

Erweiterte gebietsarithmetische Verfahren zur Simulation analoger Schaltungen

Von der Fakultät für Elektrotechnik und Informatik der
Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von Dipl.-Ing. Oliver Scharf
geboren am 17.10.1984 in Hannover

2018

1. Referent	Prof. Dr.-Ing. Erich Barke
2. Referent	Prof. Dr.-Ing. Lars Hedrich
Tag der Promotion	04. Mai 2018

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für mikroelektronische Systeme. Sie stellt das Ergebnis meiner Forschungsarbeiten im Bereich der Entwurfsautomatisierung analoger Schaltungen dar.

Ich möchte mich an dieser Stelle bei meinem Doktorvater Herrn Professor Erich Barke für die Betreuung der Arbeit, die wichtigen Impulse und die Unterstützung bei der Durchführung bedanken. Herrn Professor Lars Hedrich danke ich für die Übernahme des Koreferats der Arbeit.

Weiterhin möchte ich mich bei allen Mitarbeiterinnen und Mitarbeitern des Instituts für den Austausch und die Anmerkungen bedanken. Insbesondere danke ich Herrn Dr. Markus Olbrich für seine fachliche Unterstützung durch zahlreiche Diskussionen.

Auch meinen Eltern und Schwiegereltern gilt mein Dank für die Unterstützung. Insbesondere danke ich jedoch meiner Frau Anna und meinem Sohn Jan dafür, dass sie mich in schwierigen Zeiten unterstützt haben. Dank ihrer Geduld und ihres Verständnis konnte ich dieses Ziel erreichen.

Celle, im Juni 2018

Oliver Scharf

Kurzfassung

Zur Simulation von mit Parameterunsicherheiten behafteten Analogschaltungen können verschiedene Verfahren eingesetzt werden. Die Parameterunsicherheiten können verschiedene Ursachen wie Fertigungstoleranzen, Alterung oder wechselnde Umgebungstemperaturen haben. Herkömmliche Verfahren zur Simulation von Parameterunsicherheiten wie das Corner-Case- oder das Monte-Carlo-Verfahren besitzen den Nachteil, dass der Einschluss der Schaltungseigenschaften, die durch die verschiedenen Kombinationen der Parameter möglich sind, nicht garantiert ist. Als Alternative können gebietsarithmetische Verfahren eingesetzt werden. Sie garantieren den Einschluss aller Eigenschaften. Im Allgemeinen sind die dabei zu lösenden Gleichungssysteme implizit und nichtlinear. Zur Lösung dieser Gleichungssysteme mit Hilfe von affiner Arithmetik wurde in vorherigen Arbeiten das EPD-Verfahren vorgestellt. Die mit diesem Verfahren simulierbare Schaltungsgröße und die maximale Größe der Parameterunsicherheiten ist begrenzt. Dies liegt zum einen an der Überschätzung, die durch die affine Arithmetik verursacht wird, und zum anderen an den Eigenschaften des EPD-Verfahrens selbst.

Ein Ansatz zur Vergrößerung des Konvergenzgebiets ist der Einsatz von Gebietsaufteilungen und -zusammenfassungen. Dazu wird der Parameterraum aufgeteilt, mit den geteilten Parametern getrennt berechnet und schließlich wieder zusammengefügt. Um die Simulation durch diesen Ansatz nicht unnötig zu verlangsamen, ist es sinnvoll, die Aufteilung nur an Punkten durchzuführen, die ohne Aufteilungen nicht lösbar sind. Dazu werden im Rahmen dieser Arbeit zwei Kriterien vorgestellt anhand derer solche automatischen Aufteilungen durchgeführt werden können. Weiterhin werden verschiedene Aufteilungs- und Zusammenfassungsstrategien vorgestellt und miteinander verglichen. Verglichen werden hierbei die benötigte Laufzeit, die entstehende Überschätzung und die benötigte Anzahl der Aufteilungen.

Die Ergebnisse zeigen, dass auf diese Weise die simulierbare Schaltungsgröße und die maximale Größe der Parameterunsicherheiten vergrößert werden kann. Durch die Wahl der Strategien kann Laufzeit gegen Überschätzung eingetauscht werden. Die optimale Strategie und die mögliche Vergrößerung des Konvergenzgebiets sind dabei von der zu simulierenden Schaltung abhängig.

Schlüsselwörter: Affine Arithmetik, Schaltungssimulation, Konvergenzgebiet

Abstract

Several different methods can be used for the simulation of analog circuit with uncertain parameters. The uncertainties can have different root causes, for example manufacturing tolerances, aging or ambient temperature. Common simulation methods like the Corner Case or Monte Carlo method have the disadvantage that they do not include all characteristics caused by all the possible parameter combinations. Methods based on range arithmetics can be used as an alternative. They guarantee the inclusion of all characteristics. The equation systems that need to be solved are in general implicit and nonlinear. In previous works an algorithm called EPD method was introduced. However, the size of the circuits and the size of parameter uncertainties which can be solved by this method is limited. This is due to the overapproximation caused by the affine arithmetic and the properties of the EPD method itself.

One approach to increase the convergence range is the utilization of splitting and merging. The parameter space is divided, calculations are done with divided parameters and the results are merged afterwards. To not slow down the simulation by this approach it is reasonable to perform the split only if a solution is not possible otherwise. Two criterions are presented to perform such an automatic split decision. Furthermore, different splitting and merging strategies are compared in terms of runtime, overapproximation and required number of splits.

The results show that the circuit size and the maximum size of parameter uncertainties can be increased by this method. By choosing the right strategies runtime can be traded for overapproximation. The optimal strategies and the gain in convergence range depend on the circuit which is simulated.

Key words: Affine arithmetics, circuit simulation, convergence range

Inhaltsverzeichnis

Symbolverzeichnis	xi
Abkürzungsverzeichnis	xiii
Abbildungsverzeichnis	xv
Tabellenverzeichnis	xvii
1 Einleitung	1
1.1 Ziele der Arbeit	4
2 Simulation von Parameterunsicherheiten	5
2.1 Grundlagen von Gebietsarithmetiken	8
2.1.1 Überschätzung	9
2.1.2 Intervallarithmetik	10
2.1.3 Affine Arithmetik	13
2.1.4 Weitere Gebietsarithmetiken	17
2.1.5 Bewertung der verschiedenen Gebietsarithmetiken	17
2.2 Gebietsarithmetische Lösungsverfahren	19
2.2.1 Lineare Gleichungssysteme	19
2.2.2 Nichtlineare Gleichungssysteme	20
2.2.3 Anwendungen	22
2.2.4 Eignung zur Schaltungssimulation	23
2.3 Gleichungslösung als Optimierungsproblem	24
2.3.1 Lösung des Optimierungsproblems mit einem Brute-Force-Ansatz	25
2.4 Das EPD-Verfahren	27
2.4.1 Bedeutung des Skalierungsfaktors	30
2.4.2 Beispiel: Lösen einer Gleichung mit dem EPD-Verfahren	31
2.4.3 Transiente Simulationen	33
2.4.4 Komplexität des EPD-Verfahrens	34
3 Erweitertes EPD-Verfahren	39
3.1 Grenzen des Verfahrens	39
3.2 Maß für die Überschätzung	43
3.3 Ausnutzung von Monotonieeigenschaften	44
3.4 Gebietsaufteilungen und Gebietszusammenfassungen	47
3.4.1 Automatische Gebietsaufteilungen	51

3.5	Transiente Simulationen mit Gebietsaufteilungen	51
3.5.1	Zusammenfassungsstrategien	53
3.6	Aufteilungsstrategien	55
3.7	Extraktion von affinwertigen Eigenschaftsgrößen	56
4	Ergebnisse	59
4.1	Beispielschaltungen	60
4.2	Grenzen des EPD-Verfahrens ohne Aufteilungen	65
4.3	Laufzeitverhalten ohne Gebietsaufteilungen	65
4.4	Grenzen des EPD-Verfahrens mit Gebietsaufteilungen	70
4.5	Vergleich der Strategien zur automatischen Gebietsaufteilung	71
4.6	Vergleich der Aufteilungsstrategien	74
4.7	Ergebnisse für die verschiedenen Zusammenfassungsstrategien	79
4.7.1	Bestimmung von n für die Zusammenfassungsstrategie <i>Merge</i> n^{th} <i>Step</i>	79
4.7.2	Vergleich der Zusammenfassungsstrategien	81
4.8	Kombination der Aufteilungs- und Zusammenfassungsstrategien	83
4.9	Laufzeitverhalten mit Gebietsaufteilungen	90
4.10	Vergleich der unoptimierten mit den optimierten Einstellungen	95
5	Zusammenfassung	97
	Literaturverzeichnis	99
A	Anhang	109

Symbolverzeichnis

A	Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$
\mathbb{D}	Definitionsbereich einer Funktion
$\frac{df}{dx}$	Ableitung der Funktion f nach der Variablen x
\mathbb{E}	Globale Menge aller Abweichungssymbole $\varepsilon_1.. \varepsilon_n$
$\mathbb{E}_{\hat{x}}$	Menge der Abweichungssymbole $\varepsilon_1.. \varepsilon_n$, die Teil der affinen Form \hat{x} sind
$f^{-1}(x)$	Umkehrfunktion der Funktion $f(x)$
$f_{\text{lin}}(x)$	lineare Funktion
$f_{\text{mon}}(x)$	monotone Funktion
$f_{\text{NA}}(x)$	nichtaffine Funktion, erzeugt bei ihrer Berechnung ein NLPD-Symbol
$f_{\text{NL}}(x)$	nichtlineare Funktion
\mathbf{I}_n	Einheitsmatrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$, z.B. $\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
\mathbb{I}	Globale Menge der Indizes aller Abweichungssymbole $1..n$
$\mathbb{I}_{\hat{x}}$	Menge der Indizes der Abweichungssymbole $1..n$, die Teil der affinen Form \hat{x} sind
J	Jacobi-Matrix, Matrix mit allen Ableitungen $\frac{df}{dx}$
$\text{mid}([x])$	Mittelpunkt des Intervalls $[x]$ bzw. des Vektors oder der Matrix der Zentralwerte aus $[\vec{x}]$ bzw. $[\mathbf{x}]$
m_{OA}	mittlere relative Überschätzung, Berechnung nach Gleichung (3.1) bzw. Gleichung (3.2)
n_{iter}	Anzahl der durchgeführten Iterationen
$n_{\text{iter,max}}$	maximale Anzahl der Iterationen, nach Erreichen dieser Anzahl wird Nichtkonvergenz angenommen
$n_{\text{iter,split}}$	maximale Anzahl der Iterationen, nach Erreichen dieser Anzahl wird eine Aufteilung durchgeführt (Kriterium 1 aus Abschnitt 3.4.1)
n_e	Anzahl der Bauelemente in einer Schaltung ohne Spannungsquellen
$n_{\text{ss,max}}$	maximale Anzahl von Aufteilungen, die mit der ausgewählten Aufteilungsstrategie durchgeführt werden, siehe Algorithmus 10
$n_{\text{ts,max}}$	maximale Anzahl von Aufteilungen nach der die Lösung abgebrochen wird, siehe Algorithmus 10
n_p	Anzahl der Parameter \vec{p}
n_{splits}	aktuelle Anzahl durchgeführter Aufteilungen

Symbolverzeichnis

n_x	Anzahl der Schaltungsvariablen \vec{x}
\mathcal{P}	Parameterraum
p	Schaltungsparameter
\mathbb{R}	Menge der reellen Zahlen
r	Residuum $r = f(x)$
$\mathbb{R}^{>0}$	Menge der positiven reellen Zahlen ohne Null
$\mathbb{R}^{\geq 0}$	Menge der positiven reellen Zahlen einschließlich Null
$\mathbb{R}^{\neq 0}$	Menge der reellen Zahlen ohne Null
r_{aff}	Anzahl der affinen Simulationen, die zur Bestimmung des Minimums der Laufzeit benutzt werden
REAL_MAX	Größte darstellbare reelle Zahl
r_{MC}	Anzahl der Monte-Carlo-Simulationen, die zur Berechnung der Überschätzung m_{OA} benutzt werden
$\check{\mathbb{R}}$	Menge der Gebiete aus den reellen Zahlen
$\sigma(x)$	Sprungfunktion $\sigma(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$
V_c	Volumen der konvexen Hülle des Gebiets im Parameterraum \mathcal{P} , in dem ein Algorithmus konvergiert
\mathcal{X}	Zustandsraum
x	Skalar $x \in \mathbb{R}$
\vec{x}	Vektor $\vec{x} = (x_1, x_2, \dots, x_n)^T$
$\vec{x}_{(n)}$	n -tes Element des reellwertigen Vektors \vec{x} , bei $\vec{x} = (x_1, x_2, \dots, x_n)^T$ also x_n
\hat{x}	Affine Form
$\vec{\hat{x}}$	Affinwertiger Vektor $\vec{\hat{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)^T$
$\vec{\hat{x}}_{(n)}$	n -tes Element des affinwertiger Vektors $\vec{\hat{x}}$, bei $\vec{\hat{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)^T$ also \hat{x}_n
\check{x}	Beliebige gebietsarithmetische Darstellung von x aus $\check{\mathbb{R}}$, kann Intervall $[x]$ oder affine Form \hat{x} sein
$[x]$	Intervall $[x] = [\underline{x}, \bar{x}]$
\mathcal{Y}	Eigenschaftsraum

Abkürzungsverzeichnis

<i>AGIAS</i>	Simulator, der das erweiterte EPD-Verfahren aus Kapitel 3 implementiert
<i>AMV</i>	System zur symbolischen Behandlung von Schaltungen, implementiert in Maple
<i>BJT_INV</i>	Beispielschaltung Bipolarinverter nach Abbildung 4.2
<i>BP</i>	Beispielschaltung Bandpass nach Abbildung 4.4
<i>CC</i>	Das Corner-Case-Verfahren (CC) ist eine Methode zur Simulation von Parameterschwankungen. Die Parameter werden dabei nacheinander zu den Ecken des Parameterraumes ausgelenkt und mit diesen Werten wird eine Nominalsimulation durchgeführt.
<i>CSE</i>	Elimination von gemeinsamen Subausdrücken, englisch Common Sub-expression Elimination (CSE)
<i>DAE</i>	Algebro-Differentialgleichung bzw. -ssystem (DAE)
<i>DGL</i>	Differentialgleichung bzw. Differentialgleichungssystem (DGL)
<i>DIFF_AMP</i>	Beispielschaltung Differenzverstärker nach Abbildung 4.5
<i>EPD</i>	Enhanced Partial Deviations (EPDs) sind Abweichungssymbole ε_i , die durch den Lösungsalgorithmus zur Sicherstellung des Einschlusses des Ergebnisses eingefügt werden
<i>IFT</i>	Satz von der impliziten Funktion, englisch Implicit Function Theorem (IFT)
<i>MC</i>	Das Monte-Carlo-Verfahren (MC) ist eine Methode zur Simulation von Parameterschwankungen. Die Parameter werden dabei zufällig ausgelenkt und mit diesen Werten wird eine Nominalsimulation durchgeführt.
<i>NLPD</i>	Nonlinear Partial Deviations (NLPDs) sind Abweichungssymbole ε_i , die bei der Auswertung von nichtlinearen Funktionen entstehen
<i>NLTL</i>	Nichtlineare Übertragungsstrecke, englisch Nonlinear Transmission Line (NLTL)
<i>NLTL₃</i>	Beispielschaltung bestehend aus drei NLTL-Gliedern nach Abbildung 4.9
<i>PPD</i>	Parameter Partial Deviations (PPDs) sind Abweichungssymbole ε_i , die mit den gegebenen Parameterunsicherheiten korrespondieren
<i>SVF</i>	Beispielschaltung State Variable Filter nach Abbildung 4.6

Abbildungsverzeichnis

2.1	Simulation: Zuordnung einer Menge von Parametern zu einer Menge von Zuständen	5
2.2	Gegenüberstellung der Abbildung vom Parameterraum in den Zustandsraum für exakte und mit Unsicherheiten behaftete Parameter	6
2.3	Gegenüberstellung der durch das Monte-Carlo- und Corner-Case-Verfahren ausgewählten Punkte für die nichtmonotone Funktion $x = p^2$	7
2.4	Sensitivitätsanalyse der Funktion $x = p^2$	8
2.5	Gegenüberstellung von punkt- und gebietsarithmetischer Abbildung vom Parameterraum in den Zustandsraum	8
2.6	Überschätztes Gebiet \vec{y} im Vergleich mit dem minimalen Gebiet \vec{y}_{minimal} .	9
2.7	Gegenüberstellung des überschätzten mit dem minimalen Gebiet im Vergleich zum Definitionsbereich	10
2.8	Darstellung des Wrapping-Effekts	12
2.9	Approximation am Beispiel der Exponentialfunktion	16
2.10	Ablaufdiagramm des EPD-Verfahrens	22
2.11	Darstellung von Stichproben, die Bedingung (2.72) erfüllen	26
2.12	Grafische Darstellung der Auswertung einer affinen Funktion	30
3.1	Verschiedene Anzahlen von Nullstellen	40
3.2	Exploration eines Konvergenzgebiets in Abhängigkeit von zwei Parametern p_1 und p_2 nach Algorithmus 5	42
3.3	Exploration eines Konvergenzgebiet in Abhängigkeit von zwei Parametern p_1 und p_2 nach Algorithmus 6	42
3.4	Darstellung des dreidimensionalen Konvergenzgebiets von Algorithmus 5 für eine Beispielschaltung	43
3.5	Intervallbreiten d_A und d_{MC} zur Bestimmung der mittleren relativen Überschätzung m_{OA}	44
3.6	Parameter- und Lösungsbaum für zwei Aufteilungen	48
3.7	Überschätzung in Abhängigkeit von der Anzahl der erzwungenen Gebietsaufteilungen	49
3.8	Vergleich der Zusammenfassungsoperationen	50
3.9	Verlauf des Skalierungsfaktors s_1 über der Anzahl der Iterationen für die Gleichung $e^{\hat{x}} - \hat{p}$	51
3.10	Parameterbaum für zwei Zeitschritte t_{n-1} und t_n	52
3.11	Lösungsbäume von zwei Zeitschritten einer transienten Simulation für zwei Verfahren zur Berechnung der numerischen Integration	52

3.12	Vergleich der Zusammenfassungsstrategien	54
3.13	Zeitpunkt t_1 und t_{end} zur Extraktion von affinwertigen Eigenschaftsgrößen eines Tiefpasses	57
3.14	Tiefpassschaltung zur Extraktion von affinwertigen Eigenschaftsgrößen	58
4.1	Darstellung der Simulatorarchitektur	59
4.2	Inverter mit Bipolartransistor	61
4.3	Nonlinear Transmission Line (NLTL) nach [108, Figure 1]	62
4.4	Aktiver Bandpass nach [103, Abbildung 13.33]	62
4.5	Differenzverstärker	63
4.6	State Variable Filter nach [38, Abbildung 5.18]	64
4.7	Kettenschaltung aus n Elementen	64
4.8	Inverterkette aus n Invertern	64
4.9	Kettenschaltung aus n NLTLs	65
4.10	Laufzeit für verschiedene Werte von n_p	67
4.11	Überschätzung für verschiedene Werte von n_p	68
4.12	Laufzeit für verschiedene Werte von s_d	69
4.13	Laufzeit für verschiedene Werte von n in der Kettenschaltung NLTL $_n$	69
4.14	Simulationslaufzeit t und Anzahl der Aufteilungen für automatische Auf- teilungen nach Kriterium 1 und verschiedenen Werte von $n_{\text{iter,split}}$	72
4.15	Simulationslaufzeit t und Anzahl der Aufteilungen für automatische Auf- teilungen nach Kriterium 2 und verschiedenen Werte von Δs_{max}	73
4.16	Vergleich der Laufzeit für verschiedene Aufteilungsstrategien in den Vari- anten a, b und c	76
4.17	Vergleich der Anzahl der Aufteilungen n_{splits} für verschiedene Aufteilungs- strategien in den Varianten a, b und c	77
4.18	Vergleich der Überschätzung für verschiedene Aufteilungsstrategien in den Varianten a, b und c	78
4.19	Exploration von verschiedenen Werten für n für die Zusammenfassungs- strategie <i>Merge n^{th} Step</i>	80
4.20	Vergleich der Zusammenfassungsstrategien	82
4.21	Laufzeit und Anzahl der Aufteilungen für verschiedene Werte von s_d mit Gebietsaufteilungen	91
4.22	Laufzeit für verschiedene Werte von n in der Kettenschaltung BJT_INV $_n$	92
4.23	Laufzeit für verschiedene Werte von n in der Kettenschaltung NLTL $_n$	92
4.24	Laufzeit und Anzahl der Aufteilungen für verschiedene Werte von n_p	93
4.25	Überschätzung und Anzahl der Aufteilungen für verschiedene Werte von n_p	94
4.26	Vergleich der unoptimierten mit den optimierten Einstellungen	96

Tabellenverzeichnis

2.1	Überschätzung für Multiplikationen in Gebietsarithmetiken AA, EAA, RAA, GIA (aus [100])	18
2.2	Überschätzung und Laufzeitkomplexität der Multiplikation in den Gebietsarithmetiken AATRE, AACHA, AASEE	18
2.3	Ergebnisse und Skalierungsfaktoren der verschiedene Iterationsschritte für das Lösen von Gleichung (2.82)	33
2.4	Anzahl der Fließkommaoperationen unter Vernachlässigung der Operationen für die Spaltenpivotstrategie	37
3.1	Strategien zur Auswahl des aufzuteilenden Parameters	56
4.1	Übersicht über die verwendeten Beispielschaltungen	61
4.2	Konvergenzgebiet für das EPD-Verfahren ohne Gebietsaufteilungen	65
4.3	Konvergenzgebiet für das EPD-Verfahren mit Gebietsaufteilungen	70
4.4	Vergleich der Konvergenzgebiete für das EPD-Verfahren mit und ohne Gebietsaufteilungen	71
4.5	Vergleich der automatischen Aufteilung nach Kriterium 1 und 2	74
4.6	Vergleich der Laufzeit für die verschiedenen Kombinationen aus Aufteilungs- und Zusammenfassungsstrategien	85
4.7	Vergleich der Anzahl der Aufteilungen für die verschiedenen Kombinationen aus Aufteilungs- und Zusammenfassungsstrategien	86
4.8	Vergleich der Überschätzung für die verschiedenen Kombinationen aus Aufteilungs- und Zusammenfassungsstrategien	88

1 Einleitung

Der Mensch versucht die Realität durch Modelle begreifbar zu machen. Ein Modell ist eine vereinfachte, abstrahierende Darstellung der Realität. Die zugrunde liegende Fragestellung bestimmt den Abstraktions- bzw. Detaillierungsgrad des Modells. Dieser allgemeine Modellbegriff wird in zwei Kategorien unterteilt. Materielle Modelle sind eine verkleinerte oder vereinfachte Realisierung des realen Systems, zum Beispiel ein Architekturmodell. Ideelle Modelle dagegen sind gedankliche Konstrukte, die das reale System beschreiben (vgl. [4]). Modelle können verschiedene Beschreibungsformen besitzen. Um quantitative Aussagen und Vorhersagen anhand von Modellen treffen zu können, wird in der Regel eine formalisierte, mathematische Beschreibung gewählt. Häufig werden Systeme durch Gleichungssysteme modelliert, aber auch andere mathematische Beschreibungsformen wie Graphen sind möglich. Ideelle Modelle werden in vielen Disziplinen, wie zum Beispiel der Physik sowie in den Ingenieur- und Gesellschaftswissenschaften, eingesetzt, um Simulationen als virtuelle Experimente durchzuführen und daraus Erkenntnisse zu gewinnen. Überwiegend werden diese Simulationen mit Hilfe von Computern durchgeführt.

Reale Experimente sind in ihrer Durchführbarkeit durch verschiedene Aspekte begrenzt. Das können zum Beispiel die Kosten, die mit dem Experiment verbundenen Gefahren oder der benötigte Beobachtungszeitraum sein [8]. Diese Nachteile können virtuelle Experimente aufheben oder zumindest abmildern. Damit mit ihnen allerdings Aussagen über das reale Verhalten gemacht werden können, ist eine sorgfältige Modellbildung nötig. Zur Bildung von Modellen lassen sich drei verschiedene Verfahren unterscheiden. Beim White-Box-Ansatz ist der Aufbau des Systems vollständig bekannt und kann zur Modellierung herangezogen werden. Dagegen ist beim Black-Box-Ansatz der Aufbau des Systems unbekannt. Lediglich durch Interaktion mit dem System und dem Beobachten der Systemausgänge können Informationen über das Systemverhalten gewonnen werden. Die entstehenden Modelle werden daher Verhaltensmodelle genannt. Eine Mischform aus beiden Ansätzen ist der Grey-Box-Ansatz, bei dem einzelne innere Systemkomponenten bekannt, andere jedoch unbekannt sind (vgl. [4]). Dieser Ansatz wird häufig verwendet, um ein ausreichend genaues Modell mit begrenztem Aufwand erstellen zu können. Das Verhalten des so erhaltenen Modells sollte – so weit möglich – gegen das reale Systemverhalten validiert werden. Ein Modell ist jedoch stets eine Abstrahierung der Realität, da nur ein Teil des Verhaltens des tatsächlichen Systems durch das Modell wiedergegeben wird. Diese Komplexitätsreduktion ist beabsichtigt, um das Modell handhabbar zu machen. Zu komplexe Modelle sind in der zur Verfügung stehenden Zeit nicht berechenbar. Zusammenhänge, die bei der Bildung des Modells nicht bekannt sind oder als vernachlässigbar betrachtet werden, sind im Modell nicht enthalten und können von ihm daher auch nicht abgebildet werden. In diesen Fällen ist das Übertragen von Erkenntnissen vom Modell auf das reale System unzulässig. Die Erfassung von allen

1 Einleitung

für die Fragestellung relevanten Abhängigkeiten und Gewährleistung der numerischen Stabilität ist komplex. Daher wird die Modellbildung in der Regel iterativ durchgeführt, um ein zufriedenstellendes Ergebnis zu erreichen.

Dieser Prozess der Modellbildung wird auch auf technische Systeme angewendet. Technische Systeme realisieren mit technischen Mitteln ein bestimmtes, beabsichtigtes Verhalten. Dieses Verhalten mit seinen Eigenschaften und Grenzen wird in einer Spezifikation beschrieben. Modelle zur Beschreibung dieser Systeme können verschiedene Aufgaben besitzen (vgl. [4]). Sie können zur Analyse und Vorhersage des Systemverhaltens dienen. So lassen sich zum Beispiel Aussagen zur Stabilität oder Schwingfähigkeit des Systems treffen. Vor der Realisierung kann verifiziert werden, ob das System die spezifizierten Anforderungen erfüllt. Eine Simulation kann im Gegensatz zum Experiment bereits in der Entwurfsphase durchgeführt werden. So lassen sich Optimierungen durchführen, die später nach der Realisierung des Systems nicht oder nur sehr aufwändig möglich wären.

Auch für den Entwurf von Reglern werden Modelle benötigt. Die in der Regelungstechnik verwendeten Entwurfsmethoden setzen ein Modell des zu regelnden Systems voraus. Betreibt man ein reales System und das dazugehörige Modell parallel, kann ein Monitoring bzw. eine Diagnose des Systems durchgeführt werden. Durch Vergleich des Verhaltens von Modell und realem System können Fehler während des Betriebs erkannt werden. Diese können entsprechend gemeldet oder behandelt werden.

Das Verhalten technischer Systeme wird durch Designparameter bestimmt. Diese Parameter werden während des Entwurfs festgelegt, um das gewünschte, spezifizierte Verhalten zu realisieren. Die Parameter eines Systems sind nicht beliebig genau bekannt. Diese Unsicherheiten beeinflussen das Systemverhalten. Um diesen Einfluss quantifizieren zu können, wird ein Modell der Parameterungenauigkeiten benötigt. Darin können zum Beispiel statistische Eigenschaften wie die Fehlerwahrscheinlichkeit oder -verteilung enthalten sein.

Im Rahmen dieser Arbeit wird davon ausgegangen, dass die Parameterunsicherheiten mit ihren Minimal- und Maximalwerten bekannt sind. Über die statistische Verteilung der Unsicherheiten wird keine Aussage getroffen. Die Ursachen für Parameterunsicherheiten können vielfältig sein. Zum einen sind die Parameter stets mit unvermeidbaren Fehlern behaftet. Darunter fallen die endliche Genauigkeit von Fließkommazahlen und Fehler durch die verwendeten numerischen Simulationsalgorithmen. Andere unvermeidbare Fehler sind zum Beispiel Messfehler oder Fertigungstoleranzen. Zum anderen können sich Parameter während des Betrieb des Systems innerhalb spezifizierter Grenzen ändern. Zum Beispiel kann die Betriebstemperatur schwanken oder sich das Elastizitätsmodul von Materialien durch Alterung oder Umgebungseinflüsse ändern. Trotz dieser Parameterunsicherheiten soll sichergestellt werden, dass das System seine Aufgabe spezifikationsgemäß erfüllt. Daher müssen die dazu verwendeten Simulationsverfahren diese Unsicherheiten berücksichtigen.

Üblicherweise werden zur Simulation von mit Unsicherheiten behafteten Systemen Methoden wie das Corner-Case- oder Monte-Carlo-Verfahren verwendet. Dabei werden die Parameter von ihren Nominalwerten ausgelenkt und mit dem so gewonnenen Parametersatz wird eine Nominalsimulation durchgeführt, das heißt es wird angenommen, dass der jeweilige Parametersatz exakt bekannt sei. Die Auslenkungen der Para-

meter geschehen beim Monte-Carlo-Verfahren zufällig, während beim (vollständigen) Corner-Case-Verfahren alle Kombinationen der Parameterextremwerte gebildet werden. Außerdem existieren weitere Verfahren zur Auswahl von Stichproben wie das Latin-Hypercube-Sampling [58], diese sind in der Schaltungssimulation nicht weit verbreitet und werden daher nicht näher betrachtet. Beim Corner-Case-Verfahren ergeben sich bei n mit Unsicherheiten behafteten Parametern 2^n Simulationsdurchläufe. Trotzdem garantieren beide Verfahren nicht den Einschluss der Extremwerte des Systemverhaltens. Das Monto-Carlo-Verfahren kann dies aufgrund seiner statistischen Eigenschaften nicht leisten, das Corner-Case-Verfahren scheitert in den Fällen, in denen die Extremwerte des Schaltungsverhaltens nicht für die Extremwerte der Parameter auftreten. Als Alternative zu diesen sogenannten punktarithmetischen Verfahren existieren gebietsarithmetische Verfahren. Diese arbeiten nicht mit einzelnen Punkten im Parameterraum, sondern beschreiben Parameter als Gebiete. Dementsprechend ist das Simulationsergebnis nicht punktförmig, sondern ebenfalls ein Gebiet. Die bekannteste Gebietsarithmetik ist die Intervallarithmetik. Hierbei wird der unsichere Parameter x durch das Intervall $[\underline{x}..\bar{x}]$ dargestellt. Da die Intervalle keine Korrelationsinformationen enthalten, kommt es bereits nach wenigen Rechenschritten zu einer starken Aufweitung des Ergebnisintervalls. Als Weiterentwicklung wurde die affine Arithmetik vorgestellt [21]. Sie erhält lineare Korrelationsinformationen. Diese Informationen können sich während der Berechnung gegenseitig auslöschen und ermöglichen so einen engeren Einschluss des Lösungsgebiets.

Im Bereich des Entwurfes von analogen Schaltungen werden Simulationen hauptsächlich eingesetzt, um Vorhersagen über das Systemverhalten zu treffen und Optimierungen daran vorzunehmen. Insbesondere bei integrierten Schaltungen ist ein Fertigungslauf sehr teuer und das innere Verhalten ist nicht beobachtbar, deshalb wird „first time right silicon“ angestrebt. Dies ist nur durch den umfangreichen Einsatz von Simulationen zu erreichen. Systemparameter sind in der Schaltungssimulation die vom Designer vorgebenden Bauteilparameter, wie Widerstandswerte und Kanallängen. Mit weiter sinkenden Strukturgrößen wird der Einfluss von Fertigungstoleranzen auf diese vorgegebenen Nominalwerte immer größer. Außerdem können sich die Parameter durch verschiedene Alterungseffekte wie NBTI (negative bias temperature instability), HCI (hot carrier injection) oder Elektromigration über die Lebensdauer der Schaltung verändern. Weiterhin haben auch Umgebungsparameter wie die Temperatur einen Einfluss auf das Schaltungsverhalten. In [27] wurde ein Schaltungssimulator vorgestellt, der Schaltungen mit unsicheren Parametern mit Hilfe von affiner Arithmetik berechnen kann. Parameterunsicherheiten werden mit Hilfe von affinen Formen modelliert und ihre Auswirkungen auf das Schaltungsverhalten analysiert. Der dabei eingesetzte Lösungsalgorithmus fügt zusätzliche Abweichungssymbole, die sogenannten Enhanced Partial Deviations (EPDs) ein und skaliert diese anschließend. Er wird daher EPD-Verfahren genannt. Das so gewonnene Ergebnis garantiert den Einschluss der Lösung unter allen Parameterkombinationen, die im betrachteten Parameterraum möglich sind. Da lineare Korrelationen durch die affine Arithmetik erhalten bleiben, lassen sich Parameter, die einen besonders großen Einfluss auf das Schaltungsverhalten haben, identifizieren und gegebenenfalls manuell optimieren.

1.1 Ziele der Arbeit

Im Fokus dieser Arbeit liegt die gebietsarithmetische Simulation von Analogschaltungen unter Berücksichtigung von Unsicherheiten. Das in [27] vorgestellte EPD-Verfahren wird in dieser Arbeit analysiert und erweitert. Wie dort werden die Ungenauigkeiten durch Fließkommazahlen und numerische Algorithmen vernachlässigt und nur die Parametersicherheiten berücksichtigt. Im Rahmen dieser Arbeit werden die Simulationsarten Gleichstrom- und Transientenanalyse untersucht, weitergehende gebietsarithmetische Simulationsarten wie Wechselstrom- oder Oszillationsanalyse, wie in [27] vorgestellt, werden nicht betrachtet. Das EPD-Verfahren wird im Rahmen dieser Arbeit detaillierter dargestellt und einer genaueren Analyse unterzogen. Das Laufzeitverhalten des EPD-Verfahrens wird untersucht und mit dem der herkömmlichen punktarithmetischen Monte-Carlo- und Corner-Case-Verfahren verglichen.

Das bisherige Verfahren besitzt Limitationen hinsichtlich der maximalen Schaltungsgröße, der maximalen Größe der Parameterunsicherheiten und dem maximalen Grad der Nichtlinearität der Schaltungsgleichungen, sodass nur eine begrenzte Anzahl von Schaltungen simulierbar ist. Werden gewisse Randbedingungen verletzt, konvergiert das EPD-Verfahren nicht mehr, und es ist nicht mehr möglich, eine Aussage über das Schaltungsverhalten zu treffen. Diese Bedingungen werden untersucht und quantifiziert. Um die Konvergenz des EPD-Verfahrens auch unter diesen Bedingungen zu ermöglichen, wird eine Erweiterung des EPD-Verfahrens, basierend auf dem Aufteilen von Parametergebieten, vorgestellt. Mit diesen geteilten Parametergebieten werden die Schaltungsgleichungen wie bisher gelöst und die Ergebnisse wieder zusammengefügt, sodass sie das Ursprungsproblem lösen, das heißt, den Einschluss der Lösung sicherstellen. Durch diese Aufteilung konvergiert das EPD-Verfahren in einem größeren Bereich. Es werden Strategien vorgestellt, die eine möglichst effiziente Aufteilung ermöglichen. Die Effizienz dieser Verfahren wird wiederum bewertet und verglichen. Durch die Erweiterung sollen die Schaltungsgröße und die Größe der Parameterunsicherheiten erhöht werden, die gebietsarithmetisch untersucht werden können. Ziel ist es, das EPD-Verfahren für eine größere Anzahl von Schaltungen verwendbar zu machen und so die Einsetzbarkeit des affinen Schaltungssimulators für praxisnahe Schaltungen zu ermöglichen.

2 Simulation von Parameterunsicherheiten

Zur Bestimmung von Eigenschaften eines Systems werden häufig Simulationen eingesetzt. Eine Simulation ordnet einer Menge von Parametern eine Menge von Zustandsgrößen des Systems zu. Die Zuordnungsvorschrift wird durch das Modell des Systems beschrieben. Diese Zuordnung wird in Abbildung 2.1 dargestellt. Die Parameter des Parametervektors \vec{p} spannen den Parameterraum \mathcal{P} auf. Die Zustandsgrößen des Zustandsvektors \vec{x} spannen einen weiteren Raum, den Zustandsraum \mathcal{X} , auf. Formal definiert ist eine Simulation eine Abbildung aus dem Parameterraum \mathcal{P} in den Zustandsraum \mathcal{X} . Handelt es sich um eine transiente Simulation wird nicht ein einzelner Punkt im Zustandsraum, sondern eine Trajektorie im Zustandsraum bestimmt. Die Zuordnungsvorschrift ist in der Regel ein mathematisches Modell, d.h. eine Funktion $f : \mathcal{P} \rightarrow \mathcal{X}$. Häufig sind nicht die Zustandsgrößen selbst von Interesse, sondern auch die daraus abgeleiteten Eigenschaften. Der beschreibende Eigenschaftsraum \mathcal{Y} wird aus dem Zustandsraum wiederum durch eine Abbildung $f' : \mathcal{X} \rightarrow \mathcal{Y}$ bestimmt. Die Abbildungsvorschriften können durch verschiedene Formen von Gleichungssystemen gegeben sein. Im Fall eines linearen Gleichungssystems wird die Abbildung durch die Koeffizientenmatrix \mathbf{A} und den Vektor der Absolutglieder \vec{b} bestimmt. Bei parametrischen Systemen wird die Lösung durch die Parameter des Parametervektor \vec{p} beeinflusst. Im Folgenden sind die wichtigsten Arten von Gleichungssystemen

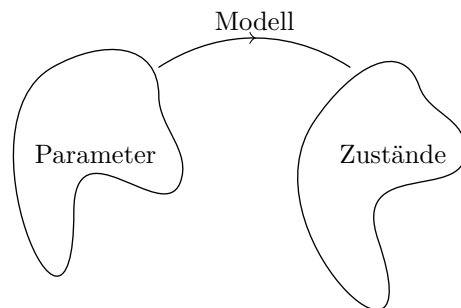


Abbildung 2.1: Simulation: Zuordnung einer Menge von Parametern zu einer Menge von Zuständen

2 Simulation von Parameterunsicherheiten

zusammengefasst:

$$\vec{x} = f_{\text{lin}}(\vec{b}) = \mathbf{A}^{-1}\vec{b} \quad \text{explizit und linear} \quad (2.1)$$

$$\vec{x} = f_{\text{lin}}(\vec{b}, \vec{p}) = \mathbf{A}(\vec{p})^{-1}\vec{b}(\vec{p}) \quad \text{explizit, linear und parametrisch} \quad (2.2)$$

$$\vec{x} = f_{\text{NL}}(\vec{p}) \quad \text{explizit und nichtlinear} \quad (2.3)$$

$$f_{\text{lin}}(\vec{x}) = \mathbf{A}\vec{x} = \vec{b} \quad \text{implizit und linear} \quad (2.4)$$

$$f_{\text{lin}}(\vec{x}) = \mathbf{A}(\vec{p})\vec{x} = \vec{b}(\vec{p}) \quad \text{implizit, linear und parametrisch} \quad (2.5)$$

$$f_{\text{NL}}(\vec{x}) = \vec{0} \quad \text{implizit und nichtlinear} \quad (2.6)$$

$$f_{\text{NL}}(\vec{x}, \vec{p}) = \vec{0} \quad \text{implizit, nichtlinear und parametrisch} \quad (2.7)$$

Für zeitabhängige Differentialgleichungssysteme (DGLs) wird unterschieden, ob die Ableitungen explizit oder implizit gegeben sind. Die Funktion f kann dabei jeweils linear oder nichtlinear sein.

$$\dot{\vec{x}} = f(\vec{x}) \quad \text{explizit} \quad (2.8)$$

$$\dot{\vec{x}} = f(\vec{x}, \vec{p}) \quad \text{explizit und parametrisch} \quad (2.9)$$

$$f(\vec{x}, \dot{\vec{x}}) = \vec{0} \quad \text{implizit} \quad (2.10)$$

$$f(\vec{x}, \dot{\vec{x}}, \vec{p}) = \vec{0} \quad \text{implizit und parametrisch.} \quad (2.11)$$

$$(2.12)$$

Sind alle Parameter $\vec{p} \in \mathcal{P}$ exakt bekannt, besteht der Parameterraum \mathcal{P} aus einem Punkt. Sind auch die Abbildungen exakt berechenbar, wird \mathcal{P} in einen einzigen Punkt im Zustandsraum \mathcal{X} bzw. Eigenschaftsraum \mathcal{Y} abgebildet. Diese Simulationsart wird als Nominalsimulation, die Lösungen werden als Nominallösungen bezeichnet. Wenn die Parameter \vec{p} mit Unsicherheiten behaftet sind, beschreiben sie ein Gebiet in \mathcal{P} , das auf ein Gebiet in \mathcal{X} bzw. \mathcal{Y} abgebildet wird. Beide Situationen werden in Abbildung 2.2 dargestellt.

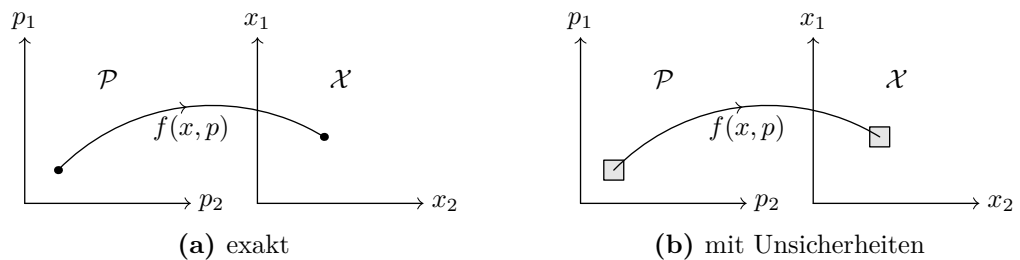


Abbildung 2.2: Gegenüberstellung der Abbildung vom Parameterraum in den Zustandsraum für exakte und mit Unsicherheiten behaftete Parameter

Verfahren zur Simulation von Systemen mit Parameterunsicherheiten lassen sich in punkt- und gebietsarithmetische Ansätze unterteilen. Bei den punktarithmetischen Verfahren wird ein Nominalparametersatz, d.h. ein einzelner Punkt im Parameterraum,

ausgewählt und mit diesem eine Nominalsimulation durchgeführt. Dies wird für mehrere Parametersätze wiederholt. Beim Corner-Case-Verfahren (CC) werden die Extremwerte der Parameter ausgewählt und mit diesen in allen möglichen Kombinationen Nominalsimulationen durchgeführt. Für n_p Parameter entspricht dies 2^{n_p} Nominalsimulationen. Dabei ist nicht sichergestellt, dass der Worst Case des Systemverhaltens in den Simulationen erfasst wird. Ist die Abbildung $\mathcal{P} \rightarrow \mathcal{Y}$ nichtmonoton, treten die Extremwerte des Eigenschaftsraumes nicht für die Extremwerte des Parameterraumes auf. Bei dem Monte-Carlo-Verfahren (MC) werden Stichproben für die mit Unsicherheiten behafteten Parameter gemäß einer Wahrscheinlichkeitsverteilung erzeugt. Mit jedem der so erzeugten n Parametersätze wird eine Nominalsimulation durchgeführt. Dabei werden die Extremwerte nur dann angenähert, wenn die Anzahl der Stichproben ausreichend hoch ist. Beide Verfahren sind beispielhaft für die nichtmonotone Funktion $x = p^2$ in Abbildung 2.3 gegenübergestellt.

Ein weiteres punktarithmetisches Verfahren ist die Sensitivitätsanalyse. Dabei wird jeder Parameter p_i einzeln von seinem Nominalwert $p_{i,\text{nom}}$ um einen kleinen Betrag Δp_i ausgelenkt. Die übrigen Parameter $p_j \in \vec{p}$ mit $j \neq i$ werden auf ihren Nominalwert $p_{j,\text{nom}}$ gesetzt. Durch die Bestimmung der Differenz $\vec{x}(p_i = p_{i,\text{nom}} + \Delta p, p_j) - \vec{x}(p_i = p_{i,\text{nom}} - \Delta p, p_j)$ kann die Empfindlichkeit aller Zustandsgrößen \vec{x} von dem Parameter p_i angegeben werden. Die Ergebnisse stellen eine Linearisierung der Empfindlichkeit im Punkt $p_{i,\text{nom}}$ dar. Ist im obigen Beispiel der Funktion $x = p^2$ mit $p_{\text{nom}} = 0$, folgt demnach eine Unabhängigkeit der Größe x vom Parameter p . Diese Situation ist in Abbildung 2.4 dargestellt. Über dem betrachteten Intervall besteht aber entgegen der aus der Sensitivitätsanalyse abgeleiteten Aussage eine signifikante Abhängigkeit. In dem Teilintervall $p < 0$ ist die Abhängigkeit monoton fallend, für $p > 0$ monoton steigend.

Im Gegensatz zu punktarithmetischen Verfahren beschreiben gebietsarithmetische Verfahren nicht einzelne Punkte in den Parameter-, Zustands- und Eigenschaftsräumen, sondern Gebiete. Aus einem Gebiet im Parameterraum wird durch die Abbildung mit

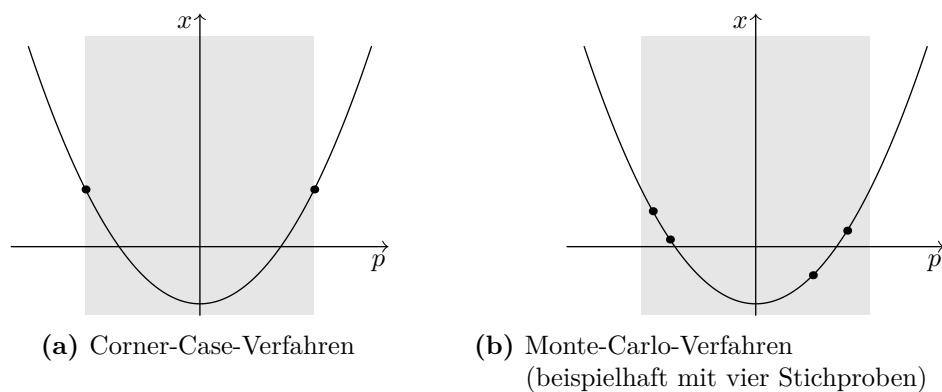


Abbildung 2.3: Gegenüberstellung der durch das Monte-Carlo- und Corner-Case-Verfahren ausgewählten Punkte für die nichtmonotone Funktion $x = p^2$. Das betrachtete Intervall ist grau dargestellt, das Minimum $p = 0$ wird in beiden Fällen nicht gefunden

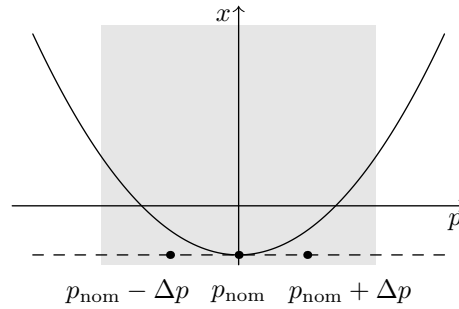


Abbildung 2.4: Sensitivitätsanalyse der Funktion $x = p^2$ im Punkt $p_{\text{nom}} = 0$, das betrachtete Intervall ist grau dargestellt

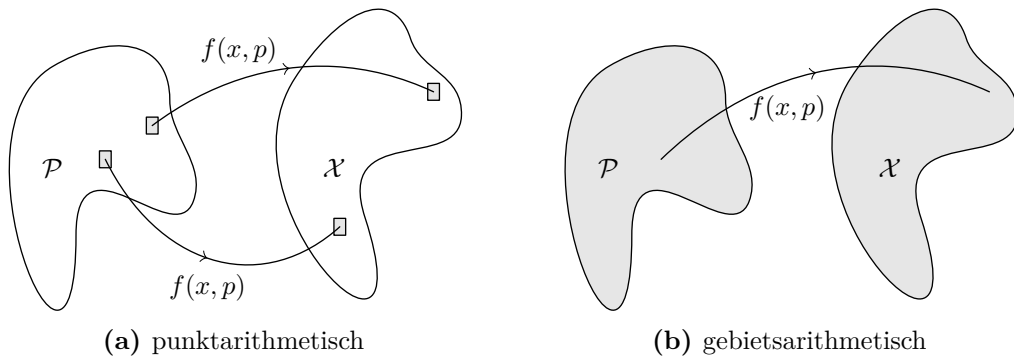


Abbildung 2.5: Gegenüberstellung von punkt- und gebietsarithmetischer Abbildung vom Parameterraum in den Zustandsraum

gebietsarithmetischen Verfahren ein Gebiet im Zustands- bzw. Eigenschaftsraum. In Abbildung 2.5 werden die punkt- gebietsarithmetischen Abbildungen gegenübergestellt. Gebietsarithmetische Verfahren sind laufezeitintensiver als punktarithmetische Verfahren, bieten jedoch den Vorteil, dass eine Simulation den kompletten Parameterraum abdeckt. Dafür garantieren sie unabhängig von den Monotonieeigenschaften der Abbildung den Einschluss. Dieser Einschluss ist jedoch nicht minimal, das heißt größer als unbedingt benötigt. Diese Eigenschaft wird als Überschätzung bezeichnet, Details hierzu folgen in Abschnitt 2.1.1.

2.1 Grundlagen von Gebietsarithmetiken

Gebietsarithmetiken beschreiben Größen nicht durch exakte Zahlen, sondern durch Gebiete. Mit Unsicherheiten behaftete Größen werden als Gebiete modelliert, sodass die tatsächliche Größe innerhalb dieses Gebiets liegt. Die Gebiete können verschiedene Darstellungsformen wie zum Beispiel Intervalle (vgl. Abschnitt 2.1.2), affine Formen (vgl. Abschnitt 2.1.3) oder andere Darstellungsformen (vgl. Abschnitt 2.1.4) besitzen. Mit

$$\check{x} \in \check{\mathbb{R}} \quad (2.13)$$

wird im Folgenden ein allgemeines Gebiet beschrieben. Eine wichtige Eigenschaft von Gebietsarithmetiken ist die Einschließungseigenschaft oder Inklusionsisotonie. Alle gebietsarithmetische Abbildungen

$$\check{f}(\check{p}) : \check{\mathbb{R}} \rightarrow \check{\mathbb{R}} \quad (2.14)$$

sind inklusionsisoton bezüglich der entsprechenden punktarithmetischen Funktion

$$f(p) : \mathbb{R} \rightarrow \mathbb{R} \quad (2.15)$$

definiert. Das heißt, es gilt:

$$\forall (q \in \mathbb{R}, \check{q} \in \check{\mathbb{R}}, q \in \check{q}) : (f(q) \in \check{f}(\check{q})) \quad (2.16)$$

und

$$\forall (\check{r}, \check{s} \in \check{\mathbb{R}}, \check{r} \subseteq \check{s}) : (\check{f}(\check{r}) \subseteq \check{f}(\check{s})). \quad (2.17)$$

Alle Punkte des Eingangsgebiets werden auf Punkte des Ausgangsgebiets abgebildet.

2.1.1 Überschätzung

Die inklusionsisotone Abbildung des Eingangs- auf das Ausgangsgebiets ist aus verschiedenen Gründen im Allgemeinen größer als unbedingt benötigt, um die Abbildung aller Punkte aus \mathcal{P} aufzunehmen. Diese Eigenschaft ist in Abbildung 2.6 grafisch dargestellt. Die minimale Abbildung ist wie folgt definiert:

Definition 2.1. Eine gebietsarithmetische Abbildung $\mathcal{P} \rightarrow \mathcal{Y}$ ist minimal, wenn sie dem Gebiet entspricht, das sich durch die Einhüllende der entsprechenden punktarithmetischen Transformationen (mit unendlicher Genauigkeit) aller Punkte $p \in \mathcal{P}$ ergibt.

Die minimale Abbildung $\vec{y}_{\text{minimal}} = \check{f}_{\text{minimal}}(\vec{x})$ ist im Allgemeinen nur sehr aufwändig berechenbar. Stattdessen wird die Abbildung $\vec{y} = \check{f}(\vec{x})$ berechnet, die \vec{y}_{minimal} inklusionsisoton einschließt:

$$\check{f}(\vec{x}) \subseteq \check{f}_{\text{minimal}}(\vec{x}) \quad (2.18)$$

$$\vec{y} \subseteq \vec{y}_{\text{minimal}}. \quad (2.19)$$

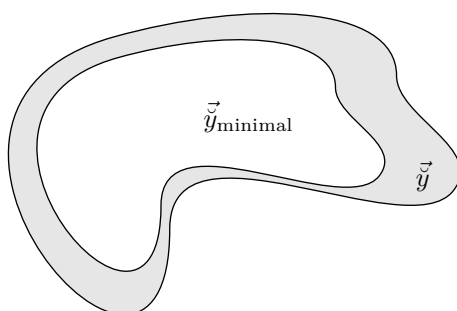


Abbildung 2.6: Überschätztes Gebiet \vec{y} im Vergleich mit dem minimalen Gebiet \vec{y}_{minimal} . Die Überschätzung ist grau dargestellt.

Die Differenz zwischen minimalem und dem tatsächlich berechneten Gebiet wird als Überschätzung (vgl. graues Gebiet in Abbildung 2.6) bezeichnet. Die Überschätzung der minimalen Abbildung ist definitionsgemäß null.

Eine Überschätzung ist unerwünscht, weil neben der Eigenschaft des garantierten Einschlusses ein möglichst enger Einschluss gewünscht ist, um möglichst scharf begrenzte Aussagen über das Systemverhalten treffen zu können. Ein zusätzliches Problem ergibt sich, wenn ein überschätztes Gebiet als Eingangsgröße für eine Funktion mit beschränktem Definitionsbereich \mathbb{D} verwendet wird. Durch die Überschätzung kann die Eingangsgröße den Definitionsbereich der Funktion verlassen, obwohl das minimale Gebiet noch innerhalb des Definitionsbereiches liegt. In diesen Fällen ist eine gebietsarithmetische Auswertung der Funktion nicht möglich. Beispiele für Funktionen mit beschränktem Definitionsbereich sind \sqrt{x} mit $x \in \mathbb{D} = \mathbb{R}^{\geq 0}$ oder $\frac{1}{x}$ mit $x \in \mathbb{D} = \mathbb{R}^{\neq 0}$. Eine solche Situation ist in Abbildung 2.7 dargestellt.

In den folgenden Abschnitten werden einige Gebietsarithmetiken vorgestellt. Details zur Überschätzung in einer dieser Gebietsarithmetiken, der affinen Arithmetik, folgen in Abschnitt 2.1.3.2.

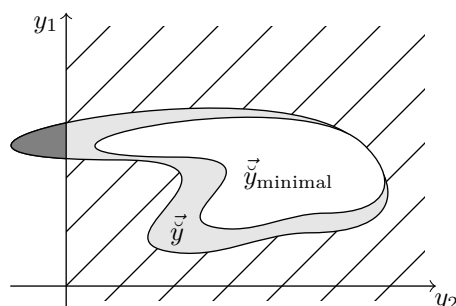


Abbildung 2.7: Gegenüberstellung des überschätzten Gebiet \vec{y} mit dem minimalen Gebiet \vec{y}_{minimal} im Vergleich zum Definitionsbereich $y_2 \in \mathbb{D} = \mathbb{R}^{>0}$ (schraffiert dargestellt). Ein Teil des überschätzten Gebiets liegt außerhalb des Definitionsbereiches (dunkelgrau dargestellt), während das minimale Gebiet vollständig innerhalb des Definitionsbereiches liegt

2.1.2 Intervallarithmetik

Die Intervallarithmetik ist die einfachste Gebietsarithmetik. Häufig wird Ramon E. Moore mit [63] aus dem Jahr 1966 als Begründer der modernen Intervallarithmetik betrachtet. Seine ersten Ideen zur Intervallarithmetik wurden jedoch bereits 1959 veröffentlicht [62]. Anstelle von Zahlen, die einen Punkt im Raum beliebig genau beschreiben, werden Intervalle zur Beschreibung von mit Unsicherheiten behafteten Größen eingesetzt. Ein Intervall $[x]$ wird durch seine Untergrenze \underline{x} und die Obergrenze \bar{x} beschrieben:

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\}. \quad (2.20)$$

Es handelt sich dabei um ein abgeschlossenes Intervall. In der Mathematik sind auch andere Definitionen von Intervallen wie offen oder halboffen üblich, in der Intervalla-

rithmetik wird der Intervallbegriff jedoch gleichbedeutend mit geschlossenem Intervall verwendet. Mit Gleichung (2.21) lässt sich auch ein intervallwertiger Vektor $[\vec{x}]$ definieren, dessen Komponenten Intervalle sind:

$$[\vec{x}] = \begin{pmatrix} [x_1] \\ [x_2] \\ \vdots \\ [x_n] \end{pmatrix}. \quad (2.21)$$

Geometrisch entspricht dies einer achsenparallelen Box. Mit dem Operator $\text{mid}([x])$ kann der Mittelpunkt des Intervalls $[x]$ bestimmt werden:

$$\text{mid}([x]) = \frac{\underline{x} + \bar{x}}{2}. \quad (2.22)$$

Er lässt sich auch komponentenweise auf intervallwertige Vektoren $[\vec{x}]$ oder Matrizen $[x]$ anwenden. Die arithmetischen Operationen Addition, Subtraktion, Multiplikation und Division werden auf die Betrachtung der Intervallgrenzen zurückgeführt:

$$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}] \quad (2.23)$$

$$[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}] \quad (2.24)$$

$$[x] \cdot [y] = [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})] \quad (2.25)$$

$$[x]/[y] = [x] \cdot [1/\bar{y}, 1/\underline{y}] \quad \text{mit} \quad 0 \notin [y]. \quad (2.26)$$

Zur Darstellung von nichtlinearen, aber monotonen Funktionen ist die Betrachtung der Intervallgrenzen ausreichend:

$$f_{\text{mon}}([x]) = \begin{cases} [f_{\text{mon}}(\underline{x}), f_{\text{mon}}(\bar{x})] & \text{wenn } f(x) \text{ monoton steigend} \\ [f_{\text{mon}}(\bar{x}), f_{\text{mon}}(\underline{x})] & \text{wenn } f(x) \text{ monoton fallend.} \end{cases} \quad (2.27)$$

Die monoton wachsende Exponentialfunktion beispielsweise lässt sich wie folgt darstellen:

$$f_{\text{mon}}([x]) = e^{[x]} = [e^{\underline{x}}, e^{\bar{x}}]. \quad (2.28)$$

Nichtmonotone Funktionen sind innerhalb eines beschränkten Gebiets monoton. So ist die Sinusfunktion $\sin(x)$ in ihrem Definitionsbereich $\mathbb{D} = \mathbb{R}$ nichtmonoton. Beschränkt man den Definitionsbereich \mathbb{D} auf $-\frac{\pi}{2} \leq x \leq \frac{\pi}{2}$, ist sie innerhalb dieses Bereichs monoton wachsend und kann dementsprechend wie eine monotone intervallarithmetische Funktion behandelt werden:

$$\sin([x]) = [\sin(\underline{x}), \sin(\bar{x})] \quad \text{für} \quad \mathbb{D} : -\frac{\pi}{2} \leq x \leq \frac{\pi}{2}. \quad (2.29)$$

Wird durch die Anwendung ein größerer Definitionsbereich benötigt, können die trigonometrischen Funktionen \arcsin , \arccos , \arctan , arccotan und \sinh durch eine Taylorapproximation mit adaptiver Ordnungswahl und speziellen Reduktionen dargestellt werden. Die

2 Simulation von Parameterunsicherheiten

übrigen trigonometrischen Funktionen verwenden eine Taylorapproximation mit fester Ordnung. Details sind in [39, 44] nachzulesen.

Eine weitere wichtige Operation ist der Schnitt zweier Intervalle:

$$[x] \cap [y] = \begin{cases} \emptyset & \text{für } \bar{x} < \underline{y} \vee \bar{y} < \underline{x} \\ [\max(x, y), \min(\bar{x}, \bar{y})] & \text{sonst.} \end{cases} \quad (2.30)$$

Die Berechnungen der Intervallarithmetik sind minimal, wenn keine Korrelationen der betrachteten Größen vorhanden sind. Abhängigkeiten der Intervalle untereinander werden von der Intervallarithmetik nicht berücksichtigt. Dies kann zu einer großen Überschätzung der Ergebnisse führen. Diese Eigenschaft wird in der Literatur als Fehlerexplosion (engl. error explosion) bezeichnet. So gilt für die Potenzfunktion

$$[x]^2 = [-1, 1]^2 = [0, 1] \quad (2.31)$$

und für die Umformung

$$[x] \cdot [x] = [-1, 1] \cdot [-1, 1] = [-1, 1]. \quad (2.32)$$

Das Intervall $[-1, 1]$ enthält die minimale Lösung $[0, 1]$, überschätzt die Lösung aber stark. Das Ausgangsgebiet $[-1, 1]$ ist doppelt so groß wie das minimale Gebiet. Dies liegt daran, dass der Multiplikationsfunktion keine Informationen über die Abhängigkeit der Eingangsgröße $[x]$ zu sich selbst vorliegen und sie daher die Eingangsgrößen als voneinander vollständig unabhängige Größen $[x]$ und $[y]$ behandelt.

Eine weitere unerwünschte Eigenschaft der Intervallarithmetik ist der sogenannte Wrapping-Effekt. Wenn das optimale Lösungsgebiet nicht zufällig selbst eine achsenparallele Box darstellt, muss es durch eine größere achsenparallele Box eingeschlossen werden, da mit der Intervallarithmetik nur achsenparallele Gebiete dargestellt werden können. Dies führt zu einer Überschätzung des Lösungsraumes, dies ist grafisch in Abbildung 2.8 dargestellt. Weitergehende Informationen zur Intervallarithmetik finden sich in [63–65] und den dortigen Referenzen.

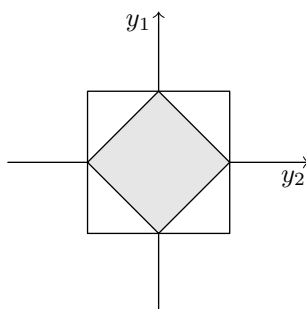


Abbildung 2.8: Darstellung des Wrapping-Effekts. Nichtachsenparallele Boxen (grau) müssen in der Intervallarithmetik durch größere achsenparallele Boxen eingeschlossen werden.

2.1.3 Affine Arithmetik

Die affine Arithmetik ist eine Weiterentwicklung der Intervallarithmetik. Sie wurde zum ersten Mal 1997 von Stolfi und de Figueiredo in [21] vorgestellt. Im Gegensatz zur Intervallarithmetik werden bei der affinen Arithmetik lineare Korrelationsinformationen während der Berechnung erhalten und mitgeführt. Korrelierende Größen können sich gegenseitig (teilweise) auslöschen und so die Überschätzung verringern. Das Abhängigkeitsproblem der Intervallarithmetik wird so verringert. In der affinen Arithmetik werden Größen durch affine Formen dargestellt:

$$\hat{x} = x_0 + \sum_{i \in \mathbb{I}_{\hat{x}}} x_i \varepsilon_i. \quad (2.33)$$

x_0 wird dabei als Zentralwert bezeichnet. Die partiellen Abweichungen $x_i \in \mathbb{R}$ werden von den Abweichungssymbolen $\varepsilon_i \in [-1, 1]$ skaliert. Das Produkt $x_i \varepsilon_i$ wird als Abweichungsterm bezeichnet. Jedes Abweichungssymbol ε_i korrespondiert mit einer Parameterunsicherheit. Abweichungssymbole mit unterschiedlichen Indizes sind linear unabhängig.

Die maximale Abweichung vom Zentralwert x_0 wird als Radius der affinen Form bezeichnet und wie folgt berechnet:

$$\text{rad}(\hat{x}) = \sum_{i \in \mathbb{I}_{\hat{x}}} |x_i|. \quad (2.34)$$

Mit dem Minimum bzw. Maximum einer affinen Form ist eine Umwandlung in ein Intervall möglich:

$$\min(\hat{x}) = x_0 - \text{rad}(\hat{x}) \quad (2.35)$$

$$\max(\hat{x}) = x_0 + \text{rad}(\hat{x}). \quad (2.36)$$

Die Umwandlung in das entsprechende Intervall lautet:

$$[x] = [\min(\hat{x}), \max(\hat{x})]. \quad (2.37)$$

Entsprechend zu Gleichung (2.21) lässt sich auch ein affinwertiger Vektor $\vec{\hat{x}}$ definieren, dessen Komponenten affine Formen sind:

$$\vec{\hat{x}} = \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{pmatrix}. \quad (2.38)$$

Das n -te Element des Vektors $\vec{\hat{x}}$ wird im Folgenden mit $\vec{\hat{x}}_{(n)}$ bezeichnet. wird das benötigt?

Arithmetische Operationen mit affinen Formen werden so definiert, dass sie als Ergebnis wieder affine Formen zurückliefern. Die inklusionsisotone Erweiterung lässt sich wie folgt

2 Simulation von Parameterunsicherheiten

darstellen:

$$\hat{f}(\hat{x}) = \alpha + \beta\hat{x} + \gamma\varepsilon_k, \quad \varepsilon_k \notin \mathbb{E}_{\hat{x}} \quad (\text{für univariate Funktionen}) \quad (2.39)$$

$$\hat{f}(\hat{x}, \hat{y}) = \alpha + \beta_x\hat{x} + \beta_y\hat{y} + \gamma\varepsilon_k, \quad \varepsilon_k \notin (\mathbb{E}_{\hat{x}} \cup \mathbb{E}_{\hat{y}}) \quad (\text{für bivariate Funktionen}). \quad (2.40)$$

Die Variable γ schließt dabei etwaige Approximationsfehler ein. Einige Operationen sind mit Hilfe von affinen Formen direkt darstellbar, sodass die Abbildung minimal erfolgt. Diese Funktionen werden affine Funktionen genannt. Zu ihnen gehören die Addition, die Subtraktion und die Multiplikation mit einer skalaren Konstanten:

$$c(\hat{x} \pm \hat{y}) = c(x_0 \pm y_0) + \sum_{i \in (\mathbb{I}_{\hat{x}} \cup \mathbb{I}_{\hat{y}})} c(x_i \pm y_i)\varepsilon_i. \quad (2.41)$$

In diesen Fällen ist die Erweiterung $\gamma = 0$. Bei allen anderen Operationen ist das Ausgangsgebiet nicht direkt als affine Form darstellbar. Daher wird ein neues Abweichungssymbol $\gamma \neq 0$ eingeführt. In der Umgebung von x_0 lässt sich die Funktion f durch die Taylor-Entwicklung

$$f(\hat{x}) = f(x_0) + f'(x_0) \cdot (\hat{x} - x_0) + \frac{f''(\nu)}{2} \cdot (\hat{x} - x_0)^2 \quad (2.42)$$

darstellen. Der quadratische Term $\frac{f''(\nu)}{2} \cdot (\hat{x} - x_0)^2$ lässt sich nicht mit Hilfe einer affinen Form ausdrücken. Zur Sicherstellung des Einschlusses wird Gleichung (2.39) um folgende Nebenbedingung ergänzt:

$$\gamma = \max_{x \in \hat{x}} (\alpha + \beta \cdot x - f(x)). \quad (2.43)$$

Zur Bestimmung von α , β und γ unter dieser Nebenbedingung können verschiedene Optimierungsverfahren eingesetzt werden. Zwei mögliche Verfahren sind die Minimalbereichserweiterung und das Tschebyscheff-Verfahren (auch Sekantenverfahren genannt). Die Minimalbereichserweiterung minimiert den Radius der Abbildung $f(\hat{x})$, während das Tschebyscheff-Verfahren den Approximationsfehler γ minimiert. Die Minimalbereichserweiterung bietet den Vorteil, Minimum und Maximum der nichtlinearen Funktion exakt darzustellen, erzeugt dafür aber einen größeren Approximationsfehler γ . Das Tschebyscheff-Verfahren erhält mehr lineare Korrelationsinformation aus dem Eingangsgebiet, über- bzw. unterschätzt dabei aber die nominalwertige Funktion an der oberen Gebietsgrenze $\max(\hat{x})$ bzw. unteren Gebietsgrenze $\min(\hat{x})$. Ein Beispiel zur Berechnung der beiden Approximationen folgt in Abschnitt 2.1.3.1. Nähere Informationen zur Bestimmung von α , β , γ mit beiden Verfahren finden sich in [21]. Eine Minimalbereichserweiterung für Multiplikation und Division ist in [89] dargestellt. Eine Erweiterung auf allgemeine multivariate Ausdrücke, d.h. Abhängigkeiten von mehreren Variablen, wird in [27, Abschnitt 3.2.6] vorgestellt. In [78] wird ein Verfahren vorgestellt, dass die Vorteile von Minimalbereichserweiterung und Tschebyscheff-Verfahren auf Kosten der Laufzeit vereint. Nach Durchführung der Approximation kann das resultierende Ausgangsgebiet in folgender Form dargestellt werden:

$$\hat{y} = f_{\text{nichtaffin}}(\hat{x}) = y_0 + \sum_{i \in \mathbb{E}_{\hat{x}}} (y_i \varepsilon_i) + y_{n+1} \varepsilon_{n+1}. \quad (2.44)$$

Der neue Abweichungsterm $y_{n+1}\varepsilon_{n+1}$ schließt den Approximationsfehler auf dem durch \hat{x} beschriebenen Gebiet ein. Er ist unkorreliert zu allen bisher bestehenden Abweichungssymbolen und erweitert die globale Menge der Indizes bzw. Abweichungssymbole:

$$\mathbb{I}_{\text{neu}} = \mathbb{I}_{\text{alt}} \cup \{n+1\}, (n+1) \notin \mathbb{I}_{\text{alt}} \quad (2.45)$$

$$\mathbb{E}_{\text{neu}} = \mathbb{E}_{\text{alt}} \cup \{\varepsilon_{n+1}\}, \varepsilon_{n+1} \notin \mathbb{E}_{\text{alt}}. \quad (2.46)$$

Diese bei der Auswertung von nichtaffinen Funktionen f_{NA} entstehenden Abweichungssymbole werden im Folgenden als Nonlinear Partial Deviations (NLPDs) bezeichnet. Die zweiten Ableitungen der meisten Elementarfunktionen sind monoton, deshalb muss in diesen Fällen keine Extremwertsuche zur Bestimmung des maximalen Approximationsfehlers durchgeführt werden. Zusammengesetzte Funktionen dagegen besitzen im Allgemeinen nichtmonotone zweite Ableitungen. Um eine aufwändige Extremwertsuche zu vermeiden, wird auf die exakte Bestimmung des maximalen Approximationsfehlers verzichtet. Stattdessen werden zusammengesetzte Funktionen in eine Abfolge von Elementarfunktionen zerlegt. Diese wird schrittweise abgearbeitet und ergibt so ein Gebiet, das das minimale Gebiet überschätzt, aber garantiert einschließt.

2.1.3.1 Beispiel: Exponentialfunktion mit Tschebyscheff-Verfahren und Minimalbereichserweiterung

Mit den Abkürzungen

$$a = \min(\hat{x}) \quad (2.47)$$

$$b = \max(\hat{x}) \quad (2.48)$$

$$f_a = e^a \quad (2.49)$$

$$f_b = e^b \quad (2.50)$$

gilt für das Tschebyscheff-Verfahren angewendet auf die Exponentialfunktion $\hat{y} = e^{\hat{x}}$ für die Hilfsgrößen α , β und ζ :

$$\alpha = \frac{f_b - f_a}{b - a} \quad (2.51)$$

$$\delta = \frac{1}{2}(f_a + \alpha(u - a - 1)) \quad \text{mit} \quad u = f^{-1}(\alpha) = \log(\alpha) \quad (2.52)$$

$$\zeta = f_a - \alpha \cdot a - \delta. \quad (2.53)$$

Für die Minimalbereichserweiterung der gleichen Funktion folgt für die Hilfsgrößen

$$\alpha = f_a \quad (2.54)$$

$$\delta = \frac{1}{2}(f_b - f_a - \alpha(b - a)) \quad (2.55)$$

$$\zeta = \frac{1}{2}(f_a + f_b - \alpha(a + b)), \quad (2.56)$$

2 Simulation von Parameterunsicherheiten

da $f'(x) = f(x) = e^x$ gilt und aus der streng monoton steigenden Eigenschaft folgt, dass die betragsmäßig kleinste Ableitung in $\min(\hat{x})$ auftritt. Mit $y_0 = \alpha x_0 + \zeta$, $y_i = \alpha x_i$, $y_{n+1} = \delta$ ergibt sich für die Auswertung der affinen Exponentialfunktion mit beiden Verfahren folgender Gesamtausdruck in der Form von Gleichung (2.44):

$$\hat{y} = \hat{f}(\hat{x}) = \alpha x_0 + \zeta + \sum_{\mathbb{I}_x} (\alpha x_i \varepsilon_i) + \delta \varepsilon_{n+1}. \quad (2.57)$$

In Abbildung 2.9 werden beide Verfahren grafisch gegenübergestellt.

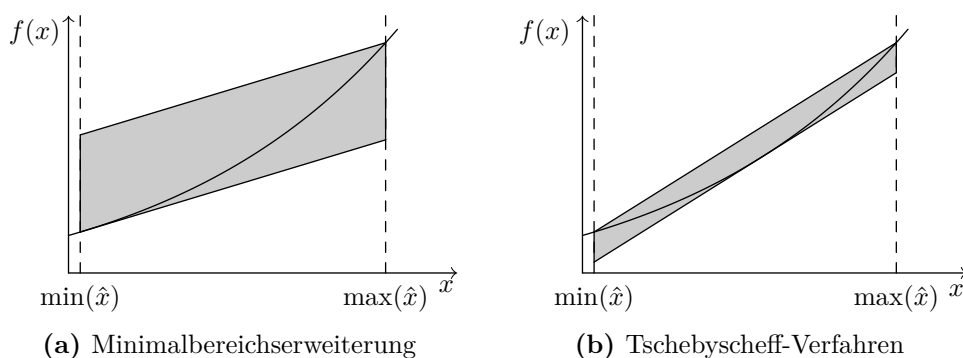


Abbildung 2.9: Approximation am Beispiel der Exponentialfunktion

2.1.3.2 Überschätzung in der affinen Arithmetik

Überschätzung entsteht in der affinen Arithmetik zum einen durch die Approximation nichtlinearer Funktionen mit dem Minimalbereichs- oder Tschebyscheff-Verfahren, zum anderen durch den Verzicht auf den optimalen Einschluss zusammengesetzter Funktionen. Die Überschätzung der affinen Arithmetik ist quadratisch von der Größe des Arguments abhängig, im Gegensatz zur linearen Abhängigkeit der Intervallarithmetik [[21] und [27, Tabelle 3.2]]. Ein möglicher Ansatz für die Minimierung der Überschätzung ist eine Verbesserung der Approximation von nichtaffinen Funktionen. Einige erweiterte Gebietsarithmetiken, die dieses Ziel verfolgen, werden in Abschnitt 2.1.4 vorgestellt. Ein weiterer Ansatz ist die Optimierung von Gleichungen. Durch Umstellen der Gleichungen kann unter Umständen eine Form gefunden werden, die mathematisch äquivalent ist, aber eine geringere Überschätzung verursacht. Dies kann durch Umordnung der Terme oder durch die Verringerung der Anzahl der nichtaffinen Operationen geschehen. Einige Beispiele für Optimierungen von einzelnen Gleichungen aus dem BSIMSOI-Transistormodell sind in [84, 85] zu finden. Im Allgemeinen ist dieses Verfahren nur begrenzt einsetzbar, da eine Gleichung des Modells wiederum in mehreren anderen eingesetzt wird. Eine Optimierung muss daher für jede dieser Einsetzungen einzeln geschehen, um die jeweilige Überschätzung möglichst gering zu halten. Die Überschätzung lässt sich des Weiteren durch die Wahl eines kleineren Eingangsgebiets verringern. Dieses ist jedoch durch die Anwendung vorgegeben. Eine Möglichkeit, um die Überschätzung zu verkleinern, ist daher, das Eingangsgebiet aufzuteilen, die Lösungen getrennt zu berechnen und sie wieder zusammenzufügen. Ein entsprechendes Verfahren wird in Abschnitt 3.4 vorgestellt.

2.1.4 Weitere Gebietsarithmetiken

Neben den bisher vorgestellten Gebietsarithmetik-Typen wurden in der Literatur weitere Typen entwickelt. Eine Verbesserung stellt die verallgemeinerte Intervallarithmetic (engl. Generalized Interval Arithmetic) von Hansen [33] dar. Sie vermeidet das Abhängigkeitsproblem der Intervallarithmetic durch eine alternative Darstellung der Intervalle und ist damit ähnlich zur affinen Arithmetik. Andere Erweiterungen der Intervallarithmetic lassen neben finiten Intervallgrenzen auch infinite zu [9, 57]. Dadurch können zum Beispiel Divisionen mit Divisoren, die die Null enthalten, durchgeführt werden. Andere Autoren schlagen eine Anpassung der Darstellung von affinen Formen vor. Eine Möglichkeit ist die Erweiterung zu einer quadratischen Arithmetik mit quadratisch-affinen Formen [27, Kapitel 3.3] und [59, Abschnitt 4.1]:

$$\tilde{x} = x_0 + \sum_{i \in \mathbb{I}_{\tilde{x}}} x_i \varepsilon_i + \sum_{i \in \mathbb{I}_{\tilde{x}}} \sum_{\substack{j \geq i \\ j \in \mathbb{I}_{\tilde{x}}}} x_{ij} \varepsilon_{ij}. \quad (2.58)$$

Dadurch können nichtaffine Funktionen mit einem Polynom 2. Ordnung angenähert und die Approximationsfehler verringert werden. In [59, 60] wird die Einführung von speziellen Abweichungstermen für positive $\varepsilon_+ \in [0, 1]$ bzw. negative $\varepsilon_- \in [-1, 0]$ Anteile vorgeschlagen (Extended Affine Arithmetic, EAA). Die Erweiterung von Kolev in [47, 49, 50] führt dagegen ein Intervall als Erweiterung der affinen Form ein. Diese Formen werden von Kolev als verallgemeinerte Intervalle (Generalized Intervals) bezeichnet, sind aber nicht identisch mit der Generalized Interval Arithmetic von Hansen [33]. Die Grundidee wird bereits in [21, Abschnitt 3.15] genannt. Um der Nomenklatur von [100] zu entsprechen, wird im Folgenden trotz der Mehrdeutigkeit die Abkürzung GIA für die verallgemeinerten Intervalle von Kolev verwendet. Die Idee einer affinen Form mit Intervallerweiterung wird auch in [82] und [109, Abschnitt 5.4] weiter verfolgt. Die letztere Erweiterung wird als Revised Affine Arithmetic (RAA) bezeichnet. Eine Verbesserung in der Auswertung einer Multiplikation von zwei affinen Formen wird in [91] vorgestellt. Dort wird eine globale Optimierung zur Bestimmung des Minimalbereichs durchgeführt. Eine optimierte Division zweier affiner Formen wird in [61] vorgestellt.

2.1.5 Bewertung der verschiedenen Gebietsarithmetiken

Ein systematischer Vergleich der verschiedenen Gebietsarithmetiken für verschiedene arithmetische Funktionen ist in der Literatur nicht zu finden. Für die Multiplikation in den Gebietsarithmetiken AA, EAA, RAA, GIA findet sich ein Vergleich in [100]. AA bezeichnet dabei affine Arithmetik mit der verbesserten Multiplikation aus [56]. Als Maß wird die relative Überschätzung RA verwendet:

$$RA = \frac{w_a}{w_c}. \quad (2.59)$$

Der Term w_c bezeichnet die Intervallbreite, die von den verschiedenen Gebietsarithmetiken berechnet wurden. Der Term w_a bezeichnet die tatsächliche Intervallbreite. Da diese

nicht direkt berechnet werden kann, wird als Näherung die Intervallbreite der Intervallarithmetik verwendet. Ist $RA < 1$ bedeutet dies also eine Verschlechterung gegenüber der Intervallarithmetik. Die Ergebnisse aus [100] werden in Tabelle 2.1 wiedergegeben. Ein Vergleich der affinwertigen Multiplikation aus [21] (AATRE), einer Multiplikation mit Tschebyscheff-Approximation (AACHA) und einem dort neu eingeführten Verfahren AASEE ist in [99] enthalten. Die Ergebnisse sind in Tabelle 2.2 zusammengefasst. Aus den Ergebnissen in Tabellen 2.1 und 2.2 wird deutlich, dass die Erweiterungen die Überschätzung verringern können. In der Regel wird dazu im Vergleich zu den Grundtypen jedoch mehr Laufzeit benötigt. Die Erweiterungen wurden so entwickelt, dass sich insbesondere die gebietsarithmetische Multiplikation mit möglichst geringer Überschätzung berechnen lässt. Die Approximation anderer arithmetischer Funktionen wird lediglich in [82] beschrieben. Inwieweit die übrigen Erweiterungen Vorteile bei der Approximation anderer arithmetischer Funktionen bieten, ist unklar, da die Erweiterungen für die Darstellung von Multiplikationen optimiert wurden. Ebenso sind in der Literatur keine Lösungsalgorithmen bekannt, die die Vorteile der Erweiterungen nutzen können. Im Folgenden werden daher ausschließlich die Grundtypen Intervall- und affine Arithmetik weiter betrachtet.

Tabelle 2.1: Überschätzung für Multiplikationen in Gebietsarithmetiken AA, EAA, RAA, GIA (aus [100])

Beispiel	AA	EAA	RAA	GIA
$(5 + \varepsilon_1 - 3\varepsilon_3) \cdot (10 - \varepsilon_1 + 2\varepsilon_2)$	0,97	0,97	0,97	1,22
$(4.5 + 0.5\varepsilon_1) \cdot (10 + 2\varepsilon_2)$	0,93	0,93	0,93	1,00
OTA ([100, Figure 2(a)])	1,52	1,52	2,61	3,20 ^a

^a mit nach [100] modifizierter GIA

Tabelle 2.2: Überschätzung und Laufzeitkomplexität der Multiplikation in den Gebietsarithmetiken AATRE, AACHA, AASEE (aus [99]). Die Variablen n_1 und n_2 bezeichnen die Anzahl der Abweichungssymbole der Faktoren der Multiplikation

	AATRE	AACHA	AASEE
RA für Case 1 ^a	1,33	1,15	1,03
Laufzeitkomplexität	$\mathcal{O}(\max(n_1, n_2))$	$\mathcal{O}((n_1 + n_2) \log(n_1 + n_2))$	$\mathcal{O}(\max(n_1, n_2))$

^a $((-0,0550\hat{x} + 0,2168)\hat{x} - 0,4645)\hat{x} + 0,9956)\hat{x} + 0,0001$ mit $\hat{x} = 0,5 + 0,5\varepsilon_1$, als Bezugsgröße w_a für RA wird hier das minimale Intervall verwendet.

2.2 Gebietsarithmetische Lösungsverfahren

Im Folgenden werden verschiedene gebietsarithmetische Lösungsverfahren aus der Literatur an Hand des damit zu lösenden Gleichungssystems gegenübergestellt.

2.2.1 Lineare Gleichungssysteme

Aus der Punktarithmetik ist das Gaußverfahren zur Lösung linearer Gleichungssysteme (2.1) und (2.4) bekannt. Mit einer Permutationsmatrix \mathbf{P} wird eine Zerlegung in eine linke untere Dreiecksmatrix \mathbf{L} und eine rechte obere Dreiecksmatrix \mathbf{R} bestimmt:

$$\mathbf{PA} = \mathbf{LR}. \quad (2.60)$$

Durch Vorwärts- und Rückwärtssubstitution können dann die Gleichungssysteme

$$\mathbf{L}\vec{y} = \mathbf{P}\vec{b} \quad (2.61)$$

$$\mathbf{R}\vec{x} = \vec{y} \quad (2.62)$$

gelöst werden. Durch Ersetzen der punktarithmetischen durch intervallararithmetische Operationen kann das Verfahren auch auf Intervalle angewendet werden. Diese Möglichkeit wurde bereits in [63] beschrieben. Durch das wiederholte Dividieren durch die Zeilen und die damit in der Intervallarithmetik verbundene Überschätzung ist dieses Verfahren nur begrenzt einsetzbar. Insbesondere darf der Divisor nicht die Null enthalten. Diese Voraussetzung wird jedoch durch die Überschätzung der vorangegangenen Rechenschritte häufig verletzt. Durch eine Vorkonditionierung der Matrix \mathbf{A} lassen sich die Konvergenzeigenschaften des Verfahrens verbessern [70, Kapitel 4.5]. Häufig wird zur Vorkonditionierung die Inverse der Mittelpunktmatrix $\text{mid}(\mathbf{A})$ eingesetzt, aber auch andere Vorkonditionierungsmatrizen sind möglich [43, Kapitel 3]. Eine Erweiterung um eine Kombination aus Intervall- und affiner Arithmetik wird in [1] vorgestellt. Auch verbesserte Verfahren wie das Gauß-Seidel-Verfahren können gebietsarithmetisch genutzt werden. Die erste Veröffentlichung einer Intervallversion des Gauß-Seidel-Algorithmus erfolgte in [37]. Mit der Vorkonditionierungsmatrix \mathbf{Y} und den Abkürzungen $\mathbf{G} = \mathbf{YA}$ und $\vec{c} = \mathbf{Y}\vec{b}$ kann Gleichung (2.4) als

$$\mathbf{G}\vec{x} = \vec{c} \quad (2.63)$$

geschrieben werden. Damit ergibt sich die intervallararithmetische Iterationsvorschrift komponentenweise zu

$$[\vec{x}]_i^{n+1} = [\vec{x}]_i^n \cap \frac{1}{[\mathbf{G}]_{i,i}} \left\{ [\vec{c}]_i - \sum_{j=1}^{i-1} [\mathbf{G}]_{i,j} [\vec{x}]_j^{k+1} - \sum_{j=i+1}^n [\mathbf{G}]_{i,j} [\vec{x}]_j^n \right\}. \quad (2.64)$$

In [42] werden Optimalitätskriterien zur Wahl der Vorkonditionierungsmatrix beschrieben, häufig wird jedoch auch hier die Inverse der Mittelpunktmatrix verwendet. Neuere Arbeiten zur Lösung von parametrischen linearen Systemen (2.5) verwenden andere

Ansätze. So verwendet [93] einen genetischen Algorithmus, um eine innere Näherung für das einschließende Intervall zu bestimmen. Eine äußere Näherung kann mit dem direkten Verfahren aus [90] bestimmt werden. Beide Ansätze setzen allerdings eine lineare Abhängigkeit von \mathbf{A} und \vec{b} von den Parametern \vec{p} voraus. Mit der Erweiterung aus [92] werden auch allgemeinere Abhängigkeiten zugelassen. Ein vergleichbares Verfahren basierend auf verallgemeinerter Intervallarithmetik wird in [73] beschrieben.

2.2.2 Nichtlineare Gleichungssysteme

Ein Verfahren zur Bestimmung von Nullstellen impliziter nichtlinearer Funktionen (2.6) ist das Bisektions-Verfahren. Dabei wird ein Intervall rekursiv verkleinert, wenn eine Nullstelle in diesem Intervall existiert. Die Verkleinerung des Intervalls geschieht durch Teilen in zwei Teilintervalle. Bereits in der punktarithmetischen Darstellung des Verfahrens werden Intervalle verwendet, um Nullstellen von Funktionen einzuschließen. Dabei wird allerdings nur eine Nullstelle innerhalb des Suchintervalls gefunden und die Rundungsfehler der Gleitkommadarstellung werden nicht berücksichtigt. Diese Nachteile können durch die Verwendung von Gebietsarithmetiken vermieden werden. Die Verwendung von affiner Arithmetik bietet neben einem engeren Einschluss den Vorteil, dass zusätzliche Informationen zur Bestimmung des Punktes, an dem das Intervall geteilt wird, aus der affinen Form gewonnen werden können. Eine eindimensionale Darstellung mit Intervallarithmetik findet sich in [21, Abschnitt 4.1.2] und mit affiner Arithmetik in [21, Abschnitt 4.1.3]. Eine Erweiterung auf mehrdimensionale Systeme und Kombinationen mit anderen Verfahren wie der Newton-Iteration sowie Monotonie- und Konvexitätstests werden in [34, 35, 37, 41, 96] vorgestellt.

Ein weiteres Lösungsverfahren ist das Newton-Verfahren. Es handelt sich um ein iteratives Verfahren nach folgender Iterationsvorschrift:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad \text{eindimensional} \quad (2.65)$$

$$\vec{x}_{n+1} = \vec{x}_n - \mathbf{J}^{-1} f(\vec{x}_n) \quad \text{mehrdimensional.} \quad (2.66)$$

Erste intervallarithmetische Newton-Ansätze ersetzen punktarithmetische durch intervallarithmetische Operationen. Dabei ist eine intervallarithmetische Inversion der Matrix $[\mathbf{J}]$ durchzuführen [63]. Durch die hohe Anzahl an benötigten Divisionen ist dies mit einer starken Überschätzung verbunden. Eine eindimensionale Darstellung der Newtoniteration mit affiner Arithmetik ist in [21] zu finden. [54] erweitert diese auf mehrdimensionale Systeme. Die affine Darstellung besitzt den Vorteil, dass die Ableitung nicht explizit berechnet werden muss, sondern sich aus der Berechnung der affinen Erweiterungen ergibt.

Eine Verbesserung der intervallarithmetischen Version ist die Variante nach Krawczyk [52]. Die lineare Gleichung (2.4) kann auf beiden Seiten mit der Matrix \mathbf{Y} multipliziert werden. \mathbf{Y} ist eine beliebige reellwertige Matrix. In der Regel wird die Inverse von $\mathbf{J}(\vec{x}_{\text{approx}})$ mit $\vec{x}_{\text{approx}} \in [\vec{x}]$ benutzt, meist mit $\vec{x}_{\text{approx}} = \text{mid}([\vec{x}])$. Mit der Abkürzung

$[\mathbf{E}] = \mathbf{I} - \mathbf{Y}[\mathbf{J}]$ wird die Iterationsvorschrift

$$[\vec{x}]^{n+1} = \mathbf{Y}b + [\mathbf{E}][\vec{x}]^n \cap [\vec{x}]^n \quad (2.67)$$

definiert. Damit wird die gebietsarithmetische Matrixinversion vermieden. Ein Startgebiet $[\vec{x}]^0$ wird durch die Bildung des Schnitts sukzessive verkleinert. Als Startgebiet wird daher ein Gebiet benötigt, das die Lösung sicher einschließt.

Eine weitere Variante stammt von Hansen [36]. Diese verwendet eine Darstellung der Jacobimatrix, bei der nur einige Komponenten aus Intervallen und die übrigen aus reellen Zahlen bestehen. Dadurch kann ein engerer Einschluss erreicht werden.

In vielen Anwendungen von gebietsarithmetischen Lösungsverfahren sollen alle möglichen Nullstellen einer nichtlinearen, impliziten Gleichung (2.6) gefunden werden. In [5, 6] werden dazu Branch-and-Bound-Verfahren, affine Arithmetik und Linear Programming kombiniert. Eine Erweiterung auf parametrische nichtlineare Systeme (2.7) wird in [97, 98] vorgestellt. Andere Lösungsverfahren setzen explizite DGLs (2.9) voraus [45, 51, 81].

Ein weiteres Lösungsverfahren zur Lösung nichtlinearer parametrischer Systeme ist das EPD-Verfahren aus [27]. Es basiert auf affiner Arithmetik. Zunächst werden die Parameterunsicherheiten vernachlässigt und die Nominallösung des nichtlinearen Gleichungssystems mit dem Newton-Raphson-Verfahren bestimmt. Aus einer Linearisierung des Gleichungssystems werden dann die Anteile der Lösung bestimmt, die mit den gegebenen Parameterunsicherheiten korrespondieren. Diese werden im Folgenden Parameter Partial Deviations (PPDs) genannt. Aufgrund der Linearisierung ist der Einschluss der Lösung nicht sichergestellt. Daher werden im nächsten Schritt EPDs hinzugefügt. Diese werden initial abgeschätzt und mit Hilfe eines Skalierungsfaktors iterativ verbessert. Dabei kann das Gebiet sowohl verkleinert, als auch vergrößert werden. In Abbildung 2.10 ist ein Ablaufdiagramm des EPD-Verfahrens dargestellt, eine detailliertere Beschreibung des Verfahrens folgt in Abschnitt 2.4.

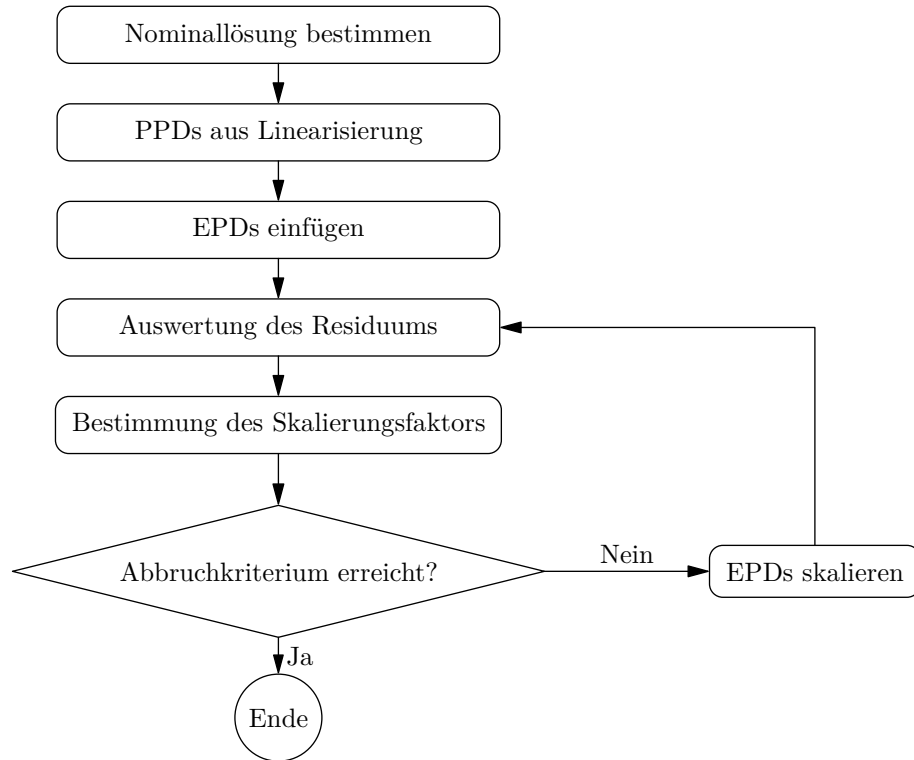


Abbildung 2.10: Ablaufdiagramm des EPD-Verfahrens

2.2.3 Anwendungen

Gebietsarithmetiken werden in verschiedenen Anwendungsgebieten eingesetzt. Zahlreiche Beispiele existieren zum Beispiel aus dem Bereich der Berechnung von Tragwerken [55, 66, 67, 71, 74, 90, 93, 94]. Die mit Unsicherheiten behafteten Parameter sind die Festigkeiten und Querschnittsflächen der Balken sowie die angreifenden Kräfte und Momente. Die dabei zugrunde liegenden Gleichungssysteme (2.4) sind linear. Andere Anwendungen stammen aus dem Bereich der Chemie. Dabei werden alle Nullstellen einer impliziten, nichtlinearen Gleichung (2.6) gesucht und durch Intervalle [5, 6] bzw. eine Kombination aus affiner und Intervallarithmetik [95] eingeschlossen. Die verwendeten Gleichungen sind dabei jedoch nicht parametrisch. Eine entsprechende Erweiterung wird in [97, 98] vorgestellt. Eine weitere Anwendung ist die Bestimmung der benötigten Bitbreiten in digitalen Systemen. Dafür ist es nicht nötig, ein implizites Gleichungssystem zu lösen, da der Signalfluss von Eingang zu Ausgang durch ein explizites Gleichungssystem (2.3) darstellbar ist. Außerdem sind die verwendeten Operationen auf Multiplikationen und Additionen beschränkt [20, 53, 75, 99, 107]. Eine Anwendung der affinen Arithmetik zum Entwurf von Verpackungen wird in [111] dargestellt. Das betrachtete Gleichungssystem ist dabei nichtlinear und explizit (Gleichung (2.3)), enthält aber im Gegensatz zu der vorgenannten Anwendung zusätzlich auch Divisionen und trigonometrische Funktionen. Auch aus dem Bereich der Schaltungssimulation stammen einige Anwendungen. Dabei

werden häufig lineare Systeme betrachtet, siehe z.B. [14, 16, 17, 48, 100–102]. Nichtlineare Probleme werden nur in wenigen Veröffentlichungen betrachtet, z.B. [15, 48, jeweils Kapitel 6] sowie [14, 54, 104]. In den vorgenannten Veröffentlichungen werden ausschließlich intervallararithmetische Verfahren behandelt. Eine Verwendung von affiner Arithmetik in Verbindung mit nichtlinearen Schaltungen wird nur durch das EPD-Verfahren aus [27–29] geleistet. Auch die Arbeiten zur Systemsimulation benutzen das EPD-Verfahren [30, 76–78]. Andere Anwendungen aus der Elektrotechnik befassen sich mit der Auslegung von Stromnetzen [31, 68, 69, 105, 106] oder der Auslegung von elektrischen Maschinen [23–25, 83]. Die Gleichungssysteme sind dabei nichtlinear explizit [31, 68, 83, 105, 106] bzw. implizit [69]. Weitere Anwendungen stammen aus dem Bereich der Abwasseraufbereitung. Die dort zugrunde liegenden DGLs sind nichtlinear und explizit (Gleichung (2.9)) [45, 51, 80, 81]. Die Entwickler der affinen Arithmetik Stolfi und De Figureido arbeiten im Bereich der Computergrafik. Dementsprechend gibt es viele Anwendungen in diesem Bereich. Die Gleichungssysteme dabei sind nichtlinear und implizit, jedoch nicht parametrisch (Gleichung (2.7)) [10, 11, 22, 46]. Zusammenfassungen weiterer Anwendungen finden sich in [40, 65].

2.2.4 Eignung zur Schaltungssimulation

Im Folgenden wird die Eignung der vorgestellten Lösungsverfahren für die Schaltungssimulation untersucht. Dazu werden zunächst die verschiedenen üblichen Simulationsarten vorgestellt.

2.2.4.1 Simulationsarten

Mit Hilfe von verschiedenen Simulationsarten werden Spannungen und Ströme in Abhängigkeit eines Schrittparameters v bestimmt. Sie unterscheiden sich in der Art der beschreibenden Gleichungssysteme und in der physikalischen Einheit des Schrittparameters v .

Gleichstromanalyse (DC) Bei der Gleichstromanalyse (DC) wird nur ein Simulationsschritt ausgeführt, daher wird keine Funktion $u(v)$ bzw. $i(v)$ verwendet. Sie bildet die Grundlage für die weiteren Simulationsarten. Das zugrunde liegende Gleichungssystem ist nichtlinear und nicht zeitabhängig, da angenommen wird, dass alle Ausgleichsvorgänge abgeschlossen sind.

Wechselstromanalyse (AC) Bei der Wechselstromanalyse (AC) wird das Gleichungssystem im DC-Arbeitspunkt (der mit der Gleichstromanalyse bestimmt wurde) linearisiert. Das Gleichungssystem wird für verschiedene Frequenzen gelöst. Der Simulationsschritt v ist eine Frequenz f .

DC-Sweep Beim DC-Sweep werden mehrere Gleichstromanalysen durchgeführt. Sie unterscheiden sich durch die Spannung einer (oder mehrerer) Spannungsquellen. Der

Simulationsschritt v ist hierbei eine Spannung u . Das zugrunde liegende Gleichungssystem ist nichtlinear und nicht zeitabhängig.

Transientenanalyse (TR) Bei der Transientenanalyse werden die Spannungen und Ströme in Abhängigkeit von der Zeit bestimmt, der Simulationsschritt v ist die simulierte Zeit t . Das Gleichungssystem ist nichtlinear, zeitabhängig und implizit (2.6). Zur Bestimmung des Anfangswerts bei $t = 0$ s wird eine Gleichstromanalyse durchgeführt. Im Gegensatz zu den anderen Simulationsarten sind hier die Simulationsschritte über die zeitlichen Ableitungen $\frac{du}{dt}$ bzw. $\frac{di}{dt}$ voneinander abhängig.

2.2.4.2 Eignung der gebietsarithmetischen Lösungsverfahren zur Schaltungssimulation

Die am häufigsten verwendete Simulationsart ist die Transientenanalyse. Sind die Schaltungsparameter mit Unsicherheiten behaftet, ist das System parametrisch, implizit und nichtlinear (2.7). Daher sind alle Lösungsverfahren für lineare Systeme nicht nutzbar. Auch Verfahren, die eine explizite Darstellung des Gleichungssystems benötigen, sind nicht einsetzbar, da die implizite im Allgemeinen nicht in eine explizite Darstellung überführt werden kann. Intervall-Newton oder -Krawczyk-Verfahren sind nicht gut geeignet, da sie eine kontrahierende Annäherung an das Lösungsgebiet darstellen. Sie benötigen ein überschätzendes Startgebiet zur Konvergenz. Dies ist jedoch bei der Schaltungssimulation nicht bekannt oder kann nur pessimistisch abgeschätzt werden. Eine zu pessimistische Abschätzung führt durch die Überschätzung jedoch zu einer Divergenz des Verfahrens. Das EPD-Verfahren erlaubt dagegen sowohl eine Kontraktion als auch eine Inflation des initialen Lösungsgebiets. Außerdem erhält es die linearen Einflüsse der Parameterunsicherheiten auf das Lösungsgebiet. Dadurch ist eine Verfolgung der Ursachen möglich. Außerdem soll zur Verbesserung des Einschlusses affine Arithmetik anstatt von Intervallarithmetik verwendet werden. Daher wird im Folgenden nur noch das EPD-Verfahren betrachtet.

2.3 Gleichungslösung als Optimierungsproblem

Im Allgemeinen sind die zu lösenden Schaltungsgleichungen in impliziter Form (2.6) gegeben. Existieren zusätzliche Parameter, ist das Gleichungssystem parametrisch (Gleichung (2.7)). Bei der Lösung einer solchen Gleichung im punktarithmetischen Fall wird \vec{x} so bestimmt, dass bei der Auswertung der Gleichung (2.7) das Residuum r vernachlässigbar klein wird:

$$\vec{r} = f_{\text{NL}}(\vec{x}, \vec{p}) < \vec{\delta}. \quad (2.68)$$

Auch in der affinen Arithmetik wird das Residuum \vec{r} für das Abbruchkriterium verwendet:

$$\vec{r} = f_{\text{NL}}(\vec{\hat{x}}, \vec{\hat{p}}). \quad (2.69)$$

Hier werden jedoch zusätzliche Bedingungen benötigt, um sicherzustellen, dass alle Lösungen eingeschlossen werden, die durch die mit Unsicherheiten behafteten Parameter

entstehen. Wird für die Elemente $\vec{x}_{(i)}$ des Vektors \vec{x} jeweils die Form

$$\vec{x}_{(i)} = \vec{x}_{(i)0} + \vec{x}_{(i)PPD}\varepsilon_{PPD} + \vec{x}_{(i)EPD}\varepsilon_{EPD} \quad (2.70)$$

angenommen, ergibt sich das Residuum in entsprechender Form:

$$\vec{r}_{(i)} = \vec{r}_{(i)0} + \vec{r}_{(i)PPD}\varepsilon_{PPD} + \vec{r}_{(i)EPD}\varepsilon_{EPD} + \vec{r}_{(i)NLPD}\varepsilon_{NLPD} \quad (2.71)$$

Die EPD-Symbole bezeichnen ein zusätzliches Abweichungssymbol, das eingeführt wird, um den Einschluss sicherzustellen. Um den Einschluss sicherzustellen muss gelten:

$$\exists \varepsilon_{EPD} \in [-1, 1] : \vec{r}_{(i)} = \vec{r}_{(i)0} + \vec{r}_{(i)PPD}\varepsilon_{PPD} + \vec{r}_{(i)EPD}\varepsilon_{EPD} + \vec{r}_{(i)NLPD}\varepsilon_{NLPD} < \vec{\delta} \quad (2.72)$$

mit

$$\varepsilon_{PPD}, \varepsilon_{NLPD} \in [-1, 1]. \quad (2.73)$$

Das zusätzlich eingefügte Abweichungssymbol x_{EPD} soll dabei möglichst klein sein, um die Überschätzung zu minimieren. In der üblichen Schreibweise für Optimierungsprobleme ergibt sich:

minimize x_{EPD}
subject to

$$\begin{aligned} \exists \varepsilon_{EPD} : \vec{r}_{(i)} &= \vec{r}_{(i)0} + \vec{r}_{(i)PPD}\varepsilon_{PPD} + \vec{r}_{(i)EPD}\varepsilon_{EPD} + \vec{r}_{(i)NLPD}\varepsilon_{NLPD} < \vec{\delta} \\ \varepsilon_{PPD}, \varepsilon_{EPD}, \varepsilon_{NLPD} &\in [-1, 1]. \end{aligned} \quad (2.74)$$

Die Berechnung des Residuums \vec{r} ist selbst wiederum ein Optimierungsproblem. Das Optimierungsziel ist vom gewählten Approximationsverfahren abhängig. Bei der Minimalbereichserweiterung wird der Radius des Residuums minimiert (vgl. [27, Gleichung 3.25]). Dagegen wird beim Tschebyscheffverfahren der Approximationsfehler γ minimiert (vgl. [27, Gleichung 3.29]).

2.3.1 Lösung des Optimierungsproblems mit einem Brute-Force-Ansatz

Das Optimierungsproblem (2.74) lässt sich unter anderem durch Ausprobieren lösen [86]. Der verwendete Algorithmus ist hier als Algorithmus 1 dargestellt. Dabei werden die Bestandteile x_0 , x_{PPD} und x_{EPD} der zu bestimmenden Lösung \hat{x} zufällig gewählt (Zeilen 7 bis 10) und überprüft, ob Bedingung (2.72) erfüllt ist (Zeile 12). Ist sie nicht erfüllt, wird die nächste Stichprobe für \hat{x} gezogen. Andernfalls wird überprüft, ob das EPD-Symbol von \hat{x} kleiner als das EPD-Symbol der bisher besten Lösung \hat{x}_{best} ist. Wenn ja, wird \hat{x}_{best} aktualisiert (Zeile 14f.). In Abbildung 2.11 sind für das Beispiel

$$f(\vec{x}, \vec{p}) = e^{\hat{x}} - \hat{p} \quad \text{mit} \quad \hat{p} = 2 + 0.7\varepsilon_1 \quad (2.75)$$

alle Stichproben, die die Bedingung (2.72) erfüllen, zusammen mit der gefundenen Lösung \hat{x}_{best} dargestellt.

Algorithmus 1 Brute-Force-Ansatz zur Lösung des Optimierungsproblems (2.74)

```

1: procedure SOLVEBRUTEFORCE( $f(\hat{x}, \hat{p}), \hat{p}, n_{\text{samples}}$ )
2:    $x_{\text{best},0} = \text{REAL\_MAX}$ 
3:    $x_{\text{best,PPD}} = \text{REAL\_MAX}$ 
4:    $x_{\text{best,EPD}} = \text{REAL\_MAX}$ 
5:    $\hat{x}_{\text{best}} = x_{\text{best},0} + x_{\text{best,PPD}}\varepsilon_{\text{PPD}} + x_{\text{best,EPD}}\varepsilon_{\text{EPD}}$ 
6:   for all  $n_{\text{samples}}$  do
7:      $x_0 = \text{RANDOM}()$ 
8:      $x_{\text{PPD}} = \text{RANDOM}()$ 
9:      $x_{\text{EPD}} = \text{RANDOM}()$ 
10:     $\hat{x} = x_0 + x_{\text{PPD}}\varepsilon_{\text{PPD}} + x_{\text{EPD}}\varepsilon_{\text{EPD}}$ 
11:     $\hat{r} = f(\hat{x}, \hat{p})$ 
12:    if  $(r_0 + r_{\text{PPD}} + r_{\text{NLPD}}) > r_{\text{EPD}} \vee (r_0 - r_{\text{PPD}} - r_{\text{NLPD}}) > -r_{\text{EPD}}$  then
13:      continue
14:    else if  $\hat{x}_{\text{EPD}} < \hat{x}_{\text{best,EPD}}$  then
15:       $\hat{x}_{\text{best}} = \hat{x}$ 
16:    end if
17:  end for
18:  return  $\hat{x}_{\text{best}}$ 
19: end procedure

```

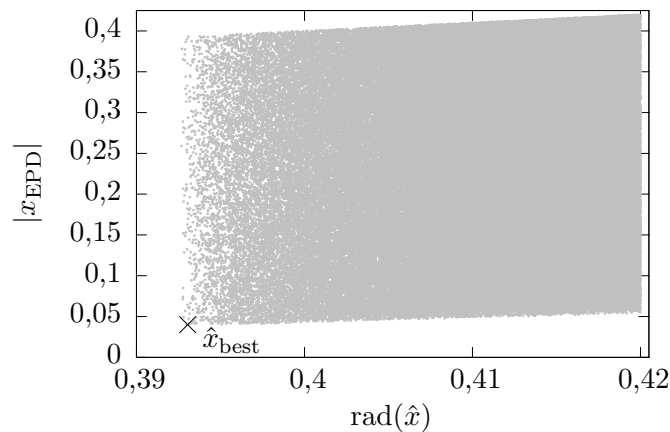


Abbildung 2.11: Darstellung von Stichproben, die Bedingung (2.72) erfüllen, zusammen mit der von Algorithmus 1 gefundenen Lösung x_{best} für das Beispiel $f(\vec{\hat{x}}, \vec{\hat{p}}) = e^{\hat{x}} - \hat{p}$ mit $\hat{p} = 2 + 0.7\varepsilon_1$ und Tschebyscheff-Approximation

2.4 Das EPD-Verfahren

Ein Algorithmus, der das Optimierungsproblem (2.74) auf effizientere Weise löst, ist das in [27] vorgestellte EPD-Verfahren. Der Grundgedanke besteht darin, die Nominallösung \vec{x}_0 als Zentralwert der affinen Form zu bestimmen und die Form anschließend um Abweichungssymbole zu erweitern. Die Parameterunsicherheiten \vec{x}_{PPD} werden aus der linearisierten Parameterabhängigkeit bestimmt. Da durch diese Linearisierung der Einschluss der Lösung nicht mehr sichergestellt werden kann, werden zusätzliche EPD-Symbole \vec{x}_{EPD} eingefügt. Ihre Größe wird durch ein iteratives Verfahren so lange angepasst, bis der Einschluss sichergestellt ist. Das EPD-Verfahren ist in Algorithmus 2 dargestellt. Die Nominallösung wird in Zeile 3 bestimmt. Dazu werden die affinen Parameter \vec{p} durch ihre Nominalwerte \vec{p}_0 ersetzt und mit Hilfe des Newton-Raphson-Verfahrens die Nominallösung \vec{x}_0 berechnet. Diese wird als Zentralwert der affinen Lösung $\vec{\hat{x}}$ verwendet (Zeile 4). Als erste Näherung für die Abweichungssymbole \vec{x}_{PPD} wird das linearisierte Gleichungssystem

$$f(\vec{x}, \vec{p}) \approx \left. \frac{df}{d\vec{x}} \right|_{\vec{x}_0, \vec{p}_0} \cdot \vec{x}_{\text{PPD}} + \left. \frac{df}{d\vec{p}} \right|_{\vec{x}_0, \vec{p}_0} \cdot \vec{p}_{\text{PPD}} = 0 \quad (2.76)$$

verwendet (Zeile 5). Mit den Abweichungssymbolen x_{PPD} wird die bestehende Lösung \hat{x} um Abweichungssymbole erweitert (Zeile 6). Durch den Linearisierungsfehler ist der Einschluss der Lösung nicht sichergestellt. Daher wird die Lösung um einen zusätzlichen Abweichungsterm $x_{\text{EPD}}\varepsilon_{\text{EPD}}$ erweitert. Seine Größe lässt sich nur iterativ bestimmen, da eine Veränderung von x_{EPD} eine (im Allgemeinen) nichtlineare Änderung an \hat{r} hervorruft, die nicht explizit beschrieben werden kann. Als Startwert für x_{EPD} wird nach [27] 10% des jeweiligen Radius angenommen (Zeile 7). Die bestehende Lösung \hat{x} wird um diese Abweichungssymbole erweitert (Zeile 8). Die Größe der EPD-Symbole wird iterativ angepasst (Zeilen 10 bis 30). Dazu wird mit der aktuellen Schätzung von \hat{x} das affine Residuum $\vec{\hat{r}}$ berechnet (Zeile 13). Daraus wird mit Hilfe der Funktion BuildEPDMatrix die EPD-Matrix \mathbf{E} in folgender Form aufgebaut (Zeile 14):

$$\mathbf{E} = \begin{bmatrix} \vec{\hat{r}}_{(0)\text{EPD}, 0} & \cdots & \vec{\hat{r}}_{(0)\text{EPD}, n_x} \\ & \ddots & \\ \vec{\hat{r}}_{(n_x)\text{EPD}, 0} & \cdots & \vec{\hat{r}}_{(n_x)\text{EPD}, n_x} \end{bmatrix}. \quad (2.77)$$

Gleichzeitig entfernt die Funktion die EPD-Symbole aus dem Residuum $\vec{\hat{r}}$. Mit der EPD-Matrix \mathbf{E} wird der affinwertige Skalierungsfaktor $\vec{\hat{v}}$ bestimmt (Zeilen 15 bis 20). Die Operationen lassen sich auch als Lösung der Gleichung

$$\mathbf{E} \cdot \vec{\hat{v}} = \vec{\hat{r}} \quad (2.78)$$

interpretieren. Mit dem affinwertigen Skalierungsfaktor $\vec{\hat{v}}$ wird dann der nominale Skalierungsfaktor \vec{s} bestimmt (Zeilen 21 bis 27). Nähere Informationen zum Skalierungsfaktor folgen in Abschnitt 2.4.1. Die eingefügten partiellen Abweichungen \vec{x}_{EPD} werden mit

2 Simulation von Parameterunsicherheiten

diesem Skalierungsfaktor skaliert (Zeile 28). Dies wird solange wiederholt, bis der Skalierungsfaktor $\vec{s}_{(i)}$ gegen eins geht und sich $\vec{x}_{(i)\text{EPD}}$ damit nicht mehr verändert oder $\vec{x}_{(i)\text{EPD}}$ verschwindet (Zeile 30). Damit ist sichergestellt, dass die erweiterte Lösung

$$\vec{\hat{x}} = \hat{x}_0 + \sum \hat{x}_{\text{PPD}} \varepsilon_{\text{PPD}} + \sum \hat{x}_{\text{EPD}} \varepsilon_{\text{EPD}} \quad (2.79)$$

das minimale Lösungsgebiet einschließt. Um den Algorithmus bei Nichtkonvergenz abzuberechnen, wird die Anzahl der Iterationen n_{iter} auf eine maximale Anzahl $n_{\text{iter,max}}$ begrenzt. Zusätzlich kann eine Optimierung der PPDs vorgenommen werden. Dies kann vor (Zeile 9) und während der affinen Lösungsiteration (Zeile 12) durchgeführt werden. Die Anzahl der Optimierungsiterationen kann dabei unterschiedlich sein ($n_{\text{opt,b}}$ bzw. $n_{\text{opt,d}}$). Die Methode `OptimizeInitialAffineSolution` ist in Algorithmus 3 dargestellt. Aus dem Residuum $\vec{\hat{r}}$ wird die PPD-Matrix \mathbf{P} aufgebaut (Zeile 4). Sie enthält in der Zeile i den Zentralwert und die PPDs aus dem jeweiligen Element $\vec{\hat{r}}_{(i)}$ des Residuums $\vec{\hat{r}}$:

$$\mathbf{P} = \begin{bmatrix} \vec{\hat{r}}_{(0)0} & \vec{\hat{r}}_{(0)\text{PPD},0} & \cdots & \vec{\hat{r}}_{(0)\text{PPD},n_p} \\ \vdots & & \ddots & \\ \vec{\hat{r}}_{(n_x)0} & \vec{\hat{r}}_{(n_x)\text{PPD},0} & \cdots & \vec{\hat{r}}_{(n_x)\text{PPD},n_p} \end{bmatrix}. \quad (2.80)$$

Aus dem linearisierten Gleichungssystem

$$\mathbf{J} \cdot \mathbf{C} = \mathbf{P} \quad (2.81)$$

wird eine Korrekturmatrix \mathbf{C} bestimmt (Zeile 5). Die Korrekturmatrix wird dann in der Methode `UpdateCentralValueAndPPDs` (Zeile 6) vom Zentralwert und den PPDs der Lösung $\vec{\hat{x}}$ subtrahiert. Diese Schritte werden n_{optimize} -mal ausgeführt. Diese Optimierung ermöglicht einen größeren Konvergenzbereich von Algorithmus 2.

Algorithmus 2 Affiner Lösungsalgorithmus nach [27]

```

1: procedure SOLVEEPD( $\vec{p}, f(\vec{x}, \vec{p})$ )
2:    $n_{\text{iter}} = 0$ 
3:    $\vec{x}_0 = \text{SOLVE}(f(\vec{x}_0, \vec{p}_0) = 0)$ 
4:    $\vec{\hat{x}} = \vec{x}_0$ 
5:    $\vec{x}_{\text{PPD}} = \left( \frac{df}{d\vec{p}} \Big|_{\vec{x}_0, \vec{p}_0} \right)^{-1} - \frac{df}{d\vec{p}} \Big|_{\vec{x}_0, \vec{p}_0} \cdot \vec{p}_{\text{PPD}}$ 
6:    $\vec{\hat{x}} = \vec{x}_0 + \vec{x}_{\text{PPD}} \varepsilon_{\text{PPD}}$ 
7:    $\vec{x}_{\text{EPD}} = 0,1 \cdot \text{rad}(\vec{\hat{x}})$ 
8:    $\vec{\hat{x}} = \vec{x}_0 + \vec{x}_{\text{PPD}} \cdot \varepsilon_{\text{PPD}} + \vec{x}_{\text{EPD}} \cdot \varepsilon_{\text{EPD}}$ 
9:    $\vec{\hat{x}} = \text{OPTIMIZEINITIALAFFINESOLUTION}(f, \vec{\hat{x}}, \vec{\hat{p}}, \mathbf{J}, n_{\text{opt},b})$ 
10:  repeat
11:     $n_{\text{iter}} = n_{\text{iter}} + 1$ 
12:     $\vec{\hat{x}} = \text{OPTIMIZEINITIALAFFINESOLUTION}(f, \vec{\hat{x}}, \vec{\hat{p}}, \mathbf{J}, n_{\text{opt},d})$ 
13:     $\vec{\hat{r}} = f(\vec{\hat{x}}, \vec{\hat{p}})$ 
14:     $\mathbf{E} = \text{BUILDEPDMATRIX}(\vec{\hat{r}})$ 
15:     $\vec{\hat{v}} = \vec{0}$ 
16:    for all row do
17:      for all col do
18:         $\vec{\hat{v}}_{(\text{row})} = \vec{\hat{v}}_{(\text{row})} + \mathbf{E}_{(\text{row},\text{col})}^{-1} \cdot \vec{\hat{r}}_{(\text{col})}$ 
19:      end for
20:    end for
21:    for all  $\vec{\hat{v}}_{(i)}$  do
22:      if  $\vec{\hat{v}}_{(i)0} > 0$  then
23:         $\vec{s}_{(i)} = \max(\vec{\hat{v}}_{(i)})$ 
24:      else
25:         $\vec{s}_{(i)} = -\min(\vec{\hat{v}}_{(i)})$ 
26:      end if
27:    end for
28:     $\vec{x}_{\text{EPD}} = \vec{s} \cdot \vec{x}_{\text{EPD}}$ 
29:     $\vec{\hat{x}} = \vec{x}_0 + \vec{x}_{\text{PPD}} \varepsilon_{\text{PPD}} + \vec{x}_{\text{EPD}} \varepsilon_{\text{EPD}}$ 
30:    until ( $|\vec{s}_{(i)}| \approx 1 \vee \vec{x}_{(i)\text{EPD}} < \epsilon \vee n_{\text{iter}} > n_{\text{iter},\text{max}}$ )
31:  return  $\vec{\hat{x}}$ 
32: end procedure

```

Algorithmus 3 Algorithmus zur Optimierung der initialen affinen Lösung

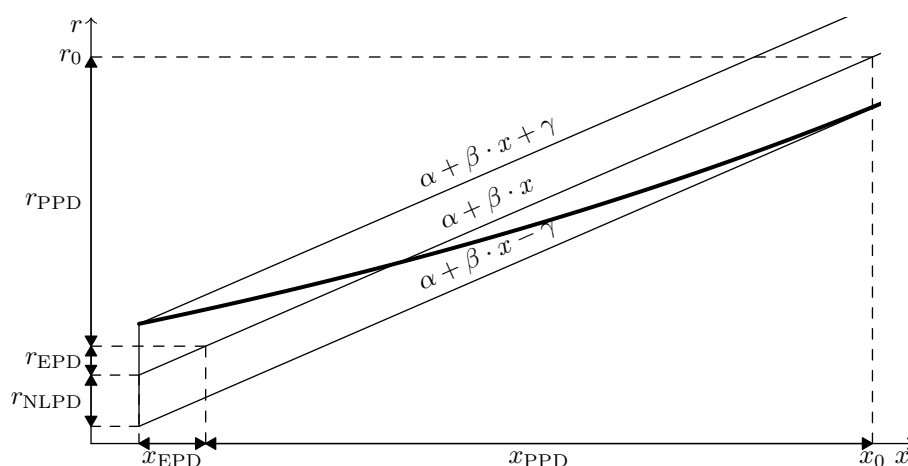
```

procedure OPTIMIZEINITIALAFFINESOLUTION( $f, \vec{\hat{x}}, \vec{\hat{p}}, \mathbf{J}, n_{\text{iterations}}$ )
  for  $0 \dots n_{\text{iterations}}$  do
     $\vec{r} = f(\vec{\hat{x}}, \vec{\hat{p}})$ 
     $\mathbf{P} = \text{BUILDPPDMATRIX}(\vec{r})$ 
     $\mathbf{C} = \mathbf{J}^{-1} \cdot \mathbf{P}$ 
     $\vec{\hat{x}} = \text{UPDATECENTRALVALUEANDPPDs}(\mathbf{C})$ 
  end for
  return  $\vec{\hat{x}}$ 
end procedure

```

2.4.1 Bedeutung des Skalierungsfaktors

Um die Bedeutung des Skalierungsfaktors zu verdeutlichen, ist in Abbildung 2.12 beispielhaft die Auswertung einer eindimensionalen affinen Funktion grafisch dargestellt. Bei der affinen Auswertung der Funktion, werden α , β und γ aus Gleichung (2.39) bzw. (2.40) bestimmt. Die Bestandteile x_0 und x_{PPD} der Eingangsgröße \hat{x} werden an der Geraden $\alpha + \beta \cdot x$ auf die Werte r_0 bzw. r_{PPD} abgebildet. Die beiden Geraden $\alpha + \beta \cdot x + \gamma$ bzw. $\alpha + \beta \cdot x - \gamma$ schließen den maximalen Linearisierungsfehler ein. Im Residuum entspricht dieser der zusätzlichen unkorrelierten, partiellen Abweichung r_{NLPD} . Die Rückabbildung auf ein entsprechendes x_{NLPD} existiert nicht, da die Umkehrfunktion von f nicht bekannt ist bzw. im allgemeinen Fall nicht existiert. Um den Einschluss der Lösung sicherzustellen, wird der Lösung \hat{x} der Abweichungsterm $x_{\text{EPD}} \varepsilon_{\text{EPD}}$ hinzugefügt. Dieser wird wieder über die Gerade $\alpha + \beta \cdot x$ auf r_{EPD} abgebildet. Gilt $|r_{\text{EPD}}| = |r_{\text{NLPD}}|$, dann existiert für jedes ε_{EPD} ein entsprechendes $\varepsilon_{\text{NLPD}}$, sodass sie sich gegenseitig kompensieren können und das Residuum zu null wird. Dieses Verfahren stellt den Einschluss sicher, verursacht jedoch eine Überschätzung. Dies wird im folgenden Beispiel deutlich.

**Abbildung 2.12:** Grafische Darstellung der Auswertung einer affinen Funktion

2.4.2 Beispiel: Lösen einer Gleichung mit dem EPD-Verfahren

Um die Funktionsweise des EPD-Verfahrens gemäß Algorithmus 2 zu verdeutlichen, wird im Folgenden ein eindimensionales Beispiel betrachtet. Vor und während der affinen Lösungsiteration wird jeweils ein Optimierungsschritt ausgeführt, d.h. $n_{\text{opt,b}} = n_{\text{opt,d}} = 1$. Die zu lösende Gleichung ist dabei wiederum die implizite Funktion

$$f(\hat{x}, \hat{p}) = e^{\hat{x}} - \hat{p} = 0 \quad (2.82)$$

mit $\hat{p} = 2 + 0,7\varepsilon_1$. Dieses Beispiel besitzt im Gegensatz zu

$$e^{\hat{x}-\hat{p}} - 1 = 0 \quad (2.83)$$

aus [27, Beispiel 4.6] einen nichtlinearen Zusammenhang zwischen dem Parameter \hat{p} und der Variablen \hat{x} . Daher verdeutlicht Gleichung (2.82) besser die Notwendigkeit EPD-Symbole einzufügen, um den Einschluss sicherzustellen. Es gilt:

$$\frac{df}{dx} = e^x \quad (2.84)$$

$$\frac{df}{dp} = -1. \quad (2.85)$$

Für Gleichung (2.82) existiert die explizite Darstellung:

$$\hat{x} = \ln(\hat{p}). \quad (2.86)$$

Dieser Zusammenhang ist monoton, daher lässt sich das minimale Intervall angeben:

$$x_{\min} = \ln(\min(\hat{p})) = \ln(2 - 0,7) \approx 0,2624 \quad (2.87)$$

$$x_{\max} = \ln(\max(\hat{p})) = \ln(2 + 0,7) \approx 0,9933 \quad (2.88)$$

$$[x_{\text{minimal}}] = [0,2624, 0,9933]. \quad (2.89)$$

Die Nominallösung von Gleichung (2.82) ist $x_0 = \ln(p_0) = \ln(2) \approx 0,6931$. Das PPD-Symbol der Lösung wird aus der Linearisierung (2.76) zu

$$x_{\text{PPD}} = - \left(\frac{df}{dx} \Big|_{x_0, p_0} \right)^{-1} \cdot \frac{df}{dp} \Big|_{x_0, p_0} \cdot p_{\text{PPD}} = \frac{1}{2} \cdot 1 \cdot 0,7 = 0,35 \quad (2.90)$$

bestimmt. Die aus diesen Werten zusammengesetzte affine Form bzw. das entsprechende Intervall

$$\hat{x} = x_0 + x_{\text{PPD}}\varepsilon_{\text{PPD}} = 0,6931 + 0,35\varepsilon_{\text{PPD}} \quad (2.91)$$

$$[x] = [0,3431, 1,0431] \quad (2.92)$$

schließt das minimale Intervall x_{minimal} nicht ein. Auch nach der Erweiterung der Lösung um das EPD-Symbol $x_{\text{EPD}} = 0,1 \cdot \text{rad}(\hat{x}) = 0,1 \cdot x_{\text{PPD}} = 0,035$ wird $[x]_{\text{minimal}}$ nicht eingeschlossen:

$$\hat{x} = x_0 + x_{\text{PPD}}\varepsilon_{\text{PPD}} + x_{\text{EPD}}\varepsilon_{\text{EPD}} = 0,6931 + 0,35\varepsilon_{\text{PPD}} + 0,035\varepsilon_{\text{EPD}} \quad (2.93)$$

$$[x] = [0,3081, 1,078]. \quad (2.94)$$

2 Simulation von Parameterunsicherheiten

Die Berechnung des Residuums mit der Tschebyscheff-Approximation ergibt:

$$\hat{r} = 0,0747 + 0,0174\varepsilon_{\text{PPD}} + 0,0717\varepsilon_{\text{EPD}} + 0,0753\varepsilon_{\text{NLPD}} \quad (2.95)$$

Durch die Optimierung verändert sich die Lösung \hat{x} zu:

$$\hat{x} = 0,6558 + 0,3413\varepsilon_{\text{PPD}} + 0,0350\varepsilon_{\text{EPD}} \quad (2.96)$$

$$[x] = [0,2795, 1,0321] \quad (2.97)$$

Dadurch verkleinert sich das Residuum

$$\hat{r} = -0,0046 + -0,0268\varepsilon_{\text{PPD}} + 0,0690\varepsilon_{\text{EPD}} + 0,0693\varepsilon_{\text{NLPD}}, \quad (2.98)$$

trotzdem wird $[x]_{\text{minimal}}$ nicht eingeschlossen. Die Bestimmung des nominalen Skalierungsfaktors mit der optimierten Lösung ergibt $s = 1,003\,526$. Damit wird das EPD-Symbol von $[x]$ gemäß $s \cdot x_{\text{EPD}}$ angepasst:

$$\hat{x} = 0,6558 + 0,3413\varepsilon_{\text{PPD}} + 0,0351\varepsilon_{\text{EPD}} \quad (2.99)$$

Die weiteren Schritte sind in Tabelle 2.3 gezeigt. Nach 11 Iterationen ist das Abbruchkriterium $s - 1 < \delta$ mit $\delta = 1 \cdot 10^{-6}$ erfüllt und die Berechnung wird abgebrochen. Die gefundene Lösung lautet:

$$\hat{x} = 0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}} \quad \text{bzw.} \quad (2.100)$$

$$[x] = [0,2623, 1,0476]. \quad (2.101)$$

Das entsprechende Residuum lautet:

$$\hat{r} = 0,0000 + 0,0000\varepsilon_{\text{PPD}} + 0,0755\varepsilon_{\text{EPD}} + 0,0755\varepsilon_{\text{NLPD}} \quad (2.102)$$

Das Intervall $[x]$ schließt das minimale Gebiet $[x]_{\text{minimal}}$ ein, überschätzt es jedoch.

Tabelle 2.3: Ergebnisse und Skalierungsfaktoren der verschiedene Iterationsschritte für das Lösen von Gleichung (2.82)

n_{iter}	\hat{x}	$[x]$	s
0	$0,6558 + 0,3413\varepsilon_{\text{PPD}} + 0,0350\varepsilon_{\text{EPD}}$	$[0,2795, 1,0321]$	1,003 53
1	$0,6581 + 0,3547\varepsilon_{\text{PPD}} + 0,0351\varepsilon_{\text{EPD}}$	$[0,2683, 1,0479]$	1,072 63
2	$0,6555 + 0,3535\varepsilon_{\text{PPD}} + 0,0377\varepsilon_{\text{EPD}}$	$[0,2644, 1,0467]$	1,006 77
3	$0,6553 + 0,3543\varepsilon_{\text{PPD}} + 0,0379\varepsilon_{\text{EPD}}$	$[0,2631, 1,0475]$	1,005 44
4	$0,6551 + 0,3543\varepsilon_{\text{PPD}} + 0,0381\varepsilon_{\text{EPD}}$	$[0,2626, 1,0476]$	1,001 29
5	$0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}}$	$[0,2624, 1,0476]$	1,000 56
6	$0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}}$	$[0,2624, 1,0476]$	1,000 17
7	$0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}}$	$[0,2624, 1,0477]$	1,000 06
8	$0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}}$	$[0,2624, 1,0477]$	1,000 02
9	$0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}}$	$[0,2624, 1,0477]$	1,000 01
10	$0,6550 + 0,3544\varepsilon_{\text{PPD}} + 0,0382\varepsilon_{\text{EPD}}$	$[0,2624, 1,0477]$	1,000 00

2.4.3 Transiente Simulationen

Für transiente Simulationen ist die Zeitschrittsteuerung sehr wichtig. Die Schrittweite h soll möglichst groß sein, um die Simulation möglichst schnell durchführen zu können, andererseits darf sie nicht zu groß werden, da andernfalls der Integrationsfehler unzulässig groß wird. Bekannte lineare Integrationsverfahren sind das explizite Eulerverfahren (engl. Forward Euler), das implizite Eulerverfahren (engl. Backward Euler), und das Trapezverfahren. Sie ersetzen die Ableitungen durch eine Näherung mit Hilfe von Differenzenquotienten. Im Folgenden wird, wenn nichts anderes genannt ist, das Trapezverfahren verwendet. Die Parameter des nichtlinearen Gleichungssystems (2.7) werden um die Lösung aus dem vergangenen Zeitschritt erweitert:

$$f(\vec{x}_n, \underbrace{\vec{p}, \vec{x}_{n-1}}_{\vec{p}_{\text{ext}}}) = 0. \quad (2.103)$$

Die eingefügten Variablen aus dem vorangegangenen Zeitschritt werden als Zustandsparameter (engl. state parameter) bezeichnet. Dieses Gleichungssystem kann mit Algorithmus 2 gelöst werden. Der Ablauf einer transienten Simulation ist in Algorithmus 4 dargestellt. Zunächst wird ein DC-Gleichungssystem f_{DC} durch Nullsetzen der Ableitungen in f erzeugt und mit dem EPD-Verfahren gelöst (Zeile 5f.) Für die weiteren Zeitschritte $t > 0$ s werden die Ableitungen entsprechend des verwendeten Integrationsverfahren ersetzt (Zeile 10) und für zeitabhängige Ausdrücke der aktuelle Wert von t eingesetzt (Zeile 11). Die Ableitungen werden dem Parametersatz hinzugefügt (Zeile 12) und die Gleichung mit Hilfe des EPD-Verfahrens gelöst (Zeile 13). Danach wird der gewählte Zeitschritt überprüft. War dieser zu groß, wird die aktuelle Lösung verworfen, eine neue, kleinere Schrittweite berechnet (Zeilen 14 bis 17) und mit dieser ein erneuter Lösungsversuch unternommen. Andernfalls wird überprüft, ob die Schrittweite vergrößert werden kann (Zeile 18f.). Ist dies nicht der Fall, wird die aktuelle Schrittweite beibehalten. Nähere

Informationen finden sich in [27, Abschnitt 5.2.3].

Algorithmus 4 Ablauf einer affinen Transientensimulation

```

1: procedure TRANSIENTSIMULATION( $f, \vec{p}$ )
2:    $n = 0$ 
3:    $t = 0$ 
4:    $h = h_{\text{init}}$ 
5:    $f_{\text{DC}} = \text{REMOVEDERIVATIVES}(f)$ 
6:    $\vec{x}_0 = \text{SOLVEEPD}(f_{\text{DC}}, \vec{p})$ 
7:   while  $t < t_{\text{end}}$  do
8:      $n = n + 1$ 
9:      $t = t + h$ 
10:     $f_{\text{RD}}(t) = \text{REPLACEDERIVATIVES}(f, \vec{x}_n, \vec{x}_{n-1}, h)$ 
11:     $f_{\text{RD}} = \text{REPLACETIME}(f_{\text{RD}}, t)$ 
12:     $\vec{p}_{\text{ext}} = \{\vec{p}, \vec{x}_{n-1}\}$ 
13:     $\vec{x}_n = \text{SOLVEAFFINE}(f_{\text{RD}}, \vec{p}_{\text{ext}})$ 
14:    if  $\text{CHECKTIMESTEP}(\vec{x}_n) == \text{tooLarge}$  then
15:       $n = n - 1$ 
16:       $t = t - h$ 
17:       $h = \text{DECREASESTEPWIDTH}()$ 
18:    else if  $\text{CHECKTIMESTEP}(\vec{x}_n) == \text{canBeEnlarged}$  then
19:       $h = \text{INCREASESTEPWIDTH}()$ 
20:    end if
21:  end while
22: end procedure

```

2.4.4 Komplexität des EPD-Verfahrens

Die Gesamtlaufzeit \mathcal{L} des EPD-Verfahrens lässt sich in die Laufzeiten für die zwei Schritte EPD-Skalierung \mathcal{L}_{EPD} (Algorithmus 2) und PPD-Optimierung $\mathcal{L}_{\text{Optimize}}$ (Algorithmus 3) unterteilen:

$$\mathcal{L} = \mathcal{L}_{\text{EPD}} + \mathcal{L}_{\text{Optimize}} \quad (2.104)$$

Beide bestehen aus dem Auswerten der affinwertigen Funktion $f(\vec{x}, \vec{p})$ und dem Lösen eines linearen Gleichungssystems mit affinwertiger rechter Seite. Da sich die Dimensionen des zu lösenden Gleichungssystem unterscheiden, werden sie getrennt betrachtet. Zur Bestimmung der Komplexität wird die Anzahl der Fließkommaoperationen nach Tabelle 2.4 zugrunde gelegt. Die Laufzeit der EPD-Skalierung wird bestimmt durch das Lösen von Gleichung (2.78) und das Auswerten des Gleichungssystems $f(\vec{x}, \vec{p})$. Die Zeit, die für das Ausführen der Funktionen BuildEPDMatrix und Solve benötigt wird, ist gemäß experimentellen Beobachtungen vernachlässigbar gering. Das Lösen des Gleichungssystems (2.78) mit affinwertiger rechter Seite \vec{r} kann unterschiedlich realisiert werden. Zum einen ist eine Realisierung als Inversion der EPD-Matrix \mathbf{E} mit anschließender

zeilenweiser Multiplikation mit dem affinwertigen Vektor $\vec{\hat{r}}$ möglich (wie in Algorithmus 2, Zeilen 15 bis 20 gezeigt). Eine weitere Möglichkeit ist die Realisierung als LR-Zerlegung mit mehreren rechten Seiten. Der affinwertige Vektor \hat{r} wird dabei als reellwertige Matrix \mathbf{R} der Form

$$\mathbf{R} = \begin{bmatrix} \vec{\hat{r}}_{(0)0} & \vec{\hat{r}}_{(0)\text{NLPD}, 0} & \cdots & \vec{\hat{r}}_{(0)\text{NLPD}, n_x \cdot n_{\text{NL}}} \\ \vdots & & \ddots & \\ \vec{\hat{r}}_{(n_x)0} & \vec{\hat{r}}_{(n_x)\text{NLPD}, 0} & \cdots & \vec{\hat{r}}_{(n_x)\text{NLPD}, n_x \cdot n_{\text{NL}}} \end{bmatrix} \quad (2.105)$$

dargestellt:

$$\mathbf{V} = \mathbf{E}^{-1} \cdot \mathbf{R} \quad (2.106)$$

Die Dimension von \mathbf{E} ist $\mathbb{R}^{n_x \times n_x}$, die von \mathbf{R} ist von der Anzahl der enthaltenen nichtlinearen Ausdrücke abhängig, die jeweils ein NLPD-Symbol generieren. Werden durchschnittlich n_{NL} nichtlineare Ausdrücke pro Gleichung angenommen, ergibt sich die Dimension zu $\mathbb{R}^{n_x \times (n_x \cdot n_{\text{NL}} + 1)}$, da jedes unabhängige NLPD-Symbol in einer separaten Spalte angeordnet werden muss. Für die Realisierung als LR-Zerlegung ergibt sich die benötigte Anzahl an Operationen \mathcal{L}_v für das Lösen von Gleichung (2.78) zu:

$$\mathcal{L}_v = n_{\text{iter}} \cdot \left(\frac{1}{3}(n_x^3 - n_x) + (n_x \cdot n_{\text{NL}} + 1) \cdot n_x^2 \right). \quad (2.107)$$

Wird die Lösung dagegen als Inversion mit anschließender Multiplikation realisiert, ergibt sich die Anzahl der Operationen $\mathcal{L}_{v'}$:

$$\mathcal{L}_{v'} = n_{\text{iter}} \cdot \left(\frac{4}{3}n_x^3 - \frac{1}{3}n_x + (n_{\text{NL}} + 1) \cdot n_x^2 \right). \quad (2.108)$$

Der Aufwand für die Auswertung einer Gleichung ist von der Anzahl der Parameter, der EPD-Symbole und der NLPD-Symbole abhängig. Die durchschnittliche Anzahl der EPD-Symbole pro Gleichung entspricht der durchschnittlichen Anzahl von Schaltungsvariablen n_{DV} , von der eine Gleichung abhängig ist. Dies entspricht der Anzahl der EPD-Symbole, die beim Auswerten einer Gleichung betrachtet werden müssen. Die genaue Anzahl der Operationen ist von den in dem Gleichungssystem verwendeten affinen Funktionen abhängig. Daher wird die Anzahl der für die Auswertung benötigten Operationen $\mathcal{L}_{\text{load}}$ in O-Notation angegeben:

$$\mathcal{L}_{\text{load}} = \mathcal{O}((n_p + n_{\text{DV}} + n_{\text{NL}}) \cdot n_x). \quad (2.109)$$

Es gilt:

$$\mathcal{L}_{\text{load}} \in \mathcal{O}(\mathcal{L}_v) \wedge \mathcal{L}_{\text{load}} \in \mathcal{O}(\mathcal{L}_{v'}). \quad (2.110)$$

Daher wird $\mathcal{L}_{\text{load}}$ im Weiteren nicht näher betrachtet.

Die für Algorithmus 3 benötigte Zeit wird durch das Lösen von

$$\mathbf{C} = \mathbf{J}^{-1} \cdot \mathbf{P} \quad (2.111)$$

und das Auswerten des Gleichungssystems $f(\vec{\hat{x}}, \vec{\hat{p}})$ bestimmt. Die Zeit für das Aufbauen der PPD-Matrix \mathbf{P} kann vernachlässigt werden. Das Lösen der Gleichung (2.111) wird

2 Simulation von Parameterunsicherheiten

$n_{\text{opt},b}$ -mal vor den affinen Lösungsiterationen (Algorithmus 2, Zeile 9) und $n_{\text{opt},d} \cdot n_{\text{iter}}$ -mal währenddessen ausgeführt. Die Optimierungen vor der affinen Lösungsiteration werden im Folgenden vernachlässigt, da in der Regel $n_{\text{opt},b} \ll n_{\text{opt},d} \cdot n_{\text{iter}}$ gilt. Die Dimension von \mathbf{J} ist $\mathbb{R}^{n_x \times n_x}$, die von \mathbf{P} ist $\mathbb{R}^{n_x \times (n_p+1)}$. Damit ergibt sich die Anzahl der Operationen \mathcal{L}_c für das Lösen von Gleichung (2.111) zu:

$$\mathcal{L}_c = n_{\text{opt},d} \cdot n_{\text{iter}} \cdot \left(\frac{1}{3}(n_x^3 - n_x) + (n_p + 1) \cdot n_x^2 \right). \quad (2.112)$$

Eine Realisierung als Inversion und anschließender Multiplikation mit einem affinerwertigen Vektor ist auch hier möglich, bringt allerdings keine Vorteile, da alle PPD-Symbole in dem entsprechenden affinerwertigen Vektor \vec{p} enthalten sind. Die Komplexität für das Auswerten entspricht Gleichung (2.109) und kann daher auch hier vernachlässigt werden. Zur Bestimmung des Worst Case der Gesamtlaufzeit werden die Annahmen $n_{\text{NL}} = n_x$, $n_p \gg 1$, $n_x \gg 1$ getroffen. Für die Gesamtlaufzeit mit Realisierung als LR-Zerlegung \mathcal{L}_{LR} ergibt sich damit:

$$\begin{aligned} \mathcal{L}_{\text{LR}} &= \mathcal{L}_v + \mathcal{L}_c \\ &= n_{\text{iter}} \cdot \left(\frac{1}{3}n_x^3(1 + n_{\text{opt},d}) - \frac{1}{3}n_x(1 + n_{\text{opt},d}) + (n_x^2 + n_p \cdot n_{\text{opt},d}) \cdot n_x^2 \right). \end{aligned} \quad (2.113)$$

Wird das Lösen dagegen als Inversion mit anschließender Multiplikation realisiert, ergibt sich für die Gesamtlaufzeit \mathcal{L}_{Inv} :

$$\begin{aligned} \mathcal{L}_{\text{Inv}} &= \mathcal{L}_v + \mathcal{L}_c \\ &= n_{\text{iter}} \cdot \left(\frac{1}{3}n_x^3(4 + n_{\text{opt},d}) - \frac{1}{3}n_x(1 + n_{\text{opt},d}) + (n_x + n_p \cdot n_{\text{opt},d}) \cdot n_x^2 \right) \end{aligned} \quad (2.114)$$

Je nach Realisierung ergibt sich also eine Laufzeitkomplexität von

$$\mathcal{O}(n_{\text{iter}} \cdot n_x^3) \quad \text{für die Inversion oder} \quad (2.115)$$

$$\mathcal{O}(n_{\text{iter}} \cdot n_x^4) \quad \text{für die LR-Zerlegung.} \quad (2.116)$$

In [27, Abschnitt 4.3] wird die Laufzeitkomplexität des EPD-Verfahrens zu $\mathcal{O}(n_{\text{iter}} \cdot n_x^4)$ bestimmt. Eine Verbesserung durch Verwendung der LR-Zerlegung wie dort angeregt, lässt sich jedoch nicht erreichen. Die im Folgenden beschriebene Implementierung verwendet Inversionen mit anschließenden Multiplikationen zur Berechnung von Gleichung (2.78), daher ist dort $\mathcal{O}(n_{\text{iter}} \cdot n_x^3)$ zu erwarten. Nicht berücksichtigt wurde die Abhängigkeit von n_{iter} von n_x , n_p und n_{NL} und n_{DV} , da diese von der Gleichung $f(\vec{x}, \vec{p})$ abhängt.

Tabelle 2.4: Anzahl der Fließkommaoperationen unter Vernachlässigung der Operationen für die Spaltenpivotstrategie, es gilt: $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\vec{b} \in \mathbb{R}^{n \times 1}$, $\mathbf{C} \in \mathbb{R}^{n \times m}$

Rechenoperation	# Fließkommaoperationen
Lösen von $\mathbf{A}\vec{x} = \vec{b}$ mit LR-Zerlegung	$\frac{1}{3}n^3 + n^2 - \frac{1}{3}n$ [110, S. 43]
Lösen von $\mathbf{A}\mathbf{X} = \mathbf{C}$ mit LR-Zerlegung	$\frac{1}{3}(n^3 - n) + mn^2$ [110, Gleichung (1.69)]
Matrix-Inversion \mathbf{A}^{-1} mit LR-Zerlegung aus $\mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{I}_n$	$\frac{4}{3}n^3 - \frac{1}{3}n$

3 Erweitertes EPD-Verfahren

3.1 Grenzen des Verfahrens

Das EPD-Verfahren ist durch verschiedene Ursachen in seiner Einsatzfähigkeit für allgemeine Gleichungssysteme und in seinem Konvergenzgebiet beschränkt. Diese werden im Folgenden dargestellt.

Ein nichtlineares Gleichungssystem (2.6) kann keine, eine oder mehrere Nullstellen (Abbildungen 3.1a bis 3.1c) besitzen. Durch die Unsicherheiten in einem parametrischen Gleichungssystem (2.7) existiert ein zusätzlicher Fall: Auch ohne nominale Nullstelle können Nullstellen durch die Parameterunsicherheiten existieren. Dieser Fall ist in Abbildung 3.1d dargestellt. Da das EPD-Verfahren zunächst die Nominallösung bestimmt und diese dann um Abweichungssymbole erweitert, kann das Verfahren in den Fällen, in denen keine Nominallösung existiert (Abbildungen 3.1a und 3.1d) keine Lösung berechnen. Existieren mehrere Nullstellen (Abbildung 3.1c), wird nur eine nominale Nullstelle gefunden, da das Newton-Raphson-Verfahren punktarithmetisch ausgeführt wird. Welche der mehrfachen nominalen Nullstellen gefunden wird, hängt von den gewählten Startwerten des Newton-Raphson-Verfahrens ab. Die gebietsarithmetische Erweiterung, die vom EPD-Verfahren berechnet wird, schließt in diesem Fall nur die Abweichungen um die einzelne, gewählte Nullstelle ein. Der globale Einschluss aller Abweichungen um alle Nullstellen ist nicht gewährleistet. Diese Beschränkung ist in der Praxis nicht relevant, da mehrfache Nullstellen nur bei Schaltungen mit mehreren stabilen Arbeitspunkten und chaotischem Verhalten auftreten können. Diese Schaltungsklassen sind auch mit herkömmlichen Simulatoren nicht behandelbar. Im Folgenden werden daher nur Schaltungen mit genau einer nominalen Nullstelle betrachtet.

Eine weitere Beschränkung ergibt sich aus der Berechnung des Residuums mit affiner Arithmetik. Durch Überschätzung kann es zu Fällen kommen, in denen Funktionen mit beschränktem Definitionsbereich nicht berechnet werden können (vgl. Abbildung 2.7). Um die Berechnung zu ermöglichen, muss die Überschätzung verringert werden. Dies lässt sich durch kleinere Parameterunsicherheiten, Umformulierungen der zu berechnenden Gleichungen oder eine verbesserte Approximation (vgl. Abschnitt 2.1.4) der Funktion mit beschränktem Definitionsbereich erreichen.

Schließlich ergibt sich eine Beschränkung aus dem EPD-Verfahren selbst. Versuche haben gezeigt, dass Systeme mit großen Abweichungen und starken Nichtlinearitäten dazu führen, dass das EPD-Verfahren nicht konvergiert. Die Einschlussbedingung (2.72) kann in diesen Fällen nicht erfüllt werden, obwohl Nominallösungen für $\min(\vec{p})$ und $\max(\vec{p})$ sowie alle Werte dazwischen existieren. Dies liegt daran, dass r_{NLPD} schneller als r_{EPD} wächst, wenn x_{EPD} zum Erreichen des Einschlusses manipuliert wird. Der Zusammenhang zwischen den Größen x_{EPD} und r_{EPD} bzw. r_{NLPD} ist nur implizit durch

3 Erweitertes EPD-Verfahren

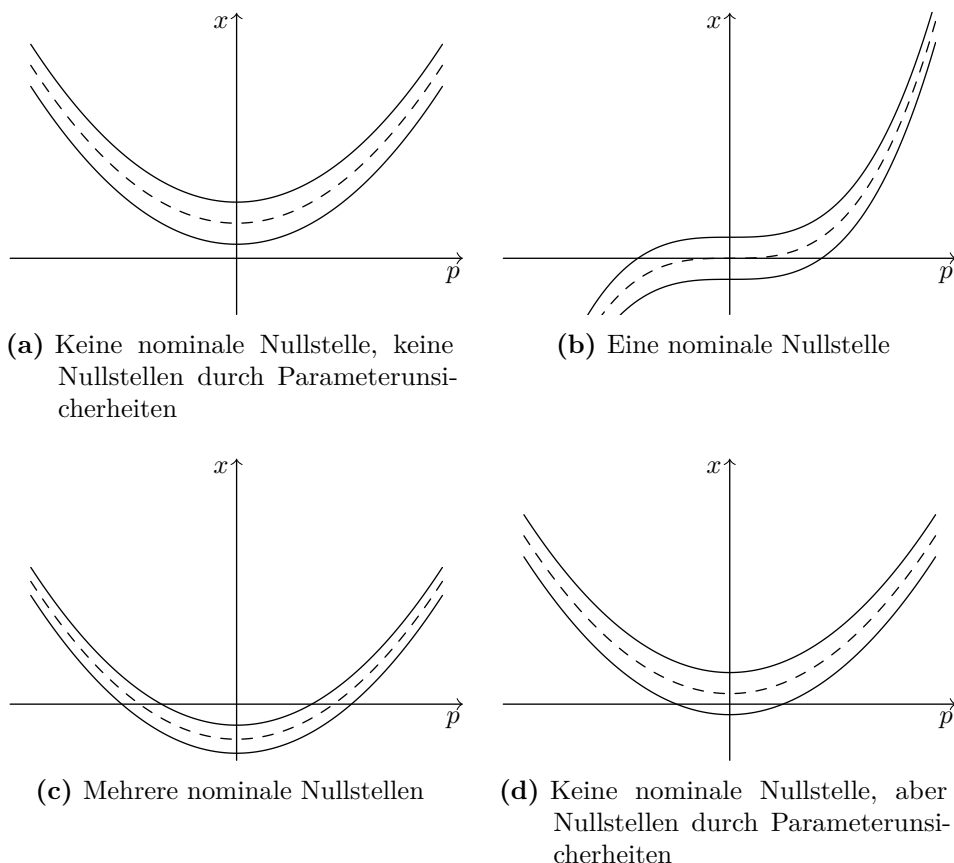


Abbildung 3.1: Verschiedene Anzahlen von Nullstellen. Die Lösung mit dem Nominalparametersatz ist gestrichelt dargestellt.

die Funktion $\vec{r} = f(\vec{x}, \vec{p})$ gegeben. Ein Konvergenzgebiet in Abhängigkeit von $\text{rad}(\vec{p})$ für eine gegebene Funktion f lässt sich daher a-priori nicht angeben. Im Folgenden werden daher zur Abschätzung des Konvergenzgebiets experimentelle Ergebnisse verwendet. Dazu werden die Parameterunsicherheiten schrittweise vergrößert und ein Lösungsversuch mit Algorithmus 2 unternommen. Ist dieser erfolgreich, wird der Punkt im Raum, der durch die Parameterunsicherheiten aufgespannt wird, entsprechend markiert. Aus dieser Punktwolke wird mit Hilfe der Delaunay-Triangulation ein Polygonnetz erzeugt, dessen konvexe Hülle schließlich dargestellt wird (Algorithmus 5). Ein so erzeugtes Ergebnis ist beispielhaft in Abbildung 3.4 dargestellt.

Algorithmus 5 besitzt den Nachteil, dass $p_{PPD_1, max} \dots p_{PPD_{n_p}, max}$ vom Benutzer so gewählt werden müssen, dass das Konvergenzgebiet des Algorithmus eingeschlossen wird. Dies lässt sich insbesondere für höhere Dimensionen nur schwer sicherstellen. Außerdem werden $n_{steps}^{n_p}$ Lösungsversuche durchgeführt, wobei n_{steps} die Anzahl der Schritte pro Parameter bezeichnet. Eine mögliche Alternative ist in Algorithmus 6 dargestellt. Hierbei werden die Abweichungen eines Parameters schrittweise erhöht, bis

das EPD-Verfahren nicht mehr konvergiert. Danach wird die Abweichung des nächsten Parameters erhöht. Dies wird solange wiederholt, bis alle Parameter bearbeitet wurden. Dadurch ist das Verfahren von der Abarbeitungsreihenfolge der Parameter abhängig. Eine beispielhafte vollständige Exploration in alle Richtungen ergibt ein Konvergenzgebiet nach Abbildung 3.2. Wird zuerst nach dem Parameter p_1 und dann nach p_2 exploriert, ergibt sich das in Abbildung 3.3a gezeigte Konvergenzgebiet. Bei der Exploration nach p_2 und nachfolgend p_1 ergibt sich das in Abbildung 3.3b gezeigte Konvergenzgebiet. Beide Explorationsrichtungen stellen daher nur eine Approximation des tatsächlichen Konvergenzgebiets aus Abbildung 3.2 dar. Für eine gegebene Parameterreihenfolge ist das Verfahren jedoch deterministisch. Als Bewertungsmaß wird das Volumen der konvexen Hülle des Konvergenzgebiets V_c verwendet. Da die Einheiten der Parameter im Allgemeinen unterschiedlich sind, ist die Einheit des Volumens eine gemischte Einheit.

Algorithmus 5 Algorithmus zur Bestimmung des Konvergenzgebiets von Algorithmus 2

```

1: procedure EXPLOREPARAMETERRANGE
2:   for  $\vec{p}_{(1)PPD_1} = p_{PPD_1,min} \cdots p_{PPD_1,max}$  do
3:     for  $\vec{p}_{(2)PPD_2} = p_{PPD_2,min} \cdots p_{PPD_2,max}$  do
4:        $\vdots$ 
5:       for  $\vec{p}_{(n_p)PPD_{n_p}} = p_{PPD_{n_p},min} \cdots p_{PPD_{n_p},max}$  do
6:         if SUCCESSFUL(SOLVEAFFINE( $\vec{x}, \vec{p}$ )) then
7:            $r = \text{MARKASUCCESSFUL}(\vec{p}, r)$   $\triangleright r \in \mathbb{R}^{n_p}$ 
8:         end if
9:       end for
10:     end for
11:   end for
12:    $hull = \text{CONVEXHULL}(\text{DELAUNAYTRIANGULATION}(r))$ 
13:    $V_c = \text{VOLUME}(hull)$ 
14:   return  $V_c$ 
15: end procedure

```

Algorithmus 6 Algorithmus zur automatischen Bestimmung des Konvergenzgebiets von Algorithmus 2

```

1: procedure AUTOEXPLOREPARAMETERRANGE
2:   maxDev = {}
3:   for parameterName ∈ listOfParameters do
4:     parameterDeviation = GETSTARTDEVIATION(parameterName)
5:     while SUCCESSFUL(SOLVEAFFINE( $\vec{x}, \vec{p}$ )) do
6:        $r = \text{MARKASUCCESSFUL}(\vec{p}, r)$   $\triangleright r \in \mathbb{R}^{n_p}$ 
7:       parameterDeviation += GETSTEPSIZE(parameterName)
8:     end while
9:     parameterDeviation -= GETSTEPSIZE(parameterName)
10:    maxDev.append(parameterDeviation)
11:  end for
12:  hull = CONVEXHULL(DELAUNAYTRIANGULATION( $r$ ))
13:  PLOT( $hull$ )
14:   $V_c = \text{VOLUME}(hull)$ 
15:  return maxDev,  $V_c$ 
16: end procedure

```

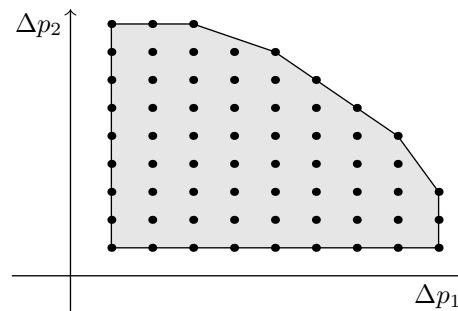


Abbildung 3.2: Exploration eines Konvergenzgebietes in Abhängigkeit von zwei Parametern p_1 und p_2 nach Algorithmus 5

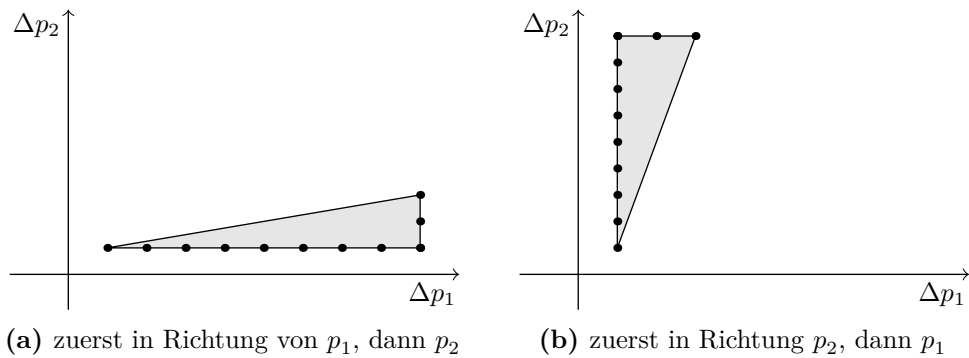


Abbildung 3.3: Exploration eines Konvergenzgebietes in Abhängigkeit von zwei Parametern p_1 und p_2 nach Algorithmus 6

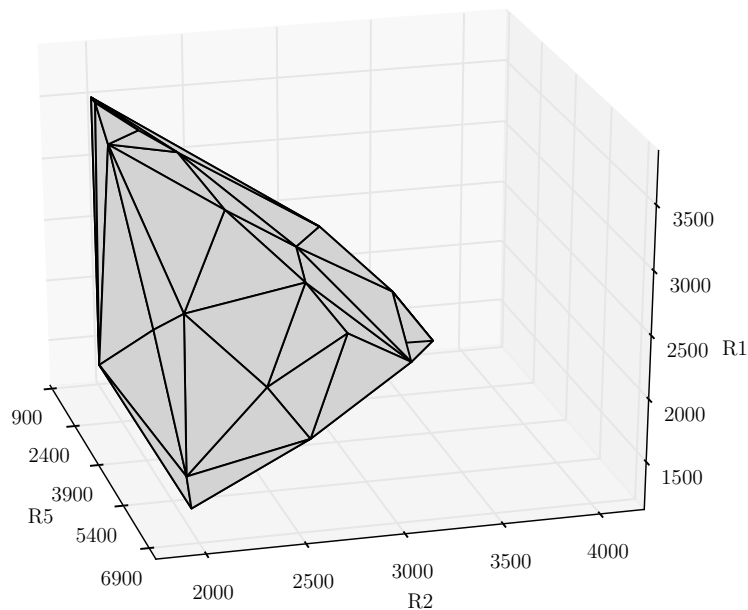


Abbildung 3.4: Darstellung des dreidimensionalen Konvergenzgebiets von Algorithmus 5 für eine Beispielschaltung

3.2 Maß für die Überschätzung

Die Einschlüsse, die mit Hilfe der affinen Arithmetik und des EPD-Verfahrens berechnet werden, sind aus verschiedenen Gründen nicht minimal, sondern mit Überschätzung behaftet (vgl. Abschnitt 2.1.3.2). Um verschiedene Ansätze vergleichen zu können, wird ein Maß für die Überschätzung benötigt. Da das minimale Gebiet im Allgemeinen nicht direkt berechnet werden kann, ist eine Abschätzung erforderlich. Dazu werden Monte-Carlo-Simulationen durchgeführt und aus diesen Stichproben der minimale und der maximale Wert für jeden Simulationsschritt v_n bestimmt. Wenn die Anzahl der Monte-Carlo-Stichproben ausreichend hoch ist, entsprechen die so bestimmten Werte näherungsweise dem minimalen Einschluss. Diese Abschätzung ist pessimistisch, da weitere Monte-Carlo-Stichproben außerhalb des durch die Minima und Maxima bestimmten Schlauches liegen können. Die Differenz zwischen Maximum und Minimum aus den Monte-Carlo-Simulationen wird als Intervallbreite d_{MC} bezeichnet. Die Extremwerte werden für jeden Zeitschritt aus r_{MC} Monte-Carlo-Simulationen bestimmt. Entsprechend ist die Intervallbreite d_A für das Ergebnis der gebietsarithmetischen Simulation definiert. Beide Intervallbreiten sind in Abbildung 3.5 schematisch dargestellt. Die Intervallbreite d_A wird, um ein relatives Maß zu erhalten, auf d_{MC} bezogen und aus den so erhaltenen Werten wird das arithmetische Mittel über alle Simulationsschritte v_n (vgl. Abschnitt 2.2.4.1)

3 Erweitertes EPD-Verfahren

gebildet. Dieses Maß wird als mittlere relative Überschätzung m_{OA} bezeichnet:

$$m_{\text{OA}} = \frac{1}{n} \sum_{v=v_0}^{v_n} \frac{d_A(v)}{d_{\text{MC}}(v)}. \quad (3.1)$$

Im Folgenden werden ausschließlich transiente Simulationen, d.h. $v = t$ betrachtet:

$$m_{\text{OA}} = \frac{1}{n} \sum_{t=t_0}^{t_n} \frac{d_A(t)}{d_{\text{MC}}(t)}. \quad (3.2)$$

Dabei ist zu beachten, dass durch die dynamische Zeitschrittsteuerung der transienten Simulation Ergebnisse der einzelnen Monte-Carlo-Simulationen und der affinen Simulation nicht notwendigerweise an den gleichen Zeitpunkten berechnet wurden. Daher werden die Zeitpunkte aus der affinen Simulation als Referenz benutzt und die Werte der Monte-Carlo-Simulationen zu diesen Zeitpunkten gegebenenfalls linear interpoliert. Damit der Einschluss der affinen durch die Monte-Carlo-Ergebnisse gegeben ist, müssen die Zeitschritte ausreichend klein sein.

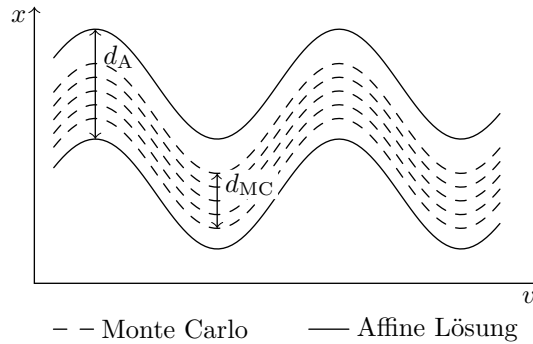


Abbildung 3.5: Intervallbreiten d_A und d_{MC} zur Bestimmung der mittleren relativen Überschätzung m_{OA}

3.3 Ausnutzung von Monotonieeigenschaften

Eine Idee zur Vergrößerung des Konvergenzgebiets des EPD-Verfahrens ist die Ausnutzung der Monotonieeigenschaften des Gleichungssystems. Monotonie ist für eine Funktion $y = f(x)$ wie folgt definiert:

$$\text{monoton steigend} \quad f(x_1) \leq f(x_2) \quad \text{für } x_1 < x_2 \quad (3.3)$$

$$\text{monoton fallend} \quad f(x_1) \geq f(x_2) \quad \text{für } x_1 < x_2. \quad (3.4)$$

Ist der Zusammenhang zwischen dem Parameter p und der Zustandsgröße x explizit, d.h. in der Form $x = f(p)$, gegeben und monoton, treten die Extremwerte der Zustandsgrößen

für die Extremwerte der Parameter auf:

$$\left. \begin{array}{l} x_{\min} = f(p_{\min}) \\ x_{\max} = f(p_{\max}) \end{array} \right\} \text{monoton steigend} \quad (3.5)$$

$$\left. \begin{array}{l} x_{\min} = f(p_{\max}) \\ x_{\max} = f(p_{\min}) \end{array} \right\} \text{monoton fallend} \quad (3.6)$$

In diesen Fällen könnte auf das EPD-Verfahren verzichtet und stattdessen Corner-Case-Simulationen mit den Extremwerten der Parameter durchgeführt werden. Die Extremwerte der Zustandsgrößen könnten so auch ohne Verwendung des EPD-Verfahrens bestimmt werden. Dadurch würde die Laufzeit verringert und die Überschätzung vermieden, da Punktarithmetik verwendet werden kann.

Die Monotonieeigenschaften können mit Hilfe der ersten Ableitung bestimmt werden:

$$\frac{df(p)}{dp} > 0 \quad \text{monoton wachsend} \quad (3.7)$$

$$\frac{df(p)}{dp} < 0 \quad \text{monoton fallend} \quad (3.8)$$

Ist ein Parameter p mit Unsicherheiten behaftet, muss die entsprechende Bedingung für alle $p \in \check{p}$ gelten. Dabei bezeichnet \check{p} das Gebiet, das alle Unsicherheiten des Parameters p enthält. Dies lässt sich zum einen durch die Betrachtung der zweiten Ableitung überprüfen. Gilt $\frac{d^2 f(p)}{dp^2} > 0$ bzw. $\frac{d^2 f(p)}{dp^2} < 0$, ist es ausreichend, die Extremwerte von p bei der Auswertung von $\frac{df(p)}{dp}$ zu betrachten. Wenn dies nicht der Fall ist, muss eine Extremwertuntersuchung durchgeführt werden, um die Bedingungen (3.7) bzw. (3.8) zu prüfen. Eine weitere Möglichkeit ist die gebietsarithmetische Auswertung der Ableitung $\frac{df(\hat{p})}{d\hat{p}}$. Dadurch können auch nichtmonotone Funktionen behandelt werden. Eine gesonderte Behandlung von nichtmonotonen Funktionen ist nicht erforderlich. Die symbolischen ersten Ableitungen liegen bereits zur Berechnung der initialen affinwertigen Lösung vor (Algorithmus 2, Zeile 5). Die zweiten Ableitungen dagegen müssten gesondert berechnet werden. Daher wird im Folgenden die gebietsarithmetische Methode zur Bestimmung der Monotonieeigenschaften verwendet.

Die bisherigen Ausführungen beziehen sich auf eine explizite Darstellung der Gleichungssysteme. Im Allgemeinen ist jedoch nur eine implizite Darstellung (2.7) des Zusammenhangs zwischen x und p bekannt. Die explizite Darstellung $x = g(p)$ existiert im Allgemeinen nicht global. Es ist aber möglich, dass lokal in der Umgebung von $f(x', p') = 0$ eine explizite Darstellung existiert. Der Satz von der impliziten Funktion (engl. Implicit Function Theorem (IFT)) bietet eine Möglichkeit, Angaben zu der lokalen Existenz von $g(p)$ machen zu können und die Ableitung $\frac{dg(p)}{dp}$ der expliziten Funktion $g(p)$ – nicht aber die Funktion $g(p)$ selbst – zu berechnen. Er lautet nach [7, Theorem 5]:

Satz 3.1. *Satz von der impliziten Funktion. Sei $X \times P$ eine offene Untermenge von $\mathbb{R}^n \times \mathbb{R}^m$ und $f : X \times P \rightarrow \mathbb{R}^n$ sei differenzierbar. Es wird angenommen, dass die Ableitung $\frac{df(x,p)}{dx}$ stetig auf $X \times P$ und invertierbar ist. Es sei $\bar{y} = f(\bar{x}, \bar{p})$. Dann existieren*

3 Erweitertes EPD-Verfahren

Nachbarschaften $U \subset X$ und $W \subset P$ von \bar{x} und \bar{p} , auf denen x eindeutig als Funktion von p definiert ist. Das bedeutet, es gibt eine Funktion $g : W \rightarrow U$, sodass gilt:

- $f(g(p), p) = \bar{y}$ für alle $p \in W$
- Für alle $p \in W$ ist $g(p)$ die eindeutige Lösung von $f(x, p) = \bar{y}$ in U . Es gilt im Besonderen $g(\bar{p}) = \bar{x}$.
- $g(p)$ ist differenzierbar auf W und es gilt

$$\begin{bmatrix} \frac{dg_1}{dp_1} & \cdots & \frac{dg_1}{dp_m} \\ \vdots & \ddots & \vdots \\ \frac{dg_n}{dp_1} & \cdots & \frac{dg_n}{dp_m} \end{bmatrix} = - \begin{bmatrix} \frac{df_1}{dx_1} & \cdots & \frac{df_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dx_1} & \cdots & \frac{df_n}{dx_n} \end{bmatrix}^{-1} \begin{bmatrix} \frac{df_1}{dp_1} & \cdots & \frac{df_1}{dp_m} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dp_1} & \cdots & \frac{df_n}{dp_m} \end{bmatrix}^{-1} \quad (3.9)$$

Die Aussagen des Satzes gelten in W und U , also auch wenn $x = g(p)$ nur lokal existiert. Er trifft allerdings keine Aussagen über die Größe der beiden Mengen. Daher bietet sich auch hier die Berechnung der Matrizen $\frac{df}{dx}$ und $\frac{df}{dp}$ mit affiner Arithmetik an. Damit die Matrix $\frac{df}{dx}$ invertierbar ist, muss

$$\det \left(\frac{df}{dx} \right) \neq 0 \quad (3.10)$$

gelten. Hierbei ergibt sich allerdings die Schwierigkeit, dass x noch nicht bekannt ist, zur Bestimmung der Ableitung mit Gleichung (3.9) jedoch benötigt wird. Eine Abschätzung aus der Linearisierung ohne zusätzliche Erweiterung wie beim EPD-Verfahren ist nicht ausreichend, da durch den Linearisierungsfehler der Einschluss nicht sichergestellt ist. Eine direkte Anwendung des IFTs zur Bestimmung der Parameterkombination, die zu den Extremwerten der Zustandsvariablen führt, ist daher nicht möglich. Denkbar ist eine Kombination des IFTs mit dem EPD-Verfahren. Dabei wird mit Hilfe des IFTs ein zusätzliches Abbruchkriterium generiert, wenn die Voraussetzungen des Satzes erfüllt sind. Eine entsprechende Erweiterung des EPD-Verfahrens (Algorithmus 2) ist in Algorithmus 7 dargestellt. Wenn die Bedingungen des IFTs erfüllt sind (Zeile 4f.), wird für alle Gleichungen und alle Parameter berechnet, für welche Kombination der komponentenweisen Minima bzw. Maxima der Parameter die Extremwerte der Zustandsgrößen erreicht werden (Zeilen 6 bis 16). Die so berechneten Extremwerte werden in dem Intervall `[border]` gespeichert (Zeile 17). Durch die Erweiterung des Abbruchkriteriums in Zeile 17 wird das Lösungsgebiet auf das Gebiet beschränkt, das durch die Extremwerte der Zustandsgrößen aufgespannt wird, solange die Voraussetzungen des IFT erfüllt sind. Experimente haben jedoch gezeigt, dass die Voraussetzungen des IFTs bereits für eine Größe von Parameterunsicherheiten verletzt werden, bei der das EPD-Verfahren noch konvergiert. Eine Erweiterung des Konvergenzgebiets mit dem IFT durch die Vermeidung von Überschätzung ist so nicht möglich. Algorithmus 7 ist daher keine sinnvolle Erweiterung des EPD-Verfahrens (Algorithmus 2) und wird daher im Folgenden nicht weiter verfolgt.

Algorithmus 7 EPD-Verfahren erweitert um Abbruchkriterium aus dem IFT

```

1: procedure SOLVEEPDWITHIFT( $\vec{p}, f(\vec{x}, \vec{p})$ )
  :
2:   repeat
  :
3:      $\vec{v} = \vec{0}$ 
4:      $isLocallyExplicit = \text{CHECKIFTPREREQUISITES} \left( \vec{f}, \vec{x}, \vec{p}, \frac{df}{dx}, \frac{df}{dp} \right)$ 
5:     if  $isLocallyExplicit$  then
6:       for all  $\vec{f}_{(i)} \in \vec{f}$  do
7:         for all  $\vec{p}_{(j)} \in \vec{p}$  do
8:           if  $\text{MONOTONICINCREASING}(\vec{f}_{(i)}, \vec{p}_{(j)})$  then
9:              $border_{j,\min} = \min(\vec{p}_{(j)})$ 
10:             $border_{j,\max} = \max(\vec{p}_{(j)})$ 
11:           else
12:              $border_{j,\min} = \max(\vec{p}_{(j)})$ 
13:             $border_{j,\max} = \min(\vec{p}_{(j)})$ 
14:           end if
15:         end for
16:       end for
17:        $[border] = [border_{\min}, border_{\max}]$ 
18:     end if
  :
19:   until ( $|\vec{s}_{(i)}| \approx 1 \vee \vec{x}_{(i)EPD} < \epsilon \vee n_{\text{iter}} > n_{\text{iter,max}} \vee (isLocallyExplicit \wedge$ 
   $\vec{x}$  not inside of  $[border])$ )
20:   return  $\vec{x}$ 
21: end procedure

```

3.4 Gebietsaufteilungen und Gebietszusammenfassungen

Ein weiterer möglicher Ansatz zur Vereinfachung von gebietsarithmetischen Problemen ist der Einsatz von Gebietsaufteilungen. Dabei werden die Ursprungsgebiete in kleinere Gebiete unterteilt. Da die Überschätzung der Gebietsarithmetiken vom Radius der Abweichung abhängig ist (vgl. Abschnitt 2.1.3.2), ermöglichen kleinere Gebiete verbesserte Konvergenzeigenschaften der Lösungsalgorithmen bzw. ermöglichen erst Konvergenz. Entsprechende Ansätze für nicht parametrische Probleme werden in [12, 13, 32, 35, 79] vorgestellt, parametrische Probleme werden in [2, 3, 51, 81] behandelt. Im Gegensatz zu den dort behandelten Problemen ist der Zusammenhang zwischen den Parametern und den Zustandsgrößen bei den Gleichungen zur Schaltungssimulation (2.7) nichtlinear. Daher ist es nicht möglich, wie in [2, 3, 51, 81] den Zustandsraum aufzuteilen, sondern die Gebietsaufteilung muss im Parameterraum stattfinden. Mit Hilfe dieser Aufteilung wird

dann die entsprechende Teilung im Zustandsraum berechnet. Die aufgeteilten Parameter werden in einem Parameterbaum und die Lösungen im Zustandsraum im entsprechenden Blatt eines Lösungsbaums gespeichert (Abbildung 3.6). Die Teilung eines Parameters $\vec{p}_{(i)}$ in die zwei Teile $\vec{p}'_{(i)}$ und $\vec{p}''_{(i)}$ kann wie folgt dargestellt werden:

$$\vec{p}'_{(i)} = p_0 - \frac{\vec{p}_{(i)}}{2} + \frac{\vec{p}_{(i)}}{2} \varepsilon_i \quad (3.11)$$

$$\vec{p}''_{(i)} = p_0 + \frac{\vec{p}_{(i)}}{2} + \frac{\vec{p}_{(i)}}{2} \varepsilon_i. \quad (3.12)$$

Die Aufteilung in mehr als zwei Teile kann analog ausgeführt werden. Auch eine Aufteilung in unterschiedlich große Gebiete ist möglich. Zur Darstellung der Anwendbarkeit von Gebietsaufteilungen werden im Folgenden vereinfachend nur Gebietshalbierungen betrachtet. Durch die Aufteilungen ergeben sich 2^{n_p} Teilprobleme, die gelöst werden müssen. Dadurch steigt die benötigte Rechenzeit an, allerdings verringert sich die Überschätzung bzw. die Lösung des Gleichungssystems für eine gegebene Größe der Parameterunsicherheiten wird erst ermöglicht. Ein einfaches Beispiel ist in Abbildung 3.7 dargestellt. Die jeweils angegebene Anzahl von Gebietsaufteilungen wird erzwungen.

In Abschnitt 3.6 werden Strategien vorgestellt, um nur einzelne Parameter aufzuteilen und so die Explosion der Komplexität zu begrenzen und gleichzeitig die Vorteile der Gebietsaufteilungen beizubehalten.

Nach den Berechnungen für die aufgeteilten Parametergebiete werden die Lösungen wieder zusammengefügt. Dabei bilden sie eine inklusionsisotone Lösung des Ursprungsproblems. Dazu werden zwei unterschiedliche Zusammenfassungsverfahren verwendet: Algorithmus 8 aus [86] und Algorithmus 9. Beide bestimmen zunächst einen neuen Zentralwert x_0 aus der Ober- bzw. Untergrenze der aufgeteilten Gebiete $\vec{x}'_{(i)}$ und $\vec{x}''_{(i)}$ (jeweils Zeilen 3 bis 5). Algorithmus 8 berechnet die neuen Abweichungssymbole als Summe der aufgeteilten Abweichungssymbole. Diese Version besitzt den Nachteil, dass Gebiete, die sich teilweise oder vollständig überlappen, eine große Überschätzung erzeugen. Solche Überlappungen treten häufig an starken Nichtlinearitäten auf. Dies ist besonders bei transienten Simulationen ein Nachteil, da hier die Überschätzung eines Zeitschritts durch die numerische Integration in die folgenden Zeitschritte einfließt. Dieser Nachteil

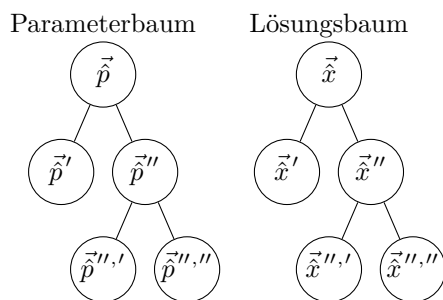


Abbildung 3.6: Parameter- und Lösungsbaum für zwei Aufteilungen

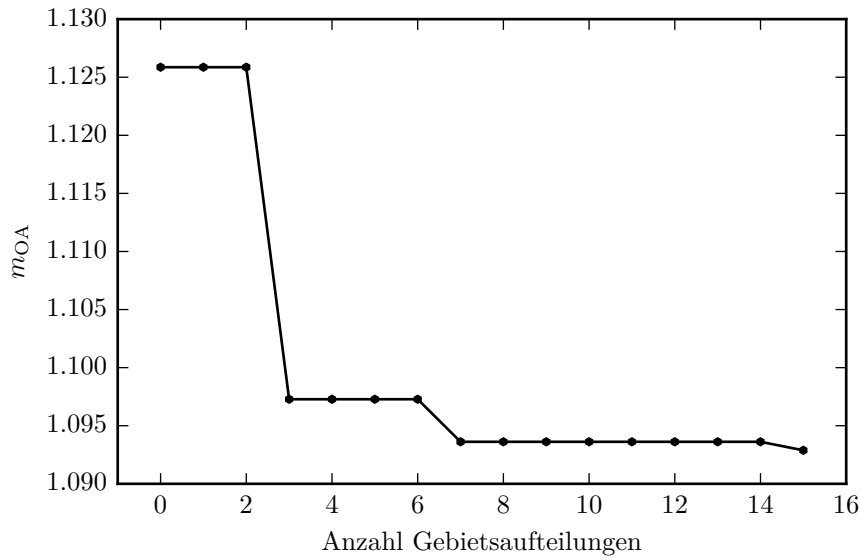


Abbildung 3.7: Überschätzung in Abhängigkeit von der Anzahl der erzwungenen Gebietsaufteilungen

wird durch die verbesserte Zusammenfassungsoperation (Algorithmus 9) vermieden. Hier werden die Abweichungssymbole auf die benötigte Größe skaliert (Zeilen 9 bis 13), sodass die Überschätzung verringert wird. In Abbildung 3.8 werden die Ergebnisse beider Zusammenfassungsoperationen für ein sich überlappendes und ein sich nicht überlappendes Gebiet grafisch gegenübergestellt.

Algorithmus 8 Zusammenfassungsoperation aus [86]

```

1: procedure MERGESOLUTIONS( $\vec{x}', \vec{x}''$ )
2:   for all  $\vec{x}'_{(i)} \in \vec{x}', \vec{x}''_{(i)} x \in \vec{x}''$  do
3:      $top = \max(\max(\vec{x}'_{(i)}), \max(\vec{x}''_{(i)}))$ 
4:      $bottom = \min(\min(\vec{x}'_{(i)}), \min(\vec{x}''_{(i)}))$ 
5:      $x_0 = \frac{bottom+top}{2}$ 
6:     for all  $i \in \{\mathbb{I}_{\vec{x}'}, \mathbb{I}_{\vec{x}''}\}$  do
7:        $x_i = x'_i + x''_i$ 
8:     end for
9:      $\vec{x}_{(i)} = x_0 + \sum_{\{\mathbb{I}_{\vec{x}'}, \mathbb{I}_{\vec{x}''}\}} x_i \varepsilon_i$ 
10:  end for
11: end procedure

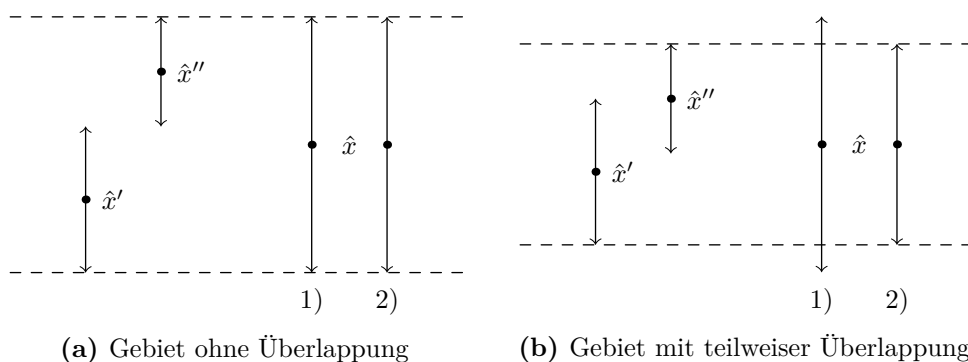
```

Algorithmus 9 Verbesserte Zusammenfassungsoperation

```

1: procedure MERGESOLUTIONS( $\vec{x}', \vec{x}''$ )
2:   for all  $\vec{x}'_{(i)} \in \vec{x}', \vec{x}''_{(i)} x \in \vec{x}''$  do
3:      $top = \max(\max(\vec{x}'_{(i)}), \max(\vec{x}''_{(i)}))$ 
4:      $bottom = \min(\min(\vec{x}'_{(i)}), \min(\vec{x}''_{(i)}))$ 
5:      $x_0 = \frac{bottom+top}{2}$ 
6:     for all  $i \in \{\mathbb{I}_{\vec{x}'}, \mathbb{I}_{\vec{x}''}\}$  do
7:        $x_i = x'_i + x''_i$ 
8:     end for
9:      $neededRadius = \frac{top-bottom}{2}$ 
10:     $scalingFactor = \frac{neededRadius}{rad(\hat{x})}$ 
11:    for all  $x_i \in \vec{x}'_{(i)}, \vec{x}''_{(i)}$  do
12:       $x_{i,new} = scalingFactor \cdot x_i$ 
13:    end for
14:     $\vec{\hat{x}}_{(i)} = x_0 + \sum_{\{\mathbb{I}_{\vec{x}'}, \mathbb{I}_{\vec{x}''}\}} x_{i,new} \cdot \varepsilon_i$ 
15:  end for
16: end procedure

```

**Abbildung 3.8:** Vergleich der Zusammenfassungsoperationen.

- 1) Zusammenfassungsoperation aus [86] (Algorithmus 8)
- 2) Verbesserte Zusammenfassungsoperation (Algorithmus 9)

3.4.1 Automatische Gebietsaufteilungen

Experimente haben gezeigt, dass nur an einzelnen Punkten Gebietsaufteilungen zur Lösung des Gleichungssystems benötigt werden. Es ist daher nicht zweckmäßig, das Parametergebiet für alle Simulationsschritte aufzuteilen. Allerdings lassen sich die Punkte, für die eine Aufteilung benötigt wird, nicht a-priori bestimmen. Daher werden im Folgenden zwei mögliche Kriterien zur automatischen Bestimmung vorgestellt. Das Kriterium 1 verwendet die Anzahl der affinen Lösungsiterationen n_{iter} . Ist diese größer als ein vorgegebener Schwellwert $n_{\text{iter,split}}$, wird eine Aufteilung durchgeführt. Das Kriterium 2 verwendet den Verlauf der Skalierungsfaktoren s_n über den Simulationsschritten als Kriterium. Bei einem konvergierenden System ist dieser Verlauf in der Regel monoton fallend. Ansteigende Werte deuten dagegen auf ein nicht konvergierendes System hin. Ein Beispiel für dieses Verhalten wird in Abbildung 3.9 gezeigt. Wird für diesen Wert ein benutzerdefinierter Schwellenwert Δs_{max} überschritten, wird eine Gebietsaufteilung durchgeführt. Bei der Abarbeitung von Lösungsbäumen wird das Maximum der Skalierungsfaktoren s_n aus den Kindknoten betrachtet.

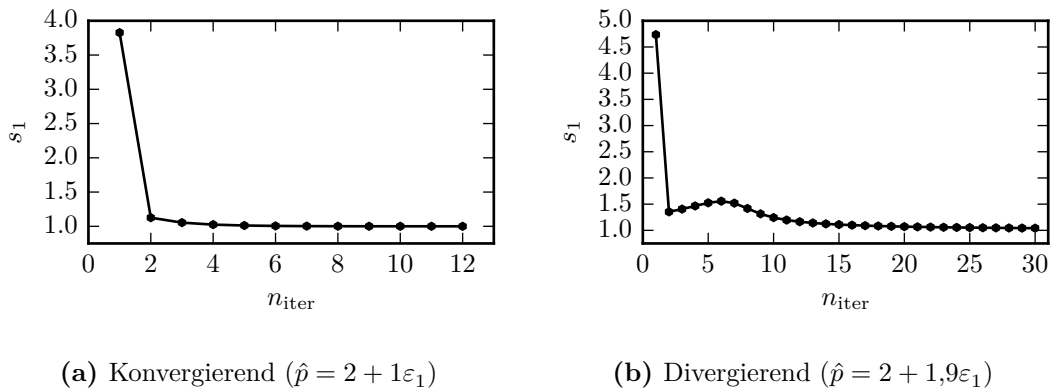


Abbildung 3.9: Verlauf des Skalierungsfaktors s_1 über der Anzahl der Iterationen für die Gleichung $e^{\hat{x}} - \hat{p}$

3.5 Transiente Simulationen mit Gebietsaufteilungen

Bei transienten Simulationen und der Verwendung von Gebietsaufteilungen ergibt sich das Problem, dass unter Umständen eine Lösung benötigt wird, die im vorangegangenen Zeitschritt nicht berechnet wurde. Dies tritt auf, wenn im aktuellen Zeitschritt t_n eine Aufteilung durchgeführt wird, die im vorangegangenen Schritt t_{n-1} nicht benötigt wurde. Der Parameterbaum für eine solche Situation ist in Abbildung 3.10 dargestellt. Zur Berechnung der numerischen Integration werden jedoch die Lösungen von mindestens einem vorangegangenen Zeitschritt benötigt. Zur Lösung dieses Problems sind zwei verschiedene Vorgehensweisen möglich. Das Backtracking-Verfahren berechnet die fehlenden Aufteilungsschritte in den vorangegangenen Zeitschritten nach. Dies ist in Abbildung 3.11a dargestellt. Das Verfahren besitzt den Nachteil, dass gegebenenfalls eine Nachberechnung

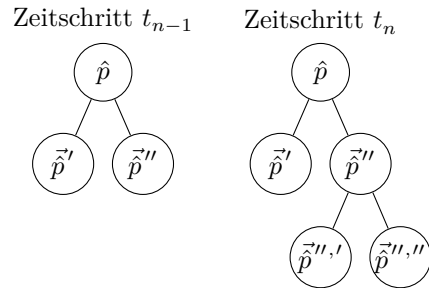
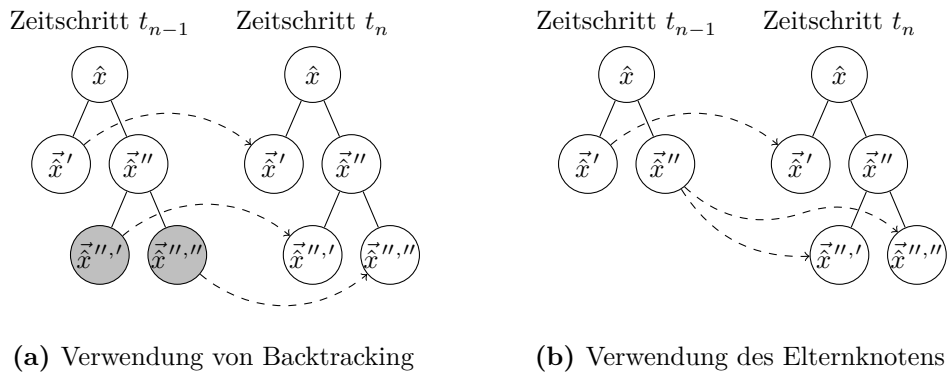


Abbildung 3.10: Parameterbaum für zwei Zeitschritte t_{n-1} und t_n . In Schritt t_n wird eine, in t_{n-1} werden zwei Aufteilungen zur Lösung benötigt.

der im aktuellen Zeitschritt t_n benötigten Aufteilung für alle vorangegangenen Zeitschritte $t_{n-1}, t_{n-2}, \dots, t_0$ stattfinden muss. Auf diese Weise werden in allen Zeitschritten nach und nach die maximale Anzahl von Aufteilungen berechnet. Experimente haben jedoch gezeigt, dass nur an wenigen Stellen viele Aufteilungen benötigt werden. Es werden daher zahlreiche Aufteilungen unnötigerweise berechnet. Um dies zu vermeiden, kann zur Berechnung der Ableitung der entsprechende Elternknoten des Lösungsbaums aus dem vorangegangenen Zeitschritt verwendet werden. Die Lösung aus dem Elternknoten schließt die aufgeteilte Lösung in den Kindknoten des Lösungsbaums garantiert ein. Diese Variante ist in Abbildung 3.11b dargestellt.



(a) Verwendung von Backtracking

(b) Verwendung des Elternknotens

Abbildung 3.11: Lösungsbaume von zwei Zeitschritten einer transienten Simulation für zwei Verfahren zur Berechnung der numerischen Integration. Die gestrichelten Pfeile zeigen die Quellen für die Lösungen der vergangenen Zeitschritte zur Berechnung der numerischen Integration. Die Aufteilungen, die zur Bestimmung der Lösung in Zeitschritt t_{n-1} nicht erforderlich waren, zur Bestimmung der numerischen Ableitung für Zeitschritt t_n jedoch nachberechnet wurden, sind grau markiert.

3.5.1 Zusammenfassungsstrategien

Bei transienten Simulationen gibt es – im Gegensatz zu den anderen Simulationsarten – verschiedene Möglichkeiten zur Zusammenfassung der aufgeteilten Lösungen. Eine Möglichkeit ist die Zusammenfassung der Lösungen nach jedem Zeitschritt, diese Strategie wird im Folgenden als *Merge Each* bezeichnet. Mit ihr werden ausschließlich Parameter von Zeitschritten aufgeteilt, die gemäß der Aufteilungskriterien eine Aufteilung benötigen. Andererseits muss in Fällen, in denen diese Aufteilung über mehrere Zeitschritte hinweg benötigt wird, der Parameterbaum in jedem Zeitschritt vollständig neu aufgebaut werden. Dies benötigt viele Iterationen, da die Entscheidungen zur Aufteilung nicht a-priori getroffen werden können. Eine gegensätzliche Strategie ist das Beibehalten einer einmal bestimmten Aufteilung über alle folgenden Zeitschritte (Strategie *Do Not Merge*). Sie bietet Vorteile, da die Aufteilung nicht in jedem Zeitschritt neu bestimmt werden muss. Allerdings werden auf diese Weise unter Umständen auch Zeitschritte mit aufgeteiltem Parametergebiet gelöst, bei denen dies aus Konvergenzgründen nicht erforderlich ist. Neben diesen beiden gegensätzlichen Strategien sind auch Kompromisse möglich. Die Strategie *Try Root First* versucht zunächst, eine Lösung ohne Gebietsaufteilung zu berechnen. Nur wenn diese nicht erfolgreich ist, wird die Aufteilung aus dem vorangegangenen Zeitschritt verwendet. Die Strategie *Merge n^{th} Step* dagegen behält die Aufteilung aus dem vorangegangenen Zeitschritt bei und führt das Zusammenfassen erst nach n Zeitschritten aus. Die Anzahl der Schritte n wird dabei vom Benutzer vorgegeben. Beispielhafte Lösungsbäume für die verschiedenen Zusammenfassungsstrategien werden in Abbildung 3.12 grafisch gegenübergestellt.

3 Erweitertes EPD-Verfahren

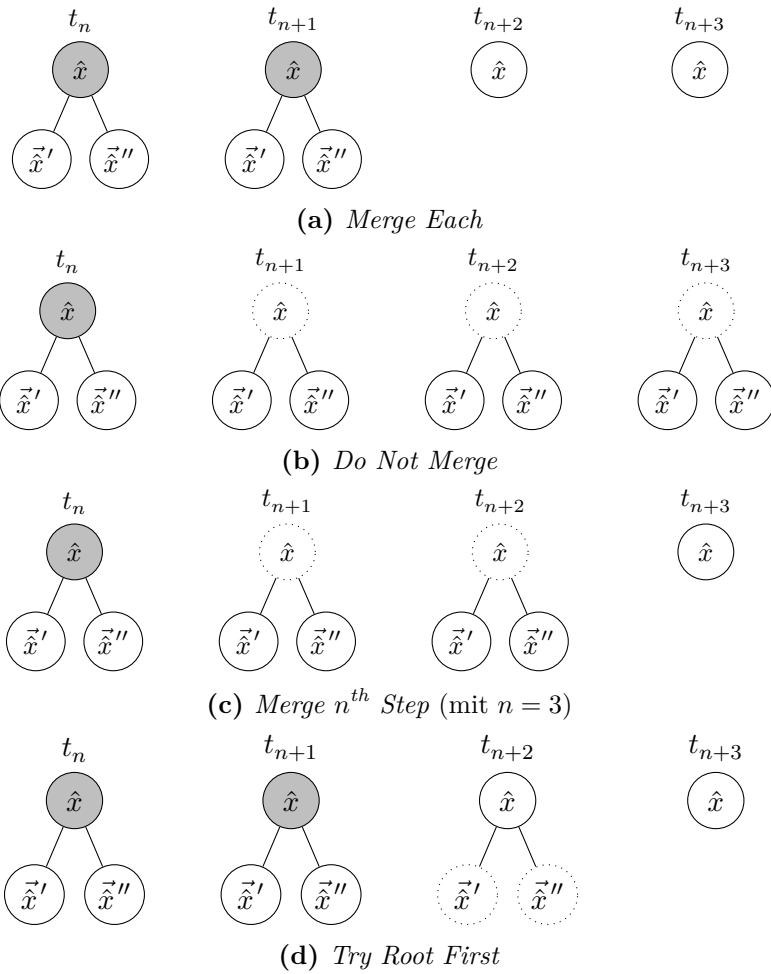


Abbildung 3.12: Vergleich der Zusammenfassungsstrategien. Es wird angenommen, dass in den ersten beiden Zeitschritten eine Aufteilung des Parameterraumes benötigt wird, in den folgenden Schritten jedoch nicht

- Knoten besucht und lösbar
- Knoten besucht und nicht lösbar
- ⊖ Knoten nicht besucht

3.6 Aufteilungsstrategien

Die Aufteilung aller Parameter verursacht bei n_p Parametern 2^{n_p} Aufteilungen, die getrennt voneinander berechnet werden. Dies führt zu einem exponentiellen Anstieg der benötigten Laufzeit. Die Auswahl aller Parameter zur Aufteilung wird als Strategie *Select All* bezeichnet. Experimente haben gezeigt, dass es häufig ausreichend ist, nur einen oder einige wenige Parameter aufzuteilen, um die Konvergenz des EPD-Verfahrens zu erreichen. Es lässt sich allerdings nicht direkt bestimmen, welcher Parameter das Konvergenzverhalten am stärksten beeinflusst. Dazu werden im Folgenden heuristische Aufteilungsstrategien vorgestellt. Sie wählen an Hand verschiedener Kriterien einen Parameter zur Aufteilung aus. Um zu verhindern, dass eine Strategie einen Parameter auswählt, dessen Aufteilung nicht zur Verbesserung des Konvergenzverhaltens führt, wird nach $n_{ss,max}$ Aufteilungen mit der gewählten Aufteilungsstrategie auf das Teilen aller Parameter (Strategie *Select All*) zurückgegriffen. Ist die Lösung mit insgesamt $n_{ts,max}$ Aufteilungen nicht möglich, wird die Lösung abgebrochen (Algorithmus 10). Die Aufteilungsstrategien verwenden die Hilfsfunktion `findMax`, die den Spaltenindex j des betragsmäßig größten Elements $a_{i,j}$ der Matrix \mathbf{A} bzw. den Index j des entsprechenden Elements $\vec{b}_{(j)}$ des Vektors \vec{b} ermittelt.

Die Aufteilungsstrategie *Select By Deviation* wählt den Parameter mit dem betragsmäßig größten Abweichungssymbol aus. Die Strategie *Select By Sensitivity* wählt dagegen den Parameter mit der größten Sensitivität aus. Diese wird aus der Sensitivitätsmatrix $\frac{df}{dp}$ bestimmt. Die beiden vorherigen Ansätze werden in der Strategie *Select By Sensitivity And Deviation* kombiniert. Dabei wird die Sensitivität mit der Größe der Parameterunsicherheiten skaliert. Der Operator \odot bezeichnet die elementweise Multiplikation (Hadamard-Multiplikation). Die Strategien *Select By EPD And PPD* und *Select By EPD And Sensitivity* verwenden im Gegensatz zu den bisher vorgestellten Strategien die EPD-Symbole, die während des Lösungsvorgangs eingefügt und angepasst werden. Beide wählen zunächst die Gleichung i mit dem betragsmäßig größten EPD-Symbol aus. Da diese Gleichung in der Regel von mehreren Parametern abhängt, wird der aufzuteilende Parameter auf zwei verschiedene Arten bestimmt. Die Strategie *Select By EPD And PPD* wählt den Parameter anhand des betragsmäßig größten PPD-Symbols in der entsprechenden Variablen $\vec{x}_{(i)}$. Bei der Strategie *Select By EPD And Sensitivity* wird dagegen der Parameter mit der größten Sensitivität gemäß $\frac{df_i}{dp}$ ausgewählt. Eine Übersicht der Strategien zur Auswahl eines aufzuteilenden Parameters ist in Tabelle 3.1 dargestellt.

Algorithmus 10 Auswahl des aufzuteilenden Parameters, selectByX mit $X \in \{\text{Deviation, Sensitivity, SensitivityAndDeviation, EPDAndPPD, EPDAndSensitivity}\}$

```

1: procedure SELECTPARAMETERS
2:    $n_{\text{splits}} = n_{\text{splits}} + 1$ 
3:   if  $n_{\text{splits}} < n_{\text{ss,max}}$  then
4:     return SELECTBYX()
5:   else if  $n_{\text{splits}} < n_{\text{ts,max}}$  then
6:     return SELECTALL()
7:   else
8:     return Failed
9:   end if
10: end procedure

```

Tabelle 3.1: Strategien zur Auswahl des aufzuteilenden Parameters

<i>Select By Deviation</i>	$\vec{d} = [\vec{p}_{(1)1} \dots \vec{p}_{(n)n}]$ return FINDMAX(\vec{d})
<i>Select By Sensitivity</i>	return FINDMAX($\frac{df}{dp}$)
<i>Select By Sensitivity And Deviation</i>	$D = \begin{bmatrix} \vec{p}_{(1)1} & \dots & \vec{p}_{(n)n} \\ \vdots & \vdots & \vdots \\ \vec{p}_{(1)1} & \dots & \vec{p}_{(n)n} \end{bmatrix}$ $A = D \odot \frac{df}{dp}$ return FINDMAX(A) [H]
<i>Select By EPD And PPD</i>	$\vec{e} = [\vec{x}_{(1)EPD} \dots \vec{x}_{(n)EPD}]$ $i = \text{FINDMAX}(\vec{e})$ $\vec{p} = [\vec{x}_{(i)PPD_0} \dots \vec{x}_{(i)PPD_{n_p}}]$ return FINDMAX(\vec{p})
<i>Select By EPD And Sensitivity</i>	$\vec{e} = [\vec{x}_{(1)EPD} \dots \vec{x}_{(n)EPD}]$ $i = \text{FINDMAX}(\vec{e})$ $\vec{b} = \left[\frac{df_i}{dp_1} \dots \frac{df_i}{dp_{n_p}} \right]$ return FINDMAX(\vec{b})

3.7 Extraktion von affinwertigen Eigenschaftsgrößen

Affine Arithmetik lässt sich außer zur Simulation von Schaltungen auch zur Systemsimulation einsetzen. Dabei ist es vorteilhaft, wenn die Korrelationen auf Schaltungsebene auf die Systemebene überführt werden können, damit auch auf Systemebene die Elimination von Abweichungssymbolen stattfinden kann. Dies führt auch auf Systemebene

zur Verringerung der Überschätzung [112]. Die Korrelationen bleiben erhalten, wenn die Extraktion von Systemparametern aus den Ergebnissen der affinwertigen Schaltungssimulation durchgeführt wird.

Beispielhaft ist im Folgenden die Extraktion der Verstärkung und der Grenzfrequenz eines Tiefpasses dargestellt. Dabei wird angenommen, dass der Tiefpass erster Ordnung ist, das heißt seine Ausgangsspannung U_{out} lässt sich durch

$$U_{\text{out}}(t) = U_{\text{end}} \left(1 - e^{-\frac{t}{\tau}}\right) \quad (3.13)$$

beschreiben. Die Spannung $U_{\text{end}} = U_{\text{out}}(t \rightarrow \infty)$ beschreibt dabei die Ausgangsspannung im eingeschwungenen Zustand. Mit Hilfe der affinwertigen Schaltungssimulation lässt sich die affinwertige Ausgangsspannung \hat{U}_{out} zu zwei Zeitpunkten t_1 und t_{end} als Sprungantwort auf die Eingangsgröße

$$U_{\text{in}}(t) = c \cdot \sigma(t) = \begin{cases} 0 & t < 0 \\ c & t \geq 0 \end{cases} \quad (3.14)$$

bestimmen. Die Konstante c gibt dabei die Sprunghöhe an. Der Zeitpunkt t_1 muss im Bereich des exponentiellen Anstiegs liegen, der zweite Zeitpunkt t_{end} dagegen im eingeschwungenen Zustand $t_{\text{end}} \approx t \rightarrow \infty$. Um eine möglichst geringe Anzahl von Abweichungssymbolen für $\hat{U}_{\text{out}}(t \rightarrow \infty)$ zu erreichen, sollte dieser Wert statt mit einer affinwertigen transienten mit Hilfe einer affinwertigen DC-Simulation bestimmt werden. Die Zeitpunkte sind beispielhaft in Abbildung 3.13 dargestellt. Damit lassen sich die affinwertige Verstärkung \hat{A} und die Grenzfrequenz $\hat{f}_g = \frac{1}{2\pi\tau}$ berechnen:

$$\hat{A} = \frac{\hat{U}_{\text{out}}(t_{\text{end}})}{\hat{U}_{\text{in}}(t_{\text{end}})} \quad (3.15)$$

$$\hat{f}_g = -\frac{\ln\left(-\frac{\hat{U}_{\text{out}}(t_1)}{\hat{U}_{\text{out}}(t_{\text{end}})} + 1\right)}{2\pi t_1}. \quad (3.16)$$

Für die in Abbildung 3.14 gezeigte Tiefpassschaltung und den dort angegebenen Bau-

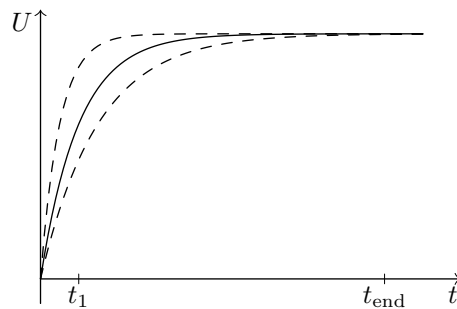


Abbildung 3.13: Zeitpunkt t_1 und t_{end} zur Extraktion von affinwertigen Eigenschaftsgrößen eines Tiefpasses

3 Erweitertes EPD-Verfahren

teilparametern ergibt für die Extraktion zum Zeitpunkt $t_1 = 0,6 \mu\text{s}$:

$$\hat{A} = 1 \pm -4,0157 \cdot 10^{-8} \varepsilon_1 \pm -1,99589 \cdot 10^{-108} \varepsilon_2 \pm -4,32424 \cdot 10^{-8} \varepsilon_3 \pm -7,15968 \cdot 10^{-21} \varepsilon_4 \pm 2,25268 \cdot 10^{-20} \varepsilon_{14} \pm \dots \quad (3.17)$$

$$\hat{f}_g = 34033,5 \pm -3608,39 \varepsilon_1 \pm -2,21021 \cdot 10^{-94} \varepsilon_2 \pm -3608,83 \varepsilon_3 \pm -7,24556 \cdot 10^{-7} \varepsilon_4 \pm 2,2797 \cdot 10^{-6} \varepsilon_{14} \pm \dots \quad (3.18)$$

Abweichungsterme mit einem Index größer als 14 besitzen nur sehr geringe partielle Abweichungen und sind daher hier nicht dargestellt. Als Intervall (berechnet aus allen Abweichungstermen) dargestellt ergibt sich:

$$[A] = [0,99999, 1] \quad (3.19)$$

$$[f_g] = [23943,4 \text{ Hz}, 44123,6 \text{ Hz}]. \quad (3.20)$$

Der Einschluss der aus dem Minimum und Maximum der Parameter berechnete Werte $A = 1$ und $f_{g,min} = 26306,6 \text{ Hz}$ bzw. $f_{g,max} = 39297,5 \text{ Hz}$ ist also gegeben. Die Extraktion weiterer affinwertiger Systemparameter aus der affinwertigen Schaltungssimulation lässt sich analog durchführen.

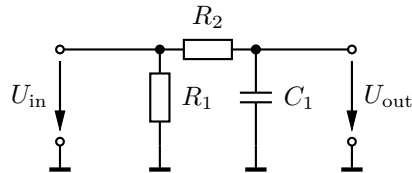


Abbildung 3.14: Tiefpassschaltung, an der beispielhaft die Extraktion von affinwertigen Eigenschaftsgrößen durchgeführt wird. $C_1 = 0,5 \text{ nF} \pm 50 \text{ pF} \varepsilon_1$, $R_1 = 10 \text{ k}\Omega \pm 1 \text{ k}\Omega \varepsilon_2$, $R_2 = 10 \text{ k}\Omega \pm 1 \text{ k}\Omega \varepsilon_3$

4 Ergebnisse

Im Rahmen dieser Arbeit wurde der Simulator *AGIAS* (A General Interval Arithmetic Simulator) implementiert. Er implementiert neben dem EPD-Verfahren aus [27] auch das erweiterte EPD-Verfahren aus Kapitel 3. Zur Erzeugung der Schaltungsgleichungen wird die Erweiterung von *AMV* (Analoge Maple-Verhaltensbeschreibung) verwendet, die in [27] für den Simulator *nonlineqs* entwickelt wurde. Die Erzeugung der Netzliste sowie die dafür benötigten Bauteilmodelle mit ihren gegebenenfalls mit Unsicherheiten behafteten Parametern sind in Maple implementiert. Der Simulationskern von *AGIAS* ist in der Programmiersprache C++ unter Verwendung der *aaf_lib* (<https://sourceforge.net/projects/aaf-lib/>) zur Berechnung der affinen Formen implementiert. Zur Verwaltung der Abhängigkeiten zwischen dem Maple- und dem C++-Teil wird ein *SCons*-basiertes Buildsystem verwendet. Diese Architektur ist grafisch in Abbildung 4.1 dargestellt. Die im Folgenden gezeigten Simulationsläufe wurden auf einem PC mit einem Intel Core-i7-Prozessor (Modell 2600K, Sandy-Bridge-Generation) und 16 GB RAM durchgeführt. Die maximale Taktfrequenz des Prozessors liegt bei 3,4 GHz bzw. 3,8 GHz wenn *Turbo Boost* aktiv ist. Das verwendete Betriebssystem ist die Linux-Distribution Fedora. Der Simulator verwendet keine parallelisierten Algorithmen, so dass von den insgesamt acht Prozessorkernen (vier physikalische plus vier durch Hyperthreading) nur ein Prozessorkern genutzt wird. Bei den folgenden Zeitmessungen wird ausschließlich die Laufzeit des C++-Teils gemessen. Dazu werden r_{aff} affine Simulationen durchgeführt und das Minimum aus ihnen berechnet um eine untere Schranke für die Laufzeit zu bestimmen und den Einfluss von Effekten wie der dynamischen Prozessortaktanpassungen zu vermeiden. Die Zeitmessungen aus [87] sind nicht direkt mit denen im Folgenden gezeigten Ergebnissen vergleichbar, da dort eine Implementierung in MATLAB verwendet wurde.

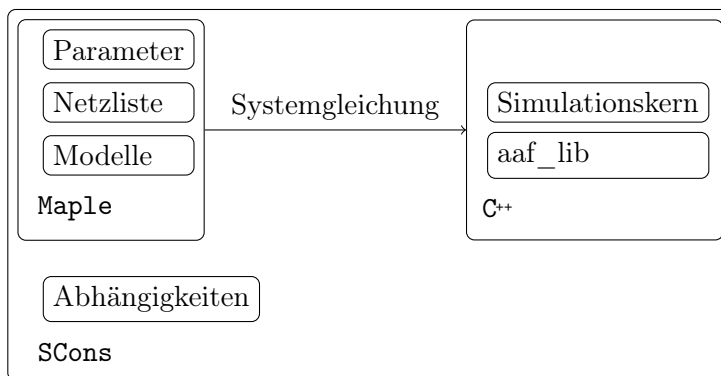


Abbildung 4.1: Darstellung der Simulatorarchitektur

Grundsätzlich lassen sich drei verschiedene Optimierungsziele für die vorgestellten Aufteilungs- und Zusammenfassungsstrategien unterscheiden: die Minimierung der Laufzeit, die Minimierung des Speicherbedarfs und die Minimierung der Überschätzung. Der Speicherverbrauch wird indirekt über die Anzahl der Aufteilungen gemessen: je höher die Anzahl der Aufteilungen, desto höher der Speicherverbrauch. Diese drei Ziele widersprechen sich im Allgemeinen. Das Hauptziel dieser Arbeit ist die Minimierung der benötigten Laufzeit.

4.1 Beispielschaltungen

Die Eigenschaften des EPD-Verfahrens und der vorgestellten Aufteilungs- und Zusammenfassungsstrategien sind stark von dem zu lösenden Gleichungssystem und damit der simulierten Schaltung abhängig. Daher werden im Folgenden zunächst einige Beispielschaltungen vorgestellt, an Hand derer im weiteren Verlauf die Eigenschaften analysiert werden. Die Beispielschaltungen wurden wegen der steigenden Anzahl der enthaltenen Bauelemente und der Linearitäts- bzw. Nichtlinearitätseigenschaften ausgewählt. Sie enthalten ausschließlich Bipolartransistoren, da alle Betriebsbereiche eines Bipolartransistors mit einer Gleichung beschrieben werden können [18]. Einfache MOSFET-Modelle wie das von Shichman und Hodges [88] verwenden dagegen abschnittsweise definierte Gleichungen. Abgesehen davon ist die Modellkomplexität mit der des Bipolartransistormodells vergleichbar. Die abschnittsweise Definition ist in Verbindung mit affiner Arithmetik problematisch, da durch die Unsicherheiten ein unscharfes Umschalten der Gleichungen erforderlich ist. Ansätze dazu finden sich in [26]. Diese werden jedoch im Rahmen dieser Arbeit nicht weiterverfolgt. MOSFET-Modelle wie das BSIM4- [72] oder das EKV-Modell [19] verwenden eine kontinuierliche Gleichung zur Modellierung aller Arbeitsbereiche. Diese ist jedoch komplexer als bei Modellen mit einer abschnittswisen Definition. Die damit verbundene Überschätzung verhindert ihren Einsatz mit affiner Arithmetik.

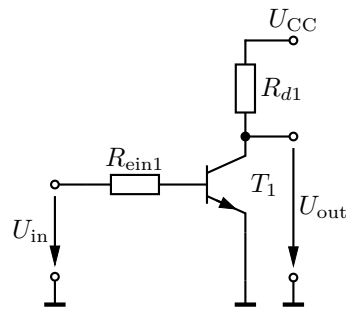
Die in den Beispielschaltungen verwendeten Operationsverstärker werden als ideal modelliert, es handelt sich also um lineare Schaltungen. Eine Übersicht über alle Beispielschaltungen, die Anzahl der enthaltenen Bauelemente n_e und die Anzahl der Schaltungsgleichungen n_x ist in Tabelle 4.1 dargestellt. Die Anzahl der affinen Parameter n_p ist als Maximum über alle Untersuchungen angegeben, in einigen Fällen wird eine geringere Anzahl angenommen. Diese Tatsache, die Größe der verwendeten Parameterunsicherheiten und weitere Simulatoreinstellungen für die verschiedenen Untersuchungen werden zusammenfassend in Anhang A angegeben.

Tabelle 4.1: Übersicht über die verwendeten Beispielschaltungen

Beispiel	Typ	n_e	n_p	n_x
BJT_INV	nichtlinear	3	2	7
BP	linear	8	7	11
DIFF_AMP	nichtlinear	8	5	14
NLTL ₃	nichtlinear	12	3	23
SVF	linear	12	9	15

Inverter

Die einfachste Schaltung mit einem Transistor ist der Inverter. Er wird im Folgenden mit BJT_INV bezeichnet und ist in Abbildung 4.2 dargestellt. In Experimenten hat sich insbesondere der Arbeitspunkt $U_{in} = 0,49 \text{ V}$ als Herausforderung für die Simulation mit affiner Arithmetik erwiesen. In diesem Arbeitspunkt werden bei zunehmender Vergrößerung der Parameterunsicherheiten bereits Aufteilungen benötigt, während sich alle anderen Arbeitspunkte noch ohne Aufteilung lösen lassen.

**Abbildung 4.2:** Inverter mit Bipolartransistor, $U_{CC} = 5 \text{ V}$

NLTL

Eine nichtlineare Übertragungsstrecke (engl. NLTL) kann durch die Ersatzschaltung in Abbildung 4.3 modelliert werden. Dabei wird das lineare Ersatzschaltbild einer Übertragungsstrecke bestehend aus Widerstand, Induktivität und Kapazität gemäß [108, Figure 1] um eine nichtlineare Diode erweitert.

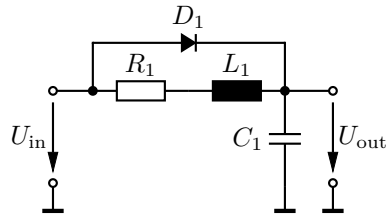


Abbildung 4.3: NLTL nach [108, Figure 1]

Bandpass

In Abbildung 4.4 ist ein aktiver Bandpass mit Einfachmitkopplung dargestellt. Er enthält einen idealen Operationsverstärker. Für $C_1 = C_2$, $R_1 = R_2 = R$, $R_4 = (k - 1)R_5$ und $R_3 = 2R$ gilt:

$$\text{Resonanzfrequenz } f_r = \frac{1}{2\pi RC} \quad (4.1)$$

$$\text{Verstärkung } A_r = \frac{k}{k - 3} \quad (4.2)$$

$$\text{Güte } Q = \frac{1}{3 - k} \quad (4.3)$$

Diese Schaltung wird im Folgenden mit BP abgekürzt.

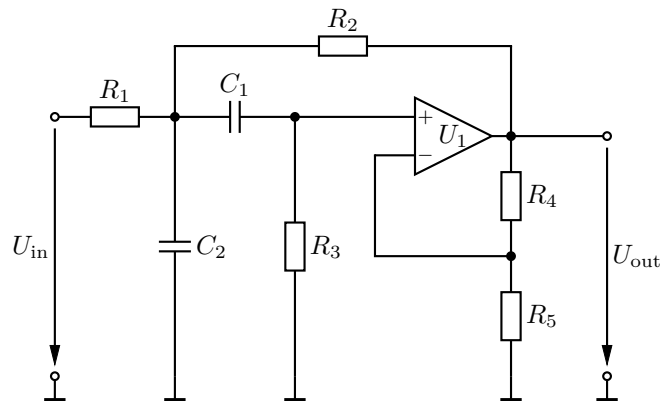


Abbildung 4.4: Aktiver Bandpass nach [103, Abbildung 13.33]

Differenzverstärker

Eine etwas komplexere Schaltung ist der Differenzverstärker in Abbildung 4.5. Er enthält drei Transistoren. Transistor T_3 ist als Stromquelle geschaltet. Die Schaltung wird im Folgenden mit DIFF_AMP abgekürzt.

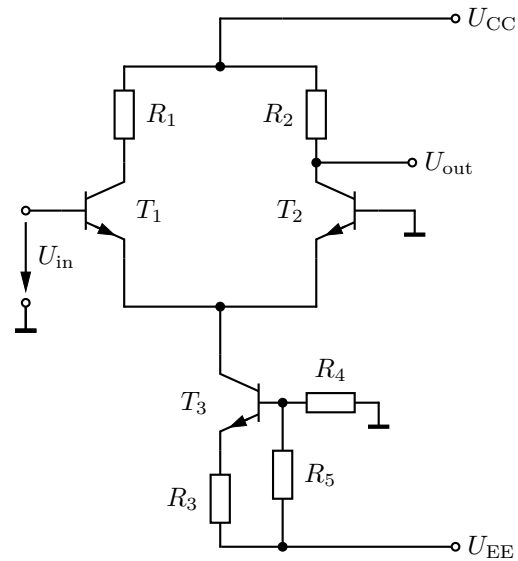


Abbildung 4.5: Differenzverstärker nach [38, Abbildung 2.70], $U_{CC} = 15\text{ V}$, $U_{EE} = -15\text{ V}$

State Variable Filter

Das State Variable Filter (auch Universalfilter genannt) ist eine konfigurierbare Filterarchitektur. Je nach Dimensionierung der Bauteile und der Wahl des Abgriffs U_{LP} , U_{HP} , U_{BP} kann ein Tiefpass-, Hochpass- oder Bandpassverhalten erzielt werden. Für die Berechnung der Übersetzung wird im Folgenden immer der Abgriff $U_{out} = U_{BP}$ verwendet. Dies vereinfacht die Auswertung, da lediglich ein Wert für die Übersetzung m_{OA} berechnet werden muss. Für $R_1 = R_2$, $R_{f1} = R_{f2}$ und $C_1 = C_2$ gilt für die Verstärkung A und die Güte Q :

$$Q = \left(1 + \frac{R_4}{R_q}\right) \left(\frac{1}{2 + \frac{R_1}{R_g}}\right) \quad (4.4)$$

$$A = \frac{R_1}{R_g}. \quad (4.5)$$

Die Schaltung wird im Folgenden mit SVF abgekürzt.

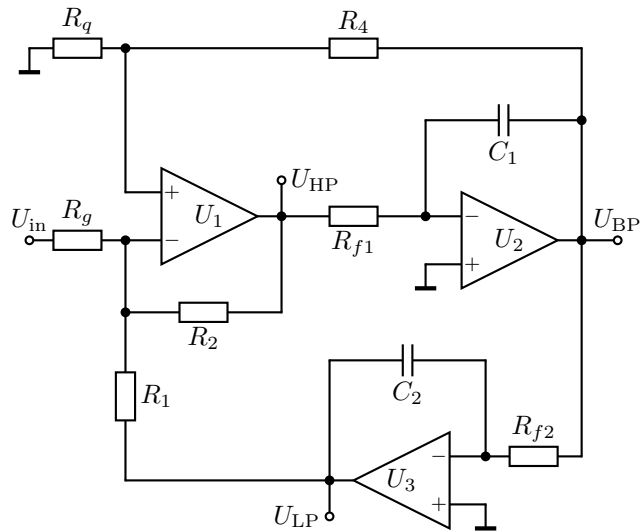


Abbildung 4.6: State Variable Filter nach [38, Abbildung 5.18]

Kettenschaltungen

Um die Abhängigkeit der Eigenschaften der Algorithmen von der Schaltungsgröße zu untersuchen, ist es nützlich, die Anzahl der Bauelemente variieren zu können. Dazu werden Kettenschaltungen aus n gleichen Grundelementen verwendet (Abbildung 4.7). Als Grundelemente werden die NLTL nach Abbildung 4.3 und der Inverter nach Abbildung 4.2 verwendet. Die entsprechenden Kettenschaltungen werden in den Abbildungen 4.8 und 4.9 dargestellt. BJT_INV_n bzw. NLTL_n bezeichnen die Schaltungen mit jeweils n Gliedern.

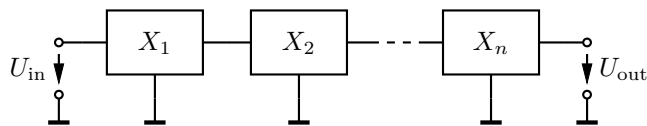


Abbildung 4.7: Kettenschaltung aus n Elementen

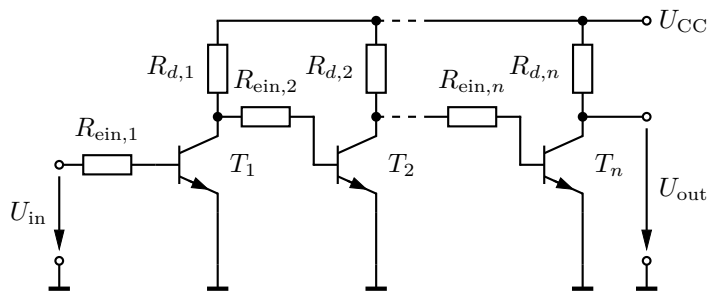
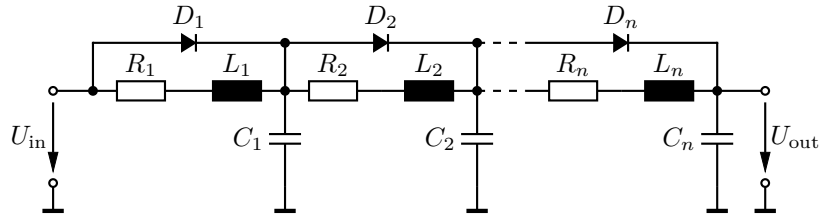


Abbildung 4.8: Inverterkette aus n Invertern (BJT_INV_n), $U_{CC} = 5\text{ V}$

Abbildung 4.9: Kettenschaltung aus n NLTLs (NLTL $_n$) nach [108, Figure 1]

4.2 Grenzen des EPD-Verfahrens ohne Aufteilungen

Zunächst werden die Grenzen des EPD-Verfahrens ohne Aufteilungen (Algorithmus 2) untersucht. Dazu wird die automatische Exploration des Konvergenzgebiets nach Algorithmus 6 verwendet. Die Ergebnisse sind in Tabelle 4.2 dargestellt. Die maximale relative Unsicherheit δp bezeichnet dabei die maximale Unsicherheit p_1 des Parameters p , für den eine konvergente Lösung berechnet werden kann, bezogen auf seinen Zentralwert p_0 :

$$\delta p = \frac{p_1}{p_0} \cdot 100 \% . \quad (4.6)$$

Die angegebene Reihenfolge der Parameter entspricht der Explorationsreihenfolge von Algorithmus 6. Das Volumen V_c der konvexen Hülle des Gebiets, in dem Algorithmus 2 konvergiert, und die maximalen relativen Unsicherheiten δp sind von der betrachteten Schaltung abhängig. Für die nichtlinearen Schaltungen BJT_INV und DIFF_AMP ergeben sich die geringsten maximalen relativen Unsicherheiten aller Beispiele. Die größte maximale relative Unsicherheit wird für den Parameter L_1 der Schaltung NLTL $_3$, die geringste für den Parameter R_1 der Schaltung DIFF_AMP erreicht.

Tabelle 4.2: Konvergenzgebiet für das EPD-Verfahren ohne Gebietsaufteilungen

Beispiel	V_c	max. relative Unsicherheit
BJT_INV	$6,25 \cdot 10^4 \Omega^2$	$\delta R_d = 5,50 \%$, $\delta R_{ein} = 5,50 \%$
BP	$1,46 \cdot 10^8 \Omega^3$	$\delta R_5 = 22,50 \%$, $\delta R_2 = 15,72 \%$, $\delta R_1 = 9,43 \%$
DIFF_AMP	$7,67 \cdot 10^3 \Omega^3$	$\delta R_1 = 0,80 \%$, $\delta R_2 = 1,47 \%$, $\delta R_3 = 2,00 \%$
NLTL3	$9 \cdot 10^{-11} \Omega F H$	$\delta R_1 = 4,77 \%$, $\delta L_1 = 92,00 \%$, $\delta C_1 = 1,83 \%$
SVF	$1,04 \cdot 10^{-4} \Omega^2 F$	$\delta R_1 = 6,00 \%$, $\delta C_1 = 23,33 \%$, $\delta R_g = 6,00 \%$

4.3 Laufzeitverhalten ohne Gebietsaufteilungen

In Abschnitt 2.4.4 wurde die Laufzeitkomplexität des EPD-Verfahrens theoretisch hergeleitet. Diese Herleitung soll im Folgenden durch Messungen bestätigt werden. Zunächst wird der Einfluss der Anzahl der Parameter n_p untersucht. Dazu werden die affinen Parameter nacheinander in nominale Parameter umgewandelt. Da nominale Parameter bereits von dem Maple-Teil (vgl. Abbildung 4.1) in das Gleichungssystem eingesetzt

werden, verringert sich so die Anzahl der Parameter, die das EPD-Verfahren (im C++-Teil) behandeln muss. Die Reihenfolge der Umwandlung von affin- zu nominalwertigen Parametern entspricht der angegebenen Reihenfolge der Parameter. Die Ergebnisse sind in den Abbildungen 4.10 und 4.11 dargestellt. Die Überschätzung wurde aus jeweils 1000 Monte-Carlo-Stichproben berechnet. Die benötigte Laufzeit steigt mit der Anzahl der Parameter n_p etwa linear an. Dies entspricht der theoretischen Herleitung in Abschnitt 2.4.4. Auch die Überschätzung steigt mit der Anzahl der Parameter tendenziell an. Lokal sind auch kleinere Verringerungen der Überschätzung zu beobachten. Dies liegt an der teilweisen Auslöschung von Abweichungssymbolen der PPDs während der Berechnung. Durch beide Effekte ist die Anzahl der Parameter, die mit dem EPD-Verfahren sinnvoll behandelt werden können, nach oben begrenzt.

Um den Einfluss der Größe der Parameterunsicherheiten auf die Laufzeit zu untersuchen, werden alle PPDs eines Schaltungsbeispiels mit dem Faktor s_d skaliert:

$$\vec{p}_{(PPD)i, \text{skaliert}} = s_d \cdot \vec{p}_{(PPD)i} \quad \forall i \in \{0 \dots n_p\}. \quad (4.7)$$

Die gemessenen Laufzeiten für $0,1 < s_d < 1$ sind in Abbildung 4.12 dargestellt. Die Größe der Parameterunsicherheiten ist so gewählt, dass für $s_d > 1$ keine konvergente Lösung gefunden werden kann. Die Abhängigkeit der Laufzeit von dem Skalierungsfaktor s_d ist linear. Für die nichtlinearen Schaltungen BJT_INV, DIFF_AMP und NLTL₃ ist die Abhängigkeit vom Skalierungsfaktor s_d geringer als für die linearen Schaltungen SVF und BP. Dies liegt daran, dass für die nichtlinearen Schaltungen die Schleife zur Anpassung der EPD-Symbole (Algorithmus 2, Zeilen 10 bis 30) häufig durchlaufen werden muss und daher der Laufzeitvorteil für kleine s_d geringer ausfällt.

Um den Einfluss der Schaltungsgröße auf das Laufzeitverhalten zu untersuchen, werden zwei Kettenschaltungen nach Abbildung 4.7 untersucht. Die Anzahl n der Elemente der Kettenschaltung wird variiert und die benötigte Laufzeit wird gemessen. Durch die Variation von n wird die Anzahl der Schaltungsgleichungen n_x und die Anzahl der Parameter n_p bestimmt. Für die Inverterkette BJT_INV _{n} nach Abbildung 4.8 ist lediglich die Lösung für $n = 1$ möglich. Die Laufzeit beträgt in diesem Fall 0,85 s. Die Ergebnisse für die Kettenschaltung aus n NLTLs (NLTL _{n}) nach Abbildung 4.9 sind in Abbildung 4.13 grafisch dargestellt. In Abbildung 4.13 sind außerdem ein kubisch und ein quartischer Fit dargestellt. Der Maximalwert für die Anzahl n der Elemente unterscheidet sich für die beiden Kettenschaltungen, da für die gewählte Größe der Unsicherheiten eine Lösung mit Algorithmus 2 von NLTL _{n} für $n > 6$ bzw. von BJT_INV _{n} für $n > 1$ nicht möglich ist. Aufgrund der geringen Anzahl der berechenbaren Datenpunkte lässt sich nicht bestimmen, ob die Abhängigkeit kubisch oder quartisch ist.

4.3 Laufzeitverhalten ohne Gebietsaufteilungen

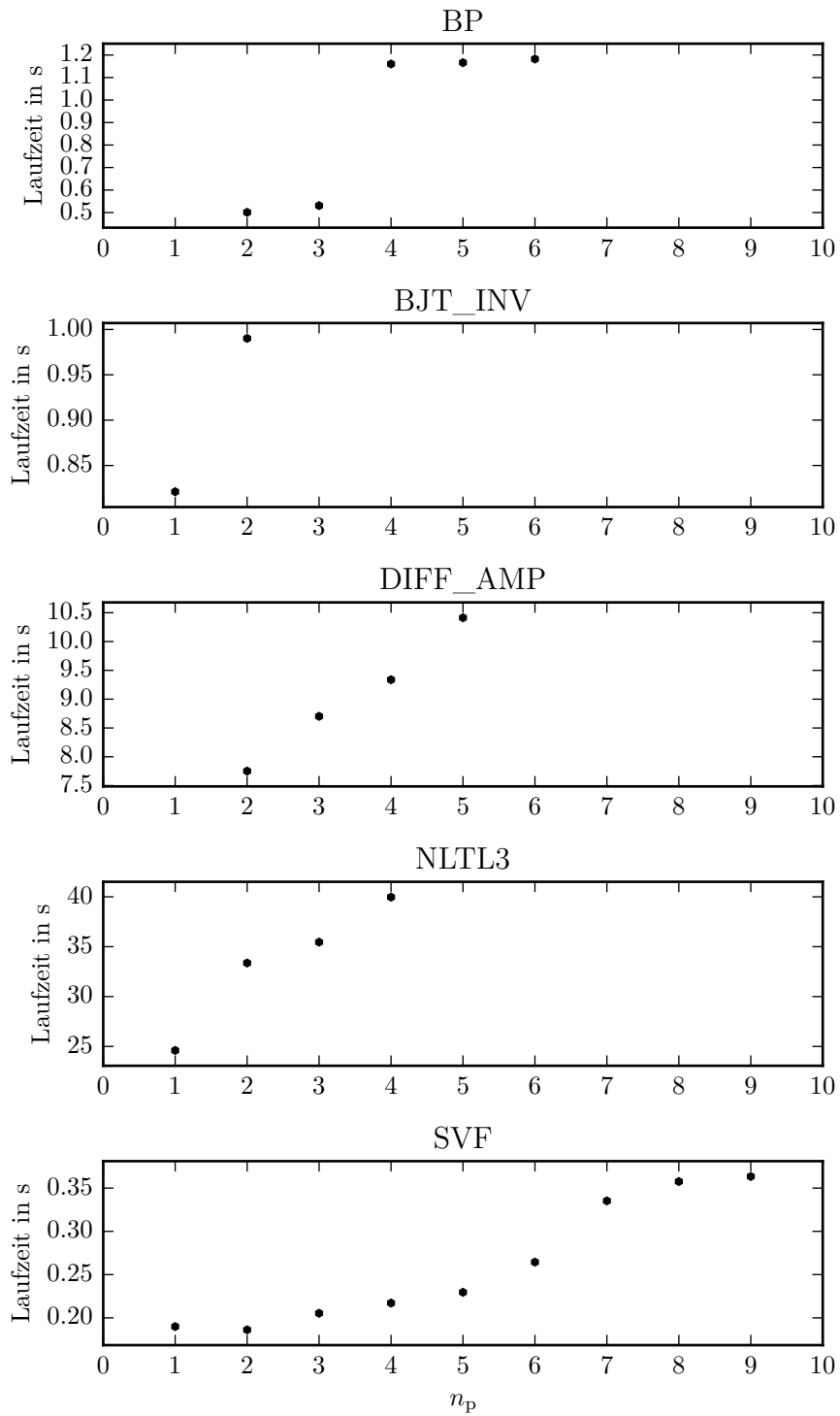


Abbildung 4.10: Laufzeit für verschiedene Werte von n_p

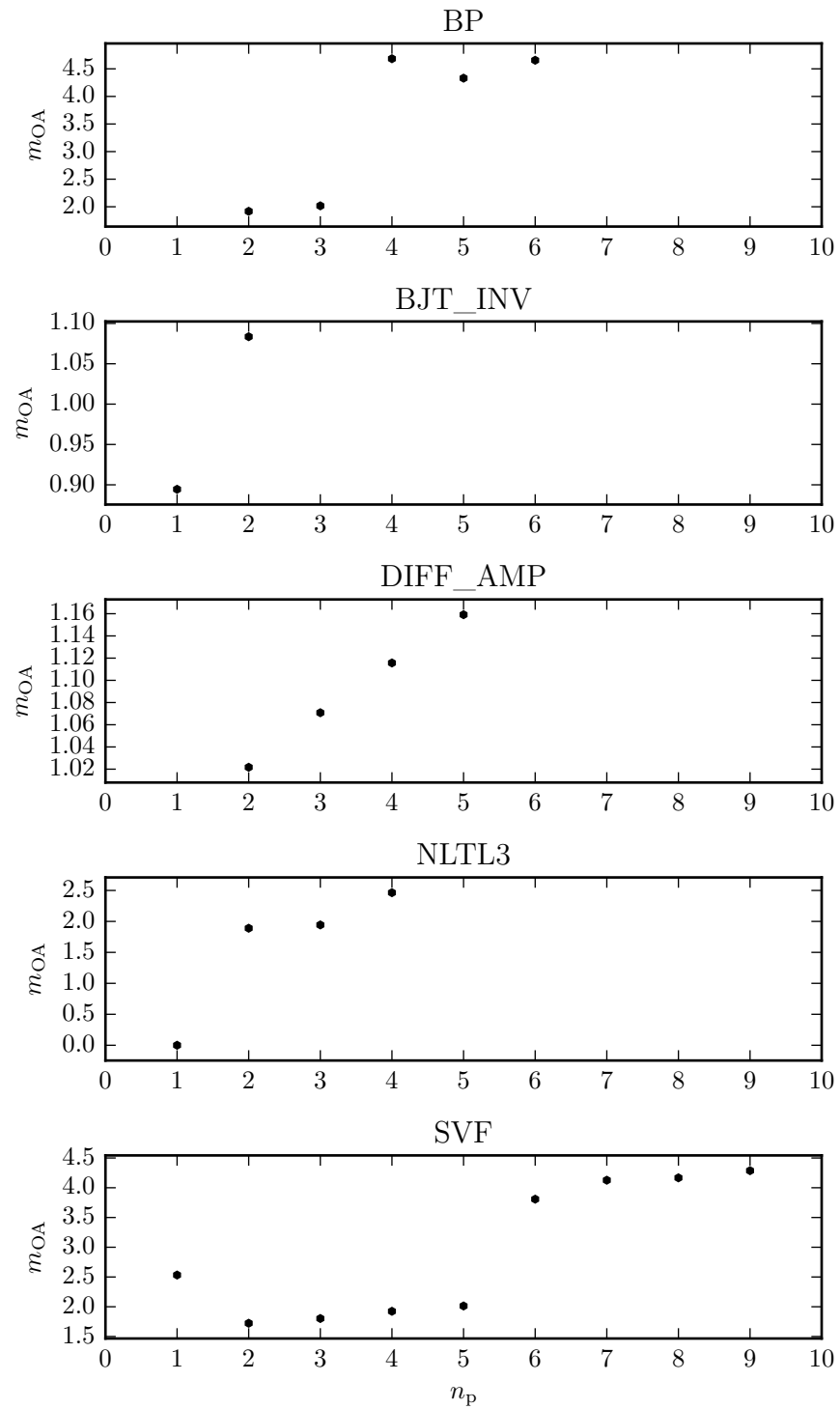


Abbildung 4.11: Überschätzung für verschiedene Werte von n_p

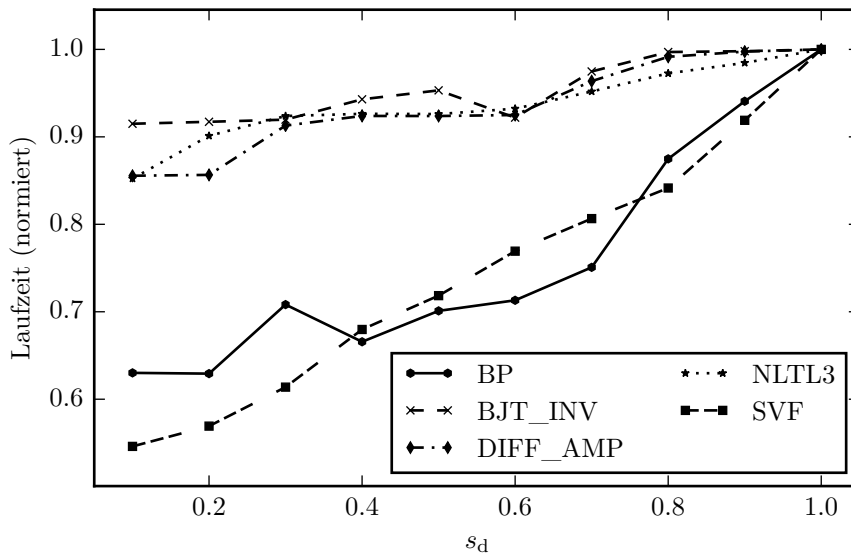


Abbildung 4.12: Laufzeit für verschiedene Werte von s_d

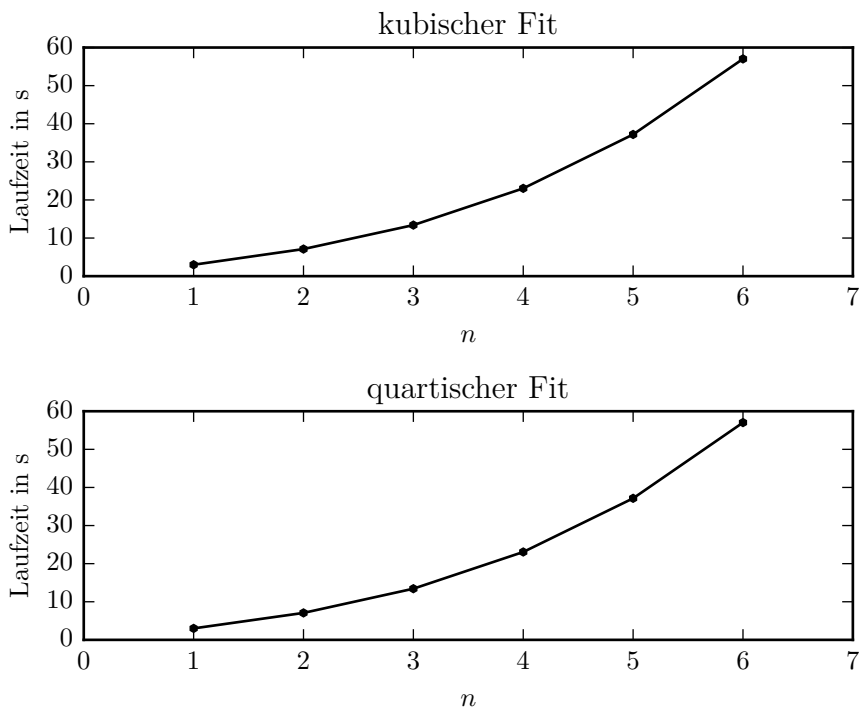


Abbildung 4.13: Laufzeit für verschiedene Werte von n in der Kettenschaltung $NLT3_n$

4.4 Grenzen des EPD-Verfahrens mit Gebietsaufteilungen

Analog zu der Methode aus Abschnitt 4.2 wird das Konvergenzgebiet für das EPD-Verfahren mit Gebietsaufteilungen nach Kapitel 3 bestimmt. Dabei werden die Aufteilungsstrategie *Select All* und die Zusammenfassungsstrategie *Merge Each* verwendet. Die Ergebnisse sind in Tabelle 4.3 dargestellt. Dabei wird deutlich, dass der Einsatz der Gebietsaufteilungen das Konvergenzgebiet des EPD-Verfahrens für alle Schaltungsbeispiele vergrößert. Die kleinste bzw. größte maximale relative Unsicherheit wird, wie in Abschnitt 4.2, für den Parameter R_1 der Schaltung DIFF_AMP bzw. für den Parameter L_1 der Schaltung NLTL₃ erreicht. Ein direkter Vergleich zwischen den Konvergenzgebieten des EPD-Verfahrens mit und ohne Gebietsaufteilung ist in Tabelle 4.4 dargestellt. Dort wird deutlich, dass der relative Beitrag zur Vergrößerung des Konvergenzgebiets stark schaltungsabhängig ist. Er reicht von 10 % für die Schaltung BJT_INV bis zu 444 % für die Schaltung SVF. In der dritten Spalte von Tabelle 4.4 ist die relative Veränderung der Unsicherheiten Δp dargestellt. Für den Parameter p mit seiner Unsicherheit p_1 berechnet sie sich gemäß:

$$\Delta p = 100 \% \cdot \frac{p_{1,\text{mg}}}{p_{1,\text{og}}} - 100 \% \quad (4.8)$$

Dabei bezeichnen $p_{1,\text{og}}$ und $p_{1,\text{mg}}$ die absoluten Unsicherheiten, die ohne bzw. mit Gebietsaufteilungen gemäß Algorithmus 6 bestimmt wurde. Für einige Fälle ergibt sich eine Verkleinerung der relativen Unsicherheiten Δp für einzelne Parameter. Dies tritt für den Parameter R_{ein} im Beispiel BJT_INV, den Parameter L_1 im Beispiel NLTL₃ und den Parameter R_g im Beispiel SVF auf. Dies ist durch die Eigenschaft von Algorithmus 6 bedingt, die Parameterunsicherheiten in der angegebenen Reihenfolge zu erhöhen. Eine andere Explorationsreihenfolge oder ein anderer Algorithmus zur Erhöhung der Parameterunsicherheiten würden auch für die oben genannten Parameter eine Vergrößerung der maximalen Unsicherheit zeigen. Mit einer festen Explorationsreihenfolge lässt sich zeigen, dass das Ziel, das Konvergenzgebiet von Algorithmus 2 durch Gebietsaufteilungen zu vergrößern, erreicht wird. Daher wird auf den Einsatz von anderen Explorationsreihenfolgen oder -algorithmen verzichtet. Die Explorationsreihenfolge bedingt auch, dass die Vergrößerung der Unsicherheit für die Parameter am Beginn der Exploration am größten ist.

Tabelle 4.3: Konvergenzgebiet für das EPD-Verfahren mit Gebietsaufteilungen

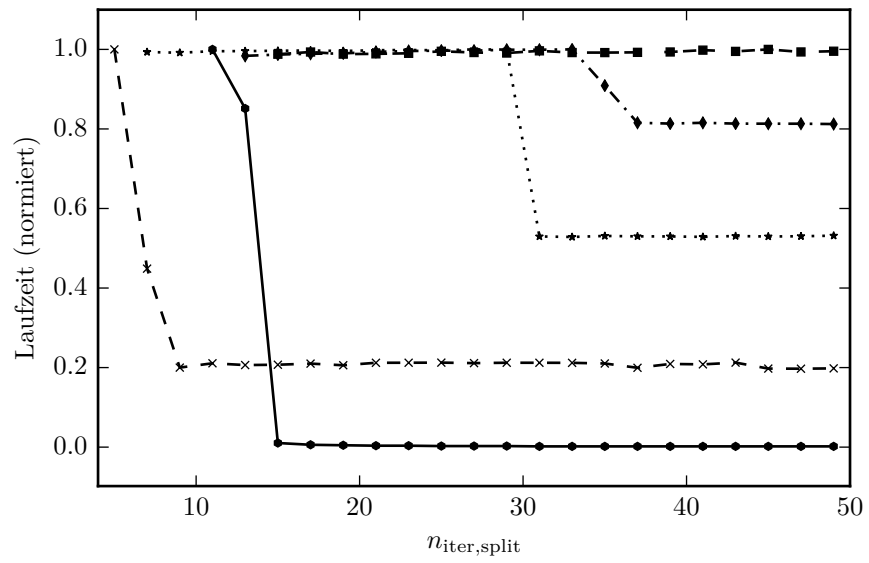
Beispiel	V_c	max. relative Unsicherheit
BJT_INV	$6,88 \cdot 10^4 \Omega^2$	$\delta R_d = 10,00 \%$, $\delta R_{\text{ein}} = 3,50 \%$
BP	$2,71 \cdot 10^8 \Omega^3$	$\delta R_5 = 37,50 \%$, $\delta R_2 = 15,72 \%$, $\delta R_1 = 9,43 \%$
DIFF_AMP	$1,62 \cdot 10^4 \Omega^3$	$\delta R_1 = 1,20 \%$, $\delta R_2 = 1,87 \%$, $\delta R_3 = 2,00 \%$
NLTL3	$1,24 \cdot 10^{-10} \Omega \text{ F H}$	$\delta R_1 = 9,81 \%$, $\delta L_1 = 57,00 \%$, $\delta C_1 = 1,83 \%$
SVF	$5,67 \cdot 10^{-4} \Omega^2 \text{ F}$	$\delta R_1 = 21,00 \%$, $\delta C_1 = 40,00 \%$, $\delta R_g = 5,00 \%$

Tabelle 4.4: Vergleich der Konvergenzgebiete für das EPD-Verfahren mit und ohne Gebietsaufteilungen

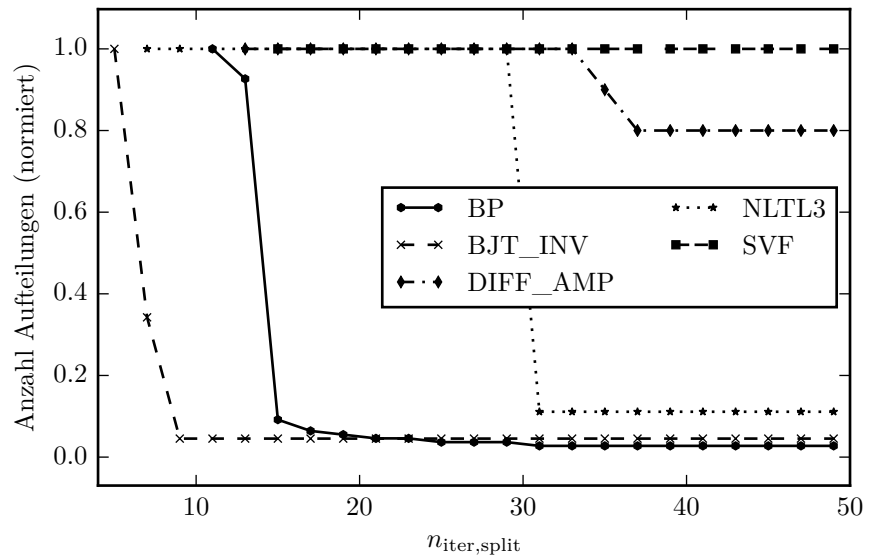
Beispiel	rel. Volumenvergrößerung	rel. Veränderung der maximalen Unsicherheiten
BJT_INV	10,00 %	$\Delta R_d = 81,82 \%$, $\Delta R_{ein} = -36,36 \%$
BP	85,71 %	$\Delta R_5 = 66,67 \%$, $\Delta R_2 = 0,00 \%$, $\Delta R_1 = 0,00 \%$
DIFF_AMP	110,87 %	$\Delta R_1 = 50,00 \%$, $\Delta R_2 = 27,27 \%$, $\Delta R_3 = 0,00 \%$
NLTL3	37,70 %	$\Delta R_1 = 105,40 \%$, $\Delta L_1 = -38,04 \%$, $\Delta C_1 = 0,00 \%$
SVF	444,00 %	$\Delta R_1 = 250,00 \%$, $\Delta C_1 = 71,43 \%$, $\Delta R_g = -16,67 \%$

4.5 Vergleich der Strategien zur automatischen Gebietsaufteilung

In Abschnitt 3.4.1 werden zwei verschiedene Strategien zur automatischen Durchführung von Gebietsaufteilungen vorgestellt. Kriterium 1 verwendet die Anzahl der Iterationen n_{iter} , Kriterium 2 den Anstieg des Skalierungsfaktors Δs . Beide Verfahren basieren auf Vergleichen mit den Schwellwerten $n_{iter,max}$ bzw. Δs_{max} , die vom Benutzer festgelegt werden. Die optimalen Werte für diese Schwellwerte, d.h. die Werte, die die geringste Anzahl Aufteilungen oder die geringste zusätzliche Laufzeit verursachen, lassen sich nicht a-priori bestimmen. Daher werden sie für die Beispielschaltungen experimentell ermittelt. Dazu werden die Schwellwerte $n_{iter,max}$ bzw. Δs_{max} variiert und die zur Simulation benötigte Laufzeit und die Anzahl der dabei durchgeführten Aufteilungen gemessen. Die Ergebnisse für automatische Gebietsaufteilungen ausschließlich nach Kriterium 1 werden in Abbildung 4.14 dargestellt. Die Laufzeit und die Anzahl der Aufteilungen werden auf den jeweiligen Maximalwert je Schaltungsbeispiel normiert angegeben. Es ist erkennbar, dass für alle Schaltungsbeispiele außer SVF die Anzahl der Gebietsaufteilungen n_{splits} und die Laufzeit t sinkt, je größer der Schwellwert $n_{iter,split}$ gewählt wird. Für das Schaltungsbeispiel SVF jedoch bleiben Laufzeit und Anzahl der Aufteilungen über $n_{iter,split}$ näherungsweise konstant. Für die anderen Schaltungsbeispiele sinken Laufzeit und Anzahl der Aufteilungen oberhalb eines bestimmten Werts für $n_{iter,max}$ stark und bleiben danach näherungsweise konstant. Dieser Schwellwert ist von der Schaltung abhängig. Für Werte $n_{iter,split} > 39$ ist jedoch das Minimum der Anzahl der Aufteilungen n_{splits} und der Laufzeit t für alle Schaltungen außer SVF erreicht. Daher wird für die folgenden Untersuchungen $n_{iter,split} = 39$ verwendet. Die Ergebnisse für automatische Gebietsaufteilungen nach Kriterium 2 werden in Abbildung 4.15 dargestellt. Auch hier sind Laufzeit und Anzahl der Aufteilungen auf den jeweiligen Maximalwert je Schaltungsbeispiel normiert. Der Parameter $n_{iter,split}$ wurde zu 300 gewählt. Dadurch wird die Aufteilung nach Kriterium 1 sehr unwahrscheinlich und benötigt mehr Zeit als die Aufteilung nach Kriterium 2. Sowohl die benötigte Laufzeit als auch die Anzahl der Aufteilungen sinken mit steigendem Δs_{max} . Für alle Beispielschaltungen ist das Minimum hinsichtlich Laufzeit und Anzahl der Aufteilungen für $\Delta s_{max} = 0,9 \cdot 10^{-4}$ erreicht. Eine weitere Erhöhung des Parameters Δs_{max} bewirkt keine Verringerung der Laufzeit oder der Anzahl der Aufteilungen mehr.



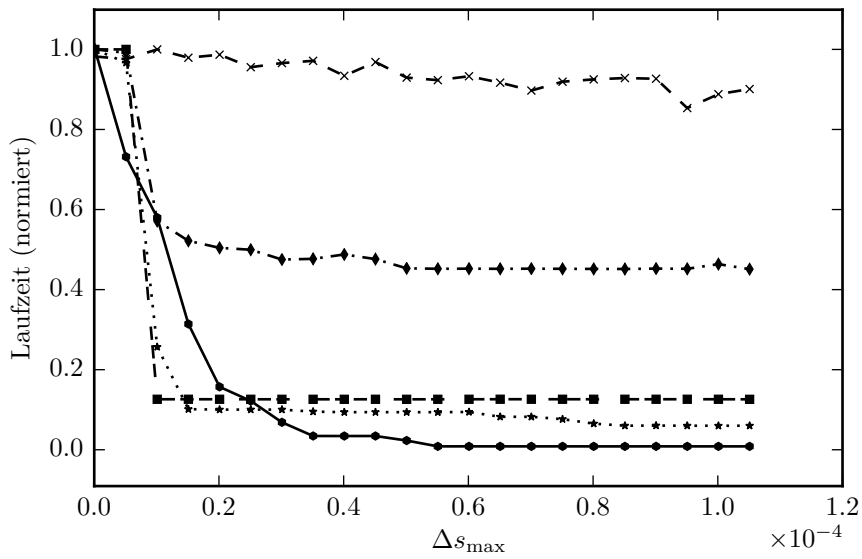
(a) Simulationslaufzeit t



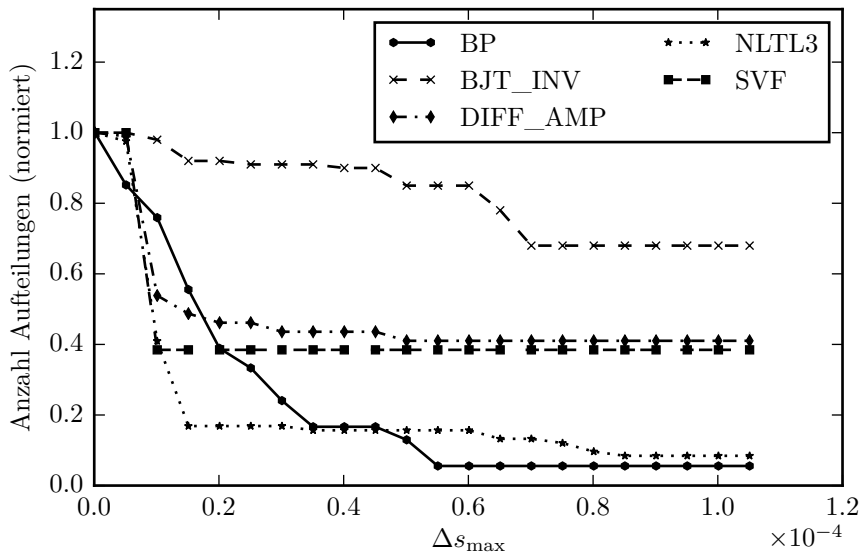
(b) Anzahl der Aufteilungen n_{splits}

Abbildung 4.14: Simulationslaufzeit t und Anzahl der Aufteilungen für automatische Aufteilungen nach Kriterium 1 und verschiedenen Werten von $n_{iter,split}$

4.5 Vergleich der Strategien zur automatischen Gebietsaufteilung



(a) Simulationslaufzeit t



(b) Anzahl der Aufteilungen n_{splits}

Abbildung 4.15: Simulationslaufzeit t und Anzahl der Aufteilungen für automatische Aufteilungen nach Kriterium 2 und verschiedenen Werten von Δs_{\max}

Daher wird dieser Wert im Folgenden verwendet.

Eine direkte Gegenüberstellung der beiden Strategien zur automatischen Gebietsaufteilung wird in Tabelle 4.5 gezeigt. Dabei wird für jedes Beispiel die Anzahl der Aufteilungen und die Laufzeit für die Aufteilung nach Kriterium 1 und Kriterium 2 mit den oben genannten Werten von $n_{\text{iter,max}}$ bzw. Δs_{max} gegenübergestellt. Zusätzlich wird die Kombination der beiden Strategien als *Kriterium 1 + 2* dargestellt. Dabei ist erkennbar, dass das Minimum sowohl der Laufzeit als auch der Anzahl der Aufteilungen für drei Beispiele für die Aufteilung gemäß Kriterium 1 auftreten. Lediglich für die Beispiele DIFF_AMP und BP ergibt bei der gleichen Anzahl von Aufteilungen einen Laufzeitvorteil für Kriterium 1. Die Aufteilung nach Kriterium 1 führt für die Schaltungsbeispiele BP, BJT_INV, DIFF_AMP und NLTL₃ zu der geringsten Laufzeit. Das Schaltungsbeispiel SVF lässt sich dagegen mit der Aufteilung nach Kriterium 2 schneller lösen. Für die Aufteilung gemäß Kriterium 1 wird etwa die 32-fache Laufzeit der Aufteilung gemäß Kriterium 2 benötigt. Die Kombination der beiden Kriterien ergibt für alle Schaltungsbeispiele die schlechtesten Eigenschaften und wird daher im Folgenden nicht mehr betrachtet.

Tabelle 4.5: Vergleich der automatischen Aufteilung nach den Kriterien 1 und 2 sowie ihrer Kombination

Beispiel	Laufzeit in s Kriterium			Anzahl Aufteilungen Kriterium		
	1	2	1 & 2	1	2	1 & 2
BP	7,63	8,01	11,25	3	3	4
BJT_INV	0,90	1,21	1,22	20	68	68
DIFF_AMP	27,98	28,68	29,30	80	80	80
NLTL ₃	17,10	29,61	30,34	1	7	7
SVF	7936,58	250,38	8027,78	26	5	26

4.6 Vergleich der Aufteilungsstrategien

Im Folgenden werden die in Abschnitt 3.6 vorgestellten Aufteilungsstrategien miteinander verglichen. Als Vergleich wird zusätzlich ein zufälliger Parameter zur Teilung ausgewählt. Diese Strategie wird als *Select By Chance* bezeichnet. Alle Schaltungsbeispiele wurden unter Verwendung der verschiedenen Aufteilungsstrategien simuliert. Dabei wird als Zusammenfassungsstrategie für alle Schaltungsbeispiele die Strategie *Merge Each* verwendet. Die Ergebnisse werden hinsichtlich der Laufzeit in Abbildung 4.16, der Anzahl der Aufteilungen in Abbildung 4.17 und hinsichtlich der Überschätzung m_{OA} in Abbildung 4.18 dargestellt. Dabei werden jeweils drei verschiedene Varianten gezeigt:

1. Variante a: $n_{\text{ss,max}} = 10$, $n_{\text{ts,max}} = 10$
2. Variante b: $n_{\text{ss,max}} = 10$, $n_{\text{ts,max}} = 20$
3. Variante c: $n_{\text{ss,max}} = 20$, $n_{\text{ts,max}} = 20$

Der Vergleich der drei Varianten untereinander zeigt, dass die Variante b zu den meisten erfolgreichen Lösungen der Gleichungssysteme führt. Von den insgesamt 35 Möglichkeiten werden mit dieser Variante 29 Möglichkeiten gelöst. In den Varianten a und c können dagegen nur jeweils 18 Möglichkeiten gelöst werden. Die benötigten Laufzeiten für die verschiedenen Varianten sind vergleichbar, wenn eine Lösung mit den übrigen Varianten möglich ist. Die Kombinationen, die mit Variante b lösbar sind, in Variante a oder c jedoch nicht, erfordern in der Regel einen höheren Laufzeitaufwand. Dieser beträgt beim Beispiel NLTL₃ etwa das dreifache, beim Beispiel BP sogar etwa das 90-fache der Zeit verglichen mit der jeweils langsamsten Strategie aus Variante a oder c. Lediglich bei dem Beispiel DIFF_AMP ergibt sich eine Verringerung der Laufzeit gegenüber der langsamsten Variante. Trotzdem ist diese langsamer als die schnellste Strategie gemäß der Varianten a und c. Daher wird im Folgenden nur Variante a verwendet. Die geringste Laufzeit ergibt sich für das Beispiel BJT_INV für die Strategie *Select By Deviation*, für das Beispiel DIFF_AMP für die Strategie *Select By Sensitivity And Deviation*. Für das Beispiel BP führen die beiden genannten Strategien zu einer vergleichbaren Laufzeit, die das Minimum gegenüber den übrigen Varianten darstellt. Das Beispiel NLTL₃ kann dagegen mit den drei Strategien *Select By Deviation*, *Select By Sensitivity* oder *Select By Sensitivity And Deviation* in minimaler Laufzeit gelöst werden. Die geringste Laufzeit für das Beispiel SVF ergibt sich für die Strategie *Select By Deviation*. Die minimale Anzahl der Aufteilungen ergibt sich für BP für die Strategien *Select All*, *Select By Deviation* und *Select By Sensitivity And Deviation*. Für das Beispiel BP ergibt sich das Minimum für die Strategien *Select All*, *Select By Deviation*, *Select By Sensitivity* und *Select By Sensitivity And Deviation*. Für die Beispiele BJT_INV und SVF führen *Select All* und *Select By Deviation* zu der minimalen Anzahl von Aufteilungen. Für DIFF_AMP führt *Select All* zu der minimalen Anzahl. Daraus lässt sich ableiten, dass die Minimierung der Anzahl der Aufteilungen nicht notwendigerweise zu der minimalen Laufzeit führt. Für alle Schaltungsbeispiele außer SVF ist die Überschätzung von der Wahl der Aufteilungsstrategie nahezu unabhängig. Für das Beispiel SVF führt die Wahl der Strategie *Select All* zur geringsten Überschätzung. Im Rahmen dieser Arbeit soll die durch Gebietsaufteilungen verursachte zusätzliche Laufzeit verringert werden. Dies kann in den meisten Fällen durch die Strategie *Select By Deviation* erreicht werden. Lediglich für ein Schaltungsbeispiel (DIFF_AMP) ergibt sich die geringste Laufzeit für *Select By Sensitivity And Deviation*, die Strategie *Select By Deviation* führt hier zu keinem konvergenten Ergebnis. Die Strategie *Select By Chance* gehört stets zu den langsamsten, eine zufällige Auswahl des Aufteilungsparameters ist daher nicht zielführend.

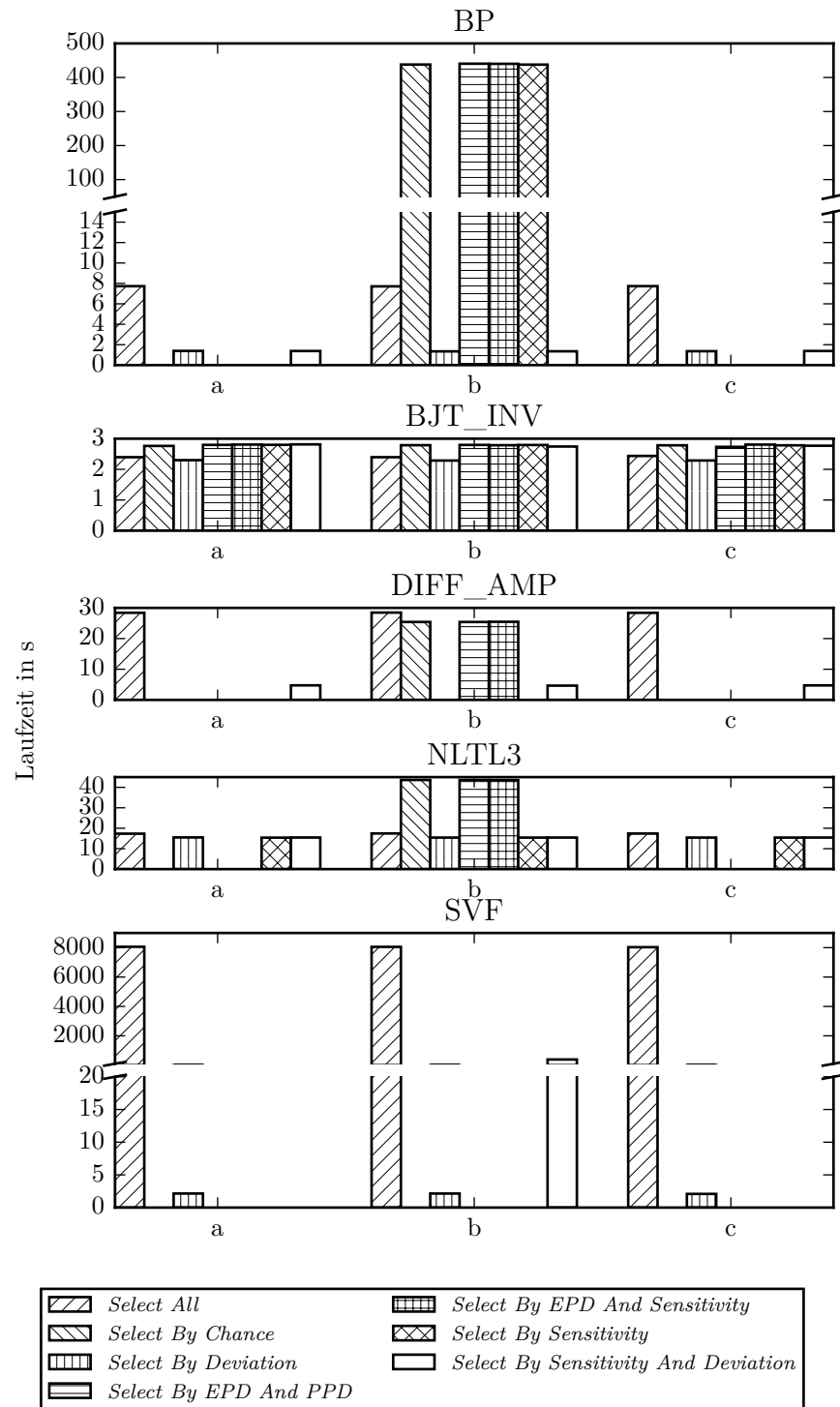


Abbildung 4.16: Vergleich der Laufzeit für verschiedene Aufteilungsstrategien in den Varianten a, b und c

4.6 Vergleich der Aufteilungsstrategien

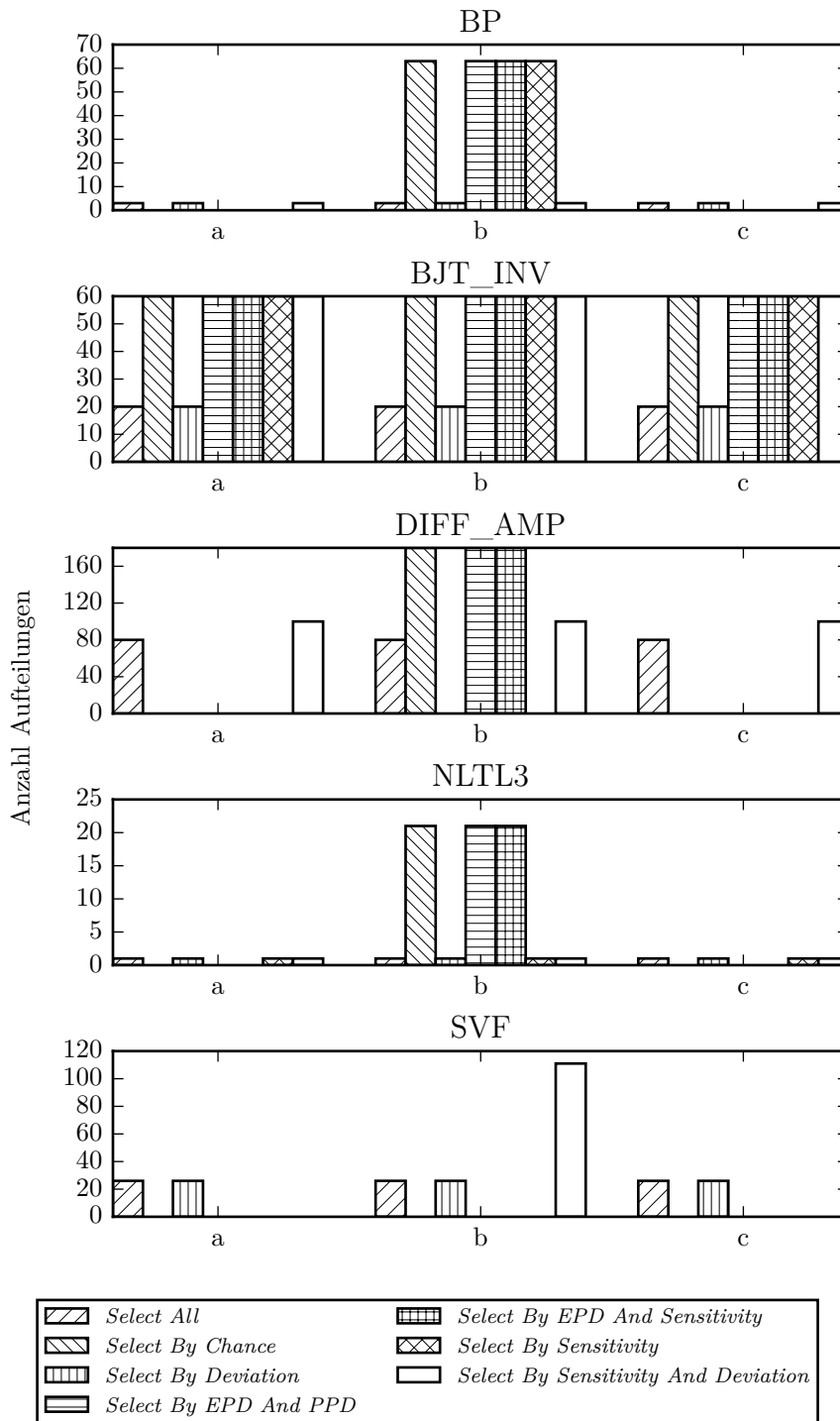


Abbildung 4.17: Vergleich der Anzahl der Aufteilungen n_{splits} für verschiedene Aufteilungsstrategien in den Varianten a, b und c

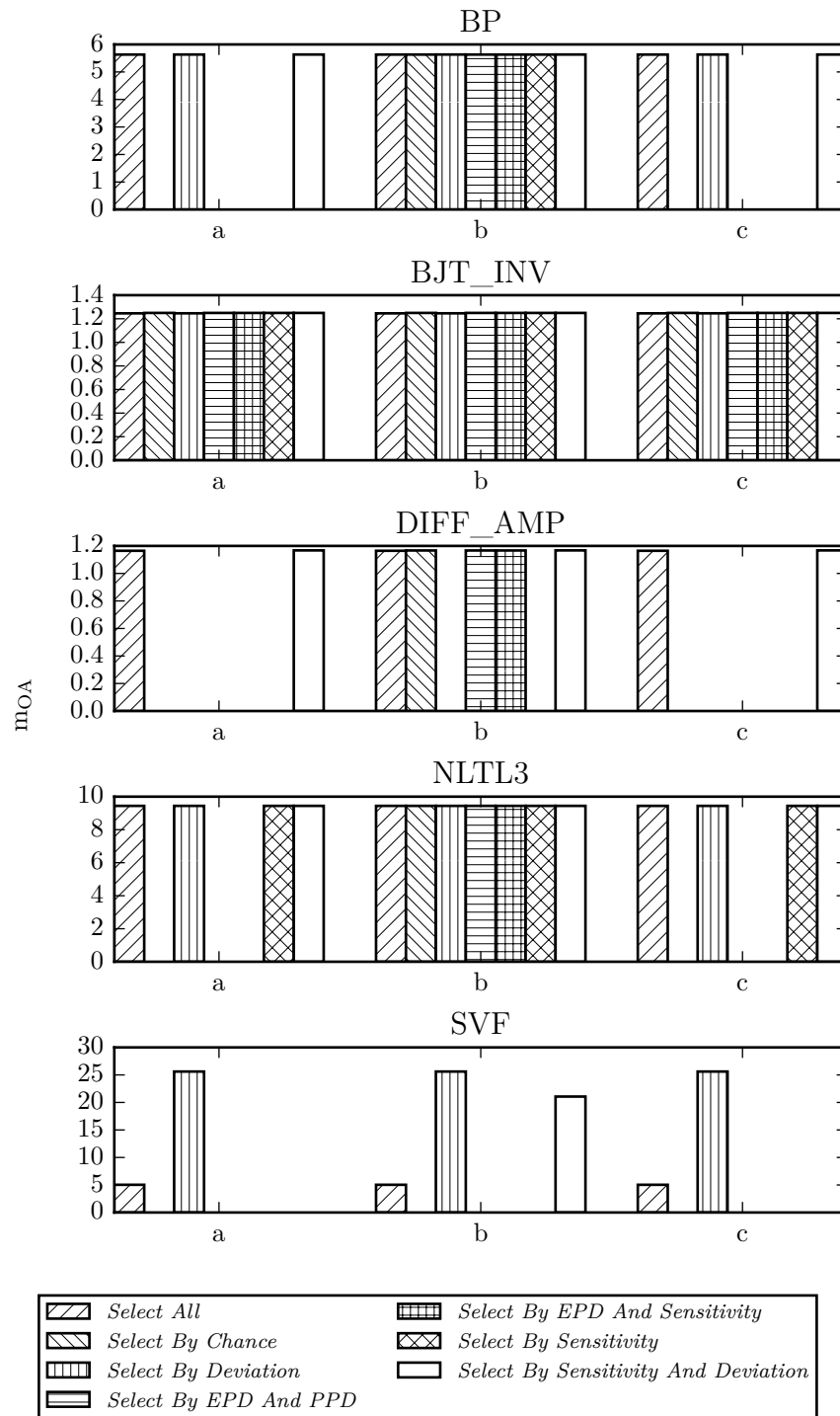


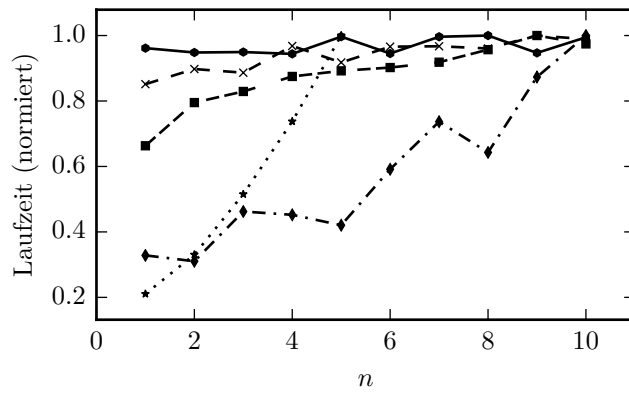
Abbildung 4.18: Vergleich der Überschätzung für verschiedene Aufteilungsstrategien in den Varianten a, b und c

4.7 Ergebnisse für die verschiedenen Zusammenfassungsstrategien

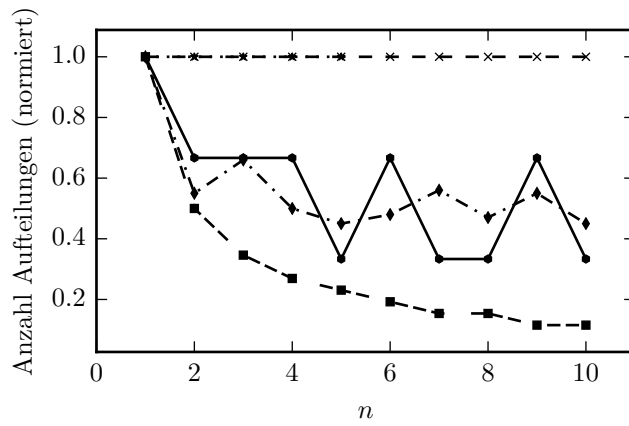
Im Folgenden werden die in Abschnitt 3.5.1 vorgestellten Zusammenfassungsstrategien miteinander verglichen. Als Aufteilungsstrategie wird dabei *Select All* verwendet. Bei der Strategie *Merge n^{th} Step* ist die Anzahl der Schritte n , nach denen eine Zusammenfassung stattfindet, einstellbar. Im Folgenden wird zunächst ein Wert für n gesucht, der für die verschiedenen Schaltungsbeispiele zu einer möglichst geringen Laufzeit führt.

4.7.1 Bestimmung von n für die Zusammenfassungsstrategie *Merge n^{th} Step*

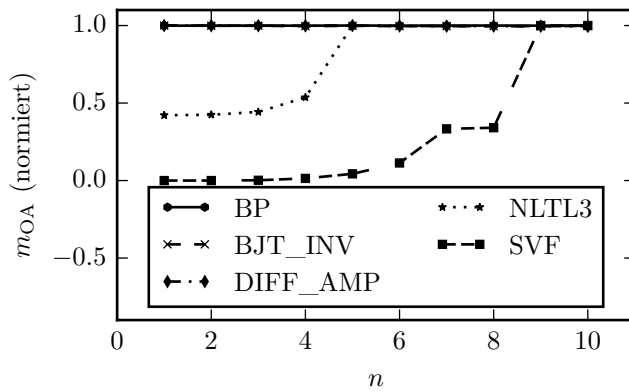
Zur Bestimmung von n für die Zusammenfassungsstrategie *Merge n^{th} Step* wird n variiert und die Laufzeit, die Überschätzung sowie die Anzahl der Aufteilungen werden gemessen. Die Ergebnisse sind in Abbildung 4.19 dargestellt. Die Ergebnisse werden auf den jeweiligen Maximalwert je Schaltungsbeispiel normiert dargestellt. Die Überschätzung m_{OA} wurde aus jeweils 1000 Monte-Carlo-Stichproben berechnet. Aus den Ergebnissen in Abbildung 4.19 ist erkennbar, dass für alle Beispiele ein steigendes n tendenziell zu einer geringeren Anzahl von Aufteilungen führt. Das oszillierende Verhalten des Schaltungsbeispiels BP deutet darauf hin, dass für dieses Gebietsaufteilungen in periodischen Abständen erforderlich sind. Für die Schaltungsbeispiele SVF und NLTL₃ ergibt sich die geringste Laufzeit für $n = 1$. Dies entspricht der Aufteilungsstrategie *Merge Each*, durch die Strategie *Merge n^{th} Step* kann in diesem Fall kein Vorteil erzielt werden. Für die Schaltungsbeispiele BJT_INV und DIFF_AMP ergibt für die geringste Laufzeit für $n = 2$. Das Beispiel BP benötigt im Fall von $n = 4$ noch etwas weniger Laufzeit als für $n = 2$. Der Unterschied ist jedoch vernachlässigbar klein. Die Überschätzung ist für alle Beispiele außer SVF und BJT_INV nahezu konstant über n . Für das Beispiel SVF ergibt sich für $n = 2$ die geringste Überschätzung, so dass hier der Vorteil der geringsten Laufzeit mit der geringsten Überschätzung zusammenfällt. Die geringste Überschätzung ergibt sich für BJT_INV für $n = 8$. Da im Rahmen dieser Arbeit hauptsächlich die zusätzliche Laufzeit durch Gebietsaufteilungen vermieden werden soll, wird im folgenden für alle Beispiele und alle Untersuchungen $n = 2$ verwendet.



(a) Laufzeit



(b) Anzahl Aufteilungen n_{splits}

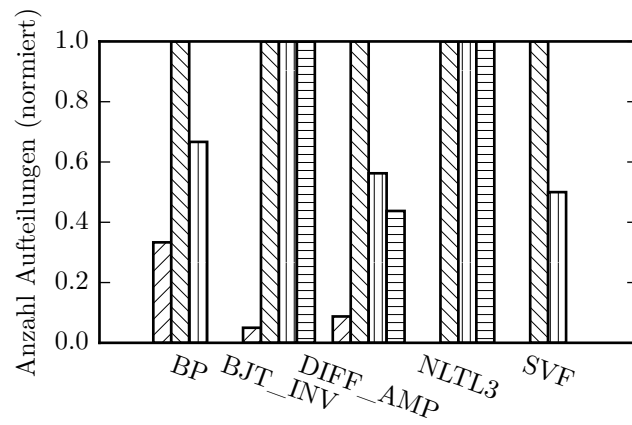


(c) Überschätzung

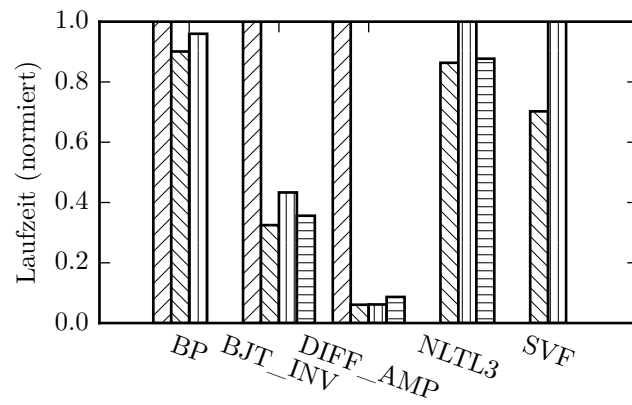
Abbildung 4.19: Exploration von verschiedenen Werten für n für die Zusammenfassungsstrategie *Merge n^{th} Step*

4.7.2 Vergleich der Zusammenfassungsstrategien

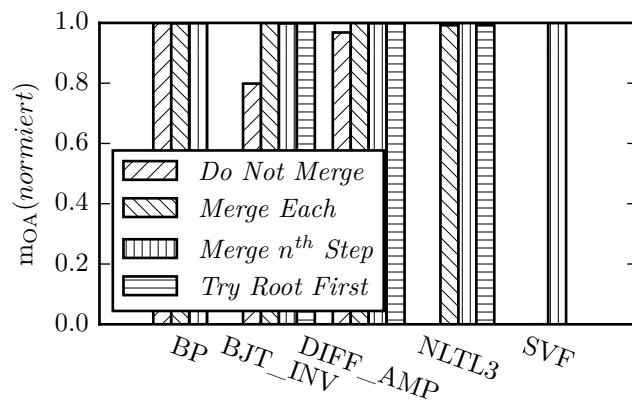
Im Folgenden werden die Eigenschaften der verschiedenen Zusammenfassungsstrategien aus Abschnitt 3.5.1 verglichen. Die Ergebnisse hinsichtlich Laufzeit, Anzahl der Aufteilungen und Überschätzung für die verschiedenen Zusammenfassungsstrategien werden in Abbildung 4.20 dargestellt. Die Überschätzung m_{OA} wird aus jeweils 1000 Monte-Carlo-Stichproben berechnet. Die Ergebnisse werden auf den jeweiligen Maximalwert je Schaltungsbeispiel normiert dargestellt. Die automatische Gebietsaufteilung wird für alle Beispiele gemäß Kriterium 1 nach 40 Iterationen ($n_{\text{iter,max}} = 40$) durchgeführt. Dabei wird die Aufteilungsstrategie *Select All* verwendet, bei der stets alle Parameter aufgeteilt werden. Für die Beispiele DIFF_AMP und BJT_INV führt die Strategie *Do Not Merge* zu der geringsten Anzahl von Aufteilungen. Im Fall BP sind *Do Not Merge* und *Merge n^{th} Step* gleichauf. Für das Beispiel SVF führt die Aufteilungsstrategie *Merge n^{th} Step* zur geringsten Anzahl der Aufteilungen. Für das Beispiel NLTL₃ sind die Strategien *Merge Each*, *Merge n^{th} Step* und *Try Root First* hinsichtlich der Anzahl der Aufteilungen gleich. Hinsichtlich der benötigten Laufzeit stellt die Strategie *Merge Each* für alle Beispielschaltungen das Optimum dar. Für die Schaltungsbeispiele BJT_INV und DIFF_AMP ergibt sich die geringste Überschätzung für die Strategie *Do Not Merge*. Hier kann also eine Abwägung von Laufzeit gegen Überschätzung durch die Wahl der Zusammenfassungsstrategie durchgeführt werden. Für die übrigen Beispiele ist die Überschätzung unabhängig von der Wahl der Zusammenfassungsstrategie. Die Strategien, für die kein Balken dargestellt ist, haben zu mehr als der vorgegebenen Anzahl von Aufteilungen ($n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 8$) geführt und werden daher als nicht konvergent betrachtet.



(a) Anzahl Aufteilungen n_{splits}



(b) Laufzeit



(c) Überschätzung

Abbildung 4.20: Vergleich der Zusammenfassungsstrategien

4.8 Kombination der Aufteilungs- und Zusammenfassungsstrategien

In den beiden vorangegangenen Abschnitten wurden die Aufteilungsstrategien mit einer festen Zusammenfassungsstrategie bzw. die Zusammenfassungsstrategien mit einer festen Aufteilungsstrategie untersucht. Durch die Kombinationsmöglichkeiten der Aufteilungs- und Zusammenfassungsstrategien ergeben sich jedoch weitere Möglichkeiten zur Minimierung der Laufzeit bzw. der Überschätzung. Diese werden im Folgenden untersucht. Die Ergebnisse sind in Tabellen 4.6 bis 4.8 dargestellt. Die Darstellung erfolgt hier absolut. Das Minimum je Schaltungsbeispiel ist kursiv dargestellt. Das Minimum der Laufzeit stellt sich für das Beispiel BJT_INV für die Kombination aus *Merge Each* und *Select By Deviation*, für das Beispiel BP für *Merge Each* und *Select By Sensitivity And Deviation* ein. Für das Beispiel DIFF_AMP ist die Kombination *Select By Sensitivity And Deviation* und *Try Root First*, für NLTL₃ *Merge Each* und *Select By Sensitivity And Deviation* sowie für SVF *Do Not Merge* und *Select By Deviation* das Optimum. Die minimale Anzahl der Aufteilungen ergibt sich für das Beispiel BJT_INV für die Zusammenfassungsstrategie *Do Not Merge* und die Aufteilungsstrategien *Select All*, *Select By Deviation* sowie *Select By Sensitivity And Deviation*. Für das Beispiel BP ist die Kombination aus *Do Not Merge* und *Select All*, *Select By Deviation* oder *Select By Sensitivity And Deviation* am besten geeignet. Für das Beispiel DIFF_AMP stellt *Do Not Merge* mit *Select By Sensitivity And Deviation* das Optimum dar. Das Beispiel NLTL₃ erlaubt viele Kombinationen, die hinsichtlich der Anzahl der Aufteilungen gleich sind: *Select By Deviation*, *Select By Sensitivity* oder *Select By Sensitivity And Deviation* mit *Merge Each*, *Try Root First* oder *Merge nth Step* sowie *Merge Each* mit *Try Root First*. Für SVF ist die Kombination aus *Select By Deviation* und *Merge Each* das Optimum. Für die meisten Beispiele führt also die Zusammenfassungsstrategie *Merge Each* zu der geringsten Laufzeit. Lediglich die Beispiele SVF und DIFF_AMP weichen hiervon ab. Bei den Aufteilungsstrategien liefern *Select By Deviation* und *Select By Sensitivity And Deviation* für die meisten Beispiele ein ähnliches Ergebnis. Die Unterschiede liegen in der Größenordnung der Messgenauigkeit. Für das Beispiel BJT_INV ist *Select By Deviation* geringfügig besser als *Select By Sensitivity And Deviation*. Für das Beispiel DIFF_AMP ist dagegen *Select By Sensitivity And Deviation* deutlich besser als *Select By Deviation*. *Select By Chance* ist in keinem Fall die schnellste Strategie, sie gehört stets zu den langsamsten. Für die Beispiele BP, NLTL₃ und SVF liefert sie für die gewählten Parameter unabhängig von der Zusammenfassungsstrategie kein konvergentes Ergebnis. Dies sind mit Ausnahme von DIFF_AMP die Beispiele mit den meisten PPDs, für das Beispiel DIFF_AMP ist die Auswahl eines geeigneten Parameters zur Aufteilung unkritischer. Die zufällige Auswahl eines Parameters zur Aufteilung ist daher nicht sinnvoll. Die geringste Überschätzung für das Beispiel BP ergibt sich für die Kombination aus *Select All* und *Merge Each*. Der Vorteil gegenüber den anderen Kombinationen ist jedoch gering, er bewegt sich im Bereich der dritten Nachkommastelle. Für die Beispiele DIFF_AMP und BJT_INV ergeben sich die geringsten Überschätzungen für die Kombination aus *Select All* und *Do Not Merge*, die geringste Überschätzung für das Beispiel SVF ergibt sich

dagegen für *Select All* und *Merge Each*. Für das Beispiel NLTL₃ ist die Überschätzung in den möglichen Kombinationen aus *Merge Each* und *Try Root First* mit *Select All*, *Select By Deviation*, *Select By Sensitivity* und *Select By Sensitivity And Deviation* am geringsten. Im Vergleich mit den Ergebnissen aus den Abbildungen 4.16 bis 4.18 sowie Abbildung 4.20 zeigt sich ein ähnliches Verhalten. Die geringste Laufzeit ergibt sich wie in Abbildung 4.20 für die meisten Beispiele für die Zusammenfassungsstrategie *Merge Each*. Lediglich für die Beispiele DIFF_AMP und SVF ist die Strategie *Try Root First* bzw. *Do Not Merge* schneller. Durch die Wahl einer geeigneten Aufteilungsstrategie ergibt sich eine Beschleunigung gegenüber der Darstellung in Abbildung 4.16 und der dortigen Aufteilungsstrategie *Select All*. Die geringste Anzahl von Aufteilungen ergibt sich für die Zusammenfassungsstrategie *Do Not Merge*. Durch die Kombinationen mit den verschiedenen Aufteilungsstrategien ergeben sich für die Beispiele DIFF_AMP, NLTL₃ und SVF Vorteile gegenüber den Ergebnissen aus Abbildung 4.17 mit der Strategie *Select All*. Die geringste Überschätzung lässt sich wie in Abbildung 4.18 stets durch die Strategie *Do Not Merge* erreichen. Da im Rahmen dieser Arbeit die Laufzeit minimiert werden soll, werden folgende optimierte Einstellungen verwendet: für BJT_INV *Select By Deviation* und *Merge Each*, für BP *Select By Sensitivity And Deviation* und *Merge Each*, für DIFF_AMP *Select By Sensitivity And Deviation* und *Try Root First*, für NLTL₃ *Select By Sensitivity And Deviation* und *Merge Each*. Für das Beispiel SVF wird die Kombination *Select By Deviation* und *Merge Each* verwendet. Die Kombination aus *Select By Deviation* und *Do Not Merge* führt hier zwar zu einer geringeren Laufzeit, jedoch ist dabei die Überschätzung so hoch, dass das Ergebnis nicht mehr sinnvoll zu verwenden ist.

4.8 Kombination der Aufteilungs- und Zusammenfassungsstrategien

Tabelle 4.6: Vergleich der Laufzeit (in Sekunden) für die verschiedenen Kombinationen aus Aufteilungs- und Zusammenfassungsstrategien für die verschiedenen Schaltungsbeispiele. Die Minima je Schaltungsbeispiel sind kursiv markiert.

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	2,43	7,34	2,61	3,15
<i>Select By Deviation</i>	2,29	3,96	2,57	2,96
<i>Select By Chance</i>	2,78	9,19	2,99	3,53
<i>Select By Sensitivity</i>	2,75	9,16	3,03	3,60
<i>Select By Sensitivity And Deviation</i>	2,74	8,27	3,00	3,50
<i>Select By EPD And PPD</i>	2,79	9,14	3,03	3,58
<i>Select By EPD And Sensitivity</i>	2,73	9,16	3,03	3,53

(a) BJT_INV

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	7,68	8,52	-	8,15
<i>Select By Deviation</i>	1,39	1,45	-	1,47
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	1,37	1,47	-	1,42
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

(b) BP

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	35,14	605,50	52,23	34,74
<i>Select By Deviation</i>	54,10	689,46	62,61	48,18
<i>Select By Chance</i>	66,21	1181,90	113,75	57,81
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	4,73	22,38	4,69	4,73
<i>Select By EPD And PPD</i>	66,51	1181,19	113,78	57,93
<i>Select By EPD And Sensitivity</i>	65,94	1179,29	113,59	57,82

(c) DIFF_AMP

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	32,86	-	34,01	54,00
<i>Select By Deviation</i>	15,53	-	15,89	16,61
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	15,56	-	15,97	-
<i>Select By Sensitivity And Deviation</i>	15,56	-	15,86	-
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

(d) NLTL₃

4 Ergebnisse

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	7969,73	18 686,12	18 847,83	10 745,07
<i>Select By Deviation</i>	3,79	2,28	2,98	3,49
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	-	-	-	-
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

(e) SVF

Tabelle 4.7: Vergleich der Anzahl der Aufteilungen für die verschiedenen Kombinationen aus Aufteilungs- und Zusammenfassungsstrategien für die verschiedenen Schaltungsbeispiele. Die Minima je Schaltungsbeispiel sind kursiv markiert.

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	20	1	20	20
<i>Select By Deviation</i>	20	1	20	20
<i>Select By Chance</i>	60	3	60	60
<i>Select By Sensitivity</i>	60	3	60	60
<i>Select By Sensitivity And Deviation</i>	60	3	60	60
<i>Select By EPD And PPD</i>	60	3	60	60
<i>Select By EPD And Sensitivity</i>	60	3	60	60

(a) BJT_INV

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	3	1	-	2
<i>Select By Deviation</i>	3	1	-	2
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	3	1	-	2
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

(b) BP

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	100	9	45	55
<i>Select By Deviation</i>	460	39	195	250
<i>Select By Chance</i>	461	45	225	240
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	100	8	40	60
<i>Select By EPD And PPD</i>	461	45	225	240
<i>Select By EPD And Sensitivity</i>	461	45	225	240

(c) DIFF_AMP

4.8 Kombination der Aufteilungs- und Zusammenfassungsstrategien

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge n^{th} Step</i>
<i>Select All</i>	9	-	9	9
<i>Select By Deviation</i>	2	-	2	2
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	2	-	2	-
<i>Select By Sensitivity And Deviation</i>	2	-	2	-
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

(d) NLTL₃

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge n^{th} Step</i>
<i>Select All</i>	26	1	1	13
<i>Select By Deviation</i>	71	3	3	39
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	-	-	-	-
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

(e) SVF

Tabelle 4.8: Vergleich der Überschätzung für die verschiedenen Kombinationen aus Aufteilungs- und Zusammenfassungsstrategien für die verschiedenen Schaltungsbeispiele. Die Minima je Schaltungsbeispiel sind kursiv markiert.

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	1,367	<i>1,086</i>	1,367	1,367
<i>Select By Deviation</i>	1,367	1,090	1,367	1,367
<i>Select By Chance</i>	1,367	1,356	1,367	1,367
<i>Select By Sensitivity</i>	1,367	1,356	1,367	1,367
<i>Select By Sensitivity And Deviation</i>	1,367	1,322	1,367	1,367
<i>Select By EPD And PPD</i>	1,367	1,356	1,367	1,367
<i>Select By EPD And Sensitivity</i>	1,367	1,356	1,367	1,367
(a) BJT_INV^a				
	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	5,632	5,640	-	5,636
<i>Select By Deviation</i>	5,634	5,638	-	5,637
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	5,634	5,638	-	5,637
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-
(b) BP				
	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	1,165	<i>1,128</i>	1,164	1,164
<i>Select By Deviation</i>	1,168	1,155	1,167	1,167
<i>Select By Chance</i>	1,167	1,158	1,167	1,167
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	1,168	1,157	1,167	1,167
<i>Select By EPD And PPD</i>	1,167	1,158	1,167	1,167
<i>Select By EPD And Sensitivity</i>	1,167	1,158	1,167	1,167
(c) DIFF_AMP				
	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge nth Step</i>
<i>Select All</i>	9,442	-	9,442	9,515
<i>Select By Deviation</i>	9,442	-	9,442	9,528
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	9,442	-	9,442	-
<i>Select By Sensitivity And Deviation</i>	9,442	-	9,442	-
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-
(d) NLTL₃				

^aDie Verwendung von Gleichung (3.2) führt bei diesem Beispiel zu Werten kleiner 1. Der Einschluss der affinen Ergebnisse ist jedoch gegeben. Daher wird bei diesem Beispiel die äquivalente, rekursive Formel $m_{OA,i+1} = m_{OA,i} + \frac{1}{i} \cdot \frac{d_A(t)}{d_{MC}(t)}$ verwendet, die geringere numerische Ungenauigkeiten verursacht.

4.8 Kombination der Aufteilungs- und Zusammenfassungsstrategien

	<i>Merge Each</i>	<i>Do Not Merge</i>	<i>Try Root First</i>	<i>Merge n^{th} Step</i>
<i>Select All</i>	5,036	$7,572 \cdot 10^{16}$	$7,572 \cdot 10^{16}$	$3,460 \cdot 10^{11}$
<i>Select By Deviation</i>	13,110	$9,610 \cdot 10^{16}$	$9,610 \cdot 10^{16}$	$1,706 \cdot 10^{12}$
<i>Select By Chance</i>	-	-	-	-
<i>Select By Sensitivity</i>	-	-	-	-
<i>Select By Sensitivity And Deviation</i>	-	-	-	-
<i>Select By EPD And PPD</i>	-	-	-	-
<i>Select By EPD And Sensitivity</i>	-	-	-	-

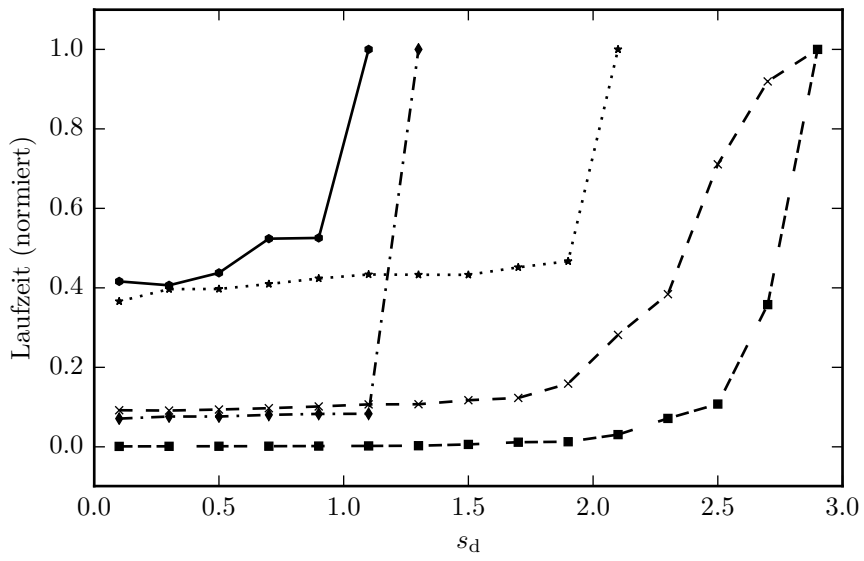
(e) SVF

4.9 Laufzeitverhalten mit Gebietsaufteilungen

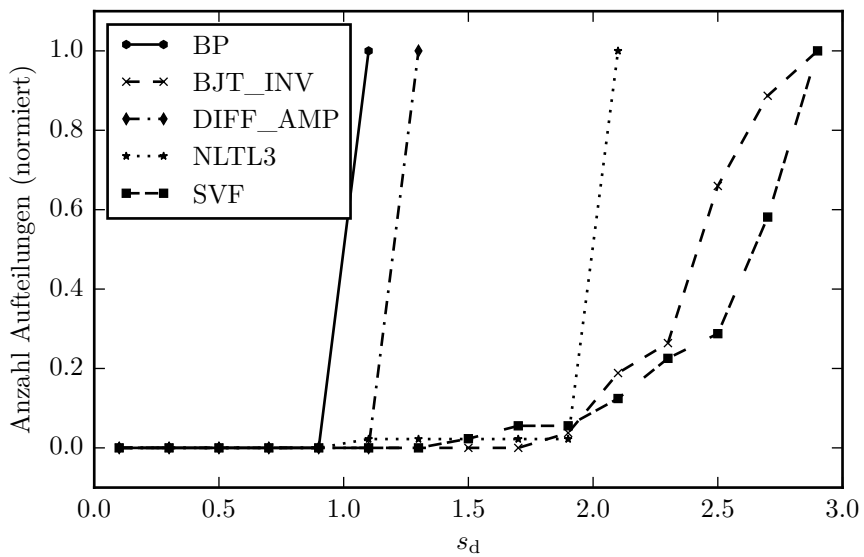
Analog zu Abschnitt 4.3 wird im Folgenden das Laufzeitverhalten des EPD-Verfahrens mit Gebietsaufteilungen gemessen. Zunächst wird die Abhängigkeit von der Parametergröße betrachtet. Dazu werden die Parameter der Schaltungsbeispiele wiederum mit dem Skalierungsfaktor s_d gemäß Gleichung (4.7) skaliert. Die Ergebnisse sind in Abbildung 4.21a auf die höchste Laufzeit je Schaltungsbeispiel normiert dargestellt. Die Größe der Parameterunsicherheiten ist wie für die Ergebnisse in Abbildung 4.12 gewählt. Für $s_d > 1$ sind daher Gebietsaufteilungen erforderlich. Ihre Anzahl ist in Abbildung 4.21b dargestellt. Im Vergleich mit Abbildung 4.12 zeigt sich, dass die Parameterunsicherheiten, für die eine konvergente Lösung möglich ist, für alle Schaltungsbeispiele vergrößert werden. Die Vergrößerung ist schaltungsabhängig. Durch die Gebietsaufteilungen verschlechtert sich jedoch das Laufzeitverhalten. Die lineare Abhängigkeit der Laufzeit vom Skalierungsfaktor s_d ohne Gebietsaufteilungen geht in eine exponentielle Abhängigkeit über. Dadurch ist die mögliche Vergrößerung des Konvergenzgebiets durch Gebietsaufteilungen begrenzt.

Die Messung der Laufzeiten mit Gebietsaufteilungen der Kettenschaltungen BJT_INV_n nach Abbildung 4.8 und NLTL_n nach Abbildung 4.9 sind in Abbildung 4.22 bzw. Abbildung 4.23 dargestellt. Zusätzlich wird die Anzahl der Aufteilungen gezeigt. Mit zunehmender Anzahl der Elemente steigt auch die Anzahl der gemäß Kriterium 1 benötigten Aufteilungen. Das Laufzeitverhalten in Abhängigkeit von der Elementanzahl n ist weiterhin kubisch, durch die Einführung der Gebietsaufteilungen ergibt sich keine Verschlechterung des Laufzeitverhaltens. Im Vergleich mit den Ergebnissen der Kettenschaltungen ohne Gebietsaufteilungen aus Abschnitt 4.3 zeigt sich, dass die maximale Anzahl der Elemente n für beide Kettenschaltungen vergrößert werden konnte. Die Erweiterung des EPD-Verfahrens um Gebietsaufteilungen vergrößert also nicht nur das Volumen V_c der konvexen Hülle des Konvergenzgebiets im Parameterraum (vgl. Tabelle 4.4), sondern auch die maximale Anzahl von Schaltungsvariablen n_x , die mit einem gebietsarithmetischen Lösungsverfahren gelöst werden kann.

Im Folgenden wird der Einfluss der Anzahl der Parameter n_p auf das Verhalten des EPD-Verfahrens mit Gebietsaufteilungen untersucht. Dazu werden die affinen Parameter nacheinander in nominale Parameter umgewandelt. Die Reihenfolge ist dabei die gleiche wie für die Ergebnisse aus Abbildungen 4.10 und 4.11. Hier sind jedoch alle Parameterunsicherheiten um dem Faktor 1,5 größer gewählt (vgl. Anhang A.2.1 mit Anhang A.8.3), so dass Aufteilungen zur Lösung erforderlich sind. Die Ergebnisse sind in Abbildungen 4.24 und 4.25 dargestellt. Die Überschätzung wurde aus jeweils 1000 Monte-Carlo-Stichproben berechnet. Das Verhalten bleibt grundsätzlich gleich, je größer die Anzahl der Parameter, desto größer die benötigte Laufzeit und die Überschätzung. Das Beispiel DIFF_AMP ist für die gewählten Größen der Parameterunsicherheiten und die maximale Anzahl von Gebietsaufteilungen nur bis $n_p = 4$ lösbar. Im Vergleich mit den Ergebnissen aus Abbildungen 4.10 und 4.11 ohne Gebietsaufteilungen zeigt sich, dass die Kurvenform nahezu gleich bleibt. Sie wird also durch die Reihenfolge und relative Größen der Unsicherheiten zueinander bestimmt. Die Laufzeit ist wie erwartet für alle Beispiele mit Gebietsaufteilungen größer als ohne Gebietsaufteilungen. Für alle Schaltungsbeispiele steigt die Überschätzung m_{OA} mit der Anzahl der Parameter n_p an.



(a) Laufzeit



(b) Anzahl der Aufteilungen

Abbildung 4.21: Laufzeit und Anzahl der Aufteilungen für verschiedene Werte von s_d mit Gebietsaufteilungen

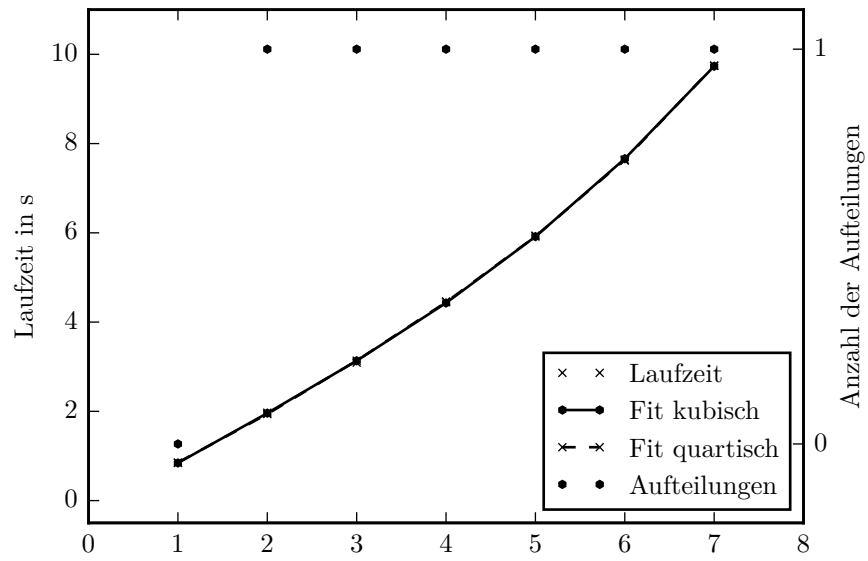


Abbildung 4.22: Laufzeit für verschiedene Werte von n in der Kettenschaltung BJT_INV_n

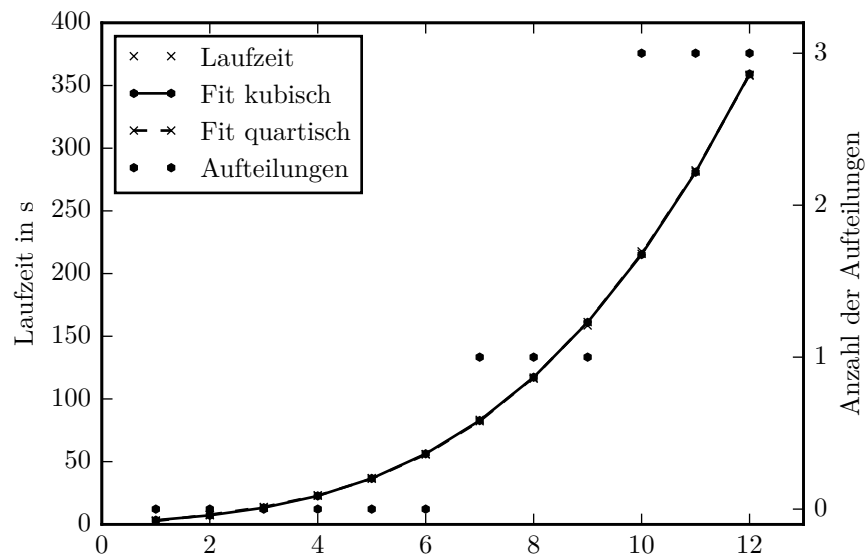


Abbildung 4.23: Laufzeit für verschiedene Werte von n in der Kettenschaltung $NLTL_n$

4.9 Laufzeitverhalten mit Gebietsaufteilungen

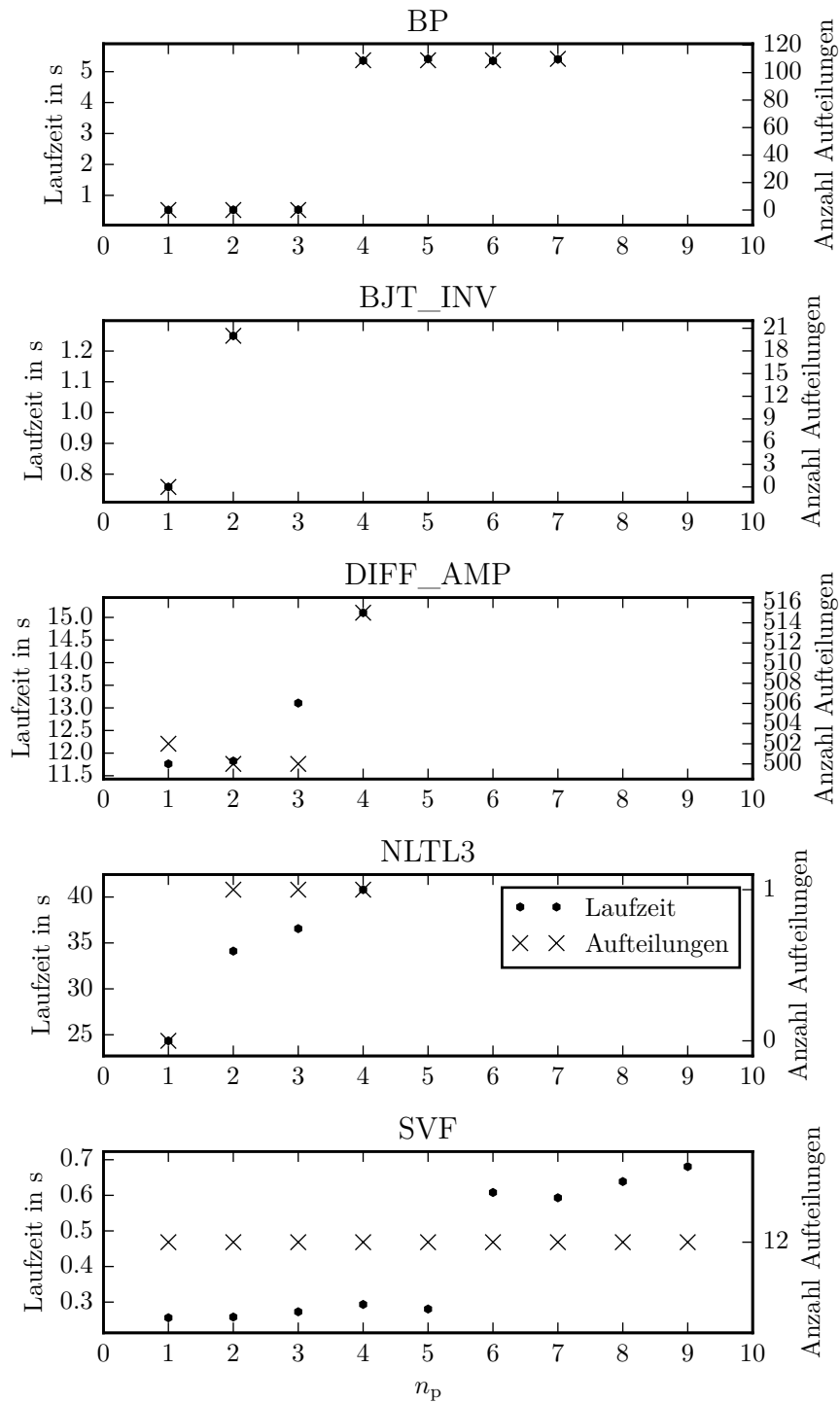


Abbildung 4.24: Laufzeit und Anzahl der Aufteilungen für verschiedene Werte von n_p

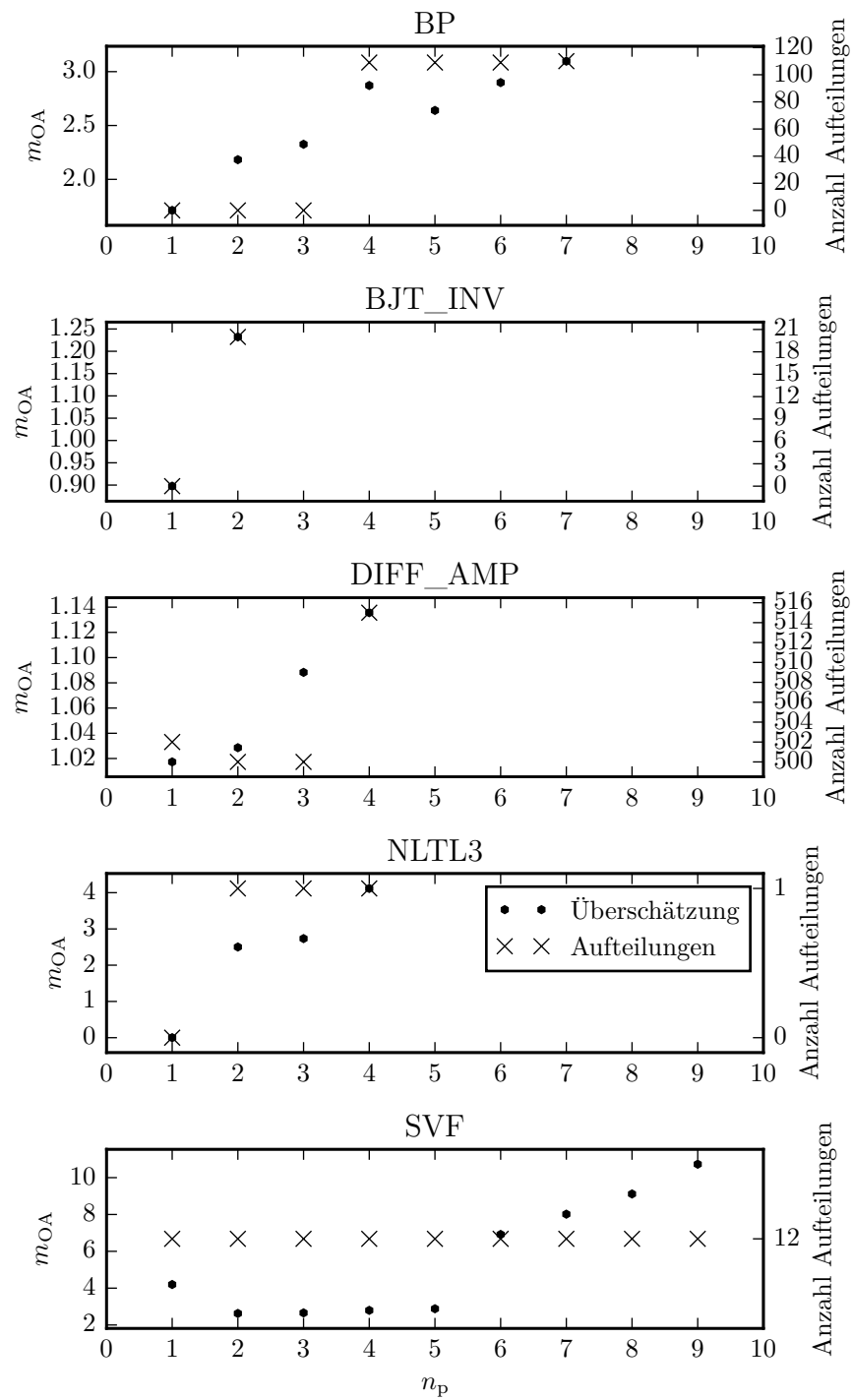
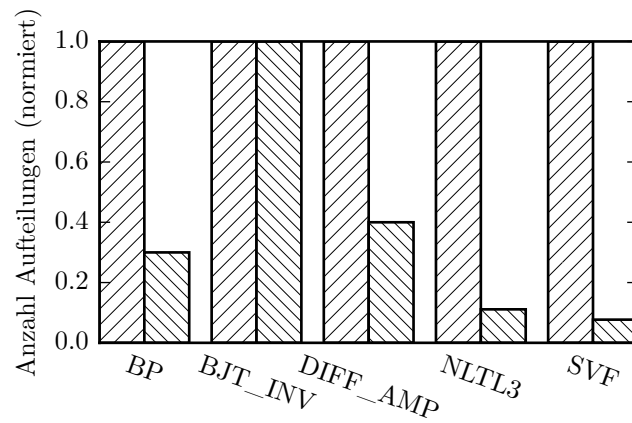


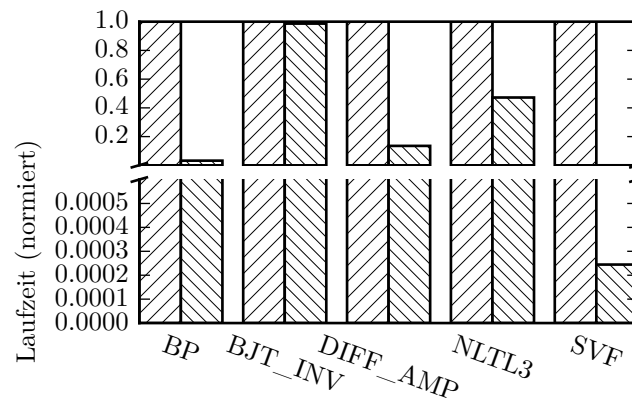
Abbildung 4.25: Überschätzung und Anzahl der Aufteilungen für verschiedene Werte von n_p

4.10 Vergleich der unoptimierten mit den optimierten Einstellungen

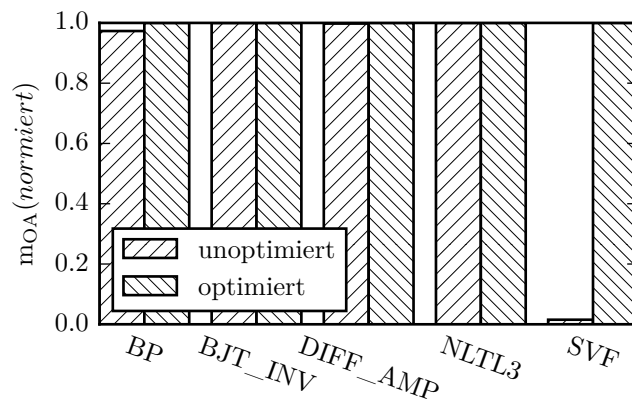
Zusammenfassend werden in Abbildung 4.26 alle Schaltungsbeispiele mit ihren unoptimierten und ihren optimierten Einstellungen gegenübergestellt. Dabei werden für jedes Schaltungsbeispiel separat die optimalen Werte aus den vorangegangenen Abschnitten bestimmt. Die so gefundenen optimierten Einstellungen werden in Anhang A.9 dargestellt. Durch die Verwendung der optimierten Einstellungen kann die benötigte Laufzeit reduziert werden. Der Anteil der durch die optimierten Einstellungen eingesparten Laufzeit ist schaltungsabhängig und reicht von 1,4% für das Beispiel BJT_INV bis 99,9% für das Beispiel SVF. Die hohe Laufzeiteinsparung für das Beispiel SVF geht mit einer starken Erhöhung der Überschätzung um etwa 98,5% einher. Für das Beispiel BP steigt die Überschätzung um etwa 3%. Für die übrigen Beispiele bleibt die Überschätzung konstant.



(a) Anzahl Aufteilungen n_{splits}



(b) Laufzeit



(c) Überschätzung

Abbildung 4.26: Vergleich der unoptimierten mit den optimierten Einstellungen

5 Zusammenfassung

Die Berücksichtigung von mit Unsicherheiten behafteten Parametern bei der Schaltungssimulation ist zur Entwicklung von robusten Schaltungen notwendig. Die Ursachen der Unsicherheiten können vielfältig sein, wie zum Beispiel die Umgebungstemperatur, Fertigungstoleranzen oder Modellungenauigkeiten. Eine Möglichkeit, sie zu berücksichtigen, ist die Verwendung von gebietsarithmetischen Simulationsverfahren wie der affinen Arithmetik. Gegenüber herkömmlichen Verfahren wie Monte-Carlo- oder Corner-Case-Simulationen bieten sie den Vorteil, dass alle Lösungen, die sich durch alle möglichen Kombinationen der unsicheren Parameter ergeben, sicher eingeschlossen werden.

Für die Schaltungssimulation wird im Allgemeinen ein Verfahren benötigt, das nicht-lineare, implizite und zeitabhängige Differentialgleichungen lösen kann. Ein solches Verfahren, das sog. EPD-Verfahren, wurde in [27] vorgestellt. Es konvergiert jedoch nur für relativ kleine Parameterunsicherheiten und Schaltungen. Die Größe des Konvergenzgebiets wird als Teil dieser Arbeit systematisch untersucht. Dazu wird ein Algorithmus zur automatischen Exploration des Konvergenzgebiets vorgestellt. Alle Untersuchungen werden anhand von fünf Beispielschaltungen mit steigender Komplexität durchgeführt. Es wird gezeigt, dass die Abhängigkeit der Laufzeit des EPD-Verfahrens von der Anzahl der Parameter und ihrer Größe jeweils linear ist. Die Abhängigkeit von der Schaltungsgröße ist dagegen polynomiell. Zur Bewertung der Überschätzung wird die mittlere relative Überschätzung *mOA* eingeführt.

Den Hauptteil dieser Arbeit stellt eine Erweiterung des EPD-Verfahrens dar, die es ermöglicht, größere Parameterunsicherheiten und größere Schaltungen zu simulieren. Dazu werden die Parameterunsicherheiten – wenn nötig – aufgeteilt, die Lösungen getrennt berechnet und wieder zu einer Gesamtlösung zusammengefügt. Dadurch kann das Konvergenzgebiet des EPD-Verfahrens aus [27] vergrößert werden. Die mögliche Vergrößerung ist schaltungsabhängig und bewegt sich zwischen 10 und 444 %.

Bei der Durchführung der Gebietsaufteilungen und -zusammenfassungen ergeben sich drei Teilprobleme:

1. die automatische Bestimmung, ob und zu welchem Zeitpunkt eine Aufteilung der Parameter erforderlich ist,
2. die Auswahl einer Parameterunsicherheit zur Aufteilung,
3. die Entscheidung, zu welchem Zeitpunkt eine Zusammenfassung der Lösung stattfindet.

Die ersten beiden Teilprobleme existieren für alle Simulationsarten, Teilproblem 3 dagegen nur für transiente Simulationen. Da die transiente Simulation die häufigste Simulationsart

ist, werden im Rahmen dieser Arbeit alle drei Teilprobleme untersucht. Für die drei Teilprobleme existieren drei verschiedene Optimierungsziele, die sich im Allgemeinen widersprechen: die Minimierung der Laufzeit, der Überschätzung oder des Speicherbedarfs. Das Hauptziel dieser Arbeit ist die Minimierung der Laufzeit.

Zur Lösung von Teilproblem 1 werden zwei verschiedene Strategien vorgeschlagen und untersucht: die Aufteilung nach Kriterium 1 oder nach Kriterium 2. Kriterium 1 verwendet die Anzahl der benötigten Iterationen n_{iter} , Kriterium 2 dagegen den Skalierungsfaktor Δs zur Bestimmung, ob eine Gebietsaufteilung durchgeführt wird. Beide Kriterien verwenden einen benutzerdefinierten Schwellwert bei dessen Überschreitung eine Gebietsaufteilung durchgeführt wird. Als geeignete Schwellwerte für n_{iter} und Δs wurden $n_{\text{iter,max}} = 39$ bzw. $\Delta s_{\text{max}} = 0,9 \cdot 10^{-4}$ bestimmt. Für drei der fünf betrachteten Beispielschaltungen führt Kriterium 1 zu einer geringeren Laufzeit als Kriterium 2. Für die übrigen zwei Schaltungen ist die Aufteilung nach Kriterium 2 schneller. Eine Kombination aus Kriterium 1 und 2 ist in jedem Fall langsamer als die Einzelkriterien und wird daher nicht weiter betrachtet.

Zur Lösung des Teilproblems 2 werden sieben verschiedene Aufteilungsstrategien vorgestellt und untersucht: *Select All*, *Select By Chance*, *Select By Deviation*, *Select By EPD And PPD*, *Select By EPD And Sensitivity*, *Select By Sensitivity*, *Select By Sensitivity And Deviation*. Für die meisten Beispielschaltungen führt die Strategie *Select By Deviation* zu der geringsten Laufzeit. Lediglich für die Beispielschaltung DIFF_AMP führt diese Strategie zu keiner konvergenten Lösung, hier ist die Strategie *Select By Sensitivity And Deviation* die schnellste. Die Wahl der Aufteilungsstrategie hat außer für das Beispiel SVF keine Auswirkung auf die Überschätzung m_{OA} .

Zur Lösung des Teilproblems 3 werden vier verschiedene Zusammenfassungsstrategien vorgestellt und untersucht: *Merge Each*, *Merge n^{th} Step*, *Try Root First*, *Do Not Merge*. Bei der Strategie *Merge n^{th} Step* ist die Schrittweite n , nach der eine Zusammenfassung durchgeführt wird, vom Benutzer einstellbar. Der Wert $n = 2$ führt zu der geringsten Laufzeit. Für alle Beispielschaltungen führt die Strategie *Merge Each* zu der geringsten Laufzeit. Für zwei Schaltungsbeispiele (DIFF_AMP und BJT_INV) wird dieser Vorteil im Vergleich zu *Do Not Merge* jedoch durch eine höhere Überschätzung eingetauscht. Durch die Wahl der Zusammenfassungsstrategie können daher unter Umständen Laufzeit und Überschätzung gegeneinander abgewogen werden.

Durch die verschiedenen Kombinationsmöglichkeiten der Aufteilungs- und Zusammenfassungsstrategien ergeben sich keine weiteren Möglichkeiten zur Verringerung der Laufzeit. Werden alle optimierten Einstellungen zur Lösung der Teilprobleme 1 bis 3 verwendet, ergibt sich insgesamt eine Laufzeiteinsparung zwischen 3 und 98 % gegenüber den unoptimierten Einstellungen.

Literaturverzeichnis

- [1] AKHMEROV, Ramil R.: Interval-affine gaussian algorithm for constrained systems. *Reliable Computing* 11 (2005), S. 323–341
- [2] ALTHOFF, M. ; STURSBURG, O. ; BUSS, M.: Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. *47th IEEE Conference on Decision and Control*. Dez. 2008, S. 4042–4048
- [3] ALTHOFF, Matthias: Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*. Philadelphia, Pennsylvania, USA: ACM, 2013, S. 173–182
- [4] AMENT, Ch.: Skript zur Vorlesung Modellbildung (II, TKS, WIW) bzw. Prozessanalyse (EIT, IN, MB). Vorlesungsskript. Technische Universität Ilmenau - Institut für Automatisierungs- und Systemtechnik - Fachgebiet Systemanalyse, Wintersemester 2013/2014
- [5] BAHAREV, Ali ; ACHTERBERG, Tobias ; RÉV, Endre: Computation of an extractive distillation column with affine arithmetic. *AIChE Journal* 55 (7) (2009), S. 1695–1704
- [6] BAHAREV, Ali ; KOLEV, Lubomir ; RÉV, Endre: Computing multiple steady states in homogeneous azeotropic and ideal two-product distillation. *AIChE Journal* 57 (6) (2011), S. 1485–1495
- [7] BORDER, KC: Notes on the implicit function theorem. Forschungsbericht. California Institute of Technology - Division of the Humanities und Social Sciences, 2013
- [8] BUNGARTZ, Hans-Joachim ; ZIMMER, Stefan ; BUCHHOLZ, Martin ; PFLÜGER, Dirk: Modellbildung und Simulation - Eine anwendungsorientierte Einführung. Springer Verlag, 2009
- [9] CAPRANI, Ole ; MADSEN, Kaj ; RALL, L. B.: Integration of interval functions. *SIAM Journal on Mathematical Analysis* 12 (1981), S. 321–341
- [10] CARVALHO NASCIMENTO, Filipe de ; PAIVA, Afonso ; FIGUEIREDO, Luiz Henrique de ; STOLFI, Jorge: Approximating implicit curves on plane and surface triangulations with affine arithmetic. *Computers & Graphics* 40 (2014), S. 36–48
- [11] COMBA, João Luiz Dohl ; STOLFI, Jorge: Affine arithmetic and its applications to computer graphics. *Proceedings of VI SIBGRAPI*. 1993, S. 9–18

- [12] DAUMAS, Marc ; LESTER, David R. ; MUÑOZ, César A.: Verified real number calculations: A library for interval arithmetic. *IEEE Transaction on Computers* 58 (2) (2009), S. 226–237
- [13] DAUMAS, Marc ; MELQUIOND, Guillaume ; MUÑOZ, César A.: Guaranteed proofs using interval arithmetic. *IEEE Symposium on Computer Arithmetic* (2005), S. 188–195
- [14] DREYER, Alexander: Combination of symbolic and interval-numeric methods for analysis of analog circuits. *Proceedings of the 8th International Workshop on Symbolic Methods and Applications in Circuit Design (SMACD 2004)*. 2004
- [15] DREYER, Alexander: Interval analysis of analog circuits with component tolerances. Dissertation. Technische Universität Kaiserslautern, 2005
- [16] DREYER, Alexander: Interval methods for analog circuits. Forschungsbericht. Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, 2006
- [17] DREYER, Alexander: Resolving parameter dependences for interval analysis of linear analog circuits. *Proceedings of the 9th International Workshop on Symbolic Methods and Applications in Circuit Design (SMACD2006)*. 2006
- [18] EBERS, J.J. ; MOLL, J.L.: Large-signal behavior of junction transistors. *Proceedings of the IRE* 42 (12) (Dez. 1954), S. 1761–1772
- [19] ENZ, Christian C. ; KRUMMENACHER, François ; VITTOZ, Eric A.: An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications. *Analog Integrated Circuits and Signal Processing* 8 (1) (Juli 1995), S. 83–114
- [20] FANG, C.F. ; RUTENBAR, R.A. ; CHEN, Tsuhan: Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs. *International Conference on Computer Aided Design (ICCAD2003)*. Nov. 2003, S. 275–282
- [21] FIGUEIREDO, L. H. de ; STOLFI, J.: Self-validated numerical methods and applications. Brazilian Mathematics Colloquium Monographs. IMPA, 1997
- [22] FIGUEIREDO, Luiz Henrique de ; STOLFI, Jorge: Adaptive enumeration of implicit surfaces with affine arithmetic. *Computer Graphics Forum* 15 (1996), S. 287–296
- [23] FITAN, E. ; MESSINE, F. ; NOGAREDE, B.: A general analytical model of electrical permanent magnet machine dedicated to optimal design. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering* 22 (4) (2003), S. 1037–1050
- [24] FITAN, E. ; MESSINE, F. ; NOGAREDE, B.: The electromagnetic actuator design problem: a general and rational approach. *IEEE Transactions on Magnetics* 40 (3) (Mai 2004), S. 1579–1590
- [25] FONTCHASTAGNER, J. ; MESSINE, F. ; LEFEVRE, Y.: Design of electrical rotating machines by associating deterministic global optimization algorithm with combinatorial analytical and numerical models. *IEEE Transactions on Magnetics* 43 (8) (Aug. 2007), S. 3411–3419

- [26] FREISFELD, Marc: Semi-symbolische Modellierung und Simulation von Unsicherheiten in analogen Schaltungen mit Hilfe stückweise affiner Abbildungen. Dissertation. Leibniz Universität Hannover - Institut für mikroelektronische Systeme, 2009
- [27] GRABOWSKI, D.: Gebietsarithmetische Verfahren zur Simulation analoger Schaltungen mit Parameterunsicherheiten. Dissertation. Leibniz Universität Hannover - Institut für Mikroelektronische Systeme, 2009
- [28] GRABOWSKI, D. ; GRIMM, C. ; BARKE, E.: Ein Verfahren zur effizienten Analyse von Schaltungen mit Parametervarianzen. *9. Workshop: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV)*. Fraunhofer IIS, 2006, S. 181–190
- [29] GRABOWSKI, D. ; OLBRICH, M. ; BARKE, E.: Analog circuit simulation using range arithmetics. *Proceedings of the ASP-DAC*. 2008, S. 762–767
- [30] GRIMM, C. ; HEUPKE, W. ; WALDSCHMIDT, K.: Analysis of mixed-signal systems with affine arithmetic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24 (1) (2005), S. 118–123
- [31] GU, Wei ; LUO, Lizi ; DING, Tao u. a.: An affine arithmetic-based algorithm for radial distribution system power flow with uncertainties. *International Journal of Electrical Power & Energy Systems* 58 (2014), S. 242–245
- [32] HANSEN, Eldon: Global optimization using interval analysis: the one-dimensional case. *Journal of Optimization Theory and Applications* 29 (3) (1979), S. 331–344
- [33] HANSEN, Eldon R.: A generalized interval arithmetic. *Interval Mathematics*. Hrsg. von NICKEL, Karl. Bd. 29. Lecture Notes in Computer Science. Springer, 1975, S. 7–18
- [34] HANSEN, Eldon R.: A globally convergent method for finding and bounding real roots. *BIT* 18 (1978), S. 415–424
- [35] HANSEN, Eldon R.: Global optimization using interval analysis - the multi-dimensional case. *Numerische Mathematik* 34 (1980), S. 247–270
- [36] HANSEN, Eldon R.: On solving systems of equations using interval arithmetic. *Mathematics of Computation* 22 (102) (1968), S. 374–384
- [37] HANSEN, Eldon ; SENGUPTA, Saumyendra: Bounding solutions of systems of equations using interval analysis. *BIT Numerical Mathematics* 21 (2) (1981), S. 203–211
- [38] HOROWITZ, P. ; HILL, W.: Hohe Schule der Elektronik 1. 6. Auflage. Elektor-Verlag GmbH, 2006
- [39] HU, Chenyi ; KEARFOTT, R. Baker ; AWAD, Abdulhamid: On bounding the range of some elementary functions in FORTRAN–77. *Interval Computations* 3 (1993), S. 29–39
- [40] JAULIN, Luc ; KIEFFER, Michel ; DIDRIT, Olivier ; WALTER, Éric: Applied interval analysis. Springer, 2001

- [41] KEARFOTT, R. B.: Abstract generalized bisection and a cost bound. *Mathematics of Computation* 49 (179) (1987), S. 187–202
- [42] KEARFOTT, R. B.: Preconditioners for the interval Gauss–Seidel method. *SIAM Journal on Numerical Analysis* 27 (3) (1990), S. 804–822
- [43] KEARFOTT, R. B.: Rigorous global search: continuous problems. Kluwer Academic Publishers, 1996
- [44] KEARFOTT, R. B. ; DAWANDE, M. ; DU, K. ; HU, Ch.: INTLIB: a portable FORTRAN 77 elementary function library. *ACM Transactions on Mathematical Software* 20 (1994), S. 447–459
- [45] KLETTING, Marco ; RAUH, Andreas ; ASCHEMANN, Harald ; HOFER, Eberhard P.: Consistency tests in guaranteed simulation of nonlinear uncertain systems with applications to an activated sludge process. *Journal of Computational and Applied Mathematics* 199 (2007), S. 213–219
- [46] KNOLL, A. ; HIJAZI, Y. ; KENSLER, A. u. a.: Fast ray tracing of arbitrary implicit surfaces with interval and affine arithmetic. *Computer Graphics Forum* 28 (1) (2009), S. 26–40
- [47] KOLEV, Lubomir V.: Automatic computation of a linear interval enclosure. *Reliable Computing* 7 (1) (2001), S. 17–28
- [48] KOLEV, Lubomir V.: Interval methods for circuit analysis. World Scientific, 1993
- [49] KOLEV, Lubomir V.: New formulae for multiplication of intervals. *Reliable Computing* 12 (4) (2006), S. 281–292
- [50] KOLEV, Lubomir V.: Optimal multiplication of G-intervals. *Reliable Computing* 13 (5) (2007), S. 399–408
- [51] KRASNOCHTANOVA, Irina ; RAUH, Andreas ; KLETTING, Marco u. a.: Interval methods as a simulation tool for the dynamics of biological wastewater treatment processes with parameter uncertainties. *Applied Mathematical Modelling* 34 (3) (2010), S. 744–762
- [52] KRAWCZYK, R.: Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehler-schranken. *Computing* 4 (3) (1969), S. 187–201
- [53] LEE, D.-U. ; GAFFAR, A.A. ; MENCER, O. ; LUK, W.: MiniBit: bit-width optimization via affine arithmetic. *Proceedings of the 42nd Design Automation Conference*. Juni 2005, S. 837–840
- [54] LEMKE, Andreas: Dimensionierung von Analogschaltungen mit formalen Methoden auf der Basis affiner Arithmetik. Dissertation. Leibniz Universität Hannover - Institut für mikroelektronische Systeme, 2004
- [55] LIU, G. L. ; CHEN, J. J. ; ZHU, Z. Q.: Reliability analysis of interval parameters bar structures with affine arithmetic. *Advanced Materials Research* 118-120 (2010)
- [56] MANSON, G.: Calculating frequency response functions for uncertain systems using complex affine analysis. *Journal of Sound and Vibration* 288 (3) (2005), S. 487–521

- [57] MARKOV, S. M.: Extended interval arithmetic involving infinite intervals. *Mathematica Balkanica* 6 (1992), S. 269–304
- [58] MCKAY, M. D. ; BECKMAN, R. J. ; CONOVER, W. J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21 (2) (Mai 1979), S. 239–245
- [59] MESSINE, Frédéric: Extensions of affine arithmetic: application to unconstrained global optimization. *Journal of Universal Computer Science* 8 (11) (2002), S. 992–1015
- [60] MESSINE, Frédéric ; TOUHAMI, Ahmed: A general reliable quadratic form: an extension of affine arithmetic. *Reliable Computing* 12 (3) (2006), S. 171–192
- [61] MIYAJIMA, Shinya ; KASHIWAGI, Masahide: A dividing method utilizing the best multiplication in affine arithmetic. *IEICE Electronics Express* 1 (7) (2004), S. 176–181
- [62] MOORE, Ramon E.: Automatic error analysis in digital computation. Forschungsbericht LMSD84821. Lockheed Missiles und Space Co., 1959
- [63] MOORE, Ramon E.: Interval analysis. Prentice-Hall, 1966
- [64] MOORE, Ramon E.: Methods and applications of interval analysis. Society for Industrial und Applied Mathematics, 1979
- [65] MOORE, Ramon E. ; KEARFOTT, R. Baker ; CLOUD, Michael J.: Introduction to interval analysis. Society for Industrial und Applied Mathematics, 2009
- [66] MUHANNA, R. ; MULLEN, R.: Uncertainty in mechanics problems - interval-based approach. *Journal of Engineering Mechanics* 127 (6) (2001), S. 557–566
- [67] MUHANNA, Rafi L. ; ZHANG, Hao ; MULLEN, Robert L.: Interval finite elements as a basis for generalized models of uncertainty in engineering mechanics. *Reliable Computing* 13 (2) (2007), S. 173–194
- [68] MUÑOZ, J. ; CAÑIZARES, C. ; BHATTACHARYA, K. ; VACCARO, A.: Affine arithmetic based methods for voltage and transient stability assessment of power systems with intermittent generation sources. *Proceedings of 2013 IREP Symposium, Bulk Power System Dynamics and Control - IX, Optimization, Security and Control of the Emerging Power Grid*. Aug. 2013, S. 1–12
- [69] MUÑOZ, J. ; CAÑIZARES, C. ; BHATTACHARYA, K. ; VACCARO, A.: An affine arithmetic-based method for voltage stability assessment of power systems with intermittent generation sources. *IEEE Transactions on Power Systems* 28 (4) (Nov. 2013), S. 4475–4487
- [70] NEUMAIER, Arnold: Interval methods for systems of equations. Cambridge University Press, 1990
- [71] NEUMAIER, Arnold ; POWNUK, Andrzej: Linear systems with large uncertainties, with applications to truss structures. *Reliable Computing* 13 (2) (2007), S. 149–172

- [72] PAYDAVOSI, Navid u. a.: BSIM4.8.0 MOSFET Model - User's Manual. Department of Electrical Engineering und Computer Sciences, University of California, Berkeley, CA 94720, 2013
- [73] POPOVA, Evgenija D.: Solving linear systems whose input data are rational functions of interval parameters. *Numerical Methods and Applications*. Hrsg. von BOYANOV, Todor ; DIMOVA, Stefka ; GEORGIEV, Krassimir ; NIKOLOV, Geno. Bd. 4310. Lecture Notes in Computer Science. Springer, 2007, S. 345–352
- [74] POPOVA, Evgenija ; IANKOV, Roumen ; BONEV, Zdravko: Bounding the response of mechanical structures with uncertainties in all the parameters. *Proceedings of the NSF Workshop on Reliable Engineering Computing (REC)* (2006), S. 245–265
- [75] PU, Yu ; HA, Yajun: An automated, efficient and static bit-width optimization methodology towards maximum bit-width-to-error tradeoff with affine arithmetic model. *Asia and South Pacific Conference on Design Automation*. Jan. 2006, S. 886–891
- [76] RADOJICIC, C. ; MORENO, J. ; PAN, Xiao ; GRIMM, C.: On more dependable assertion-based verification. *39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013)*. Nov. 2013, S. 7742–7747
- [77] RADOJICIC, C. ; SCHUPFER, F. ; RATHMAIR, M. ; GRIMM, C.: Assertion-based verification of signal processing systems with affine arithmetic. *Forum on Specification and Design Languages (FDL)*. Sep. 2012, S. 20–26
- [78] RATHMAIR, M. ; SCHUPFER, F. ; RADOJICIC, C. ; GRIMM, C.: Extended framework for system simulation with affine arithmetic. *Forum on Specification and Design Languages (FDL)*. Sep. 2012, S. 168–175
- [79] RATZ, D.: Box-splitting strategies for the interval Gauss-Seidel step in a global optimization method. *Computing* 53 (3-4) (1994), S. 337–353
- [80] RAUH, Andreas ; KLETTING, Marco ; ASCHEMANN, Harald ; HOFER, Eberhard P.: Application of interval arithmetic simulation techniques to wastewater treatment process. *Proceedings of the 23rd IASTED*. 2004, S. 287–293
- [81] RAUH, Andreas ; KLETTING, Marco ; ASCHEMANN, Harald ; HOFER, Eberhard P.: Reduction of overestimation in interval arithmetic simulation of biological wastewater treatment processes. *Journal of Computational and Applied Mathematics* 199 (2) (2007). Special Issue on Scientific Computing, Computer Arithmetic and Validated Numbers (SCAN 2004), S. 207–212
- [82] RUMP, S.M. ; KASHIWAGI, M.: Implementation and improvements of affine arithmetic. Submitted for publication. 2014
- [83] SAMARKANOV, D. ; GILLON, F. ; BROCHET, P. ; LALOY, D.: Interval arithmetic tool for optimal design of induction machines. *XXth International Conference on Electrical Machines (ICEM)*. Sep. 2012, S. 1346–1351

- [84] SCHARF, Oliver: Entwurf eines MOSFET-Modells unter Verwendung affiner Arithmetik. Diplomarbeit. Leibniz Universität Hannover - Institut für mikroelektronische Systeme, 2010
- [85] SCHARF, Oliver ; OLBRICH, Markus ; BARKE, Erich: Anwendung der affinen Arithmetik auf das BSIMSOI-Modell zur Simulation von Parameterschwankungen. *GMM Fachbericht Analog 2011; Entwicklung von Analogschaltungen mit CAE-Methoden; Schwerpunkt: Entwurf eingebetteter Sensorsysteme*. 2011, S. 49–54
- [86] SCHARF, Oliver ; OLBRICH, Markus ; BARKE, Erich: Lösungsverfahren für nicht-lineare implizite Gleichungssysteme unter Verwendung von Affiner Arithmetik und Gebietsaufteilungen. *ANALOG 2013: Entwicklung von Analogschaltungen mit CAE-Methoden*. 2013
- [87] SCHARF, Oliver ; OLBRICH, Markus ; BARKE, Erich: Split and Merge Strategies for Solving Uncertain Equations using Affine Arithmetic. *Proceedings of the 8th International Conference on Simulation Tools and Techniques, SIMUTOOLS*. 2015, S. 1–8
- [88] SHICHMAN, H. ; HODGES, D.: Modeling and simulation of insulated-gate field-effect transistor switching circuits. *IEEE Journal of Solid-State Circuits* 3 (3) (Sep. 1968), S. 285–289
- [89] SINGHEE, A. ; FANG, C.F. ; MA, J.D. ; RUTENBAR, R.A.: Probabilistic interval-valued computation: toward a practical surrogate for statistics inside CAD tools. *43rd ACM/IEEE Design Automation Conference*. 2006, S. 167–172
- [90] SKALNA, Iwona: A method for outer interval solution of parametrized systems of linear equations. *Parallel Processing and Applied Mathematics*. Hrsg. von WYRZYKOWSKI, Roman ; DONGARRA, Jack ; KARCZEWSKI, Konrad ; WASNIEWSKI, Jerzy. Bd. 6068. Lecture Notes in Computer Science. Springer, 2004, S. 485–494
- [91] SKALNA, Iwona: Algorithm for min-range multiplication of affine forms. *Numerical Algorithms* 63 (4) (2013), S. 601–614
- [92] SKALNA, Iwona: Direct method for solving parametric interval linear systems with non-affine dependencies. *Parallel Processing and Applied Mathematics*. Hrsg. von WYRZYKOWSKI, Roman ; DONGARRA, Jack ; KARCZEWSKI, Konrad ; WASNIEWSKI, Jerzy. Bd. 6068. Lecture Notes in Computer Science. Springer, 2010, S. 485–494
- [93] SKALNA, Iwona: Evolutionary optimization method for approximating the solution set hull of parametric linear systems. *Numerical Methods and Applications*. Hrsg. von BOYANOV, Todor ; DIMOVA, Stefka ; GEORGIEV, Krassimir ; NIKOLOV, Geno. Bd. 4310. Lecture Notes in Computer Science. Springer, 2007
- [94] SKALNA, Iwona: On checking the monotonicity of parametric interval solution of linear structural systems. *Parallel Processing and Applied Mathematics*. Hrsg. von WYRZYKOWSKI, Roman ; DONGARRA, Jack ; KARCZEWSKI, Konrad ; WASNIEWSKI, Jerzy. Bd. 4967. Springer, 2008, S. 1400–1409

- [95] STAUDT, P.B. ; CARDOZO, N.S.M. ; P. SOARES, R. de: Phase stability analysis using a modified affine arithmetic. *Computers & Chemical Engineering* 53 (2013), S. 190–200
- [96] STRADI-GRANADOS, Benito A.: Interval arithmetic for nonlinear problem solving. *International Journal of Engineering Mathematics* (2013)
- [97] STUBER, Matthew D. ; BARTON, Paul I.: Robust simulation and design using semi-infinite programs with implicit functions. *International Journal of Reliability and Safety* 5 (2011), S. 378–397
- [98] STUBER, Matthew David: Evaluation of process system operating envelopes. Dissertation. Massachusetts Institute of Technology, 2013
- [99] SUN, Ruiyi ; ZHANG, Yan ; CUI, Aijiao: A refined affine approximation method of multiplication for range analysis in word-length optimization. *EURASIP Journal on Advances in Signal Processing* (1) (2014), S. 36
- [100] THANIGAIVELAN, B. ; HAMILTON, T. J. ; POSTULA, A.: A comparison of interval methods in symbolic circuit analysis applications. *ANZIAM*. 2012, S. C1084–C1101
- [101] THANIGAIVELAN, B. ; POSTULA, A. ; HAMILTON, T.J.: A modified MOSFET small-signal model based on affine arithmetic concepts. *Asia Pacific Conference on Postgraduate Research in Microelectronics & Electronics (PrimeAsia)*. Jan. 2009, S. 73–76
- [102] THANIGAIVELAN, Balavelan ; POSTULA, Adam ; YONG, Ding: A self-validated computation approach to symbolic analysis of analog integrated circuits. *Proceedings of the 9th International Workshop on Symbolic Methods and Applications to Circuit Design (SMACD'06)*. 2006
- [103] TIETZE, U. ; SCHENK, Ch.: Halbleiter-Schaltungstechnik. 12. Auflage. Springer, 2002
- [104] UATRONGJIT, Sermsak: Finding all DC solutions of diode and BJT circuits by interval method and modified LP-narrowing technique. *IEEJ Transactions on Electrical and Electronic Engineering* 5 (6) (2010), S. 621–626
- [105] VACCARO, A. ; CAÑIZARES, C. A. ; BHATTACHARYA, K.: A range arithmetic-based optimization model for power flow analysis under interval uncertainty. *IEEE Transactions on Power Systems* 28 (2) (Mai 2013), S. 1179–1186
- [106] VACCARO, A. ; CAÑIZARES, C. A. ; VILLACCI, D.: An affine arithmetic-based methodology for reliable power flow analysis in the presence of data uncertainty. *IEEE Transactions on Power Systems* 25 (2) (Mai 2010), S. 624–632
- [107] VAKILI, S. ; LANGLOIS, J. M. P. ; BOIS, G.: Enhanced precision analysis for accuracy-aware bit-width optimization using affine arithmetic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32 (12) (Dez. 2013), S. 1853–1865

- [108] VASILYEV, D. ; REWIEŃSKI, M. ; WHITE, J.: A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS. *Proceeding of the Design Automation Conference*. Juni 2003, S. 490–495
- [109] VU, Xuan-Ha: Rigorous solution techniques for numerical constraint satisfaction problems. Dissertation. EPFL Lausanne, 2005
- [110] WELLER, Friedrich: Numerische Mathematik für Ingenieure und Naturwissenschaftler. Viewegs Fachbücher der Technik, 1996
- [111] YOXALL, A. ; LANGLEY, J. ; JANSON, R. u. a.: The use of uncertainty analysis for the design of container closures. *Packaging Technology and Science* 19 (3) (2006), S. 139–147
- [112] ZIVKOVIC, Carna ; OLBRICH, Markus ; SCHARF, Oliver u. a.: Hierarchical verification of AMS systems with affine arithmetic forms. *Transactions on Computer-Aided Design of Integrated Circuits and Systems* (Angenommen, im Druck)

A Anhang

A.1 Einstellungen für die Ergebnisse aus Abschnitt 4.2

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 50 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 450 \Omega$, $U_{in} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	$t_{\text{step}} = 1 \mu\text{s}$, $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{in} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 500 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 100 \Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{in} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 300 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{in} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
NLTTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L_1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{in} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$

A.2 Einstellungen für die Ergebnisse aus Abschnitt 4.3

A.2.1 Abbildungen 4.10 und 4.11

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 250 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 166,66 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 466,66 \Omega$, $U_{in} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	$t_{\text{step}} = 500 \text{ns}$, $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{in} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
NLTTL3	$t_{\text{step}} = 20 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 600 \text{pF}$, $L_1 = 10 \text{mH} \pm 500 \mu\text{H}$, $R_1 = 15,9 \text{k}\Omega \pm 795 \Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{in} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 100 \Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{in} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 3,25 \mu\text{s}$, $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{in} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$

A.2.2 Abbildung 4.12

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $t_{\text{end}} = 200 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 50 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 450 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	$t_{\text{step}} = 1 \mu\text{s}$, $t_{\text{end}} = 1 \text{ms}$	$R_1 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 600 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 160 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 160 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,272 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,272 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 2,544 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 6,0 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 800 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 13 \text{ms}$	$C_1 = 12 \text{nF} \pm 600 \text{pF}$, $L_1 = 10 \text{mH} \pm 500 \mu\text{H}$, $R_1 = 15,9 \text{k}\Omega \pm 795 \Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 1 \text{ms}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 100 \Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$

A.2.3 Abbildung 4.13 und BJT_INV_n

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 265 \mu\text{s}$, $n_{\text{iter,max}} = 40$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 45 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 85 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
NLTL	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 150 \text{pF}$, $L_1 = 10 \text{mH} \pm 125 \mu\text{H}$, $R_1 = 15,9 \text{k}\Omega \pm 200 \Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$

A.3 Einstellungen für die Ergebnisse aus Abschnitt 4.4

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $n_{\text{ts,max}} = 7$, $n_{\text{ss,max}} = 0$, Aufteilungsstrategie = <i>Select All</i> , $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 100,0 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 25,0 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	$t_{\text{step}} = 1 \mu\text{s}$, $n_{\text{ts,max}} = 7$, $n_{\text{ss,max}} = 0$, Aufteilungsstrategie = <i>Select All</i> , $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{ts,max}} = 7$, $n_{\text{ss,max}} = 0$, Aufteilungsstrategie = <i>Select All</i> , $t_{\text{end}} = 500 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 100 \Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{ts,max}} = 7$, $n_{\text{ss,max}} = 0$, Aufteilungsstrategie = <i>Select All</i> , $t_{\text{end}} = 300 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $n_{\text{ts,max}} = 7$, $n_{\text{ss,max}} = 0$, Aufteilungsstrategie = <i>Select All</i> , $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L_1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$

A.4 Einstellungen für die Ergebnisse aus Abschnitt 4.5

A.4.1 Abbildung 4.14

$r_{\text{aff}} = 3$, Kriterium 2 = deaktiviert, $n_{\text{ts,max}} = 10$, Aufteilungsstrategie = *Select All*

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
NLTTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
DIFF_AMP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 5 \mu\text{s}$, $n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 175 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{ts,max}} = 30$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$

A.4.2 Abbildung 4.14

$r_{\text{aff}} = 6$, Kriterium 2 = aktiviert, $n_{\text{iter,max}} = 300$, $n_{\text{ts,max}} = 10$, Aufteilungsstrategie = *Select All*

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
NLTTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
DIFF_AMP	Kriterium 2 = aktiviert, $t_{\text{step}} = 5 \mu\text{s}$, $n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 175 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	Kriterium 2 = aktiviert, $t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{iter,max}} = 300$, $n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{iter,max}} = 300$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$

A.4.3 Tabelle 4.5

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	$t_{\text{step}} = 5 \mu\text{s}$, $n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 175 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $n_{\text{ts,max}} = 30$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$

A.5 Einstellungen für die Ergebnisse aus Abschnitt 4.6

A.5.1 Abbildungen 4.16 bis 4.18

$r_{\text{MC}} = 1,000 \cdot 10^3$, $r_{\text{aff}} = 6$, Kriterium 2 = deaktiviert, $n_{\text{iter,max}} = 39$, Zusammenfassungsstrategie = *mergeEach*

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 125 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 5 \mu\text{s}$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 175 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
BP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{iter,max}} = 30$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_{f1} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 100 \Omega$, $R_g = 10 \text{k}\Omega \pm 100 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$

A.6 Einstellungen für die Ergebnisse aus Abschnitt 4.7

A.6.1 Abbildung 4.19

$r_{MC} = 1,000 \cdot 10^3$, $r_{aff} = 5$, r_{aff} für SVF = 1, Kriterium 2 = deaktiviert, $n_{iter,max} = 30$, $n_{ts,max} = 13$, Zusammenfassungsstrategie = *mergeEveryNthStep*, Aufteilungsstrategie = *Select All*

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{step} = 125 \mu s$, $t_{end} = 100 ms$	$R_d = 10 k\Omega \pm 250 \Omega$, $R_{ein} = 5 k\Omega \pm 700 \Omega$, $U_{in} = 2,5 V + 2,5 V \cdot \sin(2 \cdot \pi \cdot 100 Hz \cdot t)$
DIFF_AMP	Kriterium 2 = deaktiviert, $t_{step} = 5 \mu s$, $n_{ts,max} = 10$, $n_{ss,max} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{end} = 500 \mu s$	$R_1 = 7,5 k\Omega \pm 175 \Omega$, $R_2 = 7,5 k\Omega \pm 100 \Omega$, $R_3 = 1 k\Omega \pm 10 \Omega$, $R_4 = 13 k\Omega \pm 10 \Omega$, $R_5 = 2,7 k\Omega \pm 10 \Omega$, $U_{in} = 0,0 V + 100 mV \cdot \sin(2 \cdot \pi \cdot 10,000 kHz \cdot t)$
BP	Kriterium 2 = deaktiviert, $t_{step} = 3,25 \mu s$, $n_{iter,max} = 30$, $n_{ts,max} = 30$, $n_{ss,max} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{end} = 350 \mu s$	$C_1 = 10,0 nF \pm 200 pF$, $C_2 = 10,0 nF \pm 200 pF$, $R_1 = 15,9 k\Omega \pm 1,59 k\Omega$, $R_2 = 15,9 k\Omega \pm 1,59 k\Omega$, $R_3 = 31,8 k\Omega \pm 3,18 k\Omega$, $R_4 = 20 k\Omega \pm 7,5 k\Omega$, $R_5 = 20 k\Omega \pm 1,0 k\Omega$, $U_{in} = 0,0 V + 2 V \cdot \sin(2 \cdot \pi \cdot 500 Hz \cdot (t - 1 ms)) \cdot \sin(10 \cdot \pi \cdot 500 Hz \cdot (t - 1 ms))$
NLTL3	$t_{step} = 40 \mu s$, $t_{end} = 6,6 ms$	$C_1 = 12 nF \pm 1,2 nF$, $L1 = 10 mH \pm 1,0 mH$, $R_1 = 15,9 k\Omega \pm 1,59 k\Omega$, $T = 27^\circ C \pm 10^\circ C$, $U_{in} = 500 mV + 500 mV \cdot \sin(2 \cdot \pi \cdot 500 Hz \cdot t)$
SVF	$t_{step} = 10 \mu s$, $n_{iter,max} = 30$, $t_{end} = 250 \mu s$	$C_1 = 15 nF \pm 1 nF$, $C_2 = 15 nF \pm 1 nF$, $R_1 = 10 k\Omega \pm 1,000 k\Omega$, $R_2 = 10 k\Omega \pm 1,000 k\Omega$, $R_4 = 10 k\Omega \pm 1,000 k\Omega$, $R_{f1} = 10 k\Omega \pm 100 \Omega$, $R_{f2} = 10 k\Omega \pm 100 \Omega$, $R_g = 10 k\Omega \pm 100 \Omega$, $R_q = 10 k\Omega \pm 1,000 k\Omega$, $U_{in} = 0,0 V + 100 mV \cdot \sin(2 \cdot \pi \cdot 10,000 kHz \cdot t)$

A.6.2 Abbildung 4.20

n für Strategie *Merge nth Step* = 2, $r_{MC} = 1,000 \cdot 10^3$, $r_{aff} = 5$, r_{aff} für SVF = 1, Kriterium 2 = deaktiviert, $n_{iter,max} = 40$, $n_{ts,max} = 10$, $n_{ss,max} = 8$, Aufteilungsstrategie = *Select All*

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{step} = 125 \mu s$, $t_{end} = 100 ms$	$R_d = 10 k\Omega \pm 250 \Omega$, $R_{ein} = 5 k\Omega \pm 700 \Omega$, $U_{in} = 2,5 V + 2,5 V \cdot \sin(2 \cdot \pi \cdot 100 Hz \cdot t)$
DIFF_AMP	Kriterium 2 = deaktiviert, $t_{step} = 5 \mu s$, $n_{ts,max} = 10$, $n_{ss,max} = 0$, $t_{end} = 500 \mu s$	$R_1 = 7,5 k\Omega \pm 175 \Omega$, $R_2 = 7,5 k\Omega \pm 100 \Omega$, $R_3 = 1 k\Omega \pm 10 \Omega$, $R_4 = 13 k\Omega \pm 10 \Omega$, $R_5 = 2,7 k\Omega \pm 10 \Omega$, $U_{in} = 0,0 V + 100 mV \cdot \sin(2 \cdot \pi \cdot 10,000 kHz \cdot t)$
NLTL3	$t_{step} = 40 \mu s$, $t_{end} = 6,6 ms$	$C_1 = 12 nF \pm 1,2 nF$, $L1 = 10 mH \pm 1,0 mH$, $R_1 = 15,9 k\Omega \pm 1,59 k\Omega$, $T = 27^\circ C \pm 10^\circ C$, $U_{in} = 500 mV + 500 mV \cdot \sin(2 \cdot \pi \cdot 500 Hz \cdot t)$
SVF	$t_{step} = 10 \mu s$, $n_{iter,max} = 30$, $t_{end} = 250 \mu s$	$C_1 = 15 nF \pm 1 nF$, $C_2 = 15 nF \pm 1 nF$, $R_1 = 10 k\Omega \pm 1,000 k\Omega$, $R_2 = 10 k\Omega \pm 1,000 k\Omega$, $R_4 = 10 k\Omega \pm 1,000 k\Omega$, $R_{f1} = 10 k\Omega \pm 100 \Omega$, $R_{f2} = 10 k\Omega \pm 100 \Omega$, $R_g = 10 k\Omega \pm 100 \Omega$, $R_q = 10 k\Omega \pm 1,000 k\Omega$, $U_{in} = 0,0 V + 100 mV \cdot \sin(2 \cdot \pi \cdot 10,000 kHz \cdot t)$
BP	Kriterium 2 = deaktiviert, $t_{step} = 3,25 \mu s$, $n_{iter,max} = 30$, $n_{ts,max} = 30$, $n_{ss,max} = 0$, $t_{end} = 350 \mu s$	$C_1 = 10,0 nF \pm 200 pF$, $C_2 = 10,0 nF \pm 200 pF$, $R_1 = 15,9 k\Omega \pm 1,59 k\Omega$, $R_2 = 15,9 k\Omega \pm 1,59 k\Omega$, $R_3 = 31,8 k\Omega \pm 3,18 k\Omega$, $R_4 = 20 k\Omega \pm 7,5 k\Omega$, $R_5 = 20 k\Omega \pm 1,0 k\Omega$, $U_{in} = 0,0 V + 2 V \cdot \sin(2 \cdot \pi \cdot 500 Hz \cdot (t - 1 ms)) \cdot \sin(10 \cdot \pi \cdot 500 Hz \cdot (t - 1 ms))$

A.7 Einstellungen für die Ergebnisse aus Abschnitt 4.8

n für Strategie *Merge* $n^{\text{th}} \text{ Step} = 2$, $r_{\text{MC}} = 1,000 \cdot 10^3$, $r_{\text{aff}} = 6$, r_{aff} für SVF = 1, Kriterium 2 = deaktiviert, $n_{\text{iter,max}} = 30$, $n_{\text{ts,max}} = 50$, $n_{\text{ss,max}} = 30$

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 125 \mu\text{s}$, $t_{\text{end}} = 100 \text{ms}$	$R_{\text{d}} = 10 \text{k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 1,2 \text{nF}$, $L1 = 10 \text{mH} \pm 1,0 \text{mH}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
BP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $n_{\text{ts,max}} = 30$, $n_{\text{ss,max}} = 0$, $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 200 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 200 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,59 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 3,18 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 7,5 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,0 \text{k}\Omega$, $U_{\text{in}} =$ $0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_{\text{f1}} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{f2}} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{g}} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{q}} = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
DIFF_AMP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 5 \mu\text{s}$, $n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 0$, $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 175 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$

A.8 Einstellungen für die Ergebnisse aus Abschnitt 4.9

A.8.1 Abbildung 4.21a

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 500 \mu\text{s}$, $t_{\text{end}} = 200 \text{ms}$	$R_{\text{d}} = 10 \text{k}\Omega \pm 50 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 450 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	$t_{\text{step}} = 1 \mu\text{s}$, $t_{\text{end}} = 1 \text{ms}$	$R_1 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 600 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 160 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 160 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 1,272 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 1,272 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 2,544 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 6,0 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 800 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 13 \text{ms}$	$C_1 = 12 \text{nF} \pm 600 \text{pF}$, $L1 = 10 \text{mH} \pm 500 \mu\text{H}$, $R_1 = 15,9 \text{k}\Omega \pm 795 \Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $t_{\text{end}} = 1 \text{ms}$	$C_1 = 15 \text{nF} \pm 1 \text{nF}$, $C_2 = 15 \text{nF} \pm 1 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 100 \Omega$, $R_2 = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{f1}} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{f2}} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{g}} = 10 \text{k}\Omega \pm 100 \Omega$, $R_{\text{q}} = 10 \text{k}\Omega \pm 1,000 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$

A.8.2 Abbildungen 4.22 und 4.23

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 265 \mu\text{s}$, $n_{\text{iter,max}} = 40$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 45 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 85 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
NLTL	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 150 \text{pF}$, $L1 = 10 \text{mH} \pm 125 \mu\text{H}$, $R_1 = 15,9 \text{k}\Omega \pm 200 \Omega$, $T = 27^\circ\text{C} \pm 10^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$

A.8.3 Abbildungen 4.24 und 4.25

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 250 \mu\text{s}$, $n_{\text{iter,max}} = 39$, $t_{\text{end}} = 100 \text{ms}$	$R_d = 10 \text{k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{V} + 2,5 \text{V} \cdot \sin(2 \cdot \pi \cdot 100 \text{Hz} \cdot t)$
DIFF_AMP	Kriterium 2 = aktiviert, $\Delta s_{\text{max}} = 1 \cdot 10^{-12}$, $t_{\text{step}} = 500 \text{ns}$, $n_{\text{iter,max}} = 300$, $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{k}\Omega \pm 150 \Omega$, $R_2 = 7,5 \text{k}\Omega \pm 150 \Omega$, $R_3 = 1 \text{k}\Omega \pm 15 \Omega$, $R_4 = 13 \text{k}\Omega \pm 15 \Omega$, $R_5 = 2,7 \text{k}\Omega \pm 15 \Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
NLTL3	$t_{\text{step}} = 20 \mu\text{s}$, $n_{\text{iter,max}} = 39$, $t_{\text{end}} = 6,6 \text{ms}$	$C_1 = 12 \text{nF} \pm 900 \text{pF}$, $L1 = 10 \text{mH} \pm 750 \mu\text{H}$, $R_1 = 15,9 \text{k}\Omega \pm 1,1925 \text{k}\Omega$, $T = 27^\circ\text{C} \pm 15^\circ\text{C}$, $U_{\text{in}} = 500 \text{mV} + 500 \text{mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot t)$
SVF	Kriterium 2 = aktiviert, $\Delta s_{\text{max}} = 1 \cdot 10^{-12}$, $t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{iter,max}} = 300$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{nF} \pm 1,5 \text{nF}$, $C_2 = 15 \text{nF} \pm 1,5 \text{nF}$, $R_1 = 10 \text{k}\Omega \pm 150 \Omega$, $R_2 = 10 \text{k}\Omega \pm 1,500 \text{k}\Omega$, $R_4 = 10 \text{k}\Omega \pm 150 \Omega$, $R_{f1} = 10 \text{k}\Omega \pm 150 \Omega$, $R_{f2} = 10 \text{k}\Omega \pm 150 \Omega$, $R_g = 10 \text{k}\Omega \pm 150 \Omega$, $R_q = 10 \text{k}\Omega \pm 1,500 \text{k}\Omega$, $U_{\text{in}} = 0,0 \text{V} + 100 \text{mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{kHz} \cdot t)$
BP	$t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{iter,max}} = 39$, $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{nF} \pm 300 \text{pF}$, $C_2 = 10,0 \text{nF} \pm 300 \text{pF}$, $R_1 = 15,9 \text{k}\Omega \pm 2,385 \text{k}\Omega$, $R_2 = 15,9 \text{k}\Omega \pm 2,385 \text{k}\Omega$, $R_3 = 31,8 \text{k}\Omega \pm 4,77 \text{k}\Omega$, $R_4 = 20 \text{k}\Omega \pm 11,25 \text{k}\Omega$, $R_5 = 20 \text{k}\Omega \pm 1,5 \text{k}\Omega$, $U_{\text{in}} =$ $0,0 \text{V} + 2 \text{V} \cdot \sin(2 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{Hz} \cdot (t - 1 \text{ms}))$

A.9 Einstellungen für die Ergebnisse aus Abschnitt 4.10

A.9.1 Gemeinsame Einstellungen

$n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 10$, $r_{\text{MC}} = 1,000 \cdot 10^3$, $r_{\text{aff}} = 6$, r_{aff} für SVF = 1

A.9.2 Unoptimierte Einstellungen

$\Delta s_{\text{max}} = 10 \cdot 10^{-6}$, Kriterium 2 = deaktiviert, $n_{\text{iter,max}} = 15$, Zusammenfassungsstrategie = *Merge Each*, Aufteilungsstrategie = *Select All*

A Anhang

Beispiel	Einstellungen	Parameter
BJT_INV	$t_{\text{step}} = 125 \mu\text{s}$, $t_{\text{end}} = 100 \text{ ms}$	$R_d = 10 \text{ k}\Omega \pm 250 \Omega$, $R_{\text{ein}} = 5 \text{ k}\Omega \pm 700 \Omega$, $U_{\text{in}} = 2,5 \text{ V} + 2,5 \text{ V} \cdot \sin(2 \cdot \pi \cdot 100 \text{ Hz} \cdot t)$
DIFF_AMP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 5 \mu\text{s}$, $n_{\text{ts,max}} = 10$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 500 \mu\text{s}$	$R_1 = 7,5 \text{ k}\Omega \pm 175 \Omega$, $R_2 = 7,5 \text{ k}\Omega \pm 100 \Omega$, $R_3 = 1 \text{ k}\Omega \pm 10 \Omega$, $R_4 = 13 \text{ k}\Omega \pm 10 \Omega$, $R_5 = 2,7 \text{ k}\Omega \pm 10 \Omega$, $U_{\text{in}} = 0,0 \text{ V} + 100 \text{ mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{ kHz} \cdot t)$
NLTL3	$t_{\text{step}} = 40 \mu\text{s}$, $t_{\text{end}} = 6,6 \text{ ms}$	$C_1 = 12 \text{ nF} \pm 1,2 \text{ nF}$, $L1 = 10 \text{ mH} \pm 1,0 \text{ mH}$, $R_1 = 15,9 \text{ k}\Omega \pm 1,59 \text{ k}\Omega$, $T = 27 \text{ }^\circ\text{C} \pm 10 \text{ }^\circ\text{C}$, $U_{\text{in}} = 500 \text{ mV} + 500 \text{ mV} \cdot \sin(2 \cdot \pi \cdot 500 \text{ Hz} \cdot t)$
BP	Kriterium 2 = deaktiviert, $t_{\text{step}} = 3,25 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $n_{\text{ts,max}} = 30$, $n_{\text{ss,max}} = 0$, Zusammenfassungsstrategie = <i>mergeEach</i> , $t_{\text{end}} = 350 \mu\text{s}$	$C_1 = 10,0 \text{ nF} \pm 200 \text{ pF}$, $C_2 = 10,0 \text{ nF} \pm 200 \text{ pF}$, $R_1 = 15,9 \text{ k}\Omega \pm 1,59 \text{ k}\Omega$, $R_2 = 15,9 \text{ k}\Omega \pm 1,59 \text{ k}\Omega$, $R_3 = 31,8 \text{ k}\Omega \pm 3,18 \text{ k}\Omega$, $R_4 = 20 \text{ k}\Omega \pm 7,5 \text{ k}\Omega$, $R_5 = 20 \text{ k}\Omega \pm 1,0 \text{ k}\Omega$, $U_{\text{in}} = 0,0 \text{ V} + 2 \text{ V} \cdot \sin(2 \cdot \pi \cdot$ $500 \text{ Hz} \cdot (t - 1 \text{ ms})) \cdot \sin(10 \cdot \pi \cdot 500 \text{ Hz} \cdot (t - 1 \text{ ms}))$
SVF	$t_{\text{step}} = 10 \mu\text{s}$, $n_{\text{iter,max}} = 30$, $t_{\text{end}} = 250 \mu\text{s}$	$C_1 = 15 \text{ nF} \pm 1 \text{ nF}$, $C_2 = 15 \text{ nF} \pm 1 \text{ nF}$, $R_1 = 10 \text{ k}\Omega \pm 1,000 \text{ k}\Omega$, $R_2 = 10 \text{ k}\Omega \pm 1,000 \text{ k}\Omega$, $R_4 = 10 \text{ k}\Omega \pm 1,000 \text{ k}\Omega$, $R_{f1} = 10 \text{ k}\Omega \pm 100 \Omega$, $R_{f2} = 10 \text{ k}\Omega \pm 100 \Omega$, $R_g = 10 \text{ k}\Omega \pm 100 \Omega$, $R_q = 10 \text{ k}\Omega \pm 1,000 \text{ k}\Omega$, $U_{\text{in}} = 0,0 \text{ V} + 100 \text{ mV} \cdot \sin(2 \cdot \pi \cdot 10,000 \text{ kHz} \cdot t)$

A.9.3 Optimierte Einstellungen

Beispiel	Einstellungen
BJT_INV	$\Delta s_{\text{max}} = 10 \cdot 10^{-6}$, Kriterium 2 = deaktiviert, $n_{\text{iter,max}} = 39$, Zusammenfassungsstrategie = <i>Merge Each</i> , Aufteilungsstrategie = <i>Select By Deviation</i>
DIFF_AMP	$\Delta s_{\text{max}} = 90 \cdot 10^{-6}$, Kriterium 2 = aktiviert, $n_{\text{iter,max}} = 300$, Zusammenfassungsstrategie = <i>Try Root First</i> , Aufteilungsstrategie = <i>Select By Sensitivity And Deviation</i>
NLTL3	$\Delta s_{\text{max}} = 10 \cdot 10^{-6}$, Kriterium 2 = deaktiviert, $n_{\text{iter,max}} = 39$, Zusammenfassungsstrategie = <i>Merge Each</i> , Aufteilungsstrategie = <i>Select By Sensitivity And Deviation</i>
BP	$\Delta s_{\text{max}} = 10 \cdot 10^{-6}$, Kriterium 2 = deaktiviert, $n_{\text{iter,max}} = 39$, Zusammenfassungsstrategie = <i>Merge Each</i> , Aufteilungsstrategie = <i>Select By Deviation</i>
SVF	$\Delta s_{\text{max}} = 90 \cdot 10^{-6}$, Kriterium 2 = aktiviert, $n_{\text{iter,max}} = 300$, Zusammenfassungsstrategie = <i>Merge Each</i> , Aufteilungsstrategie = <i>Select By Deviation</i>

Lebenslauf

Persönliche Daten

Oliver Scharf
Geboren am 17.10.1984
in Hannover

Bildungsweg

1991 - 1997 Grundschule und Orientierungsstufe, Springe
1997 - 2004 Otto-Hahn-Gymnasium Springe, Abitur
2004 - 2010 Studium der Elektrotechnik an der Leibniz Universität Hannover

Berufspraxis

2010 - 2015 Wissenschaftlicher Mitarbeiter am Institut für Mikroelektronische Systeme, Fachgebiet Entwurfsautomatisierung, Leibniz Universität Hannover
Seit 2016 Entwicklungsingenieur bei der Baker Hughes Inteq GmbH, Celle