

Semi-symbolische Modellierung und Simulation von Unsicherheiten in analogen Schaltungen mit Hilfe stückweise affiner Abbildungen

Von der Fakultät Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieur
(abgekürzt: Dr.-Ing.)
genehmigte Dissertation

von Dipl.-Ing. Marc Freisfeld
geboren am 19. Juni 1975 in Haselünne

2009

1. Referent: Prof. Dr.-Ing. Erich Barke
2. Referent: Prof. Dr. Christoph Grimm
Tag der Promotion: 16. Juni 2009

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Mikroelektronische Systeme der Leibniz Universität Hannover.

An erster Stelle danke ich Herrn Prof. Dr.-Ing. Erich Barke für seine Unterstützung und die Übernahme des Erstreferats. Herrn Prof. Dr. Christoph Grimm danke ich für die wertvollen Hinweise und die Übernahme des Korreferats.

Großer Dank gilt auch Herrn Dr.-Ing. Markus Olbrich für die vielen wertvollen Diskussionen und Anregungen sowie das Korrekturlesen der Arbeit.

Weiterhin danke ich allen, die mir bei der Durchführung dieser Arbeit behilflich waren, insbesondere allen Kollegen des Instituts für konstruktive Diskussionen und eine angenehme Arbeitsatmosphäre.

Meinen Freunden und der Familie, die mit mir mitgefiebert haben und mich in meinem Unterfangen bestärkt haben, danke ich ebenfalls.

Besonderer Dank gebührt meiner Lebensgefährtin Claudia. Sie hat mir mit unendlicher Geduld den Rücken frei gehalten.

Inhaltsverzeichnis

Kurzfassung	ix
Abstract	xi
Symbole und Abkürzungen	xiii
1 Einführung	1
1.1 Entwurfsmethoden	1
1.2 Verifikation und Verlässlichkeit	3
1.3 Ziele der Arbeit	4
2 Modellierung integrierter Bauelemente und Schaltungen	7
2.1 Bauelementmodelle	7
2.1.1 Erzeugung von Bauelementmodellen	8
2.1.2 Tabellenmodelle	9
2.2 Schaltungsmodelle	9
2.2.1 Systemgleichungen aus Netzlisten ermitteln	9
2.2.2 Makromodelle	10
2.2.3 PWL-Modellierung	12
2.2.4 Strukturerkennung	12
2.3 Blackbox-Ansatz	13
2.4 Unsicherheiten	14
3 Semi-symbolische Simulation	17
3.1 Simulation analoger Schaltungen	17
3.1.1 Analysearten	17
3.1.2 Zeitbereichsanalyse	18
3.1.3 Stochastische Analysen	20
3.2 Symbolische Analyse	20
3.3 Bereichsarithmetiken	22
3.3.1 Intervallarithmetik	23
3.3.2 Affine Arithmetik	23
3.3.3 Überabschätzung	25
3.4 Semi-symbolische Gleichungssysteme	27
3.5 Bekannte Ansätze zur semi-symbolischen Simulation	27

4	Stückweise lineare Modellierung mit Unsicherheiten	31
4.1	Darstellungsformen	31
4.2	PWA-Zustandsmodell	33
4.2.1	Mathematische Modellbeschreibung des PWA-Zustandsmodells	33
4.2.2	Berücksichtigung von Unsicherheiten	36
4.2.3	Approximationen an den Übergängen	38
4.3	Modellerzeugung aus Kennlinienform	40
4.4	Zusammenschaltung von PWA-Modellen in Netzlisten	42
4.5	Ableitung aus funktionaler Beschreibung	47
4.5.1	Rasterbasierter Ansatz mit konstantem Fehler	47
4.5.2	PWA-Zustandsmodell aus separiertem Term	49
4.5.3	Konstruktives Verfahren zur Separation affiner Abschnitte . .	49
4.6	Gewinnung eines Schaltungsmodells aus Simulationsdaten	53
4.6.1	Approximation nichtlinearer Verläufe	53
4.6.2	Hinge-Funktionen als Basisfunktionen	55
4.6.3	Approximation mit Hinging-Hyperebenen	55
4.6.4	Überführung vom Hingemodell in eine Zustandsmodell-Darstellung	57
5	Symbolische Umgebung zur Modellierung und Simulation	61
5.1	Sur-P.A.S.S.	61
5.2	Affine-Arithmetik-Modul	61
5.3	Modul zur Modellierung	65
5.3.1	Systemdefinition	65
5.3.2	Funktionen für Hinging-Hyperebenen	66
5.3.3	Funktionen zur Ableitung aus funktionalen Beschreibungen .	67
5.3.4	Verarbeitung von Netzlisten	67
5.4	Modul zur Simulation	68
5.4.1	Lösen des linearen Komplementaritätsproblems	68
5.4.2	Neubestimmung nach Zustandswechsel	71
5.4.3	Umsetzung und Steuerung der Simulation	73
6	Modellierungs- und Simulationsergebnisse	75
6.1	Beispiel mit zwei nicht idealen Dioden	75
6.2	Beispiel mit dynamischem Verhalten	76
6.3	PWA-Transistormodelle	80
6.3.1	Transistor aus Hinging-Hyperplanes	80
6.3.2	Beispiel MOSFET-Level-1-Modell	83
6.4	Sourceschaltung	86
6.5	Operationsverstärkerstufe aus Transistorpaaren	89
7	Zusammenfassung	95
8	Literaturverzeichnis	97

Abbildungsverzeichnis

1.1	Y-Diagramm zur Darstellung des Entwurfs- und Verifikationsraums	2
1.2	Systemsimulation durch Kopplung mehrerer Simulatoren	4
2.1	Modell eines Systems S mit Eingangs-, Ausgangs- und Parametergrößen	7
2.2	Struktur des Operationsverstärkermodells nach Boyle	11
2.3	Modell eines Operationsverstärkers mit stückweise linear modellierter Stromquelle	12
2.4	Einfacher CMOS-Stromspiegel	13
2.5	Faktoren für Unsicherheiten im modellierten Verhalten eines integrier- ten Bauelements	15
3.1	Ablauf der Zeitbereichsanalyse	19
3.2	Belasteter Tiefpass	21
3.3	Vergleich beider Bereichsarithmetiken	29
4.1	Unterschiedliche Formen eines affinen Teilbereichs für kanonische und nicht kanonische Darstellung	32
4.2	Ersatzschaltbild der Zustandsmodellardarstellung für den eindimensio- nalen Fall mit zwei affinen Segmenten (Gebieten)	34
4.3	Verlauf mit konstanter Aufweitung	36
4.4	Unsicherheit mit Abschnürung	37
4.5	Beide Abschnitte der Kennlinie haben Einfluss auf \hat{y}	38
4.6	Resultierende Ausgangskennlinie $\hat{y}(\hat{x})$	39
4.7	Nominalpunktverschiebung in der Transitionsfunktion	39
4.8	Projektion auf die zweidimensionale Ebene der Eingangsgrößen	40
4.9	Erzeugung eines PWA-Zustandsmodells aus Steigung, Stützstellen und y -Achsenabschnitt	41
4.10	Netzliste mit linearen und nichtlinearen Bauelementen	44
4.11	PWA-Kennlinien für eingesetzte nichtlineare Bauelemente	45
4.12	Resultierende Kennlinie $y_{neu,1}(x_{neu,1})$	46
4.13	PWL-Funktion mit absolutem Fehler	48
4.14	Transformation für nicht separierbare Multiplikation	50
4.15	Transformation für nicht separierbare Division	50
4.16	Transformation für nicht separierbare Potenzfunktion	51
4.17	Baumstruktur entsprechend der Auswertereihenfolge	51

4.18	Auswertereihenfolge nach der Separation	52
4.19	Hinge, Hinging Hyperebenen und Hingefunktion	56
4.20	Erster Schritt der Approximation mit dem HSA-Algorithmus	58
4.21	Approximation mit Hilfe zweier Hinge-Funktionen und eingezeichneten Grenzen	59
5.1	Struktur von <i>Sur-P.A.S.S.</i>	62
6.1	Semi-symbolische Simulation einer Beispielschaltung	75
6.2	Mit Unsicherheiten behaftete Eingangserregung	76
6.3	Ergebnis der semi-symbolischen Simulation für unterschiedliche Offsetspannungen U_{Offset}	77
6.4	Beispielschaltung mit Diode und Kapazität zur semi-symbolischen Simulation im Zeitbereich	77
6.5	Eingangserregung mit Unsicherheiten	78
6.6	Ergebnis der semi-symbolischen Transientensimulation für die Spannung $U_C(t)$	78
6.7	Ergebnis der semi-symbolischen Transientensimulation für den Strom $I_{in}(t)$	79
6.8	Schaltung zur Charakterisierung des DC-Verhaltens eines integrierten Feldeffekttransistors	80
6.9	Datenpunkte zur Charakterisierung des DC-Verhaltens eines NMOS-Transistors	81
6.10	Verhalten des NMOS-Transistors modelliert durch sechs Hinge-Funktionen	82
6.11	Approximationsfehler mit symmetrischer Einschlussfunktion	83
6.12	Ergebnis der Simulation des PWA-Zustandsmodells inklusive der durch \hat{f}_{extend} modellierten Unsicherheiten	84
6.13	Hierarchie der stückweise linearen Funktionen für den linearen Betriebsbereich	85
6.14	Durch Separation und anschließende Approximation gewonnene Kennlinie eines MOS-FET als PWA-Modell	87
6.15	Fehler durch Approximation	87
6.16	Sourceschaltung mit Transistor als PWA-Zustandsmodell	88
6.17	DC-Transfersimulation für eine Sourceschaltung	89
6.18	Erste Stufe eines Operationsverstärkers mit erkannten Transistorpaarstrukturen	90
6.19	Funktionsblöcke der Operationsverstärkerstufe	90
6.20	Schaltung zur Charakterisierung des Differenzenstufe	91
6.21	Parameterabhängige Ergebniskurven zur Charakterisierung	91
6.22	Ergebnis der semi-symbolischen Simulation für die Differenzenstufe	92
6.23	Ergebnis der semi-symbolischen Simulation für die gesamte Verstärkerstufe	92

Kurzfassung

Um die Komplexität heutiger Systeme zu beherrschen, nutzt man bei der simulationsgestützten Verifikation eine Kopplung von Verhaltensbeschreibungen auf unterschiedlichen Abstraktionsebenen. Dies bedingt eine Verwendung unterschiedlicher Simulatoren. Die semi-symbolische Simulation für integrierte analoge Schaltungen ist hierbei ein ergänzender Ansatz zu den klassisch genutzten Simulatoren. Durch die semi-symbolische Simulation ist es besonders effizient möglich, Unsicherheiten im Verhalten der Schaltung oder deren ansteuernden Signale in die Simulation mit einzu beziehen. Im Gegensatz zu den traditionellen statistischen Simulationsmethoden, die vielfache Simulationsdurchläufe erfordern, wird hier durch nur einen Simulationslauf ein Ergebnis ermittelt.

Die semi-symbolische Simulation kann für Systeme, die in algebraischer Beschreibung vorliegen, angewandt werden und ist Gegenstand aktueller Forschung. Der Begriff semi-symbolisch basiert auf der Verwendung mathematischer Ausdrücke in den Systemgleichungen, die in der Affine-Arithmetik-Form (kurz: AA-Form) vorliegen. Diese Ausdrücke beschreiben ein Intervall oder im höherdimensionalen Fall ein Polytop. Durch ein oder mehrere Symbole mit definiertem Wertebereich lassen sich in diesen Ausdrücken alle Werte innerhalb des Polytops erreichen. Die AA-Form wird durch die affine Arithmetik definiert, die ein mathematisches Regelwerk für Operationen mit solchen Ausdrücken bereitstellt.

Im Rahmen dieser Arbeit wurden Verfahren für eine semi-symbolische Modellierung und semi-symbolische Simulation analoger integrierter Schaltungen entwickelt und in einem Werkzeug mit der Bezeichnung *Sur-P.A.S.S.* implementiert. Die Modellierung des Schaltungsverhaltens beruht auf stückweise affinen Abbildungen, die in einem angepassten stückweise affinen Zustandsmodell umgesetzt sind. Weiterhin wurde in *Sur-P.A.S.S.* ein auf diese Modellbeschreibung angepasster Gleichungslöser implementiert. Die Lösungsalgorithmen dieses Gleichungslösers sind dabei speziell an die Anforderungen eines mit Unsicherheiten behafteten Modells angepasst.

Stichworte: Verhaltensmodellierung, Stückweise Affines Zustandsmodell, Semi-symbolische Simulation, Analoges Schaltungsdesign

Abstract

In order to control the complexity of today's systems, typically a coupling of behavioral descriptions on different abstraction levels is used for the simulation-based verification. This causes the use of different simulators. Semi-symbolic simulation for integrated analog circuits is an additional and alternative way instead of using classical simulators. With semi-symbolic simulation it is possible to include uncertainties in the circuit's behavioral description and their input signals in an efficient way. In contrast to the traditional statistic simulation methods, which require multiple simulation runs, only one single simulation run is necessary.

Semi-symbolic simulation can be used for systems which are given in an algebraic description, and is the subject of current research. Semi-symbolic simulation is based on mathematical expressions in the model's system equations that are defined as affine arithmetic form (short: AA-form). These expressions describe an interval or in the higher-dimension case a polytope. By one or more symbols which can get a value within a defined range, every value within the polytopes can be reached. The AA-form is defined by the affine arithmetic which gives a mathematical set of rules for operations with such expressions.

In this work procedures for semi-symbolic modeling and for semi-symbolic simulation of integrated circuits are developed. The results are implemented in a tool named *Sur-P.A.S.S.*. The modeling of the circuit behavior is based on piecewise affine mappings, which are converted into an adapted piecewise affine state model. Further a solver is implemented in *Sur-P.A.S.S.* which is specialized for these state models. The solution algorithms of the solver are fitted to the requirements of a model that is affected by uncertainties.

Keywords: Behaviour Modeling, Piecewise Affine State Model, Semi-symbolic Simulation, Analog Circuit Design

Symbole und Abkürzungen

\mathbb{R}	Menge der reellen Zahlen
\mathbb{R}^n	Menge der n -dimensionalen reellen (Spalten-) Vektoren
$\mathbb{R}^{n \times m}$	Menge der $n \times m$ -Matrizen
\vec{x}	(Spalten-) Vektor
\vec{x}^T	Transponierter Vektor (Zeilenvektor)
$\vec{0}$	Nullvektor
\mathbf{M}	Matrix
\mathbf{M}^T	Transponierte Matrix
\mathbf{M}^{-1}	Inverse einer Matrix
$\mathbf{I}_{n \times n}$	Einheitsmatrix mit der Dimension $n \times n$
$f(\cdot)$	Funktion
$\vec{f}(\cdot)$	Vektorwertige Funktion
$\vec{f}(\cdot) = \vec{0}$	Gleichungssystem (vektorwertige Gleichungen)
$[a, b]$	Intervall mit unterer Grenze a und oberer Grenze b
t	Zeit
$x(t)$	Zeitabhängige Variable
$\dot{x}(t) = \frac{dx(t)}{dt}$	Zeitliche Ableitung der Variablen $x(t)$
\hat{x}	Variable in affiner Form
ε_i	Abweichungssymbol mit dem Index i einer partiellen Abweichung
ξ	Gesamtzahl aller untereinander unkorrelierten partiellen Abweichungen
$\text{rad}(\hat{x})$	Radius der affinen Form \hat{x}
$x_0 = \text{nom}(\hat{x})$	Nominalwert der affinen Form \hat{x}
$\vec{\hat{x}}$	Vektor mit Einträgen in affiner Form
$\hat{\mathbf{M}}$	Matrix mit Einträgen in affiner Form
\vec{u}, \vec{j}	Zustandsvektoren des PWA-Zustandsmodells
DAE	Algebro-Differentialgleichungssystem
MNA	Modifizierte Knotenpotentialanalyse
PWA	Stückweise affin (engl.: piecewise affine)

1 Einführung

Automatisierungswerkzeuge für das analoge Schaltungsdesign gewinnen in der Industrie ein immer stärkeres Interesse. Während die Entwurfsprozesse bei digitalen integrierten Schaltungen bereits einen hohen Automatisierungsgrad erreicht haben, gibt es im Bereich des Analog- und Mixed-Signal-Entwurfes noch ein Vielzahl an Herausforderungen. Obwohl, gemessen an der Anzahl der Transistoren, analoge Schaltungen eine weitaus geringere Komplexität haben, erhöhen die größeren Freiheiten bei den Entwurfsparametern und deren Abhängigkeiten untereinander die Entwurfskomplexität.

In integrierten Schaltungen übernehmen analoge Blöcke einerseits einfache Schnittstellenfunktionen (z.B. im Bereich der Sensorik und Aktorik), andererseits sind sie Teil hochperformanter und komplexer Systeme (z.B. in PLL's). In beiden Fällen wird aus wirtschaftlichen Erwägungen die Integration der analogen Schaltungen in Mixed-Signal Systeme vorangetrieben. Die dabei verwendeten Technologien (meist CMOS- und BiCMOS-Technologien) besitzen vergleichsweise schlechte Eigenschaften für Analogschaltungen, da sie für die auf dem Chip flächenmäßig überwiegenden digitalen Schaltungen optimiert sind.

Ein weiterer Aspekt ist die Robustheit des Entwurfs gegenüber Fertigungsschwankungen und Betriebsbedingungen. Besonders in sicherheitskritischen Anwendungen, wie beispielsweise im Automotive-Bereich, ist man auf verlässliche Verifikationsverfahren angewiesen. Damit werden sowohl durch die Entwurfskomplexität, als auch durch ungünstige Entwurfsbedingungen große Anforderungen an heutige Entwurfs- und Verifikationswerkzeuge gestellt.

1.1 Entwurfsmethoden

Der Analogentwurf ist im Vergleich zum Digitalentwurf nicht standardisierbar. Begründet ist dies in der großen Zahl von Schaltungsklassen, die je nach Einsatzgebiet der Schaltung verschiedensten physikalischen Randbedingungen unterliegen. Weiterhin erfolgt im Analogentwurf in der Praxis die Spezifikation der Entwurfsaufgabe oft parallel zum Entwurf. Die Auswahl der genutzten EDA-Entwurfswerkzeuge und der

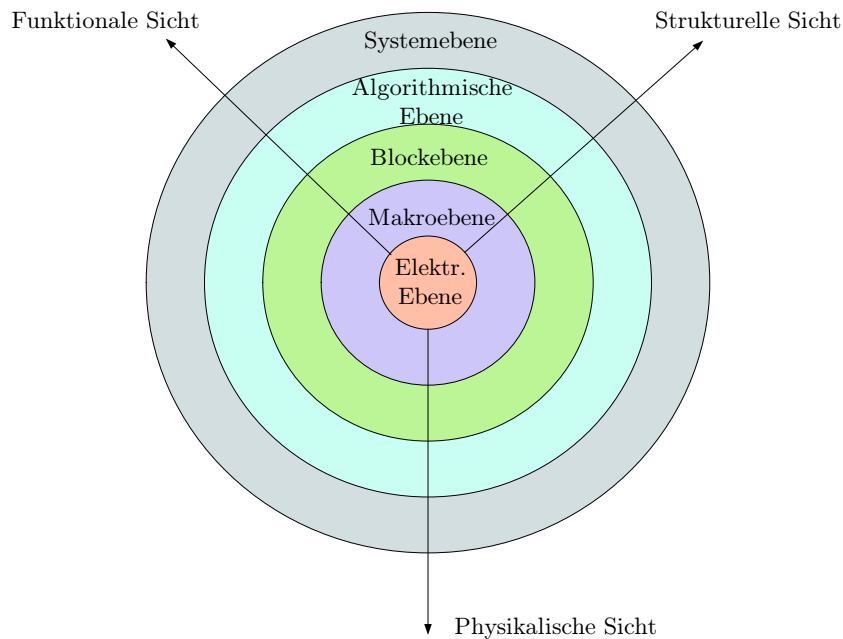


Abbildung 1.1: Y-Diagramm zur Darstellung des Entwurfs- und Verifikationsraums

damit verbundene Entwurfsablauf sind daher stark von der Entwurfsaufgabe abhängig.

Das von [Hedr04] vorgestellte Y-Diagramm, welches in Abbildung 1.1 dargestellt ist, gibt einen Überblick über den Entwurfs- und Verifikationsraum. Das Y-Diagramm besteht aus drei Achsen. Jede dieser Achsen repräsentiert einen speziellen Aspekt des Entwurfsgegenstands: die Entwurfssicht. Jede der Sichten beschreibt bestimmte Eigenschaften, die den Entwurfsgegenstand charakterisieren. In der *funktionalen Sicht* wird das Verhalten, genauer die Funktionsweise der Schaltung beschrieben. Diese besitzt meistens die Form mathematischer Gleichungs- und Ungleichungsbeziehungen. Die *strukturelle Sicht* bildet die Topologie der Schaltung ab, die sich aus ihren Komponenten und deren Verbindungen zusammensetzt. Die *physikalische Sicht* beschreibt das Layout der Schaltung in Form von Geometriedaten.

Bewegt man sich vom Mittelpunkt des Diagramms über eine der drei Achsen nach außen durchläuft man mehrere *Abstraktionsebenen*. Der Grad der Abstraktion mit der die Schaltung beschrieben wird, nimmt dabei nach außen hin immer weiter zu. Je nach Sicht und Abstraktionsebene existieren folglich unterschiedliche Repräsentationsformen einer Schaltung. Beim Entwurf und der Verifikation bewegt man sich zwischen den Sichten und über die Abstraktionsebenen hinweg.

Allgemein kann man zwischen drei Entwurfsstrategien unterscheiden [MaGi08]:

Flacher Entwurf Bei dieser Entwurfsmethode bewegt man sich ausschließlich auf der elektrischen Ebene. Dabei wird weder eine Abstraktion, noch eine Verfeine-

rung der Schaltungsbeschreibung genutzt. Der optimale Satz an Schaltungsparametern, wie beispielsweise Kanallängen und -breiten von Transistoren, wird durch systematisches Vorgehen mittels eines Designplanes [Heal93], symbolische Methoden [Hedr04] oder Optimierungsstrategien [BiHe04] gewonnen. Der Hauptnachteil bei diesem Vorgehen besteht in der eingeschränkten Schaltungskomplexität, die sich beherrschen lässt. Der Vorteil besteht jedoch darin, dass man alle nicht idealen Effekte beim Entwurf berücksichtigen kann und direkten Einblick auf deren Ursache hat. Der flache Entwurf wird hauptsächlich beim Entwickeln von Basisblöcken, wie beispielsweise VCO's oder Operationsverstärkern eingesetzt.

Bottom-Up-Strategie Diese Methode beginnt in der Regel auf der elektrischen Ebene, wobei die das Design bildenden Blöcke jeweils einzeln betrachtet und entwickelt werden. Das Gesamtsystem wird dann aus den Blöcken zusammengesetzt. Die Herausforderung liegt dabei darin, nach dem Zusammenführen ein innerhalb der Spezifikation liegendes Design zu erhalten. Hierzu wählt der Designer in der Regel Sicherheits-Margen für die Spezifikation der unteren Blöcke.

Top-Down-Strategie Meistens beginnend auf der Systemebene eines komplexen Design wird das System bei dieser Entwurfsmethode in kleinere Blöcke unterteilt [McGe03]. Hierzu werden aus dem komplexen Design Randbedingungen (constraints) für die Blöcke hergeleitet. Das Hauptproblem besteht hier darin, einen optimalen und lösbaren Satz an Randbedingungen zu bestimmen, ohne dabei das Wissen über den genauen Aufbau des Blockes zu besitzen. Stattdessen wird dieser durch Verhaltens- oder Performancemodelle repräsentiert.

1.2 Verifikation und Verlässlichkeit

Ziel der *Verifikation* von analogen Schaltungen und Systemen ist die Kontrolle auf Entwurfsvollständigkeit und -korrektheit, d.h die Überprüfung, ob alle Anforderungen durch die Schaltungsfunktionalität erfüllt werden. Zusätzlich existieren spezielle Ansätze für systematische Untersuchungen von Performanz, Robustheit oder Ausbeute (Yield), die ebenfalls zur Verifikation gerechnet werden.

Die Verifikation der Schaltungsfunktionalität erfolgt in der Praxis beinahe ausschließlich durch die Simulation des Schaltungsverhaltens und eine anschließende Bewertung durch den Entwickler. Um die Komplexität heutiger Systeme zu beherrschen, nutzt man bei der simulationsgestützten Verifikation eine Kopplung von Verhaltensbeschreibungen auf unterschiedlichen Abstraktionsebenen (Mixed-Level). Werden zusätzlich digitale Blöcke in die Betrachtung mit einbezogen, spricht man von einer Mixed-Signal-Simulation. Dies bedingt vielfach eine Kopplung unterschiedlicher Simulatoren auf Systemebene, wie Abbildung 1.2 verdeutlicht. Dabei sind jeweils eige-

ne Simulatoren für digitale, konservative und nicht-konservative Systeme üblich, so wie in neuester Entwicklung auch semi-symbolische Simulatoren.

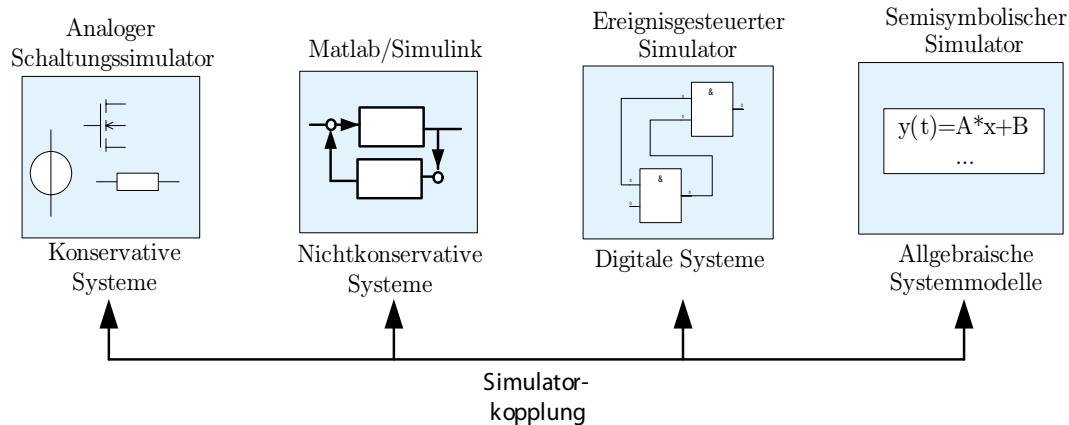


Abbildung 1.2: Systemsimulation durch Kopplung mehrerer Simulatoren

Die semi-symbolische Simulation stellt eine Ergänzung zu mehreren möglichen Simulationsarten dar. Sie kann für Systeme, die in algebraischer Beschreibung vorliegen, angewandt werden und ist aktueller Gegenstand der Forschung [GrOl08, GrGr06, GrHe04], sowie Bestandteil dieser Arbeit.

Als übergeordneter Simulator zur Kopplung und Steuerung der übrigen Simulatoren hat sich in letzter Zeit verstärkt System-C herausgebildet [VaGr03]. In diesem Kontext ergibt sich weiterhin die Perspektive, die Erweiterung SystemC-AMS als Standard für eine Anbindung der semi-symbolischen Simulation an die Systemsimulation zu etablieren.

Grundsätzlich sind alle Ergebnisse einer Simulation mit Unsicherheiten behaftet. Bei der Bewertung der Simulationsergebnisse ist dies durch den Entwickler zu berücksichtigen. Diese Unsicherheiten können unterschiedliche Ursachen haben und werden in Abschnitt 2.4 näher untersucht. Während der Verifikation ist es vielfach notwendig, trotz dieser Unsicherheiten *verlässliche* Aussagen über das Schaltungsverhalten zu erhalten. Ansatzpunkt dafür ist eine entsprechende Modellierung.

Definition 1.1 *Ein System oder Bauelement gilt als verlässlich modelliert, wenn das Ausgangsverhalten des Modells jegliches Ausgangsverhalten des Systems oder Bauelementes unter Berücksichtigung möglicher Unsicherheiten wiedergibt.*

1.3 Ziele der Arbeit

Der traditionelle Ansatz zur Berücksichtigung von Unsicherheiten baut auf statistischen Simulationen (z.B. der Monte-Carlo Simulation) auf, die vielfache Si-

ulationsdurchläufe erfordern. Dies bewirkt eine lange Gesamtsimulationszeit, die von der Anzahl der Simulationsläufe abhängt. Durch die semi-symbolische Simulation ist es im Vergleich dazu besonders effizient möglich, Unsicherheiten im Verhalten der Schaltung oder deren ansteuernden Signale mit einzubeziehen. Im Gegensatz zum traditionellen Ansatz wird hier durch nur einen einzelnen Simulationslauf ein Ergebnis ermittelt.

Wie verdeutlicht, ist die Modellierung sowohl ein wesentlicher Bestandteil des Entwurfs als auch Bestandteil der Verifikation. In der vorliegenden Arbeit wird auf eine verifikationsorientierte semi-symbolische Modellierung abgezielt. Dabei wird untersucht, wie diese zur Modellierung von Unsicherheiten verwendet werden kann, um Verhaltensmodelle zu erstellen, die

- abstrakt sind, das heißt nur einen Teil der tatsächlichen Eigenschaften einer Schaltung modellieren und gleichzeitig
- verlässlich sind, das heißt die Abweichungen zwischen modelliertem Verhalten und realem Verhalten durch Intervalle berücksichtigt werden.

Die Simulationszeit eines Modells hängt sehr stark vom Aufbau der Verhaltensmodelle ab. Allgemein gilt: Je genauer und detaillierter das Modell das reale Verhalten nachbildet, desto komplexer und somit auch langsamer lässt sich das Modell simulieren. Der oben skizzierte Ansatz unterscheidet sich insofern von dem traditionellen Ansatz, als die Modelle nur so komplex formuliert werden müssen, wie es die an Modellierung und Simulation gestellten Fragen erfordern und dennoch formal sichere und verlässliche Aussagen liefern.

Ungenauigkeiten, die sich durch Abstraktion in den Modellformulierungen ergeben, können durch Intervalle eingeschlossen werden. Dabei gilt es, ein gutes Verhältnis zwischen Konvergenz und Geschwindigkeit bei der Simulation auf der einen Seite, und der Genauigkeit des Modells auf der anderen Seite zu finden.

Diese Arbeit untersucht die Modellierung analoger Schaltungen durch stückweise affine Abbildungen¹. Hierbei sollen unterschiedliche Ansätze zur Modellbildung vorgestellt werden. Weiterhin wird das im Rahmen dieser Arbeit entwickelte Werkzeug mit der Bezeichnung *Sur-P.A.S.S.* vorgestellt. Bei *Sur-P.A.S.S.* handelt es sich um ein Paket zur Modellierung und Simulation von analogen Schaltungen. Weiterhin wird der in *Sur-P.A.S.S.* implementierte zugehörige Gleichungslöser vorgestellt. Die Lösungsalgorithmen dieses Gleichungslösers sind dabei speziell an die Anforderungen eines mit Unsicherheiten behafteten Modells angepasst.

¹Um Verwirrung zu vermeiden, sei angemerkt, dass für den Begriff *stückweise affin* in der Literatur häufig auch *stückweise linear* verwendet wird. Der mathematisch korrekte Begriff für die gemeinten Verläufe lautet jedoch *affin*.

Im folgenden Kapitel 2 werden zunächst die verschiedenen Aspekte der Modellbildung von analogen Schaltungen und Systemen behandelt und diesbezüglich wichtige Begrifflichkeiten geklärt. Kapitel 3 behandelt danach den Themenkomplex der semi-symbolischen Simulation, die ein zentraler Bestandteil dieser Arbeit ist. Zur Erläuterung wird hierbei zunächst auf allgemeine Simulationsalgorithmen eingegangen. Anschließend wird die Affine Arithmetik, die Basis der hier vorgestellten semi-symbolischen Simulation ist, erläutert.

Die beiden Kapitel 4 und 5 stellen die im Rahmen dieser Arbeit entwickelten und weiterentwickelten Methoden zur Modellierung und Simulation von Schaltungen, die mit Unsicherheiten behaftet sind, vor. Die unterschiedlichen Verfahren nutzen Informationen sowohl aus gegebenen Spezifikationen, als auch aus Schaltungsbeschreibungen, um den drei in Abschnitt 1.1 vorgestellten Entwurfsmethoden (Top-Down, Bottom-Up und flacher Entwurf) Rechnung zu tragen. Die Simulationsergebnisse von einigen Schaltungsbeispielen befinden sich in Kapitel 6. Alle Ergebnisse sind in dem im Rahmen dieser Arbeit entstandenen Gesamtpaket *Sur-P.A.S.S.* implementiert.

2 Modellierung integrierter Bauelemente und Schaltungen

Möchte man das Verhalten von integrierten analogen Schaltungen oder Systemen untersuchen, benötigt man ein mathematisches Modell, welches das reale Verhalten der Schaltung oder des Systems widerspiegelt. Die Modellfunktion kann dabei entweder mit symbolischen Verfahren [GiSa91] oder durch numerische Modellierung [FeDi91] generiert werden.

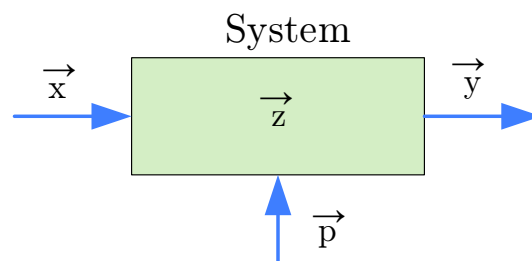


Abbildung 2.1: Modell eines Systems S mit Eingangs-, Ausgangs- und Parametergrößen

Im Folgenden soll der Begriff System gemäß Abbildung 2.1 verwendet werden. Demnach beschreibt das mathematische Modell des Systems S die Beziehung zwischen dem Eingangsvektor \vec{x} (Stimuli) und dem Ausgangsvektor \vec{y} . Der Vektor \vec{z} bezeichnet die inneren Variablen des Systems. Weiterhin wird das Verhalten des Systems durch den Parametervektor \vec{p} beeinflusst. Grundsätzlich besteht kein Unterschied zwischen den Eingangsvektoren \vec{x} und \vec{p} . Im Folgenden sollen jedoch alle zeitunabhängigen Eingangsgrößen als Parameter bezeichnet und somit \vec{p} zugeordnet werden, alle zeitabhängigen Eingangsgrößen dem Vektor \vec{x} .

2.1 Bauelementmodelle

Die Bauelemente sind die Grundbausteine, aus denen eine Schaltung besteht. Möchte man das Verhalten von integrierten Schaltungen auf elektrischer Ebene modellieren, muss man hauptsächlich das Verhalten von MOS-Transistoren und Bipolar-Transistoren, Widerständen bzw. Leitwerten, Kapazitäten, Induktivitäten sowie gesteuerten Strom- und Spannungsquellen betrachten.

Soll das Bauelementmodell auf die Systemdefinition gemäß Abbildung 2.1 übertragen werden, ist eine gerichtete Zuordnung der elektrischen Größen zu den Eingängen und Ausgängen des Systems erforderlich. In den meisten Fällen definiert man die Klemmenspannungen oder Potentiale der Bauelemente als Eingangsgrößen \vec{x} und die Klemmenströme als Ausgangsgröße \vec{y} . Als einfache Beispiele werden ein idealer Widerstand durch die Gleichung

$$i_R(t) = \frac{u_2(t) - u_1(t)}{R}$$

und ein Kondensator durch die Gleichung

$$i_C = \frac{d}{dt}(C \cdot u_1(t) - u_2(t))$$

beschrieben. $u_1(t)$ bzw. $u_2(t)$ sind die an den Knoten des Bauteiles anliegenden Potentiale, $i_R(t)$ der resultierende Strom. Der Parametervektor \vec{p} aus Abbildung 2.1 wird aus dem Widerstandswert R , bzw. der Kapazitätsgröße C gebildet.

Auch komplexere Bauelementmodelle wie beispielsweise für einen Transistor geben die Klemmenströme in Abhängigkeit der Klemmenpotentiale und Eingangsparameter wider. Die mathematische Formulierung der Modellgleichungen erlaubt dabei meistens eine explizite Bestimmung der Ausgangsströme bei gegebenen Eingangsgrößen.

Bei den heutigen Genauigkeitsanforderungen an die Modellierung für moderne Technologien unterliegen die gebräuchlichen Transistormodelle einer ständigen Weiterentwicklung und stellen die Modellentwickler vor immer größer werdende Herausforderungen.

2.1.1 Erzeugung von Bauelementmodellen

Die sogenannten physikalischen Bauelementmodelle beruhen auf der Bauelementephysik, genauer einer Approximation der physikalischen Eigenschaften des Elementes. Bei Transistoren basieren die Modellparameter auf Größen wie der Dicke der Oxidschicht, der Höhe der Substratdotierung, der Ladungsträgerbeweglichkeit und vielen anderen. In der Vergangenheit wurden derartige Modelle sehr stark genutzt, aber aufgrund der hohen Komplexität in aktuellen Technologien sind sie inzwischen inpraktikabel.

Rein empirische Modelle basieren im Gegensatz zu den physikalischen Modellen auf Kurvenanpassung. Bei der Wahl der genutzten Parameter steht im Vordergrund, bei möglichst kleiner Zahl an Parametern eine bestmögliche Kurvenanpassung an vorliegende Messdaten zu erreichen. Die Parameter haben dabei keinen oder nur noch indirekt Bezug zu den physikalischen Parametern.

Aktuell genutzte Modelle, wie sie beispielsweise in Simulatoren wie Spice, Spectre [Cad08] oder Titan [FeSc92] eingesetzt werden, sind Hybride aus physikalischen und empirischen Modellen. Die gebräuchlichsten Modelle für MOS-Feldeffekttransistoren sind das EKV-Modell [BuLa98] und die Modelle aus der BSIM-Reihe der University of California, Berkeley.

2.1.2 Tabellenmodelle

Als eine weitere Klasse an Modellen lassen sich die sogenannten Tabellenmodelle identifizieren. Bei Tabellenmodellen werden zuvor gemessene Größen eines Bauelements, wie Spannungen und Ströme, in Form von Tabellen abgelegt, auf die später während der Simulation zurückgegriffen wird. Werte, die nicht in der Tabelle gespeichert sind, werden während der Simulation durch Interpolation ermittelt. Jedoch können nur solche Werte bestimmt werden, die innerhalb des Messbereichs liegen. Eine Extrapolation über den vor der Simulation gemessenen Bereich hinaus kann in der Regel nicht erfolgen, da Tabellenmodelle technologieunabhängig sind und daher die nötigen Informationen über den physikalischen Aufbau des Bauelements nicht enthalten.

Für verschiedene Parameter \vec{p} (wie Bauelementgeometrien bzw. verschiedene Temperaturen) müssen unterschiedliche Tabellen abgespeichert werden, was ein Nachteil gegenüber einem analytischen Modellierungsansatz ist. Der Vorteil von Tabellenmodellen ist jedoch in der Regel eine kürzere Simulationszeit, da die mathematischen Beschreibungen analytischer Modelle komplexer sind.

2.2 Schaltungsmodelle

Um das Schaltungsverhalten zu modellieren, gibt es je nach Abstraktionsebene verschiedene Ansätze. Die in den folgenden Unterabschnitten betrachteten Ansätze setzen dabei Wissen zumindest über den prinzipiellen Aufbau der Schaltung voraus und sind daher im Kontext einer Bottom-Up-Modellierung zu sehen.

2.2.1 Systemgleichungen aus Netzlisten ermitteln

Eine Schaltung setzt sich aus mehreren Bauelementen zusammen. Die strukturellen Informationen der Schaltung werden in einer Netzliste erfasst. In einem Schaltungssimulator, wie beispielsweise SPICE, beruht die Beschreibung des elektrischen Systems auf den kirchhoffschen Erhaltungssätzen. Das dynamische Verhalten der Ge-

samtschaltung wird durch ein nichtlineares Algebro-Differentialgleichungssystem beschrieben (kurz: DAE-System) :

$$\vec{f}(\vec{x}, \vec{z}(t), \dot{\vec{z}}(t), \vec{p}) = \vec{0} \quad (2.1)$$

Der Vektor $\vec{z}(t)$ repräsentiert die unbekanntes zeitabhängigen Größen in der Schaltung. Dies können die Knotenpotentiale oder zusätzliche Zweigströme durch Quellen oder Induktivitäten sein. Die zeitlichen Ableitungen der unbekanntes Größen werden als $\dot{\vec{z}}(t)$ dargestellt. Der Vektor \vec{p} beinhaltet die zeitunabhängigen Schaltungsparameter. Die Eingangsgrößen, die als Erregung für das System dienen, werden mit $\vec{x}(t)$ bezeichnet. Zum Ermitteln von Systemgleichungen aus einer gegebenen Netzliste bedient man sich in der Regel der modifizierten Knotenformulierung (engl. Modified-Nodal-Analysis, kurz: MNA) oder der Tableauformulierung.

In der Tableauformulierung [HaBr71] werden sowohl alle Knoten- als auch alle Maschengleichungen für das System aufgestellt. Die resultierende Systemmatrix ist somit sehr groß. Ein großer Vorteil der Tableauformulierung besteht darin, dass im Lösungsvektor alle Zweigströme und Knotenspannungen enthalten sind und dadurch keine zusätzlichen Berechnungen zum Erhalt von Systemgrößen mehr durchgeführt werden müssen. Dem Nachteil der großen Systemmatrix kann durch den Einsatz von Sparse-Matrix Techniken begegnet werden, da die aufgestellte Systemmatrix nur schwach besetzt ist.

Die modifizierte Knotenformulierung benutzt – anders als die Tableauformulierung – zunächst nur die Knotenpotentiale der Netzliste als unabhängige Variablen. Lediglich für die Bauelemente, die nicht spannungsgesteuert dargestellt werden können, werden die Systemmatrix und der Lösungsvektor bezüglich der Zweigströme erweitert.

2.2.2 Makromodelle

Eine Abstraktionsebene über der elektrischen Ebene nutzt man die sogenannten Makromodelle. Modellierung auf der Makroebene erfordert vom Modellentwickler detaillierte Kenntnisse über die Schaltung. Die Schaltung wird dabei in verschiedene Funktionsblöcke unterteilt, für die jeweils eigene Gleichungen definiert werden. Um das Modell weiterhin einfach in einer Schaltungssimulation auf elektrischer Ebene einsetzen zu können, werden die Funktionsblöcke häufig durch einfache elektrische Bauelemente (gesteuerte Quelle, Widerstände, Kapazitäten) nachgebildet.

Aufgrund des vereinfachten Aufbaus und mathematisch weniger komplexen Formulierungen zeichnen sich Makromodelle durch eine erhöhte Simulationsgeschwindigkeit aus. Jedoch ist es nicht möglich, technologieabhängige Parameter in die Modellierung einfließen zu lassen, so dass ein einfaches Übertragen auf andere Prozesse

schwierig ist. Die lange Zeit, die Entwickler zum Erstellen und Modifizieren des Makromodells benötigen, ist ebenfalls ein Nachteil.

Ein sehr bekanntes, empirisch ermitteltes Makromodell ist das Operationsverstärkermodell nach Boyle et. al. [BoCh74], von dem in Folge eine Vielzahl von weiteren Modellen abgeleitet wurden. In Abbildung 2.2 ist die Struktur des Modells dargestellt.

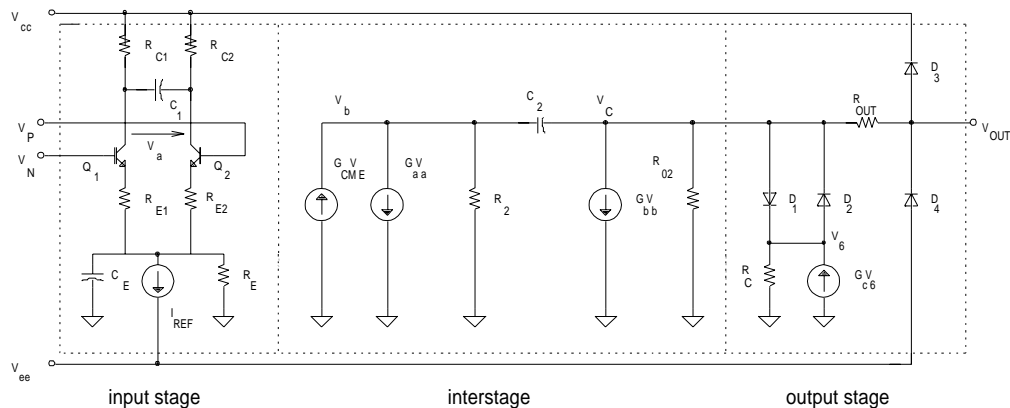


Abbildung 2.2: Struktur des Operationsverstärkermodells nach Boyle

Die Grundidee dieses Makromodells ist die Trennung der Eingangsstufe des Operationsverstärkers von den restlichen Schaltungsteilen. In der Eingangsstufe eines Differenzverstärkers wird die durch Transistoren realisierte Stromquelle (meist durch Stromspiegel implementiert) durch eine ideale konstante Stromquelle für I_{ref} ersetzt. Die übrigen Eigenschaften werden durch gesteuerte Quellen, RC-Glieder und Dioden modelliert. Aufgrund der Leistungsfähigkeit dieses Ansatzes folgten Erweiterungen, um neue Transistortechnologien, wie JFET und MOSFET, abdecken zu können, sowie Ergänzungen bezüglich des Frequenzverhaltens und Berücksichtigung von Betriebschwankungen (Temperatur, etc.).

Die verbesserte Simulationsgeschwindigkeit dieses Modells gegenüber der vollständigen Netzlistenbeschreibung wird jedoch durch einige Nachteile erkauft, die letztlich einer allgemeinen Verwendbarkeit entgegenstehen. So muss der Entwickler sowohl detaillierte Kenntnisse über den Aufbau des Operationsverstärkers besitzen, als auch die genauen Einsatzbedingungen des Modells wissen. Es werden ausschließlich Eigenschaften und Effekte wiedergegeben, die der Entwickler in das Modell hat einfließen lassen. Weiterhin kann das Modell beispielsweise nicht unabhängig von der Last benutzt werden, so dass ein Redesign für die spezielle Umgebung notwendig ist. Da die Modelle nicht parametrierbar sind, können sie weder einfach an neue Aufgaben angepasst werden, noch für eine automatische Modellerzeugung und Modellanpassung bei neuen Schaltungen genutzt werden.

Die Charakteristika der modellierten Schaltung werden durch die Kennlinien der im Modell genutzten linearen und nicht linearen Bauelemente erzeugt. Dies bedingt eine nur eingeschränkte Gültigkeit bezüglich des Arbeitsbereiches des Modells.

2.2.3 PWL-Modellierung

Ebenfalls zur Methode der Makromodellierung lässt sich der Ansatz zählen, das Schaltungsverhalten durch Makroblöcke mit stückweise linearem Verhalten wiederzugeben. Im Allgemeinen kann so eine weitere Reduktion der Komplexität des Schaltungsmodells erzielt werden, da nicht lineare Blöcke auf diese Weise ersetzt werden können.

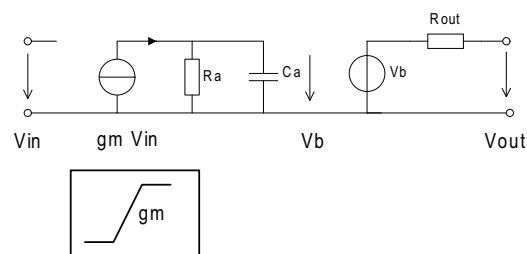


Abbildung 2.3: Modell eines Operationsverstärkers mit stückweise linear modellierter Stromquelle

Abbildung 2.3 zeigt hierzu exemplarisch die Realisierung eines Operationsverstärkermodells. Der nichtlineare Verlauf für die Stromverstärkung wird hier durch eine stückweise lineare Funktion nachgebildet. Diese Technik wurde bereits früh in speziellen Simulatoren [KaOh75] [StEi90] eingesetzt, welche sich zu Nutze machen, dass die Modellgleichungen in spezifizierten Regionen aus linearen Funktionen bestehen. Die Lösung des mathematischen Problems besteht somit in der Lösung eines linearen Gleichungssystems und der Unterscheidung, dass der Lösungsraum in mehrere lineare - oder genauer affine - Bereiche aufgeteilt ist. Damit verbunden ist somit eine Suche nach dem für die Lösung aktuell gültigen Bereich.

In dieser Arbeit wird eine Modellierung mit Hilfe einer stückweise affinen Beschreibung genutzt, auf die in Kapitel 4 detailliert eingegangen wird.

2.2.4 Strukturerkennung

Anhand von vordefinierten Regeln lassen sich in Netzlisten Substrukturen erkennen, für die eine gesonderte abstraktere Modellierung sinnvoll erscheint. Diese Regeln können meist vom Anwender angegeben werden. Das in [OIRe01, OIPo01] vorgestellte

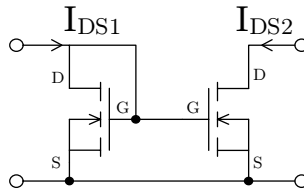


Abbildung 2.4: Einfacher CMOS-Stromspiegel

Werkzeug beispielsweise bedient sich hierzu einer auf Graphenisomorphie basierenden Methode.

Zur Einstellung von Referenzströmen in integrierten Schaltungen werden beispielsweise Stromspiegel eingesetzt. Die einfachste Realisierung eines solchen Stromspiegels besteht aus zwei Transistoren und ist in Abbildung 2.4 dargestellt. Der in den linken Zweig hineinfließende Strom dient als Referenz. Durch die Verbindung von Drain und Gate stellt sich am Gate der Transistoren ein Potential ein, das den Fluss des Referenzstroms durch den linken Transistor sicherstellt. Da beide Transistoren eine identische Gate-Source-Spannung aufweisen, können Ein- und Ausgangstrom zueinander in Beziehung gesetzt werden.

In integrierten Schaltungen sind die Kanallängen beider Transistoren im Allgemeinen gleich, um Effekten wie Kanallängenmodulation und Beweglichkeitsmodulation aus dem Weg zu gehen. Trifft dies zu, kann die Gleichung

$$\frac{I_{D2}}{I_{D1}} = \frac{W_2}{W_1} \quad (2.2)$$

abgeleitet werden, wobei W_1 und W_2 die Kanalweiten der beiden Transistoren sind. Somit ist ein sehr einfaches Modell eines Stromspiegels beschrieben, welches zwei wesentlich komplexere Gleichungssysteme von zwei zusammenhängenden Transistoren ersetzen kann. Die Strukturerkennung ist folglich ein Instrument, das unterstützend für das Erzeugen von Makromodellen eingesetzt werden kann.

Weitere relevante Strukturen aus Transistorpaarungen sind Differenzstufe oder Levelshifter. Das Extrahieren solch einfacher Strukturen ist mit Hilfe einer Strukturerkennung gut zu realisieren. Für komplexere Strukturen wie beispielsweise Operationsverstärker oder Transkonduktanzverstärker ist das Erstellen eindeutiger Regeln jedoch grundsätzlich problematisch.

2.3 Blackbox-Ansatz

Ein völlig anderer Ansatz zur Modellierung ist der sogenannte Blackbox-Ansatz. Bei dieser Art der Modellierung handelt es sich um eine rein numerische Modellierung,

die keinerlei Kenntnis der eigentlichen Schaltung voraussetzt. Hierbei werden in der Regel folgende Schritte durchgeführt:

- Modellparameter festlegen
- Zu modellierenden Betriebsbereich bestimmen
- Mathematische Modellbeschreibung auswählen
- Mehrere Schaltungssimulationen für zuvor definierte Parametersätze durchführen
- Koeffizienten des Modells ermitteln

Als Modellparameter werden dabei diejenigen Parameter gewählt, die beim späteren Modell variierbar sein sollen. Bezogen auf das System in Abbildung 2.1 sind dies die Vektoren \vec{x} und \vec{p} . Zur Erstellung des Modells werden an mehreren Punkten im Parameterraum, den sogenannten Entwurfspunkten, Simulationen der zu modellierenden Schaltung durchgeführt. Der Parameterraum bildet den Betriebsbereich, für den das Modell ausgelegt sein soll. Zur Auswahl der Entwurfspunkte können dabei statische und dynamische Verfahren verwendet werden, d.h. sie werden beispielsweise über ein festes Raster vorab statisch definiert, oder die Wahl der Entwurfspunkte wird dynamisch während der Simulationen vorgenommen.

Bei der Wahl der mathematischen Modellbeschreibung legt man zunächst die Basisfunktionen fest, mit der modelliert werden soll (z.B. Polynome, Splines, ...). Die Beschreibung sollte dabei der zu modellierenden Eigenschaft angepasst sein, um die Modellierungsfehler möglichst klein halten zu können. Modelliert man beispielsweise über ein durch zwei Grenzen definiertes Gebiet, besitzen die Eigenschaften an den Randbereichen oft eine stark abweichende Charakteristik gegenüber den Eigenschaften im inneren Bereich.

Aus den Simulationsdaten werden durch anschließendes Fitting, Interpolation oder Approximation die Koeffizienten der mathematischen Modellbeschreibung ermittelt.

2.4 Unsicherheiten

Wie einleitend erwähnt, sind die Ergebnisse einer Simulation grundsätzlich mit Unsicherheiten behaftet. Betrachtet man das Modell eines integrierten Bauelements, wirken sich die in Abbildung 2.5 dargestellten Faktoren auf das Verhalten aus. So gilt unabhängig von der Modellierungsart, dass je höher der Abstraktionsgrad des Modells ist, desto schlechter wird das reale Verhalten angenähert (*Abstraktionsfehler*).

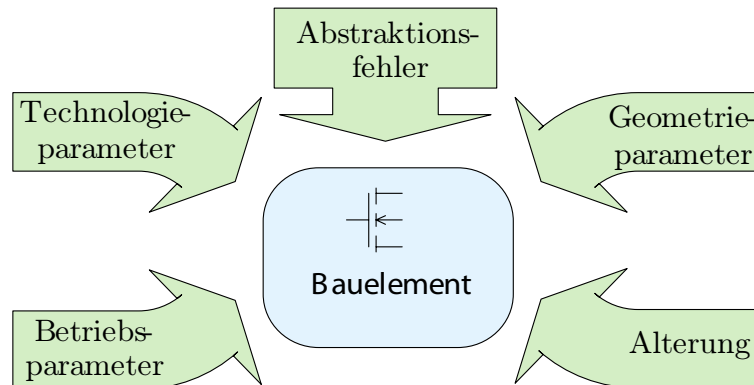


Abbildung 2.5: Faktoren für Unsicherheiten im modellierten Verhalten eines integrierten Bauelements

Weitere Variationen sind die Abweichungen von den ursprünglich vom Entwickler angenommenen Betriebs-, Technologie- oder Schaltungsparametergrößen. Sie beeinflussen das Schaltungsverhalten bzw. die gewünschten Performancegrößen und Charakteristiken der Schaltungen. Als beeinflussende Betriebsparameter werden Größen wie Versorgungsspannung oder Temperatur bezeichnet. Ihr Schwankungsbereich ist meistens durch die Spezifikation mit einer Ober- und einer Untergrenze angegeben.

Die Anzahl statistisch signifikant schwankender Technologie- und Geometrieparameter wird mit fortschreitenden Technologien immer größer und die Schwankungsbreiten nehmen weiter zu. Die resultierenden Variationseffekte können in einer digitalen Schaltung beispielsweise abweichende Verzögerungszeiten auf den Signalleitungen und damit eine Verschiebung der maximal zulässigen Taktfrequenz sein. In analogen Schaltungen wirken sich Variationen jedoch noch wesentlich stärker auf das Verhalten aus. Dabei gibt es Abweichungen, die den gesamten Chip gleichermaßen betreffen, aber auch Streuungen nominal gleicher Eigenschaften untereinander innerhalb des Chips.

Eine Art der Variationen stellen die geometrischen Variationen dar. Sie beziehen sich auf Abweichungen in den geometrischen Strukturen von MOSFETs oder anderen Bauelementen. Ein kritischer Parameter bezüglich des Verhaltens von MOSFETs ist beispielsweise die Oxidschichtdicke, die jedoch im Herstellungsprozess gut beherrscht wird. Variationen treten somit hauptsächlich zwischen unterschiedlichen Wafern auf. Zwischen zwei Chips auf ein und dem selben Wafer sind die Variationen wesentlich kleiner. Dies gilt es insbesondere bei aneinander angepassten Elementen zu berücksichtigen.

Weitere geometrische Variationen betreffen die lateralen Ausdehnungen, also Längen und Weiten, auf dem Chip. Sie haben ihre Ursache im photolithographischen Herstellungsprozess. Auch hier sind insbesondere MOSFETs empfindlich, da die effektive Kanallänge einen starken Einfluss auf das Verhalten des Drain-Source-Stromes hat.

Eine weitere Klasse bilden die Variationen von technologischen Parametern. Einen starken Effekt haben hier Abweichungen in der Dotierung des Substrates. Dabei werden Ionen des Dotierungselements in einem elektrischen Feld (50 kV bis 200 kV) beschleunigt und dringen beim Auftreffen in die Siliziumoberfläche des Chips ein. Die Eindringtiefe und die Konzentration werden über die Beschleunigungsspannung und den Ionenstrom gesteuert und unterliegen lokalen Schwankungen. In MOSFETs wirken sich diese Schwankungen auf entscheidende elektrische Parameter, wie beispielsweise die Threshold-Spannung aus.

Als letzter Punkt sei hier noch der Einfluss durch Alterung genannt. Die Bauelemente einer integrierten Schaltung sind im normalen Betrieb elektrischen Belastungen ausgesetzt. Hohe Feldstärken bei geringsten Abmessungen sorgen dabei auf physikalischer Ebene für einen beschleunigten Alterungsprozess. Die bekanntesten Effekte sind Hot Carrier Injection und Negative Bias Temperature Instability [KuKi06].

3 Semi-symbolische Simulation

3.1 Simulation analoger Schaltungen

Für eine Simulation von analogen Schaltungen und Systemen benötigt man eine mathematische Beschreibung des Verhaltens. Die Herleitung dieser Beschreibung wird als Modellierung bezeichnet und wurde in Kapitel 2 behandelt. Je nach gewünschter Genauigkeit, Abstraktionsgrad und Einsatzgebiet können dabei unterschiedliche Ansätze für die Modellierung verfolgt werden. Sobald das System mathematisch beschrieben ist, lässt sich das analytische Verhalten des Systems vorhersagen.

Definition 3.1 *Eine Simulation ist eine Auswertung bzw. das Lösen der mathematischen Systembeschreibung unter Berücksichtigung gegebener Eingangserregungen. In die Betrachtung können des Weiteren Zeit- oder Frequenzverlauf der Eingangs- und Systemgrößen miteinbezogen werden.*

3.1.1 Analysearten

Man unterscheidet zwischen drei Grundanalysearten:

Arbeitspunktanalyse Die Arbeitspunktanalyse (engl. DC-Operating Point Analysis) berechnet den eingeschwungenen Zustand eines Systems für $t \rightarrow \infty$ bei konstanten Eingangsgrößen. Somit können aus der mathematischen Systembeschreibung alle zeitlichen Ableitungen $\vec{z}(t)$ zu Null gesetzt werden. Im Allgemeinen erhält man so ein nichtlineares algebraisches Gleichungssystem, das beispielsweise mit Hilfe des Newton-Raphson-Algorithmus gelöst werden kann.

Frequenz-Analyse Hier wird das Verhalten einer Schaltung als Funktion der Frequenz berechnet. Um dies zu erreichen, wird das System in einem zuvor ermittelten Arbeitspunkt linearisiert. Nach erfolgter Berechnung stehen alle Spannungen und Ströme nach Betrag und Phase der Schaltung im betrachteten Frequenzbereich zu Verfügung.

Zeitbereichsanalyse Die Zeitbereichsanalyse, auch als Transientenanalyse bezeichnet, berechnet das Systemverhalten über der Zeit.

Aufbauend auf diesen Grundanalysen sind folgende weiterführende Analysen gebräuchlich:

Parameteranalyse Die Parameteranalyse, oder auch Parameter Sweep Analyse, setzt sich aus mehrfach hintereinander ausgeführten Arbeitspunktanalysen zusammen. Hierbei werden ein oder mehrere Parameter in \vec{p} über einen zuvor definierten Wertebereich verändert.

DC-Transfer-Analyse Die DC-Transfer- oder auch DC-Sweep-Analyse, besteht wie die Parameteranalyse aus mehrfach ausgeführten Arbeitspunktanalysen. Hierbei wird jedoch die Eingangserregung \vec{x} über einen zuvor definierten Wertebereich verändert.

Fourier-Analyse Diese Analyse ermöglicht die Untersuchung der DC- und Frequenzanteile innerhalb einer Schaltung. Voraussetzung für diese Berechnung ist eine vorangegangene Transientenanalyse mit ausreichender Simulationszeit, so dass der eingeschwungene Zustand erreicht wird.

Des Weiteren sind einige spezielle Analysen gebräuchlich, die in Entwurf und Verifikation zum Einsatz kommen:

- Rausch-Analyse (engl. Noise Analysis)
- Pol-Nullstellen-Analyse
- Empfindlichkeits-Analyse (engl. Sensitivity Analysis)
- Verzerrung-Analyse (engl. Distorsion Analysis)

Die gebräuchlichste Analyseart ist die Zeitbereichsanalyse, auf die im folgenden Abschnitt näher eingegangen wird.

3.1.2 Zeitbereichsanalyse

Ausgangspunkt für die Zeitbereichsanalyse sind die in Gleichung (2.1) vorgestellten DAEs. Der prinzipielle Ablauf einer Zeitbereichsanalyse ist in Abbildung 3.1 dargestellt. Zur Lösung der DAEs werden numerische Integrationsverfahren eingesetzt, da sich in der Regel keine explizite Integration durchführen lässt. Als typische Integrationsverfahren sind das Trapez- und das Backward-Euler-Verfahren im Einsatz, als Standard in Spice wird der Gear-Algorithmus eingesetzt [Spir90].

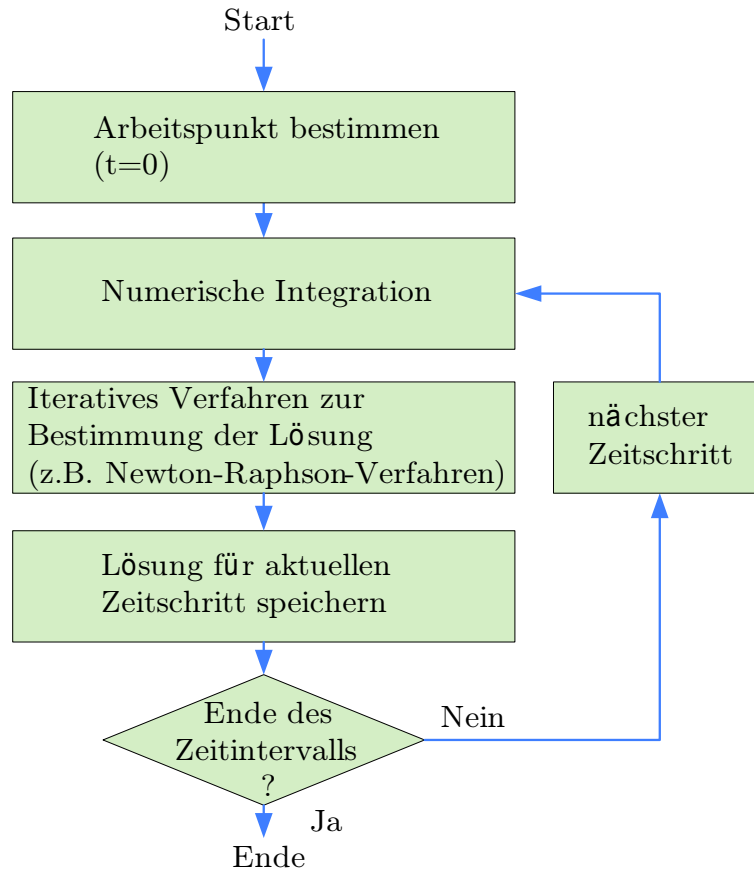


Abbildung 3.1: Ablauf der Zeitbereichsanalyse

Aufgrund von Nichtlinearitäten in der Systemmatrix werden iterative Algorithmen wie das Newton-Raphson-Verfahren benötigt, um eine Lösung zu jedem beliebigen Zeitpunkt ermitteln zu können. Die grundlegende Idee dieses Verfahrens ist, die Funktion in einem Ausgangspunkt zu linearisieren, d. h. ihre Tangente zu bestimmen, und die Nullstelle der Tangente als verbesserte Näherung der Nullstelle der Funktion zu verwenden. Die erhaltene Näherung dient als Ausgangspunkt für einen weiteren Verbesserungsschritt. Diese Iteration erfolgt, bis die Änderung in der Näherungslösung eine festgesetzte Schranke unterschritten hat.

Aufgrund numerischer Instabilität dieses Verfahrens oder aufgrund von Unstetigkeiten in Modellgleichungen von Schaltungselementen kann es dabei zu Konvergenzproblemen beim Lösungsprozess kommen. Hierdurch wird die Simulationszeit signifikant erhöht, da zusätzliche Iterationen benötigt werden. Letztlich ist nicht garantiert, dass sich eine Lösung ermitteln lässt.

3.1.3 Stochastische Analysen

Mit stochastischen Analysen kann untersucht werden, wie sich ändernde Bauteileigenschaften bzw. Toleranzen auf das Verhalten einer Schaltung auswirken. Hierbei werden mehrere Analysedurchläufe ausgeführt und bei jedem Durchlauf die Bauteilparameter \vec{p} entsprechend der eingestellten Verteilungsart und Toleranz variiert. Diese Simulationsart wird als *Monte-Carlo-Simulation* bezeichnet. Sie lässt sich mit den Grundanalysearten DC-Sweep, AC-Sweep und der Transientenanalyse verbinden.

Die Monte-Carlo-Simulation ist eine Sampling-Methode, da die Eingangsgrößen zufällig aus einer Wahrscheinlichkeitsverteilung, meist einer Normal-Verteilung, genommen werden. Für aussagekräftige Ergebnisse müssen die Verteilungen entsprechend nah an der Realität liegen. Die Ergebnisse der Monte-Carlo-Simulation können Kurvenscharen sein, die alle ermittelten Lösungen oder Wahrscheinlichkeitsverteilungen darstellen.

Grundsätzlich gilt, dass, je mehr Samples genommen werden (also je mehr Simulationsläufe mit variierenden Parametern durchgeführt werden), desto zuverlässiger sind die erzielten Ergebnisse. Eine formal sichere Aussage, ob das tatsächliche Schaltungsverhalten durch die Simulationen erfasst wurde, gibt es nur im Sinne statistischer Aussagen.

3.2 Symbolische Analyse

Die symbolische Analyse ist eine Technik zur Modellierung und Berechnung des Verhaltens bzw. zur Bestimmung charakteristischer Eigenschaften einer Schaltung. Dazu nutzt man symbolische algebraische Berechnungsmethoden. Ausgangspunkt ist meist ein aus der vollständigen Schaltungsbeschreibung gewonnenes symbolisches nichtlineares Algebro-Differentialgleichungssystem

Definition 3.2 *Ein symbolisches Gleichungssystem liegt vor, wenn ein oder mehrere Parameter \vec{p} des Gleichungssystems nicht als numerischer Zahlenwert vorliegen, sondern durch ein Symbol repräsentiert werden.*

Je nach Zielsetzung werden unterschiedliche Algorithmen zur Analyse genutzt, die jedoch in der Regel eine Vereinfachung der Komplexität des Gleichungssystems vornehmen. Die unterschiedlichen Ziele der symbolischen Analyse können dabei sein:

- Erzeugung von Gleichungen, die zur automatischen oder manuellen Dimensionierung der Schaltung eingesetzt werden können,

- Gewinnung von dominanten Pol-/ und Nullstellen für Stabilitätsuntersuchungen,
- besseres Verständnis der Funktionsweise einer Schaltung,
- Vereinfachung des Gleichungssystems zur Gewinnung eines abstrahierten Verhaltensmodells.

Die überwiegende Zahl der entwickelten Algorithmen untersucht dabei lineare Schaltungen [GiSa91, Lin91, WaGi98]. Diese Verfahren haben mittlerweile einen Entwicklungsstand erreicht, bei dem ein Einsatz im industriellen Umfeld möglich ist. Die Verfahren zur Analyse nichtlinearer Schaltungen sind im Gegensatz dazu noch Gegenstand der Forschung. Dies liegt zum Teil an der mathematisch schwieriger zu behandelnden Darstellung durch nichtlineare Algebro-Differentialgleichungssysteme.

Die Verfahren werden meist in Verbindung mit Computeralgebrasystemen eingesetzt, um die notwendigen analytischen Umformungen ausführen zu lassen. Bekannte Systeme sind hier AMV [Hedr04], als Erweiterungspaket zum Computeralgebrasystem Maple, oder Analog-Insydes [Fra08], welches auf Mathematica aufbaut. Alle auf Schaltungsebene bezogenen Ergebnisse dieser Arbeit nutzen Analog-Insydes, um ein symbolisches Gleichungssystem aus einer Netzlistenbeschreibung zu gewinnen.

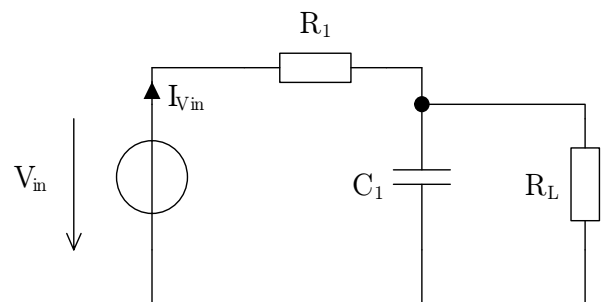


Abbildung 3.2: Belasteter Tiefpass

Zur Verdeutlichung wird das Beispiel eines belasteten Tiefpasses betrachtet, der in Abbildung 3.2 dargestellt ist. Eine entsprechende Netzlistenbeschreibung in Analog-Insydes besitzt folgende Form:

```
Netlist[
  {C1, {Node1, 0}, Type -> Capacitor, Symbolic -> C1},
  {R1, {Node0, N1}, Type -> Resistor, Symbolic -> R1},
  {RL, {Node1, 0}, Type -> Resistor, Symbolic -> RLoad},

  {Vin, {Node0, 0}, Type -> VoltageSource, Symbolic -> Vin}
]
```

Unter Anwendung der modifizierten Knotenanalyse (siehe Abschnitt 2.2.1) lassen sich die Netzwerkgleichungen im Frequenzbereich in Matrixform aufstellen:

$$\begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_L} + C_1 s & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} V_{Node0} \\ V_{Node1} \\ I_{in} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ V_{in} \end{pmatrix} \quad (3.1)$$

Hiernach erfolgt die symbolische Auflösung der Gleichungen nach dem Knotenpotential V_{Node1} und anschließend die Berechnung des Teilverhältnisses $H(s)$ mit Division durch V_{in} . Es ergibt sich

$$H(s) = \frac{R_L}{R_1 + R_L + C_1 R_1 R_L s}. \quad (3.2)$$

Diese symbolische Darstellung der Übertragungsfunktion $H(s)$ ermöglicht einen guten Einblick in den Einfluss der Parameter R_1 , C_1 und R_L .

Verwendet man Bauelemente mit komplexeren Bauelementgleichungen wächst das Gleichungssystem schnell auf mehrere hundert Gleichungen an. Dies bedingt dann in Folge die bereits weiter oben erwähnten Vereinfachungsmethoden, um ähnlich wie in Gleichung (3.2) einen aussagekräftigen Ausdruck zu erhalten.

Neben der Betrachtung im Frequenzbereich lassen sich die Gleichungen alternativ auch im Zeitbereich erzeugen. Für das Beispiel in Abbildung 3.2 ergibt sich:

$$\begin{aligned} \frac{I_{in}(t) + V_{Node0}(t) - V_{Node1}(t)}{R_1} &= 0 \\ \frac{V_{Node1}(t)}{R_L} + \frac{-V_{Node0}(t) + V_{Node1}(t)}{R_1} + C_1 \frac{\delta V_{Node1}(t)}{\delta t} &= 0 \\ V_{Node0}(t) - V_{in} &= 0 \end{aligned}$$

3.3 Bereichsarithmetiken

Ausgangspunkt für den Einsatz von Bereichsarithmetiken im Simulationsprozess ist die Möglichkeit, Unsicherheiten in den Modellparametern und Stimuli durch ein Intervall, das alle möglichen Unsicherheiten einschließt, wiedergeben zu können. Im Folgenden wird zunächst die Intervallarithmetik, die erstmals 1966 von Moore [Moor66] vorgestellt wurde, und in der Vergangenheit beispielsweise für Toleranzanalysen eingesetzt wurde [Kole93], erläutert. Danach wird die später entwickelte Affine Arithmetik, die auch als Grundlage dieser Arbeit dient, vorgestellt und abschließend ein Vergleich beider Arithmetiken vorgenommen.

3.3.1 Intervallarithmetik

Die Intervallarithmetik bildet die Grundlage, mathematische Operationen mit Intervallen oder im mehrdimensionalen Fall mit Gebieten $X \subseteq \mathbb{R}^n$ (Hyperkuben) durchführen zu können. Die Gebiete $X = [\underline{X}, \overline{X}]$ werden durch eine obere und eine untere Grenze dargestellt.

Die Intervallarithmetik definiert für verschiedene mathematische Operationen $f(X)$ Einschlussfunktionen $F(X)$. Dabei gilt $f(X) \subseteq F(X) = [F_L(X), F_H(X)]$, so dass man von einem sicheren Einschluss von $f(X)$ innerhalb der Grenzen von $F(X)$ spricht.

Zur Veranschaulichung werden im Folgenden zwei Intervalle $[a, b]$ und $[c, d]$ betrachtet. Für die Operationen $f = \{+, -, \cdot, /\}$ definiert die Intervallarithmetik:

$$\begin{aligned} [a, b] - [c, d] &:= [a - d, b - c] \\ [a, b] + [c, d] &:= [a + c, b + d] \\ [a, b] \cdot [c, d] &:= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \\ [a, b] / [c, d] &:= [a, b] \cdot [1/d, 1/c] \quad \text{wenn } 0 \notin [c, d] \end{aligned}$$

Grundsätzlich führt eine Verschachtelung mehrerer Funktionen über X zu immer größer werdenden Gebieten (Aufweitung), die letztlich keine Interpretation der Ergebnisse mehr zulassen.

3.3.2 Affine Arithmetik

Um die Problematik der Aufweitungen in der Intervallarithmetik zu verringern, wurde von Stolfi et al. [AnCo94] die affine Arithmetik (AA) entwickelt. In der AA werden Intervalle durch Polynome erster Ordnung ausgedrückt. Diese sogenannte affine Form definiert sich als

$$\hat{x} = x_0 + \sum_{i=1}^{\xi} x_i \varepsilon_i \quad \text{mit } \varepsilon_i = [-1, 1]. \quad (3.3)$$

Hier und im Folgenden werden Variablen in affiner Form durch \hat{x} gekennzeichnet. Die Variable x_0 wird als Nominal- oder Zentralwert bezeichnet. Die Koeffizienten x_i sind numerische Größen und bestimmen die Größe der partiellen Abweichung vom Wert x_0 . Über die Symbole ε_i lassen sich die Abhängigkeiten zwischen unterschiedlichen \hat{x} definieren. Die Gesamtzahl aller partiellen Abweichungen ist ξ . Der Radius einer affinen Größe ist definiert durch

$$\text{rad}(\hat{x}) = \sum_{i=1}^{\xi} |x_i| \quad (3.4)$$

und bezeichnet die maximale Abweichung von x_0 .

Ein Intervall $X = [\underline{x}, \bar{x}]$ lässt sich über folgende Vorschrift in eine affine Form überführen:

$$\hat{x} = \frac{\underline{x} + \bar{x}}{2} + \frac{\underline{x} - \bar{x}}{2} \epsilon_{\xi+1} \quad (3.5)$$

Ein Ausdruck \hat{x} in AA-Form kann mit Hilfe der rad -Funktion in ein Intervall zurückgeführt werden:

$$X = [x_0 - \text{rad}(\hat{x}), x_0 + \text{rad}(\hat{x})] \quad (3.6)$$

Die mathematischen Operationen in der AA lassen sich in die Klasse der affinen und der nicht-affinen Operationen aufteilen. Zur Klasse der affinen Operationen zählen die folgenden Operationen:

- Addition
- Subtraktion
- skalare Multiplikation

Gleichung (3.7) zeigt, wie sich diese auf einfache mathematische Operationen zurückführen lassen. Die Größen \hat{x} und \hat{y} sollen hierbei in affiner Form vorliegen und es gelte $\alpha, \beta, \gamma \in \mathbb{R}$.

$$\alpha \hat{x} + \beta \hat{y} + \gamma = (\alpha x_0 + \beta y_0 + \gamma) + \sum_{i=1}^{\xi} (\alpha x_i + \beta y_i) \quad (3.7)$$

Zur Klasse der nicht-affinen Operationen op_{na} zählen beispielsweise Multiplikation, Division, $\exp(\cdot)$, $\ln(\cdot)$ oder $\text{sqrt}(\cdot)$. So entspricht eine Multiplikation zweier affiner Formen im Ergebnis keiner affinen Form mehr:

$$\begin{aligned} \hat{x} \cdot \hat{y} &= (x_0 + \sum_{i=1}^{\xi} x_i \epsilon_i) \cdot (y_0 + \sum_{i=1}^{\xi} y_i \epsilon_i) \\ &= x_0 y_0 + \sum_{i=1}^{\xi} (x_0 y_i + x_i y_0) \epsilon_i + \left(\sum_{i=1}^{\xi} x_i \epsilon_i \right) \cdot \left(\sum_{i=1}^{\xi} y_i \epsilon_i \right). \end{aligned}$$

Um dennoch eine affine Form zu erhalten, wird das Ergebnis durch eine Funktion \hat{f}_{approx} approximiert:

$$\hat{y} = op_{na}(\hat{x}) = \hat{f}_{approx}(\hat{x}) + \underbrace{y_{\xi+1} \epsilon_{\xi+1}}_{\hat{f}_{extend}(\hat{x}, \hat{y})} \quad (3.8)$$

Der zusätzliche Term $\hat{f}_{extend} = y_{\xi+1}\varepsilon_{\xi+1}$ verursacht dabei eine zusätzliche symmetrische Aufweitung und wird so gewählt, dass ein sicherer Einschluss des exakten Ergebnisses gewährleistet ist. Eine effiziente Berechnung ergibt sich beispielsweise durch

$$\hat{x} \cdot \hat{y} = x_0 y_0 + \underbrace{\sum_{i=1}^{\xi} (x_0 y_i + x_i y_0) \varepsilon_i}_{\hat{f}_{approx}(\hat{x}, \hat{y})} + \text{rad}(\hat{x}) \text{rad}(\hat{y}) \varepsilon_{\xi+1}. \quad (3.9)$$

Als weiteres Beispiel ist im Folgenden die Approximation der $1/x$ -Operation dargestellt:

$$\begin{aligned} \hat{y} = op_{na}(\hat{x}) & : \hat{x} \rightarrow \frac{1}{\hat{x}} \\ a & = x_0 - \text{rad}(\hat{x}), \quad b = x_0 + \text{rad}(\hat{x}) \\ \alpha & = \frac{-1}{b^2} \\ y_a & = \frac{1}{a} - \alpha \cdot a, \quad y_b = 2 \cdot \frac{1}{b} \\ \beta & = 0.5 \cdot (y_a - y_b) \\ \gamma & = 0.5 \cdot (y_a + y_b) \\ \hat{y} & = \underbrace{(\alpha \cdot x_0 + \gamma) + \alpha \sum_{i=1}^{\xi} (x_i \cdot \varepsilon_i)}_{\hat{f}_{approx}(\hat{x})} + \beta \cdot \varepsilon_{\xi+1} \end{aligned} \quad (3.10)$$

Wie in den Gleichungen (3.9) und (3.10) deutlich wird, werden die Korrelationen über ε_i bewahrt.

3.3.3 Überabschätzung

Beide Bereichsarithmetiken (IA und AA) liefern formal sichere Einschüsse des exakten Ergebnisses, welches in der Regel durch eine Überabschätzung des tatsächlichen exakten Ergebnisintervalls erreicht wird. Anhand der Funktion

$$f(x) = (10+x)(10-x) \quad \text{mit } x \in [-\delta, \delta] \quad (3.11)$$

soll dies verdeutlicht werden. Die exakte Berechnung führt zu dem Ergebnis $F_{exakt}(x) = [100 - \delta^2, 100]$, was einer Aufweitung des Ergebnisintervalls von $\text{rad}_{exakt} = \delta^2$ entspricht.

Eine Auswertung von Gleichung (3.11) mit der IA ergibt:

$$\begin{aligned} 10 + x &= [10 - \delta, 10 + \delta] \\ 10 - x &= [10 - \delta, 10 + \delta] \\ F(x) &= (10 + x)(10 - x) \\ &= [(10 - \delta)^2, (10 + \delta)^2] \end{aligned}$$

Das Ergebnisintervall besitzt somit eine Aufweitung von

$$\text{rad}_{IA} = (10 + \delta)^2 - (10 - \delta)^2 = 40\delta. \quad (3.12)$$

Nutzt man hingegen die Affine Arithmetik, bleiben die Korrelationen zwischen den Operanden erhalten:

$$\begin{aligned} f(x) &= (10 + x)(10 - x) \quad \text{mit } x = 0 + \delta \cdot \varepsilon \\ 10 + x &= 10 + \delta \cdot \varepsilon \\ 10 - x &= 10 - \delta \cdot \varepsilon \\ (10 + x)(10 - x) &= 100 - \delta^2 \cdot \varepsilon \end{aligned}$$

Das Ergebnisintervall besitzt eine Aufweitung von $\text{rad}_{AA} = 2\delta^2$ und ist somit exakter als $\text{rad}_{IA} = 40\delta$. Die Überabschätzung durch die AA ist aber nicht zwangsläufig kleiner als bei der IA, wie im obigen Beispiel bei der Wahl $\delta > 20$ deutlich wird.

Abbildung 3.3 zeigt die Ergebnisse für die Berechnungen mit beiden Bereichsarithmetiken, wobei zur Veranschaulichung jeweils mehrere aufeinanderfolgende Intervalle mit einer Breite $2\delta = 4$ dargestellt sind.

Die wesentlichen Vorteile der AA begründen sich in der Tatsache, dass lineare Korrelationen zwischen den Operanden während der Berechnungen erhalten bleiben. Ebenso wurde bereits 1975 ein vergleichbarer Ansatz (*generalized intervall arithmetic*) von Hansen [Hans08, Hans92] vorgeschlagen.

Aufbauend auf den Ergebnissen von [AnCo94] wurden durch Messine [Mess02] Erweiterungen der Affinen Arithmetik um sogenannte Quadratische Formen vorgenommen. Ein vergleichbarer Ansatz wird auch durch Grabowski et al. [GrOI08] verfolgt. Die Absicht dieser Ansätze ist eine weitere Verringerung der Aufweitung, indem auch Korrelationen zwischen den quadratischen Termen während der Berechnung erhalten bleiben.

Zusammengefasst zeigt sich, dass die AA eine sinnvolle Alternative zur IA darstellt. Der Vorteil liegt in der Tatsache, dass Korrelationsinformationen mitgeführt werden. Erkauft wird dieser Vorteile jedoch durch komplexere Berechnung und somit höhere Laufzeiten.

3.4 Semi-symbolische Gleichungssysteme

Betrachtet man erneut das in Abschnitt 2.2.1 vorgestellte Gleichungssystem eines Schaltungsmodells und lässt für die Elemente im Parametervektor \vec{p} affine Formen zu, spricht man von einem semi-symbolischen Gleichungssystem:

Definition 3.3 *Ein symbolisches Gleichungssystem, in dem mindestens einer der symbolischen Parameter \vec{p} durch einen Ausdruck in affiner Form ersetzt wird und für alle übrigen Symbole eine numerische Größe angegeben ist, wird zu einem semi-symbolischen Gleichungssystem.*

Ebenso kann auch der Eingangsvektor neben rein numerischen Eingangswerten affine Formen enthalten. Das Gleichungssystem lässt sich dann wie folgt beschreiben:

$$\vec{f}(\vec{x}, \vec{z}(t), \vec{z}(t), \vec{p}) = \vec{0} \quad (3.13)$$

Die semi-symbolische Simulation beruht auf speziellen Lösungsverfahren für solche semi-symbolischen Gleichungssysteme.

3.5 Bekannte Ansätze zur semi-symbolischen Simulation

Die affine Arithmetik lässt sich in den Gleichungslösungsprozess bei der Schaltungssimulation einsetzen. Die mathematischen Regeln und Approximationen werden hierbei auf die jeweiligen Algorithmen des Simulators übertragen.

Die Idee der semi-symbolischen Simulation wurde zunächst zur Untersuchung von (nicht-konservativen) Systemen auf höherer Abstraktionsebene angewandt [GrHe04]. Diese Systeme sind z.B. Regelkreise, die mit Hilfe von Blockdiagrammen beschrieben werden können. Diese Beschreibung erlaubt eine explizite Formulierung der Ausgangsgröße als Funktion der Eingangsgrößen.

Ein erweiterter Ansatz wurde von [GrGr06] präsentiert. Dieser Ansatz ermöglicht das Lösen von konservativen Systemen, also von Schaltungsmodellen mit Strom- und Spannungsbeziehungen. Ein wesentlicher Unterschied zur Blockdiagrammbeschreibung ist die implizite Form von nichtlinearen Gleichungssystemen (siehe Abschnitt 2.2.1), die das Verhalten der Schaltung beschreiben.

Ausgangspunkt dieses Verfahrens ist eine Formulierung des nichtlinearen Gleichungssystems in symbolischer Form gemäß Definition 3.2. In einem ersten Schritt wird

zunächst die nominale Lösung \vec{z}_0 des Systems, d.h. unter Vernachlässigung aller Unsicherheiten, ermittelt. Hierzu können übliche Standardverfahren genutzt werden.

In einem zweiten Schritt wird dann mit Hilfe der partiellen Ableitungen

$$\mathbf{J}_0 = \left. \frac{\delta \mathbf{F}(\vec{z}, \vec{p})}{\delta \vec{z}} \right|_{\vec{z}_0, \vec{p}_0} \quad \text{und} \quad \mathbf{P}_0 = \left. \frac{\delta \mathbf{F}(\vec{z}, \vec{p})}{\delta \vec{p}} \right|_{\vec{z}_0, \vec{p}_0} \quad (3.14)$$

eine initiale affine Lösung $\vec{z}_{initial}$ bestimmt:

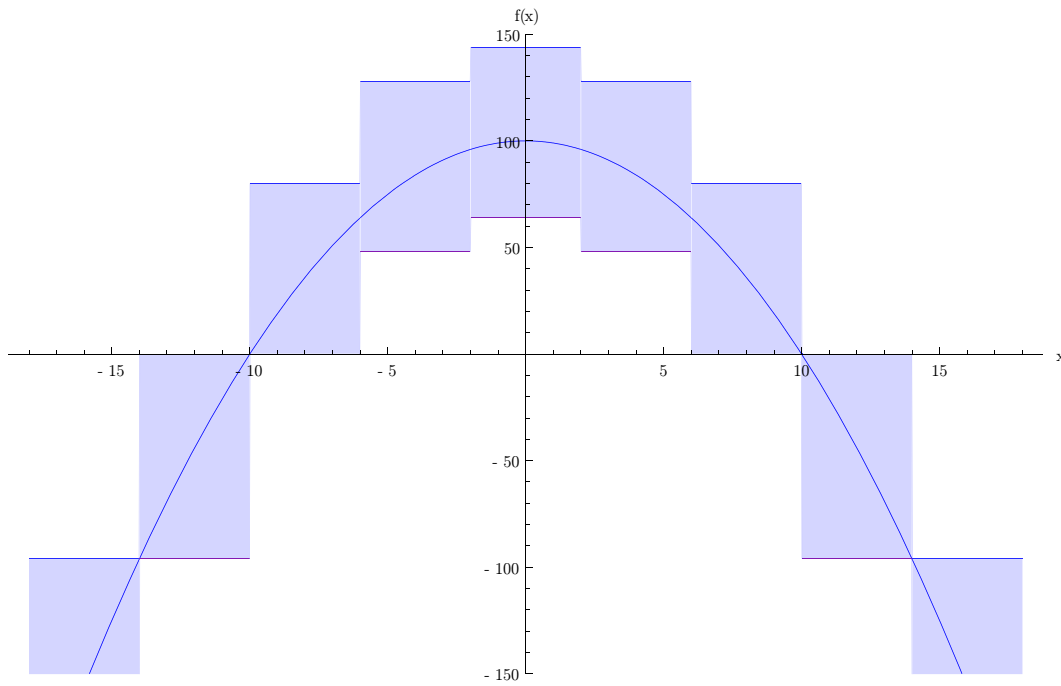
$$\mathbf{J}_0(\vec{z} - \vec{z}_0) + \mathbf{P}_0(\vec{p} - \vec{p}_0) = \vec{0} \quad (3.15)$$

$$\Rightarrow \vec{z}_{initial} = \vec{z}_0 - \mathbf{J}_0^{-1} \cdot \mathbf{P}_0 \cdot (\vec{p} - \vec{p}_0) \quad (3.16)$$

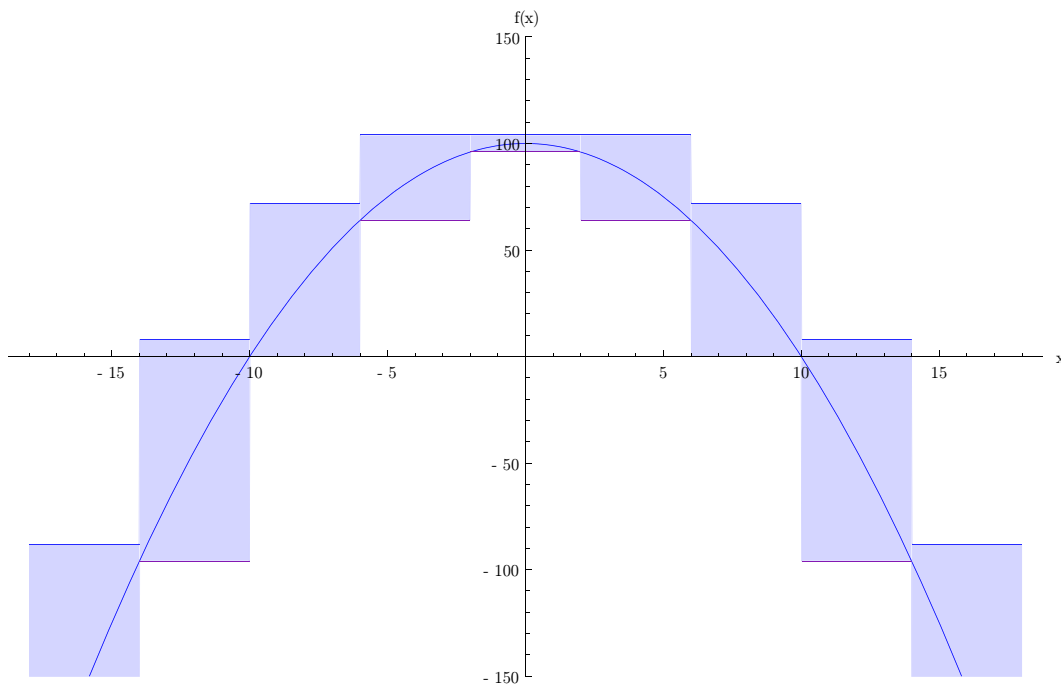
Da \mathbf{F} in der Regel nicht affine Operationen enthält, die durch die in Abschnitt 3.3.2 beschriebenen Methoden approximiert werden, garantiert die so ermittelte Lösung $\vec{z}_{initial}$ keinen sicheren Einschluss. In einem dritten Schritt wird deshalb $\vec{z}_{initial}$ um eine zusätzliche partielle Abweichung \vec{z}_{ext} ergänzt zu:

$$\vec{z} = \vec{z}_{initial} + \vec{z}_{ext} \quad (3.17)$$

In einem iterativen Verfahren wird dabei versucht, \vec{z}_{ext} möglichst optimal bezüglich einer geringen Überabschätzung zu bestimmen.



(a) Intervallarithmetik



(b) Affine Arithmetik

Abbildung 3.3: Vergleich beider Bereichsarithmetiken

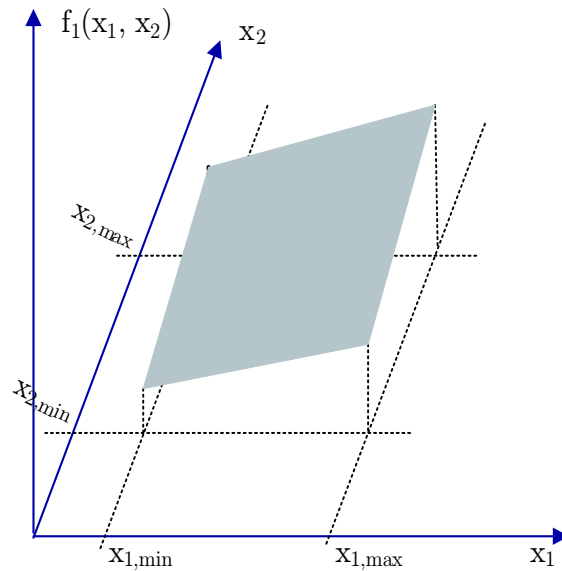
4 Stückweise lineare Modellierung mit Unsicherheiten

In Kapitel 2 wurden verschiedene Möglichkeiten der Modellierung von integrierten Schaltungen vorgestellt. Je nachdem, ob eine möglichst genaue Analyse des Schaltungsverhaltens verlangt ist oder der Schwerpunkt auf schnelle Simulationen gelegt wird, erscheinen unterschiedliche Modellbeschreibungen sinnvoll. Durch die in Kapitel 3 vorgestellte semi-symbolische Simulation besteht die Möglichkeit, mit Hilfe affiner Terme in den Modellgleichungen Unsicherheiten im Schaltungsverhalten (siehe Abschnitt 2.4) nachzubilden.

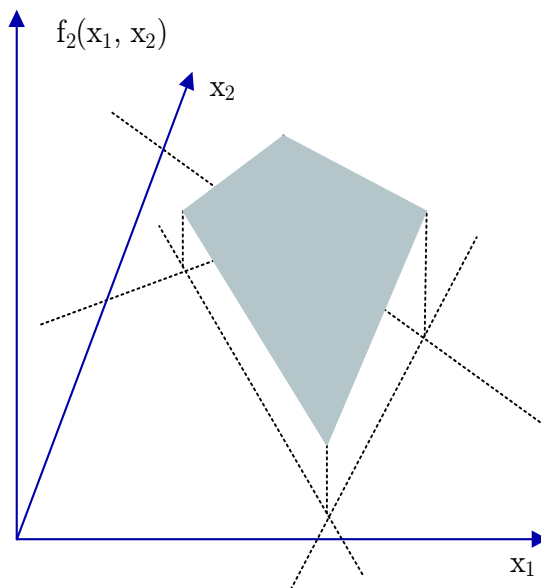
Das in Abschnitt 3.5 skizzierte Verfahren nach [GrOl08] ermöglicht es, beliebige mit Unsicherheiten belegte DAE-Systeme zu simulieren. Der in dieser Arbeit entwickelte und im Folgenden vorgestellte Ansatz legt seinen Schwerpunkt auf eine Modellierung durch stückweise affine Kurvenverläufe. Wie hierzu bereits in Abschnitt 2.2.3 dargestellt, lassen sich solche Modelle verhältnismäßig schnell simulieren. Dieser Vorteil wird jedoch durch geringere Genauigkeit und den damit verbundenen Verlust von Verlässlichkeit der Simulationsergebnisse erkaufte. Geht man jedoch davon aus, dass diese systematisch bedingten Ungenauigkeiten der Modellierung durch affine Terme eingeschlossen werden, ist eine Verlässlichkeit der Simulationsergebnisse wieder gewährleistet.

4.1 Darstellungsformen

Für die Abbildung stückweise affiner Kurvenverläufe wurden in der Vergangenheit verschiedene Darstellungsformen entwickelt. Ziel der Ansätze ist es, auch bei hoher Partitionierung, das heißt bei einer Zerlegung des Wertebereichs in viele affine Teilbereiche, eine möglichst kompakte mathematische Beschreibung zu haben. Bereits 1977 stellten Chua und Kang [ChKa77] eine erste für die Schaltungsmodellierung optimierte Darstellungsform vor. In den folgenden Jahren beschäftigten sich weitere Autoren [GüGo91] [KaCh90] mit einer Erweiterung des Ansatzes, um eine größere Schaltungsklasse abdecken zu können.



(a) Beispiel für eine kanonische Partitionierung



(b) Beispiel für eine nicht kanonische Partitionierung

Abbildung 4.1: Unterschiedliche Formen eines affinen Teilbereichs für kanonische und nicht kanonische Darstellung

Grundsätzlich ist zu unterscheiden, ob die Partitionierung des Modells achsenparallel (kanonisch) oder unabhängig von den Achsen erfolgt. Zur Verdeutlichung ist in Abbildung 4.1 für beide Darstellungen je ein affiner Teilbereich der Funktionen $f_1(\vec{x})$ und $f_2(\vec{x})$ mit $\vec{x}^T = (x_1, x_2)$ dargestellt. Die Bereichsgrenzen in Abbildung 4.1(a) können linear unabhängig voneinander durch $x_{1,min}$ und $x_{2,min}$ bzw. $x_{1,max}$ und $x_{2,max}$ angegeben werden. In Abbildung 4.1(b) sind die Grenzen jeweils eine Funktion des Eingangsvek-

tors \vec{x} .

Die oben zitierten Modelle zählen dabei zur Klasse der kanonischen stückweise linearen Modelle (CPWL-Modelle). Zur zweiten Klasse zählen beispielsweise die in dieser Arbeit verwendeten PWA-Zustandsmodelle und Modelle, die sich aus sogenannten Schanier-Hyperebenen (engl.: Hinging Hyperplanes, [Brei93]) zusammensetzen und in Abschnitt 4.6.2 detailliert vorgestellt werden. Durch die beliebige Form, die die Gebiete einnehmen können, sind im Allgemeinen bei gleicher Zahl an Gebieten bessere Approximationen des realen Schaltungsverhalten zu erwarten. Der Nachteil einer nicht kanonischen Darstellung ist, dass aus einem gegebenen Eingangsvektor nicht direkt auf das zugehörige Gebiet geschlossen werden kann. Man spricht in diesem Zusammenhang auch von einer impliziten Darstellung im Gegensatz zu der expliziten Darstellung der CPWL-Modelle.

4.2 PWA-Zustandsmodell

Diese Arbeit nutzt die PWA-Zustandsmodellldarstellung (PWA = Piecewise Affine), wie sie erstmals 1986 durch Bokhoven [Rueh86] vorgeschlagen wurde. Hierbei handelt es sich um eine nicht kanonische Modellldarstellung. Die Modellldarstellung wurde im Rahmen dieser Dissertation erweitert, so dass sich Kennlinien mit Unsicherheiten versehen lassen. Ein weiterer Aspekt für die Auswahl der PWA-Zustandsmodelle ist, dass sich – wie im Abschnitt 5.4.1 noch gezeigt wird – besondere Vorteile für eine Simulation trotz modellierter Unsicherheiten ergeben.

4.2.1 Mathematische Modellbeschreibung des PWA-Zustandsmodells

Das PWA-Zustandsmodell setzt sich aus folgenden Gleichungen zusammen:

$$PWASyst : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

$$\vec{0} = \mathbf{I}_{m \times m} \vec{y} + \mathbf{A} \vec{x} + \mathbf{B} \vec{u} + \vec{f} \quad (4.1)$$

$$\vec{j} = \mathbf{I}_{k \times k} \vec{u} + \mathbf{C} \vec{x} + \mathbf{D} \vec{y} + \vec{g} \quad (4.2)$$

$$\vec{u} \geq \vec{0}, \vec{j} \geq \vec{0}, \vec{u}^T \cdot \vec{j} = \vec{0} \quad (4.3)$$

Die Eingangsgrößen sind im Eingangsvektor \vec{x} zusammengefasst, die Ausgangsgrößen stehen in Vektor \vec{y} , der sich explizit aus Gleichung (4.1) bestimmen lässt, da $\mathbf{I}_{m \times m}$ eine Einheitsmatrix ist. Die Vektoren \vec{u} und \vec{j} bilden den internen Zustand des Modells ab. Gleichung (4.1) wird im Folgenden als affine Abbildung, Gleichung (4.2) als

Zustandsgleichung und Gleichung (4.3) als Komplementaritätsbedingung bezeichnet. Die Grenzen zwischen den einzelnen Gebieten sind nicht direkt durch die Systemgleichungen erfasst, sondern sind implizit über die Gleichungen (4.2) und (4.3) definiert.

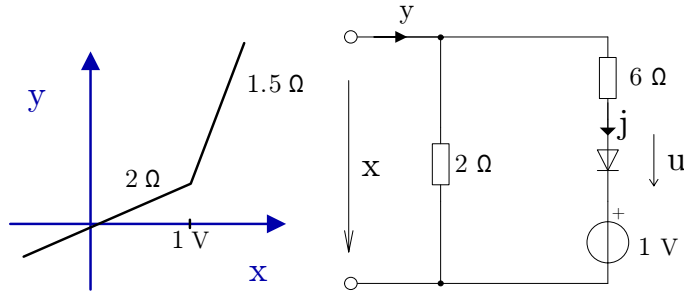


Abbildung 4.2: Ersatzschaltbild der Zustandsmodellldarstellung für den eindimensionalen Fall mit zwei affinen Segmenten (Gebieten)

Um die grundsätzliche Idee des Zustandsmodells zu veranschaulichen, ist in Abbildung 4.2 ein Ersatzschaltbild dargestellt. Für den eindimensionalen Fall ist dies möglich, wobei hier ein nichtlineares Zweitor-Element mit der Klemmenspannung als Eingangsgröße x und dem Klemmenstrom als Ausgangsgröße y (Kennlinie entsprechend linker Abbildung) modelliert wird. Das Ersatzschaltbild besteht aus linearen Widerständen (die auch negative Widerstandswerte annehmen können), einer idealen Spannungsquelle und einer idealen Diode. Durch das Anfügen weiterer Diodenzweige, ließe sich jeweils ein weiteres zusätzliches lineares Segment in der Kennlinie definieren. Da sich die Kennlinie in Abbildung 4.2 nur in zwei Abschnitte gliedert, bzw. nur eine Stützstelle bei 1 V besitzt, gibt es in der Ersatzschaltung auch nur einen einzigen Diodenzweig. Die Strom-Spannungsbeziehung der idealen Diode lässt sich formal fassen durch

$$u \geq 0, j \geq 0, u \cdot j = 0, \quad (4.4)$$

was der Komplementaritätsbedingung aus Gleichung (4.3) entspricht.

Abhängig davon, ob die Diode leitet oder nicht, erhält man für das Gesamtsystem bezüglich der Eingangsgröße x ein unterschiedliches lineares System. Ändert sich der Zustand der Diode, wird im Folgenden auch von einem Zustandswechsel des Gesamtsystems gesprochen.

Der Strom y aus Abbildung 4.2 lässt sich bei leitender Diode, also bei $u = 0\text{ V}$, berechnen durch

$$y = \frac{1}{2} \cdot x + \frac{1}{6}(x - 1) = \frac{2}{3} \cdot x - \frac{1}{6}. \quad (4.5)$$

Übertragen auf Gleichung (4.1) bestimmen sich so die Systemmatrizen zu

$$\mathbf{A} = \left(-\frac{1}{2}\right), \quad \vec{f} = \left(-\frac{1}{6}\right), \quad \mathbf{B} = \left(\frac{1}{6}\right),$$

wobei für die Matrix \mathbf{B} der Leitwert des Diodenzweigs herangezogen wird.

Die Zustandsgleichung (4.2) berechnet sich aus dem Diodenzweig der Ersatzschaltung mit

$$j = -1 \cdot x + 1 \cdot u + 1, \quad (4.6)$$

so dass sich weiterhin ergibt:

$$\mathbf{C} = (-1), \quad \mathbf{D} = (0), \quad \vec{g} = (1).$$

Aufgrund der Modelldefinition mit der separat definierten Komplementaritätsbedingung ist es möglich, das Verhalten in einem Zustand durch zwei affine Systemgleichungen (der Affinen Abbildung und der Zustandsgleichung) darzustellen. Ihre Auswertung kann sehr effizient durchgeführt werden, sofern alle Diodenzweige leiten:

$$\vec{y} = -\mathbf{A}\vec{x} - \vec{f} \quad (4.7)$$

Durch die beiden affinen Gleichungen wird der Einsatz von affinen Formen in den Koeffizientenmatrizen und Variablen stark vereinfacht. Somit eignet sich die PWA-Zustandsmodellldarstellung sehr gut für die in Kapitel 3 vorgestellte semi-symbolische Simulation.

Ziel während der Simulation ist es, die Koeffizientenmatrizen bezüglich eines Eingangsvektors \vec{x} in der Form zu bestimmen, dass die Komplementaritätsbedingung bei $\vec{u} = \vec{0}$ erfüllt bleibt. Um die Gültigkeit der Komplementaritätsbedingung zu überprüfen, wird der Vektor \vec{j} mit Hilfe der Gleichung (4.2) ermittelt. Es muss gelten:

$$j_i \geq 0, \forall i \in \{1, 2, \dots, k\} \quad \text{mit} \quad \vec{u} = \vec{0}. \quad (4.8)$$

Ist diese Bedingung verletzt, muss ein neuer gültiger Zustand ermittelt werden. Für das Beispiel aus Abbildung 4.2 ergibt sich folgendes neues System¹, wenn $x < 1$ V gegeben ist:

$$y = \frac{1}{2} \cdot x + \frac{1}{6} \cdot u \quad (4.9)$$

$$j = -1 \cdot x + 1 \cdot u + 1 \quad (4.10)$$

Die neuen Systemmatrizen auf eine möglichst effiziente Weise zu ermitteln, stellt das Hauptproblem während einer Simulation dar. Das entsprechende Vorgehen ist in Kapitel 5 beschrieben.

¹Das Ersatzschaltbild verliert seine Gültigkeit.

4.2.2 Berücksichtigung von Unsicherheiten

Um Unsicherheiten zu modellieren, lassen sich die Einträge in den Matrizen \mathbf{A} und \mathbf{B} und dem Vektor \vec{f} um Abweichungen in affiner Form erweitern. Das entsprechende System wird damit definiert durch:

$$PWASyst : \mathbb{R}^n \rightarrow \mathbb{R}^m,$$

$$\vec{0} = \mathbf{I}_{m \times m} \vec{y} + \hat{\mathbf{A}} \vec{x} + \hat{\mathbf{B}} \vec{u} + \vec{f}, \quad (4.11)$$

$$\vec{j} = \mathbf{I}_{k \times k} \vec{u} + \mathbf{C} \vec{x} + \mathbf{D} \vec{y} + \vec{g}, \quad (4.12)$$

$$\vec{u} \geq \vec{0}, \vec{j} \geq \vec{0}, \vec{u}^T \cdot \vec{j} = \vec{0} \quad (4.13)$$

Anhand eines Beispiels, in dem eine Kennlinie eines nichtlinearen Widerstandes gemäß Abbildung 4.3 modelliert werden soll, wird verdeutlicht, welchen Einfluss affine Formen in den Systemmatrizen haben. Dargestellt ist der Strom durch den Widerstand

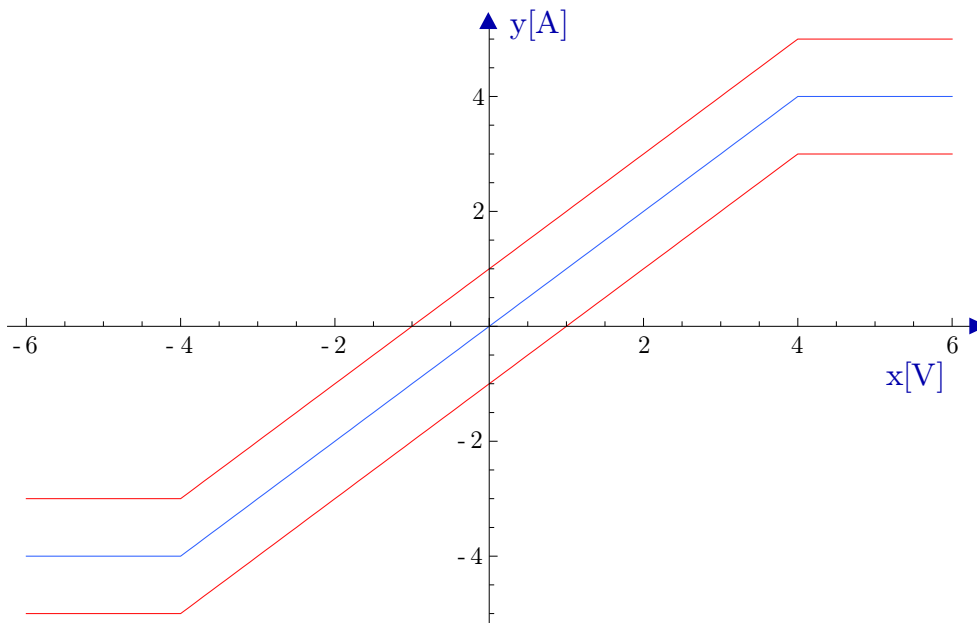


Abbildung 4.3: Verlauf mit konstanter Aufweitung

in Abhängigkeit von der Eingangsspannung. Entsprechend der Variablenbezeichnung für das PWA-Zustandsmodell wird die Eingangsspannung mit x und der Ausgangsstrom mit y bezeichnet. Die Kennlinie besitzt eine konstante Abweichung vom Nominalwert des Ausgangsstroms von $di = 1 \text{ mA}$.

Für die Kennlinie lassen sich folgende Systemmatrizen bestimmen:

$$\mathbf{A} = \begin{pmatrix} 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -1 & 1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} -1 \\ -1 \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} 4 + di \cdot \varepsilon_1 \end{pmatrix}, \quad \vec{g} = \begin{pmatrix} -4 \\ 4 \end{pmatrix}$$

Zu erkennen ist hier eine konstante Aufweitung von y , die durch die Unsicherheit im Vektor \vec{f} bewirkt wird.

Unsicherheiten können ebenfalls in den Matrizen \mathbf{A} und \mathbf{B} eingefügt werden. Hier beeinflussen sie die Richtungsableitung bezüglich des Eingangsvektors x . Wählt man die Systemmatrizen wie folgt, ergibt sich eine Kennlinie entsprechend Abbildung 4.4:

$$\mathbf{A} = \begin{pmatrix} 0 \end{pmatrix}, \quad \hat{\mathbf{B}} = \begin{pmatrix} -1 - \frac{1}{4} \cdot \varepsilon_1 & 1 + \frac{1}{4} \cdot \varepsilon_1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} -1 \\ -1 \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} 4 + 1 \cdot \varepsilon_1 + 0.4 \cdot \varepsilon_2 \end{pmatrix}, \quad \vec{g} = \begin{pmatrix} -4 \\ 4 \end{pmatrix}$$

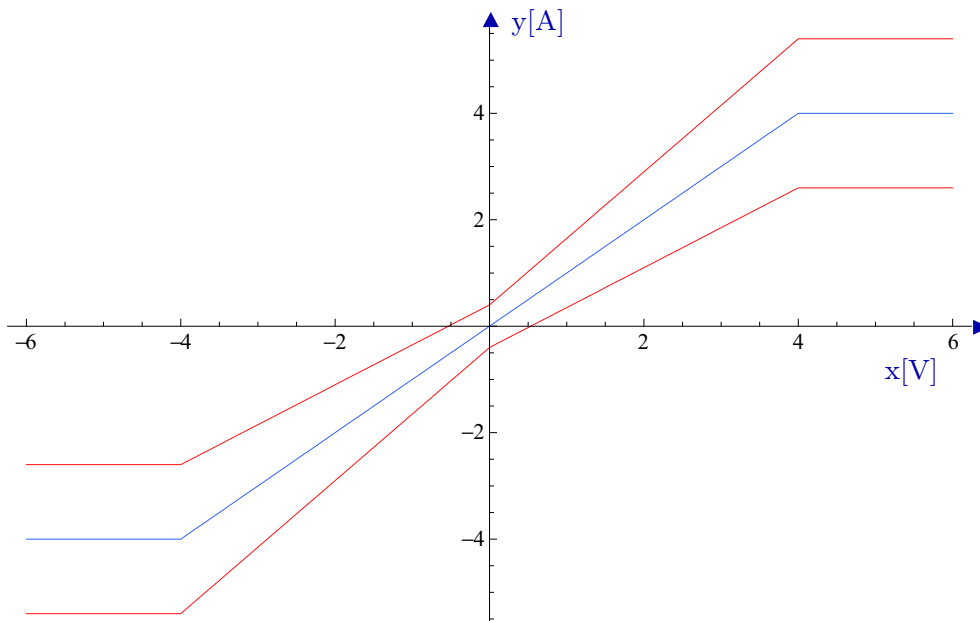


Abbildung 4.4: Unsicherheit mit Abschnürung

Durch ε_1 stehen die Größen in $\hat{\mathbf{B}}$ und \vec{f} zueinander in Korrelation. Die durch \vec{f} verursachte konstante Aufweitung kann somit abhängig von \vec{x} erhöht bzw. verringert werden. Durch gezielte Wahl der Koeffizienten ist es möglich, den abgebildeten Kurvenverlauf, der ein Abschnüren der Unsicherheit bei $x = 0$ aufweist, zu konstruieren.

Der Ausgangsvektor lässt sich trotz vorhandener Unsicherheiten sehr effizient lösen. Ist der Eingangsvektor selbst nicht mit affinen Formen besetzt, werden ausschließlich affine Operationen ausgeführt und es entstehen keinerlei zusätzliche Abweichungsterme (Aufweitungen) im Ergebnisintervall.

Ist auch der Eingangsvektor $\vec{\hat{x}}$ mit affinen Formen besetzt, d.h. die Eingangserregung des Systems ist mit Unsicherheiten behaftet, ergeben sich nicht affine Operationen und somit eine Überabschätzung.

4.2.3 Approximationen an den Übergängen

Sofern für den gegebenen Eingangsvektor $\vec{\hat{x}}$ die Komplementaritätsbedingung mit $\vec{j}_0 \pm \text{rad}(\vec{j})$ erfüllt ist, ist es ohne Belang, ob die Matrizen oder Vektoren in Gleichung (4.11) mit affinen Formen besetzt sind oder nicht. Mit der Verwendung der affinen Arithmetik erfolgt die semi-symbolische Berechnung von $\vec{\hat{y}}$ und die damit verbundene Skalarmultiplikation analog zur Berechnung ohne affine Formen.

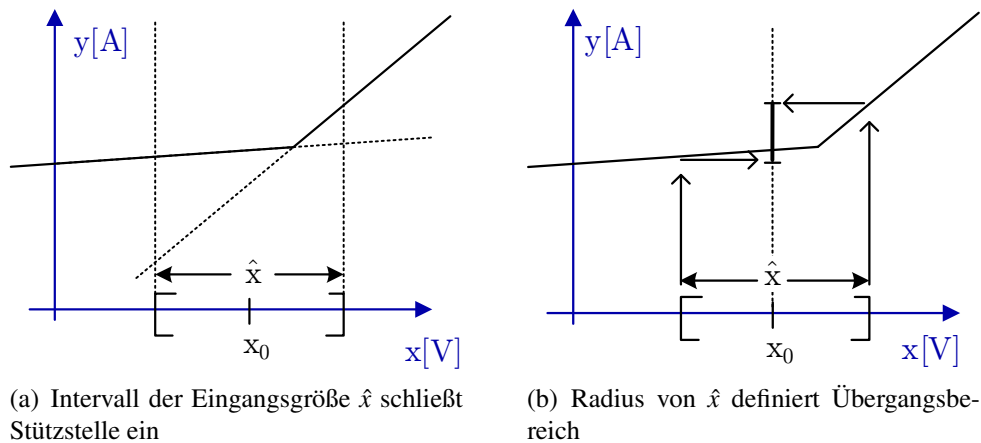


Abbildung 4.5: Beide Abschnitte der Kennlinie haben Einfluss auf \hat{y}

Eine besondere Betrachtung wird jedoch an den Übergängen zwischen zwei Gebieten nötig. Hier ist es möglich, dass der Eingangsvektor einen Bereich aufspannt, der mehr als ein affines Teilgebiet des Modells einschließt. Folglich sind mehrere Systembeschreibungen mit unterschiedlichen Koeffizientenmatrizen in die Auswertung einzubeziehen. Zur Verdeutlichung sei wiederum der eindimensionale Fall betrachtet. Wie aus Abbildung 4.5(a) ersichtlich, sind bei gegebenem \hat{x} sowohl die rechte, als auch die linke Seite der Kennlinie zur Bestimmung von \hat{y} zu berücksichtigen. In Abbildung 4.5(b) ist skizziert, wie sich aus der maximalen Abweichung $x_0 \pm \text{rad}(\hat{x})$ die Grenzen für \hat{y} ergeben. Durchläuft man die Kennlinie mit variablem x_0 und konstanter Aufweitung, erhält man einen Verlauf entsprechend Abbildung 4.6. Bezüglich der Nominalkurve besteht jedoch keine Symmetrie mehr zu den ermittelten Grenzen

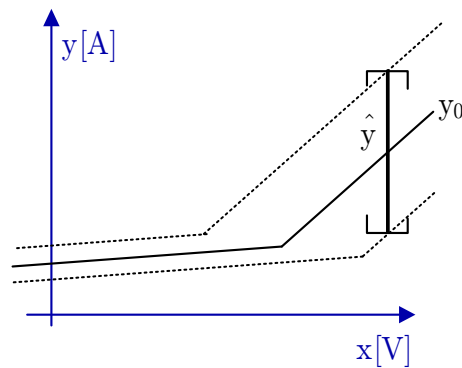
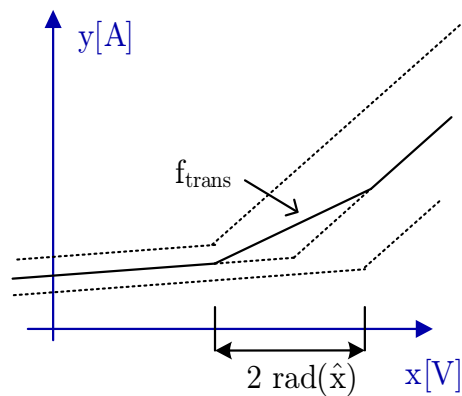
Abbildung 4.6: Resultierende Ausgangskennlinie $\hat{y}(\hat{x})$ 

Abbildung 4.7: Nominalpunktverschiebung in der Transitionsfunktion

(gestrichelte Linie). Das Ergebnis ließe sich folglich nicht durch eine affine Form darstellen. Ebenfalls gingen Korrelationsinformationen aus \hat{x} verloren.

Als Lösungsansatz wird eine Transitionsfunktion

$$\hat{y} = f_{trans}(PWAS_{\text{yst}1}, PWAS_{\text{yst}2}, x_0, \text{rad}(\hat{x})) \quad (4.14)$$

bestimmt, die einen Verlauf entsprechend Abbildung 4.7 aufweist. Die Steigung der Transitionsfunktion ist hierbei der arithmetische Mittelwert aus den beiden Teilsteigungen. Der Nominalwert ist um einen konstanten Anteil erhöht, der sich aus der Gesamtbreite der Eingangsgröße ($2 \cdot \text{rad}(\hat{x})$) ergibt. Ausgehend von den beiden Randpunkten der Transitionsfunktion werden die beiden Geradenstücke durch eine affine Funktion ineinander überführt. Die Abweichungsterme ändern sich dabei linear, so dass nicht nur bezüglich der Nominalkurve, sondern auch bezüglich der Abweichungen keine Unstetigkeiten auftreten.

Diese Betrachtung lässt sich auf den mehrdimensionalen Fall erweitern. Übertragen

auf die Nomenklatur des PWA-Modells ist \vec{y} im Übergang definiert durch:

$$\vec{y} = -((\mathbf{A}_{neu} \cdot \vec{x}_0 + \vec{f}_{neu} + 0.5(\hat{\mathbf{A}}_2 \cdot \vec{x}_r - \hat{\mathbf{A}}_1 \cdot \vec{x}_r)) + ((\hat{\mathbf{A}}_2 + \hat{\mathbf{A}}_1)/2 - 0.5\vec{\Phi}(\hat{\mathbf{A}}_2 - \hat{\mathbf{A}}_1)) \cdot (\hat{x} - \vec{x}_0)). \quad (4.15)$$

mit

$$\hat{\mathbf{A}}_{neu} = (\hat{\mathbf{A}}_1 + \hat{\mathbf{A}}_2)/2 \quad (4.16)$$

$$\hat{\mathbf{f}}_{neu} = (\hat{\mathbf{f}}_1 + \hat{\mathbf{f}}_2)/2 \quad (4.17)$$

$$\vec{x}_r = \text{rad}(\vec{\hat{x}}) \quad (4.18)$$

Der Vektor $\vec{\Phi} \in \mathbb{R}^m$ gibt, wie in Abbildung 4.8 dargestellt, die Distanz zu dem eingeschlossenen Stützpunkt an.

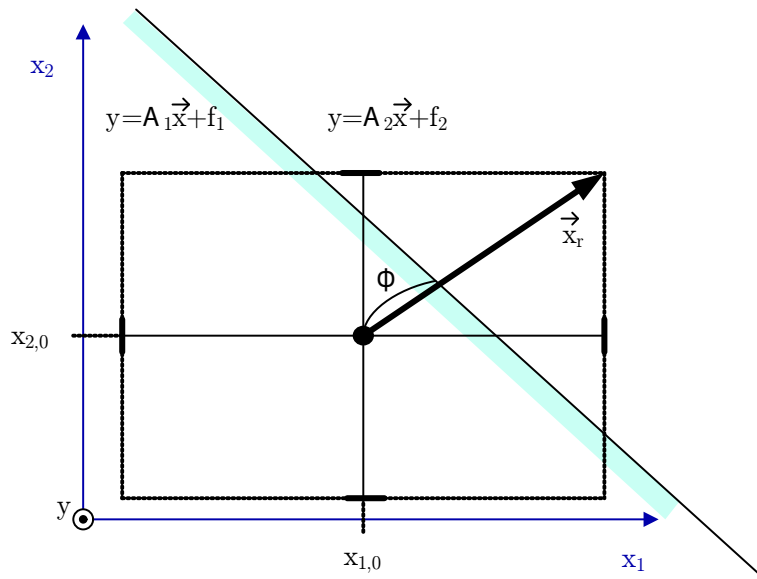


Abbildung 4.8: Projektion auf die zweidimensionale Ebene der Eingangsgrößen

4.3 Modellerzeugung aus Kennlinienform

Ist die gewünschte Kennlinie für die zu modellierende Schaltung bekannt, d.h. sind die Steigungen, Stützstellen und Schnittpunkte mit der y-Achse gegeben, lässt sich im eindimensionalen Fall ein entsprechendes PWA-Zustandsmodell herleiten.

Mit $\hat{a}_1, \hat{a}_2, \dots, \hat{a}_{k+1}$ für die Steigungen der Geradenstücke, x_1, x_2, \dots, x_k für die Positionen der Stützstellen und b als y-Achsenabschnitt des ersten Geradenabschnitts (siehe

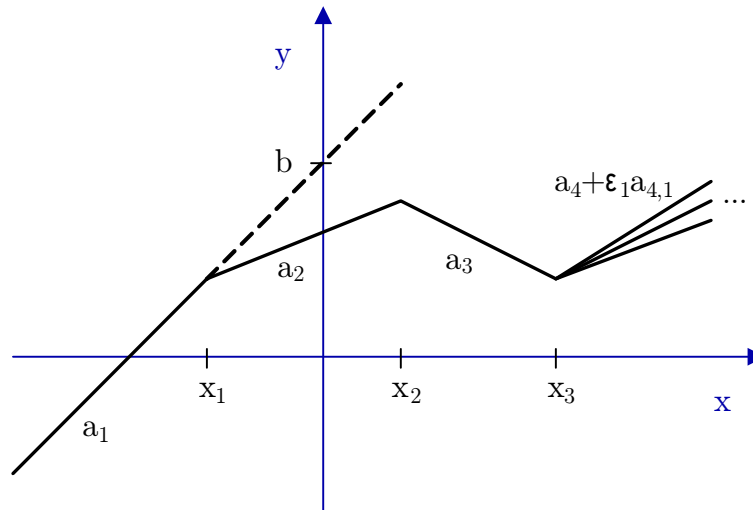


Abbildung 4.9: Erzeugung eines PWA-Zustandsmodells aus Steigung, Stützstellen und y-Achsenabschnitt

Abbildung 4.9) ergibt sich ein PWA-Zustandsmodell mit den Systemmatrizen

$$\mathbf{A} = (-a_1), \quad \mathbf{B} = (a_1 - a_2 \quad a_2 - a_3 \quad \cdots \quad a_k - a_{k+1}), \quad \mathbf{C} = \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \vec{f} = (-b), \quad \vec{g} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix}.$$

Unsicherheiten, beispielsweise in der Steigung einzelner Segmente (siehe Abbildung 4.9), gehen dabei direkt in die Matrizen $\hat{\mathbf{A}}$ und $\hat{\mathbf{B}}$ ein. In dem derart berechneten Modell ist die Komplementaritätsbedingung für $x < x_1$ erfüllt.

Für Systeme mit mehreren Eingängen ist eine Herleitung eines PWA-Zustandsmodells aus einer Kennlinienform nur schwer möglich. Vor allem wenn die Wechsel zwischen den affinen Teilgebieten nicht achsenparallel sind (nicht kanonisch), ist zwischen den Koeffizienten des Modells und den Kenngrößen des Systems kein Zusammenhang mehr ersichtlich. In den Abschnitten 4.4, 4.5 und 4.6 werden drei Möglichkeiten vorgestellt, mit denen eine automatische Generierung von Modellen aus beliebigen Kennlinien erfolgen kann.

4.4 Zusammenschaltung von PWA-Modellen in Netzlisten

Als Grundlage zur Erzeugung eines PWA-Zustandsmodells dient in diesem Abschnitt eine algebraische Schaltungsbeschreibung, wie sie in Kapitel 3.2 vorgestellt wurde. Ausgangspunkt ist eine Netzliste, die sowohl lineare Bauelemente als auch nichtlineare Bauelemente mit bekannten PWA-Zustandsmodellierungen enthalten kann. Für eine Modellierung komplexerer Schaltungsfunktionalität (siehe Kapitel 2.2.3) werden im Folgenden sechs Basiselemente zur Verfügung gestellt (siehe auch *Sur-P.A.S.S.* PWA-Netlisting Paket, Abschnitt 5.3.4):

- Stromgesteuerte Stromquelle (CCC)
- Stromgesteuerte Spannungsquelle (CCV)
- Spannungsgesteuerte Stromquelle (VCC)
- Spannungsgesteuerte Spannungsquelle (VCV)
- PWA-Widerstand (Zweipol)
- PWA-Konduktanz (Zweipol)

Jedes dieser Basiselemente besitzt eine eindimensionale funktionale Zuordnung

$$PWASyst : \mathbb{R} \rightarrow \mathbb{R},$$

so dass sich die Systemmatrizen abhängig von den gegebenen Kennlinien mit der in Abschnitt 4.3 vorgestellten Methode bestimmen lassen. Sofern das PWA-Modell bekannt ist, sind auch Bauelemente mit mehreren Klemmen, wie beispielsweise Transistoren oder Transistorpaarungen, einsetzbar.

Betrachtet man bei κ PWA-Bauelementen zunächst nur deren Ein- und Ausgangsgrößen ($\vec{y}_{NL,1}, \dots, \vec{y}_{NL,\kappa}$ und $\vec{x}_{NL,1}, \dots, \vec{x}_{NL,\kappa}$) erhält man mit Hilfe der MNA ein lineares Gleichungssystem. Berücksichtigt man zusätzlich die Systemmatrizen A , B und f aus Gleichung (4.11) für die affine Abbildung, lässt sich folgende erweiterte Koeffizientenmatrix aufstellen:

\vec{y}_{neu}	$\vec{y}_{NL,1}$...	$\vec{y}_{NL,\kappa}$	$\vec{x}_{NL,1}$...	$\vec{x}_{NL,\kappa}$	\vec{u}_1	...	\vec{u}_κ	\vec{x}_{neu}	
0	$\begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$		0	\mathbf{A}_1		0	\mathbf{B}_1		0	0	\vec{f}_1
		\ddots			\ddots			\ddots			\vdots
0	0		$\begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$	0		\mathbf{A}_κ	0		\mathbf{B}_κ	0	\vec{f}_κ
Gleichungen aus Netzliste							0	...	0	Gl.	0

\vec{x}_{neu} repräsentiert dabei alle Eingangsgrößen der Gesamtschaltung und \vec{y}_{neu} alle Ausgangsgrößen. Zur Gewinnung der Systemmatrizen, die das PWA-Modell für die Gesamtschaltung bilden, wird die erweiterte Koeffizientenmatrix in die reduzierte Stufenform gebracht. Die Einträge für alle $\vec{x}_{NL,i}$ und $\vec{y}_{NL,i}$ werden dabei eliminiert. \mathbf{A}_{neu} , \mathbf{B}_{neu} , \vec{f}_{neu} finden sich durch Koeffizientenvergleich in der reduzierten Form wie folgt wieder:

\vec{y}_{neu}	$\vec{y}_{NL,1}$...	$\vec{y}_{NL,\kappa}$	$\vec{x}_{NL,1}$...	$\vec{x}_{NL,\kappa}$	\vec{u}_1	...	\vec{u}_κ	\vec{x}_{neu}
$\begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$	0		0	0		0	\mathbf{B}_{neu}		\mathbf{A}_{neu}	\vec{f}_{neu}

Für die Zustandsgleichungen (4.12), also für die Bestimmung der Matrizen \mathbf{D}_{neu} , \mathbf{C}_{neu} und \vec{g}_{neu} , ist das Vorgehen vergleichbar. Mit einer angepassten Spaltenanordnung sieht die Koeffizientenmatrix wie folgt aus:

\vec{u}_1	...	\vec{u}_κ	$\vec{x}_{NL,1}$...	$\vec{x}_{NL,\kappa}$	$\vec{y}_{NL,1}$...	$\vec{y}_{NL,\kappa}$	\vec{y}_{neu}	\vec{x}_{neu}	
$\begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$		0	\mathbf{C}_1		0	\mathbf{D}_1		0	0	0	\vec{g}_1
	\ddots			\ddots			\ddots				\vdots
0		$\begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$	0		\mathbf{C}_1	0		\mathbf{D}_1	0	0	\vec{g}_κ
0		0	Gleichungen aus Netzliste								

Auch hier werden alle Einträge für $\vec{x}_{NL,i}$ und $\vec{y}_{NL,i}$ eliminiert. Entsprechend der Vorgaben für die Zustandsgleichung steht vor dem Zustandsvektor \vec{u} eine Einheitsmatrix:

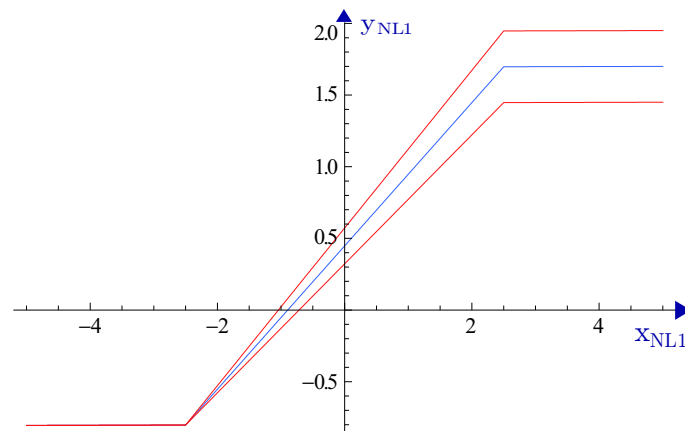
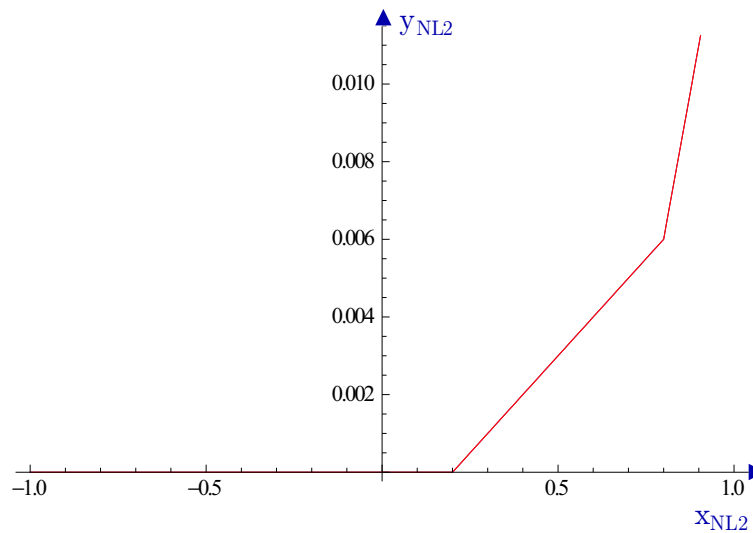
(a) Kennlinie für $PWLSyst_1$ (Widerstand)(b) Kennlinie für $PWLSyst_2$ (VCV)

Abbildung 4.11: PWA-Kennlinien für eingesetzte nichtlineare Bauelemente

chungen:

$$0 = i_{ein} + \frac{R_1 \cdot \mathbf{A}^{(a)} \cdot \mathbf{A}^{(b)}}{R_1 \cdot \mathbf{A}^{(b)} - 1} u_{ein} + \begin{pmatrix} \mathbf{B}_1^{(a)} & \mathbf{B}_2^{(a)} & \frac{R_1 \mathbf{A}^{(a)} \mathbf{B}_1^{(b)}}{R_1 \mathbf{A}^{(b)} - 1} & \frac{R_1 \mathbf{A}^{(a)} \mathbf{B}_2^{(b)}}{R_1 \mathbf{A}^{(b)} - 1} \end{pmatrix} \begin{pmatrix} \vec{u}_{NL,1} \\ \vec{u}_{NL,2} \end{pmatrix} + \left(f^{(a)} + \frac{R_1 \mathbf{A}^{(a)} f^{(b)}}{R_1 \mathbf{A}^{(b)} - 1} \right)$$

Wendet man die der Abbildung 4.11 entsprechenden Koeffizienten auf die Systemmatrizen an und setzt $R_1 = 100 \, \Omega$, ergibt sich eine resultierende Kennlinie entsprechend Abbildung 4.12.

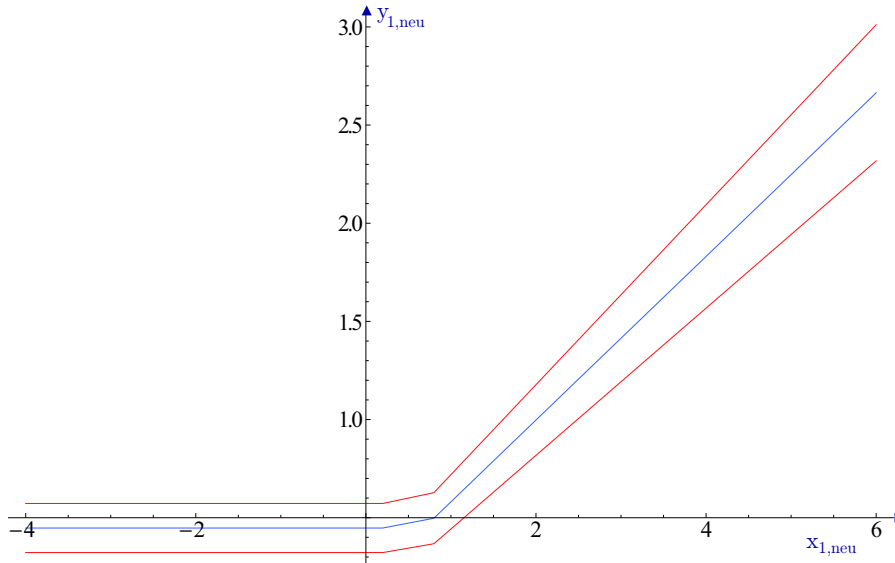


Abbildung 4.12: Resultierende Kennlinie $y_{neu,1}(x_{neu,1})$

4.5 Ableitung aus funktionaler Beschreibung

Liegt für die zu modellierende Schaltung bereits eine algebraische Beschreibung in expliziter Form vor, ist es möglich, das Modell direkt aus der nichtlinearen Gleichung zu konstruieren. Je nach Beschaffenheit der Funktion und der Anzahl der Eingangsvariablen gibt es zahlreiche Ansätze zur Approximation ([Cant71],[HuHo98],[ImIr86]), die sich jedoch nur für eine kleine Zahl von Eingangsvariablen eignen. Für die Modellierung von Schaltungs- oder Bauelementverhalten benötigt man jedoch oftmals mehr als zwei Eingangsgrößen.

Eine Möglichkeit der Realisierung führt über die Separation der Funktion in mehrere Funktionen, die jeweils nur von einer Variablen abhängen.

Nach dem Theorem von Kolmogorov [Kolm63] ist es möglich, jede beliebige Funktion, die von mehreren Veränderlichen abhängt, als Superposition von Funktionen, die nur von einer Veränderlichen abhängen, darzustellen. Liegt also eine Beschreibung für $y_0 = f(x_1, \dots, x_t)$ vor, kann man diese umformen zu:

$$y_n = \sum_{i=n+1}^s g_{n,i}(y_i) + \sum_{i=1}^t f_{n,i}(x_i) \quad \text{mit } n = \{0 \dots s\}, g_{s,s+1} = 0. \quad (4.19)$$

Beispielsweise lässt sich so die Funktion $f(x_1, x_2) = x_1 \cdot x_2$ durch

$$\begin{aligned} f(x_1, x_2) &= y_0 \\ y_0 &= \frac{y_1^2}{2} - \frac{x_1^2}{2} - \frac{x_2^2}{2} \\ y_1 &= x_1 + x_2 \end{aligned}$$

darstellen.

Für jede einzelne Funktion $f_{n,i}(x_i)$ und $g_{n,i}(y_i)$ kann nun im Gegensatz zu $f(x_1, \dots, x_t)$ eine stückweise affine Beschreibung gefunden werden, wie im Folgenden anhand eines Algorithmus für $f_{n,i}(x_i)$ gezeigt wird.

4.5.1 Rasterbasierter Ansatz mit konstantem Fehler

Als Grundlage für diesen Ansatz wird der zu betrachtende Eingangsbereich für x_i diskretisiert und die Funktionsergebnisse werden an den diskretisierten Punkten $x_i^{(k)}$ ermittelt, so dass sich eine Punktmenge $P_i = (x_i^{(k)}, f(x_i^{(k)}))$ mit $k = \{1, \dots, m\}$ ergibt.

Der maximale Fehler, um den eine Gerade mit der Funktion $f_{lin}(x) = \alpha * x + \beta$ von allen P_i abweicht, lässt sich über die Unendlich-Norm (auch Maximum-Norm genannt) definieren:

$$\delta(m) := \|x_i^{(k)} - f_{lin}(x_i^{(k)})\|_\infty = \max_{k=1}^m (|x_i^{(k)} - f_{lin}(x_i^{(k)})|) \quad (4.20)$$

Gesucht sind die Parameter α und β der linearen Funktion $f_{lin}(x)$, für die $\delta(m)$ am kleinsten ist. Aus dem Ergebnis lässt sich anschließend

$$\hat{f}_{lin}(x) = \alpha x + \delta(m)\epsilon_{NL} + \beta \quad (4.21)$$

herleiten.

Um eine stückweise lineare Beschreibung zu finden, bei der je Segment der Fehler δ kleiner als ein vorgegebener Fehler δ_{max} ist, lässt sich folgender Algorithmus verwenden:

Algorithmus 1 : Finde PWL-Funktion

Eingabe : P_i, m, δ_{max}

solange $m > 0$ **tue**

wiederhole

 Finde Minimum für $\delta(n)$;

wenn $\delta(n) \leq \delta_{max}$ **dann**

 | $\alpha_{neu} = \alpha, \beta_{neu} = \beta$

Ende

bis $\delta(n) \geq \delta_{max}$ **oder** $n == m$;

 Neues PWL-Segment ($\alpha_{neu} * x + \delta_{max}\epsilon_{NL} + \beta_{neu}$);

$P_i = \{(x_i^{(k)}, f(x_i^{(k)})) | k = \{n-1, n, \dots, m\}\}$;

$n = 1$;

$m = |P_i|$

Ende

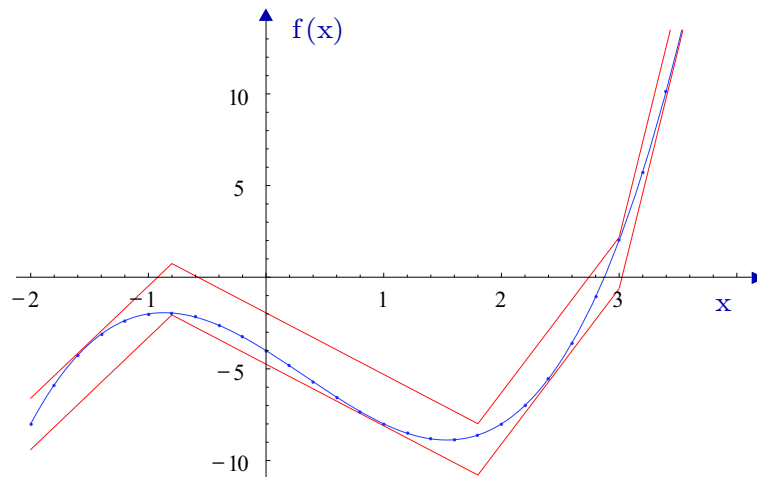


Abbildung 4.13: PWL-Funktion mit absolutem Fehler

In Abbildung 4.13 ist exemplarisch das Ergebnis von Algorithmus 1 für die Funktion $f(x) = x^3 - (x+2)^2$ mit einem Eingangsbereich von $-2 \leq x \leq 4$ dargestellt.

Aus den so ermittelten linearen Funktionen lässt sich das Zustandsmodell berechnen. Dabei findet die in Abschnitt 4.3 vorgestellte Berechnungsvorschrift Verwendung.

4.5.2 PWA-Zustandsmodell aus separiertem Term

Um das PWA-Zustandsmodell für alle y_n aus Gleichung (4.19) zu erzeugen, muss die entsprechende Summation der zuvor ermittelten Zustandsmodelle vorgenommen werden. Bei einer Summation von $k = s + t$ PWA-Zustandsmodellen dieser Art mit jeweils unterschiedlichen Eingangsgrößen y_1, y_2, \dots, y_s und x_1, x_2, \dots, x_t bilden sich die Systemmatrizen des neuen Zustandsmodells $PWASyst_i$ aus den einzelnen Systemmatrizen wie folgt:

$$\begin{aligned} \mathbf{A}_{neu} &= \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_k \end{pmatrix}, & \mathbf{B}_{neu} &= \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_k \end{pmatrix}, \\ \mathbf{C}_{neu} &= \begin{pmatrix} \mathbf{C}_1 & & & \\ & \mathbf{C}_2 & & \\ & & \ddots & \\ & & & \mathbf{C}_k \end{pmatrix}, & \mathbf{D}_{neu} &= 0, \\ f_{neu} &= f_1 + f_2 + \cdots + f_k, & \vec{g}_{neu}^T &= \begin{pmatrix} \vec{g}_1^T & \vec{g}_2^T & \cdots & \vec{g}_k^T \end{pmatrix}. \end{aligned}$$

4.5.3 Konstruktives Verfahren zur Separation affiner Abschnitte

In [Yama96] wurde ein konstruktives Verfahren vorgestellt, mit dem sich eine Separation automatisiert durchführen lässt. Dieses Verfahren findet im weiteren Verlauf Verwendung und soll deshalb zunächst kurz erläutert werden. Allgemein wird dabei davon ausgegangen, dass sich die zu untersuchende Funktion aus vier arithmetischen Operationen (+, -, *, /), unären Operationen (sin(.), exp(.), log(.), sqrt(.), etc.) sowie der Potenzfunktion (^) zusammensetzt. Im Folgenden wird aus der Funktion eine Baumstruktur entsprechend der mathematischen Auswertereihenfolge erzeugt.

Dieser Baum wird nun nach separierbaren und nicht separierbaren Zweigen durchsucht. Die Knoten werden markiert. Über einen Regelsatz werden die nicht separierbaren Äste im Weiteren durch das Einführen einer neuen Hilfsvariablen y_{s+1} in separierbare Äste transformiert. Die Transformationen hängen dabei von den Operatoren ab und sind in den Abbildungen 4.14 bis 4.16 dargestellt.

Als Beispiel ist in Abbildung 4.17 ein Baum zur Auswertereihenfolge für die Funktion $f(x_1, x_2) = \frac{(x_1+1)^3}{x_2} + \sin(x_1 + (3x_2))$ dargestellt.

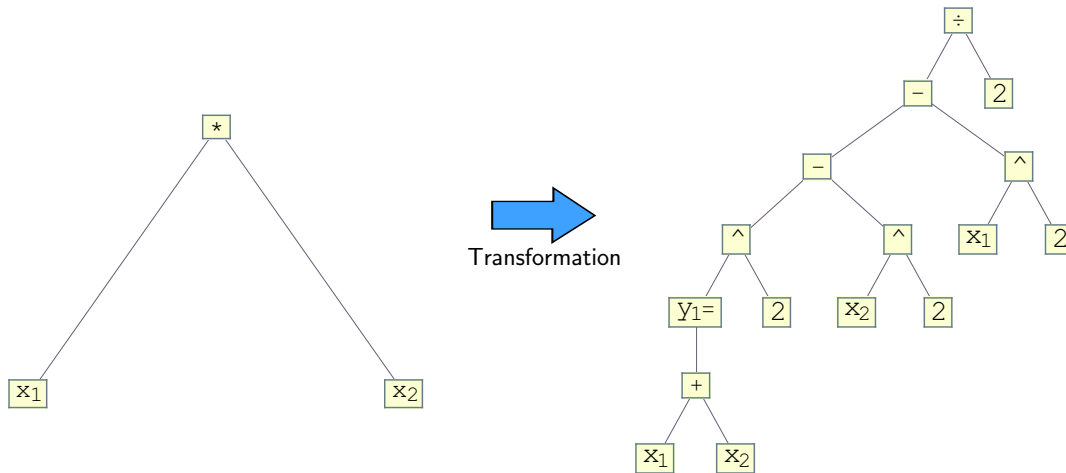


Abbildung 4.14: Transformation für nicht separierbare Multiplikation

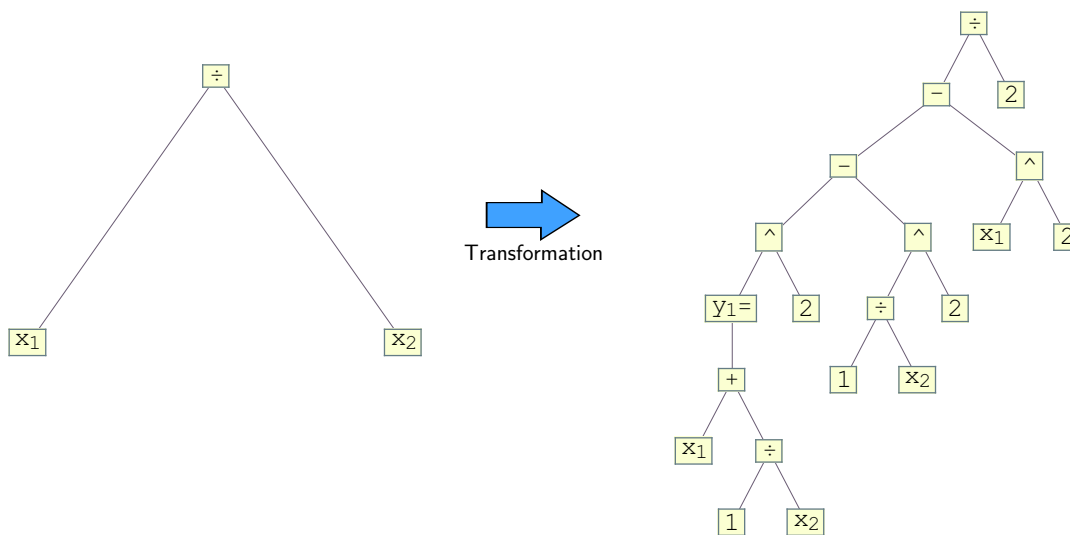


Abbildung 4.15: Transformation für nicht separierbare Division

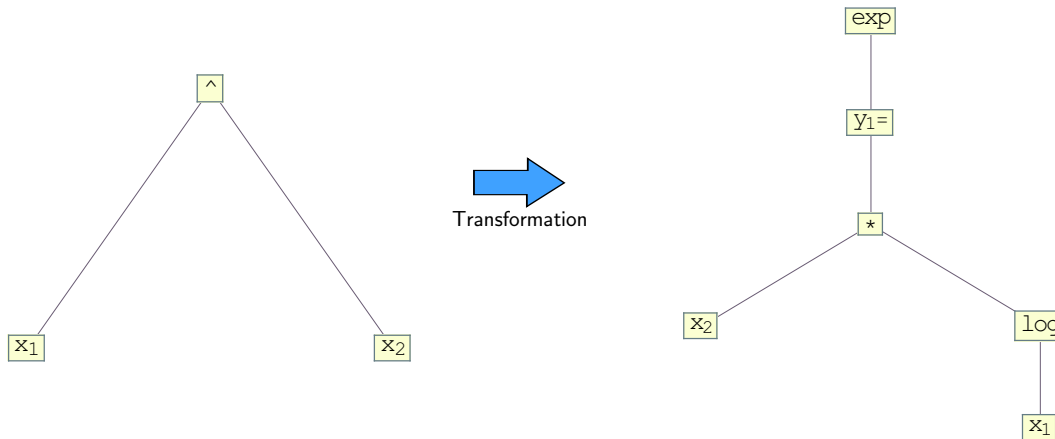


Abbildung 4.16: Transformation für nicht separierbare Potenzfunktion

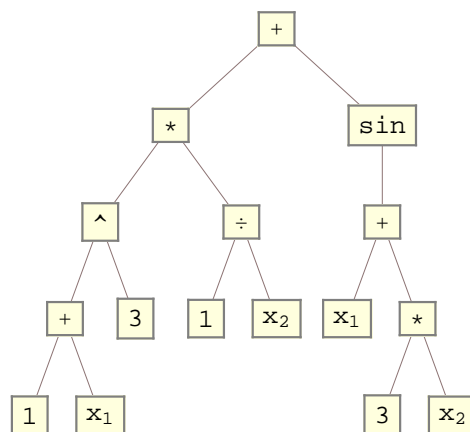


Abbildung 4.17: Baumstruktur entsprechend der Auswertereihenfolge

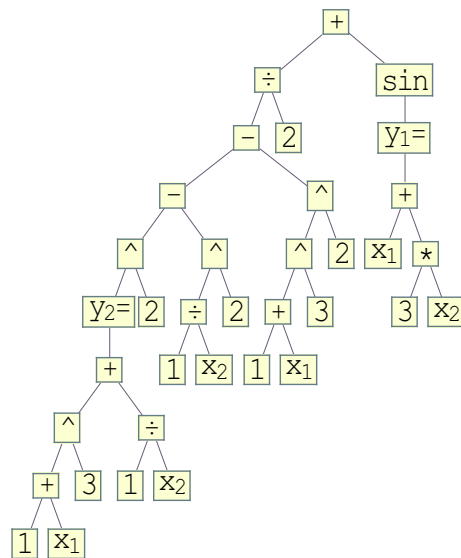


Abbildung 4.18: Auswertereihenfolge nach der Separation

4.6 Gewinnung eines Schaltungsmodells aus Simulationsdaten

In vielen Fällen ist es nicht bzw. nur mit hohem Aufwand möglich, das PWA-Zustandsmodell mit den in Abschnitt 4.5 beschriebenen Methoden direkt aus einer funktionalen Beschreibung abzuleiten. Grund hierfür kann zum einen eine zu hohe Komplexität der Schaltung sein. Zwar lassen sich mit Hilfe einer Partitionierung durch eine Strukturerkennung Unterblöcke bilden, die sich mit den vorherigen Verfahren ausreichend modellieren lassen (siehe Beispiel in Abschnitt 6.5) und damit die Komplexität des Systems senken, jedoch bleiben häufig viele einzelne komplexe Bauelementmodelle (in der Regel Transistoren) in der Netzliste stehen. Außerdem liefert ein Zusammenschalten der aus den Unterblöcken entstandenen PWA-Zustandsmodelle zu einem Gesamtsystem nicht immer zufriedenstellende Ergebnisse, da die Wechselwirkungen zwischen den Unterblöcken nicht hinreichend berücksichtigt werden können. Ein zweiter Grund kann sein, dass für die zu modellierende Schaltung überhaupt keine funktionale Beschreibung vorliegt. In den folgenden Abschnitten betrachtet man das System als Black-Box, für das sich mit Hilfe einer Approximation eine algebraische Beschreibung erzeugen lässt.

4.6.1 Approximation nichtlinearer Verläufe

Dieser Abschnitt befasst sich mit den Möglichkeiten, eine stückweise Beschreibung des Schaltungsverhaltens aus Simulationsergebnissen zu erzeugen. Der Ansatz ist somit dem im Abschnitt 2.3 vorgestellten Black-Box Ansatz zuzuordnen.

Da die zu erzeugenden PWA-Modelle mit Hilfe der affinen Arithmetik Ergebnisintervalle bilden können, soll der Ansatz hierbei Ergebnisse aus mehreren Simulationsläufen mit statistischen Abweichungen in einem einzigen PWA-Modell einschließen können.

Die nichtlinearen Verläufe seien durch eine Menge von p d -dimensionalen Datenpunkten und den zugehörigen reellen Funktionswerten gegeben:

$$P = (\vec{x}_i, y_i) \quad \text{mit } \vec{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = \{1, \dots, p\} \quad (4.22)$$

Gesucht ist eine analytische Funktion $\hat{y} = f(\vec{x})$, die die den Daten zugrunde liegende Funktion bestmöglich annähert und in affiner Form zurückliefert. Im Gegensatz zu einer Interpolation, d.h. einer exakten Übereinstimmung der Funktion an den gegebenen Datenpunkten, soll hier eine allgemeinere Approximationsfunktion konstruiert werden, welche in den gegebenen Datenpunkten mit einer vorgegebenen Toleranz vom tatsächlichen Wert der Funktion abweichen darf.

So lässt sich beispielsweise eine Monte-Carlo-Simulation durchführen, die alle relevanten Variationen in den Technologie- und Betriebsparametern berücksichtigt. Als Ergebnis liegt dann eine große Anzahl gestreuter Simulationskurven vor, die man durch ein PWA-Zustandsmodell annähern kann. Durch Einfügen affiner Terme in die Systemmatrizen erzielt man im Weiteren eine Aufweitung, mit der sich alle Simulationskurven einschließen lassen.

Für eine geschlossene Formulierung wird in der Regel eine Expansion von Basisfunktionen h mit

$$f(\vec{x}) = \sum_{i=1}^K h_i(\vec{x}) \quad (4.23)$$

zur Approximation verwendet, wobei K die Anzahl der Basisfunktionen angibt. Die bekannten Approximationsverfahren unterscheiden sich danach, welche Basisfunktionen verwendet werden. Die Bandbreite reicht von polynomialen Basisfunktionen und Splines bis hin zu trigonometrischen und sigmoidalen Funktionen.

Um festzulegen, wie gut $f(\vec{x})$ die Punktmenge P approximiert, wird eine Kostenfunktion für den Fehler definiert. Üblicherweise benutzt man den quadratischen Fehler und versucht, diesen über die Methode der kleinsten Quadrate zu minimieren:

$$\text{minimize} \left(\sum_{i=1}^p (y_i - f_i(\vec{x}))^2 \right) \quad (4.24)$$

Da die Basisfunktionen h_i für PWA-Modelle stückweise definiert sind und die erste Ableitung an den Übergängen nicht kontinuierlich ist, ergeben sich Konvergenzprobleme bei der Optimierung. Als Lösung dieses Optimierungsproblems wird beispielsweise von Pittman [PiMu00] die Verwendung eines genetischen Algorithmus vorgeschlagen. Das Verfahren lässt sich allerdings nur für $\vec{x} \in \mathbb{R}^2$ einsetzen. Für allgemeinere Verläufe wird die Anzahl der zu optimierenden Parameter schnell zu hoch. Ist $f : \mathbb{R}^d \rightarrow \mathbb{R}$ konvex, ist dieses Problem auch für große P lösbar (siehe [BoVa04] [MaBo08]).

Liegt bereits eine Partitionierung der Daten vor, vereinfacht sich das Problem. Eine Approximation ist so nur für jeden einzelnen Bereich nötig, wobei dann im Falle einer PWA-Modellierung jeweils ein lokales affines Modell zu finden ist. Die Basisfunktion aus Gleichung 4.23 entspricht $h(\vec{x}) = A \cdot \vec{x} + \hat{f}$ und es gilt $K = 1$. Ist ein stetiger Übergang zwischen den lokalen Modellen gewünscht (wie für die PWA-Zustandsmodelle der Fall), müssen zusätzliche Randbedingungen bei den lokalen Approximationen eingeführt werden.

Die Methode, mit der die nichtlinearen Verläufe approximiert werden sollen, hängt unmittelbar mit der gewünschten Darstellungsform von f zusammen. Für die PWA-Zustandsmodell-Darstellung ist dabei keine direkte Herleitung möglich, da die Wechsel zwischen den affinen Abbildungen (also der Rand der einzelnen Partitionen) nur indirekt über die Komplementaritätsbedingung in die Modellgleichungen einfließen.

Benötigt werden folglich Basisfunktionen, für die eine Umrechnung von f in ein PWA-Zustandsmodell einfach möglich ist. Des Weiteren soll das Approximationsverfahren für allgemeine Verläufe und große P einsetzbar sein.

Mit den erstmals in [Brei93] vorgestellten Hinge-Funktionen als Basisfunktion lassen sich beide Forderungen erfüllen. Weiterhin ist man bei der Modellierung, genauso wie bei den PWA-Zustandsmodellen, nicht auf eine kanonische Darstellung beschränkt. Die Definition der Hinge-Funktionen und ein Approximationsverfahren sind im folgenden Abschnitt beschrieben.

4.6.2 Hinge-Funktionen als Basisfunktionen

Eine Hinge-Funktion $h(\vec{w})$ besteht aus zwei miteinander verbundenen Hyperebenen. Die beiden Hyperebenen sind definiert durch

$$y_+ = \Theta_+^T \vec{w} \quad \text{und} \quad y_- = \Theta_-^T \vec{w}. \quad (4.25)$$

Dabei ist $\vec{w} \in \mathbb{R}^d$ der Eingangsvektor mit $\vec{w}^T = \{w_0, w_1, \dots, w_d\}$ und $w_0 = 1$. Die Hingefunktion bildet den Eingangsvektor auf die Größe $y \in \mathbb{R}$ ab. Weiterhin gilt

$$\{w : \quad \Delta^T \vec{w} = 0\} \quad (4.26)$$

mit $\Delta = \Theta_+ - \Theta_-$. Der Unterraum des Eingangsraumes, der Gleichung 4.26 erfüllt, wird als *Hinge* einer *Hinging Hyperebene* bezeichnet (siehe Abbildung 4.19).

Die *Hingefunktion* wird über zwei mögliche Gleichungen formuliert:

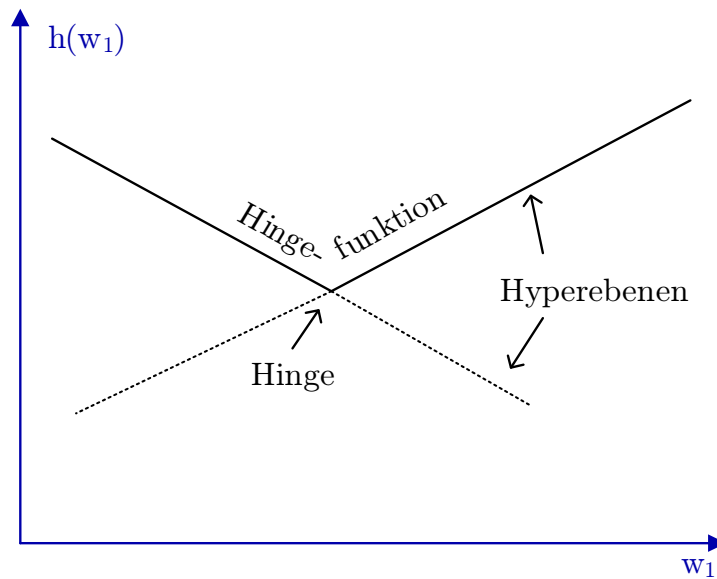
$$y = h(\vec{w}) = \max(\Theta_+^T \vec{w}, \Theta_-^T \vec{w}) \quad (4.27)$$

$$y = h(\vec{w}) = \min(\Theta_+^T \vec{w}, \Theta_-^T \vec{w}) \quad (4.28)$$

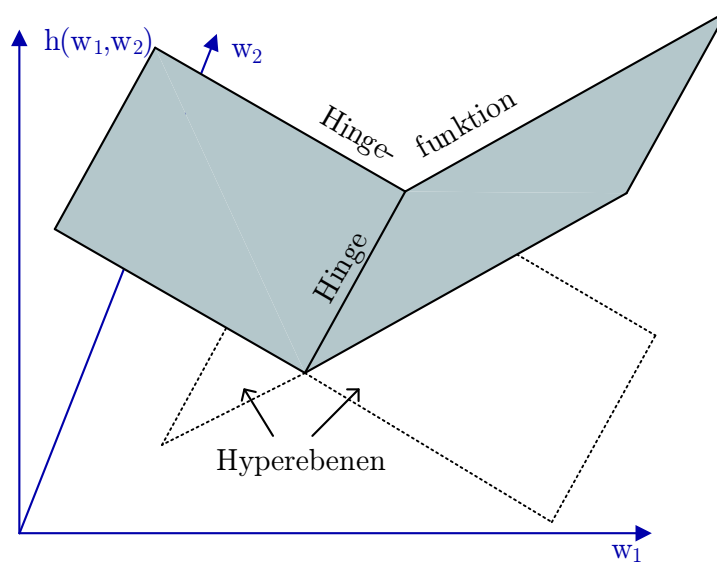
Je nachdem, ob ein konvexer oder konkaver Verlauf gewünscht ist, ist Gleichung 4.27 oder Gleichung 4.28 zu wählen.

4.6.3 Approximation mit Hinging-Hyperebenen

Die verfolgte Strategie bei der Approximation mit Hinging-Hyperebenen beruht auf einer schrittweisen Erhöhung der Anzahl an Hingefunktionen in Gleichung 4.23. Man beginnt zunächst bei $K = 1$ mit folgendem Ablauf, der im Weiteren als Hingesuchalgorithmus (HSA) bezeichnet wird:



(a) Hingefunktion mit einer Eingangsgröße



(b) Hingefunktion mit zwei Eingangsgrößen

Abbildung 4.19: Hinge, Hinging Hyperebenen und Hingefunktion

Algorithmus 2 : Hingesuchalgorithmus**Eingabe** : Punktmenge P **Ausgabe** : Θ_+ , Θ_- $\{S_+, S_- \leftarrow \text{FindeInitialeVerteilung}(P);$ **wiederhole** $\Theta_+ \leftarrow \text{LeastSquare}(S_+);$ $\Theta_- \leftarrow \text{LeastSquare}(S_-);$ $S_+ \leftarrow \{\vec{x} : \vec{x}^T (\Theta_+ - \Theta_-) > 0\};$ $S_- \leftarrow \{\vec{x} : \vec{x}^T (\Theta_+ - \Theta_-) \leq 0\};$ **bis keine Änderungen der Mengen S_+ und S_- ;**

Auf diese Weise erhält man eine erste Hingefunktion $h_1(\Theta_-, \Theta_+)$, die die Punktmenge P approximiert. Ein Beispiel hierzu ist in Abbildung 4.20 gegeben. Um die zweite Hingefunktion h_2 zu bestimmen, bildet man die Differenz (das Residuum, siehe Abbildung 4.20(b)) zwischen den Punkten in P und der Hingefunktion mit $\tilde{y}^{(1)} = y - h_1$. Mit Hilfe des HSA bildet man h_2 aus der Punktmenge $(\tilde{x}_i, \tilde{y}_i^{(1)})$.

Nun wird erneut h_1 angepasst, indem man $\tilde{y}^{(2)} = y - h_2$ für den HSA zugrunde legt. Zwischen der Anpassung von h_1 und h_2 wird jetzt solange iteriert, bis beide Hingefunktionen konvergieren.

Möchte man eine dritte Hingefunktion hinzufügen, geht man ähnlich vor. Zuerst bestimmt man $\tilde{y}^{(1,2)} = y - h_1 - h_2$ und ermittelt mit Hilfe des HSA die Funktion h_3 . Über mehrere Iterationen mit dem HSA passt man nun die Hingefunktionen an. Dabei nutzt man jeweils $\tilde{y}^{(2,3)}$, $\tilde{y}^{(1,3)}$ und $\tilde{y}^{(1,2)}$. Die Approximation mit Hingefunktionen wurde von zahlreichen Autoren untersucht [Erns98, PuSj98, LiWa03, QuJr04]. Dabei wurden Vorschläge zur Verbesserung der Laufzeit und der Konvergenz des Approximationsverfahrens gemacht. Zur Modellierung von Schaltungen ist der oben beschriebene Algorithmus jedoch ausreichend.

Im folgenden Abschnitt wird gezeigt, wie die so gebildete Summe von Hingefunktionen in das PWA-Zustandsmodell überführt wird.

4.6.4 Überführung vom Hingemodell in eine Zustandsmodell-Darstellung

Eine Hingefunktion besitzt genau zwei Hyperebenen. Vergleicht man Gleichung (4.26) mit der Zustandsgleichung (siehe Gleichung 4.2) des PWA-Zustandsmodells

$$j = u + \mathbf{C}\vec{v} + \mathbf{D}\vec{i} + g$$

und berücksichtigt, dass am Übergang zwischen zwei Hyperebenen $j = 0$ und $u = 0$ sind, lassen sich \mathbf{C} und \vec{g} durch Koeffizientenvergleich bestimmen. Da keine Abhängigkeit von \vec{i} besteht, gilt $\mathbf{D} = 0$. Die Matrizen der Zustandsgleichungen ergeben sich somit zu

$$\mathbf{C}_i = \begin{pmatrix} \Theta_{+,1} - \Theta_{-,1} \\ \Theta_{+,2} - \Theta_{-,2} \\ \dots \\ \Theta_{+,d} - \Theta_{-,d} \end{pmatrix}, \quad \mathbf{D}_i = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \quad \vec{g}_i = (\Theta_{+,0} - \Theta_{-,0}).$$

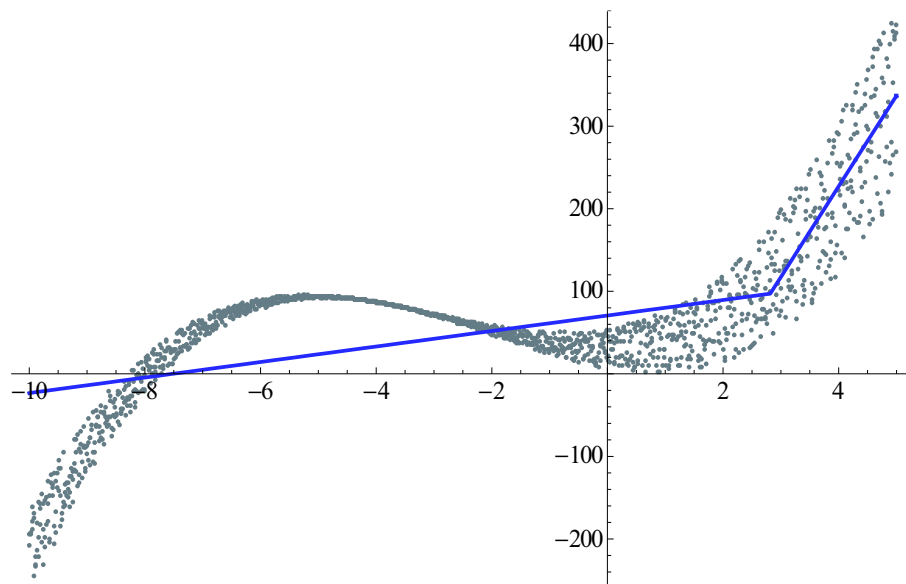
Ist die Hingefunktion über die Max-Beziehung definiert, ergibt sich des Weiteren eine gültige Abbildung mit den Matrizen

$$\mathbf{A}_i = (\Theta_{+,1} \quad \Theta_{+,2} \quad \dots \quad \Theta_{+,d}), \quad \mathbf{B}_i = (-1), \quad \vec{f}_i = (\Theta_{+,0}).$$

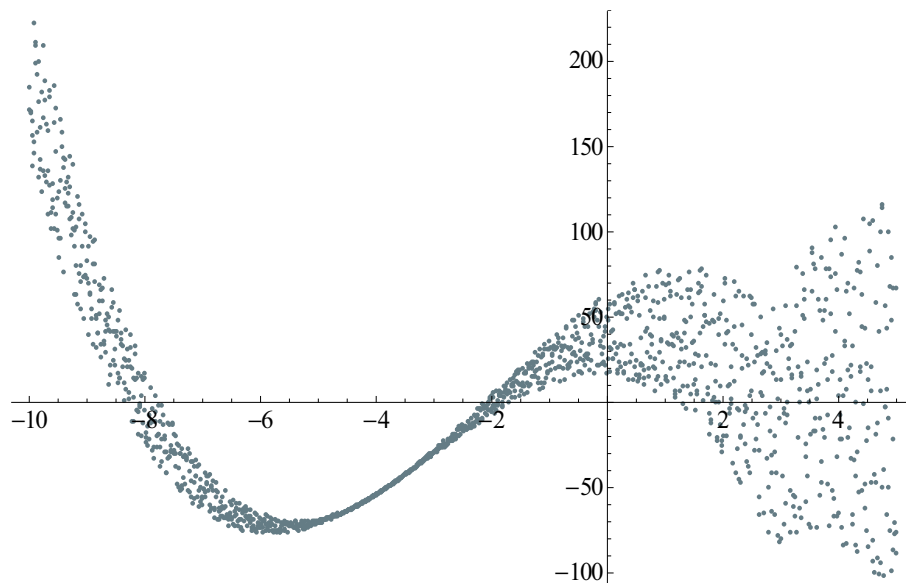
Liegt eine Min-Beziehung vor, ergeben sich die Matrizen zu

$$\mathbf{A} = \begin{pmatrix} \Theta_{-,1} & \Theta_{-,2} & \dots & \Theta_{-,d} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} \Theta_{-,0} \end{pmatrix}.$$

Zum Überführen einer Summe aus mehreren Hingefunktionen in ein PWA-Zustands-



(a) Daten durch erste Hingefunktion angenähert



(b) Residuum $\bar{y}^{(1)} = y - h_1$

Abbildung 4.20: Erster Schritt der Approximation mit dem HSA-Algorithmus

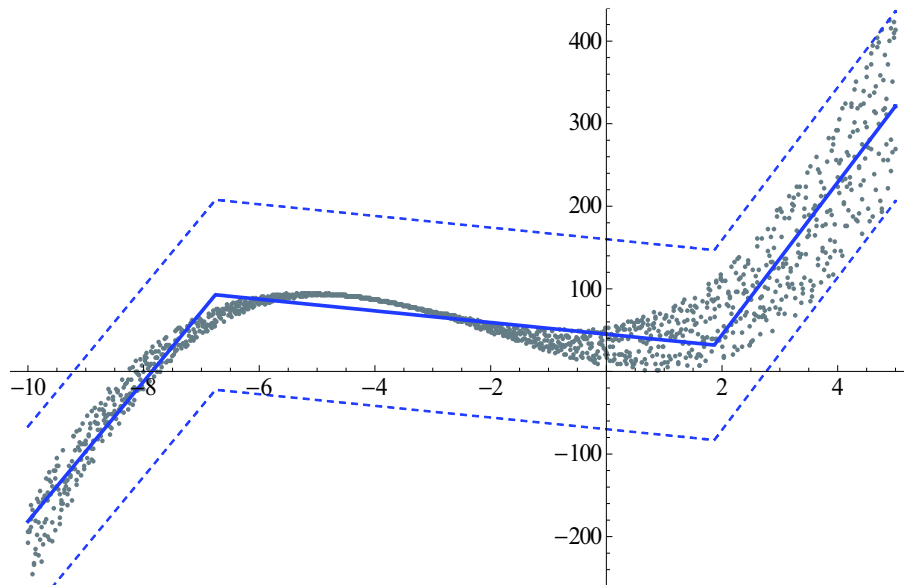


Abbildung 4.21: Approximation mit Hilfe zweier Hinge-Funktionen und eingezeichneten Grenzen

modell, bilden sich die Systemmatrizen aus den einzelnen PWA-Systemen nach

$$\begin{aligned}
 \mathbf{A}_{ges} &= \sum_{i=1}^K \mathbf{A}_i, & \mathbf{B}_{ges} &= \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \dots & \mathbf{B}_K \end{pmatrix}, \\
 \mathbf{C}_{ges}^T &= \begin{pmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \dots & \mathbf{C}_K \end{pmatrix}, & \mathbf{D}_{ges}^T &= \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \dots & \mathbf{D}_K \end{pmatrix}, \\
 \vec{f}_{ges} &= \sum_{i=1}^K \vec{f}_i, & \vec{g}_{ges}^T &= \begin{pmatrix} \vec{g}_1 & \vec{g}_2 & \dots & \vec{g}_K \end{pmatrix}.
 \end{aligned}$$

5 Symbolische Umgebung zur Modellierung und Simulation

Nachdem im vorangegangenen Kapitel der Aufbau des PWA-Zustandsmodells und verschiedene theoretische Ansätze zur Modellgewinnung vorgestellt wurden, wird in diesem Kapitel auf die praktische Realisierung der Ansätze in einem geschlossenen Funktionspaket (genannt *Sur-P.A.S.S.*) für die Mathematiksoftware Mathematica eingegangen.

5.1 Sur-P.A.S.S.

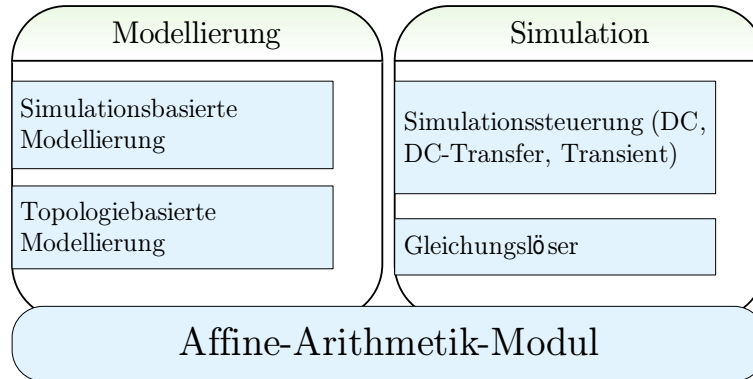
Bei *Sur-P.A.S.S.* handelt es sich um ein Paket zur Modellierung und Simulation von analogen Schaltungen, das im Rahmen dieser Arbeit entstanden ist. Der Name ergibt sich aus der Abkürzung für **P**iecewise **A**ffine **S**emisymbolic **S**imulation. Die Funktionalität baut auf der Mathematiksoftware *Mathematica* auf. Zur Unterstützung von Netzlisten analoger Schaltungen bedient sich *Sur-P.A.S.S.* zusätzlich der Funktionen aus dem Erweiterungspaket *Analog Insydes* [Fra08].

Im Einzelnen gliedert sich *Sur-P.A.S.S.* in drei Module. Ein Modul dient der Modellierung von Schaltungen, das zweite Modul ermöglicht die Simulation der erzeugten Schaltungsmodelle. Beide Module bedienen sich, wie in Abbildung 5.1 dargestellt, der Funktionalität aus dem dritten, dem Affine-Arithmetik-Modul. Alle drei Module werden in den folgenden Abschnitten näher beschrieben.

5.2 Affine-Arithmetik-Modul

Das in dieser Arbeit implementierte Affine-Arithmetik-Modul erweitert die mathematischen Operationen in Mathematica. Es besteht aus den Paketen

- Arithmetik.m
- AAFUtils.m

Abbildung 5.1: Struktur von *Sur-P.A.S.S.*

Durch das Modul wird neben den in Mathematica definierten numerischen Größen ein Objekt vom Typ AAF eingeführt:

```

AAF [
  NominalValue,
  {
    {DeviationIndex1, DeviationValue1},
    {DeviationIndex2, DeviationValue2},
    ...
  },
  UncorrelatedDeviation
]
  
```

Das AAF-Objekt repräsentiert einen Ausdruck in AA-Form

$$\hat{x} = x_0 + \sum_{i=1}^{\xi} x_i \epsilon_i + x_{NL} \epsilon_{NL} \quad \text{mit } \epsilon_i, \epsilon_{NL} = [-1, 1]. \quad (5.1)$$

Die Bezeichnung *NominalValue* steht für x_0 , die darauf folgende Liste von Wertepaaren gibt den Summenterm $\sum_{i=1}^{\xi}$ wieder. Dabei entspricht *DeviationIndex* dem Index i und *DeviationValue* dem zugehörigen x_i .

Gleichung (5.1) unterscheidet sich durch den Term $x_{NL} \epsilon_{NL}$ leicht von der Definition in Abschnitt 3.3.2. Unkorrelierte Abweichungen, wie sie beispielsweise durch nicht affine Operationen entstehen, werden durch das Symbol *UncorrelatedDeviation* bzw. $x_{NL} \epsilon_{NL}$ wiedergegeben. Eine Erhöhung des Indexes ξ findet bei der Durchführung mathematischer Operationen nicht statt. Stattdessen erhöht jede nicht affine Operation den Eintrag für x_{NL} .

Alle mathematischen Ausdrücke, die für den Lösungsprozess des PWA-Zustandsmodells benötigt werden, lassen sich durch eine Verkettung von Multiplikationen, Additionen, Quadrieren und Inversionen mit einem bzw. zwei Operanden wiedergeben.

Grund hierfür ist die Tatsache, dass für die Auswertung der Systemmatrizen lediglich die lineare Algebra angewendet werden muss. Alle nichtlinearen Beziehungen sind wegen des Aufbaus des PWA-Zustandsmodells implizit über die Komplementaritätsbedingung gegeben.

Für Berechnungen mit Operanden in affiner Form wurden in *Sur-P.A.S.S.* spezialisierte Routinen entwickelt. In vereinfachter Darstellung in Anlehnung an die Mathematica-Syntax ergibt sich beispielsweise das Ergebnis aus einer Multiplikation zweier AAF-Objekte durch:

```
Times[v_AAF, w_AAF] :=
  AAF[
    Nominalwert von v * Nominalwert von w,
    (* Liste über alle Indizes i *)
    {{i, NominalValue[v] * wi
      + NominalValue[w] * vi},
     ...
    },
    Radius von v * Radius von w
  ]
```

Rückgabewert ist wiederum ein AAF-Objekt. Die Routine entspricht der Rechenvorschrift

$$\hat{x} \cdot \hat{w} = (v_0 \cdot w_0) + \sum_{i=1}^{\xi} ((v_0 \cdot w_i + w_0 \cdot v_i) \epsilon_i) + \text{rad}(\hat{x}) \cdot \text{rad}(\hat{w}) \cdot \epsilon_{NL}.$$

Weitere Spezialisierungen wurden für

- die skalare Multiplikation: `Times[_AAF, _?NumberQ]`,
- die Additionen zweier Intervalle: `Plus[_AAF, _AAF]` und
- die Addition eines Intervalls mit einem Skalar: `Plus[_AAF, _?NumberQ]`

realisiert, die jeweils die Rechenvorschriften aus Abschnitt 3.3.2 umsetzen.

Affine Formen lassen sich in *Sur-P.A.S.S.* direkt durch AAF-Objekte angeben oder durch die beiden Funktionen

```
BuildAAF[NominalValue, {{DeviationSymbol, DeviationValue}, ...}]
```

und

```
I2AAF[{untere Grenze, obere Grenze}, DeviationSymbol]
```

erzeugen. Sie registrieren die angegebenen Symbole `DeviationSymbol` für die Abweichung in einer globalen Liste. Sollte eines der Symbole bereits in einem zuvor definierten Intervall genutzt werden, wird der Index im AAF-Objekt entsprechend gleich gewählt, ansonsten wird ein neuer Index erzeugt ($\xi = \xi + 1$). Als Gegenstück kann die Funktion

```
DecomposeAAF[_AAF]
```

genutzt werden. Durch sie werden alle AAF-Objekte in Mathematica-Standardausdrücke überführt. Der Einsatz beider Funktionen wird anhand des folgenden Beispiels deutlich:

```
In[1] := intervall1 = BuildAAF[x0, {{epsA,x1},{epsB,x2}}]
Out[1] = AAF[x0,{{1,x1},{2,x2}},0]
```

```
In[2] := intervall2 = BuildAAF[x0, {{epsB,x2},{epsC,x3}}]
Out[2] = AAF[x0,{{2,x2},{3,x3}},0]
```

```
In[3] := result = intervall1 + c1 * intervall2 + c2
Out[3] = c2+AAF[x0,{{1,x1},{2,x2}},0]+c1 AAF[x0,{{2,x2},{3,x3}},0]
```

```
In[4] := DecomposeAAF[result]
Out[4] = c2+x0+epsA x1+epsB x2+c1 (x0+epsB x2+epsC x3)
```

Zunächst wird den Symbolen `intervall1` und `intervall2` durch `BuildAAF` je ein AAF-Objekt zugeordnet. Die partiellen Abweichungen werden dabei nicht durch einen gemeinsamen Index in Korrelation gesetzt, sondern durch gleiche Symbole `epsA`, `epsB` und `epsC`. Die nachfolgende Berechnung von `result` erfolgt mit den AAF-Objekten. Das Ergebnis wird dann mit `DecomposeAAF` wieder in eine Standardformel überführt, indem alle AAF-Objekte aufgelöst werden.

Des Weiteren wurden folgende Operationen implementiert, die sowohl auf AAF-Objekte, als auch auf Listen und Matrizen mit AAF-Objekten anwendbar sind:

- `RadAAF[x_]`: Radiusfunktion
- `NomAAF[x_]`: Gibt Nominalwert zurück
- `negBound[x_]`: Gibt Nominalwert abzüglich Radius zurück
- `posBound[x_]`: Gibt Nominalwert zuzüglich Radius zurück

5.3 Modul zur Modellierung

Das Modul zur Modellierung setzt die in Kapitel 4 vorgestellten Modellierungsansätze softwaretechnisch um. Es besteht aus mehreren Paketen, die im Folgenden erläutert werden.

5.3.1 Systemdefinition

Die Definition von PWA-Zustandsmodellen und grundlegende Funktionen zum Umgang befinden sich in *Sur-P.A.S.S.* im Paket:

- PWASystemDefs.m

Das Modell ist dort über das Objekt

```
PWLSyst[{  
    A_?MatrixQ,  
    B_?MatrixQ,  
    C_?MatrixQ,  
    D_?MatrixQ,  
    f_?MatrixQ,  
    g_?MatrixQ  
}],  
SystemType->"DC"|"Transient"]
```

definiert, welches auf unterschiedlichen Wegen erzeugt werden kann. Neben der direkten Erzeugung durch explizite Angabe der Systemmatrizen lässt sich das Objekt über die Funktion

```
GenSystemMatrices[{Steigungen, ...} ,  
                  {Stützstellen, ...},  
                  {y-Achsenabschnitt}]
```

erzeugen. Die Funktion führt dabei die in Abschnitt 4.3 angegebenen Berechnungen aus.

5.3.2 Funktionen für Hinging-Hyperebenen

Die in Abschnitt 4.6 vorgestellten Hinge-Funktionen werden in *Sur-P.A.S.S.* durch das Objekt

```
HingeObj["Min"|"Max", {{ThetaN1, ThetaP2, ...},
                       {ThetaP1, ThetaP2, ...}}]
```

ausgedrückt. Die Listen mit `ThetaN` und `ThetaP` entsprechen den Vektoren Θ_+^T und Θ_-^T aus Gleichung (4.27) bzw. Gleichung (4.28).

Alle notwendigen Funktionen zum Umgang mit Hinge-Objekten befinden sich im Paket

- `Hinge.m`

Für eine Herleitung eines Hinge-Objektes aus bereits vorliegenden Simulationsdaten dient die Funktion

```
FindHingeFit[Pointlist_, NumberOfHinges_, InitialHingeObjs___List],
```

die den Hinge-Suchalgorithmus von Seite 56 implementiert. Die Funktion führt im weiteren Verlauf die in Abschnitt 4.6.3 beschriebene iterative Verbesserung bei der Modellierung durch Hinging-Hyperebenen durch. Eingabegröße ist neben den Datenpunkten und der gewünschten Anzahl resultierender Hinge-Objekte optional auch eine initiale Liste bereits vorgegebener Hinge-Objekte. Als Ergebnis wird eine Liste von Hinge-Objekten geliefert.

Aus dieser Liste lässt sich schließlich mit Hilfe von

```
GenSystemMatrices[HingeObjList_List]
```

ein `PWLSystem`-Objekt erzeugen, das zur Simulation verwendet werden kann.

5.3.3 Funktionen zur Ableitung aus funktionalen Beschreibungen

In den beiden Paketen

- `FuncApproximation.m`
- `FuncSeperation.m`

befinden sich die notwendigen Routinen, um funktionale Systembeschreibungen in PWA-Zustandsmodelle zu wandeln.

Somit ist es möglich, über einen Aufruf von `SeparateFunction` eine beliebige Funktion mit mehreren Eingangsvariablen nach dem Verfahren aus Abschnitt 4.5.2 zu separieren. Als Rückgabe erhält man eine Liste mit voneinander unabhängigen Funktionen mit jeweils einer Veränderlichen.

Diese Liste kann unter Angabe von Wertebereichen für die Eingangsvariablen des zu erzeugenden PWA-Zustandsmodells an die Funktion `LinApproxLoop` übergeben werden, wodurch eine Approximation der Einzelfunktionen durch stückweise affine Funktionen erfolgt. Daraus kann im Anschluss ein `PWASys`t-Objekt erzeugt werden.

5.3.4 Verarbeitung von Netzlisten

Um Netzlisten in Mathematica verarbeiten zu können, bzw. Netzlisten anderer Simulatoren in Mathematica importieren zu können, nutzt *Sur-P.A.S.S.* das Erweiterungspaket *Analog-Insydes* [Fra08]. In *Analog-Insydes* wird eine eigene Syntax zur Darstellung von Netzlisten definiert. Neben Verhaltensmodellen für Standardbauelemente ist es in *Analog-Insydes* zusätzlich möglich, eigene Modelle zu implementieren.

In *Sur-P.A.S.S.* wird diese Möglichkeit genutzt, um eigene PWA-Modelle für Standardblöcke zu erstellen. Diese Standardblöcke definieren die Anzahl der Klemmen und geben die Strom-Spannungsbeziehungen als unbestimmte Funktion vor. Da *Analog-Insydes* nicht mit den in *Sur-P.A.S.S.* definierten `PWASys`t-Objekten umgehen kann, dienen diese hier als Platzhalter. Wie in Abschnitt 4.4 vorgestellt, sind die möglichen Blöcke stromgesteuerte Strom- und Spannungsquellen (`CCC`, `CCV`) und spannungsgesteuerte Strom- und Spannungsquellen (`VCC`, `VCV`). Als passive Zweipole wurden der Widerstand und die Konduktanz implementiert.

Mit *Analog-Insydes* lässt sich für die vorliegenden Schaltung eine Modifizierte Knotenanalyse durchführen und die Netzliste somit in ein symbolisches DAE-System überführen. Die Variablen der Ein- und Ausgangsgrößen aller enthaltenen PWA-Modelle liegen hier weiterhin als Platzhalter vor. Mit der Funktion `BuildPWLSys`t kann

das DAE-System schließlich in ein neues PWLSyst-Objekt überführt werden. Die Hierarchien werden dabei aufgelöst. Weitere notwendige Angaben für BuildPWLSyst sind alle in der Netzliste verwendeten PWASyst-Objekte und die Knotenpotentiale bzw. Zweigströme, die als Ein- und Ausgangsgrößen des neuen PWA-Zustandsmodells dienen sollen.

Für den Umgang mit Netzlisten aus *Analog-Insydes* und zum Umwandeln von DAE-Objekten in ein PWA-Zustandsmodell wurden für *Sur-P.A.S.S.* im Rahmen dieser Arbeit folgende Pakete implementiert :

- PWA-Netlisting.m
- FlattenPWLSystem.m

5.4 Modul zur Simulation

In den Paketen

- SolveAAF.m
- Simulation.m

ist die Funktionalität für eine Simulation von PWA-Zustandsmodellen enthalten. In `SolveAAF.m` sind dabei die Funktionen implementiert, mit denen die Lösung bei gegebenem PWASyst-Objekt und Eingangsvektor \vec{x} berechnet werden. In `Simulation.m` ist die Simulatorsteuerung für die unterschiedlichen Simulationsarten implementiert.

In den folgenden Abschnitten werden zunächst die theoretischen Grundlagen für den Aufbau des in *Sur-P.A.S.S.* enthaltenen Simulators behandelt. Der letzte Abschnitt zeigt, wie Simulationen in *Sur-P.A.S.S.* durchgeführt werden.

5.4.1 Lösen des linearen Komplementaritätsproblems

Wie in Kapitel 4 bereits beschrieben, wird durch den Aufbau der Gleichungen (4.11 - 4.13) ein lineares Komplementaritätsproblem (engl.: Linear Complementary Problem - LCP) beschrieben. Das LCP steht in Beziehung zu einer großen Sammlung an mathematischen Problemen mit verschiedensten Lösungsverfahren [CoPa92]. Allgemein lässt sich das LCP definieren durch:

Definition 5.1 Für eine gegebene rationale Matrix $\mathbf{M} \in \mathbb{R}^{k \times k}$ und einen gegebenen rationalen Vektor $\vec{q} \in \mathbb{R}^k$ sollen die Vektoren $\vec{j}, \vec{u} \in \mathbb{R}^k$ gefunden werden. Dabei sollen folgende drei Bedingungen eingehalten werden:

- $\vec{q} = \vec{j} - \mathbf{M}\vec{u}$
- $j_i, u_i \geq 0$ für alle i
- $j \cdot u = 0$ mit $j_i = 0$ oder $u_i = 0$ für alle i

Es existieren zahlreiche Lösungsverfahren, die ein effizientes Lösen des LCP ermöglichen [Lemk67, vdp74, vB81]. Aufgrund seiner Einfachheit wird zur Lösung im Folgenden der Algorithmus nach Katzenelson [Katz65] verwendet. Dieser Lösungsalgorithmus zählt zur Klasse der Homotopieverfahren.

Bei Homotopieverfahren wird ein zusätzlicher Parameter, im Folgenden λ genannt, zum Problem hinzugefügt, um dann einen Lösungspfad im so erweiterten Lösungsraum zu finden. Möchte man ein Problem der Form $\vec{j} = \vec{f}(\vec{u}, \vec{q})$ lösen, für das bereits eine Lösung $\vec{j} = \vec{f}(\vec{u}, \vec{q}_{start})$ bekannt ist, wird das Problem reformuliert zu

$$\vec{j} = \vec{f}^*(\vec{u}, \vec{q}, \lambda) = \vec{f}(\vec{u}, \vec{q}_{start} + \lambda(\vec{q} - \vec{q}_{start})). \quad (5.2)$$

Das Verfahren startet mit $\lambda = 0$ und erhöht kontinuierlich den Wert von λ bis $\lambda = 1$. Bei Erreichen von $\lambda = 1$ wird somit wieder das ursprüngliche Problem beschrieben, wobei die entsprechende Lösung entlang des Pfades $0 \leq \lambda \leq 1$ ermittelt wurde.

Bezogen auf die Gleichung aus Definition 5.1 ergibt sich

$$\vec{j} = \mathbf{M} \cdot \vec{u} + \vec{q}_{start} + \lambda(\vec{q} - \vec{q}_{start}). \quad (5.3)$$

Dabei wird vorausgesetzt, dass die bekannte Lösung \vec{q}_{start} gültig ist für $\vec{u} = \vec{0}$ und somit die Komplementaritätsbedingung erfüllt ist. Dies bedingt gleichzeitig, dass alle Elemente in \vec{j} positiv oder gleich Null sind. Erhöht man nun den Homotopieparameter λ , erreicht man – sofern nicht \vec{q}_{start} bereits eine gültige Lösung des Gesamtproblems darstellt – einen Wert λ_z , bei dem ein Element j_p mit $p \in \{1, \dots, k\}$ negativ werden muss. Dies ist genau dann der Fall, wenn $q_{z,p} = q_{start,p} + \lambda_z(q_p - q_{start,p})$ zu Null wird und durch eine weitere Erhöhung von λ einen negativen Wert einnehmen würde.

Das Verfahren nimmt eine Vertauschung der beiden Elemente j_p und u_p vor, was aufgrund der Komplementarität erlaubt ist. Dies führt zu einem neuen Gleichungssystem der Form:

$$\vec{j}^* = \mathbf{M}^* \cdot \vec{u}^* + \vec{q}_{start}^* + \lambda(\vec{q}^* - \vec{q}_{start}^*). \quad (5.4)$$

Die Komplementaritätsbedingung ist damit erfüllt für $\vec{u}^* = \vec{0}$ und $\vec{j}^* = \vec{q}_{start}^* + \lambda_z(\vec{q}^* - \vec{q}_{start}^*)$, so dass eine weitere Erhöhung von λ vorgenommen werden kann. Das Verfahren wird solange fortgesetzt, bis $\lambda = 1$ erreicht wird. An jedem Punkt, an dem die Komplementarität verletzt wird, erfolgt eine erneute Vertauschung innerhalb von \vec{u}^* und \vec{j}^* .

Folgendes Beispiel soll dies verdeutlichen, wobei die Lösung für die Vektoren \vec{j} und \vec{u} gesucht ist:

$$\begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 2 & -1 \\ \frac{1}{2} & 1 \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \underbrace{\begin{pmatrix} -4 \\ 1 \end{pmatrix}}_{\vec{q}}. \quad (5.5)$$

Die einfache Lösung $\vec{u} = \vec{0}$ verletzt offensichtlich die Komplementaritätsbedingung. Reformuliert man \vec{q} aus Gleichung (5.5) unter Verwendung des Homotopieparameter λ zu $\vec{q}_z = \vec{q}_{start} + \lambda(\vec{q} - \vec{q}_{start})$ und nimmt als gültige Startlösung $\vec{q}_{start}^T = (2, 1)$, dann ergibt sich:

$$\begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} -6 \\ 0 \end{pmatrix}. \quad (5.6)$$

Im weiteren Verlauf wird λ von Null an bis Eins erhöht. Bis einschließlich $\lambda_z = \frac{1}{3}$ ist für die einfache Lösung $\vec{u} = \vec{0}$ die Komplementarität erfüllt, da alle Elemente von \vec{q}_z positiv bleiben. Für λ_z wird das erste Element jedoch zu Null, so dass dementsprechend ein Tausch von j_p und u_p mit $p = 1$ vorgenommen wird.

Das neue System lautet damit:

$$\begin{pmatrix} u_1 \\ j_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{5}{4} \end{pmatrix} \begin{pmatrix} j_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} -1 \\ \frac{1}{2} \end{pmatrix} + \lambda \left(\begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} -1 \\ \frac{1}{2} \end{pmatrix} \right). \quad (5.7)$$

Der Homotopieparameter λ lässt sich weiter erhöhen, wobei mit $(j_1, u_2)^T = (0, 0)^T$ die Komplementaritätsbedingung bis $\lambda = 1$ erfüllt bleibt. Gleichung (5.7) ergibt weiterhin $(u_1, j_2)^T = (2, 2)^T$, so dass im Ergebnis eine Lösung für die Vektoren \vec{j} und \vec{u} bezüglich Gleichung (5.5) vorliegt.

Betrachtet man das vollständige PWA-Zustandsmodell, lässt sich das oben skizzierte Verfahren wie folgt übertragen:

Zunächst wird definiert

$$\vec{q}_{start}(\vec{x}_{start}) = (\mathbf{C} - \mathbf{D} \cdot \mathbf{A}) \cdot \vec{x}_{start} + (\vec{g} - \mathbf{D} \cdot \vec{f}). \quad (5.8)$$

Die Herleitung erfolgt aus den Gleichungen des Zustandsmodells, wobei deren Gültigkeit für $\vec{u} = \vec{0}$ angenommen wird:

$$\vec{0} = \mathbf{I}_{m \times m} \vec{y} + \hat{\mathbf{A}} \vec{x} + \hat{\mathbf{B}} \vec{u} + \vec{f} \quad (5.9)$$

$$\vec{j} = \mathbf{I}_{k \times k} \vec{u} + \mathbf{C} \vec{x} + \mathbf{D} \vec{y} + \vec{g} \quad (5.10)$$

$$\Rightarrow \vec{j} = \underbrace{\mathbf{I}_{k \times k}}_M \vec{u} + \underbrace{(\mathbf{C} - \mathbf{D} \cdot \mathbf{A}) \cdot \vec{x} + (\vec{g} - \mathbf{D} \cdot \vec{f})}_{\vec{q}_{start}} \quad (5.11)$$

Für das LCP existiert immer eine eindeutige Lösung, sofern die Matrix \mathbf{M} positiv definit ist [CoPa92]. Eine quadratische symmetrische bzw. hermitesche Matrix ist positiv definit, falls alle Eigenwerte größer als Null sind. Da \mathbf{M} für das PWA-Zustandsmodell per Definition eine Einheitsmatrix ist, ist diese Forderung immer erfüllt. Das vorgestellte Homotopieverfahren führt somit immer zu einer Lösung und lässt sich über folgende Formulierung durchführen:

$$\begin{aligned} \vec{j} &= \underline{\mathbf{I}} \cdot \vec{u} + \vec{q}_{start} + \lambda(\vec{q} - \vec{q}_{start}) \\ \Rightarrow \vec{j} &= \underline{\mathbf{I}} \cdot \vec{u} + (\mathbf{C} - \mathbf{D} \cdot \mathbf{A}) \cdot \vec{x}_{start} + (\vec{g} - \mathbf{D} \cdot \vec{f}) + \lambda(\mathbf{C} - \mathbf{D} \cdot \mathbf{A}) \cdot (\vec{x} - \vec{x}_{start}). \end{aligned} \quad (5.12)$$

Hierbei ist zu beachten, dass bei einer Vertauschung von j_p und u_p auch Gleichung (5.9) betroffen ist und entsprechend umformuliert werden muss. In Analogie zur Ersatzanordnung aus Abschnitt 4.2.1 für das PWA-Zustandsmodell wird von einem *Zustandswechsel* gesprochen. Die Systemmatrizen müssen entsprechend neu bestimmt werden, was im folgenden Abschnitt detailliert behandelt wird.

5.4.2 Neubestimmung nach Zustandswechsel

Die Neubestimmung der Systemmatrizen ist aufgrund einer Vertauschung der beiden Elemente u_p mit j_p aus den Zustandsvektoren notwendig. Für das p -te Element ergibt sich aus den Zustandsgleichungen:

$$j_p = \begin{pmatrix} d_{p1} & \dots & d_{pm} \end{pmatrix} \cdot \vec{y} + \begin{pmatrix} c_{p1} & \dots & c_{pn} \end{pmatrix} \cdot \vec{x} + u_p + g_p. \quad (5.13)$$

Folglich ist

$\mathbf{C}^* := \mathbf{C}$ mit negiertem Vorzeichen in p -ter Zeile

$\mathbf{D}^* := \mathbf{D}$ mit negiertem Vorzeichen in p -ter Zeile

$\vec{g}^* := \vec{g}$ mit negiertem Vorzeichen in p -ter Zeile

Setzt man Gleichung (5.13) in Gleichung (5.9) ein, ergibt sich

$$\vec{0} = \mathbf{I}_{m \times m} \vec{y} + \hat{\mathbf{A}} \vec{x} + \hat{\mathbf{B}} \begin{pmatrix} u_1 \\ \vdots \\ u_{p-1} \\ - (d_{p1} \ \dots \ d_{pm}) \cdot \vec{y} - (c_{p1} \ \dots \ c_{pn}) \cdot \vec{x} - g_i + j_i \\ u_{p+1} \\ \vdots \\ u_k \end{pmatrix} + \vec{f}$$

Gruppiert man zusammengehörige Ausdrücke, folgt weiter:

$$\vec{0} = \left(\mathbf{I}_{m \times m} - \hat{\mathbf{B}} \cdot \begin{bmatrix} 0 & \dots & 0 \\ d_{p1} & \dots & d_{pm} \\ 0 & \dots & 0 \end{bmatrix} \right) \vec{y} + \left(\hat{\mathbf{A}} - \hat{\mathbf{B}} \cdot \begin{bmatrix} 0 & \dots & 0 \\ d_{p1} & \dots & d_{pm} \\ 0 & \dots & 0 \end{bmatrix} \right) \vec{x} + \hat{\mathbf{B}} \begin{pmatrix} u_1 \\ \vdots \\ u_{p-1} \\ j_i \\ u_{p+1} \\ \vdots \\ u_k \end{pmatrix} + \vec{f}$$

Um das System wieder in die Standardform zu überführen, muss das System mit der Inversen

$$\mathbf{S} = \left(\mathbf{I}_{m \times m} - \hat{\mathbf{B}} \cdot \begin{bmatrix} 0 & \dots & 0 \\ d_{p1} & \dots & d_{pm} \\ 0 & \dots & 0 \end{bmatrix} \right)^{-1} \quad (5.14)$$

multipliziert werden. Mit $\vec{d}_i^T = (d_{p1} \ \dots \ d_{pm})$ und $\vec{b}_i^T = (b_{1i} \ \dots \ d_{mi})$ lässt sich die Berechnung von \mathbf{S} mit der Sherman-Morrison-Formel durchführen:

$$\mathbf{S} = (\mathbf{M} - \vec{b} \vec{d}^T) = \mathbf{M}^{-1} + \frac{1}{1 - \mathbf{M}^{-1} \vec{d}^T \vec{b} \mathbf{M}^{-1}} \mathbf{M}^{-1} \vec{b} \vec{d}^T \mathbf{M}^{-1} \quad (5.15)$$

Da $\mathbf{M}^{-1} = \mathbf{I}_{m \times m}$, ergibt sich $\mathbf{S} = \mathbf{I} + \frac{\vec{b} \vec{d}^T}{1 - \vec{d}^T \vec{b}}$.

Die neuen Systemmatrizen ergeben sich somit zu:

$$\mathbf{A}^* = \mathbf{S}(\hat{\mathbf{A}} - \vec{b} \vec{c}^T) \quad (5.16)$$

$$\mathbf{B}^* = \mathbf{S} \hat{\mathbf{B}} \quad (5.17)$$

$$\vec{f}^* = \mathbf{S}(\vec{f} - \vec{b} \vec{g}^T) \quad (5.18)$$

5.4.3 Umsetzung und Steuerung der Simulation

Die Auswertung des PWA-Zustandsmodells bei gegebenen Eingangsvektor \vec{x} wird in *Sur-P.A.S.S.* über die Funktion `PWASolve` vorgenommen. Neben dem `PWLSyst`-Objekt und dem Eingangsvektor erhält die Funktion einen Startvektor \vec{x}_{start} , der als Anfangslösung für das in Abschnitt 5.4.1 vorgestellte Homotopieverfahren dient.

Ist der Startvektor für das gegebene `PWLSyst`-Objekt unbekannt, lässt sich dieser bestimmen durch Finden eine Lösung für die Ungleichung

$$(\mathbf{C} - \mathbf{D} \cdot \mathbf{A}) \cdot \vec{x}_{start} + (\vec{g} - \mathbf{D} \cdot \vec{f}) \geq \vec{0}, \quad (5.19)$$

was in *Sur-P.A.S.S.* durch die Funktion `FindStartVector[_PWLSyst]` ausgeführt werden kann.

Der in `PWASolve` implementierte Algorithmus lautet wie folgt:

Algorithmus 3 : PWASolve

```

Eingabe :  $\vec{x}_{start}$ ,  $\vec{x}$ , PWA-System
Ausgabe : Ausgangsvektor  $\vec{y}$ , neues PWA-System
/* Nominale Lösung finden */
wenn keine Verletzung der Komplementarität für  $\vec{u} = \vec{0}$  dann
  | Systemmatrizen beibehalten
sonst
  | Neue Systemmatrizen bestimmen
Ende
;
/* Unsicherheiten berücksichtigen */
wenn  $\vec{j}_{neu}$  negative Werte einschließt dann
  | Matrizen des Transitionssystem bestimmen
Ende
;

```

Da bei einer Simulation der Eingangsvektor oft in nur kleinen Schritten geändert wird, ist die Wahrscheinlichkeit hoch, dass sich der Ausgangsvektor aus dem gegebenen System über

$$\vec{y} = -\hat{\mathbf{A}}\vec{x} - \vec{f} \quad (5.20)$$

bestimmen lässt, sich \vec{x} also im selben affinen Teilbereich befindet wie \vec{x}_{start} . Um die Gültigkeit des aktuellen Zustands zu überprüfen, wird der Vektor \vec{j} mit Hilfe der Zustandsgleichung (4.2) ermittelt. Die Betrachtung erfolgt dabei nur mit den nominalen Größen von \vec{x} . Um die Komplementaritätsbedingung zu erfüllen, muss gelten:

$$j_i \geq 0, \forall i \in \{1, 2, \dots, k\} \quad \text{mit} \quad \vec{u} = \vec{0}. \quad (5.21)$$

Befindet sich das System in keinem gültigen Zustand, wird das Homotopieverfahren entsprechend Abschnitt 5.4.1 durchgeführt, wobei die neuen Koeffizientenmatrizen entsprechend dem Verfahren in Abschnitt 5.4.2 ermittelt werden.

Nachdem sichergestellt ist, dass für den nominalen Fall ein gültiges System vorliegt, wird überprüft, ob eine Approximation entsprechend Abschnitt 4.2.3 notwendig ist. Indikator hierfür ist mindestens ein negativer Eintrag $j_{neg,i}$ im Zustandsvektor $\vec{j}_{neg} = \vec{j}_0 - \text{rad}(\hat{j})$, wobei diesmal alle affinen Formen sowohl in den Koeffizientenmatrizen, als auch im Eingangsvektor ausgewertet werden.

Um die Antwort des PWA-Zustandsmodells auf kontinuierliche Verläufe der Eingangserregung \vec{x} , wie beispielsweise bei einer DC-Simulation, zu ermitteln, bietet *Sur-P.A.S.S.* die Funktion `SolvePWASyst` an. Hier wird der gewählte Eingangsbereich für eine oder mehrere Veränderliche abgefahren und für jede Eingangsbelegung von \vec{x} die Funktion `PWASolve` aufgerufen. Der jeweilige Rückgabewert und das eventuell aktualisierte PWA-Zustandsmodell werden für den Folgeaufruf übernommen.

Im Anschluss an die Simulation wird das Ergebnis sowohl im eindimensionalen Fall, als auch im zweidimensionalen Fall graphisch ausgegeben (siehe Simulationsergebnisse in Kapitel 6).

6 Modellierungs- und Simulationsergebnisse

In diesem Kapitel werden einige Schaltungsbeispiele mit Hilfe des vorgestellten Werkzeugs *Sur-P.A.S.S.* behandelt. Dabei werden die in den vorangegangenen Kapiteln angeführten Modellierungsmethoden verwendet.

6.1 Beispiel mit zwei nicht idealen Dioden

Als erstes Beispiel für eine Simulation auf elektrischer Ebene wird die Schaltung gemäß Abbildung 6.1(a) betrachtet.

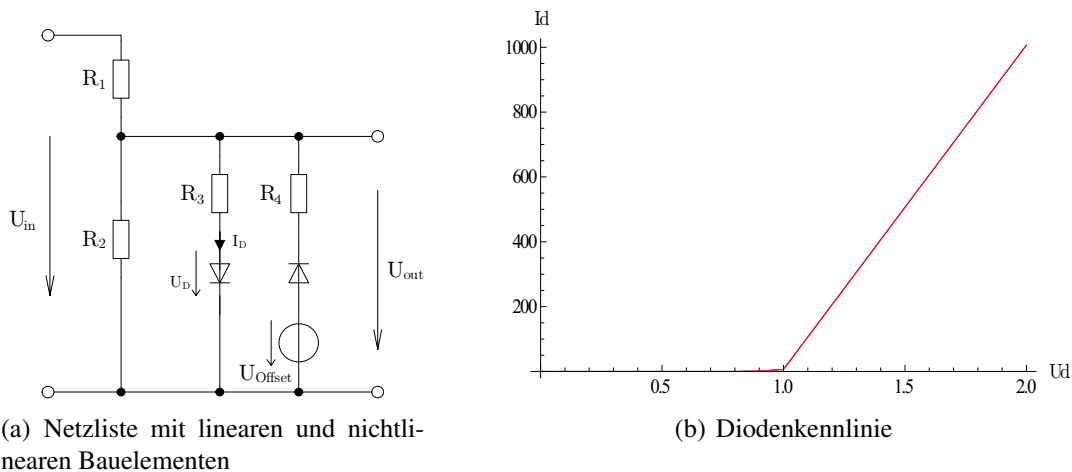


Abbildung 6.1: Semi-symbolische Simulation einer Beispielschaltung

Die Schaltung besitzt zwei nichtlineare Bauelemente (nicht-ideale Dioden), deren jeweils identische Kennlinie in Abbildung 6.1(b) dargestellt ist. Die Zustandsvektoren des Diodenmodells besitzen die Größe $k = 6$.

Die beiden Dioden gehen entsprechend Abschnitt 5.3.4 als nichtlineare Zweipole in eine *Analog-Insydes*-Netzliste ein. Einer der beiden Diodenzweige ist über eine ideale

Spannungsquelle vorgespannt. Aus der Netzliste wird nun mit Hilfe von *Analog-Insydes* ein DAE-Objekt erzeugt, aus dem abschließend das PWA-Zustandsmodell der Gesamtschaltung mit $k = 12$ gewonnen wird. Eingangsgrößen des neuen Modells sind U_{Offset} und U_{in} .

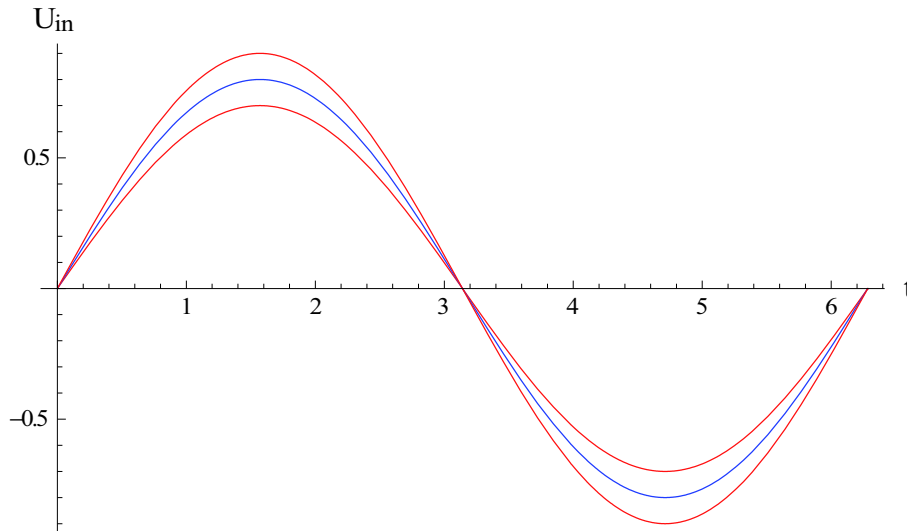


Abbildung 6.2: Mit Unsicherheiten behaftete Eingangserregung

Für die semi-symbolische Simulation wird eine sinusförmige Eingangserregung entsprechend Abbildung 6.2 mit

$$U_{in} = (0.8 \text{ [V]} + 0.1 \text{ [V]} \cdot \varepsilon_1) \cdot \sin(\omega t) \quad \text{mit } \omega = 2\pi \cdot 1 \text{ Hz}$$

an das erste Klemmenpaar gelegt. In Abbildung 6.3 ist das Simulationsergebnis für drei unterschiedliche Offsetspannungen U_{Offset} dargestellt. Die Ergebnisse besitzen die erwartete Form.

6.2 Beispiel mit dynamischem Verhalten

Für die nächste untersuchte Schaltung wird eine semi-symbolische Transientenanalyse durchgeführt. Der Aufbau der Schaltung ist in Abbildung 6.4 dargestellt. Aus der Netzliste wird, wie im vorherigen Beispiel, mit Hilfe von *Analog-Insydes* ein DAE-Objekt erzeugt.

Das zeitabhängige Eingangssignal $U_{in}(t)$ ist ein Rechtecksignal, wobei die Amplitude des High-Pegels $\hat{U}_{in,high} = 1 \text{ V} + 0.1 \text{ V} \cdot \varepsilon_1$ beträgt. Wie bei einer Standard-Transientenanalyse von nichtlinearen Netzwerken, wird eine Integrationsformel (hier: Backward-Euler) genutzt, um die zeitlichen Ableitungen aus dem Gleichungssystem zu eliminieren (siehe Abschnitt 3.1.2). Auf diese Weise wird der durch die Kapazität

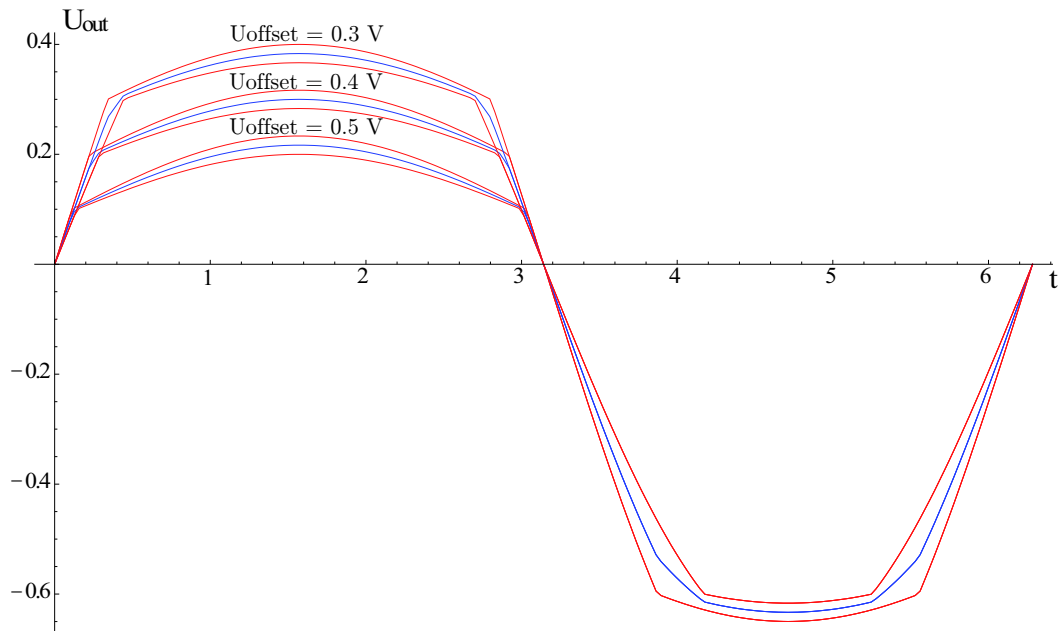


Abbildung 6.3: Ergebnis der semi-symbolischen Simulation für unterschiedliche Offsetspannungen U_{offset}

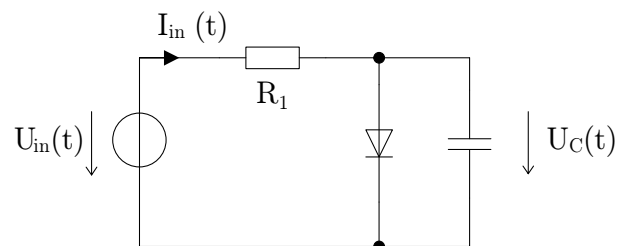


Abbildung 6.4: Beispielschaltung mit Diode und Kapazität zur semi-symbolischen Simulation im Zeitbereich

bewirkte Ausdruck dU_C/dt ersetzt, so dass sich ein PWA-Zustandsmodell in der hier vorgestellten Form erzeugen lässt.

Abbildung 6.6 zeigt den zeitlichen Verlauf der Spannung $U_C(t)$. Nach der kurzen Einschwingphase bis $t \approx 3$ s ist ein geglättetes Signal zu erkennen, welches periodisch der Eingangserregung folgt und die zu erwartende Form aufweist. Die Unsicherheiten aus der Amplitude $\hat{U}_{in,high}$ finden sich ebenfalls wieder. Abbildung 6.7 zeigt den Strom, der von der Quelle U_{in} ausgeht. Auch hier zeigt sich der erwartete Verlauf inklusive Unsicherheit in semi-symbolischer Form.

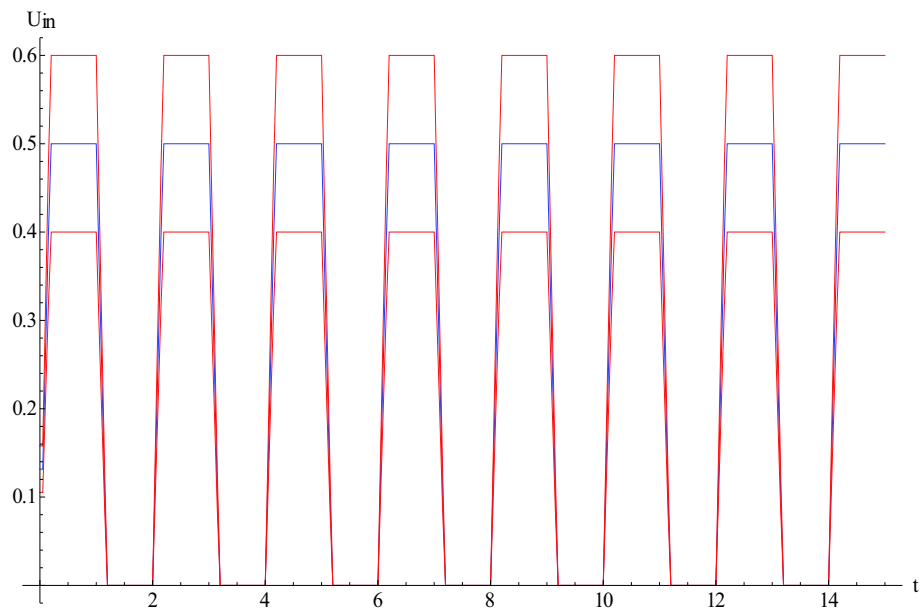
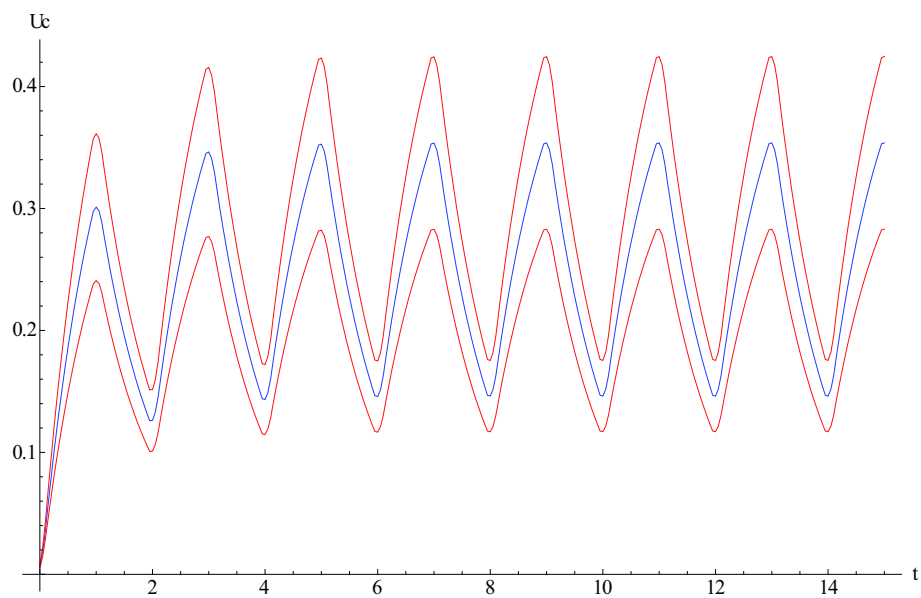


Abbildung 6.5: Eingangserregung mit Unsicherheiten

Abbildung 6.6: Ergebnis der semi-symbolischen Transientensimulation für die Spannung $U_C(t)$

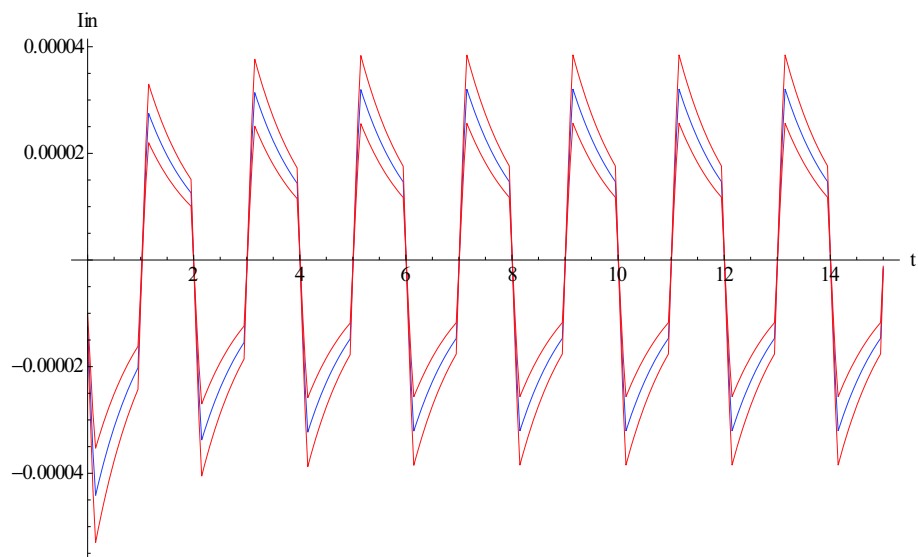


Abbildung 6.7: Ergebnis der semi-symbolischen Transientensimulation für den Strom $I_{in}(t)$

6.3 PWA-Transistormodelle

Im Folgenden werden zwei integrierte Feldeffekttransistoren untersucht. Hierzu werden die in Kapitel 4 vorgestellten Modellierungsverfahren eingesetzt. Für den ersten Transistor erfolgt die Modellierung mit Hilfe von Simulationsdaten, wie in Abschnitt 4.6 beschrieben. Für den zweiten Transistor wird ein PWA-Zustandsmodell entsprechend Abschnitt 4.5 mit Hilfe eines algebraischen Verhaltensmodells des Transistors gewonnen.

6.3.1 Transistor aus Hinging-Hyperplanes

Grundlage für die Modellierung bildet ein mit Hilfe eines Schaltungssimulators gewonnener Satz von Datenpunkten. Die Daten wurden mit der Schaltung aus Abbildung 6.8 erzeugt, indem mehrere DC-Transfer-Simulationen über die beiden Eingangsgrößen U_{DS} mit $U_{DS} = \{0 \text{ V} \dots 1.5 \text{ V}\}$ und U_{GS} mit $U_{GS} = \{0 \text{ V} \dots 1.5 \text{ V}\}$ durchgeführt wurden.

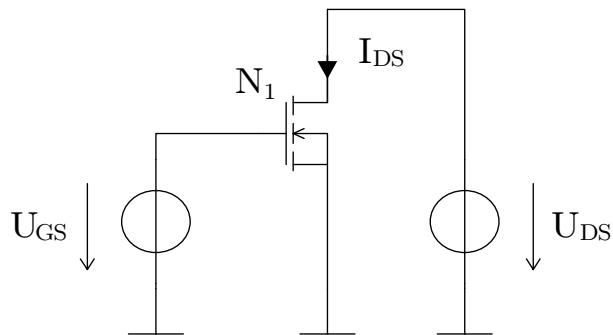


Abbildung 6.8: Schaltung zur Charakterisierung des DC-Verhaltens eines integrierten Feldeffekttransistors

Als Ausgangsgröße wurde der Drainstrom I_{DS} des Transistors gewählt. Die resultierenden Datenpunkte sind in Abbildung 6.9 dargestellt.

Aus diesem Datensatz wurde des Weiteren über die Funktion FindHingeFit eine Liste von sechs Hinge-Objekten gewonnen:

```
{
HingeObj[Max,{{-0.000833261, 0.00119173, 0.000315286},
              {-0.000364922, 0.000707548, 0.000265167}}],
HingeObj[Min,{{0.000204353, -0.000347215, 0.000464957},
              {0.0000580928, 0.000109854, -0.000124497}}],
HingeObj[Min,{{0.000261339, -0.000393211,
```

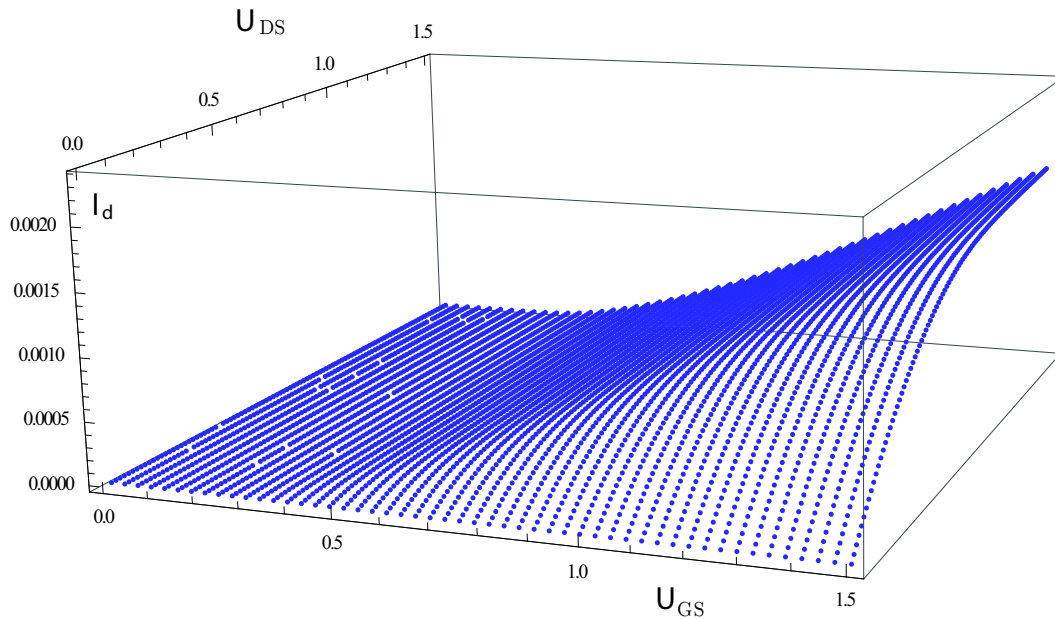


Abbildung 6.9: Datenpunkte zur Charakterisierung des DC-Verhaltens eines NMOS-Transistors

```

    0.000655594}, {0.0000322612, 0.0000618101, -0.0000724998}}],
HingeObj[Min,{{0.0000137, 0.0000309506, -0.0000350584},
               {0.000495639, -0.000572596, 0.00113021}}],
HingeObj[Max,{{-0.000050941, 0.0000475764, 1.59199*10^-6},
               {0.0000820179, -0.000368417, -4.56353*10^-6}}],
HingeObj[Max,{{-0.000222136, 0.000195612, 0.0000142153},
               {0.000177681, -0.000529433, -0.0000282522}}]
}

```

In Abbildung 6.10 sind die resultierenden Hyperebenen gemeinsam mit den Datenpunkten über den Eingangsgrößen aufgetragen. Die Approximation gibt die Charakteristik des Transistors gut wieder. Der dabei gemachte Approximationsfehler

$$f_{err}(U_{GS}, U_{DS})$$

ist in Abbildung 6.11 dargestellt.

Des Weiteren lässt sich eine Einschlussfunktion für den Approximationsfehler defi-

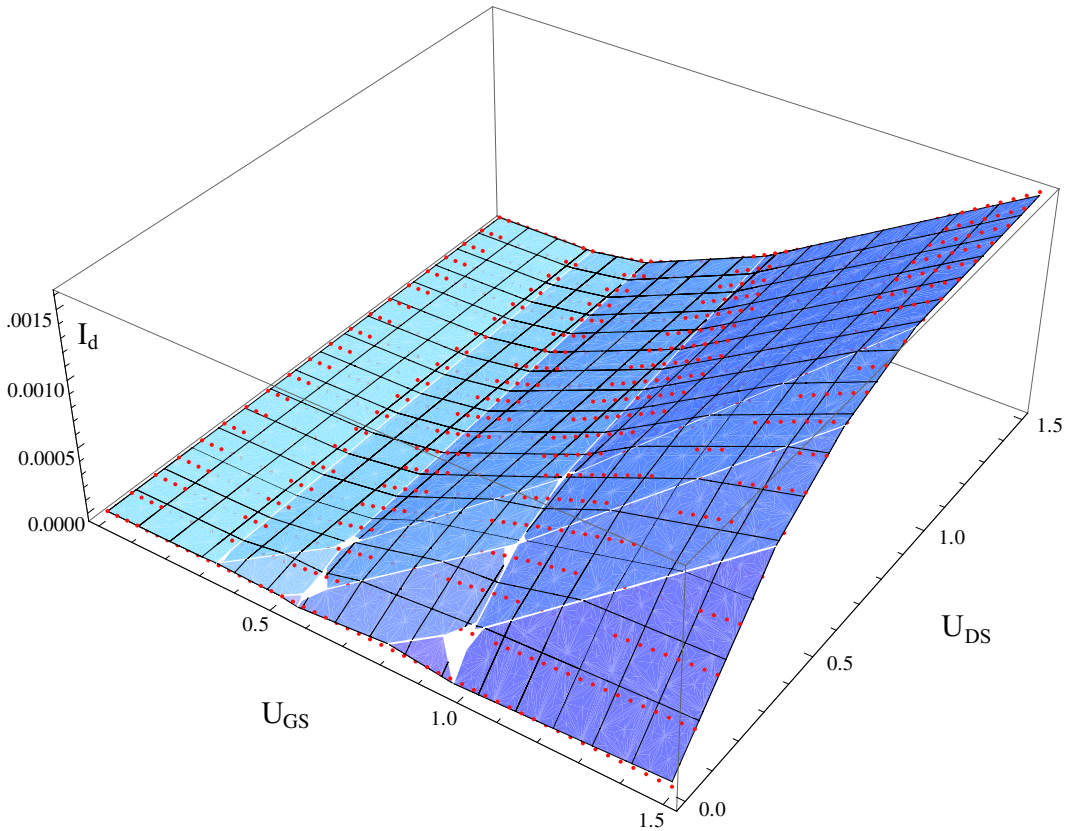


Abbildung 6.10: Verhalten des NMOS-Transistors modelliert durch sechs Hinge-Funktionen

nieren durch

$$\hat{f}_{extend}(U_{GS}, U_{DS}) = (0.0000104 \cdot U_{GS} - 0.000008 \cdot U_{DS} + 0.000022) \cdot \epsilon_1.$$

Die Grenzen dieser Einschlussfunktion

$$\begin{aligned} \underline{\hat{f}}_{extend} &= -\text{rad}(\hat{f}_{extend}(U_{GS}, U_{DS})) \\ \overline{\hat{f}}_{extend} &= \text{rad}(\hat{f}_{extend}(U_{GS}, U_{DS})) \end{aligned}$$

sind ebenfalls in Abbildung 6.11 eingetragen. Zusammengefasst ergibt sich ein PWA-Zustandsmodell mit Zustandsvektoren der Größe $k = 6$, mit dem abschließend eine semi-symbolische DC-Transfersimulation durchgeführt wurde (siehe Abbildung 6.12). Die Systemmatrizen lauten:

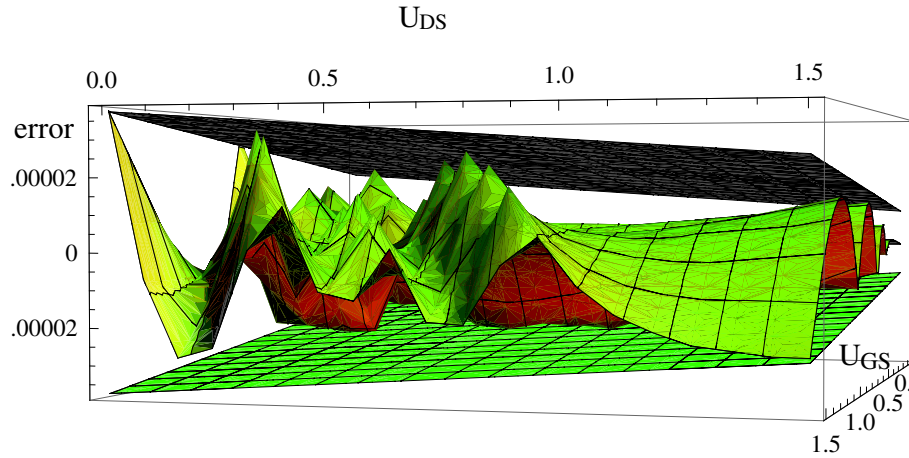


Abbildung 6.11: Approximationsfehler mit symmetrischer Einschlussfunktion

$$\mathbf{A} = \left(-0.000012312 + 0.0000104 \cdot \varepsilon_1 \quad -2.956324879 \cdot 10^{-7} - 8 \cdot 10^{-6} \cdot \varepsilon_1 \right),$$

$$\mathbf{B} = \begin{pmatrix} -1 & 1 & 1 & 1 & -1 & -1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} -0.0004842 & -0.0000501 \\ -0.0004571 & 0.0005895 \\ -0.000455 & 0.0007281 \\ -0.0006035 & 0.0011653 \\ -0.000416 & -6.2 \cdot 10^{-6} \\ -0.000725 & -0.0000425 \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{f} = \left(1.16925143 \cdot 10^{-6} + 0.000022 \cdot \varepsilon_1 \right), \quad \vec{g} = \begin{pmatrix} 0.0004683 \\ 0.0001463 \\ 0.0002291 \\ 0.0004819 \\ 0.000133 \\ 0.0003998 \end{pmatrix}.$$

6.3.2 Beispiel MOSFET-Level-1-Modell

In diesem Abschnitt erfolgt die Modellierung auf Grundlage eines algebraischen Verhaltensmodells. Das Verhaltensmodell basiert auf dem Modell von Shichman und Hodges [ShHo68] (MOSFET-Level-1-Modell). Es unterscheidet drei Arbeitsbereiche des Transistors, die sich durch die Spannungsdifferenzen zwischen Gate, Source und Drain definieren. Ist die Gate-Source-Spannung U_{GS} kleiner als die Schwellspannung U_{th} , kann sich kein leitender Kanal ausbilden. Der Transistor befindet sich im

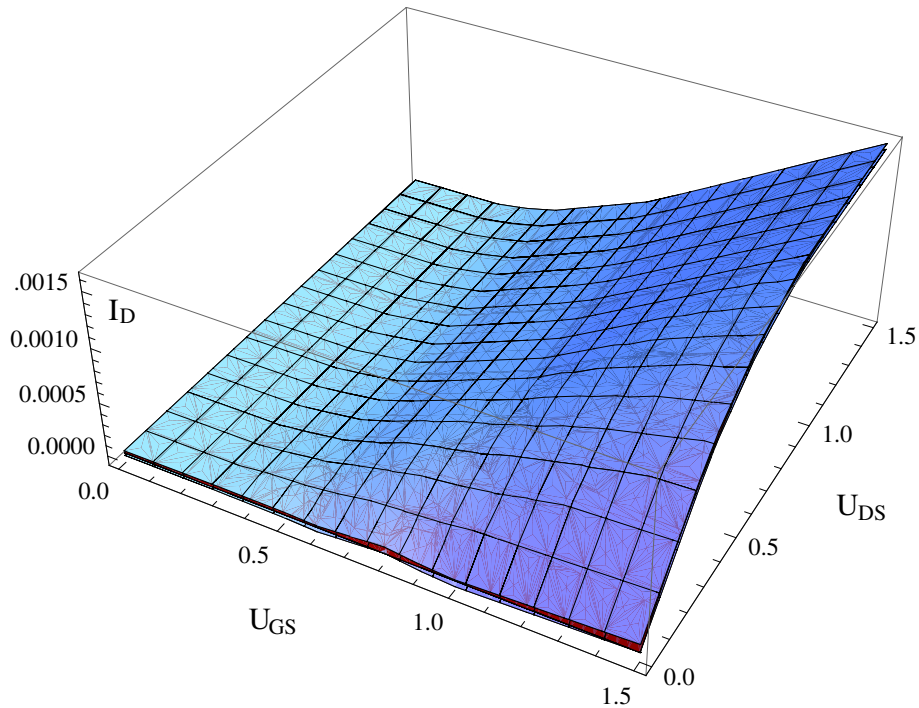


Abbildung 6.12: Ergebnis der Simulation des PWA-Zustandsmodells inklusive der durch \hat{f}_{extend} modellierten Unsicherheiten

Sperrbereich. Der Drainstrom I_D ist hier unabhängig von der Drain-Source-Spannung U_{DS} . Überschreitet U_{GS} die Schwellspannung U_{th} , bildet sich ein Kanal, durch den ein Strom I_D fließen kann. I_D ist dabei für kleine U_{DS} näherungsweise proportional zu U_{DS} . Dieser Bereich wird daher als linearer Bereich bezeichnet. Wird U_{DS} über $U_{DS,sat}$ und U_{GS} über U_{th} erhöht, erfolgt auf der Drainseite des Kanals eine Abschnürung. I_D steigt in diesem sogenannten Sättigungsbereich nur noch sehr schwach an. Zusammengefasst ergibt sich der Drain-Strom I_D als

$$I_D = \begin{cases} K_n \frac{W}{L} U_{DS} (U_{GS} - U_{th} - \frac{U_{DS}}{2}) (1 + \lambda U_{DS}) & \text{für } U_{GS} \geq U_{th}, 0 \leq U_{DS} \leq U_{DS,sat} \\ K_n \frac{W}{2L} (U_{GS} - U_{th})^2 (1 + \lambda \cdot U_{DS}) & \text{für } U_{GS} \geq U_{th}, U_{DS} \geq U_{DS,sat} \\ 0 & \text{für } U_{GS} < U_{th} \end{cases} \quad (6.1)$$

Für die Parameter aus Gleichung (6.1) wird nun $K_n = 522.1 \cdot 10^{-6}$, $W = 10 \mu\text{m}$, $L = 10 \mu\text{m}$, $\lambda = 360 \cdot 10^{-3}$ und $U_{th} = 252.2 \text{ mV}$ eingesetzt. Mit Hilfe der im vorherigen Abschnitt vorgestellten Separation zerlegt sich die Gleichung für den linearen Betrieb

in

$$I_d = 50000 \cdot y_1^2 - 50000 \cdot y_6^2 - 1.36294 \cdot 10^{-12} U_{DS}^2 \quad (6.2a)$$

$$y_1 = y_2 + 5.221 \cdot 10^{-9} U_{DS} \quad (6.2b)$$

$$y_2 = -\frac{1}{2} + \frac{y_3^2}{2} - \frac{y_5^2}{2} - \frac{9}{25} U_{DS} - \frac{81}{1250} U_{DS}^2 \quad (6.2c)$$

$$y_3 = 1 + y_4 + \frac{9}{25} U_{DS} \quad (6.2d)$$

$$y_4 = -0.2522 - \frac{U_{DS}}{2} + U_{GS} \quad (6.2e)$$

$$y_5 = -0.2522 - \frac{U_{DS}}{2} + U_{GS} \quad (6.2f)$$

$$y_6 = -\frac{1}{2} + \frac{y_7^2}{2} - \frac{y_9^2}{2} - \frac{9}{25} U_{DS} - \frac{81}{1250} U_{DS}^2 \quad (6.2g)$$

$$y_7 = 1 + y_8 + \frac{9}{25} U_{DS} \quad (6.2h)$$

$$y_8 = -0.2522 - \frac{1}{2} U_{DS} + U_{GS} \quad (6.2i)$$

$$y_9 = -0.2522 - \frac{1}{2} U_{DS} + U_{GS} \quad (6.2j)$$

Wie in Abschnitt 4.5.2 bereits allgemein formuliert, setzen sich die Zeilen in den Gleichungen (6.2) jeweils aus einer Summe von nichtlinearen Funktionen zusammen, die nur von einer Variablen abhängen.

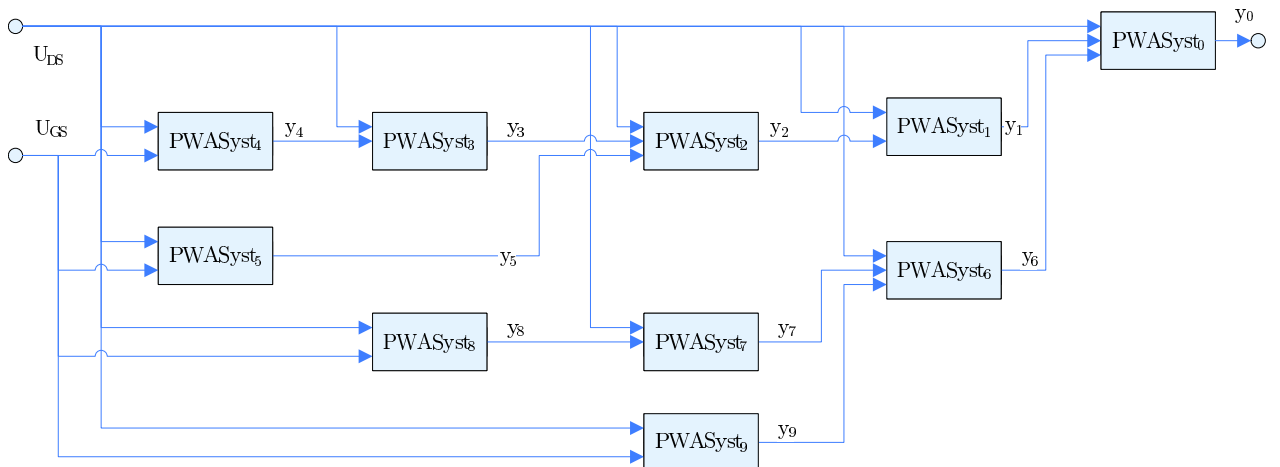


Abbildung 6.13: Hierarchie der stückweise linearen Funktionen für den linearen Betriebsbereich

Ein- und Ausgangsgrößen dieser PWA-Zustandsmodelle stehen durch die Gleichungen in (6.2) zueinander in Beziehung, so dass letztlich die Größe I_d durch eine hierarchische Verknüpfung (siehe Abbildung 6.13) mehrerer Modelle (hier $PWASyst_1$ bis

$PWAS_{yst9}$) gebildet wird. Diese PWA-Modelle wurden mit der rasterbasierten Methode und einer Summation gewonnen.

Die Hierarchie des Gesamtmodells kann im Weiteren für die Simulation (siehe Kapitel 5.4) erhalten bleiben. Zur Bestimmung von y_0 müssen dabei lediglich die Hilfsvariablen y_i mit $i = \{1 \dots 9\}$ schrittweise von links nach rechts ausgewertet werden. Analog zu dem Vorgehen in Kapitel 4.4 lässt sich jedoch aus einer hierarchischen Anordnung auch eine flache Darstellung herleiten. Anstelle der Strukturinformationen aus der Netzliste bildet sich die Konnektivitätsmatrix aus den Gleichungen (6.2a - 6.2j).

Für den Sättigungsbereich des Transistors ist das Vorgehen bei der Modellgewinnung gleich. Nach der Separation ergibt sich

$$\begin{aligned} I_d &= -0.000131053 + 0.000130525 y_1^2 - 0.000093978 U_{DS} \\ &\quad - 0.000016916 U_{DS}^2 + 8.37508 \cdot 10^{-6} U_{GS} - 0.0000498121 U_{GS}^2 \\ &\quad + 0.000131674 U_{GS}^3 - 0.000130525 U_{GS}^4 \\ y_1 &= 1.0636 + \frac{9}{25} U_{DS} - 0.5044 U_{GS} + U_{GS}^2 \end{aligned}$$

unter Verwendung derselben Technologie- und Dimensionierungsparameter. Auch hier kann für beide Gleichungen jeweils ein PWA-Zustandsmodell ermittelt werden.

In Abbildung 6.14 ist das Ergebnis beider Approximationen dargestellt. Dabei wurde der Drainstrom I_d des zusammengefassten Gesamtsystems mit $0 \text{ V} \leq U_{DS} \leq 1.5 \text{ V}$ und $0 \text{ V} \leq U_{GS} \leq 1.5 \text{ V}$ ermittelt. Der ebenfalls über den Eingangsgrößen aufgetragene Approximationsfehler ist in Abbildung 6.15 dargestellt. Wie ersichtlich, alterniert der Fehler um eine Abweichung von Null. Der maximale absolute Fehler beträgt $36 \mu\text{A}$, was bezogen auf einen Drainstrom $I_d = 400 \mu\text{A}$ einen relativen Fehler von 9% ergibt. Die betragsmäßig größeren Abweichungen ergeben sich dabei im linearen Bereich, was in der stärkeren Verknüpfung der PWA-Systeme begründet ist.

Jedes PWA-System wurde nach der rasterbasierten Methode gewonnen, wobei jeweils ein absoluter Fehler $absErr = 1\% \cdot (y_{i,max} - y_{i,min})$ zugelassen wurde, was im Durchschnitt zu 13 Segmenten pro Eingangsvariable geführt hat.

Prinzipiell lässt sich durch einen kleineren zulässigen Fehler für die Einzelsysteme auch eine höhere Genauigkeit des Gesamtsystems erzielen, was im Gegenzug jedoch eine stärkere Partitionierung und somit größere Systemmatrizen des PWA-Zustandsmodells zur Folge hat.

6.4 Sourceschaltung

Als weitere Schaltung wird die Sourceschaltung aus Abbildung 6.16 untersucht. Für

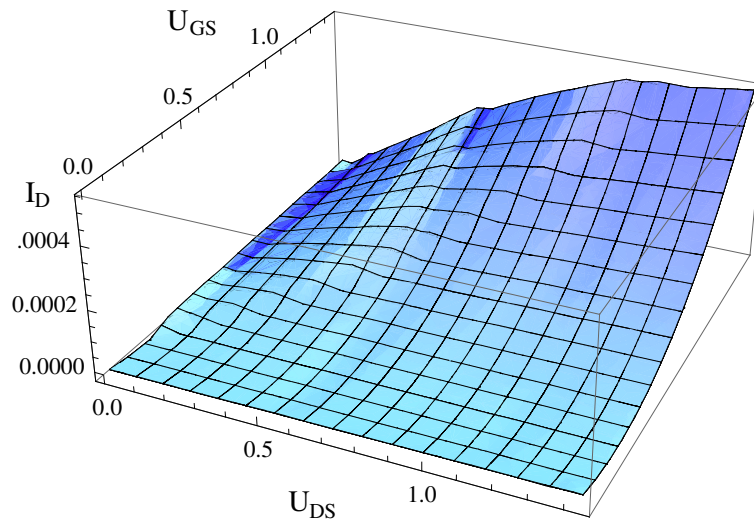


Abbildung 6.14: Durch Separation und anschließende Approximation gewonnene Kennlinie eines MOS-FET als PWA-Modell

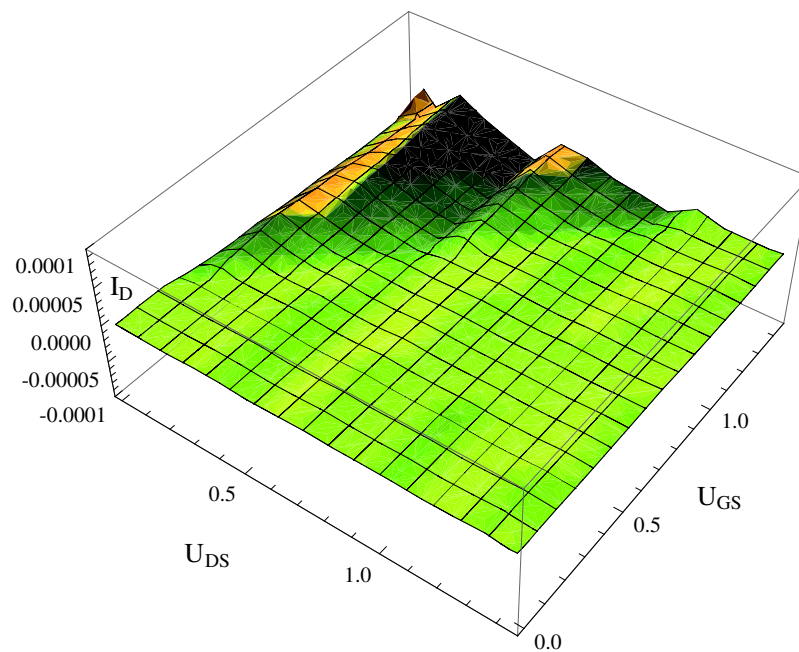


Abbildung 6.15: Fehler durch Approximation

den verwendeten Transistor wird das in Abschnitt 6.3.1 ermittelte PWA-Zustandsmodell mit einer Dimension von $k = 6$ für die beiden Zustandsvektoren \vec{u} und \vec{j} eingesetzt. Die Netzliste mit den beiden Widerständen R_1 und R_L wird, wie in den vorherigen Beispielen auch, mit Hilfe von *Analog-Insydes* in ein symbolisches DAE-System überführt. Aus diesem wird dann durch die Funktion `BuildPWLSyst` ein PWA-Zu-

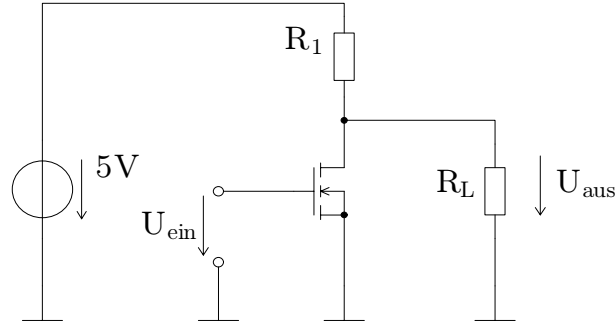


Abbildung 6.16: Sourceschaltung mit Transistor als PWA-Zustandsmodell

standsmodell berechnet, welches folgende Systemmatrizen besitzt:

$$\mathbf{A} = \left(\frac{0.0000246249 - 0.0000208 \cdot \varepsilon_1}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \right), \quad \mathbf{B}^T = \begin{pmatrix} \frac{2}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \\ -2 \\ \frac{2}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \\ -2 \\ \frac{2}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \\ -2 \\ \frac{2}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \\ \frac{2}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \\ \frac{2}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} -0.000484179 \\ -0.000457069 \\ -0.000455021 \\ -0.000603546 \\ -0.000415993 \\ -0.000725045 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} -0.0000501194 \\ 0.000589454 \\ 0.000728094 \\ 0.00116527 \\ -6.15552 \cdot 10^{-6} \\ -0.0000424675 \end{pmatrix},$$

$$\vec{f} = \left(-\frac{3/R_1 + 2.3385 \cdot 10^{-6} + 0.000044 \cdot \varepsilon_1}{2/R_1 + 2/R_L + 5.91265 \cdot 10^{-7} + 0.000016 \cdot \varepsilon_1} \right), \quad \vec{g} = \begin{pmatrix} 0.000468339 \\ 0.00014626 \\ 0.000229077 \\ 0.000481939 \\ 0.000132959 \\ 0.000399817 \end{pmatrix}.$$

Wie zu erkennen ist, besitzt das resultierende Modell der Sourceschaltung wiederum $k = 6$ Zustandsgleichungen. In Abbildung 6.17 ist das Ergebnis einer semi-symbolischen DC-Transfer-Simulation über einen Eingangsbereich von $U_{ein} = \{0 \text{ V} \dots 5 \text{ V}\}$ dargestellt. Dabei wurden Kurven für drei unterschiedliche Lastwiderstände erzeugt:

$$R_L = \{1 \text{ M}\Omega, 10 \text{ k}\Omega, 1 \text{ k}\Omega\}.$$

Der Widerstand $R_1 = 5 \text{ k}\Omega$ wurde mit einer Unsicherheit von $200 \text{ }\Omega$ versehen. R_1 geht also folglich mit der affinen Form

$$\hat{R}_1 = 5 \text{ k}\Omega + 200 \text{ }\Omega \cdot \varepsilon_1 \quad (6.3)$$

in das Modell ein. Die Ausgangskurven berücksichtigen somit einerseits die Unsicherheit aufgrund des Modellierungsfehlers beim Transistor, als auch andererseits die Unsicherheit durch die Abweichungen für \hat{R}_1 .

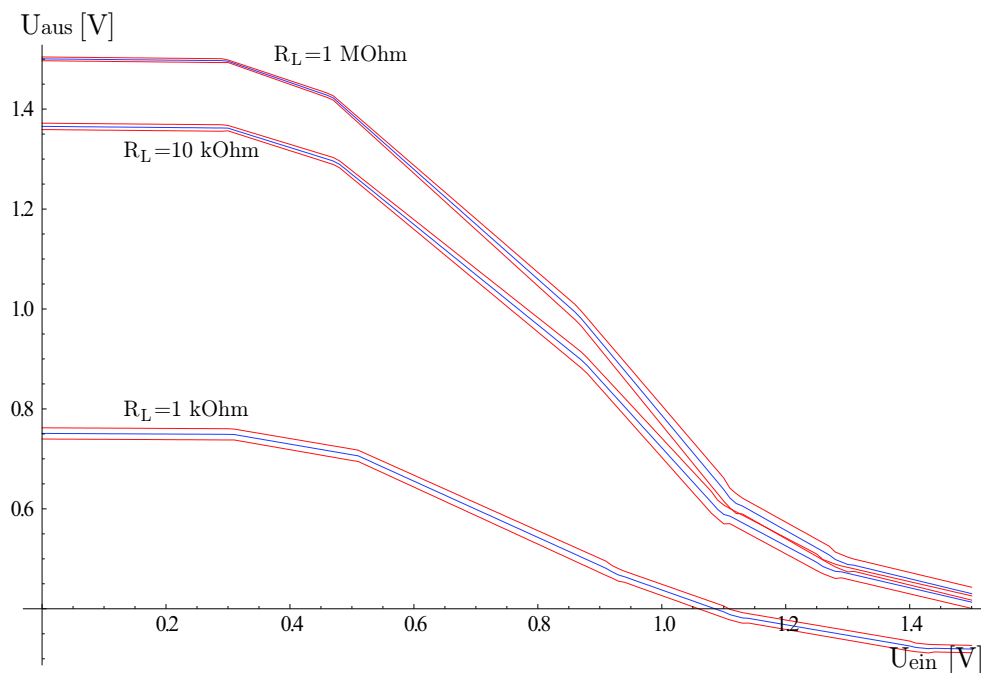


Abbildung 6.17: DC-Transfersimulation für eine Sourceschaltung

6.5 Operationsverstärkerstufe aus Transistorpaaren

In diesem Abschnitt wird die erste Stufe eines Operationsverstärkers gemäß Abbildung 6.18 betrachtet. Die beiden p-Kanal Transistoren P_0 und P_1 bilden einen Stromspiegel, die Transistoren M_0, M_1, N_0 und N_1 bilden zusammen eine Differenzpaar mit aktiver Last. Zur weiteren Modellierung werden diese beiden Funktionsblöcke getrennt voneinander betrachtet. Es ergibt sich ein Aufbau gemäß Abbildung 6.19.

Zur Modellierung der Differenzstufe wird zunächst wieder eine Simulation mit einem Standardsimulator durchgeführt, um den Funktionsblock zu charakterisieren. Die entsprechende Schaltung ist in Abbildung 6.20 dargestellt. Die Spannungen der Span-

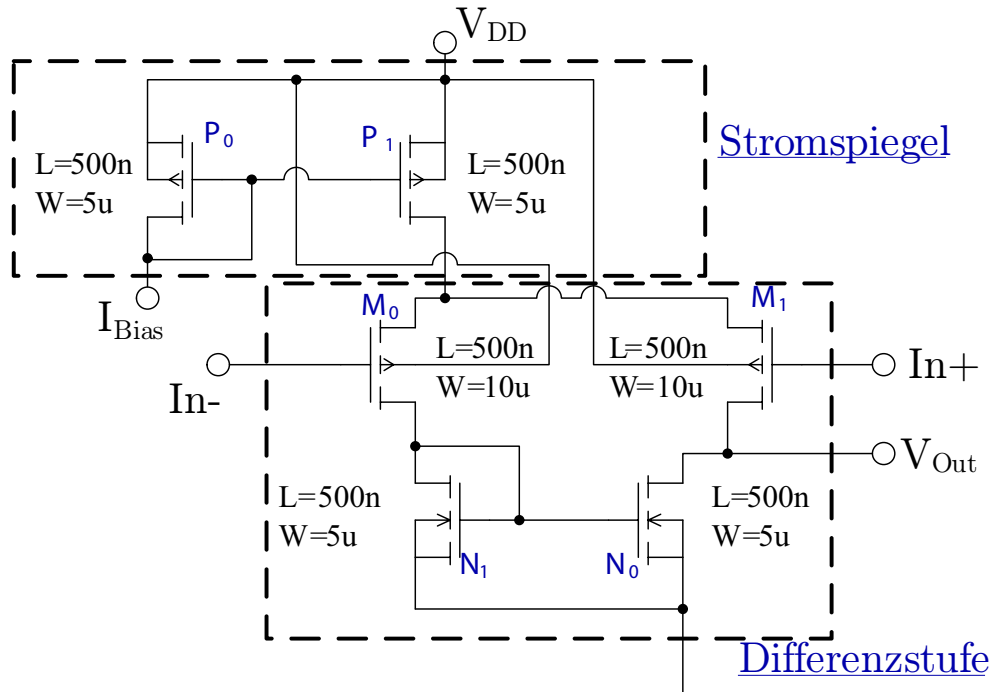


Abbildung 6.18: Erste Stufe eines Operationsverstärkers mit erkannten Transistorpaarstrukturen

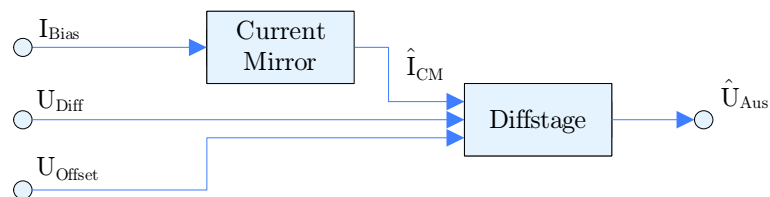


Abbildung 6.19: Funktionsblöcke der Operationsverstärkerstufe

nungsquellen V_0 und V_1 berechnen sich aus

$$V_{DC0} = U_{Offset} - \frac{U_{Diff}}{2}$$

$$V_{DC1} = U_{Offset} + \frac{U_{Diff}}{2}$$

Die variierenden Eingangsgrößen für die Simulation sind U_{Offset} als Gleichtaktspannung der Differenzstufe, U_{Diff} als Differenzspannung, sowie der Bias-Strom I_{CM} der Stromquelle I_6 . Die Spannung an der Klemme U_{out} bildet die Ausgangsgröße. Die erzeugten Ausgangskurven sind in Abbildung 6.21 skizziert. Aus den resultierenden Datenpunkten konnte mit Hilfe des Hinge-Modellierungsansatzes ein PWA-Zustandsmodell

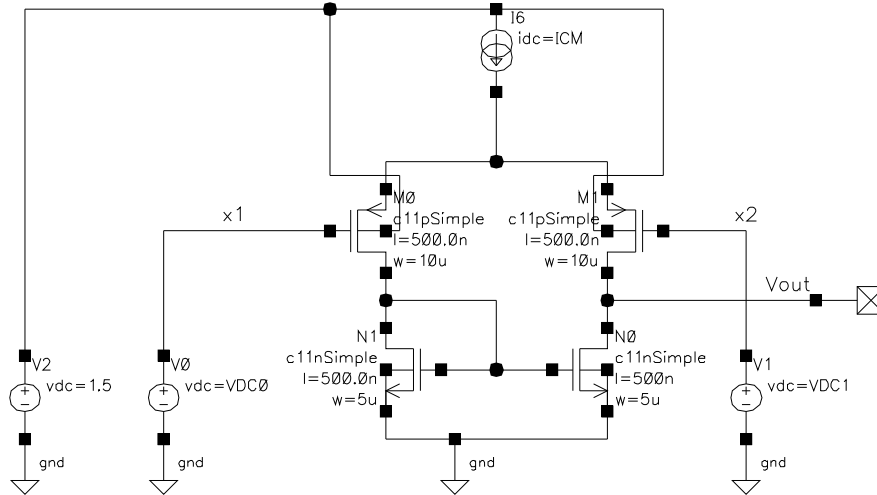


Abbildung 6.20: Schaltung zur Charakterisierung des Differenzstufe

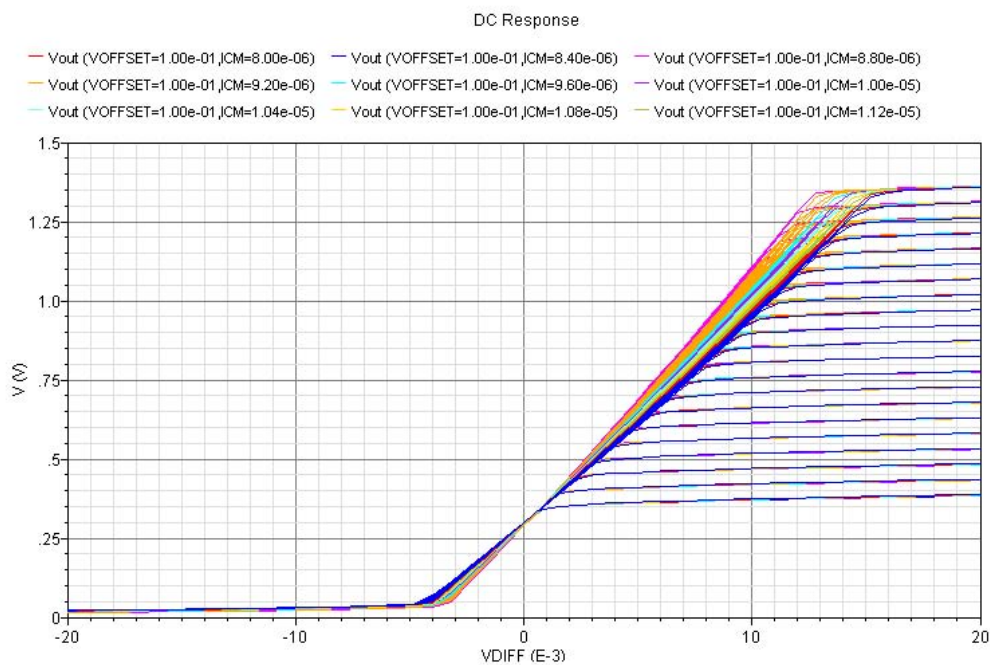


Abbildung 6.21: Parameterabhängige Ergebniskurven zur Charakterisierung

generiert werden, welches einen Eingangsvektor mit dem Rang 3 besitzt:

$$\vec{x} = \begin{pmatrix} U_{Diff} \\ U_{Offset} \\ I_{CM} \end{pmatrix}$$

In Abbildung 6.22 ist die zwei-dimensionale Ergebniskurve der Simulation mit *Sur-P.A.S.S.* dargestellt. Für den Biasstrom wurde dabei $I_{CM} = 10 \mu\text{A}$ festgelegt. Der ge-

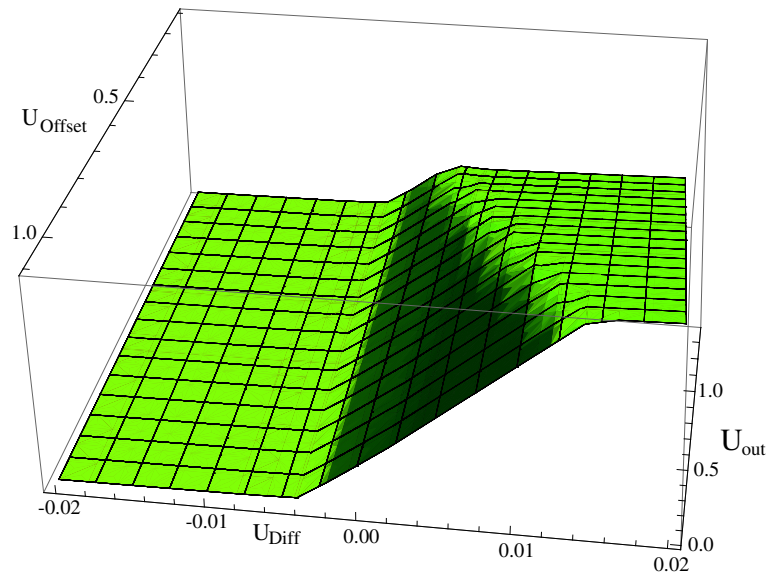


Abbildung 6.22: Ergebnis der semi-symbolischen Simulation für die Differenzstufe zeigte Verlauf entspricht dabei den Erwartungen.

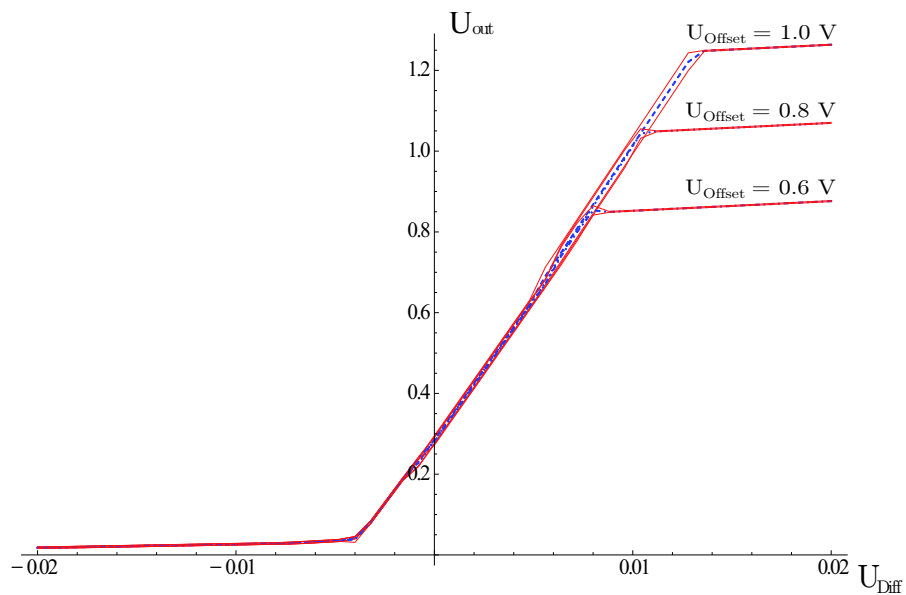


Abbildung 6.23: Ergebnis der semi-symbolischen Simulation für die gesamte Verstärkerstufe

Für den Stromspiegel sei nun angenommen, dass der Ausgangsstrom aufgrund von Parameterschwankungen mit Unsicherheiten behaftet ist. Dieser ist deshalb im Folgenden durch den affinen Ausdruck

$$I_{CM} = I_{Bias} + 1 \mu\text{A} \cdot \epsilon_1 \quad (6.4)$$

modelliert.

Mit $I_{Bias} = 10 \mu\text{A}$ ergeben sich aus der semi-symbolischen Simulation für die Gesamtschaltung die Ergebniskurven entsprechend Abbildung 6.23. Hierbei wurden die Kurven für drei unterschiedliche Offsetspannungen über dem Eingang U_{Diff} aufgenommen. Die Kurven entsprechen dabei dem zu erwartenden Ergebnis.

7 Zusammenfassung

Aufgrund der fortschreitenden Miniaturisierung in der Chipherstellung wird es immer wichtiger, den Einfluss von Prozessschwankungen auf das Verhalten von Schaltungen in die Verifikation mit einzubeziehen. Um die Unsicherheiten im Schaltungsverhalten zu berücksichtigen, werden in der Regel eine Vielzahl von Simulationsläufen durchgeführt (Monte-Carlo-Simulation). Je verlässlicher die Aussagen sein sollen, desto mehr Simulationsläufe müssen durchgeführt werden. Dies führt in der Konsequenz zu langen Simulationszeiten. Eine Möglichkeit, die Simulationen zu beschleunigen, besteht in mathematisch möglichst einfachen Modellgleichungen. Hierdurch ergeben sich jedoch zwangsläufig stärkere Abweichungen zwischen dem Modell und dem realen Schaltungsverhalten.

In dieser Arbeit wurden zwei unterschiedliche Verfahren zur semi-symbolischen Modellierung integrierter analoger Schaltungen auf Basis der affinen Arithmetik entwickelt. In dem ersten Verfahren wird das Modell aus einer algebraischen Verhaltensbeschreibung der Schaltung gewonnen. Diese Beschreibung kann dabei aus einer Netzliste und zuvor definierten Bauelementmodellen bestehen oder direkt gegeben sein. Das zweite Verfahren basiert auf einem Black-Box-Ansatz, bei dem das Modell anhand von zuvor gewonnenen Simulationsdaten ermittelt wird. Beide Verfahren erzeugen ein Modell, das sich in stückweise affine Bereiche aufgliedert.

Im Rahmen dieser Arbeit wurden die entwickelten Modellierungsverfahren für eine semi-symbolische Modellierung und ein semi-symbolischer Simulator softwaretechnisch umgesetzt und zu einem Paket mit der Bezeichnung *SUR-P.A.S.S.* zusammengefasst. Das Schaltungsverhalten wird dabei durch ein angepasstes stückweise affines Zustandsmodell wiedergegeben. Die Lösungsalgorithmen des Simulators wurden speziell an die Anforderungen eines mit Unsicherheiten behafteten stückweise affinen Zustandsmodells angepasst.

Durch die semi-symbolische Modellierung und Simulation ist es möglich, Unsicherheiten im Verhalten der Schaltung zu berücksichtigen, die einerseits durch den Einfluss von Prozessschwankungen, andererseits durch Abweichungen in den Modellgleichungen verursacht werden können. Erreicht wird dies durch affine Ausdrücke in den Modellparametern. Da für eine Simulation dieser Modelle, anders als bei einer Monte-Carlo-Simulation, nur ein einziger Simulationsdurchlauf nötig ist, erreicht man einen Geschwindigkeitsvorteil.

Anhand von mehreren Schaltungsbeispielen wurden die Modellierungsverfahren validiert. Mit dem entwickelten Simulator wurden weiterhin statische und eine dynamische semi-symbolische Simulation durchgeführt. Hierbei wurden mit Hilfe von affinen Ausdrücken, sowohl in den elektrischen Eingangserregungen, als auch in den Modellparametern, Unsicherheiten vorgegeben.

8 Literaturverzeichnis

- [AnCo94] M. V. A. Andrade, J. L. D. Comba, J. Stolfi, *Affine Arithmetic (Extended Abstract)*, in *INTERVAL '94, St. Petersburg*, 1994
- [AnGr93] K. J. Antreich, H. Gräß, E. Helmut, C. U. Wieser, *Practical Methods for Worst-Case and Yield Analysis of Analog Integrated Circuits*, *International Journal of High Speed Electronics and Systems*, Bd. 4(3), S. 261–282, 1993
- [BeRo69] R. Bellman, R. Roth, *Curve Fitting by Segmented Straight Lines*, *J. Amer. Statist. Assoc.*, Bd. 64, S. 1079–1084, 1969
- [BiHe04] T. Binder, C. Heitzinger, S. Selberher, *A Study on Global and Local Optimization Techniques for TCAD Analysis Tasks*, *IEEE Trans. on CAD of Integrated Circuits and Systems*, S. 814–822, 2004
- [BoCh74] G. Boyle, B. Chon, D. Pederson, *Macromodeling of IC OpAmp*, *IEEE Journal. of Solid-state Circ.*, Bd. 6, 1974
- [BoVa04] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press., 2004
- [Brei93] L. Breiman, *Hinging Hyperplanes for Regression, Classification, and Function Approximation*, in *IEEE Trans. Inf. Theory*, Bd. 39, S. 999–1013, 1993
- [BuLa98] M. Bucher, C. Lallement, C. Enz, F. Theodoloz, F. Krummenacher, *The EPFL-EKV MOSFET Model Equations for Simulation*, Technical report, Electronics Laboratories, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1998
- [Cad07] Cadence Design Systems, *Cadence Design Framework II User Guide - Product Version 6.1.0*, 2007
- [Cad08] Cadence Design Systems, *Virtuoso Spectre Circuit Simulator User Guide*, 2008
- [Cant71] A. Cantoni, *Optimal Curve Fitting With Piecewise Linear Functions*, *IEEE Trans. Comput.*, Bd. 20, Nr. 1, S. 59–67, 1971

- [ChDe85] L. Chua, A.-C. Deng, *Canonical Piecewise-Linear Analysis. II. Tracing Driving-Point and Transfer Characteristics*, IEEE Trans. Circuits and Systems, Bd. 32, Nr. 5, S. 417–444, 1985
- [ChDe87] L. Chua, C. Desoer, E. Kuh, *Linear and Nonlinear Circuits*, McGraw-Hill, New York, 1987
- [ChKa77] L. O. Chua, S. M. Kang, *Section-wise Piecewise Linear Functions: Canonical Representation, Properties and Applications*, in *Proc. IEEE*, Bd. 65, S. 915–929, 1977
- [CoPa92] R. W. Cottle, J.-S. Pang, R. E. Stone, *The Linear Complementary Problem*, Academic Press Inc., 1992
- [CoYa85] P. Cox, P. Yang, S. Mahant-Shetti, *Statistical Modeling for Efficient Parametric Yield Estimation of MOS VLSI Circuits*, in *IEEE Transactions on Electron Devices*, S. 471–478, 1985
- [Erns98] S. Ernst, *Hinging Hyperplane Trees for Approximation and Identification*, in *Proceedings of the 37th IEEE Conference on Decision and Control*, Bd. 2, S. 1266–1271, 1998
- [FeDi91] P. Feldmann, S. Director, *A Macromodeling Based Approach for Efficient IC Yield Optimization*, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, S. 2260–2263, 1191
- [FeSc92] U. Feldmann, R. Schulz, U. Wever, H. Wriedt, Q. Zheng, *Algorithms for Modern Circuit Simulation*, Arch. Elektron. und Übertragungstechnik, Bd. 46, S. 274–285, 1992
- [Fra08] Fraunhofer Institute for Industrial Mathematics, *Analog Insydes for Mathematica*, 2008
- [FrOl07] M. Freisfeld, M. Olbrich, C. Grimm, E. Barke, *Verwendung von Gebietsarithmetiken zum Entwurf robuster Schaltungen und Systeme*, in *1. GMM/GI/ITG-Fachtagung Zuverlässigkeit und Entwurf*, S. 131–136, 2007
- [GiSa91] G. Gielen, W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Kluwer Academic Publishers, 1991
- [Good94] M. T. Goodrich, *Efficient Piecewise-Linear Function Approximation Using the Uniform Metric*, SCG '94: Proceedings of the Tenth Annual Symposium on Computational Geometry, S. 322–331, 1994
- [GrGr06] D. Grabowski, C. Grimm, E. Barke, *Semi-Symbolic Modeling and Simulation of Circuits and Systems*, in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, S. 986–990, 2006

- [GrHe04] C. Grimm, W. Heupke, K. Waldschmidt, *Refinement of Mixed-Signal Systems with Affine Arithmetic*, in *Design and Test in Europe 2004 (DATE '04)*, S. 372–377, Paris, France, 2004
- [GrOl08] D. Grabowski, M. Olbrich, E. Barke, *Analog Circuit Simulation Using Range Arithmetics*, in *Proceedings of the ASP-DAC*, S. 762–767, 2008
- [GüGo91] C. Güzelis, I. C. Goknar, *A Canonical Representation for Piecewise-Affine Maps and its Applications to Circuit Analysis*, in *IEEE Transactions on Circuits and Systems*, Bd. 38, S. 1342–1354, 1991
- [HaBr71] G. Hachtel, R. Brayton, F. Gustavson, *The Sparse Tableau Approach to Network Analysis and Design*, *IEEE Transactions on Circuit Theory*, Bd. 18, S. 101–113, 1971
- [Hans92] E. R. Hansen, *Bounding the Solution of Interval Linear Equations*, *SIAM J. Numer. Anal.*, Bd. 29, Nr. 5, S. 1493–1503, 1992
- [Hans08] E. R. Hansen, *A Generalized Interval Arithmetic*, In *Interval Mathematics: Lecture Notes in Computer Science*, Bd. 29, S. 7–18, 2008
- [Heal93] R. Henderson, et. al., *A Spreadsheet Interface for Analog Design Knowledge Capture and Reuse*, in *Proceedings of the IEEE Custom Integrated Circuits Conference*, S. 1331–1334, 1993
- [Hedr04] L. Hedrich, *Entwurf Integrierter Analoger Schaltungen mit Hilfe Symbolischer Methoden*, Logos Berlin, 2004
- [HoBr94] B. Hosticka, W. Brockherde, R. Klinke, R. Kokozibski, *Design Methodology for Analog Monolithic Circuits*, *IEEE TCS I: Fundamental, Theory and Applications*, S. 387–394, 1994
- [HuHo98] D. R. Hush, B. Horne, *Efficient Algorithms for Function Approximation with Piecewise Linear Sigmoidal Networks*, *IEEE Transactions on Neural Networks*, Bd. 9, S. 1129–1141, 1998
- [ImIr86] H. Imai, M. Iri, *An Optimal Algorithm for Approximating a Piecewise Linear Function*, *J. Inform. Process.*, Bd. 9, Nr. 3, S. 159–162, 1986
- [KaCh90] C. Kahlert, L. Chua, *A Generalized Canonical Piecewise-linear Representation*, in *IEEE Transactions on Circuits and Systems*, Bd. 37, S. 373–383, 1990
- [KaOh75] K. Kawakita, T. Ohtsuki., *NETCAR 2: A Circuit Analysis Program*, in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, S. 92–95, 1975

- [Katz65] J. Katzenelson, *An Algorithm for Solving Nonlinear Resistor Networks*, in *Bell System Technical Journal*, Bd. 44, S. 1605–1620, 1965
- [KeLe92] T. A. M. Kevenaer, D. M. W. Leenaerts, *A Comparison of Piecewise Linear Model Descriptions*, *IEEE Trans. Circ. Syst. Part I*, Bd. 39, S. 996–1004, 1992
- [Kole93] L. V. Kolev, *Interval Methods for Circuits Analysis*, World Scientific Publishing, 1993
- [Kolm63] A. N. Kolmogorov, *On the Representation of Continuous Functions of many Variables by Superpositions of Continuous Functions of one Variable and Addition*, in *American Mathematical Biophysics*, Bd. 5, S. 55–59, 1963
- [KrLe96] W. Kruiskamp, D. Leenaerts, *Behavioral and Macro Modeling using Piecewise Linear Techniques*, *Int. J. Analog Integr. Circuits Signal Processing*, Bd. 10, S. 67–76, 1996
- [KuKi06] S. V. Kumar, C. H. Kim, S. S. Sapatnekar, *An Analytical Model for Negative Bias Temperature Instability*, in *ICCAD '06: Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, S. 493–496, ACM, 2006
- [LeBo98] D. M. W. Leenaerts, W. M. G. van Bokhoven, *Piecewise Linear Modeling and Analysis*, Kluwer Academic Publisher, 1998
- [Lee04] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, 2004
- [LeHe02] A. Lemke, L. Hedrich, E. Barke, *Analog Circuit Sizing Based on Formal Methods Using Affine Arithmetic*, in *Proceedings of ICCAD 2002*, S. 486–489, 2002
- [Lemk67] C. E. Lemke, *On Complementary Pivot Theory*, in *Mathematics of the Decision Sciences. Part I*, Bd. 11, S. 95–114, 1967
- [Lin91] P.-M. Lin, *Symbolic Network Analysis*, Elsevier, 1991
- [LiNe86] J. C. Lin, J. H. Nevin, *A Modified Time Domain Model for Nonlinear Analysis of an Operational Amplifier*, *IEEE Journal of Solid-State Circuits*, Bd. SC-21, S. 478–483, 1986
- [LiWa03] X. Li, S. Wang, W. Wang, *A Novel Nonlinear Approximation Approach: Enhanced Hinge-Finding Algorithm*, in *Acta Automatica Sinica*, Bd. 29, S. 8–13, 2003

- [MaBo08] A. Magnani, S. Boyd, *Convex Piecewise-Linear Fitting*, Optimization and Engineering, 2008
- [MaGi08] E. Martens, G. Gielen, *High-Level Modeling and Synthesis of Analog Integrated Systems*, Springer Verlag, 2008
- [McGe03] M. S. McCorquodale, F. H. Gebara, K. L. Kraver, E. D. Marsman, R. M. Senger, R. B. Brown, *A Top-Down Microsystems Design Methodology and Associated Challenges*, in *DATE '03: Proceedings of the Conference on Design, Automation and Test in Europe*, S. 292–296, IEEE Computer Society, 2003
- [Mess02] F. Messine, *Extensions of Affine Arithmetic: Application to Unconstrained Global Optimization*, in *Journal of Universal Computer Science*, Vol. 8, Issue 11, S. 992–1015, 2002
- [Moor66] R. E. Moore, *Interval Analysis*, Prentice-Hall, 1966
- [OcRu96] E. S. Ochotta, R. A. Rutenbar, L. R. Carley, *Synthesis of High-Performance Analog Circuits in ASTRX/OBLX*, in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, S. 273–294, 1996
- [OlPo01] M. Olbrich, R. Popp, L. N athke, L. Hedrich, E. Barke, *A Combined Structural and Symbolic Method for Automatic Behavioral Modeling of Nonlinear Analog Circuits*, in *Proceedings of the 15th European Conference on Circuit Theory and Design (ECCTD 01)*, S. 229–232, Helsinki University of Technology, Espoo, Finland, 2001
- [OlRe01] M. Olbrich, A. Rein, E. Barke, *An Improved Hierarchical Classification Algorithm for Structural Analysis of Integrated Circuits*, in *DATE2001: 4th Design Automation and Test in Europe*, S. 807 ff., 2001
- [PiMu00] J. Pittman, C. Murthy, *Fitting Optimal Piecewise Linear Functions Using Genetic Algorithms*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 22, S. 701–718, IEEE Computer Society, 2000
- [PuSj98] P. Pucar, J. Sj oberg, *On the Hinge-Finding Algorithm for Hinge Hyperplanes*, in *IEEE Trans. Inf. Theory*, Bd. 44, S. 1310–1319, 1998
- [QuJr04] M. Queiroz, C. H. Jr., J. Judice, *On Finding Global Optima for the Hinge Fitting Problem*, in *Computer and Operations Research*, Bd. 31, S. 101–122, 2004
- [Rueh86] A. Ruehli (Hrsg.), *Circuit Analysis, Simulation and Design*, Kap. Piecewise Linear Analysis and Simulation, North Holland Publishing, 1986

- [ShHo68] H. Shichman, D. Hodges, *Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits*, in *IEEE J. Solid-State Circuits*, Bd. SC-3, S. 285–289, 1968
- [Spir90] H. Spiro, *Simulation Integrierter Schaltungen*, Oldenbourg Verlag, 1990
- [StEi90] M. T. v. Stiphout, J. T. J. v. Eijndhoven, H. W. Buurman, *PLATO: A New Piecewise Linear Simulation Tool*, Proceedings of the European Design Automation Conference (EDAC), S. 235–239, 1990
- [TiSh00] M. W. Tian, C.-J. R. Shi, *Worst Case Tolerance Analysis of Linear Analog Circuits Using Sensitivity Bands*, *IEEE Transactions on Circuits and Systems*, Bd. 47, Nr. 8, S. 1138–1145, 2000
- [VaGr03] A. Vachoux, C. Grimm, K. Einwich, *Analog and Mixed-Signal Modeling with SystemC-AMS*, Proceedings of the 2003 International Symposium on Circuits and Systems, Bd. 3, S. 914–917, 2003
- [vB81] W. van Bokhoven, *Piecewise-Linear Modeling and Analysis*, Kluwer, 1981
- [vdP74] C. v. d. Panne, *A Complementary Variant of Lemke's Method for the Linear Complementary Problem*, *Mathematical Programming*, Bd. 7, Nr. 1, S. 283–310, 1974
- [WaGi98] P. Wambacq, G. Gielen, W. Sansen, *Symbolic Network Analysis Methods for Practical Analog Integrated Circuits: A Survey*, *IEEE Transactions on Circuits and Systems II*, Bd. 45, S. 1331–1341, 1998
- [Yama96] K. Yamamura, *An Algorithm for Representing Nonseparable Functions by Separable Functions*, in *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Bd. E79-A, 1996
- [ZhOl06] M. Zhang, M. Olbrich, H. Kinzelbach, D. Seider, E. Barke, *A Fast and Accurate Monte Carlo Method for Interconnect Variation*, in *ICICDT 2006 Proceedings*, S. 207–210, 2006

Lebenslauf

Persönliche Angaben

Marc Freisfeld
Dipl.-Ing. der Elektrotechnik
Geburtsdatum 19.06.1975 in Haselünne
Staatsangehörigkeit deutsch

Wissenschaftlicher Werdegang

- 10/96 - 05/03 Studium der Elektrotechnik Fachrichtung Technische Informatik, Leibniz Universität Hannover, Abschluss Diplom-Ingenieur
- 04/99 - 03/01 Wissenschaftliche Hilfskraft
- 07/03 - 12/08 Wissenschaftlicher Mitarbeiter am Institut für Mikroelektronische Systeme (www.ims.uni-hannover.de), Leibniz Universität Hannover