufwand zu vermeiden, kann die neue Ausgangsmatrix statt mit dem Kronecker-Produkt auch mit Hilfe des Khatri-Rao-Produkts $*$ gewonnen werden

$$\tilde{\mathbf{C}} = (\mathbf{C}^{\top} * \mathbf{C}^{\top})^{\top} \quad (4.175)$$

wodurch die Mischprodukte entfallen [166]. Die Effektivwerte der Ausgangsgrößen über das Zeitintervall von t_0 bis $t_0 + T$ ergeben sich schließlich zu:

$$\mathbf{y}_{\text{RMS}} = \sqrt{\frac{1}{T} [\mathbf{0} \quad \tilde{\mathbf{C}}] e^{\mathbf{A}_{\text{RMS}} T} \begin{bmatrix} \tilde{\mathbf{x}}(t_0) \\ \mathbf{0} \end{bmatrix}}. \quad (4.176)$$

Der Zustandsvektor in (4.168) besitzt einige redundante Zustände, da alle Mischprodukte doppelt auftreten (Permutation mit Wiederholung). Eine von diesen überflüssigen Zuständen bereinigte Zustandsraumdarstellung zur Berechnung des Effektivwerts ist Bestandteil des Abschnitts B.1 in Anhang B.

Überspannt das zu betrachtende Zeitintervall von t_0 bis $t_0 + T$ mehrere Zeitintervalle mit unterschiedlichen Zustandsraumdarstellungen, so muss für jedes dieser Teilintervalle getrennt das Quadrat des Effektivwerts berechnet werden und die Teillösungen zum Quadrat des Gesamteffektivwerts mittels gewichteter Addition zusammengesetzt werden. Eine anschließende Berechnung der Quadratwurzel liefert schließlich den gesuchten Gesamteffektivwert.

4.6. Zusammenfassung des Ablaufs

Die rechnergestützte, transiente Schaltungsanalyse, wie sie in diesem und dem vorhergehenden Kapitel präsentiert wird, setzt als Ausgangspunkt einzig die Schaltungsbeschreibung in Form einer SPICE-Netzliste voraus. Simulationsdetails, wie etwa der Betriebsmodus einer getakteten, leistungselektronischen Schaltung oder die Umschaltbedingungen der als stückweise linear definierten Schaltungselemente (PWL-Elemente), werden selbstständig erkannt und müssen nicht vom Anwender vorgegeben werden.

Dem Flussdiagramm aus Abb. 4.19 kann der stark abstrahierte Programmablauf entnommen werden. ① Die Anfangswerte der Spannungen und Ströme aller Kapazitäten bzw. Induktivitäten werden, wenn in der Netzliste nicht anders vorgegeben, zu Beginn der Simulation zu null angenommen. Der Anfangszustand jedes PWL-Elements, d. h.

das anfängliche Segment seiner stückweise linearen Kennlinie, ist beliebig, da im weiteren Verlauf ein konsistenter Zustand (falls vorhanden) automatisch hergestellt wird. ② Die Aufstellung des Gesamtgleichungssystems nach (3.35) geschieht unter Anwendung grundlegender Zusammenhänge der Netzwerktheorie. Der dafür nötige rechentechnische Aufwand ist aufgrund der vergleichsweise geringen Anzahl an Elementen leistungselektronischer Schaltungen gering. ③ Ob ein Netzwerk verboten ist oder nicht zeigt sich anschließend an der letzten Zeile der in reduzierter Zeilenstufenform vorliegenden Matrix des Gesamtgleichungssystems⁴⁸. ④ Gegebenenfalls muss eine übergeordnete Programmschleife die Zustandskombination aller als stückweise linear definierten Schaltungselemente (PWL-Kombination) iterativ anpassen, sodass schließlich ein konsistentes Netzwerk entsteht. ⑤ Sollte dabei keine Kombination gefunden werden, die ein konsistentes Netzwerk ermöglicht, wird die Simulation mit einer Fehlermeldung abgebrochen. ⑥ Anschließend können konsistente Anfangswerte gefunden werden sowie die Zustandsraumdarstellung formuliert werden. ⑦ Das Netzwerk ist zum jetzigen Zeitpunkt zwar nicht verboten, jedoch konnte bisher nicht ermittelt werden, ob die Zustände sämtlicher PWL-Elemente konsistent sind, d. h. ob die Ungleichung (4.73) verletzt oder erfüllt wird⁴⁹. Dies kann nun nachgeholt werden und die PWL-Kombination muss ggf. wiederum in einer übergeordneten Programmschleife angepasst werden. ⑧ Befinden sich alle PWL-Elemente in einem gültigen Zustand sollte die Zustandsraumdarstellung sowie die zugehörigen Anfangswerte für einen späteren Wiedergebrauch und der nachgelagerten Auswertung durch den Anwender abgespeichert werden. Ein lineares, zeitinvariantes System wäre damit vollständig beschrieben. ⑨ Allerdings verfügen leistungselektronische Schaltungen typischerweise über nichtlineare und zeitvariante Schaltungselemente. Wechselt beispielsweise eine Diode ihren Zustand von leitfähig in sperrend oder springt eine pulsformige Spannungsquelle auf ein anderes Spannungsniveau, so muss zum einen dieser Zeitpunkt erkannt werden, ⑩ abgespeichert und anschließend mit einer geänderten PWL-Kombination wiederum innerhalb einer Programmschleife begonnen werden. ⑪ Fällt dieser Zeitpunkt mit dem Simulationsende zusammen oder überschreitet er dieses, wird die Analyse beendet.

⁴⁸ Zwei einfache Beispiele verbotener Netzwerke sind eine leerlaufende Stromquelle und eine kurzgeschlossene Spannungsquelle.

⁴⁹ An dieser Stelle sei nochmals darauf hingewiesen, dass sowohl ideale Dioden als auch als stückweise linear definierte nichtlineare Kapazitäten und Induktivitäten alle mittels der Ungleichung (4.73) auf konsistente Zustände überprüft werden. Eine Unterscheidung zwischen „schwach“ nichtlinearen Schaltungselementen (nichtlineare Kapazitäten/Induktivitäten/Widerstände) mit nur leichten unterschiedlichen Parametern zwischen benachbarten PWL-Zuständen und „stark“ nichtlinearen (ideale Dioden) mit großen Änderungen zwischen benachbarter PWL-Zustände, findet im Rahmen dieser Arbeit nicht statt.

4. Rechnergestützte Simulation mit Hilfe der Zustandsraumdarstellung

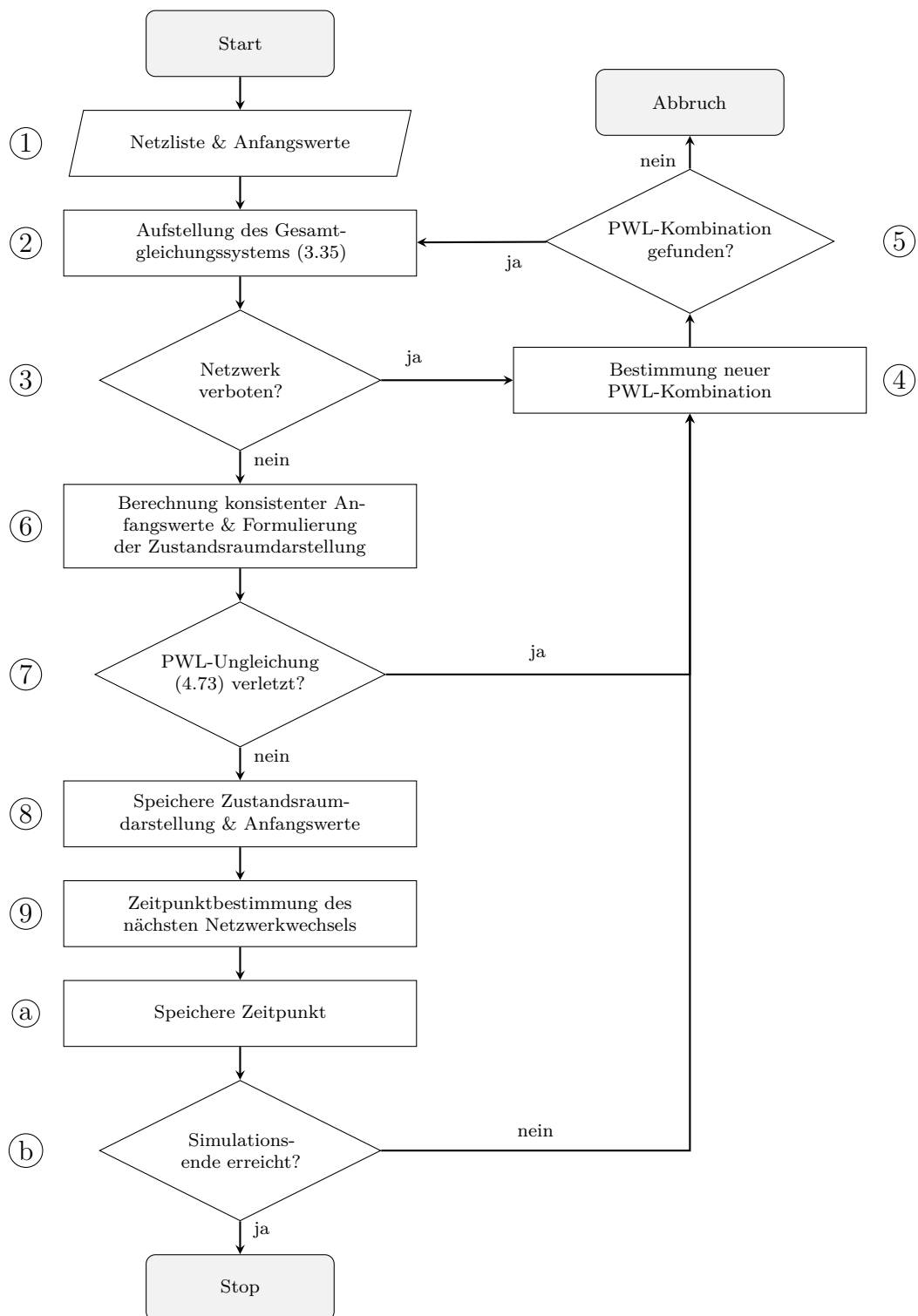


Abbildung 4.19.: Stark abstrahierter Ablauf der rechnergestützten, transienten Schaltungsanalyse

5. Verifizierung der Algorithmen und Lösungsstrategien

Die in den Kapiteln 3 und 4 beschriebenen Algorithmen und Lösungsstrategien wurden im Rahmen dieser Arbeit in Form eines Matlab¹-Programms zur rechnergestützten Auswertung umgesetzt. Die Eingabe der Schaltung und der erforderlichen Simulationsparameter erfolgt als SPICE-Netzliste. Die Ausgabe des Programms besteht in seiner Grundform als Abfolge von Anfangszustände aller Kapazitätsspannungen und Induktivitätsströmen sowie den Zeitdauern der aufeinanderfolgenden Zeitintervalle. Mit Hilfe der ebenfalls abgespeicherten Zustandsraumdarstellung, d. h. der System- und Ausgangsmatrizen, sowie der Zeitpunkte aller Unstetigkeiten kann jeder beliebige Spannungs- und Stromverlauf im Netzwerk in einem nachgelagerten Schritt mit beliebiger zeitlicher Auflösung rekonstruiert werden. Im Folgenden werden die Simulationsergebnisse einiger exemplarischer Schaltungen mit den Ergebnissen analytischer Rechnungen bzw. den Ergebnissen aus SPICE verglichen. Sämtliche SPICE-Simulationen dieses Kapitels wurden mit der Software LTspice IV² (Version 4.21i) durchgeführt.

5.1. Demonstrations-Schaltungen

Es folgen zunächst einige spezielle Demonstrations-Schaltungen, die dazu dienen sollen bestimmte Problemfälle, wie etwa inkonsistente Anfangswerte, gezielt zu provozieren.

5.1.1. Inkonsistente Anfangswerte und relativer Fehler

Die Schaltung aus Abb. 5.1a dient zum einen der Erzeugung inkonsistenter Anfangswerte und zum anderen weist sie den charakteristischen exponentiellen Strom- bzw. Spannungsverlaufs eines RC-Glieds auf, wodurch die Ergebnisse direkt mit einer analytischen Lösung verglichen werden können. Das Simulationsprogramm muss also zum einen konsistente Anfangswerte und zum anderen den über eine Differentialgleichung ge-

¹ Matlab ist eine kommerzielle Software des US-amerikanischen Unternehmens MathWorks zur Lösung mathematischer Probleme und zur grafischen Darstellung der Ergebnisse.

² Wenn nichts anderes angegeben ist, so wird das Programm stets mit den Simulationsparametern (Toleranzen, Integrationsverfahren,...) im Auslieferungszustand betrieben.

5. Verifizierung der Algorithmen und Lösungsstrategien

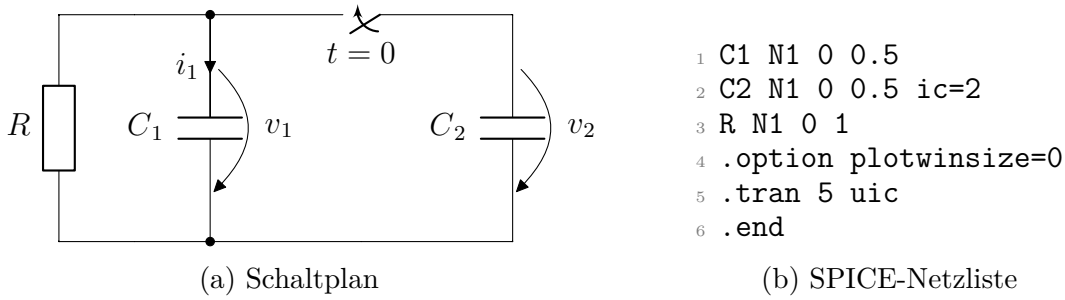


Abbildung 5.1.: Schaltung eines RC-Glieds (links) mit inkonsistenten Anfangswerten $v_1(0^-) = 0$ und $v_2(0^-) = 2\text{ V}$ sowie zugehöriger Netzliste (rechts)

gebenen Strom- bzw. Spannungsverlauf bestimmen. Die Schaltungselemente besitzen die in der Netzliste aus Abb. 5.1b aufgeführten Werte: $R = 1\ \Omega$ und $C_1 = C_2 = 0,5\text{ F}$. Der Schalter (welcher nur im Schaltplan jedoch nicht in der Netzliste vorkommt) wird zum Zeitpunkt $t = 0$ geschlossen, wodurch die beiden Kapazitäten mit den unterschiedlichen Anfangswerten $v_1(0^-) = 0$ und $v_2(0^-) = 2\text{ V}$ augenblicklich auf den neuen, konsistenten Anfangswert $v_1(0^+) = v_2(0^+) = 1\text{ V}$ umgeladen werden. Nach dem impulsbehafteten Umladevorgang nimmt die Kapazitätsspannung mit

$$v_1(t) = v_2(t) = 1\text{ V} e^{\lambda t} \quad \text{mit} \quad \lambda = -(R(C_1 + C_2))^{-1} = -1\text{ s}^{-1} \quad (5.1)$$

den charakteristischen, exponentiell abnehmenden Verlauf an.

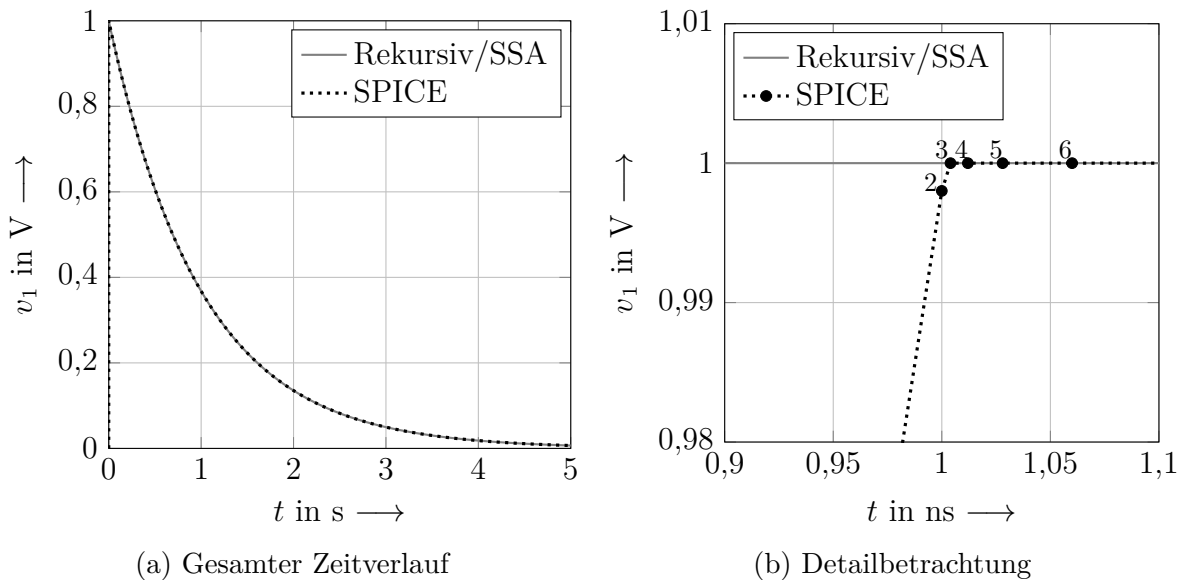


Abbildung 5.2.: Vergleich der mit SPICE und den mittels (5.2) rekursiv berechneten Spannungszeitverläufen der Simulation im Zustandsraum (SSA)

Die zugehörigen simulierten Zeitverläufe können Abb. 5.2a entnommen werden. Der Beginn der Simulation wird in Abb. 5.2b nochmals vergrößert (mit anderer Zeitskala) dargestellt. Hieraus wird der Unterschied in der Bestimmung des konsistenten Anfangswert ersichtlich. SPICE findet unter Zuhilfenahme einiger weniger, sehr kurzer Zeitschritte und dem impliziten Euler-Verfahren den korrekten Anfangswert. Die Simulation im Zustandsraum hingegen (in Abb. 5.2 mit „rekursiv/SSA“ bezeichnet) ermittelt den korrekten Anfangswert in einem der transienten Simulation vorgelagerten Schritt (mit dem Lösungsansatz aus Abschnitt 4.2).

Wie eingangs erwähnt, wird der zeitlich hoch aufgelöste Zeitverlauf während der Simulation in Zustandsraumdarstellung nicht ermittelt. Er wird mithilfe der Ausgangsmatrix, der Systemmatrix und der Anfangswerte in einem nachgelagerten Schritt rekonstruiert. Die zeitliche Auflösung ist dann beliebig wählbar. Eine Möglichkeit der Rekonstruktion mit äquidistanter Abtastung besteht in der rekursiven Berechnung

$$v_1((n+1)h) = e^{\lambda h} v_1(nh) \quad \text{mit} \quad v_1(0) = v_1(0^+). \quad (5.2)$$

Da im Beispiel nach Abb. 5.1a lediglich eine einzige Zustandsgröße vorhanden ist, ergibt sich im vorliegenden Beispiel statt einer Transitionsmatrix der skalare Wert $e^{\lambda h}$. Die damit berechnete Lösung wird in Abb. 5.2a als „Rekursiv“ bezeichnet. Eine Alternative besteht in der Auswertung mit Hilfe von Transitionsmatrizen unterschiedlicher Intervallweiten, deren Dauer sich nach der Folge von Zweierpotenzen richtet, ähnlich wie in Abschnitt 4.4.4 beschrieben. Sie wird im folgenden mit „Binär“ bezeichnet³.

Obwohl die Zeitverläufe aus Abb. 5.2a scheinbar aufeinanderliegen besitzen sie eine relative Genauigkeit, die sich um mehrere Zehnerpotenzen unterscheidet. Der relative Fehler ist definiert als

$$\text{relativer Fehler} := \left| \frac{\text{exakte Lösung} - \text{Lösung des numerischen Verfahrens}}{\text{exakte Lösung}} \right|. \quad (5.3)$$

Die dafür benötigte exakte Lösung lässt sich mit Hilfe der „Symbolic Math Toolbox“ in Matlab ermitteln⁴. Während die in SPICE verwendete numerische Integration stets einem Kompromiss aus Schrittweite und Genauigkeit unterliegt, wird der relative Fehler der mit „rekursiv“ und „binär“ bezeichneten Lösungen hauptsächlich durch Rundungsfehler der Gleitkommaarithmetik hervorgerufen. Zum Vergleich zeigt Abb. 5.3 mit der Bezeichnung „Minimum“ den relativen Fehler der auf „double“-Genauigkeit gerundeten exakten Lösung. Diese stellt damit den minimal erzielbaren relativen Fehler dar.

³ Die Lösung dieses Ansatzes („binär“) ist allerdings aus Gründen der Übersichtlichkeit nicht in Abb. 5.2a eingezeichnet.

⁴ Die „Symbolic Math Toolbox“ in Matlab erlaubt mittels der `vpa()`-Funktion die Verwendung einer virtuellen Gleitkommaarithmetik mit beliebig hoher Genauigkeit.

5. Verifizierung der Algorithmen und Lösungsstrategien

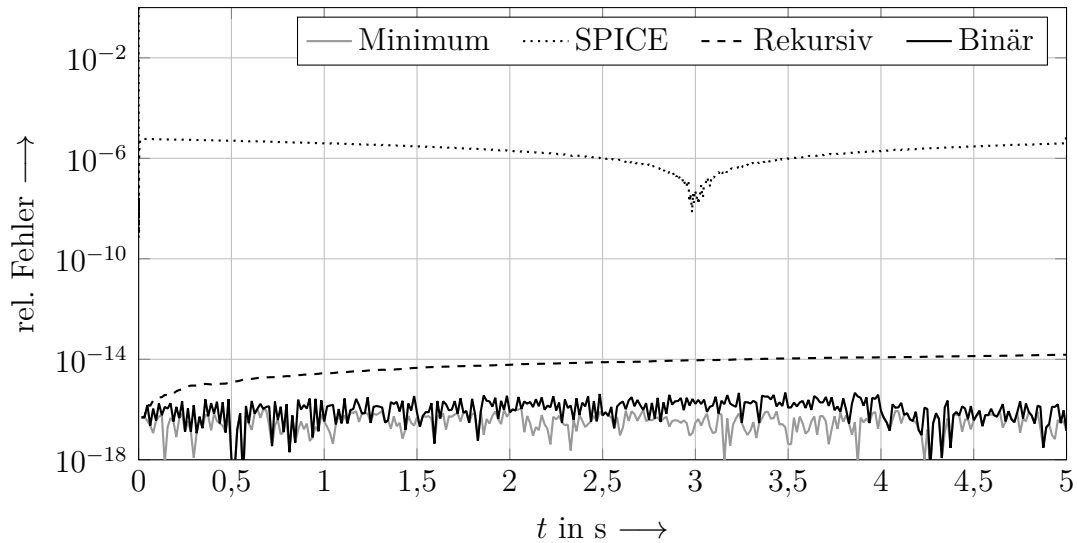


Abbildung 5.3.: Vergleich der relativen Fehler der SPICE-Simulation, der mit „Rekursiv“ bezeichneten Berechnung nach (5.2), der mit „Binär“ bezeichneten Berechnung nach Abschnitt 4.4.4 und dem minimal erzielbaren Fehler

5.1.2. Schaltungen mit versteckten Bedingungen

Die Bestimmung konsistenter Anfangswerte kann sich noch deutlich schwieriger als im zuvor gezeigten Abschnitt gestalten. Den Grund dafür bilden die sogenannten versteckten Bedingungen (vgl. Abschnitt 2.2.4). Das Beispiel einer Schaltung mit versteckten Bedingungen zeigt Abb. 5.5. Die kaskadierten spannungsgesteuerten Stromquellen sorgen zusammen mit den Induktivitäten für eine zunehmende zeitliche Differentiation der ursprünglichen Spannung über dem Schwingkreis links, bestehend aus L_1 und C_1 . Der korrekte Anfangswert jeder Induktivität ist daher die Spannung über der am Eingang der jeweiligen gesteuerten Quelle anliegenden Induktivität multipliziert mit dem Leitwert $\frac{1}{\Omega}$. Diese Spannung stellt wiederum, aufgrund der differentiellen Strom-Spannungsbeziehung an Induktivitäten, die Ableitung des Quellenstroms der gesteuerten Quelle dar. Der durch den Schwingkreis angeregte sinusförmige Spannungsverlauf⁵ setzt sich demnach um eine jeweils 90° voreilende Schwingung entlang der Kaskade fort. Die Oszillationsfrequenz, die Induktivitätswerte und die Verstärkungen der gesteuerten Quellen sind so gewählt, dass die Amplitude aller Ableitungen konstant bleibt (s. Werte der Netzliste aus Abb. 5.4).

Ein Vergleich der Simulationsergebnisse zur Netzliste aus Abb. 5.4 zeigt Abb. 5.6. Die Ergebnisse der SPICE-Simulation besitzen einen tiefgestellten Index „spice“, diejenigen der Simulation im Zustandsraum einen tiefgestellten Index „ss“. Der grafische Vergleich zeigt,

⁵ Statt des LC-Schwingkreises könnte auch eine sinusförmige Spannungsquelle eingesetzt werden. Allerdings ermöglicht der Schwingkreis, aufgrund der unterschiedlichen Quellenmodellierungen, eine bessere Vergleichbarkeit zwischen den Ergebnissen aus SPICE und der Simulation im Zustandsraum.

```

1 L1 N1 0 1 ic=-1
2 C1 N1 0 1
3 G1 0 N2 N1 0 1
4 L2 N2 0 1
5 G2 0 N3 N2 0 1
6 L3 N3 0 1
7 .tran 0 {4*pi} uic
8 .end

```

Abbildung 5.4.: SPICE-Netzliste zur Schaltung in Abb. 5.5

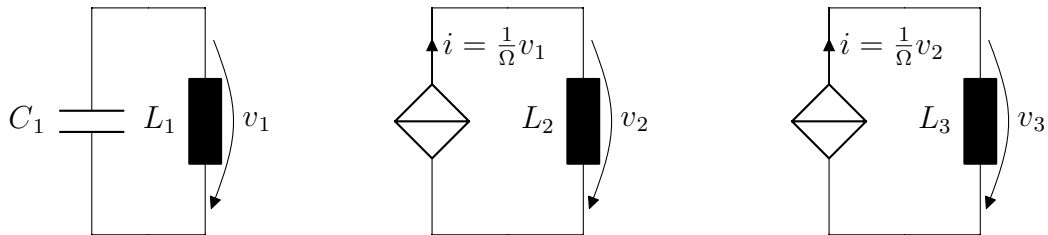


Abbildung 5.5.: Schaltung (Parameter nach Abb. 5.4) mit versteckten Bedingungen (nach [111])

dass die Ergebnisse übereinstimmen. Auf eine Auswertung und Vergleich des relativen Fehlers, wie in Abb. 5.3 geschehen, wird an dieser Stelle verzichtet. Zudem fehlen in Abb. 5.6 die ersten vier Zeitpunkte der SPICE-Zeitverläufe, da diese aufgrund der inkonsistenten Anfangswerte extrem hohe Werte aufweisen. Tabelle 5.1 zeigt daher die vernachlässigten Ergebnisse der ersten drei Zeitschritte (d. h. der ersten vier Zeitpunkte) der Simulation in SPICE. Trotz der versteckten Bedingungen liefert das implizite Euler-Verfahren nach dem zweiten Zeitschritt alle Anfangswerte als korrekt.

Tabelle 5.1.: Gerundete Ergebnisse der Simulation in SPICE

Zeit in s	v_1 in V	v_2 in V	v_3 in V	i_1 in A	i_2 in A	i_3 in A
0	0	0	0	0	0	0
10^{-9}	10^{-9}	0,998	$5 \cdot 10^8$	-0,998	10^{-9}	0,998
$1,004 \cdot 10^{-9}$	$1,004 \cdot 10^{-9}$	1	$3,333 \cdot 10^8$	-1	$1,004 \cdot 10^{-9}$	1
$1,012 \cdot 10^{-9}$	$1,012 \cdot 10^{-9}$	1	0	-1	$1,012 \cdot 10^{-9}$	1

Versteckte Bedingungen stellen jedoch nicht nur ein Problem bei der Bestimmung konsistenter Anfangswerte dar, sondern können auch während der transienten Simulation dafür sorgen, dass die Ergebnisse unbrauchbar werden. Dazu genügt bereits, lediglich eine weitere Stufe an die Kaskade aus Abb. 5.5 anzuhängen (s. Abb. 5.7).

5. Verifizierung der Algorithmen und Lösungsstrategien

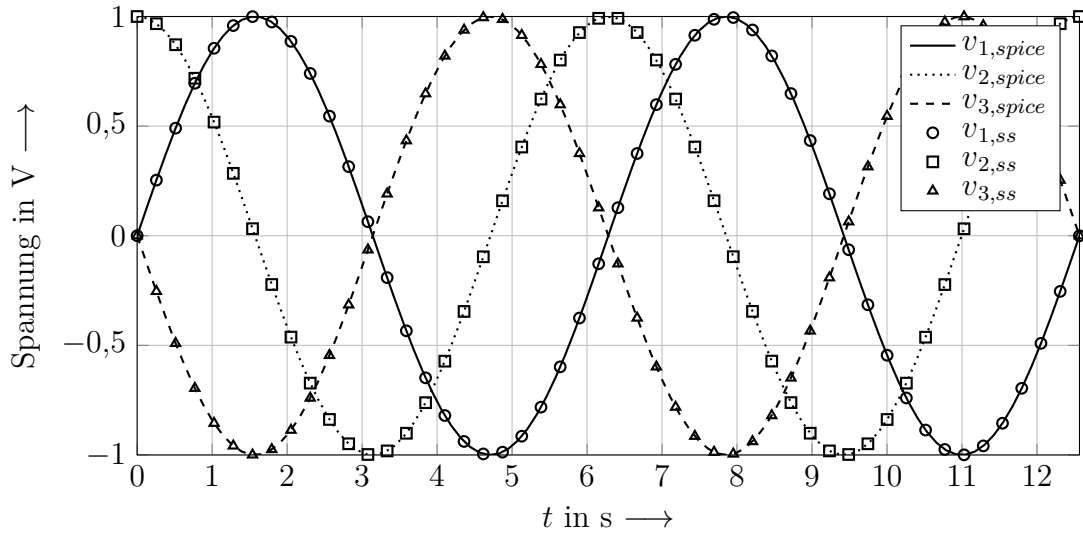


Abbildung 5.6.: Simulationsergebnisse zur Netzliste aus Abb. 5.4

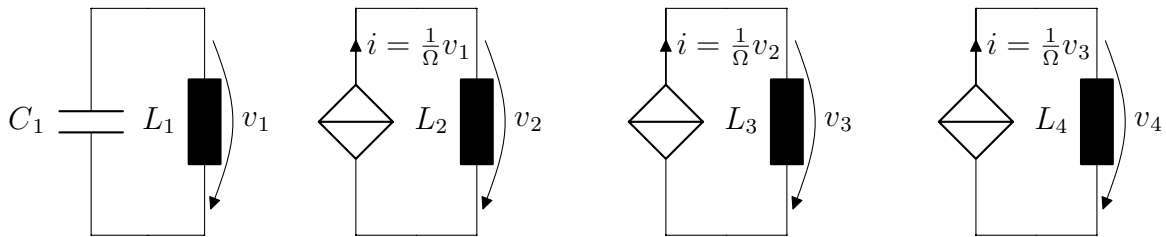


Abbildung 5.7.: Erweiterte Schaltung mit versteckten Bedingungen nach [111], sämtliche Induktivitäten besitzen 1 H, die Kapazität besitzt 1 F, die Anfangswerte sind identisch mit der Netzliste aus Abb. 5.4

Die Ergebnisse der zugehörigen transienten Simulation zeigt Abb. 5.8. Während die Simulation im Zustandsraum die korrekten Zeitverläufe liefert, kann die Spannung über der neu hinzugekommenen Induktivität (in der Schaltung nach Abb. 5.7 ganz rechts) in SPICE nicht mehr korrekt bestimmt werden. Dabei ist zu beachten, dass die Problematik hier nicht aufgrund der Differentiation von Eingangssignalen (Quellspannungen bzw. -ströme) hervorgerufen wird. Die Simulation im Zustandsraum ist davon prinzipbedingt nicht betroffen. Sobald konsistente Anfangswerte sowie die Zustandsraumdarstellung mit den Methoden aus Kapitel 4 bestimmt wurden, verläuft die Simulation ohne weitere Probleme. Tatsächlich stellen, unabhängig von der Anzahl der kaskadierten Glieder, ausschließlich die Spannung über der Kapazität C_1 und der Strom durch die Induktivität L_1 des Schwingkreises Zustandsgrößen dar. Sämtliche andere Netzwerkgrößen können aus diesen beiden Zeitverläufen gewonnen werden. Bei Simulation im Zustandsraum ist der Anzahl an Gliedern theoretisch keine Grenze gesetzt.

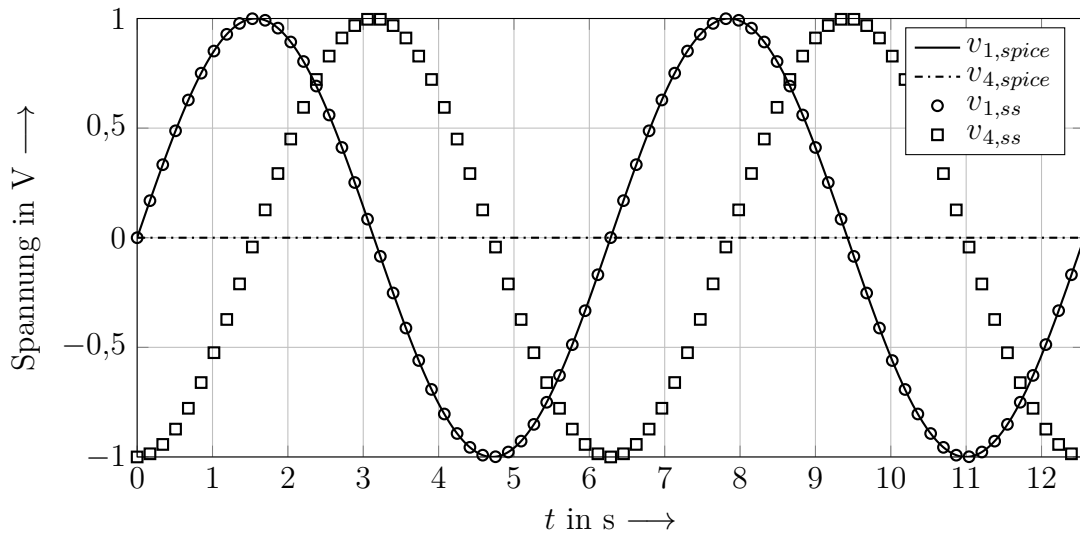


Abbildung 5.8.: Simulationsergebnisse zur Schaltung aus Abb. 5.7

5.1.3. Impulsbehaftete Schaltvorgänge und PWL-Elemente

Im Folgenden soll die Lösungsstrategie aus Abschnitt 4.2 verifiziert werden, die konsistente Anfangswerte nicht nur für lineare, sondern auch für als stückweise linear (PWL) definierte, nichtlineare Kapazitäten und Induktivitäten bestimmt. Dazu genügt eine ähnliche Schaltung wie diejenige aus Abb. 5.1a, allerdings nun mit zwei PWL-Kapazitäten (s. Abb. 5.9).

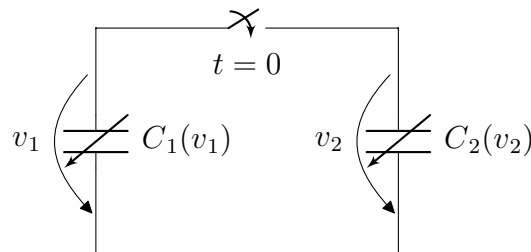


Abbildung 5.9.: Schaltung mit zwei nichtlinearen Kapazitäten und inkonsistenten Anfangswerten $v_{1-} \neq v_{2-}$

Die $q-v$ -Verläufe als auch die zugehörigen $c-v$ -Verläufe der beiden Kapazitäten sind in Abb. 5.10 dargestellt. Die (inkonsistenten) Anfangswerte der Kapazitätsspannungen betragen $v_{1-} = 0 \text{ V}$ bzw. $v_{2-} = 10 \text{ V}$. Eine analytische Rechnung auf Grundlage der Ladungserhaltung liefert mit den gegebenen Verläufen und Anfangswerten einen konsistenten Anfangswert von

$$v_{1+} = v_{2+} = 6,02 \text{ V.} \quad (5.4)$$

Bevor jedoch eine rechnergestützte Auswertung durchgeführt werden kann, muss das

5. Verifizierung der Algorithmen und Lösungsstrategien

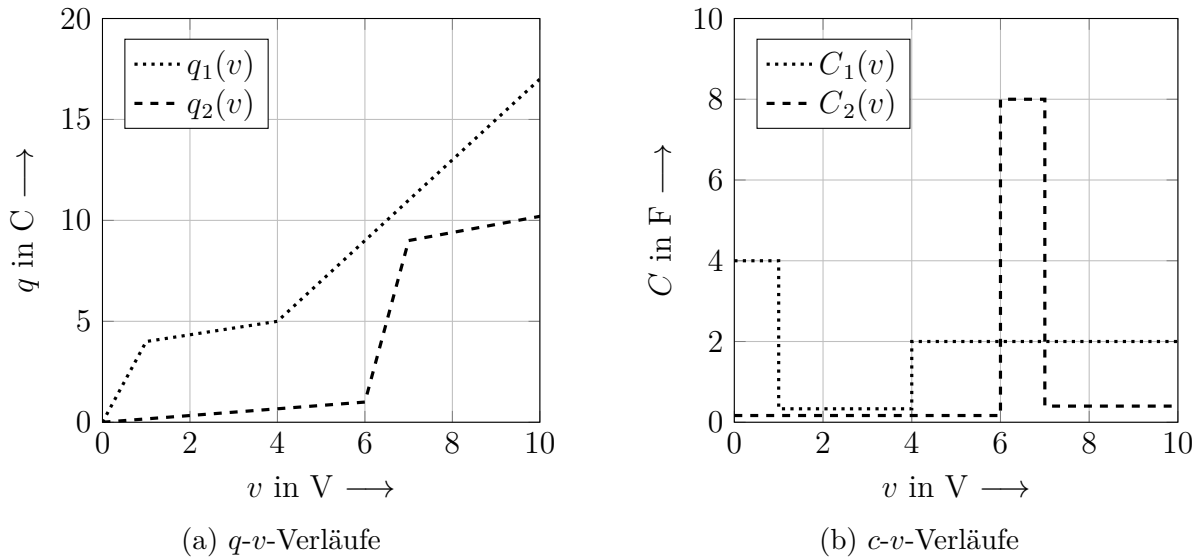


Abbildung 5.10.: Beispiele für zwei stückweise linearisierte, nichtlineare Kapazitäten

Schaltungsmodell in SPICE noch angepasst werden. SPICE ignoriert den `.ic`-Befehl für nichtlineare Kapazitäten, sodass hier eine alternative Vorgehensweise nötig wird. Die Simulation wird mit einem `.tran`-Befehl ohne das Attribut `uic` ausgeführt wodurch SPICE eine vorgelagerte DC-Analyse durchführt, um die beiden Anfangswerte der Kapazitäten zu bestimmen. Zur Vorgabe der gewünschten Anfangswerte sind daher zwei DC-Spannungsquellen mit in Reihe geschalteten, großen Induktivitäten, notwendig. Beide werden jeweils parallel zu den beiden Kapazitäten verschaltet (s. Abb. 5.11a). Die großen Induktivitäten werden während der DC-Analyse durch einen Kurzschluss ersetzt. Für die transiente Analyse stellen sie jedoch eine sehr große Impedanz dar. Die Simulation im Zustandsraum kommt ohne eine solche Hilfsbeschaltung aus.

Nachdem zum Zeitpunkt $t = 0$ der Schalter geschlossen wird, beginnt in SPICE der Ladungsausgleich (s. Abb. 5.12). Die Simulation im Zustandsraum ist nicht auf die Berechnung eines transienten Ausgleichsvorgangs angewiesen und findet mit der Lösungsstrategie aus Abschnitt 4.2 augenblicklich den konsistenten Anfangswert. Beide Simulationen liefern schließlich das erwartete Ergebnis von 6,02 V.

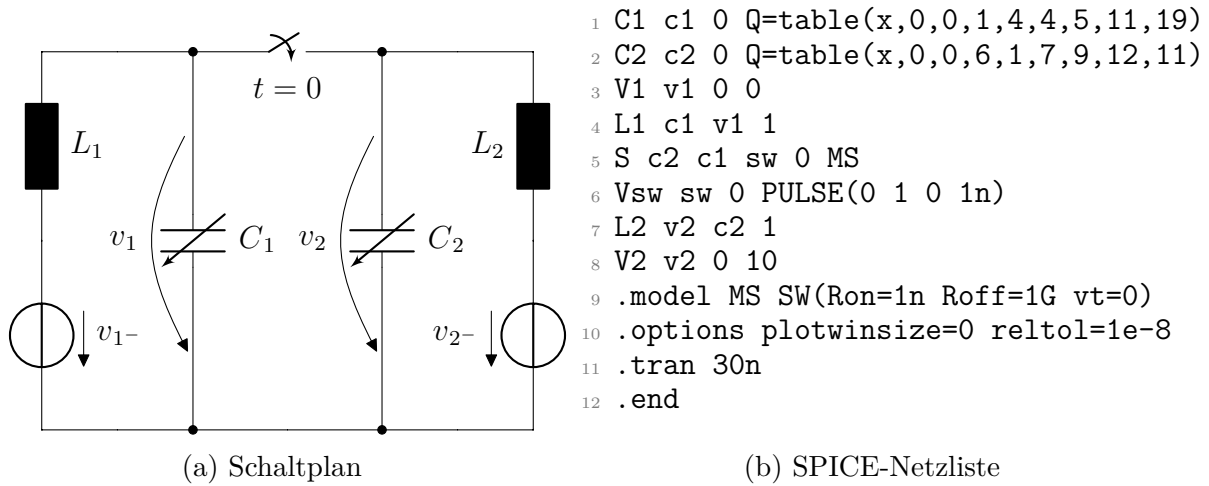


Abbildung 5.11.: Schaltplan und Netzliste mit Hilfsbeschaltung zur Realisierung benutzerdefinierter Anfangswerte

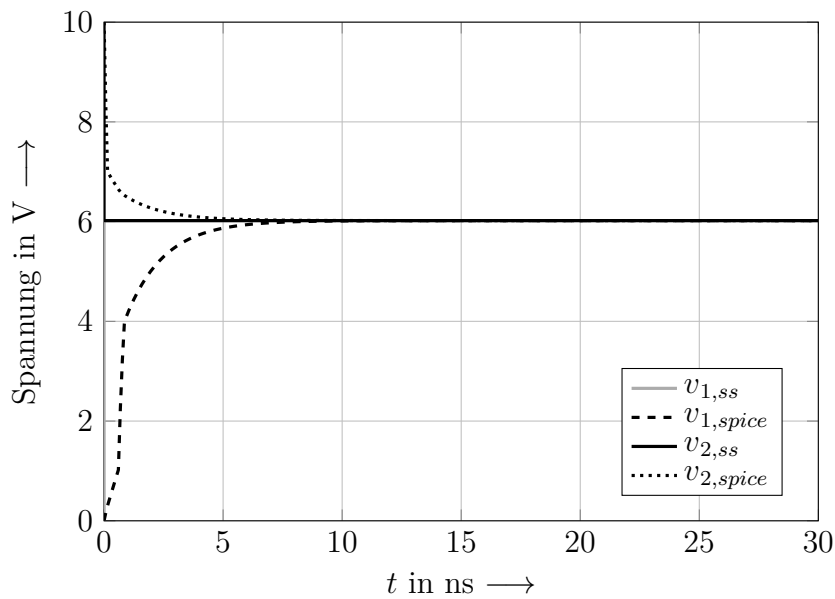


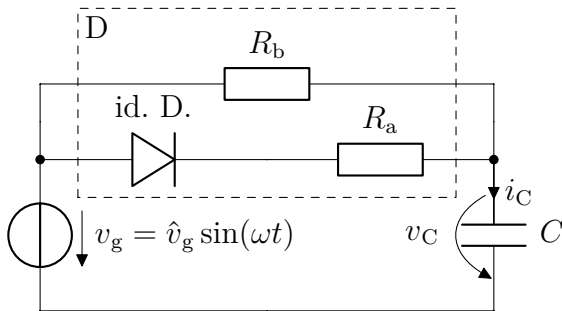
Abbildung 5.12.: Simulationsergebnisse zur Netzliste aus Abb. 5.11b, die Simulation im Zustandsraum besitzt den Index „ss“, die Simulation in SPICE den Index „spice“

5.1.4. Zuverlässige Detektion von Unstetigkeiten

Ein Schaltungsbeispiel zur Demonstration von schwer detektierbaren Unstetigkeiten ist bereits aus Abschnitt 2.2.6 bekannt. Es handelt sich dabei um den in Abb. 5.13a gezeigten Spitzenwertgleichrichter an einer Sinusspannungsquelle (die Werte ergeben sich aus der

5. Verifizierung der Algorithmen und Lösungsstrategien

zugehörigen Netzliste in Abb. 5.13b).



(a) Schaltplan

```

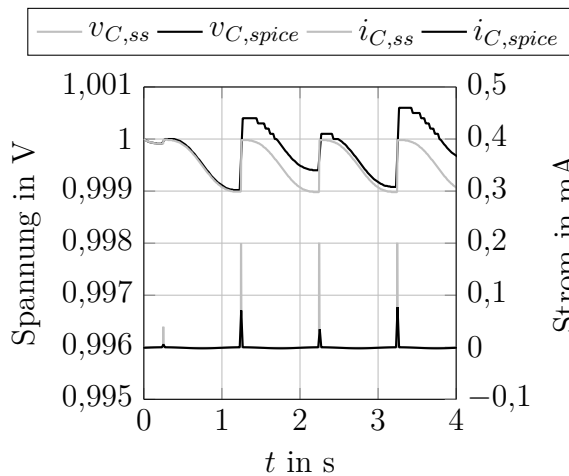
1 C N002 0 1m ic=1
2 Vg N001 0 SINE(0 1 1)
3 D N001 N002 MD
4 .model MD D(Ron=1 Roff=1Meg)
5 .options plotwinsize=0
6 .tran 10 uic
7 .end

```

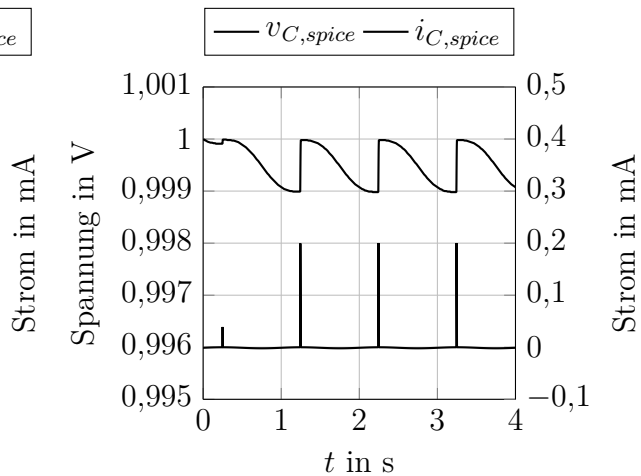
(b) SPICE-Netzliste

Abbildung 5.13.: Spitzenwertdetektor an Spannungsquelle („id. D.“ steht für ideale Diode, $R_{\text{off}} = R_b$, $R_{\text{on}} = \frac{R_a R_b}{R_a + R_b}$)

Da SPICE den sinusförmigen Spannungsverlauf der Quelle nicht fein genug abtastet, werden nicht alle Leitintervalle der Diode korrekt detektiert. Die Simulation in Zustandsraum hingegen verwendet den in Abschnitt 4.4.5 vorgestellten Algorithmus, der die Unstetigkeiten des Spitzenwertgleichrichters zuverlässig detektiert. Die Simulationsergebnisse zeigt Abb. 5.14a.



(a) $\text{reltol}=1\text{e-}3$



(b) $\text{reltol}=1\text{e-}8$

Abbildung 5.14.: Simulationsergebnisse der Netzliste aus Abb. 5.13b für SPICE und der Simulation im Zustandsraum mit zuverlässiger Detektion der Unstetigkeiten

Anhand des Kapazitätsstroms als auch der -spannung lassen sich deutliche Unterschiede erkennen. Während die Ergebnisse aus SPICE ein chaotisches Verhalten zeigen, besitzt die mittels der Simulation im Zustandsraum gefundene Lösung den erwarteten Zeitverlauf.

Um SPICE zu kleineren Zeitschritten zu zwingen, kann der Parameter `reitol`, der im Auslieferungszustand einen Wert von 0,001 besitzt, hin zu kleineren Werten geändert werden [7]. Eine zweite Simulation mit einem Parameter von `reitol=1e-8` liefert schließlich auch in SPICE das erwartete Ergebnis (s. Abb. 5.14b).

5.1.5. Unterbestimmtes Netzwerk

Unterbestimmte Netzwerke stellen eine Schwierigkeit dar, die lediglich diejenigen Simulationsprogramme beherrschen müssen, welche ideale Schaltelemente (Schaltern, Dioden, etc.) zulassen. Sie treten immer dann auf, wenn zwei oder mehr dieser idealen Schaltelemente im leitenden Zustand eine Masche bilden bzw. im sperrenden Zustand eine Trennmenge. Der ideale Vollbrückengleichrichter aus Abb. 5.15a kann als Beispiel eines solchen temporär unterbestimmten Netzwerks dienen [84]. Die Kapazität wird periodisch mit der doppelten Frequenz der Spannungsquelle über den Brückengleichrichter nachgeladen. Im Zeitintervall, in dem der Betrag der Spannung über der Quelle unterhalb der Kondensatorspannung liegt, sperren alle vier Dioden des Brückengleichrichters. Die Netzwerkzweige der Dioden bilden dann zusammen eine Trennmenge aus vier nullwertigen Stromquellen. Einzelne Zweigspannungen können in einem solchen Fall nicht mehr eindeutig bestimmt werden, ausschließlich die Summe zweier in Reihe liegender, sperrender Dioden (D_1 und D_2 bzw. D_3 und D_4) ist durch die Spannung über der Kapazität festgelegt.

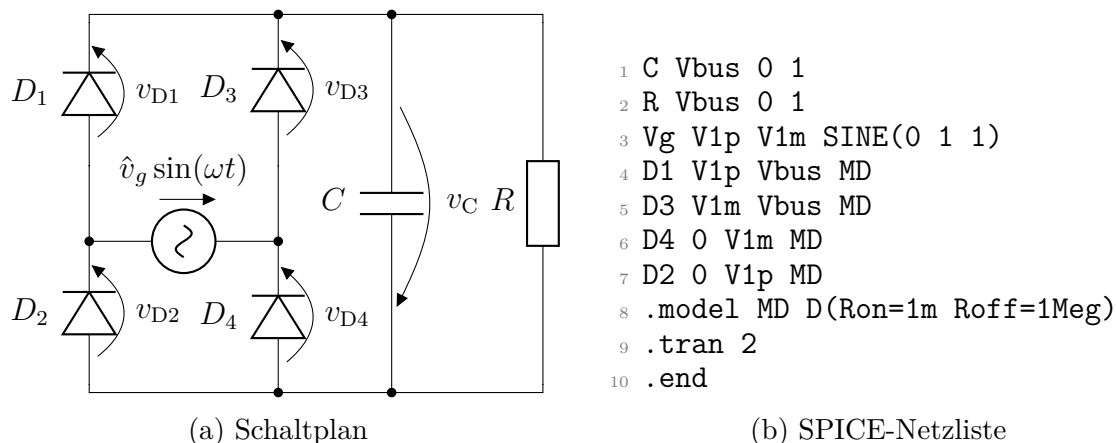


Abbildung 5.15.: Beispiel eines temporär unterbestimmten Netzwerks: Ein idealer Brückengleichrichter ($\hat{v}_g = 1 \text{ V}$, $\omega = 2\pi \text{ Hz}$, $C = 1 \text{ F}$, $R = 1 \Omega$)

Für SPICE stellt die Sperrphase kein Problem dar, denn ideale Dioden sind hier nicht zulässig. Dies zwingt den Anwender, einen endlichen Widerstandswert für die Leit- und Sperrphase anzugeben (vgl. Diodenmodell in Abb. 5.15b, Zeile 8). Dank der Methode der virtuellen Widerstände sind die Spannungen allerdings auch mit idealen Schaltern bestimmbar [73]. Das Ergebnis beider Simulationen zeigt Abb. 5.16. Der Anwender muss

5. Verifizierung der Algorithmen und Lösungsstrategien

sich allerdings im Klaren darüber sein, dass sich im realen Fall die Sperrspannungen entsprechend der Impedanzen der verwendeten Dioden aufteilen. Unterschiedliche Impedanzen führen folglich auf eine ungleichmäßige Spannungsbelastung aller Dioden. Der Unterschied zwischen der Simulation im Zustandsraum und in SPICE ist, dass das Diodenmodell im Zustandsraum durch Einfügen geeigneter parasitärer Elemente zwar auf den realen Fall angepasst werden kann, umgekehrt kann das Modell in SPICE jedoch nicht auf den idealen Fall vereinfacht werden.

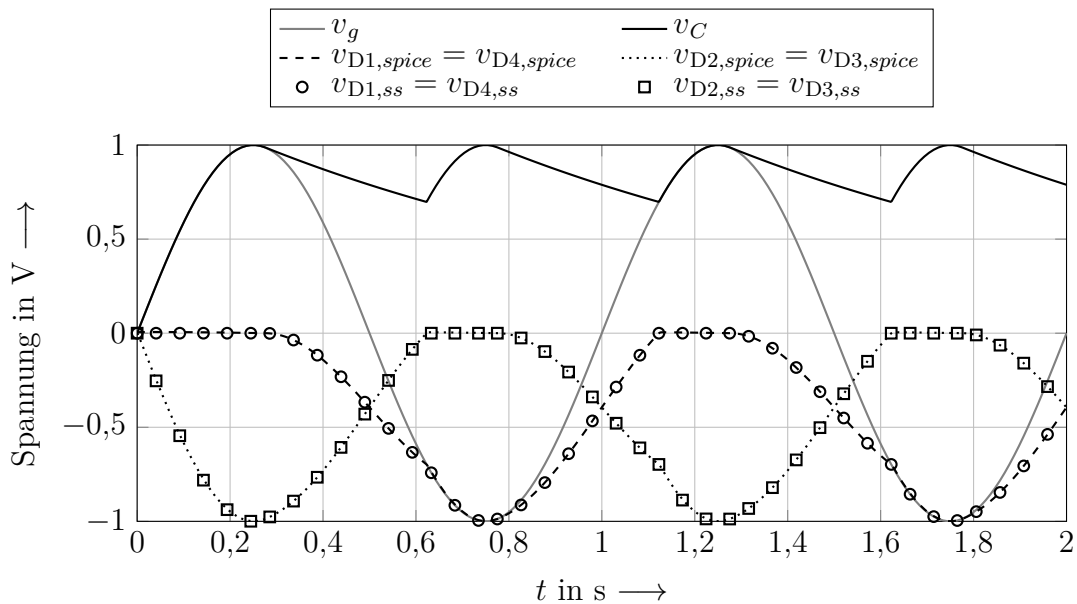


Abbildung 5.16.: Simulationsergebnisse zur Netzliste aus Abb. 5.15b, die Simulation in Zustandsraum (Index „ss“) verwendet ideale Dioden ($D(R_{on}=0, R_{off}=\infty)$), die Simulation in SPICE (Index „spice“) das Diodenmodell aus Abb. 5.15b, Zeile 8

5.2. Getaktete, leistungselektronische Schaltungen

Die in diesem Abschnitt simulierten Schaltungen stammen aus dem Bereich der höherfrequent getakteten Leistungselektronik [9]. Im Vergleich zu den nicht-getakteten (bspw. Linearregler) beziehungsweise zu den netzsynchron getakteten leistungselektronischen Schaltungen (bspw. Phasenanschnittsteuerung) ergeben sich durch die freie Wahl der Taktfrequenz erhebliche Vorteile. So können durch eine optimale Wahl der Schaltfrequenz deutlich kleinere, leichtere, kostengünstigere und effizientere passive Schaltungskomponenten

ten verwendet werden⁶.

Die in den folgenden Abschnitten behandelten Beispiele stellen nur eine kleine Auswahl an exemplarischen Schaltungen dar. Sie dienen dazu, die Simulationsergebnisse des in dieser Arbeit entwickelten Programms mittels Vergleichs-Simulationen in SPICE zu verifizieren.

5.2.1. Spannungsverdopplung mit Ladungspumpe

Ladungspumpen werden in der (leistungselektronischen) Schaltungstechnik häufig für Hilfsspannungs- oder Versorgungsspannungsaufgaben mit geringer Leistungsanforderung verwendet [167]. Häufig vorkommende Beispiele sind die Bootstrapping-Schaltung zur Versorgung des oben liegenden Halbleiterschalters einer Halbbrückenschaltung [167] oder der Spannungsverdoppler nach Abb. 5.17, wie er etwa auch im MAX232 [168], einem sehr weit verbreiteten integrierten Schaltkreis, Anwendung findet.

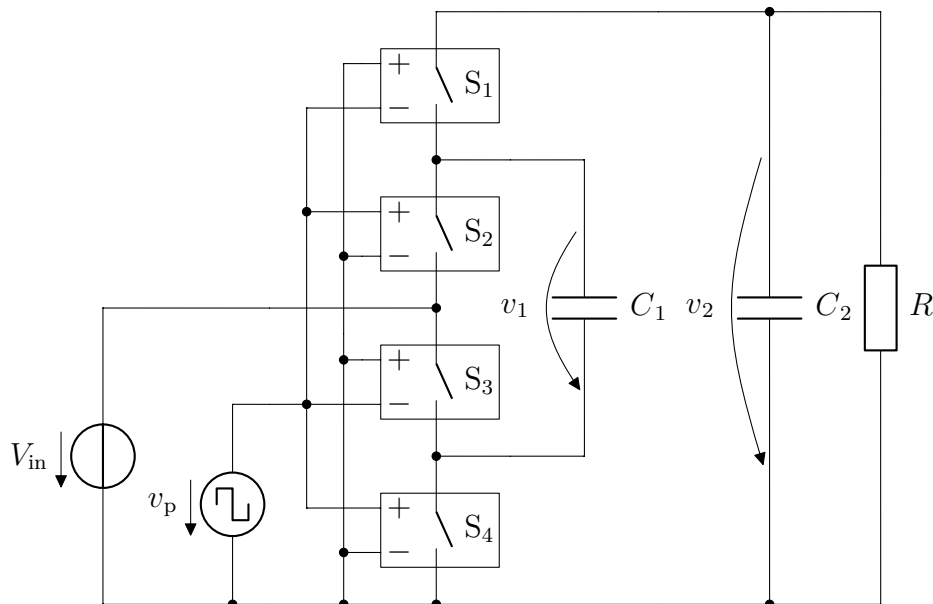


Abbildung 5.17.: Schaltung eines Spannungsverdopplers mit Ladungspumpe

Eine Schaltungsanalyse anhand des Schaltplans und der Netzliste ergibt, dass die Kapazität C_1 durch synchrones Takten der Schalter S_1 und S_3 bzw. S_2 und S_4 abwechselnd in Reihe bzw. parallel zur Eingangsspannungsquelle geschaltet wird. Im unbelasteten Zustand, d. h. ohne den Lastwiderstand R , wird die Ausgangskapazität C_2 durch die periodische Taktung bis auf das doppelte der Eingangsspannung aufgeladen. Unter Belastung bricht die Spannung entsprechend ein (vgl. Abb. 5.17). Die Ladung bzw. Entladung der

⁶ Ein eindrückliches Beispiel sind Ladegeräte für Mobiltelefone. Moderne, auf höherfrequent getakteter Leistungselektronik basierende Ladegeräte können mittlerweile in den Stecker integriert werden. Dies war mit den ursprünglichen Geräten, die mit einem Netztransformator arbeiteten, nicht möglich.

5. Verifizierung der Algorithmen und Lösungsstrategien

```
1 Vin in 0 5
2 Vp vp 0 PULSE(-1 1 10u 1n 1n 10u 20u)
3 C1 c1p c1m 1u
4 C2 out 0 1u
5 R out 0 1k
6 S1 out c1p 0 vp MS
7 S2 c1p in vp 0 MS
8 S3 in c1m 0 vp MS
9 S4 c1m 0 vp 0 MS
10 .model MS SW(Ron=1m Roff=1Meg Vt=0)
11 .tran 200u uic
12 .options plotwinsize=0
13 .end
```

Abbildung 5.18.: SPICE-Netzliste zur Schaltung aus Abb. 5.17

Kapazitäten geschieht über die sehr kleinen Leitwiderstände der Schalter (*hier*: 1 m Ω). Diese parasitären Widerstände sind einzig dazu nötig, impulsbehaftete Umladevorgänge zu vermeiden und dadurch eine Simulation in SPICE zu ermöglichen. Die Simulation im Zustandsraum, wie sie in dieser Arbeit vorgestellt wird, benötigt diese zusätzlichen Widerstände nicht. Die Umladung der Kapazitäten geschieht in diesem Fall impulsbehaftet und daher unendlich schnell.

Die Simulationsergebnisse aus Abb. 5.19 zeigen, dass die Zeitverläufe der beiden Simulationsmethoden auf den ersten Blick keine Unterschiede aufweisen. Erst eine genaue Betrachtung eines exemplarischen Schaltvorgangs aus Abb. 5.20 zeigt die Auswirkungen der parasitären Widerstände. Die Marker (Kreise) des Zeitverlaufs $v_{2,spice}$ kennzeichnen dabei die Zeitpunkte, für die SPICE die Zeitverläufe abspeichert⁷ (ohne Datenkompression). Während die Simulation in Zustandsraum mit den Methoden aus Kapitel 4 lediglich die Start- und Endwerte vor und nach jedem Schaltvorgang berechnet, ist SPICE darauf angewiesen, den eigentlich unstetigen Zeitverlauf durch parasitäre Schaltungselemente⁸ zu verstetigen.

Die Simulation der Ladungspumpe soll an dieser Stelle zwei Sachverhalte verdeutlichen:

- Impulsbehaftete Umladevorgänge werden von dem in dieser Arbeit entwickelten Simulationsprogramm korrekt simuliert und
- parasitäre Leit- und Sperrwiderstände sind zur Simulation der grundlegenden Funktionsweise einer getakteten, leistungselektronischen Schaltung typischerweise unnötig.

⁷ Der tatsächliche Berechnungsaufwand liegt jedoch weitaus höher, da SPICE für jeden Zeitschritt ein implizites Gleichungssystem iterativ löst und da aufgrund des PWL-Schaltermodells sowie der Pulsquelle einige der bereits berechneten Zeitpunkte wieder verworfen werden.

⁸ Nicht nur parasitäre Schaltungselemente sondern auch die endlichen Anstiegs- und Abfallzeiten von Pulsspannungsquellen sorgen für einen stetigen Verlauf aller Zeitverläufe der SPICE-Simulation.

5.2. Getaktete, leistungselektronische Schaltungen

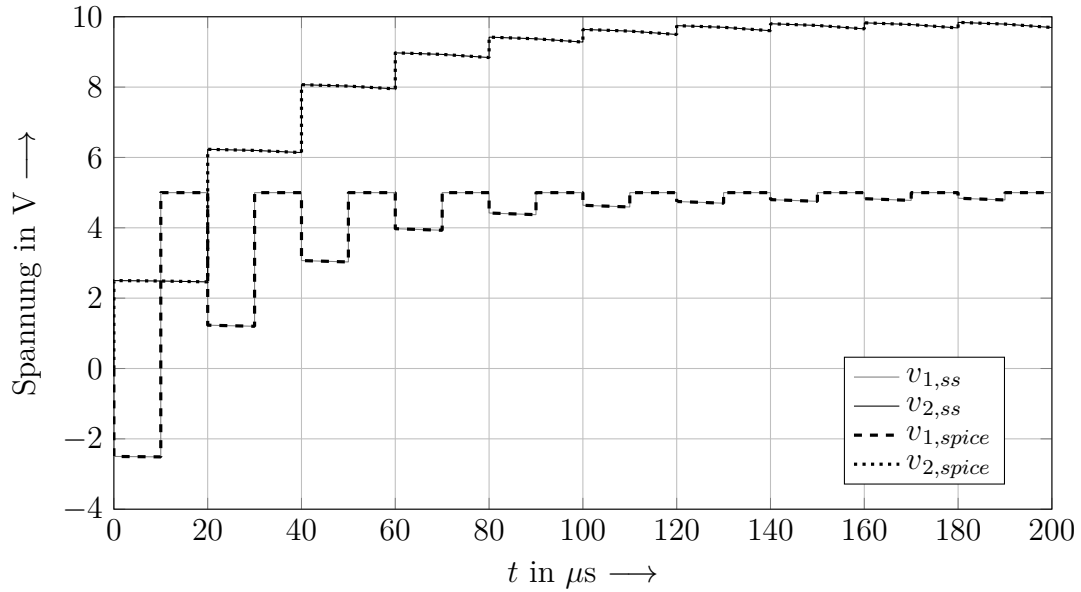


Abbildung 5.19.: Simulationsergebnisse zur Netzliste aus Abb. 5.18, die Simulation in Zustandsraum (Index „ss“) verwendet ideale Schalter ($SW(Ron=0$ $Roff=inf)$), die Simulation in SPICE (Index „spice“) das Schaltermodell aus Abb. 5.18, Zeile 10

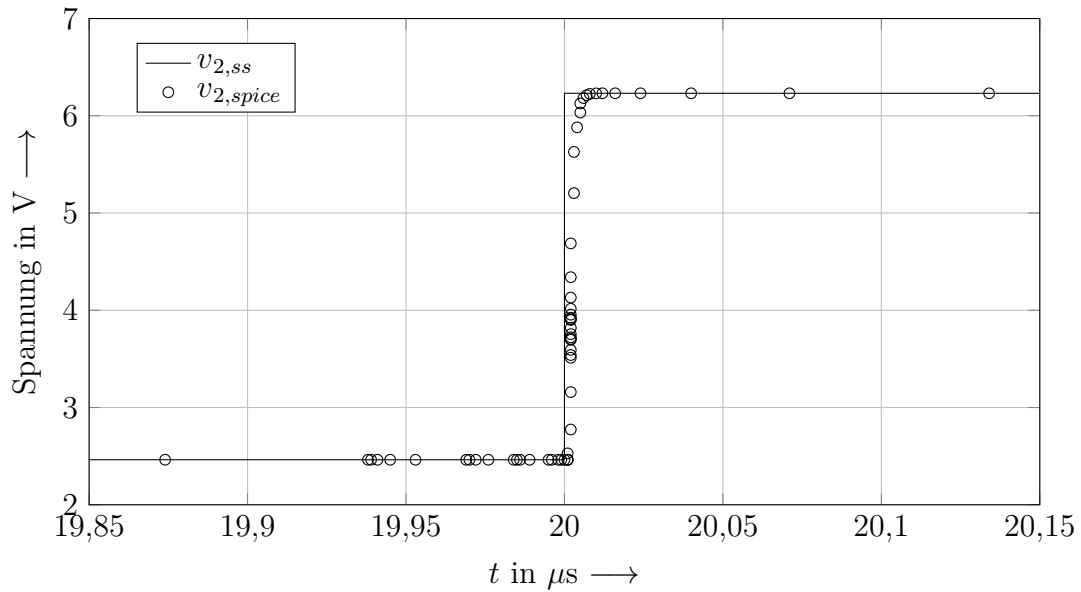


Abbildung 5.20.: Simulationsergebnisse zur Netzliste aus Abb. 5.18, die Simulation in Zustandsraum (Index „ss“) verwendet ideale Schalter ($SW(Ron=0$ $Roff=inf)$), die Simulation in SPICE (Index „spice“) das Schaltermodell aus Abb. 5.18, Zeile 10

5.2.2. DC-DC Schaltwandler mit Spannungsregelung

Als Beispiel einer umfangreicheren getakteten, leistungselektronischen Schaltung dient der geregelte DC-DC-Spannungswandler aus Abb. 5.21. Die Schaltung ist aus [50] entnommen und wird in ähnlicher Weise auch zur Verifikation der Schaltungssimulatoren in [52,96] verwendet. Da in allen Arbeiten, einschließlich der vorliegenden, identische Werte der Schaltungselemente eingesetzt werden, sind die Simulationsergebnisse miteinander vergleichbar.

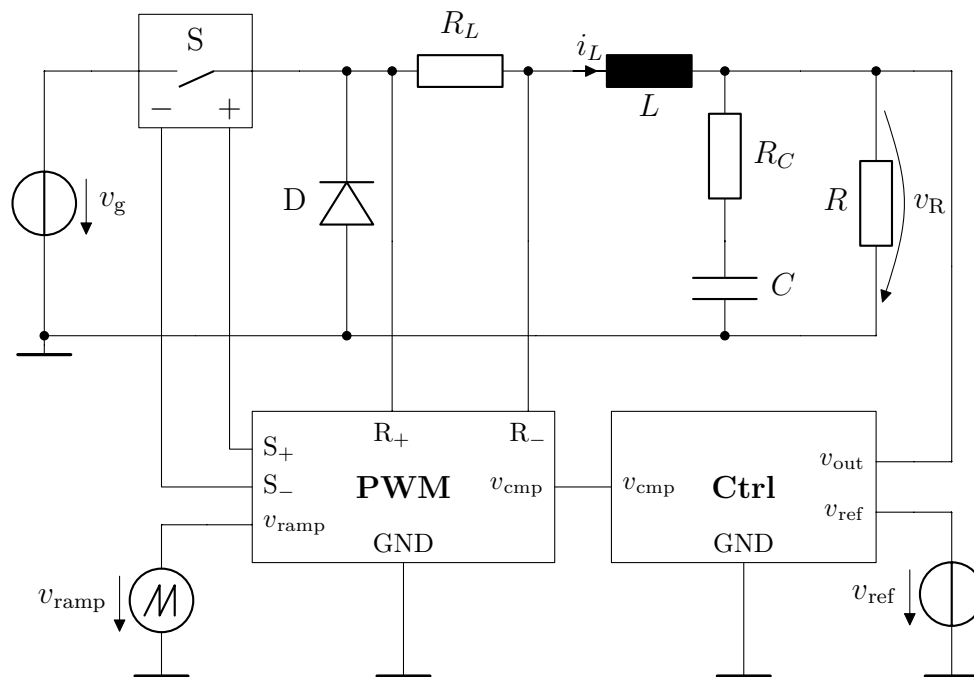


Abbildung 5.21.: Schaltplan eines DC-DC-Spannungswandlers mit Funktionsblöcken der Taktsignalerzeugung (PWM) und Spannungsregelung (Ctrl)

Der DC-DC-Spannungswandler aus Abb. 5.21 setzt die Eingangsspannung v_g auf eine niedrigere Ausgangsspannung herab. Er besitzt in der Literatur mehrere äquivalente Bezeichnungen: Buck-Konverter, Down-Konverter, Tiefsetzsteller, u. a. Das Funktionsprinzip kann, stark vereinfacht, mit Hilfe einer Rechteckspannungsquelle und eines LC-Tiefpassfilters erklärt werden. Im kontinuierlichen Betrieb („Continuous Conduction Mode“, CCM) leitet entweder der Schalter S oder die Diode D ⁹. Die Einheit aus Eingangsspannungsquelle, Schalter und Diode kann daher im CCM durch eine Rechteckspannungsquelle ersetzt werden, an der abwechselnd die Eingangsspannung oder das Massepotential anliegt. Der

⁹ Der lückende Betrieb („Discontinuous Conduction Mode“, DCM) zeichnet sich dadurch aus, dass unterhalb einer gewissen Ausgangsleistung der Spulenstrom negativ werden würde, was jedoch die Diode verhindert. Daraus resultiert ein zusätzliches, drittes Teilintervall, in dem weder Schalter noch Diode leiten.

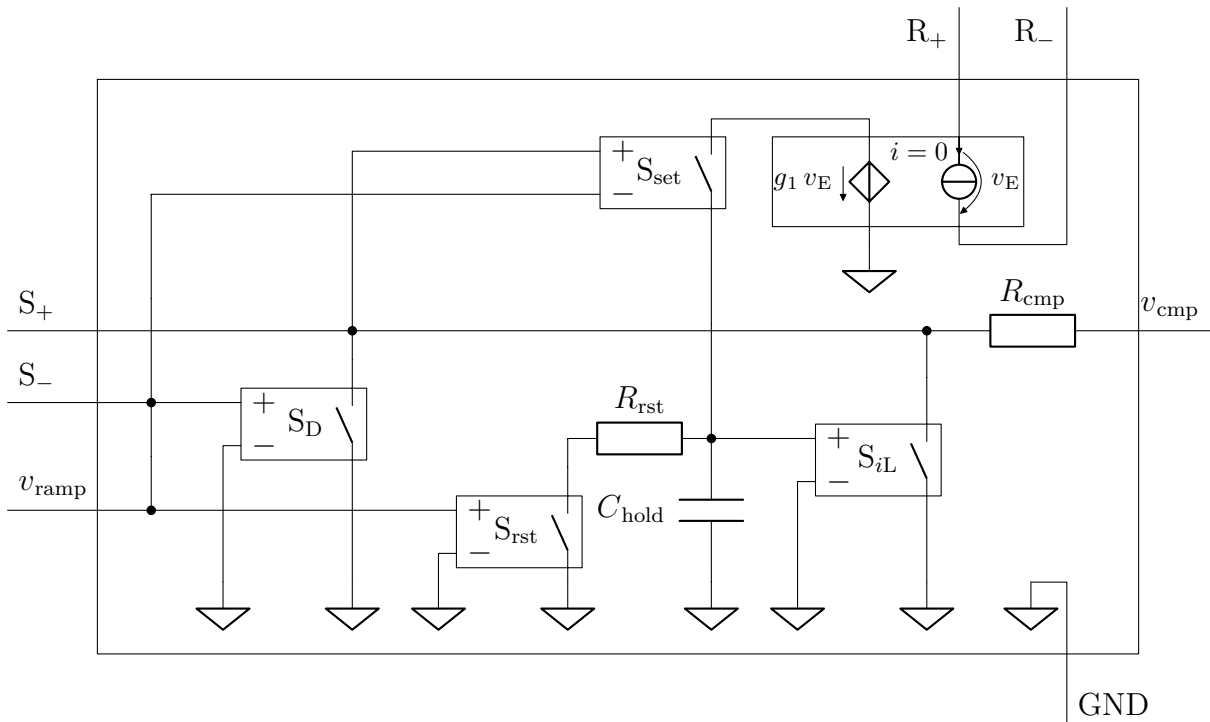


Abbildung 5.22.: Schaltplan des Funktionsblocks der Taktsignalerzeugung mit zusätzlichen Kontrollmechanismen zur Begrenzung des maximalen Induktivitätsstroms und des maximalen Tastgrads (PWM). Die Funktionsweise kann [50] entnommen werden.

Mittelwert dieses Rechtecksignals hängt von der Einschaltdauer des Schalters im Verhältnis zur Taktperiodendauer ab. Dieses Verhältnis wird auch als Tastgrad bezeichnet. Das anschließende, belastete LC-Tiefpassfilter, bestehend aus der Induktivität L , der Kapazität C und dem Widerstand R , besitzt eine Grenzfrequenz deutlich unterhalb der Taktfrequenz, sodass am Ausgang der Schaltung eine Gleichspannung v_R mit einem sehr kleinen überlagerten Wechselanteil zur Verfügung steht. Eine Regelschleife, bestehend aus einem Fehlerverstärker mit Kompensationsnetzwerk (s. Abb. 5.23) sowie einer Einheit zur Taktsignalerzeugung (s. Abb. 5.22), sorgt für eine im Mittel gleichbleibende Ausgangsspannung, unabhängig von Last und Eingangsspannung. Die Einheit zur Taktsignalerzeugung verfügt über die zusätzlichen Kontrollmechanismen der Strombegrenzung und der maximalen Tastgradbegrenzung [50], wie aus dem Schaltplan in Abb. 5.22 ersichtlich wird. Alle Parameter, wie etwa der maximale Induktivitätsstrom als auch der maximale Tastgrad, können mit Hilfe der in der Netzliste in Abb. 5.24 verzeichneten Parameter ermittelt werden.

Abb. 5.25 stellt die Simulationsergebnisse des im Rahmen dieser Arbeit entwickelten Programms und SPICE gegenüber. Die Ergebnisse der Simulation im Zustandsraum (Index „ss“) aus Abb. 5.25a bzw. Abb. 5.25b stimmen gut mit den Ergebnissen aus [50, 52, 96]

5. Verifizierung der Algorithmen und Lösungsstrategien

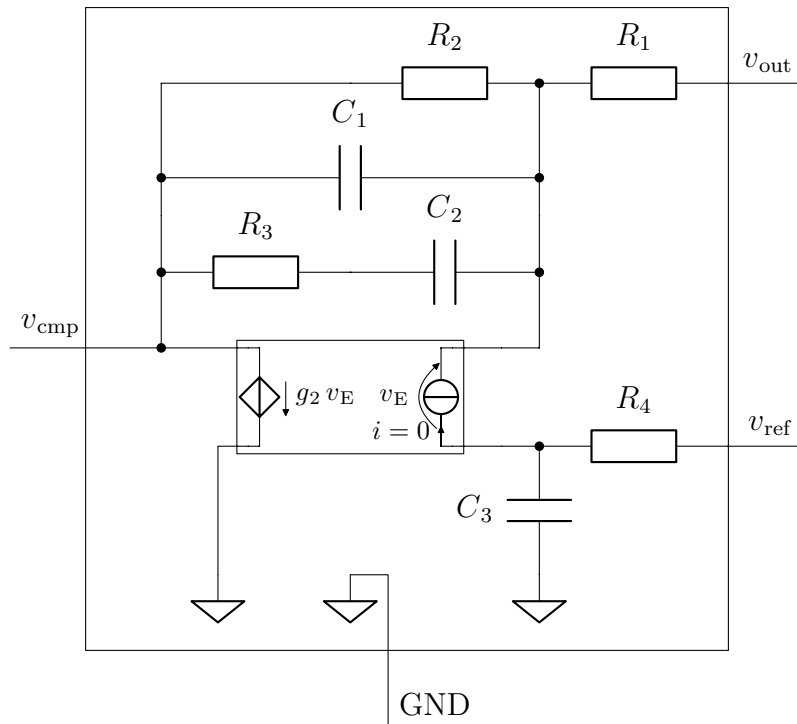


Abbildung 5.23.: Schaltplan des Funktionsblocks der Spannungsregelung (Ctrl) mit Sanftanlauf (R_4 , C_3)

überein. SPICE (Index „spice“) liefert jedoch für die Standard-Einstellungen der Simulationsparameter mit $\text{reltol}=1\text{e-}3$ in Abb. 5.25c bzw. Abb. 5.25d deutliche Abweichungen. Anpassung des Parameters¹⁰ nach $\text{reltol}=1\text{e-}6$ liefert dann die gewünschten Ergebnisse, wie Abb. 5.25e bzw. Abb. 5.25f entnommen werden kann.

Die Simulation des DC-DC Schaltwandlers soll an dieser Stelle zwei Sachverhalte verdeutlichen:

- Umfangreichere Schaltungen mit Regelung und Kontrollmechanismen können auch mit der begrenzten Auswahl an Schaltungselementen sowohl problemlos implementiert als auch simuliert werden und
- die Simulation im Zustandsraum benötigt keine Anpassung der Simulationsparameter, um ein korrektes Ergebnis zu liefern.

¹⁰ Nach [7] wird der Parameter reltol zur Bestimmung der maximal zulässigen Schrittweite verwendet.

5.2. Getaktete, leistungselektronische Schaltungen

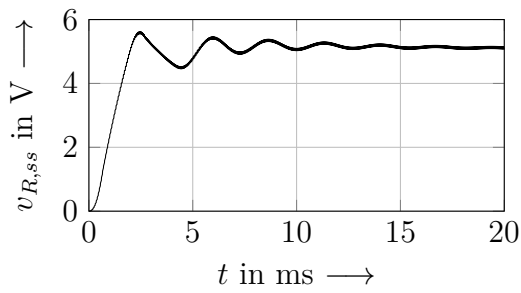
```

1 ** DC-DC Schaltwandler *****
2 Vg      in      0          20
3 Vramp   S-      0      PULSE(0 10 0 {50u-1n} 1n 0 50u)
4 Vref    vRef    0          5
5 S       in      R+      S+      S-      MS0
6 D       0       R+          MD
7 R1      R+      R-          0.25
8 L       R-      vR          200u
9 Rc      vR      c+          0.1
10 C      c+      0          1m
11 R      vR      0          5
12 ** PWM-Block *****
13 E2      E2+     0          R+      R-      4
14 Rcmp    vCmp    S+          1k
15 Rrst    Chold+  Rrst-      0.1
16 Chold   Chold+  0          1u
17 Sd      S+      0          S-      0      MS85
18 Sil     S+      0          Chold+  0      MS4
19 Srst    Rrst-   0          S-      0      MS85
20 Sset    E2+     Chold+    S+      S-      MS0
21 ** Ctrl-Block *****
22 E1      vCmp    0          C3+     R1-     100k
23 R1      vR      R1-        600
24 R2      R1-     vCmp       300k
25 R3      C2-     vCmp       4.7k
26 R4      vRef    C3+        1k
27 C1      R1-     vCmp       2u
28 C2      R1-     C2-        3.3u
29 C3      C3+     0          1.8u
30 ** Schalter und Dioden Modelle *****
31 .model  MD      D(Ron=1m Roff=1Meg)
32 .model  MS0     SW(Ron=1m Roff=1Meg Vt=0)
33 .model  MS85    SW(Ron=1m Roff=1Meg Vt=8.5)
34 .model  MS4     SW(Ron=1m Roff=1Meg Vt=4)
35 ** Simuliere 20ms *****
36 .tran 20m uic
37 .end

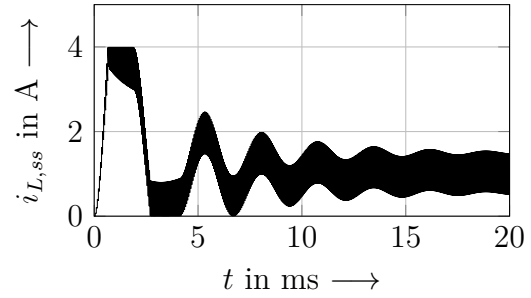
```

Abbildung 5.24.: SPICE-Netzliste zur Schaltung aus Abb. 5.21 bis Abb. 5.23

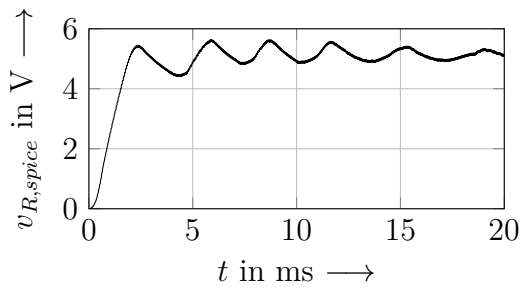
5. Verifizierung der Algorithmen und Lösungsstrategien



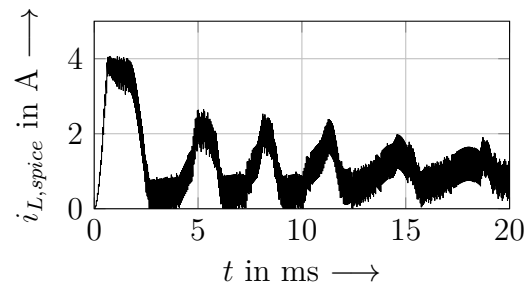
(a) v_R im Zustandsraum



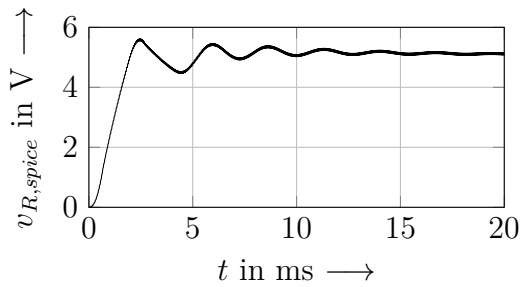
(b) i_L im Zustandsraum



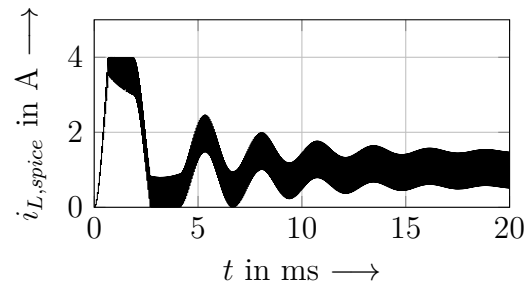
(c) v_R in SPICE mit $\text{reltol}=1\text{e-}3$



(d) i_L in SPICE mit $\text{reltol}=1\text{e-}3$



(e) v_R in SPICE mit $\text{reltol}=1\text{e-}6$



(f) i_L in SPICE mit $\text{reltol}=1\text{e-}6$

Abbildung 5.25.: Simulationsergebnisse zur Netzliste aus Abb. 5.24, die Simulation in Zustandsraum (Index „ss“) verwendet ausschließlich ideale Schalter und Dioden ($R_{on}=0$, $R_{off}=\text{inf}$), die Simulation in SPICE (Index „spice“) die Modelle aus Abb. 5.24, Zeile 31-34

6. Zusammenfassung und Ausblick

Die vorliegende Dissertation beschäftigt sich mit der rechnergestützten, transienten Schaltungsanalyse. Das einführende Kapitel hebt die Bedeutung der rechnergestützten Simulation für die moderne Schaltungsentwicklung hervor. Im anschließenden Kapitel werden die Lösungsstrategien unterschiedlicher Schaltungssimulatoren (darunter auch SPICE) in ihren Grundzügen erklärt und anschließend ihre Eignung zur Simulation getakteter, leistungselektronischer Schaltungen bewertet. Der Vergleich zeigt, dass die Formulierung im Zustandsraum viele Vorteile gegenüber dem üblicherweise verwendeten modifizierten Knotenpotenzialverfahren besitzt. Ebenso zeigt sich, dass die stückweise lineare Modellierung nichtlinearer Schaltungselemente (PWL) für viele leistungselektronische Schaltungen ausreichend genaue Ergebnisse liefert. Gegenüber der in SPICE verwendeten Linearisierung im Arbeitspunkt (Newton-Raphson-Verfahren) kann dadurch in vielen Fällen der Rechenaufwand vermindert und eine potentielle Quelle von Konvergenzproblemen eliminiert werden. Allerdings existiert in der Literatur nicht für jede der Teilaufgaben innerhalb der rechnergestützten Schaltungsanalyse im Zustandsraum eine zufriedenstellende Lösung. So stellt vor allem die Simulation von Schaltungen mit idealen Schaltelementen eine besondere Herausforderung dar. Eine andere Schwierigkeit besteht im effizienten und schnellen Auffinden der Zeitpunkte intern gesteuerter Unstetigkeiten (bspw. den Schaltzeitpunkten idealer Dioden). Der Hauptteil der Arbeit beschäftigt sich daher mit Verfahren, die einen Beitrag dazu liefern die angesprochenen Schwächen zu beheben.

So beschreibt Kapitel 3 die rechnergestützte Formulierung der Zustandsraumdarstellung mittels einer rein algebraischen Methode, die keine besonderen Ansprüche an die Schaltung stellt. Dadurch lassen sich auch sogenannte aktive Netzwerke, d. h. Netzwerke mit gesteuerten Quellen, analysieren. Sämtliche unabhängige Strom- und Spannungsquellen werden durch äquivalente Ersatzschaltungen substituiert, den sogenannten Eingangsgrößenmodellen. Die Beschreibung im Zustandsraum besitzt dadurch weder einen Quellenvektor noch eine Eingangs- bzw. Durchgangsmatrix. Die Problematik der Stromaufteilung parallel geschalteter ideal leitender Schalter oder der Spannungsaufteilung in Reihe geschalteter ideal sperrender Schalter wird durch die Methode der virtuellen Widerstände behoben. Sogenannte versteckte Bedingungen werden sicher erkannt und mittels einer zusätzlichen Gleichung, die sich durch zeitliches Ableiten der letzten Zeile im Gleichungssystem ergibt, gelöst.

Im darauffolgende Kapitel 4 wird die Verarbeitung impulsbehafteter Schaltvorgänge, einer der Hauptschwierigkeiten im Zusammenhang mit idealen Schaltelementen, thematisiert.

6. Zusammenfassung und Ausblick

Die Literatur bezeichnet diesen Umstand auch als inkonsistente Anfangswerte. Mit Hilfe eines Lösungsansatzes basierend auf dem Ladungs- bzw. Flusserhaltungsprinzip lassen sich nach einem impulsbehafteten Schaltvorgang konsistente Anfangswerte finden, die selbst dann korrekt sind, wenn die als stückweise linear modellierten Kapazitäten und Induktivitäten ihre Werte während des Schaltvorgangs ändern. Die auftretenden Impulse werden anhand ihres Gewichts erfasst und können wiederum selbst zum Auffinden der konsistenten Zustände von PWL-Schaltungselementen dienen.

Die transiente Simulation getakteter Leistungselektronik im Zustandsraum mit zeitabhängigen Schaltungselementen (bspw. Pulsquellen) und PWL-Schaltungselementen (bspw. ideale Dioden oder nichtlineare Kapazitäten) besteht aus einer zeitlichen Abfolge (sich häufig periodisch wiederholender) linearer Netzwerke. Sobald eine vorgegebene Zeitschwelle oder die relevante Größe in Form einer Spannungs- bzw. Stromgrenze überschreitet, muss der Schaltungssimulator das Netzwerk entsprechend anpassen. Während die Umschaltzeitpunkte zeitgesteuerter Elemente bekannt sind, stellen spannungs- bzw. stromabhängige PWL-Schaltungselemente für Simulationsprogramme eine Herausforderung dar. Die unbekanntenen Umschaltzeitpunkte müssen stets zuverlässig, exakt und ohne ein Übermaß an Rechenaufwand detektiert und bestimmt werden. Im Rahmen dieser Arbeit wurde dafür ein Ansatz aus der Intervallarithmetik verwendet, welcher die zu überwachenden Zeitverläufe mittels Polynome ersten Grads abschätzt. Durch eine konservative Abschätzung nach oben bzw. unten gelingt eine zuverlässige Eingrenzung und es können keine Netzwerkwechsel übergangen werden. Mittels eines iterativen Vorgehens werden die Zeitintervalle mit möglichem Netzwerkwechsel in immer kleiner werdende Intervalle, bis hin zur Maschinengenauigkeit, eingegrenzt. Die Schrankenfunktionen als Polynome ersten Grads erlauben eine nahezu quadratische Konvergenzordnung, die sich wiederum vorteilhaft auf den Rechenaufwand auswirkt. Allerdings bietet das Verfahren nicht in jedem Fall die gewünschten Vorteile. Ein Beispiel dafür sind nichtlineare oszillatorische Systeme [160]. Sobald der Spannungsverlauf über nichtlinearen Kapazitäten oder der Stromverlauf durch nichtlineare Induktivitäten zu Schwingungen angeregt wird, ist es nicht mehr möglich diese Schwingung mit der hier vorgestellten Methode über mehrere Perioden hinweg abzuschätzen. Eine möglicher Lösungsansatz könnte hier in einem ähnlichen Vorgehen wie bei den Exponentiellen Integratoren liegen [104]. Das bedeutet, dass aus dem nichtlinearen System dominante lineare Komponenten extrahiert werden und die verbleibenden, nichtlinearen dann bspw. durch numerische Integration gelöst werden. Dieser Ansatz wird allerdings im Rahmen der vorliegenden Arbeit nicht weiter untersucht.

Dank der Formulierung des Netzwerks im Zustandsraum minimiert sich der nötige Speicheraufwand, mit dem das Ergebnis der transienten Simulation festgehalten und zu einem späteren Zeitpunkt verlustfrei rekonstruiert werden kann. Wichtige Kenngrößen (bspw. Effektiv- und Mittelwert) können auch ohne Rekonstruktion des Zeitverlaufs, einzig anhand der gespeicherten Zeitpunkte von Netzwerkwechseln, den Anfangswerten und den Systemmatrizen berechnet werden. Die dafür nötigen mathematischen Operationen werden im Rahmen dieser Arbeit hergeleitet.

Wie bereits erwähnt, besteht die transiente Simulation getakteter, leistungselektronischer Schaltungen im Zustandsraum typischerweise aus einer periodischen Abfolge weniger unterschiedlicher Netzwerke mit jeweils veränderten Anfangswerten. Es lohnt sich daher einmal berechnete Ergebnisse abzuspeichern und wiederzuverwenden. Dazu zählen u. a. die Systemmatrizen und die Schalttransformationsgleichungen. Zusätzlich wäre es wünschenswert, Algorithmen zu implementieren, die den eingeschwungenen Zustand, d. h. die Netzwerkabfolge samt den zugehörigen Anfangswerten im stationären Betrieb, direkt ermitteln können [85, 169–174]. Im Rahmen dieser Arbeit wird ein solcher Lösungsansatz zum schnellen Auffinden des stationären Betriebszustands nicht berücksichtigt. Um die Simulationsgeschwindigkeit weiterhin zu erhöhen, könnten zukünftige Arbeiten Strategien untersuchen, welche die besonderen Simulationsbedingungen einer Parametervariation aufgreifen und versuchen den Simulationsaufwand so gering wie möglich zu halten. Denkbar wären eine symbolische Repräsentation der Zustandsraumdarstellung. Die vollständige Neuberechnung aller Matrizen für jeden einzelnen Parametersatz wäre dadurch nicht mehr nötig. Zusätzlich könnten Ergebnisse aus vorhergehenden Simulationsdurchgängen mit nur leicht veränderten Parametern wiederverwendet werden, um bspw. die Zeitpunkte von Netzwerkwechseln schneller bestimmen zu können.

Im abschließenden Kapitel werden anhand einiger Beispielschaltungen die korrekte Funktion des vorgestellten Schaltungssimulators verifiziert und die Ergebnisse mit denen des Schaltungssimulators SPICE verglichen. Es zeigt sich, dass die im Rahmen dieser Arbeit entwickelten Lösungsansätze auch mit schwierigen Situationen umgehen können und in keinem der Fälle auf eine Anpassung der Simulationsparameter durch den Benutzer angewiesen sind, wie dies bei SPICE der Fall ist.

Eine mögliche Erweiterung des Simulationsprogramms besteht in der automatisierten Durchführung einer AC-Analyse. Diese liefert nützliche Erkenntnisse, die für die Auslegung der Regelung leistungselektronischer Schaltungen benötigt werden. Eine besonders für die Zustandsraumdarstellung geeignete Methode stellt das „Sampled-Data Modeling“ dar [175].

Das in der Einleitung definierte Ziel der Arbeit, die Entwicklung eines Simulationsprogramms, welches das Potenzial der vereinfachten, stückweise linearen Modellierung bestmöglich für eine noch effizientere und schnellere Simulation ausnutzt, wird größtenteils erfüllt. Zum einen können viele der Schwierigkeiten, die sich im Zusammenhang mit idealen Schaltelementen ergeben, gelöst werden. Zum anderen wird die Linearität der Teilnetzwerke dahingehend besonders ausgenutzt, dass durch die programmintern in analytischer Form vorliegenden Zeitverläufe der Ströme und Spannungen eine Anwendung von effizienten Methoden der Intervallarithmetik ermöglicht wird.

A. Ergänzungen zu Kapitel 3

A.1. Übersicht der Netzlistensyntax: Schaltungselemente und Modelldefinitionen

SPICE-Netzlisten sind zeilenweise aufgebaut. Das Zeilenende muss nicht gesondert markiert werden, ein Zeilenumbruch genügt. Der Inhalt einer Zeile besteht aus einer der drei folgenden Möglichkeiten:

- Kommentare, gekennzeichnet durch ein vorangestellten Stern „*“
- Direktiven, Modelldefinitionen oder Befehle, gekennzeichnet durch einen vorangestellten Punkt „.“
- Schaltungselemente, gekennzeichnet durch einen vorangestellten Buchstaben „X“.

Eine Netzliste muss aus mindestens einem Schaltungselement, einem Analysebefehl (bspw. „.tran“) und der „.end“-Direktive bestehen, die das Ende der Netzliste kennzeichnet. Die Schaltungselemente, die mit dem in dieser Arbeit vorgestellten Programm kompatibel sind, werden nun aufgelistet und kurz beschrieben. Aufgrund der Beschränkung auf (abschnittsweise) lineare Netzwerke ist der Umfang gegenüber SPICE deutlich eingeschränkt. Bei einigen Elementen ist der Parameterumfang gegenüber SPICE eingeschränkt, bei anderen wiederum erweitert¹. Wie in SPICE üblich informiert eine vom Programm selbstständig generierte Logdatei den Anwender über alle die Simulation betreffenden Warnungen und Fehler, also auch über nicht unterstützte Netzwerkelemente. An dieser Stelle sei nochmals erwähnt, dass ausschließlich das Verbraucherzählpeilsystem verwendet wird. Strom und Spannung jeden Zweigs zeigen in die gleiche Richtung, vom positiven zum negativen Pol.

A.1.1. Widerstände, Kapazitäten und Induktivitäten

Die Syntax für R, L, C-Elemente besteht aus vier durch eine beliebige Anzahl an Leerzeichen voneinander getrennten Einträgen. Die Zeile beginnt mit dem zum Element zugehörigen Buchstaben direkt gefolgt von einer beliebigen Zeichenkombination. Diese dient zur Unterscheidung mehrerer Elemente des gleichen Typs. Anschließend folgen nacheinander die Bezeichnung des Startknotens und des Endknotens, denen der positive beziehungsweise

¹ Das bedeutet, dass die Netzliste nicht mehr vollständig SPICE-kompatibel ist und eine Simulation in SPICE nicht in allen Fällen möglich sein wird.

A. Ergänzungen zu Kapitel 3

negative Pol zugeordnet sind. Die Knotennamen können aus einer beliebigen Zeichenabfolge von Nummern und Buchstaben bestehen². Der letzte Eintrag wird als der jeweilige Bauteilwert in der Grundeinheit (Ohm, Farad und Henry) interpretiert. Tab. A.1 gibt je ein Beispiel für die drei Bauelemente. Zusätzliche Parameter, wie etwa eine Temperaturabhängigkeit oder parasitäre Elemente (ESR, ESL), sind zwar in manchen SPICE-Derivaten zulässig, nicht aber in dieser Arbeit. Sie werden ignoriert und der Anwender über die Logdatei darüber informiert.

Tabelle A.1.: Syntax für Widerstand, Kapazität und Induktivität

	Name	+Knoten	-Knoten	Wert in Ω , F, H
Widerstand	R5	n4	n3	1k
Kapazität	C12	K1	K2	100n
Induktivität	L2	N001	0	100e-6

A.1.2. Gleichspannungs- und Gleichstromquellen

Die Zusammensetzung der Syntax der Gleichspannungs- und Gleichstromquellen entspricht der für Widerstände, Kapazitäten und Induktivitäten. Einzig die Einheiten sind geändert. Zusätzliche Parameter wie parasitäre Elemente (Widerstände, Kapazitäten) sind zwar in SPICE zulässig, nicht aber in dieser Arbeit.

Tabelle A.2.: Syntax für Gleichspannungs- und Gleichstromquellen

	Name	+Knoten	-Knoten	Wert in V, A
Spannungsquelle	V2	N004	N002	12
Stromquelle	I1	43	2	-10m

A.1.3. Sinus-Quellen

Die Syntax der Spannungs- und Stromquellen mit sinusförmigem Zeitverlauf umfasst insgesamt zwölf Einträge. Die ersten drei Einträge entsprechen denen der Gleichspannungs- und Gleichstromquellen. Es folgt ein Identifikator mit Klammer „SINE(“ als vierter Eintrag. Die Bedeutung der restlichen Einträge geht aus Tab. A.3 hervor. Der vorletzte Eintrag, die Periodenanzahl nach der die Quelle nur noch den Offsetwert als Gleichspannung ausgibt, ist optional. Wird nichts angegeben, so wird Periode um Periode bis zum Simulationsende

² Der Knotenname „0“ besitzt in SPICE eine besondere Bedeutung, da er den Bezugsknoten kennzeichnet.

A.1. Übersicht der Netzlistensyntax: Schaltungselemente und Modelldefinitionen

wiederholt. Der letzte Eintrag besteht aus dem Zeichen „)“: Zusätzliche Parameter wie parasitäre Elemente (Widerstände, Kapazitäten) sind zwar in SPICE zulässig, nicht aber in dieser Arbeit.

Tabelle A.3.: Syntax für Sinus-Quellen

	Name	+Knoten	-Knoten	
Spannungsquelle	V0	N1	N2	SINE(
Stromquelle	Isine	2	3	SINE(
	Offset in V, A	Ampl. in V, A	Freq. in Hz	Verzug in s
Spannungsquelle	-0.5k	1k	1Meg	0
Stromquelle	0	10m	50	0
	Dämpfung in $\frac{1}{s}$	Phase in Grad	Perioden	
Spannungsquelle	0	90)
Stromquelle	0	0	10)

A.1.4. Puls-Quellen

Die Syntax der Puls-Quelle besteht insgesamt aus dreizehn Einträgen. Im Gegensatz zur Sinusquelle muss nun statt „SINE(“ der Identifikator „PULSE(“ verwendet werden. Die Angabe einer Periodenanzahl als zwölfter Parameter ist, wie bei der Sinus-Quelle, optional. Der letzte Eintrag besteht aus dem Zeichen „)“. Im Gegensatz zu SPICE sind in dieser Arbeit Anstiegs- und Abfallzeiten von null zulässig. Dadurch ist der Zeitverlauf nicht auf Trapez- bzw. Dreiecksform beschränkt, sondern es können auch Rechteck- und Sägezahnverläufe realisiert werden. Zusätzliche Parameter wie parasitäre Elemente (Widerstände, Kapazitäten) sind zwar in SPICE zulässig, nicht aber in dieser Arbeit. Zur Bedeutung sämtlicher Parameter aus Tab. A.4 sei auf [128] verwiesen.

Tabelle A.4.: Syntax für Pulsquellen

	Name	+Knoten	-Knoten	
Spannungsquelle	V0	N1	N2	PULSE(
Stromquelle	I0	2	3	PULSE(
	Startwert in V, A	Endwert in V, A	t_d in s	t_{rise} in s
Spannungsquelle	-1	1	0	0
Stromquelle	0	1	0	0
	t_{fall} in s	t_{on} in s	T in s	Perioden
Spannungsquelle	0	5u	10u)
Stromquelle	0	1u	2u	10)
t_d : Startverzögerung	t_{fall} : Abfallzeit	T : Periodendauer		
t_{rise} : Anstiegszeit	t_{on} : Einschaltdauer			

A.1.5. Übertrager

Eine magnetische Kopplung von zwei oder mehr Induktivitäten wird in SPICE über eine Zeile beginnend mit „K“ realisiert. Direkt im Anschluss folgt, ohne Leerzeichen, ein beliebiger Name. Es folgen nun mindestens zwei Einträge mit Induktivitätsnamen. Die Gesamtanzahl ist beliebig, allerdings müssen sämtliche Induktivitäten innerhalb der Netzliste definiert sein. Der letzte Eintrag ist der Wert des Kopplungsfaktors, der im Bereich zwischen -1 und 1 liegen muss. Da jeder nichtideale, streubehaftete Übertrager mit Hilfe eines idealen Übertragers und zusätzlichen Induktivitäten modelliert werden kann, wird im Rahmen dieser Arbeit auf dessen Implementierung verzichtet. Den idealen Übertrager als eigenständiges Schaltungselement gibt es in SPICE allerdings nicht. Deshalb wird statt des Kopplungsfaktors der Identifikator „ideal“ als letzter Eintrag verwendet. Die Selbstinduktivitäten werden dann nicht modelliert, sie dienen lediglich zur Bestimmung der Übersetzungsverhältnisse.

Tabelle A.5.: Syntax für ideale Übertrager

	Name	Induktivität	...	Induktivität	Kopplungsfaktor
idealer Übertrager	K7	L1	...	Ln	ideal

A.1.6. Gesteuerte Quellen

Die vier linear gesteuerten Quellen können aufgrund ihrer Syntax in zwei Gruppen eingeteilt werden: spannungsgesteuert und stromgesteuert. Erstere besitzen insgesamt vier

A.1. Übersicht der Netzlistensyntax: Schaltungselemente und Modelldefinitionen

Knoten, wobei die ersten beiden die Anschlüsse der Quelle darstellen und die anderen beiden für den Steuereingang zuständig sind. Quellen mit Stromsteuerung besitzen nur zwei Knoten für die Quelle, der Steuereingang wird über eine Hilfsspannungsquelle realisiert. Bei beiden Gruppen ist der jeweils letzte Eintrag der Zeile die Verstärkung. Im Gegensatz zu SPICE ist es bei stromgesteuerten Quellen nicht unbedingt notwendig eine Hilfsspannungsquelle zu definieren. Statt des Namens einer Spannungsquelle „Vh“ kann in der Syntax der stromgesteuerten Quellen auch jeder andere Elementname verwendet werden, also bspw. „R2“.

Tabelle A.6.: Syntax für spannungsgesteuerte Quellen

	Name	Quelle		Eingang		Verstärkung
		+Knoten	-Knoten	+Knoten	-Knoten	
Spannungsquelle	Etest	N5	N4	N16	N14	1e-3
Stromquelle	G	a	b	c	d	-0.333

Tabelle A.7.: Syntax für stromgesteuerte Quellen

	Name	+Knoten	-Knoten	Hilfsquelle	Verstärkung
Spannungsquelle	H15	N1	N2	Vh	3.14
Stromquelle	F1	N023	N004	R2	1Meg

A.1.7. Schalter

Spannungsgesteuerte Schalter besitzen, ähnlich wie spannungsgesteuerte Quellen, insgesamt vier Knoten. Der Schalter selbst liegt zwischen den ersten beiden Knoten, der Steuereingang an den anderen beiden. Stromgesteuerte Schalter besitzen, genau wie stromgesteuerte Quellen, nur zwei Knoten wobei der Steuereingang über eine Hilfsspannungsquelle (oder jedes andere Bauelement) realisiert ist.

Tabelle A.8.: Syntax für spannungsgesteuerte Schalter

Name	Schalter		Eingang		Modell
	+Knoten	-Knoten	+Knoten	-Knoten	
S2	N2	N2	N4	N23	MySwitch

Statt eines Werts besitzen Schalter als letzten Eintrag ihrer Syntax einen frei wählbaren Modellnamen, im Beispiel nach Tab. A.8 und A.9 ist dies der Eintrag „MySwitch“. Dieses

A. Ergänzungen zu Kapitel 3

Tabelle A.9.: Syntax für stromgesteuerte Schalter

Name	+Knoten	-Knoten	Hilfsquelle	Modell
<code>Wsw</code>	<code>N3</code>	<code>N5</code>	<code>V3</code>	<code>MySwitch</code>

Schaltermmodell muss zusätzlich an anderer Stelle in der Netzliste definiert sein. Die Syntax eines Schalters besteht demnach aus zwei Teilen. Während die Standard-SPICE-Syntax diverse Modellparameter vorsieht, werden in Tab. A.10 nur zwei berücksichtigt. Zum einen der Schwellwert, ab dem der Schalter seinen Zustand wechselt, und zum anderen ein Hysteresewert, der zum Schwellwert hinzuaddiert bzw. subtrahiert wird, um eine Schalthysterese zu erzeugen. Alle anderen Parameter werden ignoriert, worüber der Anwender in der Logdatei informiert wird.

Tabelle A.10.: Syntax für Schaltermodelle

Name	Schwellwert in V/A	Hysteresewert in V/A
<code>.model MySwitch SW(</code>	<code>Vt=0.75</code>	<code>Vh=0.25</code>

A.1.8. Ideale Dioden

Die Syntax der Diode besteht aus zwei Teilen: Die Zeile mit dem Namen, den beiden Knoten und dem Modellname sowie eine Zeile mit der Modelldefinition. Die Standard-SPICE-Syntax sieht diverse Modellparameter vor, um das Verhalten einer realen Diode so genau wie möglich vorherzusagen. Es wird allerdings kein einziger davon berücksichtigt, worüber der Anwender in der Logdatei informiert wird. Bei Bedarf kann die Diodenkennlinie mit zusätzlichen Schaltungselementen beeinflusst werden. Häufig genügt eine Kombination aus Spannungsquelle und Widerstand in Reihe zur Diode, um ihre Kennlinie in Vorwärtsrichtung mit zwei Geradenstücken annähern zu können [176]. Diese müssen dann explizit in der Netzliste als eigenständige Elemente vorgesehen werden.

Tabelle A.11.: Syntax der Diode

Name	+Knoten (Anode)	-Knoten (Kathode)	Modell
<code>D2</code>	<code>N01</code>	<code>0</code>	<code>~</code>

A.2. Übersicht der Netzlistensyntax: Befehle und Direktiven

Alle Befehle und Direktiven, die mit dem in dieser Arbeit vorgestellten Programm kompatibel sind, werden in diesem Kapitel aufgelistet und kurz beschrieben.

A.2.1. Transiente Analyse und eingeschwungener Zustand

Die Anweisung zur transienten Simulation beinhaltet mehrere Zeitparameter. Das Abspeichern von Zeitverläufen benötigt einen Zeitschritt, mit dem die Signale abgetastet werden. Diese Schrittweite wird über den Parameter T_{step} mitgeteilt³. Den Anfang und das Ende der Aufzeichnung geben die beiden Parameter t_{start} bzw. t_{end} an. Der Parameter dT_{max} ist nur für SPICE relevant, da er den maximalen Zeitschritt der numerischen Integrationsmethode angibt.

Tabelle A.12.: Syntax für transiente Analyse

	T_{step}	t_{end}	t_{start}	dT_{max}	
Transiente Analyse	.TRAN	100n	100m	90m	~
T_{step} : Schrittweite in s		t_{start} : Startzeitpunkt in s		t_{end} : Endzeitpunkt in s	dT_{max} : max. Schrittweite in s

A.2.2. Zeitverläufe

Mit der Direktive „.SAVE“ kann dem Simulationsprogramm mitgeteilt werden, welche Größen abgespeichert werden sollen. Dies können sämtliche Ströme und Spannungen innerhalb der Schaltung sein. Tab. A.13 zeigt mehrere Möglichkeiten, wie dem Programm die gewünschten Spannungen mitgeteilt werden können. Der erste Eintrag „V(R1)“ gibt die Zweigspannung über dem Widerstand R1 aus, der zweite Eintrag „V(N01,N05)“ gibt die Spannungsdifferenz zwischen den Knoten N01 und N05 aus. Darüber hinaus gibt es noch die Möglichkeit nur einen Knoten anzugeben, bspw. „V(N01)“, wodurch die Spannungsdifferenz zwischen dem Knoten N01 und dem Bezugsknoten (Bezeichner: 0) ausgegeben wird. Das Schlüsselwort `dialogbox` ermöglicht die nachträgliche Auswahl der gewünschten Spannungen und Ströme aus einer Liste, die nach Abschluss der Simulation in einem Dialogfenster geöffnet wird. Die Schrittweite, mit der die Signale abgetastet und abgespeichert werden, sowie der Aufzeichnungszeitraum werden über den Befehl .TRAN (s. Kapitel A.2.1) mitgeteilt.

³ Bei manchen SPICE-Derivaten ist der Parameter T_{step} optional.

A. Ergänzungen zu Kapitel 3

Tabelle A.13.: Syntax für das Abspeichern von Zeitverläufen

	V/I	V/I	...	V/I	Schlüsselwort
.SAVE	V(R1)	V(N01,N05)	...	I(S1)	
.SAVE	~	~	...	~	dialogbox

V: Spannung I: Strom

A.2.3. Kennwerte

Zusätzlich zu den Zeitverläufen (s. Kapitel zur Direktive `.SAVE`) können bei Bedarf auch Kennwerte ausgewählter Größen berechnet und abgespeichert werden. Die Standard-SPICE-Syntax sieht dafür die Direktive `.MEAS` vor. Während der Funktionsumfang in SPICE deutlich mehr Möglichkeiten bietet [128], soll sich in dieser Arbeit auf die wesentlichen konzentriert werden. Das Zeitintervall, welches allen `.MEAS`-Direktive zugrunde liegt, wird mittels des `.TRAN`-Befehls definiert. Die Zeitspanne reicht entweder von t_{start} bis t_{end} (s. Tab. A.12) oder über eine Periode des eingeschwungenen Zustands (Modifikator `steady` nach Tab. A.12). Tab. A.14 gibt je ein Beispiel für jede Auswertemöglichkeit.

Tabelle A.14.: Syntax für das Abspeichern von Kennwerten

		Name	Identifikator	Größe
Mittelwert	.MEAS	Messung1	AVG	V(N01,N05)
Maximalwert	.MEAS	Messung2	MAX	V(N03)
Minimalwert	.MEAS	Vcmin	MIN	V(C2)
Effektivwert	.MEAS	IL5RMS	RMS	I(L5)
Integral	.MEAS	Vr2integ	INTEG	V(R2)

A.2.4. Anfangswerte

Falls vom Anwender keine anderen Anfangswerte vorgegeben sind, so werden zu Beginn der transienten Analyse alle Kondensatorspannungen und Induktivitätsströme zu null gesetzt. Die Angabe von Anfangswerten mittels der `.IC`-Anweisung ist deshalb optional. Sollten die vorgegebenen Anfangswerte nicht konsistent sein, korrigiert die in Kapitel 4.2 vorgestellte Vorgehensweise diese automatisch. Der Anwender wird in diesem Fall, wie auch in SPICE üblich, über einen Eintrag in der Logdatei über die tatsächlichen Anfangswerte informiert. Im Gegensatz zur Standard-SPICE-Syntax akzeptiert die `.IC`-Anweisung ausschließlich Kondensatorspannungen und Induktivitätsströme.

Tabelle A.15.: Syntax für Anfangswerte

	V/I	V/I	...	V/I
.IC	V(C23)=10	I(L11)=-2	...	V(C3)=1

A.3. Formulieren der Zustandsdifferentialgleichung

Nicht jedes Netzwerk führt mit der in Kapitel 3.3 gezeigten Methode unmittelbar auf ein Zustandsdifferentialgleichungssystem in Normalform. In Kapitel 3.3.1 ergibt sich für die gegebene Schaltung des Buck-Boost-Wandlers und der speziellen Schaltkombination zwar ein Gleichungssystem, das eine direkte Formulierung in Normalform erlaubt, jedoch ist dies eher als eine Ausnahme anzusehen. Ein allgemeingültiges Simulationsprogramm sollte jedoch mit sämtlichen Schaltungen und Schaltkombinationen umgehen können, für die eine physikalisch sinnvolle Lösung des Netzwerks existiert. Auch mehrdeutige Fälle⁴ sollen nicht ausgeschlossen werden. Einzig die Fälle, in denen das Netzwerk keine Lösung zulässt, bspw. für den Fall, dass zwei unabh. Spannungsquellen parallel geschaltet sind, können und sollten mit einer Fehlermeldung ausgeschlossen werden.

Ausgangspunkt der nachfolgenden Betrachtungen ist die Matrix des Gleichungssystems, welches sich aus den Kirchhoff'schen Regeln, den Strom-/Spannungsbeziehungen der Bauelemente und der Eingangsgrößenmodellierung, d. h. dem Ersetzen der unabhängigen Quellen durch äquivalente Ersatznetzwerke, ergibt. Für die weitere Vorgehensweise müssen die Spalten der Matrix von links nach rechts nach der Reihenfolge der folgenden Vektoren, wie in (A.1) zu sehen, sortiert sein [131]:

1. \mathbf{w} enthält alle Spannungen und Ströme der Widerstände, gest. Quellen⁵ und idealen Übertrager
2. $\tilde{\mathbf{u}}_{\text{aux}}$ enthält alle Ströme der Hilfsspannungsquellen und Spannungen der Hilfsstromquellen⁶
3. $\dot{\mathbf{x}}$ enthält die Ableitungen aller Kondensatorspannungen und Induktivitätsströme
4. \mathbf{x} enthält alle Kondensatorspannungen und Induktivitätsströme
5. \mathbf{u}_{aux} enthält alle (nullwertigen) Ströme der Hilfsstromquellen und (nullwertigen) Spannungen der Hilfsspannungsquellen

Das Gleichungssystem nach (A.1) zeigt die prinzipielle Struktur der sortierten und in

⁴ Ein mehrdeutiger Fall tritt in leistungselektronischen Schaltungen bspw. dann auf, wenn zwei parallel liegende, ideale Schalter gleichzeitig eingeschaltet sind. Die Stromaufteilung auf beide Schalter ist dann nicht eindeutig bestimmbar. Die Behandlung solcher Mehrdeutigkeiten ist Gegenstand von Kapitel A.3.1.

⁵ Die nullwertige Größe der Hilfsquelle am Steuereingang ist nicht im Vektor \mathbf{w} sondern im Vektor \mathbf{u}_{aux} .

⁶ Hilfsquellen sind alle unabh. Quellen, die nicht explizit vom Anwender in der Netzliste übergeben werden.

Zeilenstufenform gebrachten Matrix.

$$\begin{bmatrix} \mathbf{T}^{(0)} & \dots & \dots & \dots & \dots \\ & \mathbf{Q}^{(0)} & \dots & \dots & \dots \\ & & \mathbf{M}^{(0)} & -\mathbf{A}^{(0)} & -\mathbf{B}_{\text{aux}}^{(0)} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \tilde{\mathbf{u}}_{\text{aux}} \\ \dot{\mathbf{x}} \\ \mathbf{x} \\ \mathbf{u}_{\text{aux}} \end{bmatrix} = \mathbf{0}. \quad (\text{A.1})$$

Durch die Zeilenstufenform ist es ausreichend, nur die letzten Zeilen der Matrix zu betrachten, um eine Aussage über die notwendigen Schritte treffen zu können. Es genügt also der Teil

$$\begin{bmatrix} \mathbf{M}^{(0)} & -\mathbf{A}^{(0)} & -\mathbf{B}_{\text{aux}}^{(0)} \end{bmatrix} \quad (\text{A.2})$$

aus der Gesamtmatrix. Nach jedem Schritt wird die Matrix wieder in Zeilenstufenform gebracht und die Entscheidungsfindung beginnt erneut. Es entsteht ein iterativer Algorithmus [4]. Dieser Prozess wird solange durchlaufen, bis sich entweder eine Normalform finden lässt oder sich das Netzwerk als nicht lösbar herausstellt. Zur Kennzeichnung des Iterationsdurchgangs wird ein hochgestellter Zählindex in Klammern (n) eingeführt. Dieser ist zu Beginn null (vgl. (A.1)) und erhöht sich mit jedem Iterationsschritt um eins. Das Vorgehen in jedem Iterationsschritt ist davon abhängig, welche Untermatrizen ausschließlich eine nullwertige letzte Zeile aufweisen und in welchen Einträge ungleich null vorhanden sind. Nach [4] gibt es drei Fälle:

1. Alle Einträge der letzten Zeilen aller Untermatrizen sind nullwertig.
2. Alle Einträge der letzten Zeilen in $\mathbf{M}^{(0)}$ und $\mathbf{A}^{(0)}$ sind nullwertig, einer oder mehrere in $\mathbf{B}_{\text{aux}}^{(0)}$ nicht.
3. Alle Einträge der letzten Zeile in $\mathbf{M}^{(0)}$ sind nullwertig, einer oder mehrere sowohl in $\mathbf{A}^{(0)}$ als auch in $\mathbf{B}_{\text{aux}}^{(0)}$ nicht.

Besteht die letzte Zeile ausschließlich aus nullwertigen Einträgen (erster Fall), so wird die Simulation abgebrochen. Das Netzwerk lässt keine Lösung zu. Tritt der zweite Fall ein, wird die Methode der virtuellen Widerstände eingesetzt. Im dritten Fall reduziert sich die Anzahl der Zustandsvariablen um eins. Das Vorgehen für die beiden letztgenannten Lösungsansätze wird in den folgenden Abschnitten näher erläutert.

A.3.1. Methode der virtuellen Widerstände

Eine in der Leistungselektronik häufig anzutreffende Schaltung ist der aus vier Dioden bestehende Brückengleichrichter. In Abb. A.1 wird er als Netzgleichrichter mit Glättungskondensator und ohmscher Last eingesetzt. Möchte der Anwender den Gleichrichter mit idealen Dioden simulieren, so müsste die Simulation nach kurzer Zeit abgebrochen werden. Der Grund liegt in der Sperrphase, in der alle vier Dioden sperren. Während dieser Zeit besteht die Schaltung aus zwei getrennten Netzwerken, auf der einen Seite die Quelle und

A.3. Formulieren der Zustandsdifferentialgleichung

auf der anderen der Kondensator mit Last. Es gibt keine eindeutig festgelegte Potentialdifferenz zwischen beiden Teilnetzwerken. Die Spannungen über den Dioden können nicht eindeutig ermittelt werden, das zugehörige Gleichungssystem ist unterbestimmt. Eine einfache Lösung dieses Problems ist der Verzicht auf ideale Dioden. Der Anwender wird dazu gezwungen, einen Widerstandswert anzugeben, der im Fall der sperrenden Diode statt des Leerlaufs wirksam wird. Übliche Widerstandswerte liegen im Mega-/Gigaohmbereich [128] und verursachen damit einen kleinen Sperrstrom. Der durch den Sperrstrom verursachte Spannungsabfall ergibt direkt die gesuchte Sperrspannung über der jeweiligen Diode.

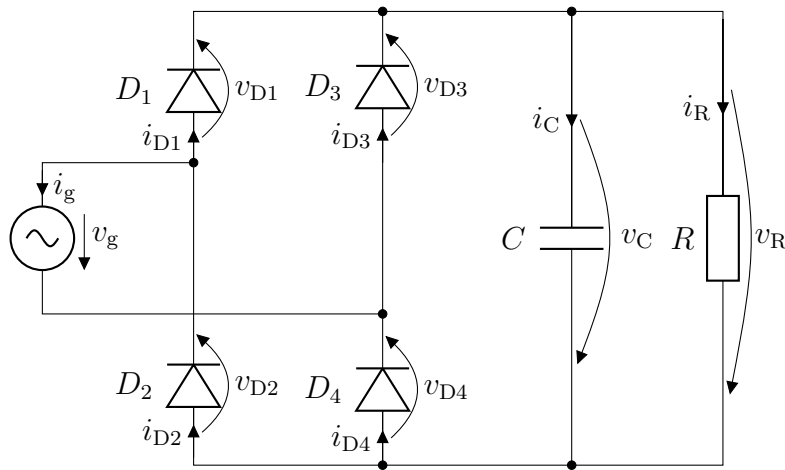


Abbildung A.1.: Brückengleichrichter mit Kondensator und ohmscher Last

Alternativ zur Einführung von Ersatzwiderständen bietet sich die Verwendung eines unendlich kleinen bzw. unendlich großen Widerstands an [73]. Für das Beispiel des sperrenden Brückengleichrichters bedeutet dies, dass statt der vier sperrenden Dioden vier unendlich große Widerstände eingesetzt werden. Die Vorgehensweise aus Kapitel 3⁷ führt zunächst

⁷ Aus Gründen der Übersichtlichkeit wurde hier auf die Eingangsgrößenmodellierung verzichtet. Das Vorgehen aus Kapitel 3 würde die Sinusquelle durch ein Quellenmodell mit Kapazität, Induktivität und Widerstand, sowie einer gesteuerten Spannungsquelle ersetzen.

A. Ergänzungen zu Kapitel 3

auf die Matrix in (A.3).

$$\begin{array}{r}
 \begin{array}{c} i_R \quad v_R \quad i_g \quad v_{D1} \quad v_{D2} \quad v_{D3} \quad v_{D4} \quad \dot{v}_C \quad v_C \quad i_{D1} \quad i_{D2} \quad i_{D3} \quad i_{D4} \quad v_g \\
 \left[\begin{array}{cccccccc|c|c|cccc|c}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & C & 0 & -1 & 0 & -1 & 0 & 0 \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 & & & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 & & & & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & & & & & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & 0 & 1 & \frac{1}{RC} & -\frac{1}{C} & 0 & -\frac{1}{C} & 0 & 0 \\
 & & & & & & & 0 & 0 & 1 & -1 & 1 & -1 & 0 \\
 \end{array} \right. \\
 \end{array}
 \end{array} \quad (A.3)$$

Die vorletzte Zeile in (A.3) beinhaltet bereits die gesuchte Differentialgleichung. Es wäre deshalb unnötig die Simulation nur wegen der Problematik der ungewissen Sperrspannungen abzurechnen. Die letzte Zeile liefert:

$$i_{D1} - i_{D2} + i_{D3} - i_{D4} = 0. \quad (A.4)$$

Diese Gleichung besitzt keinen Informationsgehalt, denn die vier Diodenströme i_{D1} bis i_{D4} sind während der Sperrphase null. Wenn die Dioden allerdings statt durch einen Leerlauf nun durch virtuelle, unendlich große Sperrwiderstände R_{sperr} ersetzt werden, so ergibt sich ein Zusammenhang zwischen Sperrstrom und -spannung.

$$v_{Di} = R_{\text{sperr}} i_{Di} \quad (A.5)$$

Wird Gleichung (A.4) auf beiden Seiten mit R_{sperr} multipliziert und anschließend der Zusammenhang aus Gleichung (A.5) eingesetzt, so ergibt sich:

$$v_{D1} - v_{D2} + v_{D3} - v_{D4} = 0. \quad (A.6)$$

Diese Gleichung ersetzt nun die bisherige letzte Zeile der Matrix, was in (A.7) durch die

A.3. Formulieren der Zustandsdifferentialgleichung

grau eingefärbten Elemente illustriert ist.

$$\begin{array}{c}
 \tilde{\mathbf{u}}_{\text{aux}}^T \qquad \qquad \mathbf{u}_{\text{aux}}^T \\
 \begin{array}{cccccccc|cc|cccc|c}
 i_R & v_R & i_g & v_{D1} & v_{D2} & v_{D3} & v_{D4} & \dot{v}_C & v_C & i_{D1} & i_{D2} & i_{D3} & i_{D4} & v_g \\
 \hline
 1 & 0 & 0 & 0 & 0 & 0 & 0 & C & 0 & -1 & 0 & -1 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{RC} & -\frac{1}{C} & 0 & -\frac{1}{C} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 \\
 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array}
 \end{array} \quad (\text{A.7})$$

Elementare Zeilenumformungen erlauben schließlich die Darstellung nach (A.8) in Zeilenstufenform.

$$\begin{array}{c}
 \begin{array}{cccccccc|cc|cccc|c}
 i_R & v_R & i_g & v_{D1} & v_{D2} & v_{D3} & v_{D4} & \dot{v}_C & v_C & i_{D1} & i_{D2} & i_{D3} & i_{D4} & v_g \\
 \hline
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{R} & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 & & & 1 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} \\
 & & & & 1 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} \\
 & & & & & 1 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} \\
 & & & & & & 1 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} \\
 & & & & & & & 1 & \frac{1}{RC} & -\frac{1}{C} & 0 & -\frac{1}{C} & 0 & 0
 \end{array}
 \end{array} \quad (\text{A.8})$$

Im Allgemeinen zeichnet sich der Fall notwendiger virtueller Widerstände durch eine bestimmte Form der letzten Zeile der Matrix (in Zeilenstufenform) aus:

- Die letzte Zeile der Untermatrix $\mathbf{M}^{(n)}$ muss ausschließlich aus Nullen bestehen.
- Die letzte Zeile der Untermatrix $\mathbf{A}^{(n)}$ muss ausschließlich aus Nullen bestehen.
- Die letzte Zeile der Untermatrizen $\mathbf{B}_{\text{aux}}^{(n)}$ muss mindestens einen Eintrag ungleich null aufweisen.

In allgemeiner Schreibweise ergibt sich für die letzte Zeile dann:

$$\begin{aligned}
 & [0 \ 0 \ \dots \ 0] \tilde{\mathbf{u}}_{\text{aux}} + [0 \ 0 \ \dots \ 0] \dot{\mathbf{x}} + \\
 & \qquad \qquad \qquad [0 \ 0 \ \dots \ 0] \mathbf{x} + [b_1 \ b_2 \ \dots \ b_n] \mathbf{u}_{\text{aux}} = 0. \quad (\text{A.9})
 \end{aligned}$$

A. Ergänzungen zu Kapitel 3

Diese Zeile wird nun ersetzt durch die neue Zeile

$$\begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} \tilde{\mathbf{u}}_{\text{aux}} + \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \dot{\mathbf{x}} + \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \mathbf{u}_{\text{aux}} = 0. \quad (\text{A.10})$$

in der die Elemente b_1 bis b_n aus dem Bereich des Vektors \mathbf{u}_{aux} in den Bereich des Vektors $\tilde{\mathbf{u}}_{\text{aux}}$ verschoben werden. Diese Vorgehensweise ist gleichbedeutend mit der Einführung virtueller Widerstände. Im Beispiel der Matrix aus (A.7) werden demnach die von null verschiedenen Einträge der letzten Zeile um sechs Spalten nach links verschoben.

A.3.2. Reduzierung der Anzahl an Zustandsvariablen

Die Vorgehensweise nach Kapitel 3.3.1 sieht zunächst alle Kapazitätsspannungen und Induktivitätsströme im Netzwerk als Zustandsvariablen an. Diese Annahme trifft zwar in vielen Fällen zu, wie das folgende Beispiel jedoch zeigt, ist dies nicht allgemeingültig. Durch eine geeignete Methode müssen deshalb diejenigen Größen identifiziert werden, die nicht zu den Zustandsvariablen gezählt werden können. Als Beispiel einer Schaltung, die

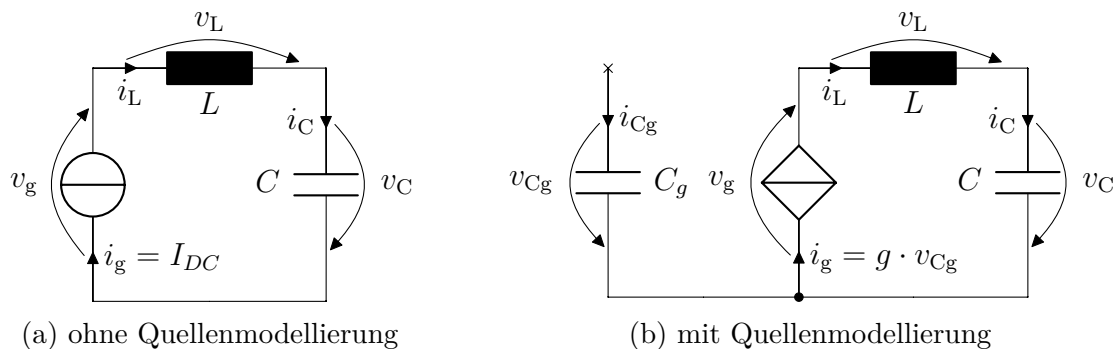


Abbildung A.2.: Beispielschaltung mit einer statt zwei Zustandsgrößen

weniger Zustandsvariablen als Kapazitäten und Induktivitäten besitzt, soll die einfache Reihenschaltung aus Abb. A.2a dienen. Die Vorgehensweise nach Kapitel 3.3.1 ersetzt die Gleichstromquelle durch das zugehörige Quellenmodell (vgl. Abb. A.2b). Außerdem wird zunächst angenommen, dass die Spannungen v_C und v_{C_g} als auch der Strom i_L Zustandsgrößen sind. Die Matrix in Zeilenstufenform aus (A.11) zeigt das vorläufige

A.3. Formulieren der Zustandsdifferentialgleichung

Ergebnis.

$$\begin{array}{cccc|ccc} v_g & i_g & \dot{v}_C & \dot{i}_L & \dot{v}_{Cg} & v_C & i_L & v_{Cg} \\ \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & L & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -g \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{g}{C} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -g \end{array} \right] \end{array} \quad (\text{A.11})$$

$\mathbf{M}^{(0)} \qquad \qquad \mathbf{-A}^{(0)}$

Die letzte Zeile der Untermatrix $\mathbf{M}^{(0)}$ in (A.11) besteht ausschließlich aus Nullen. $\mathbf{M}^{(0)}$ kann daher nicht in Einheitsform gebracht werden, wie es für das direkte Ablesen der Zustandsdifferentialgleichung nötig ist. Stattdessen liefert die letzte Zeile aus (A.11) den Zusammenhang:

$$i_L = g \cdot v_{Cg}. \quad (\text{A.12})$$

Der Induktivitätsstrom i_L ist demnach zu jedem Zeitpunkt durch die Kapazitätsspannung v_{Cg} und Transkonduktanz g vorgegeben. Dies gilt auch für den Zeitpunkt $t = 0$, weshalb der Anfangswert $i_L(0)$ nicht mehr frei gewählt werden kann. Die Größe i_L verliert dadurch den Status einer Zustandsgröße⁸. Die zugehörigen Spalten mit i_L und \dot{i}_L müssen daher durch Spaltenverschiebung aus den Untermatrizen $\mathbf{M}^{(0)}$ bzw. $\mathbf{A}^{(0)}$ entfernt werden. Das Ergebnis in Zeilenstufenform zeigt die Matrix aus (A.13).

$$\begin{array}{cccc|ccc} v_g & i_g & \dot{i}_L & i_L & \dot{v}_C & \dot{v}_{Cg} & v_C & v_{Cg} \\ \left[\begin{array}{cccc|ccc} 1 & 0 & L & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -g \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -g \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -\frac{g}{C} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right] \end{array} \quad (\text{A.13})$$

Durch Entfernen einer Variablen aus der Gruppe der Zustandsgrößen reduzieren sich entsprechend auch die Dimensionen der Untermatrizen $\mathbf{M}^{(1)}$ bzw. $\mathbf{A}^{(1)}$. Das gesuchte Differentialgleichungssystem kann nun mit Hilfe der letzten beiden Zeilen aus (A.13) direkt angegeben werden. Dennoch ist das Netzwerk nicht vollständig lösbar. Die Größen \dot{i}_L und damit auch die Größe v_g können nicht bestimmt werden. Es fehlt eine zusätzliche Gleichung. Diese ergibt sich durch einmalige zeitliche Ableitung der letzten Zeile in (A.11):

$$\dot{i}_L = g \cdot \dot{v}_{Cg}. \quad (\text{A.14})$$

⁸ In [139] werden alle Größen eines Netzwerks als Zustände bezeichnet, deren Start- bzw. Anfangswerte notwendig und zugleich ausreichend sind, um die Zeitverläufe aller Größen im Netzwerk ab dem Startzeitpunkt eindeutig bestimmen zu können.

A. Ergänzungen zu Kapitel 3

Die Matrix in reduzierter Zeilenstufenform und mit (A.14) als zusätzlicher Zeile lautet⁹:

$$\begin{array}{cccc|cc|cc}
 v_g & i_g & \dot{i}_L & i_L & \dot{v}_C & \dot{v}_{C_g} & v_C & v_{C_g} \\
 \left[\begin{array}{cccc|cc|cc}
 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & -g \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & -g \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & -\frac{g}{C} \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
 \end{array} \right] & & & & & & & & (A.15)
 \end{array}$$

Die Differentialgleichung ergibt sich aus den letzten beiden Zeilen der Matrix in (A.15). Zudem können nun (nach vorheriger Lösung der Differentialgleichung) sämtliche Größen des Netzwerks ermittelt werden.

Im Allgemeinen zeichnet sich der Fall einer verschwindenden Zustandsgröße durch eine bestimmte Form der letzten Zeile der Matrix (in Zeilenstufenform) aus:

- Die letzte Zeile der Untermatrix $\mathbf{M}^{(n)}$ muss ausschließlich aus Nullen bestehen.
- Die letzte Zeile der Untermatrix $\mathbf{A}^{(n)}$ muss mindestens einen Eintrag ungleich null besitzen.
- Die letzte Zeile der Untermatrix $\mathbf{B}_{\text{aux}}^{(n)}$ ist unerheblich und wird im folgenden nicht weiter berücksichtigt.

In allgemeiner Schreibweise ergibt sich für die letzte Zeile dann:

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \dot{\mathbf{x}} + \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \mathbf{x} = 0. \quad (A.16)$$

Ein beliebiges Element x_j des Vektors $\mathbf{x}^{(n)}$, dessen Koeffizient a_j ungleich null ist, lässt sich durch die anderen Elemente in $\mathbf{x}^{(n)}$ ausdrücken. Es soll im Folgenden nun nicht mehr den Status einer Zustandsgröße besitzen. Die zugehörige Spalte sowie die Spalte der zugehörigen abgeleiteten Größe \dot{x}_j sollte demnach innerhalb der Matrix nach links in die Gruppe des Spaltenvektors \mathbf{w} verschoben werden. Durch zeitliche Ableitung der letzten Zeile ergibt sich:

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \dot{\mathbf{x}} = 0. \quad (A.17)$$

Gleichung (A.17) wird als zusätzliche Zeile in die Matrix des bestehenden Gleichungssystems integriert.

Nach diesen Schritten wird der hochgestellte Zählindex in Klammern um eins erhöht und die Matrix des Gleichungssystems wird wieder in Zeilenstufenform gebracht. Die quadratischen Untermatrizen $\mathbf{M}^{(n+1)}$ und $\mathbf{A}^{(n+1)}$ sind nun um eine Dimension kleiner. Die Vektoren $\dot{\mathbf{x}}^{(n+1)}$ und $\mathbf{x}^{(n+1)}$ besitzen ein Element weniger als zuvor.

⁹ Da hier $\dot{v}_{C_g} = 0$ gilt (Gleichstromquelle) folgt daraus $\dot{i}_L = 0$ und schließlich $v_g + v_C = 0$, weshalb in (A.15) der Induktivitätswert L keine Rolle mehr spielt.

A.3. Formulieren der Zustandsdifferentialgleichung

Das vorgestellte Verfahren stammt ursprünglich aus [64] und stellt aus mathematischer Sicht die Umformung eines differentiell-algebraischen Gleichungssystems (DAE) in ein gewöhnliches Differentialgleichungssystem (ODE) dar, das den besonderen Bedingungen eines Zustandsraumsystems genügt [59]. In Differential-algebraischen Gleichungssystemen sind gewöhnliche Differentialgleichungen und algebraische Gleichungen miteinander gekoppelt, was bei der Analyse elektrischer Netzwerke typischerweise der Fall ist [21, 27].

B. Ergänzungen zu Kapitel 4

B.1. Effektivwertberechnung mit reduziertem Zustandsraumsystem

Die erweiterte Zustandsraumdarstellung aus Unterabschnitt (4.5.2) besitzt einige redundante Zustände, da alle Mischprodukte doppelt auftreten: x_1x_2 und x_2x_1 , x_1x_3 und x_3x_1 usw. Diese überflüssigen Zustände können jedoch mit Hilfe der Eliminationsmatrix \mathbf{L} mit

$$\tilde{\mathbf{x}}_{\text{red}} = \mathbf{L}\tilde{\mathbf{x}} \quad (\text{B.1})$$

bzw. der Duplikationsmatrix \mathbf{D} mit

$$\tilde{\mathbf{x}} = \mathbf{D}\tilde{\mathbf{x}}_{\text{red}} \quad (\text{B.2})$$

eliminiert werden [177]. Aus

$$\frac{d}{dt}\tilde{\mathbf{x}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} \quad (\text{B.3})$$

folgt mit (B.1) und (B.2)

$$\begin{aligned} \mathbf{D}\frac{d}{dt}\tilde{\mathbf{x}}_{\text{red}} &= \tilde{\mathbf{A}}\mathbf{D}\tilde{\mathbf{x}}_{\text{red}} \\ \underbrace{\mathbf{LD}}_{=\mathbf{1}}\frac{d}{dt}\tilde{\mathbf{x}}_{\text{red}} &= \mathbf{L}\tilde{\mathbf{A}}\mathbf{D}\tilde{\mathbf{x}}_{\text{red}} \\ \frac{d}{dt}\tilde{\mathbf{x}}_{\text{red}} &= \tilde{\mathbf{A}}_{\text{red}}\tilde{\mathbf{x}}_{\text{red}}. \end{aligned} \quad (\text{B.4})$$

$$(\text{B.5})$$

Mit den zusätzlichen Hilfszuständen zur Mittelwertbildung lautet die erweiterte Systemmatrix

$$\mathbf{A}_{\text{RMS,red}} = \begin{bmatrix} \tilde{\mathbf{A}}_{\text{red}} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}. \quad (\text{B.6})$$

Die Ausgangsmatrix ohne Redundanzen lässt sich über

$$\tilde{\mathbf{C}}_{\text{red}} = \tilde{\mathbf{C}}\mathbf{D} \quad (\text{B.7})$$

berechnen.

B. Ergänzungen zu Kapitel 4

Die Effektivwerte der Ausgangsgrößen über das Zeitintervall von t_0 bis $t_0 + T$ ergeben sich mit der reduzierten Zustandsraumdarstellung zu:

$$\mathbf{y}_{\text{RMS}} = \sqrt{\frac{1}{T} \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{C}}_{\text{red}} \end{bmatrix} e^{\mathbf{A}_{\text{RMS,red}} T} \begin{bmatrix} \tilde{\mathbf{x}}_{\text{red}}(t_0) \\ \mathbf{0} \end{bmatrix}}. \quad (\text{B.8})$$

Überspannt das zu betrachtende Zeitintervall von t_0 bis $t_0 + T$ mehrere Zeitintervalle mit unterschiedlichen Zustandsraumdarstellungen, so muss für jedes dieser Teilintervalle getrennt das Quadrat des Effektivwerts berechnet werden und die Teillösungen zum Quadrat des Gesamteffektivwerts mittels gewichteter Addition zusammengesetzt werden. Eine anschließende Berechnung der Quadratwurzel liefert schließlich den gesuchten Gesamteffektivwert.

Symbole und Abkürzungen

Matrizen

Symbol	Beschreibung
$\mathbf{0}$	Nullmatrix bzw. Nullvektor
$\mathbf{1}$	Einheitsmatrix
\mathbf{A}	Systemmatrix
\mathbf{A}_a	Inzidenzmatrix
\mathbf{A}_{rref}	Inzidenzmatrix in reduzierter Zeilenstufenform
$\tilde{\mathbf{A}}$	Um die Eingangsgrößenmodelle erweiterte Systemmatrix
\mathbf{B}	Fundamentale Maschenmatrix bzw. Eingangsmatrix bzw. Untermatrix der modifizierten Knotenpotentialdarstellung bzw. eine zur Systemmatrix ähnliche Matrix
\mathbf{B}_t	Untermatrix der fundamentalen Maschenmatrix
\mathbf{B}_{aux}	Untermatrix im Gesamtgleichungssystem
\mathbf{C}	Ausgangsmatrix bzw. Untermatrix der modifizierten Knotenpotentialdarstellung bzw. Diagonalmatrix mit Kapazitäten/Induktivitäten
\mathbf{C}_{pwl}	Ausgangsmatrix der Ströme/Spannungen nichtlinearer, als stückweise linear definierter Schaltungselemente
\mathbf{D}	Fundamentale Schnittmengenmatrix bzw. Durchgangsmatrix bzw. Untermatrix der modifizierten Knotenpotentialdarstellung bzw. Duplikationsmatrix
\mathbf{D}_1	Untermatrix der fundamentalen Trennmengenmatrix
\mathbf{E}	Untermatrix im Gesamtgleichungssystem bzw. Koeffizientenmatrix einer Tschebyscheff-Entwicklung
\mathbf{F}	Matrix eines linearen Gleichungssystems bzw. Matrix der Schalttransformationsgleichung
\mathbf{G}	Matrix der Schalttransformationsgleichung
\mathbf{J}	Jacobimatrix bzw. Matrix in Jordanscher Normalform
\mathbf{K}	Matrix mit Zweigparametern

Vektoren

Symbol	Beschreibung
L	Matrix des Ungleichungssystems der nichtlinearen, als stückweise linear definierten Schaltungselemente bzw. Eliminationsmatrix
M	Untermatrix im Gesamtgleichungssystem bzw. Hilfsmatrix in der Blockdiagonalisierung
Q	Untermatrix im Gesamtgleichungssystem bzw. Transformationsmatrix
S	Untermatrix im Gesamtgleichungssystem bzw. Transformationsmatrix einer Ähnlichkeitstransformation
T	Untermatrix im Gesamtgleichungssystem bzw. Matrix in Schurscher Normalform
Y	Knotenadmittanzmatrix

Vektoren

Symbol	Beschreibung
δ	Spaltenvektor mit Impulsfunktionen
b	Spaltenvektor mit Quellfunktionen
f	Spaltenvektor mit Quellen bzw. Spaltenvektor zeitabhängiger Funktionen
i_r	Spaltenvektor ausgesuchter Zweigströme in der MNA
i	Spaltenvektor der Zweigströme
j	Spaltenvektor mit Quellen
$\dot{\mathbf{q}}$	Spaltenvektor mit den komplementären Größen der Zustandsgrößen
$\tilde{\mathbf{q}}$	Spaltenvektor mit den komplementären Größen der abhängigen Zustandsgrößen
q	Spaltenvektor mit Quellgrößen bzw. Spaltenvektor mit Strömen bzw. Spannungen von Kapazitäten bzw. Induktivitäten
r	Spaltenvektor eines Ungleichungssystems
r_n	Nichtlinearer Restterm (exponentielle Integratoren)
u	Spaltenvektor mit Strömen unabh. Stromquellen und Spannungen unabh. Spannungsquellen
$\tilde{\mathbf{u}}$	Spaltenvektor mit Strömen unabh. Spannungsquellen und Spannungen unabh. Stromquellen
\mathbf{u}_{aux}	Spaltenvektor mit Strömen unabh. Hilfsstromquellen und Spannungen unabh. Hilfsspannungsquellen

Symbol	Beschreibung
$\tilde{\mathbf{u}}_{\text{aux}}$	Spaltenvektor mit Strömen unabh. Hilfsspannungsquellen und Spannungen unabh. Hilfsstromquellen
\mathbf{v}	Spaltenvektor der Zweigspannungen
\mathbf{v}_k	Spaltenvektor der Knotenpotentiale
\mathbf{w}	Spaltenvektor mit Strömen und Spannungen von Widerständen, gesteuerten Quellen und Übertragern
$\tilde{\mathbf{x}}$	Um die Eingangsgrößenmodelle erweiterter Vektor der Zustandsgrößen
\mathbf{x}	Spaltenvektor der Zustandsgrößen bzw. Spaltenvektor der Unbekannten eines linearen Gleichungssystems
$\tilde{\mathbf{x}}$	Spaltenvektor abhängiger Zustandsgrößen
\mathbf{y}	Spaltenvektor der Ausgangsvariablen

Skalare

Symbol	Beschreibung
C	Kapazität
I	Zeitunabhängiger elektrischer Strom
L	Induktivität
N	Windungszahl
P	Polynomanteil eines Tschebyscheff-Modells
R	Ohmscher Widerstand
T	Elementarer Zeitschritt bzw. Periodendauer
V	Zeitunabhängige elektrische Spannung
Z_0	Schwingkreiskennwert
c	Differentielle Kapazität bzw. eine Konstante, häufig verwendet in der Lösung von Differentialgleichungen
e	Verstärkung einer spannungsgesteuerten Spannungsquelle
f	Frequenz bzw. universeller Funktionsname
g	Verstärkung bzw. Transkonduktanz einer spannungsgesteuerten Stromquelle
h	Zeitschritt
i	Zeitabhängiger Strom bzw. universelle Zählvariable mit $i \in \mathbb{N}_0$
j	Universelle Zählvariable mit $j \in \mathbb{N}_0$
k	Anzahl der Knoten eines elektrischen Netzwerks
m	Universelle Zählvariable mit $m \in \mathbb{N}_0$

Einheiten

Symbol	Beschreibung
n	Universelle Zählvariable mit $n \in \mathbb{N}_0$
q	Elektrische Ladung
s	Universelle Variable
t	Zeitvariable
v	Zeitabhängige elektrische Spannung
x	Zustandsgröße bzw. universelles Funktionsargument
z	Anzahl der Zweige eines elektrischen Netzwerks
Δ	Intervallanteil eines Tschebyscheff-Modells
ϕ	Magnetischer Fluss
δ	Impulsfunktion (Delta-Distribution)
λ	Eigenwert bzw. verketteter magnetischer Fluss
ω	Kreisfrequenz
φ	Phasenverschiebung bzw. elektrisches Skalarpotential
σ	Sprungfunktion (Heaviside-Funktion)
τ	Zeitkonstante bzw. Integrationsvariable

Konstanten

Symbol	Wert	Beschreibung
e	2,718 281 8 ...	Eulersche Zahl
π	3,141 592 6 ...	Kreiszahl

Einheiten

Symbol	Einheit	Beschreibung
A	Ampere	Elektrische Stromstärke
C	Coulomb	Elektrische Ladung
°	Grad	Winkelmaß
F	Farad	Elektrische Kapazität

Symbol	Einheit	Beschreibung
H	Henry	Induktivität
Hz	Hertz	Frequenz
Ω	Ohm	Elektrischer Widerstand
ppm	Millionstel	Hilfsmaßeinheit
rad	Radian	Winkelmaß
s	Sekunden	Zeit
V	Volt	Elektrische Spannung

Abkürzungen

BDF	eine Gruppe numerischer Integrationsverfahren ("Backward Differentiation Formulas")
CCM	Kontinuierlicher Betrieb ("Continuous Conduction Mode")
D	Abkürzung für Diode
DAE	Differential-algebraisches Gleichungssystem ("Differential algebraic equation system")
DCM	Lückender Betrieb ("Discontinuous Conduction Mode")
DGL	Differentialgleichung bzw. Differentialgleichungssystem
LGS	Lineares Gleichungssystem
LTE	Lokaler Abbruchfehler ("Local Truncation Error")
MNA	Modifiziertes Knotenpotentialverfahren ("Modified Nodal Analysis")
ODE	Gewöhnliches Differentialgleichungssystem ("Ordinary differential equation system")
PWL	Stückweise linear ("piecewise linear")
SPICE	Simulation Program with Integrated Circuit Emphasis
SSA	Zustandsraumverfahren ("State-Space Analysis")
STA	Tableauverfahren ("Sparse Tableau Analysis")

Tabellenverzeichnis

4.1.	Ergebnisse aller Iterationen der Nullstellensuche für die exemplarische Funktion (4.93)	118
4.2.	Ergebnisse aller Iterationen der Nullstellensuche für die exemplarische Funktion (4.93)	129
5.1.	Gerundete Ergebnisse der Simulation in SPICE	141
A.1.	Syntax für Widerstand, Kapazität und Induktivität	162
A.2.	Syntax für Gleichspannungs- und Gleichstromquellen	162
A.3.	Syntax für Sinus-Quellen	163
A.4.	Syntax für Pulsquellen	164
A.5.	Syntax für ideale Übertrager	164
A.6.	Syntax für spannungsgesteuerte Quellen	165
A.7.	Syntax für stromgesteuerte Quellen	165
A.8.	Syntax für spannungsgesteuerte Schalter	165
A.9.	Syntax für stromgesteuerte Schalter	166
A.10.	Syntax für Schaltermodelle	166
A.11.	Syntax der Diode	166
A.12.	Syntax für transiente Analyse	167
A.13.	Syntax für das Abspeichern von Zeitverläufen	168
A.14.	Syntax für das Abspeichern von Kennwerten	168
A.15.	Syntax für Anfangswerte	169

Abbildungsverzeichnis

2.1.	Beispiel eines Netzlisteneintrags, Schaltbilds und Ausfüllmusters eines Widerstands in der Tableaudarstellung	10
2.2.	Beispiel eines Netzlisteneintrags, Schaltbilds und Ausfüllmusters eines Widerstands in der modifizierten Knotenpotentialdarstellung	12
2.3.	Einfaches Beispielnetzwerk zur Demonstration numerischer Integration	15
2.4.	Exakte Lösung (durchgezogen) und Lösung des expliziten Euler-Verfahrens (gestrichelt) für unterschiedliche Schrittweiten h mit $\lambda = -1 \text{ s}^{-1}$ und $v_{C,0} = 1 \text{ V}$, nach [89]	17
2.5.	Beispielnetzwerk mit $i_g(t) = \hat{i}_g(1 - e^{\lambda_g t})$, wobei $ \lambda_g $ deutlich kleiner als $ \lambda $ gewählt wird (vereinfacht aus [4] entnommen)	18
2.6.	Exakte (durchgezogen) und Näherungslösung (gestrichelt) des Netzwerks aus Abb. 2.5 mit $\lambda = -1 \text{ s}^{-1}$, $\lambda_g = -0,1 \text{ s}^{-1}$, $v_{C,0} = 1 \text{ V}$ und $R\hat{i}_g = 1 \text{ V}$, bei $t = 5 \text{ s}$ springt der Zeitschritt der Näherungslösung von $h = 0,25 \text{ s}$ auf $h_g = 2,5 \text{ s}$	18
2.7.	Exakte Lösung (durchgezogen) und Lösungen des expliziten (links, gestrichelt) und impliziten (rechts, gestrichelt) Euler-Verfahrens der DGL (2.7) für eine Schrittweite $h = 2,1 \text{ s}$ und $\lambda = -1 \text{ s}^{-1}$, $v_{C,0} = 1 \text{ V}$; die Pfeile geben das Richtungsfeld der DGL wieder	21
2.8.	Mit implizitem Eulerverfahren generiertes „companion model“ einer Kapazität	22
2.9.	Netzwerk aus Abb. 2.5 mit beliebiger Quellfunktion $i_g(t)$	23
2.10.	Stückweise konstante Näherung (durchgezogen mit Punkten) einer beispielhaften Spannungsquelle (gestrichelt)	24
2.11.	Eingangsgrößenmodellierung einer sinusförmigen Spannungsquelle	25
2.12.	Beispielnetzwerk mit nichtlinearem Schaltungselement	27
2.13.	v_D - i_D -Diagramm mit Kennlinien zur grafischen Bestimmung der Lösung mit $v_g = 1 \text{ V}$, $R = 1 \Omega$, $i_0 = 1 \text{ nA}$ und $v_T = 28 \text{ mV}$	28
2.14.	Lokale lineare Approximation der Diodenkennlinie mit Tangenten (dünne, durchgehende Linien) und Zwischenergebnissen (nummerierte Marker)	28
2.15.	Stückweise lineare Approximation der Diodenkennlinie mit drei Geradenstücken	29
2.16.	Mit dem Newton-Raphson-Verfahren generiertes „companion model“ einer Diode	30

Abbildungsverzeichnis

2.17.	Mit der lokalen linearen Approximation und dem impliziten Eulerverfahren generiertes „companion model“ einer nichtlinearen Kapazität . . .	32
2.18.	Stückweise lineare Approximation (gestrichelt) der Diodenkennlinie (durchgezogen, Diodenparameter aus Abb. 2.13) mit unterschiedlicher Anzahl an Segmenten	34
2.19.	Zwei Beispiele stückweiser linearisierter Kennlinien (durchgezogen) mit beispielhafter Quellenkennlinie (gepunktet) und Teillösungen (a,b) des Newton-Raphson-Verfahrens	35
2.20.	Netzwerk mit stromgesteuerter Spannungsquelle und sogenannten versteckten Bedingungen ($v_g = 1 \text{ V} \sin(\omega t)$, $\omega = 1 \frac{\text{rad}}{\text{s}}$, $C_1 = 1 \text{ F}$, $C_2 = \text{beliebig}$, $g = 1 \Omega$)	38
2.21.	Reihenschaltung aus Spannungsquelle mit $v_g \neq 0 \text{ V}$ und Ersatzschaltung einer Kapazität des „companion model“ des impliziten Eulerverfahrens mit inkonsistentem Anfangswert $v_{C,0} = 0 \text{ V}$ (links) sowie zugehöriger Strom- und Spannungsverlauf (rechts)	38
2.22.	Parallelschaltung zweier Kapazitäts-Ersatzschaltungen des impliziten Eulerverfahrens („companion model“) mit unterschiedlichen Anfangswerten $v_{1,0} \neq v_{2,0}$	39
2.23.	Kurzschließen zweier Kapazitäten unterschiedlichen Anfangswerts mit idealem Schalter zum Zeitpunkt $t = 0$ (links) sowie zugehörigen Spannungsverläufen (rechts)	40
2.24.	Spitzenwertdetektor an Spannungsquelle („id. D.“ steht für ideale Diode, $R_{\text{off}} = R_b$, $R_{\text{on}} = \frac{R_a R_b}{R_a + R_b}$)	43
2.25.	LTspice IV (Version 4.21i) Simulationsergebnis der Netzliste aus Abb. 2.24b	43
2.26.	Ausschnitt aus dem Simulationsergebnis in Abb. 2.25	44
2.27.	Transiente Schaltungsanalyse (ohne Ein- und Ausgabe und ohne Berücksichtigung eventueller Konvergenzprobleme) in SPICE	47
3.1.	Schaltplan des Buck-Boost-Wandlers mit Schalteransteuerung, bestehend aus Quelle V_p und Steuereingang S,2	55
3.2.	Schaltbild des Widerstands mit Zählpfeilen	62
3.3.	Schaltbild des idealen n -Wicklungsübertragers	64
3.4.	Schaltbild von Induktivität (links) und Kapazität (rechts) mit Zählpfeilen	64
3.5.	Schaltbild von Schalter (links) und Diode (rechts) mit Zählpfeilen . . .	66
3.6.	Spannungsgesteuerte Stromquelle (rechts) mit Hilfsstromquelle (links) .	68
3.7.	Stromgesteuerte Spannungsquelle (rechts) mit Hilfsspannungsquelle (links)	68
3.8.	Schaltbild von Spannungsquelle (links) und Stromquelle (rechts) mit Zählpfeilen	69
3.9.	Eingangsgößenmodellierung einer Gleichspannungsquelle, auf einen zusätzlichen Eingangszweig mit nullwertiger Stromquelle (vgl. Abschnitt 3.2.5) kann hier und in den folgenden Quellenmodellen verzichtet werden . . .	70

3.10.	Eingangsgrößenmodellierung einer Pulsspannungsquelle, die Verstärkung der spannungsgesteuerten Quelle beträgt eins	71
3.11.	Eingangsgrößenmodellierung einer Sinusspannungsquelle, die Verstärkung der spannungsgesteuerten Quelle beträgt eins	72
3.12.	Schaltplan des Buck-Boost-Wandlers mit Hilfsquelle I_y	76
4.1.	Kurzschließen zweier Kapazitäten unterschiedlichen Anfangswerts mit idealem Schalter zum Zeitpunkt $t = 0$ (links) sowie zugehörige Spannungsverläufe (rechts)	87
4.2.	Beispiele für stückweise linearisierte q - v -Verläufe	88
4.3.	Schaltung mit einer eindeutigen Zuordnung der unabhängigen und abhängigen Zustandsgröße	93
4.4.	Schaltung mit zwei abhängigen Zustandsgrößen sowie Dirac-Impulsen erster und zweiter Ordnung	95
4.5.	Prozess um aus der Matrix des Gesamtgleichungssystems nach (4.37) die Matrix des Gleichungssystems mit der reduzierten Anzahl an Zustandsvariablen (4.41) zu generieren	98
4.6.	Ist die Kapazität zu Beginn negativ geladen ($v_{C,0} < 0$) sperrt die Diode, ist sie positiv geladen ($v_{C,0} > 0$) tritt ein Dirac-Impuls auf. Ist sie vollständig entladen ($v_{C,0} = 0$), so sind beide Zustände der Diode (leitend/sperrend) erlaubt.	99
4.7.	Zeitverlauf einer Pulsspannungsquelle mit $1 \mu\text{s}$ Verzögerung und Begrenzung auf zwei Perioden	104
4.8.	Approximation der Funktion e^{-at} mit $a = 1 \text{ s}^{-1}$ über die ersten beiden Glieder ihrer Taylorreihe	107
4.9.	Approximation der Funktion e^{-at} mit $a = 1 \text{ s}^{-1}$ über die ersten beiden Glieder ihrer Tschebyscheffentwicklung	109
4.10.	Beispielfunktion (4.93) mit oberer und unterer Schranke (4.98a) bzw. (4.98b) 115	
4.11.	Obere und untere Schranke (4.99a) bzw. (4.99b) mit skaliertem Zeitachse $t_{2,a}$ sowie die ursprüngliche Funktion $f(t)$ auf der Zeitachse t	116
4.12.	Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $0 \leq t \leq 0,25 \text{ s}$. Das gefüllte Rechteck markierte den Bereich, in dem zuvor ein Nulldurchgang ausgeschlossen werden konnte.	117
4.13.	Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $0,5 \text{ s} \leq t \leq 1 \text{ s}$ 118	
4.14.	Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $0,75 \text{ s} \leq t \leq 1 \text{ s}$. Das gefüllte Rechteck markiert den Bereich, in dem zuvor ein Nulldurchgang ausgeschlossen werden konnte.	119
4.15.	Die Funktion $f(t)$ mit oberer und unterer Schranke im Intervall $\frac{13}{16} \leq t \leq \frac{15}{16}$. Die gefüllten Rechtecke markieren den Bereich, in dem zuvor ein Nulldurchgang ausgeschlossen werden konnte.	120

Abbildungsverzeichnis

4.16.	Lösung (4.106) eines exemplarischen, gedämpft oszillatorischen Differentialgleichungssystems (4.106) mit deutlich kleinerer Oszillationsperiodendauer $T_{\text{osc}} = 10^{-3}\text{s}$ als dargestellter Intervallweite	121
4.17.	Beispielfunktion (4.106) mit oberer und unterer Schranke (4.154) bzw. (4.155)	128
4.18.	Integrieren einer beliebigen Spannung v_x mittels spannungsgesteuerter Stromquelle und einer Hilfs-Zustandsgröße $v_{x,\text{int}}$	131
4.19.	Stark abstrahierter Ablauf der rechnergestützten, transienten Schaltungsanalyse	136
5.1.	Schaltung eines RC-Glieds (links) mit inkonsistenten Anfangswerten $v_1(0^-) = 0$ und $v_2(0^-) = 2\text{V}$ sowie zugehöriger Netzliste (rechts)	138
5.2.	Vergleich der mit SPICE und den mittels (5.2) rekursiv berechneten Spannungszeitverläufen der Simulation im Zustandsraum (SSA)	138
5.3.	Vergleich der relativen Fehler der SPICE-Simulation, der mit „Rekursiv“ bezeichneten Berechnung nach (5.2), der mit „Binär“ bezeichneten Berechnung nach Abschnitt 4.4.4 und dem minimal erzielbaren Fehler	140
5.4.	SPICE-Netzliste zur Schaltung in Abb. 5.5	141
5.5.	Schaltung (Parameter nach Abb. 5.4) mit versteckten Bedingungen (nach [111])	141
5.6.	Simulationsergebnisse zur Netzliste aus Abb. 5.4	142
5.7.	Erweiterte Schaltung mit versteckten Bedingungen nach [111], sämtliche Induktivitäten besitzen 1 H, die Kapazität besitzt 1 F, die Anfangswerte sind identisch mit der Netzliste aus Abb. 5.4	142
5.8.	Simulationsergebnisse zur Schaltung aus Abb. 5.7	143
5.9.	Schaltung mit zwei nichtlinearen Kapazitäten und inkonsistenten Anfangswerten $v_{1-} \neq v_{2-}$	143
5.10.	Beispiele für zwei stückweise linearisierte, nichtlineare Kapazitäten	144
5.11.	Schaltplan und Netzliste mit Hilfsbeschaltung zur Realisierung benutzerdefinierter Anfangswerte	145
5.12.	Simulationsergebnisse zur Netzliste aus Abb. 5.11b, die Simulation im Zustandsraum besitzt den Index „ss“, die Simulation in SPICE den Index „spice“	145
5.13.	Spitzenwertdetektor an Spannungsquelle („id. D.“ steht für ideale Diode, $R_{\text{off}} = R_b$, $R_{\text{on}} = \frac{R_a R_b}{R_a + R_b}$)	146
5.14.	Simulationsergebnisse der Netzliste aus Abb. 5.13b für SPICE und der Simulation im Zustandsraum mit zuverlässiger Detektion der Unstetigkeiten	146
5.15.	Beispiel eines temporär unterbestimmten Netzwerks: Ein idealer Brückengleichrichter ($\hat{v}_g = 1\text{V}$, $\omega = 2\pi\text{Hz}$, $C = 1\text{F}$, $R = 1\Omega$)	147
5.16.	Simulationsergebnisse zur Netzliste aus Abb. 5.15b, die Simulation in Zustandsraum (Index „ss“) verwendet ideale Dioden ($D(R_{\text{on}}=0\ R_{\text{off}}=\text{inf})$), die Simulation in SPICE (Index „spice“) das Diodenmodell aus Abb. 5.15b, Zeile 8	148
5.17.	Schaltung eines Spannungsverdopplers mit Ladungspumpe	149

5.18.	SPICE-Netzliste zur Schaltung aus Abb. 5.17	150
5.19.	Simulationsergebnisse zur Netzliste aus Abb. 5.18, die Simulation in Zustandsraum (Index „ss“) verwendet ideale Schalter ($\text{SW}(\text{Ron}=0 \text{ Roff}=\text{inf})$), die Simulation in SPICE (Index „spice“) das Schaltermodell aus Abb. 5.18, Zeile 10	151
5.20.	Simulationsergebnisse zur Netzliste aus Abb. 5.18, die Simulation in Zustandsraum (Index „ss“) verwendet ideale Schalter ($\text{SW}(\text{Ron}=0 \text{ Roff}=\text{inf})$), die Simulation in SPICE (Index „spice“) das Schaltermodell aus Abb. 5.18, Zeile 10	151
5.21.	Schaltplan eines DC-DC-Spannungswandlers mit Funktionsblöcken der Taktsignalerzeugung (PWM) und Spannungsregelung (Ctrl)	152
5.22.	Schaltplan des Funktionsblocks der Taktsignalerzeugung mit zusätzlichen Kontrollmechanismen zur Begrenzung des maximalen Induktivitätsstroms und des maximalen Tastgrads (PWM). Die Funktionsweise kann [50] entnommen werden.	153
5.23.	Schaltplan des Funktionsblocks der Spannungsregelung (Ctrl) mit Sanftanlauf (R_4, C_3)	154
5.24.	SPICE-Netzliste zur Schaltung aus Abb. 5.21 bis Abb. 5.23	155
5.25.	Simulationsergebnisse zur Netzliste aus Abb. 5.24, die Simulation in Zustandsraum (Index „ss“) verwendet ausschließlich ideale Schalter und Dioden ($\text{Ron}=0, \text{Roff}=\text{inf}$), die Simulation in SPICE (Index „spice“) die Modelle aus Abb. 5.24, Zeile 31-34	156
A.1.	Brückengleichrichter mit Kondensator und ohmscher Last	171
A.2.	Beispielschaltung mit einer statt zwei Zustandsgrößen	174

Literatur

- [1] M. Wackernagel und W. Rees, *Our ecological footprint: reducing human impact on the earth*. New Society Publishers, 1998, Band 9.
- [2] M. Lenzen, C. B. Hansson, und S. Bond, „On the bioproductivity and land-disturbance metrics of the Ecological Footprint,” S. 6–10, 2007.
- [3] F. Blaabjerg, K. Ma, und Y. Yang, „Power electronics - The key technology for Renewable Energy Systems,” in *Proc. Ninth Int. Conf. Ecological Vehicles and Renewable Energies (EVER)*, März 2014, S. 1–11.
- [4] L. O. Chua und P.-M. Lin, *Computer-Aided Analysis of Electronic Circuits*. Englewood Cliffs: Prentice Hall, 1975.
- [5] V. J. Thottuvelil, „Challenges in computer-based analysis/simulation and design of power electronic circuits,” in *Proceedings of the IEEE Workshop on Computers in Power Electronics*, Aug. 1988, S. 1–7.
- [6] T. G. Wilson, „Life after the schematic: the impact of circuit operation on the physical realization of electronic power supplies,” *Proceedings of the IEEE*, Band 76, Nr. 4, S. 325–334, Apr. 1988.
- [7] K. S. Kundert, *The Designer’s Guide to Spice and Spectre*. Springer, Jan. 1995.
- [8] P. Li, L. M. Silveira, und P. Feldmann, *Simulation and verification of electronic and biological systems*. Springer Science & Business Media, 2011.
- [9] R. Erickson und D. Maksimovic, *Fundamentals of Power Electronics*, 2. Aufl. New York: Springer Science+Business Media, 2001.
- [10] V. J. Thottuvelil, D. Chin, und G. C. Verghese, „Hierarchical approaches to modeling high-power-factor AC-DC converters,” *IEEE Transactions on Power Electronics*, Band 6, Nr. 2, S. 179–187, Apr. 1991.
- [11] J. Sun und H. Grotstollen, „Symbolic analysis methods for averaged modeling of switching power converters,” *IEEE Transactions on Power Electronics*, Band 12, Nr. 3, S. 537–546, Mai 1997.
- [12] D. Kübrich, „Beitrag zur Optimierung einphasiger Leistungsfaktorkorrekturschaltungen,” Diss., Universität Erlangen-Nürnberg, 2011.

- [13] N. Mohan, W. P. Robbins, T. M. Undeland, R. Nilssen, und O. Mo, „Simulation of power electronic and motion control systems-an overview,” *Proceedings of the IEEE*, Band 82, Nr. 8, S. 1287–1302, Aug. 1994.
- [14] D. Maksimovic, A. M. Stankovic, V. J. Thottuvelil, und G. C. Verghese, „Modeling and simulation of power electronic converters,” *Proceedings of the IEEE*, Band 89, Nr. 6, S. 898–912, Juni 2001.
- [15] V. J. Thottuvelil, F. S. Tsai, und D. Moore, „Application of switched-circuit simulators in power electronics design,” in *Proceedings of the Applied Power Electronics Conference (APEC)*, März 1993, S. 484–490.
- [16] M.-L. Liou, „Exact Analysis of Linear Circuits Containing Periodically Operated Switches with Applications,” *IEEE Transactions on Circuit Theory*, Band 19, Nr. 2, S. 146–154, März 1972.
- [17] R. C. Wong, H. A. Owen, und T. G. Wilson, „A fast algorithm for the time-domain simulation of switched-mode piecewise-linear systems,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1984, S. 281–296.
- [18] R. J. Dirkman, „The simulation of general circuits containing ideal switches,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1987, S. 185–194.
- [19] M. Albach, *Grundlagen der Elektrotechnik 1*, 3. Aufl. München: Pearson Studium, 2011.
- [20] G. Kron, *Tensor Analysis of Networks*. New York: Wiley, 1939.
- [21] M. Günther, U. Feldmann, und J. ter Maten, „Modelling and Discretization of Circuit Problems,” in *Numerical Methods in Electromagnetics*, Serie Handbook of Numerical Analysis. Elsevier, 2005, Band 13, S. 523 – 659.
- [22] G. Hachtel, R. Brayton, und F. Gustavson, „The Sparse Tableau Approach to Network Analysis and Design,” *IEEE Transactions on Circuit Theory*, Band 18, Nr. 1, S. 101–113, Jan. 1971.
- [23] W. Weeks, A. Jimenez, G. Mahoney, D. Mehta, H. Qassemzadeh, und T. Scott, „Algorithms for ASTAP - A network-analysis program,” *IEEE Transactions on Circuit Theory*, Band 20, Nr. 6, S. 628–634, Nov. 1973.
- [24] J. Demel und S. Selberherr, „Application of the complete tableau approach in JANAP,” *Electrosoft*, Band 2, Nr. 6, S. 243–260, 1991.
- [25] F. Yuan und A. Opal, „Computer methods for switched circuits,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 50, Nr. 8, S. 1013–1024, Aug. 2003.
- [26] R. Unbehauen, *Grundlagen der Elektrotechnik 1*, 5. Aufl. Berlin: Springer-Verlag GmbH, 1999.

- [27] R. Unbehauen, *Grundlagen der Elektrotechnik 2*, 5. Aufl. Berlin: Springer-Verlag GmbH, 2000.
- [28] C.-W. Ho, A. Ruehli, und P. Brennan, „The modified nodal approach to network analysis,” *IEEE Transactions on Circuits and Systems*, Band 22, Nr. 6, S. 504–509, Juni 1975.
- [29] D. Pederson, „A historical review of circuit simulation,” *IEEE Transactions on Circuits and Systems*, Band 31, Nr. 1, S. 103–111, Jan. 1984.
- [30] L. W. Nagel und R. Rohrer, „Computer Analysis of Nonlinear Circuits, Excluding Radiation (CANCER),” *Journal of Solid-State Circuits*, Band SC-6, Nr. 4, S. 166–182, Aug. 1971.
- [31] L. W. Nagel und D. Pederson, „SPICE (Simulation Program with Integrated Circuit Emphasis),” EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M382, Apr. 1973.
- [32] L. W. Nagel, „SPICE2: A Computer Program to Simulate Semiconductor Circuits,” Diss., EECS Department, University of California, Berkeley, 1975.
- [33] T. L. Quarles, „Analysis of Performance and Convergence Issues for Circuit Simulation,” Diss., EECS Department, University of California, Berkeley, 1989.
- [34] K. G. Nichols, T. J. Kazmierski, M. Zwolinski, und A. D. Brown, „Overview of SPICE-like circuit simulation algorithms,” *IEE Proceedings - Circuits, Devices and Systems*, Band 141, Nr. 4, S. 242–250, Aug. 1994.
- [35] C. Gammell und M. Engelhardt. (2014, Apr.) An Interview with Mike Engelhardt – SPICE Simulator Synteresis. Interview. [Online]. Verfügbar: <http://theamphour.com/196-an-interview-with-mike-engelhardt-spice-simulator-synteresis/>
- [36] *Saber User Guide*, Synopsys, Inc., Sep. 2004.
- [37] (2016, Okt.) PSIM Frequently Asked Questions. Powersim, Inc. [Online]. Verfügbar: <https://powersimtech.com/support/resources/faq/>
- [38] *Multisim - User Guide*, National Instruments, Austin, Texas, USA, Jan. 2007.
- [39] P. Pejovic und D. Maksimovic, „A method for fast time-domain simulation of networks with switches,” *IEEE Transactions on Power Electronics*, Band 9, Nr. 4, S. 449–456, Juli 1994.
- [40] P. Pejovic und D. Maksimovic, „A new algorithm for simulation of power electronic systems using piecewise-linear device models,” *IEEE Transactions on Power Electronics*, Band 10, Nr. 3, S. 340–348, Mai 1995.
- [41] P. Pejovic und D. Maksimovic, „PETS-a simulation tool for power electronics,” in *Proceedings of the IEEE Workshop on Computers in Power Electronics*, Aug. 1996, S. 1–8.

- [42] P. J. van Duijsen, „Multilevel modeling and simulation of power electronic systems,” in *Proceedings of the European Conference on Power Electronics and Applications*, Band 4, Sep. 1993, S. 347–352.
- [43] A. Davis, *Gnucap users manual*, Sep. 2006.
- [44] *Simulation System Simplorer 6.0 User Manual*, 5. Aufl., Ansoft Corporation, Pittsburgh, Okt. 2002.
- [45] R. Nilssen und O. Mo, „KREAN, a new simulation program for power electronic circuits,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, 1990, S. 506–511.
- [46] J. Mahseredjian und F. Alvarado, „Creating an Electromagnetic Transients Program in MATLAB: MatEMTP,” *IEEE Transactions on Power Delivery*, Band 12, Nr. 1, S. 380–388, Jan. 1997.
- [47] J. Mahseredjian, S. Denetière, L. Dubé, B. Khodabakhchian, und L. Gérin-Lajoie, „On a new approach for the simulation of transients in power systems,” *Electric Power Systems Research*, Band 77, Nr. 11, S. 1514 – 1520, 2007.
- [48] *EMTDC - User's Guide*, 4. Aufl., Manitoba HVDC Research Centre, Winnipeg, Canada, 2010.
- [49] S. Jahn, M. Margraf, V. Habchi, und R. Jacob, *QUCS - Technical Papers*, 2007.
- [50] D. Bedrosian und J. Vlach, „Time-domain analysis of networks with internally controlled switches,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 39, Nr. 3, S. 199–212, März 1992.
- [51] J. Valsa und J. Vlach, „SWANN-a program for analysis of switched analog nonlinear networks,” in *Proceedings of the IEEE International Symposium on Circuits & Systems (ISCAS)*, Band 3, Apr. 1995, S. 1752–1755 vol.3.
- [52] H. S. H. Chung und A. Ioinovici, „Fast computer-aided simulation of switching power regulators based on progressive analysis of the switches' state,” *IEEE Transactions on Power Electronics*, Band 9, Nr. 2, S. 206–212, März 1994.
- [53] B. K. H. Wong und H. S. Chung, „An efficient technique for the time-domain simulation of power electronic circuits,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 45, Nr. 4, S. 364–376, Apr. 1998.
- [54] A. B. Yildiz, „A MNA-based unified ideal switch model for analysis of switching circuits,” *Journal of Circuits, Systems, and Computers*, Band 22, Nr. 6, S. 1350046 (12 Seiten), 2013.
- [55] J. D. Lavers, H. Jin, und R. W. Y. Cheung, „Analysis of power electronic circuits with feedback control: a general approach,” *IEE Proceedings B - Electric Power Applications*, Band 137, Nr. 4, S. 213–222, Juli 1990.
- [56] F. H. Branin, G. R. Hogsett, R. L. Lunde, und L. E. Kugel, „ECAP II-a new

- electronic circuit analysis program," *IEEE Journal of Solid-State Circuits*, Band 6, Nr. 4, S. 146–166, Aug. 1971.
- [57] J. Vlach und K. Singhal, *Computer Methods for Circuit Analysis and Design*, Serie Van Nostrand Reinhold Electrical/Computer Science and Engineering Series. Springer, 1983.
- [58] U. M. Ascher und L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.
- [59] E. S. Kuh und R. A. Rohrer, „The state-variable approach to network analysis," *Proceedings of the IEEE*, Band 53, Nr. 7, S. 672–686, Juli 1965.
- [60] A. M. Sommariva, „State-space equations of regular and strictly topologically degenerate linear lumped time-invariant networks: the multiport method," *International Journal of Circuit Theory and Applications*, Band 29, Nr. 5, S. 435–453, 2001.
- [61] P. Bryant, „The Explicit Form of Bashkow's A Matrix," *IRE Transactions on Circuit Theory*, Band 9, Nr. 3, S. 303–306, Sep. 1962.
- [62] A. Dervisoglu, „Bashkow's 'A Matrix for Active RLC Networks'," *IEEE Transactions on Circuit Theory*, Band 11, Nr. 3, S. 404–406, Sep. 1964.
- [63] T. Bashkow, „The A Matrix, New Network Description," *IRE Transactions on Circuit Theory*, Band 4, Nr. 3, S. 117–119, Sep. 1957.
- [64] A. Dervisoglu, „State Models of active RLC networks," Coordinate Science Laboratory University of Illinois, Urbana, Illinois, Tech. Rep. Report R-237, Dez. 1964.
- [65] C. J. Hsiao, R. B. Ridley, H. Naitoh, und F. C. Lee, „Circuit-oriented discrete-time modeling and simulation for switching converters¹," in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1987, S. 167–176.
- [66] Y.-S. Huang und S.-P. Chan, „A New State-variable Formulation Of Network Equations For Linear Networks," in *Proceedings of the Asilomar Conference on Circuits, Systems and Computers*, Nov. 1977, S. 396–400.
- [67] S. Natarajan, „A systematic method for obtaining state equations using MNA," *IEE Proceedings G (Circuits, Devices and Systems)*, Band 138, Nr. 3, S. 341–346, Juni 1991.
- [68] Y. Kang, „Systematic method for obtaining state-space representation of nonlinear dynamic circuits using MNA," *Electronics Letters*, Band 28, Nr. 21, S. 2028–2030, Okt. 1992.
- [69] Q. Chen, S. H. Weng, und C. K. Cheng, „A Practical Regularization Technique for Modified Nodal Analysis in Large-Scale Time-Domain Circuit Simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Band 31, Nr. 7, S. 1031–1040, Juli 2012.

- [70] P. Lin, „Formulation of hybrid matrices for linear multiports containing controlled sources,” *IEEE Transactions on Circuits and Systems*, Band 21, Nr. 2, S. 169–175, März 1974.
- [71] A. Massarini und U. Reggiani, „An efficient algorithm for the formulation of state equations and output equations for networks with ideal switches,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Band 3, Mai 2001, S. 521–524 vol. 2.
- [72] J. H. Allmeling und W. P. Hammer, „PLECS-piece-wise linear electrical circuit simulation for Simulink,” in *Proceedings of the IEEE International Conference on Power Electronics and Drive Systems (PEDS)*, Band 1, 1999, S. 355–360.
- [73] W. Hammer, „Dynamic Modeling of Line and Capacitor Commutated Converters for HVDC Power Transmission,” Diss., Eidgenössische Technische Hochschule (ETH), Zürich, 2003.
- [74] E. Purslow, „Solvability and analysis of linear active networks by use of the state equations,” *IEEE Transactions on Circuit Theory*, Band 17, Nr. 4, S. 469–475, Nov. 1970.
- [75] D. Li, „Time-domain Computer Analysis Methods for Power Electronic Circuits,” Diss., Portland State University, Portland, 2001.
- [76] R. Tymerski, „A fast time domain simulator for power electronic systems,” in *Proceedings of the Applied Power Electronics Conference (APEC)*, März 1993, S. 477–483.
- [77] *SimPowerSystems User’s Guide*, 3. Aufl., The MathWorks, Inc., 2013.
- [78] R. C. Wong, „Control of energy-storage dc-to-dc converters: The large-signal time-domain approach,” Diss., Duke University, Durham, 1984.
- [79] F. C. Y. Lee und Y. Yu, „Computer-Aided Analysis and Simulation of Switched DC-DC Converters,” *IEEE Transactions on Industry Applications*, Band IA-15, Nr. 5, S. 511–520, Sep. 1979.
- [80] F. C. Y. Lee, R. P. Iwens, Y. Yu, und J. E. Triner, „Generalized Computer-Aided Discrete Time-Domain Modeling and Analysis of DC-DC Converters,” *IEEE Transactions on Industrial Electronics and Control Instrumentation*, Band IECI-26, Nr. 2, S. 58–69, Mai 1979.
- [81] S.-H. Weng, „Circuit Simulation via Matrix Exponential Method,” Diss., University of California, San Diego, 2013.
- [82] M. Köksal und Y. Tokad, „State-space formulation of linear circuits containing periodically operated switches,” *International Journal of Circuit Theory and Applications*, Band 5, Nr. 2, S. 155–170, 1977.
- [83] J. M. Burdio und A. Martinez, „A unified discrete-time state-space model for switching converters,” *IEEE Transactions on Power Electronics*, Band 10, Nr. 6, S.

- 694–707, Nov. 1995.
- [84] S. M. Albeaik, „State space modeling, analysis, and simulation of ideal switched RLCM networks,” Masterarbeit, The Pennsylvania State University, State College, Pennsylvania, 2014.
- [85] J. C. Contreras-Sampayo, J. Usaola-Garcia, und A. R. Wood, „Steady-state algorithm for switching power electronic devices,” *IEE Proceedings-Electric Power Applications*, Band 148, Nr. 3, S. 245–250, Mai 2001.
- [86] A. R. de Jesus Silva und D. C. Martins, „Numerical simulation of power static converters: diagonalization method of dynamical matrix,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Band 2, Juni 1996, S. 1655–1660 vol.2.
- [87] F. Bordry und D. Martins, „Simulation of power electronic systems - computational method using variable step,” in *Proceedings of the European Conference on Power Electronics and Applications (EPE)*, 1985.
- [88] N. Femia und M. Vitelli, „Time-domain analysis of switching converters based on a discrete-time transition model of the spectral coefficients of state variables,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 50, Nr. 11, S. 1447–1460, Nov. 2003.
- [89] K. Janschek, *Systementwurf mechatronischer Systeme: Methoden – Modelle – Konzepte*. Springer Berlin Heidelberg, 2009.
- [90] F. H. Branin, „Computer methods of network analysis,” *Proceedings of the IEEE*, Band 55, Nr. 11, S. 1787–1801, Nov. 1967.
- [91] C. Moler und C. V. Loan, „Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later,” *SIAM Review*, Band 45, Nr. 1, S. 3–49, 2003.
- [92] J. Sun und H. Grotstollen, „Fast time-domain simulation by waveform relaxation methods,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 44, Nr. 8, S. 660–666, Aug. 1997.
- [93] D. Skowronn, D. Li, und R. Tymerski, „Simulation of networks with ideal switches,” *International Journal of Electronics*, Band 77, Nr. 5, S. 715–730, 1994.
- [94] A. Opal, „The transition matrix for linear circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Band 16, Nr. 5, S. 427–436, Mai 1997.
- [95] *PLECS - User Manual Version 4.1*, Plexim GmbH, Zurich, 2018.
- [96] A. M. Luciano und A. G. M. Strollo, „A fast time-domain algorithm for the simulation of switching power converters,” *IEEE Transactions on Power Electronics*, Band 5, Nr. 3, S. 363–370, Juli 1990.
- [97] G. Ludyk, *CAE von Dynamischen Systemen - Analyse, Simulation, Entwurf von*

Regelungssystemen. Berlin Heidelberg New York: Springer-Verlag, 1990.

- [98] J. A. Kojori und J. D. Lavers, „An improved algorithm for fast discrete time domain modelling of power electronic circuits,” in *Proceedings of the IEEE Annual Meeting of the Industry Applications Society*, Okt. 1988, S. 1032–1039 vol.1.
- [99] A. Opal, „Sampled data simulation of linear and nonlinear circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Band 15, Nr. 3, S. 295–307, März 1996.
- [100] D. M. W. Leenaerts und W. M. G. Van Bokhoven, *Piecewise Linear Modeling and Analysis.* Boston, MA: Springer US, 1998.
- [101] F. N. Najm, *Circuit Simulation.* Hoboken, New Jersey: John Wiley & Sons, Inc., 2010.
- [102] J. D. Lavers und Y. Kang, „A new approach for power electronic system modelling using MNA,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1993, S. 402–407.
- [103] K. Strehmel, H. Podhaisky, und R. Weiner, *Numerik gewöhnlicher Differentialgleichungen: Nichtsteife, steife und differential-algebraische Gleichungen.* Wiesbaden: Vieweg+Teubner Verlag, 2012.
- [104] M. Hochbruck und A. Ostermann, „Exponential integrators,” *Acta Numerica*, Band 19, S. 209–286, 2010.
- [105] A. H. Al-Mohy und N. J. Higham, „Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators,” *SIAM Journal on Scientific Computing*, Band 33, Nr. 2, S. 488–511, 2011.
- [106] P. Pejovic und D. Maksimovic, „An algorithm for solving piecewise-linear networks that include elements with discontinuous characteristics,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 43, Nr. 6, S. 453–460, Juni 1996.
- [107] N. Femia, G. Spagnuolo, und M. Vitelli, „Unified analysis of synchronous commutations in switching converters,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 49, Nr. 8, S. 1150–1166, Aug. 2002.
- [108] A. Massarini, U. Reggiani, und M. K. Kazimierczuk, „Analysis of networks with ideal switches by state equations,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 44, Nr. 8, S. 692–697, Aug. 1997.
- [109] M. Köksal und S. Herdem, „Ordinary Differential Equations with Strong Nonlinearities and Their Numerical Solutions with Abrupt Changes,” *Applied Mathematics and Computation*, Band 119, Nr. 2-3, S. 249–264, Apr. 2001.
- [110] Y.-W. Jan, „A switched-capacitor analysis metal-oxide-silicon circuit simulator,” Diss., Ohio University, Athens, 1999.

- [111] J. Vlach, J. M. Wojciechowski, und A. Opal, „Analysis of nonlinear networks with inconsistent initial conditions,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 42, Nr. 4, S. 195–200, Apr. 1995.
- [112] H. W. Dommel, *Electromagnetic Transients Program. Reference Manual (EMTP theory book)*. Portland: Bonneville Power Administration, 1986.
- [113] M. T. Engelhardt, „SPICE Differentiation,” *LT Journal of Analog Innovation*, Band 24, Nr. 4, S. 10–16, Jan. 2015.
- [114] J. R. Marti und J. Lin, „Suppression of numerical oscillations in the EMTP power systems,” *IEEE Transactions on Power Systems*, Band 4, Nr. 2, S. 739–747, Mai 1989.
- [115] J. Wojciechowski, Z. Michalski, und J. Vlach, „Numerical-symbolic large-signal simulation of switched circuits,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Band 4, Mai 1996, S. 620–623 vol.4.
- [116] M. Köksal und Y. Tokad, „Determination of initial conditions in active RLC networks,” *Circuit theory and applications*, Band 6, 1978.
- [117] E. Zeidler, *Springer-Taschenbuch der Mathematik*. Springer, Jan. 2013.
- [118] P. Kuffel, K. Kent, und G. Irwin, „The implementation and effectiveness of linear interpolation within digital simulation,” *International Journal of Electrical Power & Energy Systems*, Band 19, Nr. 4, S. 221 – 227, 1997.
- [119] P. Maffezzoni, L. Codecasa, und D. D’Amore, „Event-driven time-domain simulation of closed-loop switched circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Band 25, Nr. 11, S. 2413–2426, 2006.
- [120] N. Femia, G. Spagnuolo, und M. Vitelli, „Steady-state analysis of soft-switching converters,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 49, Nr. 7, S. 939–954, Juli 2002.
- [121] N. Femia, „A robust and fast convergent interval analysis method for the calculation of internally controlled switching instants,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 43, Nr. 3, S. 191–199, März 1996.
- [122] N. Femia und V. Tucci, „An optimized method for the time domain analysis of switched RLC networks,” in *Proceedings of the IEEE International Symposium on Circuits & Systems*, Band 1. IEEE, 1996, S. 569–572.
- [123] A. Cirillo und N. Femia, „Calculation of internally controlled switching instants in switched RLC circuits using interval analysis,” in *Proceedings of the IEEE International Symposium on Circuits & Systems*, Band 1. IEEE, 1996, S. 593–596.
- [124] G. T. Kostakis, S. N. Manias, und N. I. Margaris, „A generalized method for calculating the RMS values of switching power converters,” *IEEE Transactions on Power*

Electronics, Band 15, Nr. 4, S. 616–625, Juli 2000.

- [125] J. Vlach und A. Opal, „Modern CAD methods for analysis of switched networks,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 44, Nr. 8, S. 759–762, Aug. 1997.
- [126] C. Gear, „Simultaneous Numerical Solution of Differential-Algebraic Equations,” *IEEE Transactions on Circuit Theory*, Band 18, Nr. 1, S. 89–95, Jan. 1971.
- [127] C. Tischendorf, „Regularization of Electrical Circuits,” *IFAC-PapersOnLine*, Band 48, Nr. 1, S. 312 – 313, 2015.
- [128] G. Brocrad, *Simulation in LTSpice IV - Handbuch, Methoden und Anwendungen*. Künzelsau: Swiridoff, 2013.
- [129] „Elektrische Netze; Begriffe für die Topologie elektrischer Netze und Graphentheorie,” Apr. 1988.
- [130] R. W. Erickson und D. Maksimovic, „A multiple-winding magnetics model having directly measurable parameters,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Band 2, Mai 1998, S. 1472–1478.
- [131] A. Massarini und U. Reggiani, „Computer-aided time-domain large-signal analysis of networks with switches,” in *Proceedings of the IEEE International Symposium on Industrial Electronics*, Band 2, Juni 1996, S. 567–572.
- [132] N. J. Higham und E. Deadman, „A Catalogue of Software for Matrix Functions. Version 2.0,” Manchester Institute for Mathematical Sciences, The University of Manchester, Tech. Rep., März 2016.
- [133] N. J. Higham, *Functions of matrices: theory and computation*. SIAM, 2008.
- [134] N. Higham, *Accuracy and Stability of Numerical Algorithms*, 2. Aufl. Society for Industrial and Applied Mathematics, 2002.
- [135] P. I. Davies und N. J. Higham, „A Schur-Parlett algorithm for computing matrix functions,” *SIAM Journal on Matrix Analysis and Applications*, Band 25, Nr. 2, S. 464–485, 2003.
- [136] B. N. Parlett und K. C. Ng, „Development of an Accurate Algorithm for EXP (Bt),” University of California, Berkeley, Center for Pure and Applied Mathematics, Tech. Rep., 1985.
- [137] G. Moore, „Orthogonal polynomial expansions for the matrix exponential,” *Linear Algebra and its Applications*, Band 435, Nr. 3, S. 537–559, 2011.
- [138] C. A. Bavely und G. Stewart, „An algorithm for computing reducing subspaces by block diagonalization,” *SIAM Journal on Numerical Analysis*, Band 16, Nr. 2, S. 359–367, 1979.
- [139] C. Pottle, *System Analysis by Digital Computer*. New York-London-Sydney: John

- Wiley & Sons, Inc., 1966, Kap. State-Space Techniques for General Active Network Analysis, S. 59–98.
- [140] M. Tanaka, „Formulations for switching transformation matrices of large switched networks,” in *Proceedings of the IEEE International Symposium on Circuits & Systems*, Juni 1988, S. 1487–1490 vol.2.
- [141] A. Saichev und W. Woyczynski, *Distributions in the Physical and Engineering Sciences, Volume 2: Linear and Nonlinear Dynamics in Continuous Media*, Serie Applied and Numerical Harmonic Analysis. Springer New York, 2013.
- [142] Y. Demir und M. Köksal, „State space suitable modelling of power semiconductors with ideal switches, resistors, constant voltage and current sources,” *Electrical Engineering*, Band 86, Nr. 3, S. 135–145, 2004.
- [143] A. R. de Jesus Silva, „Programa SCVOLT - Simulação numérica de conversores estáticos, método do voltímetro, empregando a técnica do passo de cálculo variável,” Diss., Universidade Federal de Santa Catarina, Florianópolis, 1989.
- [144] O. Föllinger, F. Dörrscheidt, und M. Klittich, *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*, Serie Hüthig Studium. Hüthig, 2008.
- [145] F. Bordry, A. Beuret, und P. Proudlock, „Numeric simulation of thyristor power converters taking into consideration the firing-circuits and the closed-loop control,” in *Proceedings of the IMACS World Congress on Scientific Computation*, 1988.
- [146] S. S. Kelkar und F. C. Y. Lee, „A fast time domain digital simulation technique for power converters: Application to a buck converter with feedforward compensation,” *IEEE Transactions on Power Electronics*, Band PE-1, Nr. 1, S. 21–31, Jan. 1986.
- [147] J. P. Boyd, *Solving Transcendental Equations: The Chebyshev Polynomial Proxy and Other Numerical Rootfinders, Perturbation Series, and Oracles*. SIAM, 2014, Band 139.
- [148] P. Daponte, D. Grimaldi, A. Molinaro, und Y. D. Sergeyev, „An algorithm for finding the zero crossing of time signals with Lipschitzean derivatives,” *Measurement*, Band 16, Nr. 1, S. 37–49, 1995.
- [149] O. Caprani, L. Hvidegaard, M. Mortensen, und T. Schneider, „Robust and efficient ray intersection of implicit surfaces,” *Reliable Computing*, Band 6, Nr. 1, S. 9–21, 2000.
- [150] L. G. Casado, I. F. García, und Y. D. Sergeyev, „Interval branch and bound algorithm for finding the first-zero-crossing-point in one-dimensional functions,” *Reliable Computing*, Band 6, Nr. 2, S. 179–191, 2000.
- [151] L. G. Casado, I. García, und Y. D. Sergeyev, „Interval algorithms for finding the minimal root in a set of multiextremal one-dimensional nondifferentiable functions,” *SIAM Journal on Scientific Computing*, Band 24, Nr. 2, S. 359–376, 2002.

- [152] M. A. Wolfe, „On first zero crossing points,” *Applied Mathematics and Computation*, Band 150, Nr. 2, S. 467–479, 2004.
- [153] J. P. Boyd, „Computing real roots of a polynomial in Chebyshev series form through subdivision with linear testing and cubic solves,” *Applied Mathematics and Computation*, Band 174, Nr. 2, S. 1642–1658, 2006.
- [154] J. P. Boyd, „Computing real roots of a polynomial in Chebyshev series form through subdivision,” *Applied Numerical Mathematics*, Band 56, Nr. 8, S. 1077–1091, 2006.
- [155] H. A. Le Thi, M. Ouanes, und A. Zidna, „Computing real zeros of a polynomial by branch and bound and branch and reduce algorithms,” *Yugoslav Journal of Operations Research*, Band 24, Nr. 1, S. 53–69, 2014.
- [156] W. H. Press, S. A. Teukolsky, W. T. Vetterling, und B. P. Flannery, *Numerical recipes*, 3. Aufl. Cambridge University Press, Cambridge, 2007.
- [157] K. K. Tse, H. S. H. Hung, und S. Y. Hui, „Quadratic state-space modeling technique for analysis and simulation of power electronic converters,” *IEEE Transactions on Power Electronics*, Band 14, Nr. 6, S. 1086–1100, Nov. 1999.
- [158] K. K. Tse, H. S. Chung, und S. Y. R. Hui, „Stepwise quadratic state-space modeling technique for simulation of power electronics circuits,” *IEEE Transactions on Industrial Electronics*, Band 46, Nr. 1, S. 91–99, Feb. 1999.
- [159] A. Gil, J. Segura, und N. Temme, *Numerical Methods for Special Functions*. Society for Industrial and Applied Mathematics, 2007.
- [160] L. R. Petzold, „An Efficient Numerical Method for Highly Oscillatory Ordinary Differential Equations,” *SIAM Journal on Numerical Analysis*, Band 18, Nr. 3, S. 455–479, 1981.
- [161] N. Brisebarre und M. M. Joldes, „Chebyshev Interpolation Polynomial-based Tools for Rigorous Computing,” in *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2010, S. 147–154.
- [162] W. Tucker, *Validated numerics*. Princeton University Press, Princeton, NJ, 2011.
- [163] C. V. Loan, „Computing integrals involving the matrix exponential,” *IEEE Transactions on Automatic Control*, Band 23, Nr. 3, S. 395–404, Juni 1978.
- [164] H. R. Visser und P. P. J. van den Bosch, „Modelling of periodically switching networks,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1991, S. 67–73.
- [165] D. Bernstein, *Scalar, Vector, and Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, 2018.
- [166] C. G. Khatri und C. R. Rao, „Solutions to Some Functional Equations and Their Applications to Characterization of Probability Distributions,” *Sankhyā: The Indian Journal of Statistics*, Band 30, Nr. 2, S. 167–180, 1968.

- [167] F. Zach, *Leistungselektronik - Ein Handbuch Band 1 / Band 2*. Berlin Heidelberg New York: Springer-Verlag, 2015.
- [168] *+5V-Powered, Multichannel RS-232 Drivers/Receivers*, Maxim Integrated, Jan. 2015, Rev. 17.
- [169] T. J. Aprille und T. N. Trick, „Steady-state analysis of nonlinear circuits with periodic inputs,” *Proceedings of the IEEE*, Band 60, Nr. 1, S. 108–114, Jan. 1972.
- [170] E. A. El-Bidweihy und K. Al-Badwaihyy, „Steady-State Analysis of Static Power Converters,” *IEEE Transactions on Industry Applications*, Band IA-18, Nr. 4, S. 405–410, Juli 1982.
- [171] R. C. Wong, „Accelerated convergence to the steady-state solution of closed-loop regulated switching-mode systems as obtained through simulation,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1987, S. 682–692.
- [172] D. G. Bedrosian und J. Vlach, „An accelerated steady-state method for networks with internally controlled switches,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Band 39, Nr. 7, S. 520–530, Juli 1992.
- [173] D. Maksimovic, „Automated steady-state analysis of switching power converters using a general-purpose simulation tool,” in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC)*, Juni 1997.
- [174] B. K. H. Wong und H. S.-H. Chung, „Steady-state analysis of PWM DC/DC switching regulators using iterative cycle time-domain simulation,” *IEEE Transactions on Industrial Electronics*, Band 45, Nr. 3, S. 421–432, Juni 1998.
- [175] G. C. Verghese, M. E. Elbuluk, und J. G. Kassakian, „A General Approach to Sampled-Data Modeling for Power Electronic Circuits,” *IEEE Transactions on Power Electronics*, Band PE-1, Nr. 2, S. 76–89, Apr. 1986.
- [176] J. Schönberger, *Dynamics and Control of Switched Electronic Systems*. Springer-Verlag London, 2012, Kap. An Overview of Simulation Tools, S. 391–416.
- [177] J. R. Magnus und H. Neudecker, „The Elimination Matrix: Some Lemmas and Applications,” *SIAM Journal on Algebraic Discrete Methods*, Band 1, Nr. 4, S. 422–449, Dez. 1980.