DISSERTATION

Design Refinement of Communication Systems Applying Range Based Simulations

Submitted at the Faculty of Electrical Engineering and Information Technology, Vienna University of Technology in partial fulfillment of the requirements for the degree of Doktor der technischen Wissenschaften (equals Ph.D.)

under supervision of

Univ.-Prof. Dr. Christoph Grimm
Institut number: 384
Institute of Computer Technology
Vienna University of Technology

and

Prof. Dr. Lars Hedrich
Electronic Design Methodology
Department of Computer Science
J.W.Goethe University Frankfurt/Main

by

Dipl.-Ing.(FH) Florian Schupfer
Matr.Nr. 9626886
Universumstrasse 24/6, 1200 Vienna

Date ______________________________

Abstract

Electronic systems and circuits are typically designed to fulfill a specified ideal behavior and are considered to be implemented by their nominal quantities. Realizations of system designs however always deviate in their system parameters from the ideal implementation. Process variations deviate component properties, part tolerances introduce variations to the system realization and general uncertainties add deviations to the system behavior. System quality improvements should result in a robust system behavior and primarily targets communication systems within this work. The variation of system parameters especially occurs in analog and mixed-signal systems and raises the question on how to efficiently analyze and improve their behavior. The behavior of analogue systems is not only specified by the nominal design parameters but is also influenced by parameter variations caused by implementation decisions and process variations. Traditional multi-run simulations do not keep pace in their simulation efficiency with the rising demand for computation power. Classical numeric simulations loose the correlation to causing factors. The numerical result simply provides a scalar quantity which does not hold any information on the causing contributor. System refinement techniques are difficult to apply as initially correlations or sensitivities have to be determined to achieve a deterministic optimization goal. A system analysis is considered to be the first and crucial step in a refinement design process. Semi-symbolic simulations are a novel simulation technique that provide the potential to avoid and overcome these restrictions. The combined numerical and symbolic modeling and simulation approach allows to compute the simulation quantities simultaneously for a complete range of varying system parameters. This reduces the multi-run effort of traditional numeric simulations to a single simulation run with the cost of an increased computation complexity. The symbolic representatives of the resulting simulation quantities keep the correlations to the system parameters and allow a backward behavior analysis. The objectives within this thesis are to create a simulation and analysis environment which allows a parameter impact estimation of deviated system models. Deviations of system quantities and parameters are modeled by range descriptions and considered in a following simulation step. Identified refinement candidates should be updated/modified iteratively to increase the system quality and to improve the robustness and reliability of the designed systems. The symbolic nature of deviation representations is considered to support the identification of refinement parameters. The range based system response should be decomposed into the contributing sub ranges giving, a measure on the impact of every sub ranges associated deviation effect on the overall behavior. A simulation (semi-symbolic) guided refinement candidate identification allows an efficient system quality improvement. All methodologies are finally combined in the "'MARC refinement design flow"' which supports the semi-symbolic simulation, analysis and deterministic identification of refinement candidates in one environment.

Kurzfassung

Elektronische Systeme und Schaltungen werden typischerweise so entworfen, dass sie ein spezifiziertes, ideales Verhalten erfüllen und mit nominalen Systemgrößen implementiert werden. Realisierungen von Systemen haben aber immer Abweichungen ihrer Systemparameter von der idealen Implementierung zur Folge. Prozess Variationen verschieben Bauteileigenschaften, Toleranzen erzeugen Abweichungen in der Realisierung und generelle Unsicherheiten verursachen Abweichungen des Systemverhaltens. Verbesserungen der Systemqualität sollen zu einem robusteren Verhalten führen und in dieser Arbeit speziell für Kommunikationssysteme gezeigt werden. Eine Schwankung von Systemparametern tritt insbesonders bei analogen und mixed-signal Systemen auf. Das Verhalten von analogen Systemen ist nicht nur durch seine nominellen Designparameter spezifiziert, es ist auch durch Parameterabweichungen die durch Implementierungsentscheidungen und Prozessvariationen verursacht werden beeinflusst. Traditionelle Multi-Run Simulationen können mit den gestiegenen Leistungsanforderungen in ihrer Simulationseffizienz nicht schritthalten. Klassische numerische Simulationen verlieren die Korrelation zu den beeinflussenden Faktoren. Die numerischen Resultate liefern einfach skalare Größen welche keine Informationen über Ursachen beinhalten. Techniken zur Verbesserung der Systemeigenschaften sind schwierig anzuwenden, da zuerst Korrelationen und Empfindlichkeiten bestimmt werden müssen um ein deterministische Optimierung zu erreichen. Eine Systemanalyse wird daher als erster und wichtigster Schritt erachtet um einen Designprozess welcher auf eine Verbesserung der Systemeigenschaften basiert, zu ermöglichen. Semi-Symbolische Simulationen sind eine neuartige Simulationstechnik welche das Potential haben diese Einschränkungen zu überwinden. Der kombinierte numerische und symbolische Modellierungs- und Simulationsansatz ermöglicht es die Simulationsgrößen gleichzeitig für einen ganzen Bereich von variierenden Systemparametern zu berechnen und damit den Multi-Run Aufwand auf einen einzigen Simulationslauf zu reduzieren. Die symbolische Darstellung des Simulationsergebnisses beinhaltet weiterhin Korrelationen zu den Systemparametern und ermöglicht eine inverse Verhaltensanalyse. Die Ziele dieser Arbeit sind es eine Simulations- und Analyseumgebung zu schaffen, welche eine integrierte Parameterauswirkungsabschätzung von Systemen mit Parameterabweichungen ermöglicht. Abweichungen von Systemgrößen und Parametern werden durch eine Bereichsbeschreibung modelliert und im folgenden Simulationsschritt berücksichtigt. Identifizierte Verfeinerungskandidaten sollen schrittweise modifiziert werden und dadurch die Systemqualität beziehungsweise die Robustheit und Zuverlässigkeit des Systems verbessert werden. Die symbolische Eigenschaft der Abweichungsdarstellung unterstützt die Identifikation von Verbesserungskandidaten. Die bereichsbasierte Systemantwort soll in seine beitragenden Unterbereiche zerlegt werden, um ein Maß für den Beitrag jedes Abweichungseffektes zum gesamten Systemverhalten zu bekommen. Eine durch eine Simulation (Semi-Symbolisch) geleitete Auswahl von Verbesserungskandidaten erlaubt eine effiziente Verbesserung der Systemqualität. Alle Methodiken werden schließlich in einem "'MARC refinement design flow"' kombiniert welcher Semi-Symbolische Simulationen, die Analyse und eine deterministische Identifikation von Verbesserungskandidaten in einer Umgebung erlaubt.

Acknowledgements

As first and most important person I want to thank my partner Katharina for her never ending support during the last years of scientific work and thesis writing.
My special thank goes to Prof. Dr. Christoph Grimm who always encouraged me in my work and who offered me the chance to work in the fantastic working environment I found at the Institute of Computer Technology. My thank also goes to Prof. Dr. Lars Hedrich who was willed to supervise this work.
My thanks go to my friends and colleagues Peter Brunmayr and Jiong Ou for all the valueable discussions and distractions. I have to specially thank my project colleagues Carna Radojicic and Michael Rathmair for their contribution to this work, the discussion about challenging problems and their pleasant collaboration. Last but not least I want to thank all my other colleagues at the Institute of Computer Technology for the interesting last years.

Table of Contents

Abbreviations

AA	Affine Arithmetic
AAF	Affine Arithmetic Form
AC	Alternating Current
ACI	Adjacent Carrier Interference
ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AMS	Analog/Mixed Signal
AM	Amplitude Modulation
BDD	Binary Decision Diagram
BER	Bit Error Rate
BSIM	Berkley Short-channel IGFET Model
CPS	Cyber Physical Systems
CPU	Central Processing Unit
DAE	Differential Algebraic Equation
DC	Direct Current
DE	Discrete Event
DFT	Discrete Fourier Transformation
DoE	Design of Experiments
DPE	Data Processing Engine
DSP	Digital Signal Processor
ECU	Electronic Control Unit
ELN	Electrical Linear Networks
EVA	Extreme Value Analysis
EVM	Error Vector Magnitude
HW	Hardware
IA	Interval Arithmetic
ICI	Inter Channel Interference
IGFET	Insulated Gate Field Effect Transistor
ISI	Inter Symbol Interference
LGPL	GNU Lesser General Public License
LNA	Low Noise Amplifier
LO	Local Oscillator
LP	Low Pass
LSF	Linear Signal Flow
LTF	Laplace Transfer Function
MARC	Methodology for Accurate and Robust Communication Systems
MC	Monte Carlo Simulation
MNA	Modified Nodal Analysis
MoC	Model of Computation
MOSFET	Metal Oxide Semiconductor Field Effect Transistor

NP	Nondeterministic Polynomial Time
OBDD	Ordered Binary Decision Diagram
OSCI	Open SystemC Initiative
RF	Radio Frequency
RSS	Root Square Sum
RTL	Register Transfer Level
SAG	Simplification After Generation
SAT	Satisfiablity
SBG	Simplification Before Generation
SDF	Synchronous Data Flow
SDG	Simplification During Generation
SDR	Signal to Deviation Ratio
SIR	Signal to Interferer Ratio
SNR	Signal to Noise Ratio
SoC	Systems on Chip
SPICE	Simulation Program with Integrated Circuit Emphasis
STE	Symbolic Trajectory Evaluation
SW	Software
TDF	Timed Data Flow
TLM	Transaction Level Modeling
VS	Variation Space

1 Introduction

1.1 Problem statement

Our modern society and environment is steeped by technological solutions and applications. A considerable group of our every day devices are embedded systems which summarize in environment "‘embedded"’ electronic systems. A more modern and specific term for such devices is a "‘cyber physical system"’ which especially indicates the common interaction of these systems with their physical environment. The integration of cyber physical systems into our every days environment poses a potential safety issue to the applications and the health of operating persons. Electronic control units (ECU) for instance are used to operate originally mechanical driven processes in the automotive or aeronautical field. If braking, steering or other essential operating processes fail due to system errors, persons are most likely to get harmed. Apart from strictly safety critical systems the omnipresent presence of electronic devices with their most probably implemented wireless communication connection introduce a new source of potential distortion. Cross interference potentially causes the systems to at least do not meet the projected behavior or worse to completely fail. The reliability and robustness of implemented systems becomes a central constraint during the design process.

Cyber physical systems combine the "‘embedded system"’ with interfaces to its surrounding physical environment and typically comprise of several heterogeneous subsystems (analogue HW/digital HW/SW) which are functionally interwoven. Software signal processing algorithms are used to reconstruct distorted signal characteristics, analogue pre-processing units transform the received signals to processable representations and hand them over to digital demodulation units. Every design weakness influences the behavior of the adjacent domain. The system can be classified as mixed-signal system, as the system contains both analogue as well as digital signal representations.

The analysis and verification of digital sub domains of a system (either software or digital hardware) already reached a mature level. Formal methods have been introduced which already support industrial design processes and help to improve the reliability of purely digital systems. For instance model checking [BCM+90] allows to verify the behavior of modeled and abstracted systems with respect to property descriptions using temporal logic. Nevertheless, the analogue domain lacks of an efficient formalized verification and refinement methodology.

Additionally, the behavior of analogue systems is not only specified by the nominal design parameters but also are influenced by parameter variations caused by implementation decisions and process variations. All parameters of the analogue subsystem consist of a nominal design parameter associated with a potential variation of this value. The traditional and most widely used methodology for analyzing the behavior of deviated analogue and mixed-signal systems are multi-run simulations. Every simulation run produces a specific system behavior caused by the respective parameter values. The systems parameters are consequently adapted either by random selection [Rub81] or by guided choices [Mh02] within the potential variation boundaries. The resulting set of behavior characteristics is following analyzed to estimate the effective system behavior of the deviated system model. In order to obtain a solid statistical confidence a significant number of simulation runs providing distinct behavior characteristics have to be performed.

Analogue electronic systems still represent important interfacing and pre-processing units. Contrary to last decades assumption that digital systems take over most analogue functionalities, the fact that cyber physical systems are embedded and interact with their physical environment still requires various analogue subsystems [GRB+06], [4]. Analogue systems do not scale down compare able to digital circuits with advances in the semiconductor industry. Analysis and verification methodologies still exist at an early stage of development if not remain on an academic level. The growth rates of cyber physical systems are driven by the desire to add computation intelligence to an increasing number of our daily life applications. Hence, the need for efficient and applicable analysis and verification methodologies also become of central interest [1].
Analogue and mixed-signal sub domains are typically evaluated by conducting SPICE simulations [6]. Models of predefined components are thereby wired up to represent the intended circuit characteristic. This circuits netlist is following analyzed for obtaining the respective system equations in a differential equation form and finally solved by using approximation algorithms. A typical device model for SPICE simulations is the so called BSIM (Berkley Short-channel IGFET Model) MOSFET transistor model. Figure 1.1 [CH02] shows the typical application of such a BSIM model.

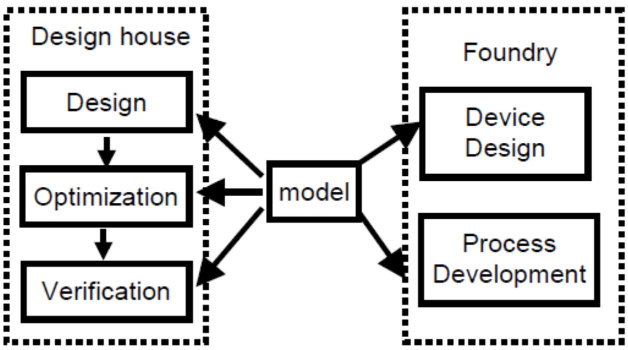

Figure 1.1: BSIM model application

The BSIM model acts as connector between the semiconductor foundry and the system designer with his need for realistic system evaluations Figure. Real physical effects are abstracted and added to the device model in the equational description, configurable with the device parameters. Advances in semiconductor integration processes lead to additional sub micron physical effects which have to be considered in improved device models. Moreover the influence of process variations on the integrated systems is increasing with higher integration densities and thus the minimal achievable variation figures are rising.

Figure 1.2 [CH02] shows the evolution of the BSIM device models since their creation in 1987. Every progress in integration techniques required an increased accuracy and complexity of the simulation models to capture all influencing effects. This higher complexity of the device models directly translates into an increased number of configuration parameters. An additional impact of the success in device shrinking is a significant increase of the system parameters variations caused by process deviations, inaccuracies in sub-micron structures and matching imperfections. As previously mentioned, multi-run simulations are the traditional means to evaluate and verify analogue and mixed-signal systems with deviation effects. Unfortunately, the increased parameter variations and also the general increased number of model parameters twofold negatively influence the number of necessary simulation runs in multi-run approaches. This trend seriously effects the simulation performance in industrial verification applications.

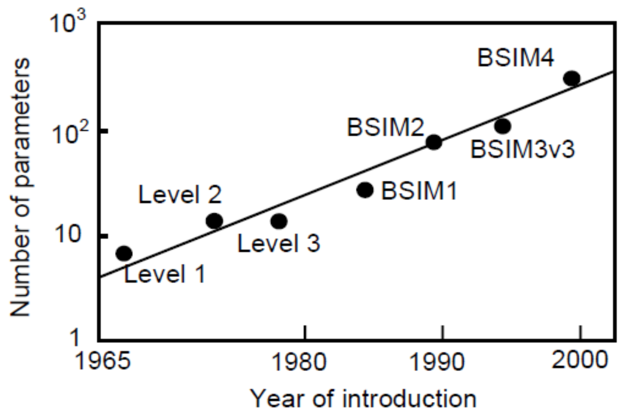

Figure 1.2: BSIM transistor model evolution

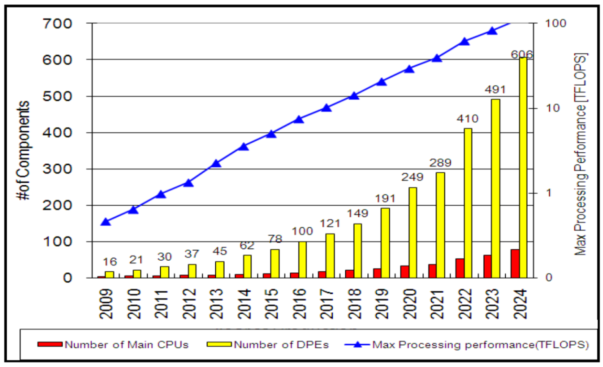

Figure 1.3: ITRS SoC complexity prognosis

Figure 1.3 predicts the complexity of typical cyber physical systems (or the embedded system subclass) over a time period of two decades [3]. The figure visualizes the trend for embedded systems to a steady increase in system complexity. The detailed graphs show the increase in the number of integrated CPUs, the correlated increase in data processing engines (DPE) and the resulting trend in overall processing performance. As the cyber physical systems interact with their physical environment, processing performance gains directly correlate with intensified analogue and mixed-signal interfacing usage. Thus, although the design process and especially the analysis and verification techniques for analogue and mixed-signal systems do not follow the down scaling trend of digital domains, analog applications tend to follow the complexity rise.

Moore's Law predicts in the most optimistic scenario a doubling of the transistor numbers in integrated circuits every 18 month. Following Moore's Law a doubling of the integrated transistors scales to an 40 % increase in computation power [OH05].

$$P(m) = T(0) * 0.4 * 2^{\frac{m}{18}} \tag{1.1}$$

Equation 1.1 predicts Moore's computation power evolution where $P(m)$ represents the predicted computation power given by Moore's Law, $T(0)$ an initial transistor figure and m the time under investigation in months.

On the other hand the simulation performance is negatively influenced by

- a steady increase in overall system complexity - goes along with an increased number of system components

- a steady increase in device models accuracy - increases the amount of parameters which are used to configure and specify the device models
- an increase in part variations caused by device shrinking - the increased variance of integrated components cause an increase in parameter variations which have to be considered in the device models

which form in their combined effect a sub-space to the parameter space that has to be analyzed. This negative influence on the simulation performance can be considered as opposing factor to the computation power which is denoted in this scope as simulation effort. As intermediate characteristic the variation space (VS) is defined as the sub-space constituted by the combined simulation effects,

$$VS \propto Variance(m) * Parameters(n) * Complexity(u) \tag{1.2}$$

$$\text{Simulation Effort(m,n,u)} = O(m * n * u) \tag{1.3}$$

where equation 1.2 illustrates that the single, approximately linear increasing (deduced from figure 1.2 1.3) effects constitute to a higher dimensional variation space (VS). The increase in system complexity influences the number of "‘embedded"’ integrated components in the system. Each component again consists of plenty of defining parameters whose amount additionally increase with improvements in device models accuracy. The success in system miniaturization finally leads to a steady increase in minimum achievable parameter variance causing deviations of the system parameters from their designed intended values. The variation space is formed by the combination of all three influencing effects and follows an exponential growth when enhanced. This variation space spans a sub-space of the original systems parameter space and influences the simulation effort (number of simulation runs for multi-run approaches) in an exponential way.

Comparing the advances in computation power gain of equation 1.1 with the higher order, exponential growth characteristic of the simulation effort in equation 1.3 reveals a spread between demands and feasibility. Multi-run simulations face a simulation performance gap which interferes with the desire for an efficient analysis and verification methodology. Existing exhaustive simulation methods reach restricting barriers and advanced or even novel analysis methodologies are required to keep pace with the system evolution trends.

1.2 Objectives and realization

The unabated trend to integrate analogue and mixed-signal systems into cyber physical systems raise the question on how to efficiently analyze and improve their behavior. Traditional multi-run simulations do not keep pace in their simulation efficiency with the rising demand for computation power. Classical numeric simulations loose the correlation from their originating behavior contributors. The numerical result simply provides a scalar quantity which does not hold any information on the causing contributors. System refinement techniques are difficult to apply as initially correlations or sensitivities have to be determined to achieve a deterministic optimization goal. A system analysis is considered to be the first and crucial step in a refinement design process. The system behavior has to be evaluated and assessed for its improvement potential. Sensitivities of system parameters are vital at this point to be able to rate the impact factor of every system quantity on the analyzed system behavior. System quantities and their parameters do not appear with their nominal designed value when considering real realizations for the evaluation and refinement process. Deviations from the ideal system parameters always influence the behavior of design implementations. For that reason a simulation and analysis methodology is demanded that not just simulates ideal system behavior but also considers deviation effects in the evaluation process.

Semi-symbolic simulations [GHW04] are a novel simulation technique that provide the potential to avoid and overcome these restrictions. The combined numerical and symbolic modeling and simulation approach allows to compute the simulation quantities simultaneously for a complete range of varying system parameters. This reduces the multi-run effort to a single simulation run with the cost of an increased computation complexity. The symbolic representatives of the resulting simulation quantities keep the correlations to the system parameters and allow a backward behavior analysis. Problematic system behavior can be analyzed for its worst contributing source and thus the systems be refined well-directed in its properties to improve the quality and robustness of the system. The symbolic nature of the simulation provides an implicit sensitivity measure which allows the analysis of system quantities for their correlations and dependency on the original deviation sources.

Semi-symbolic simulations have been extended in their ability for system analysis in recent years [SKG+11, GOGB07, OSG11a]. Range based system quantities have been transformed to the frequency domain which allows the full field of spectral analysis techniques applied on them. Semi-symbolic simulations have been used for a reachability analysis, have been extended from the system level to transistor level and have been applied on DSP systems to determine the optimal bit width in the system when considering the quantization error in the design. All these extensions simplify the analysis of systems under deviation influence and broadens the usage of semi-symbolic simulations.

1.2.1 Hypothesis

The research question within this thesis is if there is a mathematical or deterministic way to identify system parameter dependencies within a system and to determine their single impact on the overall system behavior. Traditional system design methods rely on numerical system simulations which analyze the behavior by use of corner case analyzes or Monte Carlo simulations. All these methods lack a possibility to trace back the cause of erratic system behavior to its causing system parameter. The hypothesis is that a semi-symbolic simulation based on Affine Arithmetic [FS97] or range arithmetic [GOGB07] is capable of computing the system output quantities as functions of the original parameters, deviations or uncertainties. By means of this method heterogeneous systems are simulate able in a symbolic way, showing finally the impact of every single parameter and design decision on the overall system behavior. Contrary to numerical methods which rely on statistical methods to analyze system behavior (Monte Carlo simulation) a semi-symbolic simulation guarantees the containing of the **entirely** reachable (caused by the modeled parameter variations) output quantities in the range based simulation result. One single simulation run provides the pessimistic, guaranteed system behavior caused by the considered parameter variations and their combinations. Multi-run simulations in contrast require a high number of simulation runs (N) to achieve an acceptable statistical confidence interval, where the variance scales down with $\sqrt{N}$ [TD06]. The objectives within this thesis are to create a simulation and analysis environment which allows an integrated parameter impact estimation of deviated system models. Deviations of system quantities and parameters are modeled by range descriptions and considered in a following simulation step. Identified refinement candidates should be updated/modified iteratively and the system quality increased to improve the robustness and reliability of the designed systems. As stated in the hypothesis, the starting point of the thesis is the usage of Affine Arithmetic for semi-symbolic system simulations [GHW04, GOGB07, FS97]. The symbolic nature of deviation representations is considered to support the identification of refinement parameters deduced from the range based system response. The range based quantities should be disassembled into the contributing sub ranges giving a measure on the impact of every sub ranges associated deviation effect on the overall behavior. A simulation (semi-symbolic) guided refinement candidate identification will allow an efficient system quality improvement with simultaneously minimizing the number of simulation runs.

1.2.2 Contribution to the field

Among publications in related research fields, especially the work targeting semi-symbolic simulations and in particular the simulation of systems under process and parameter variations contribute to the thesis topic. Semi-symbolic simulations have been already addressed in previous publications [GHW04] but utilizing the particular symbolic properties for a refinement design flow and analysis enhancements into the frequency domain extend their field of applications.

This work is based on the following publications:

F.Schupfer and C.Grimm. Towards more Dependable Verification of Mixed-Signal Systems. In: *Dagstuhl Seminar Proceedings:Verification over discrete-continuous boundaries*, 2010.

F.Schupfer, M.Kärgel, C.Grimm, M.Olbrich and E.Barke. Towards Abstract Analysis Techniques for Range Based System Simulations. In:*System Specification and Design Languages: Selected Contributions from Fdl 2012*. pp105-121, Springer New York,2011.

F.Schupfer, M.Svarc, C.Radojicic and C.Grimm. A Range Based System Simulation and Refinement Design Flow. In:*Industry Adoption of the SystemC AMS standard*, 2011.

F.Schupfer, C.Radojicic, J.G.O.Wenninger and C.Grimm. System Refinement Design Flow based on Semi-Symbolic Simulations. In:*Proceedings of the 10th IEEE Africon (2011)*, 2011.

J.Ou, F.Schupfer and C.Grimm. System Level Communication System Design using Extended SystemC AMS Building Block Library. In:*Tagungsband Austrochip 2011*, pp39-43, 2011.

J.Ou, F.Schupfer and C.Grimm. Modeling Quantization Error of DSP Systems using SystemC AMS. In:*Proceedings of the VW FEDA 2011*. Southampton,UK,2011.

[SG10] summarizes the idea of semi-symbolic simulations, presents the advances reached in recent years and introduces potential enhancements to the analysis capabilities. The full variety of range based simulations is discussed which comprise of simulations on the abstracted system level down to the elementary circuit description on the transistor level. [SKG+11] introduces a comprehensive discussion of a Fourier Transform for range based systems. Analyzing a system behavior solely in the time domain does not suffice the complex analysis challenges and requirements of modern systems. This applies especially when considering frequency translation structures as used in communication systems. The introduction of the frequency domain for range based systems offers a completely new class of analysis methodologies and fundamentally enhances the analysis potential of semi-symbolic simulations. A range based refinement design flow was first introduced in [SSRG11] and enhanced in [SRWG11]. The "‘symbolic"’ nature of semi-symbolic simulations is used to identify appropriate refinement candidates and to iteratively refine the system properties. The refinement procedure is pursued until an acceptable system robustness and reliability is reached. Quality metrics for deciding about the systems performance remain a crucial weakness but first approaches are presented. Modeling a range based system model remains a challenging and incomplete solved task. To improve the modeling techniques a very basic building block library was created and introduced in [OSG11b]. [OSG11a] finally describes the modeling approach for quantization effects in signal processing systems and discusses further system uncertainties and how to possibly model them in future applications.

The trend in recent years show an integration of cyber physical systems into our daily life. Not just the number of "'embedded"' systems increases also the complexity of the single systems show a steady increase. The analysis and verification methodologies for purely digital systems are already at a mature stage. The evaluation and analysis of analog or mixed-signal systems on the other hand remains inefficient. Multi-run simulations provide an industry approved methodology but the steady increase in system complexity and considered system variations will restrict the efficiency of this techniques. Semi-symbolic simulations promise an answer to this simulation performance dilemma. They allow a semi-symbolic simulation of the nominal system quantities with their corresponding variations and thus the considering of a continuous number of parameter values in just one simulation run. The work within this thesis enhances the overall analysis capabilities by introducing the frequency domain for range based signals. Range based modeling techniques are supported by the creation of a building block library for typical deviation effects. Finally a range based refinement design flow was introduced which utilizes the symbolic nature of the range based signals to identify refinement candidates and iteratively updates the system properties.

The outline of the thesis is given in the following:
Chapter 1 formulates the motivation for the use of semi-symbolic simulations over traditional multi-run simulations which also would be suitable for the proposed objective. The advantage in simulation performance for systems under the influence of system parameter deviations justifies the development of the proposed simulation and analysis methodology. The efficient semi-symbolic simulation environment is extended to a range based refinement design flow. The scientific publications used as basis for the thesis objective are named and discussed in detail.

Chapter 2 summarizes the state of the art and related field of system simulations with the purpose to analyze and improve the system quality. Especially multi-run simulations (the main competitor in this field) are reviewed and compared against the proposed approach. System analysis techniques are presented and discussed for their applicability in the presented refinement design flow. Related work presents available methodologies for system analysis. It presents the semi-symbolic simulation environment used in this work and discusses past achievements and approaches in this field.

Chapter 3 describes the simulation framework in general. It briefly summarizes the SystemC AMS simulation framework, describes the range based simulation approach and presents enhancements achieved to extend the simulation and analysis capabilities.

Chapter 4 presents the main part of the thesis, the "'MARC refinement design flow"' with its design environment. It combines the SystemC AMS simulation framework with the range based extension and uses it to analyze and refine deviation affected system models.

Chapter 5 presents a selection of experimental case studies which should proof the efficiency and easiness of the proposed design refinement flow.

Chapter 6 concludes the thesis and will discuss the design environment and refinement capabilities of the approach. A section which will discuss future extensions to the presented methodology will end the work.

2 State of the Art

System analysis utilizing simulation techniques are a widely accepted approach to handle complex system models and to limit the computation complexity. Contrary to symbolic analysis where analytical formulas for the system behavior are deduced from the system equations a simulation based analysis offers a flexible and computation efficient methodology to handle even complex systems. System components can be added or removed to the system model without significant effort. The system behavior is simply influenced by adding/removing the functionality of the block to the data path and hence changing the overall functionality implicitly. Simulation environments in this work will be used in two different classes. The first and most important one uses a data flow *Model of Computation (MoC)* to describe and simulate the system model. SystemC AMS is a member of this class of simulation environments and is in the following used as basis for the semi-symbolic simulations. The second, less important representative is an *Electrical Network* description of the system and a SPICE like solving of range based simulation models. A system analysis utilizes the simulation results to either visually evaluate the system behavior and its properties or to (semi)automatically detect undesired system behavior. The most widely used analysis domains used are the **time domain** and the **frequency domain**. The time sequential execution and display in the time domain supports the humans notion on how processes behave in nature. The frequency domain characteristics of a signal allow a concise illustration of periodic events within complex signals.

2.1 Time domain analysis

The time domain represents the most natural illustration of signal characteristics describing the timed behavior of systems. The time domain representation is specially suited for stochastic signals. Most systems are specified not only by areas of allowed and restricted signal values but also by their timed behavior. System simulations act as basis for the analysis operation. The computed signal characteristics are determined by reading the system models input signals and calculating the output signal by applying the output relation together with the systems inner quantities. This computation process is sequentially enhanced in time.

2.1.1 Transient simulation

A transient simulation computes a time response of a circuit on an arbitrary input signal. This simulation category takes into account all non-linear effects of the simulated model and does not restrict the input signal characteristics. Generally speaking, the simulation engine is solving a non-linear differential algebraic equation (DAE) system to compute the simulation quantities. The general equation set

$$F(\underline{x}(t), \underline{\dot{x}}(t), \underline{u(t)}, t) = \underline{0} \tag{2.1}$$

specifies this DAE for a system model where $\underline{x}(t)$ is the vector of time dependent variables and $\underline{u(t)}$ is the input vector of the circuit. The equation set is usually (in SPICE like environments) derived by the **Modified Nodal Analysis (MNA)** and the equation set is solved by numerical approximation techniques [Vla94].

2.1.2 Timed Data Flow

The *Timed Data Flow (TDF)* is an advancement of the known *Synchronous Data Flow (SDF)* [BVGE10]. SDF is a dataflow Model of Computation (MoC) which restricts the process input and output rates to fixed numbers. It was introduced by [LM87] in 1987 and restricts the original concept of Kahn process networks [Kah74] to fixed rates. This restriction in flexibility allows a compile-time static scheduling of the simulation model. A static schedule offers a significant increase in simulation performance as the single processes are invoked always in the same order. TDF introduces the physical dimension *time* to the description. The MoC still determines a static schedule for process executions but a delay in cyclic dependencies has to be specified explicitly. A "‘sample time"’ at which the simulation cluster cyclically is invoked has to be specified which activates the cluster at discrete-times. The timed data flow MoC was introduced to describe in particular analog and mixed-signal models in the SystemC AMS environment.

2.2 Frequency domain analysis

The analysis of signals in the frequency domain introduces advanced possibilities to evaluate their behavior. Periodic components of a complex superimposed signal can be characterized in the frequency domain by a single spectral component. A signal can thus be analyzed for its spectral components and analyzed for their position when constructing/analysing a frequency spectra. Communication systems and especially the analog front end typically uses frequency translation components like mixers to construct or reconstruct the baseband signals. Signal processing applications also regularly modify the spectral appearance of handled signals. All signal modifications which influence the spectral constitution of the signal, demand for a possibility to evaluate the correct operation. A frequency domain analysis facilitates this requirement.

2.2.1 Fourier analysis

A widely accepted methodology to analyze the frequency behavior of signals is a Fourier analysis. It bases on the mathematical concept of Fourier series that a general signal may be represented by an infinite sum of sine and cosine functions at integral multiples of the fundamental frequency

$$y(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \Big(a_k cos\,(k\omega t) + b_k sin\,(k\omega t) \Big) \tag{2.2}$$

where a_0 is the DC portion and a_k and b_k represents the weights of each spectral quantity. The spectral components of the general signal are represented as scaled sine and cosine pairs which constitute to the complex frequency spectra of the signal. The transition from the time to the frequency domain is usually accomplished by the Fourier transform. Thereby the input signal is transformed into the frequency domain by executing an integral transform calculating a complex valued frequency value for every spectral point.

2.2.2 z- and Laplace domain

The z- and the Laplace transform are a modification of the Fourier transform where not the frequencies are the scope of analysis- For the Laplace transform the signals are resolved into their moments. The z-domain is the time discretized counterpart to the continuous Laplace domain. Transfer functions of complex processing blocks are quite often specified in the Laplace's s-domain. When having a transfer function in the s-domain advanced analysis methodologies like the pole-zero analysis becomes feasible.

2.3 Affine Arithmetic

Affine Arithmetic (AA) is a methodology for a self validated range propagation introduced by [FS97]. It bases on Interval Arithmetic (IA) [Moo66] which was created by R. Moore. IA is a model for self validated numerical computations where the quantity is represented by an interval and mathematical operations yield in an interval that guaranteed includes the unknown result. The operations add, subtract, multiply and division are defined in IA which allows the processing of the intervals. IA is widely known for its use in error propagation calculations [Loz83] and error bounds estimation. The intervals are simply described by the upper and lower boundary. This simple description with just the two numeric boundaries results in the major drawback of Interval Arithmetic. The intervals do not hold any information about the origin of the interval. Thus, mathematical operations on two or more intervals are always expected to be operations on non-correlated intervals. This leads to the situation that the subtraction of one interval with itself leads not to the expected [0..0] interval, but to an interval greater than the original [FS97, p.36]. This effect is referred to as dependency problem. Electrical systems and in particular control systems often contain feedback loops. These feedback loops are intended to subtract the sensor measurement from the reference input. The controller therefor reduces the systems error. When using IA to describe and manipulate systems with feedback loops the feed backed signals computes to a wider interval. The interval result is said to be over-approximated. Systems modeled with Interval Arithmetic will never reduce their interval size, they are steadily widened throughout the computation process. This widening of the intervals is the reason that IA never gained real importance for range based system simulations. Interval Arithmetic tends to a significant over-approximation of the resulting intervals which is named the **error explosion problem**.
Affine Arithmetic (AA) overcomes this dependency problem by labeling the single ranges with symbolic identifiers and considering range correlations during mathematical operations. AA automatically keeps track of correlations between quantities which is considered as a self validated range propagation. AA reduces the significant over-approximation of IA in long computation chains or when using systems with signal cancellation structures.
Nevertheless, Affine Arithmetic guarantees to include the unknown calculation result in the resulting range quantity. It determines the ranges in a pessimistic way, considering the full set of possible results and guaranteeing a full coverage. If approximations of the resulting range apply, over-approximations are used to guarantee the pessimistic nature.

Affine Arithmetic is a methodology to define ranges as superposition of a *center value* with a set of $\mathcal{N}_{\tilde{x}}$ subranges. In AA a partially unknown quantity is described as first-order polynomial spanning a range over the area where the quantity resides in. The first order polynomial [FS97, 43]

$$\tilde{x} = x_0 + x_1\epsilon_1 + x_2\epsilon_2 + ... + x_n\epsilon_i \qquad \epsilon_i \in [-1, 1] \tag{2.3}$$

represents a typical AA description where x_0 represents the the center value, the symbolic identifier ϵ_i generates an interval between $[-1\_\_1]$ and the partial deviation x_i scales the range to the intended value. Each deviation symbol ϵ_i represents a source of uncertainty or deviation and thus is used to keep track of the range correlations. A key feature of Affine Arithmetic is that one deviation term may contribute to more system quantities, either through the original source or as result from the computation process. These shared symbols indicate a correlation between the quantities and especially their sub-ranges. Along with the formulation of mathematical operations on these ranges an arithmetic is established [FS97]. In order to compute with Affine Arithmetic appropriate mathematical operations will be provided that compute on Affine Forms instead of the usual real valued quantities. Uncertainties in systems are modeled as ranges and these ranges are modeled by so called Affine Forms. Mathematical calculations on such Affine Forms are defined and provide at least pessimistic approximations to allow worst case evaluations. $\mathcal{N}_{\tilde{x}}$ defines a set of natural numbers identifying all deviation terms $x_i\epsilon_i$ in symbol $\tilde{x}$.

$$\tilde{x} = x_0 + \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i\epsilon_i \qquad \epsilon_i \in [-1, 1] \tag{2.4}$$

The superposition of the central value with a sum of deviation terms indicate the formalism which is used to model deviated systems. The central value holds the nominal, designed quantities and the set of deviation terms represent all single uncertainties and inaccuracies which influence the system behavior. All quantities of the system model are described by this range based approach enabling the computation of the impact of the even cross correlated deviation effects on the overall system behavior. Different affine signals can contain identical deviation terms which would indicate a cross correlation of this signals. The dependency conservation of AA allows a conciser description of the resulting ranges. [FS97, 45] shows that the joint range of two correlated AA signals form a convex polygon symmetric around the central points (x_0, y_0). Interval Arithmetic without any correlation information specifies this identical joint range as rectangular introducing a significant wider range.

Affine Arithmetic not only defines how to efficiently model intervals in symbolic ranges it also specifies mathematical operations on these affine quantities. When computing with Affine Forms modified mathematical operations have to be provided which handle affine input signals and calculate the pessimistic resulting ranges. The mathematical operations can be divided into two classes, affine operations which solve in exact results and non-affine operations which are derived as pessimistic approximations. Pessimistic approximations are considered to safely contain the actual operation result but over-approximates the range, since the exact result can not be formally determined. Affine operations are the addition and subtraction of Affine Forms as well as the multiplication of Affine Forms by numeric values defined as:

$$\tilde{x} \pm \tilde{y} = (x_0 \pm y_0) + \sum_{i \in \mathcal{N}_{\tilde{x}}} (x_i \pm y_i)\epsilon_i \tag{2.5}$$

$$c\tilde{x} = cx_0 + \sum_{i \in \mathcal{N}_{\tilde{x}}} cx_i \epsilon_i \tag{2.6}$$

$$\tilde{x} \pm c = (x_0 \pm c) + \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i \epsilon_i \tag{2.7}$$

where $\tilde{x}$ and $\tilde{y}$ are affine quantities ($\tilde{x}, \tilde{y} \in AAF$) c represents a real value ($c \in \mathcal{R}$). The main advantage of affine operations are the interval exact solutions of the calculation which allows to use tight ranges to describe the system quantities.

Non-affine operations are derived by an approximation of the resulting Affine Form. Since the approximation is considered as being pessimistic, non-affine operations are the source of over-approximations which influences the simulation expressiveness negatively. The resulting affine range is estimated by calculating a first order approximation and adding an additional deviation term accounting for the approximation error. The Chebyshev approximation is originally [FS97, 57] used to determine the approximation term. The additional deviation term contribute to the over-approximation of the whole system. If many non-affine operations are used in a system or if chained non-affine calculations are performed the over-approximation can reach considerable figures. Significant over-approximations can prohibit the usage of semi-symbolic simulations as the system behavior is concealed by wide ranges.
Non-affine operations in the original work of [FS97] comprise of

- square root of Affine Form
- exponential function
- reciprocal function
- multiplication of Affine Forms
- division of Affine Form

where the approximations are found either by using the Chebyshev approximation or by using the min-range approximation.

Improvements on Affine Arithmetic have been introduced in recent years to reduce the over-approximation effects of original Affine Arithmetic. For instance, the Quadratic Arithmetic [GOGB07, GOB08] has been introduced which adds multiplications and the square function of Affine Forms to the affine operations. This is reached by using a second order approximation instead of the original first order approach. An increase in range accuracy is here bought by an increase in computation effort. The usability of this method strongly depends on the application as also Affine Arithmetic reduces the over-approximation of Interval Arithmetic on the costs of computation efficiency.

2.4 Related Work

Refinement decisions for systems at the very beginning require an accurate knowledge about the system behavior and the impact of varying quantities on it. Multi-run simulations are the most wide-spread approach to analyze and evaluate system properties under the influence of deviation effects. This exhaustive simulation approach offers a good estimation when performing enough simulation runs but at the same time looses its efficiency or even applicability if the number of simulation runs get to high. The number of simulation runs can be reduced by statistical approaches such as 'Design of Experiments' [RGDP10] for finding worst case parameter sets, or 'Importance Sampling' [SR07] for a more accurate and focused estimation of the systems statistical properties. Unfortunately multi-run simulations are time consuming and possess an exponential dependency between the number of necessary runs and the number of system parameters. A sensitivity analysis identifies input-output relations by using statistical techniques applied on multi-run results [Kle95]. Over the past years formal verification techniques proofed their capabilities of verifying digital systems but provide limited information for system refinement techniques.

2.4.1 Multi-run simulation methodologies

2.4.1.1 Monte-Carlo simulation

Simulations of systems are easier to study and obtain than calculating analytical expressions for the problem. The well established Monte Carlo simulation [Rub81], a stochastic computer simulation technique, approaches the system behavior and the influence of parameter variations by repeated system simulations using random parameter variations. Contrary to deterministic simulations, Monte-Carlo methods generate statistical descriptions of the system behavior formed by the variable and continuous verification space. With each simulated verification space realization, the variance of the stochastic system model is reduced, thus improving the reliability and accuracy of the constructed model. For a sufficient high number of simulation runs the system characteristic can be determined fairly precisely. However, the continuous parameter space is added a new dimension whenever a new parameter is introduced to the system. Accordingly, the number of necessary simulations for a given statistically confidence grows exponentially with parameter space dimension.

Importance Sampling is a technique to significantly reduce the number of Monte-Carlo simulation runs for a given statistical confidence. Generally speaking, importance sampling is a variance reduction technique where parameters of increased significance are sampled more comprehensively. For a wise chosen sample probability density function the estimator variance can be dramatically reduced, thus reducing the number of simulation runs. Nevertheless, choosing a "‘wise"’ sample probability density which considers and favors "‘important"’ samples is tricky.

Monte-Carlo simulation and importance sampling are methodologies to determine the system characteristic from numerical simulation runs. They rely on repeated simulation runs with either random or guided parameter variations. The exponential dependency of the variation space size from the constituting system parameters cause the simulation performance to get a restricting factor for these approaches. An analysis approach based on numerical simulations basically results in numerical quantities which determine the system behavior. For an analysis methodology this outcome is precise and sufficient. When not just trying to analyze the systems behavior but also to determine refinement steps a numerical result is not enough. The result only reflects the combined signals but it does not hold any information on how strong the single system functions (or better parameters) influence the result.

2.4.1.2 Worst Case analysis

Circuits and systems are usually subject to parameter variations from their nominal specified value. A Worst Case analysis tries to estimate the Worst Case maximum and minimum performance of the deviated system. It is a widely accepted industry adopted methodology which is used in three major techniques:
Extreme Value Analysis (EVA): Also denoted as absolute Worst Case method which calculates the systems extreme behavior by simply considering the parameters extreme values. EVA requires medium performance costs but can result in pessimistic estimations of the Worst Case behavior.

Root Square Sum (RSS): Summarizes the standard deviations of all part tolerance distributions to obtain the overall Worst-Case bounds deduced from the systems standard deviation. Needs knowledge of standard deviations for every considered parameter.

Monte Carlo (MC): Provides the most realistic Worst Case estimate from the presented methods. Performs an empirical determination of the statistical behavior of the system by repeated simulations using random parameter variations. Unfortunately this method demands for very time and computation power consuming system simulations.

Corner Case simulations are used to reduce the number of necessary system simulations for determining the Worst Case. If the system response is monotonic with respect to parameter variations, one corner parameter set results in the overall system worst case behavior. This significantly reduces the number of simulation runs as the Corner Case analysis has a quadratic dependency on the parameter space dimension compared to the exponential correlation of the Worst Case analysis.

A Worst Case analysis basically relies on multiple simulation runs to determine the system characteristic and its behavior. As the number of system parameters that have to be considered reach quickly large numbers also the number of simulation runs to cover all these parameters reaches enormous counts. The simulation performance poses again the limiting factor for realistic, industrial sized systems. The RSS methodology does not require multi-run simulations but the standard deviation of all components and their parameters are difficult to obtain.

2.4.1.3 Design of Experiments

Design of Experiments (DoE) creates a *metamodel* from experimental data which can be used to validate the original system behavior [Kle08]. The experimental data is collected from simulations of the original system model or measurements on already realized systems. The *metamodel* is an approximation of the Input/Output relationship deduced from the original simulation model. The DoE metamodel is created using first-order or even higher order polynomials and may also consider factor interactions in the abstracted model. [RGDP10] uses Design of Experiments for a model-based verification, calculating a sensitivity and worst-case analysis. The model based verification approach significantly reduces the number of simulation runs, necessary. The *metamodel* can be divided into three major parts [RGDP10]. The *main effect* which models the first-order approximation, an *interaction effect* which handles interactions between factors and the *quadratic effect* which improves the model accuracy by a second-order estimate. When neglecting the second-order estimate the DoE model is defined identical to an Affine Arithmetic description. While DoE uses a stochastic analysis to predict the model property, Affine Arithmetic and the related semi-symbolic modeling approach constructs the system model from a deterministic nominal model which is expanded by the worst-case deviation effects.

2.4.1.4 Sensitivity analysis

Sensitivity analysis methodologies can be basically divided into two distinct classes.
Analytical methods [FRV96]: which basically calculate the system function as analytical formula. The system function contains sub expressions for every quantity that influences the system behavior. System dependencies from single contributors or even interdependencies can be analyzed by solving the equation with respect to the parameter under view.
Sample based methods [HJSS06]: where the system is repeatedly tested or simulated for identifying input-output parameter relations. Impacts of system parameters on the output behavior can be estimated by simple input-output scatter plot analysis or by identifying the main contributing factors in the system, among others.

Both, analytical and sample based methods are restricted in their usage for realistic sized designs. Deriving the analytical formula of the system function is influenced by the exponential dependency of the expression length from the circuit size. Sample based methods are stochastic based and face the same problem, that they demand many simulation runs to obtain a predictable model quality.

2.4.1.5 Symbolic- and Semi-symbolic analysis

Symbolic and semi-symbolic analysis techniques derive a symbolic system representation which can be used for a system behavior analysis. As additional feature the system function also provide sub expressions which can be used for the interpretation of inner system dependencies . The symbolic methods generate system functions as analytical formula showing the dependencies of the single subexpressions. This analytical formula can be used to calculate transfer functions, poles and zeros, root loci, Bode plots or even harmonic distortion quantities. A sensitivity analysis is computed by deriving the symbolic system functions with respect to the respective system parameter. Symbolic analysis is applicable on linear systems and systems with weak non-linearities [WGS90]. For real sized circuits the complexity of the formula becomes a restricting factor. Therefor, a formula simplification by approximation has been introduced [FRV96]. A reduction of formula complexity is achieved by discarding of non-significant expressions in the original system function. The decision about the significance of the single sub expressions is found by a numeric estimation of the symbolic parameters. The approximation operations can be distinguished into three classes:
Simplification After Generation ***SAG****:* First the full symbolic formula is computed and following the formula is approximated by removing minor contributions. As the symbolic expression is firstly constructed for the full system this technique is restricted to small and medium complexity circuits [FRV96].
Simplification During Generation ***SDG****:* If the simplification operation is performed directly during the formula generation only the dominant contributions are considered in the approximation.
Simplification Before Generation ***SBG****:* The computational complexity for system solving and thus the complexity of the resulting symbolic function grows exponentially with circuit size. Thus,

simplifying the original system structure significantly improves the computation efficiency. Recent SBG approaches simplify the system equations prior to the transformation operation [GRGR+99].

Semi-symbolic techniques combine symbolic descriptions with numerical solutions. Parameters of interest are modeled as symbols, whereas the remaining system quantities are summarized as numerical results. Plain numerical system results indicate the system behavior but do not help in understanding the inner dependencies. A semi-symbolic approach in this scope allows a balancing between the expressiveness of the system function and the computational effort to derive it.

2.4.2 Formal verification/analysis methodologies

2.4.2.1 SAT solving, Theorem proving

Properties of specifications and systems are expressible in propositional logic. These boolean expressions can be checked for their formal satisfiability or the opposite, unsatisfiability. Intentional system behavior is proofed to be assignable by SAT-solving the property expression. In complexity theory a satisfiability problem (SAT-problem) is a decision problem which evaluates boolean expressions. The Cook-Levin [Coo71] theorem proofs that the boolean satisfiability problem is NP-complete. However, the NP-completeness just states that there is no general efficient SAT-solving algorithm but many efficient implementations for practical problems exist. SAT-solvers are used in automated theorem proving which is used in integrated circuit design and verification. Efficient implementations of SAT-solvers for specific fields of applications circumvent computation restrictions of NP-complete algorithms. Nevertheless, obtaining boolean expressions which usefully describe the properties of practical circuits limit this verification approach.

2.4.2.2 Model checking

Over the past years formal verification techniques proofed their capabilities for verifying digital systems. Model checking [MGP99, BCM+90] verifies a specification, given as temporal logic description against the circuit property, represented as transition model. The use of a temporal description language allows to include temporal requirements into the verification process and results in a reachability analysis using a state space exploration method. If the desired system property is not met a counterexample is produced which can be used to correct and improve the system behavior. Its main restriction is the state explosion problem [Val98], where the number of model states become immense for practical systems. Improvements have been achieved in reducing the number of necessary states. Abstraction [CGJ+03] verifies the model on an abstracted level to reduce the complexity of the description. Partial order reduction [MGP99, God96] exploits the commutativity of concurrently executed transitions to construct a reduced state graph. These improvements shift the frontier for model checking but do not generally solve the state explosion problem. Model checking has recently also been successfully applied on analog and mixed-signal (AMS) systems. In general the continuous state-space of the analog domain is partitioned into geometric objects, representing states of the system model [JH08, GPHB06]. Trajectories between the geometric objects can be calculated which completes the abstracted automaton model. Traditional model checking techniques are finally used to verify the model against a temporal logic formulation of the specification. The state explosion problem still influences the verification efficiency but the discretization of the continuous state space and determining the state transitions consumes additional computation power. Recent advances

[CGJ+03, MGP99, God96] in abstraction and partial order reduction overcome the restrictions from the state-explosion-problem and allow a model checking of real sized systems.

2.4.2.3 Equivalence checking

A common industry adopted methodology to compare a systems behavior is equivalence checking. With this technique two representations of a system are compared for their equivalence. The basic approach was to check whether an obtained gate- or netlist is still equivalent in its behavior to the original RTL description [HC98]. During a digital design flow potential modifications on the original description are executed which alter the behavior of the finally implemented design. In order to detect such inequalities in a deterministic way equivalence checking has been introduced. The trend in recent years goes towards designing systems at higher levels of abstraction thus also equivalence checking has been extended to the system level description [KLM05]. Equivalence checking basically uses two techniques for reasoning about the behaviors.

- comparing binary decision diagrams (BDD)
- checking satisfiability with SAT-solver

As used in model-checking techniques the systems are represented as binary decision diagrams. When transforming the BDDs into canonical ordered binary decision diagrams (OBDD) the equivalence of the descriptions can be checked by a simple diagram comparison. The system descriptions which are to be compared may also be represented as boolean expressions. SAT-solvers are an efficient means to reason about the formulas and perform the equivalence check. Equivalence checking has also been extended to the analog and mixed-signal domain. [HKH04] divides the continuous analog state space into planes representing areas of identical behavior and thus creating a discretized state space which finally is comparable utilizing traditional equivalence checking techniques.

2.4.2.4 Semi-symbolic approaches

A novel technique of handling system deviations by adding them as ranges to the system model is the semi-symbolic simulation approach [HGW05, GHW05, GGB06b, LAOW05, Gra09]. Thereby Affine Arithmetic [FS97] is used to describe and compute the system model and to simulate the system behavior. The semi-symbolic approach provides an interesting compromise between the efficiency and completeness of formal methods with the usability of simulation based verifications. The completeness of the verification does not hold in a full formal sense but can be shown for the effects of the range modeled parameters and signals on the system. An established way to perform a range based semi-symbolic simulation uses the SystemC AMS modeling and simulation environment [GHW04] for system level and a numerical spicelike environment for transistor-level simulations [GOB08]. The SystemC AMS environment can easily be extended by using an Affine Arithmetic library, which overloads certain computation related operations, introducing the semi-symbolic methodology [HGW05]. In such environments initially uncertain parameters or quantities will be modeled using ranges instead of single numerical values which allows a semi-symbolic computation of the system quantities. The quantities comprise of a numeric central value and a symbolic deviation term. The deviation term holds a numerical value which scales the interval and a range symbol which forms the interval itself. This explains the name semi-symbolic simulation, simply referring to the mixed representations. Although the most published work concentrates on the system level also transistor level circuits are simulate able when solving the non-linear differential equations by using Affine Arithmetic [GOB08]. [GHW05] uses semi-symbolic simulation to analyze the convergence behavior of control loops in presence of uncertainties. [GGB06a] and [GHW04] additionally enhance the semi-symbolic simulation for simulating non-linear analog circuits and obtaining refinement information to improve the system quality. The problem of over-approximation is addressed in recent works where the Affine Arithmetic is enhanced by additional affine operations which result in exact solutions for a higher number of mathematical operations [SLMW03, MT06, GOGB07]. The main approach in enhancing the affine operations is to replace the first order approximations with higher order terms. Semi-symbolic approaches are also used for an implicit error analysis of systems which provides tighter estimates as it supports error cancellation. The first work targeting floating point errors in digital signal processing (DSP) systems was [FCR03]. Quantization and rounding errors are modeled using Affine Arithmetic and propagated through the system model. As resulting quantity an error bound caused by the single error sources, their correlations and interdependencies is obtained which estimates the bound more accurate as it considers dependencies. [LAOW05] also uses Affine Arithmetic to estimate the error bounds of DSP systems but extends the methodology by an optimization process. [LCNT07] finally modifies the Affine Arithmetic approximation process to improve the calculation performance to again compute the error bound of a DSP system and optimize the system for the optimal bit width in each stage. The improvement in this approach simply discards the higher order remains in the first-order approximation for non affine operations. Neglecting the higher order effects may result in computation results located outside of the predicted range area loosing the pessimistic paradigm and loosing the guarantee for a signal inclusion.

2.4.2.5 Symbolic trajectory evaluation

A derivative of model checking which uses a form of symbolic simulations is the symbolic trajectory evaluation (STE) [HSB95]. It can be characterized as lattice-based model checking technology which is in its field more immune to the state explosion problem. The generalized STE (GSTE) improves the original symbolic trajectory evaluation which is limited to properties over finite time to unbounded properties ranging to infinite time [YS03, CR09]. In STE the circuit is described by a four valued (1,0,X,T) node representation [YS03]. This "'quaternary"' abstraction of the circuit significantly reduces the state space of the system model. The simulation of this quaternary model results in a comparison of STE assertions against the timed symbolic simulation trace. Symbolic trajectory evaluation is industry proofed for medium to large scale hardware designs [YS03]. It supports the automation of verification processes and is already used by Intel, Compaq, IBM and Motorola.

2.5 Discussion of related work

The work introduced in this thesis uses semi-symbolic simulations to simulate and analyze circuits and systems. Based on this simulation the system behavior and possible refinement candidates are deduced. One class of already established analysis methodologies are multi-run simulations. The Monte-Carlo simulation randomly varies system parameters and simulated the model repeatedly. The resulting simulation behavior is used to predict the system characteristics. As

the number of system parameters in complex systems become huge and also the variance of the system parameters is increasing, simulation times become a limiting factor for this approach.
Importance Sampling was introduced to reduce the number of necessary simulation runs. It can be considered as guided multi-run simulation and provides valuable improvements. However, Importance Sampling does not solve the multi-run problem where an increase in system parameters exponentially increase the number of simulation runs. A semi-symbolic simulation approach avoids this problems. The symbolic handling of parameter variations allow finding the system characteristics of deviated systems as an area of potential values in just one simulation run, thus avoiding the multi-run performance problem.
A Worst Case analysis also bases on a multi run approach. The simulation performance problem again poses a limiting factor for this technique. The Root Square Sum (RSS) methodology does not require multiple simulation runs but the standard deviation figures of all part tolerances are usually not easy to obtain. All these numerical methodologies lack a statement about the correlation between the system behavior and its causing parameters. For a refinement of systems not just the system characteristic but also potential refinement candidates in the systems parameters have to be identified. Semi-symbolic simulations with their symbolic nature allow a back tracking of deviation ranges from the resulting system behavior back to their originating sources. By doing so, refinement candidates can easily be identified and their impact on the system behavior evaluated.
Design of Experiments (DoE) creates a metamodel from experimental or simulated data. The metamodel is an approximation of the Input/Output relation and is used to verify the system behavior on an abstracted, reduced complexity model. The abstraction process relies on a stochastic analyses to predict the model properties and thus potentially neglects important system characteristics.
Semi-symbolic simulations provide a pessimistic approach which guarantee a inclusion of resulting quantities and never results in an underestimation of systems characteristics. A sensitivity analysis either calculates the system function as analytical formula or identifies the Input/Output parameter relation in a sample based technique. Either approaches are restricted in their usage for realistic sized designs.
Symbolic- and semi-symbolic analysis approaches derive an analytical formula as characterization of the system behavior. Sub expressions can be used to predict inner dependencies and to analyze the system. The construction of the system function is expensive although several simplification operations exist. The computation complexity restricts this methodology to medium sized systems. Semi-symbolic approaches simplify the formula by replacing less interesting sub expressions with numerical values, thus keeping just the symbolic parts which are of special interest. Symbolic analysis offers a great expressiveness of its system functions to correlations and dependencies. The complex deriving process restricts its usage to medium sized systems.
SAT solving and theorem proving are techniques classified as formal verification methodologies. Properties of specifications and systems can be expressed in propositional logic. These boolean expressions can be checked for their satisfiability, or unsatisfiability by SAT-solvers for theorem proving. Constructing a propositional logic function describing a system behavior can become complex. Also the SAT-solving is NP-complete and restricted in its computation efficiency. Obtaining boolean expressions which usefully describe the properties of practical systems limits this verification approach and propositional logic descriptions for analog quantities are not available anyway.
Model checking is a formal verification methodology which evaluates specification properties expressed as temporal logic description against a system model typically given as BDD. The methodology is applied on purely digital systems but has also been advanced to the analog domain in

an academic approach. Model checking suffers of the state explosion problem which restricts the usability of the methodology for complex system models. Also the lack of support for analog and mixed-signal systems restricts the usability in the scope of this thesis.
Equivalence checking uses a comparison of system descriptions to analyze the behavior of implemented systems, or to prove the equivalence of two representations of a system. The primary focus of equivalence checking is on digital systems but has also been extended to the analog and mixed-signal domain. Again, the support for the analog domain is academic and the partitioning of the continuous analog state space into planes of identical behavior is complex.
The symbolic trajectory evaluation is a derivative of model checking which uses a form of symbolic simulations. It can be characterized as lattice-based model checking technology which is in its field more immune to the state explosion problem. It operates on digital systems and is currently not usable on analog or mixed-signal systems.
Semi-symbolic simulation approaches are a novel class of system simulation and analysis technique. The combination of symbolic representatives with numerical quantities allow an efficient system simulation along with an improved analysis capability. The symbolic identifiers define ranges which are superimposed on the numerical system model. The idea is to describe deviations or variations of system parameters as ranges and simulate the deviated system model. Contrary to the multi-run approaches the system response containing all deviation effects is found in just one simulation run providing a pessimistic worst case bound but offering additionally subranges representing the impact of every deviation quantity on the output signal. The semi-symbolic technique is applicable on analog as well as on mixed-signal systems and has been introduced for system level simulations but also transistor level circuit simulations.

3 Simulation framework

Semi-symbolic simulations in this scope combine conventional numerical simulation environments with symbolic enhancements. The system simulation is computed in a numerical way whereas the system deviations and uncertainties are modeled as ranges. The system response is finally computed and represented as the numerical result superimposed with a set of ranges describing the single deviations.

As simulation environment SystemC AMS is used which is extended by an Affine Arithmetic library. The uncertain system model parameters, as for instance, tolerance values, voltage offset deviations or quantization effects are thereby characterized as ranges which represent the continuous deviation of the nominal parameter value. The resulting range based system model is finally simulated in a numerical way but additionally considers the symbolic ranges and the mathematical processing on them. The mathematical operations are defined by Affine Arithmetic and are implemented in the enhancement library. This library is included into the SystemC AMS environment and overloads the computation related operations with their specific range based counterparts.

3.1 SystemC AMS

SystemC AMS [10] is a C++ class library introduced for modeling and simulating of especially analog and mixed-signal systems. It bases on the original SystemC [7] C++ hardware description environment but extends the functionality for descriptions of analog mixed-signal behavior. SystemC is an object oriented modeling and simulation environment designed to describe complex digital hardware structures that uses *Discrete Event* as simulation model. Analog behavior is characterized by continuous time and different forms of system description. SystemC AMS accounts for this varieties and offers several distinct *Models of Computation* (MoC) to support system descriptions on different levels of abstraction, with varying representations.
The experimental *Frauenhofer* proof-of-concept SystemC AMS environment [2] integrates a simulation engine and tracing capabilities. The implementation is available as C++ class library which is compiled to the specific target system and which complies to the OSCI SystemC AMS Extensions Standard 1.0 language syntax. Simulation domains include transient simulation, AC simulation and frequency domain simulations.
A *Model of Computation* (MoC) defines the semantic for a model description by specifying its meaning and defining how the modeled system behaves. A model of computation can also be

considered as an automaton which accepts models that have been described following its semantic. A MoC is independent from the describing language or elements, it simply has to obey the definition rules.

- ***Discrete event model (DE)***: The DE model is used for discrete processes and is implemented in the SystemC superclass of SystemC AMS. DE uses explicit synchronization signals during the simulation of the system model. The events are scheduled in an event queue and are sequentially processed by using simulation delta cycles. Processes are activated by either elapsed time or events. Discrete event is the most widely used MoC as it is also used for Verilog and VHDL simulations for distributed process simulations. *Transaction level modeling* (TLM) was introduced as enhancement of DE in the SystemC environment. With TLM bus transactions are abstracted as message passing allowing the designer a SW orientated, easier model description.
- ***Continuous-time signal flow (LSF)***: The linear signal flow (LSF) MoC is used to describe directed signals in the continuous time domain. SystemC AMS implements predefined blocks to describe transfer functions or to specify and solve systems constituting of primitive modules (adders, integrators, differentiators). LSF solves differential and algebraic equations numerically (DAE) at appropriate time steps.
- ***Electrical linear network (ELN)***: Electrical linear networks are build from network elements that assemble an electrical circuit. Network elements are predefined in SystemC AMS and contain for example resistors, capacitors, nodes, converters and sources. ELN is a SPICE like network description but is restricted to linear elements. The equation system of the connected electrical network is generated in an elaboration phase and solved numerically during system simulation.
- ***Timed data flow (TDF)***: Timed data flow is timed derivate of the original *Synchronous Dataflow* (SDF). It bases on the original Khan process networks but demands a predefined, finite number of port of input/output tokens for each port of its processes. This restriction allows a static scheduling of the model processes and significantly improves the simulation performance. TDF is especially to model and simulate DSP behavior as tokens can be considered as data samples and the computation process follows a data flow structure. TDF also supports multi-rate specifications to allow the description of interpolators or decimators in decoder blocks. The cluster schedule is calculated in an elaboration phase allowing the cyclic execution of the cluster modules during simulation. TDF also supports the use of transfer functions, either given in s-domain polynomial form, s-domain pole/zero representation or state space equations.

Figure 3.1 shows the layered composition and the time synchronization of SystemC AMS. As SystemC AMS bases on the SystemC implementation the SystemC specific elements, its simulation system time and the DE model of computation is the foundation of the simulation environment. A general synchronization layer connects the SystemC AMS signals to the underlying SystemC time base. The implemented MoCs are on top of this synchronization layer to ensure a permanent synchronization of the continuous SystemC AMS simulation time with the originating SystemC time base.

Figure 3.2 gives a typical complex mixed-signal system, as specified by [8] as motivation for mixed-signal extensions to SystemC.

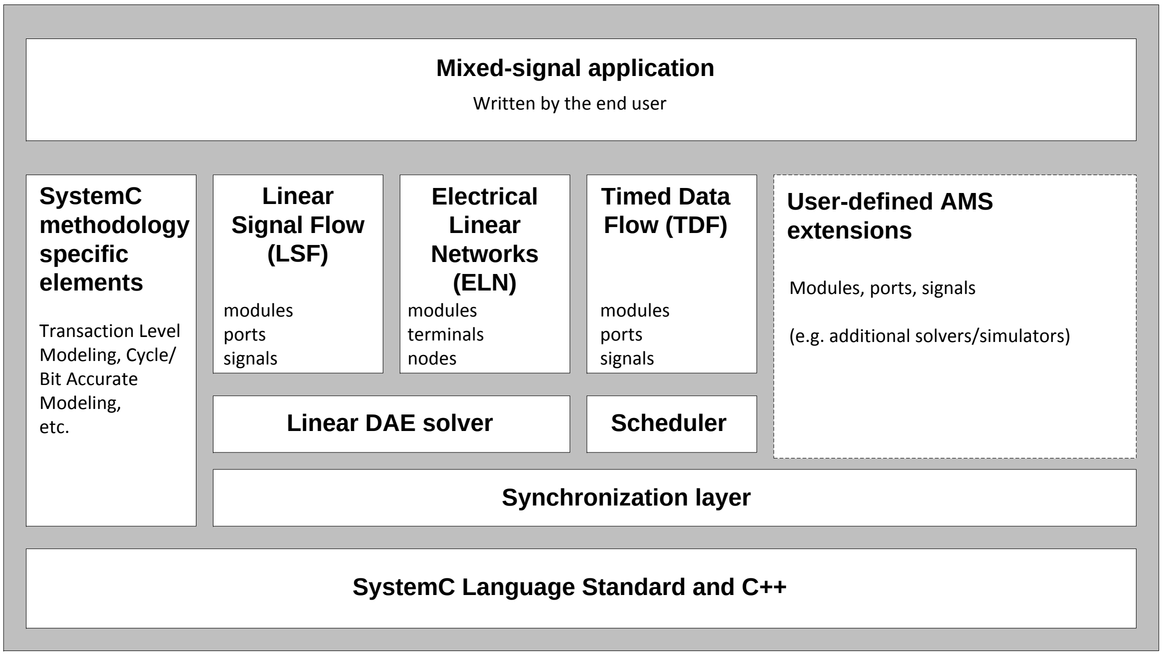

Figure 3.1: SystemC AMS engine and its layers

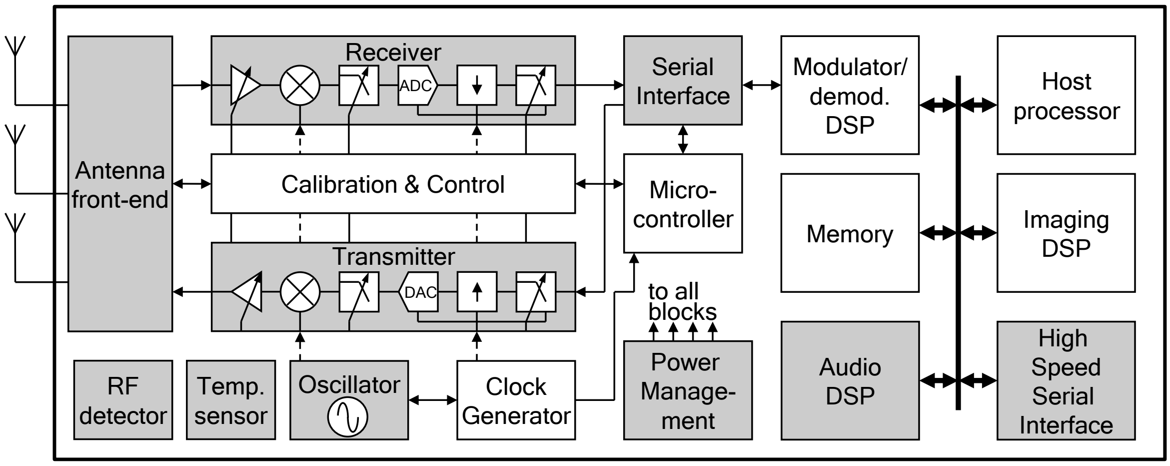

Figure 3.2: Typical mixed-signal transceiver

Heterogeneous systems integrate subsystems of different domains which are functionally interwoven. The transceiver example shows a mixed-signal system which will be described also at different levels of abstraction. The RF front end, RF detector and temperature sensor fully comprise of analog circuits. The receiver and transmitter blocks implement analog functionality but as well digital control signals which form mixed-signal modules. The rightmost block all represent digital hardware, also running software codes. These are used to compute the signals and to control the whole system through the application. The system view ranges from analog circuits in the RF parts to the system level and even application level in the digital, software parts.

3.2 Semi-symbolic modeling and simulation

Electronic systems (whether analog, digital or mixed-signal circuit) are designed to meet an ideal behavior. The actual realization of the system introduces uncertainties and potential variations in all system parameters. Component values will not reach the nominal values, they are source of part tolerances and deviations. Semi-symbolic simulations account for this inconsistency. Deviations of system parameters are considered by symbolic range identifiers which are superimposed the nominal system model. This range based system model is used for simulation and provides the system behavior as range of potential signal characteristics in just one simulation run. All parameter deviations contribute to the resulting signal range and are considered following the concept of Affine Arithmetic. Parameter deviations which directly or indirectly contribute to a range signal can be identified as sub-range and tracked back to their origin via the symbolic range identifiers.

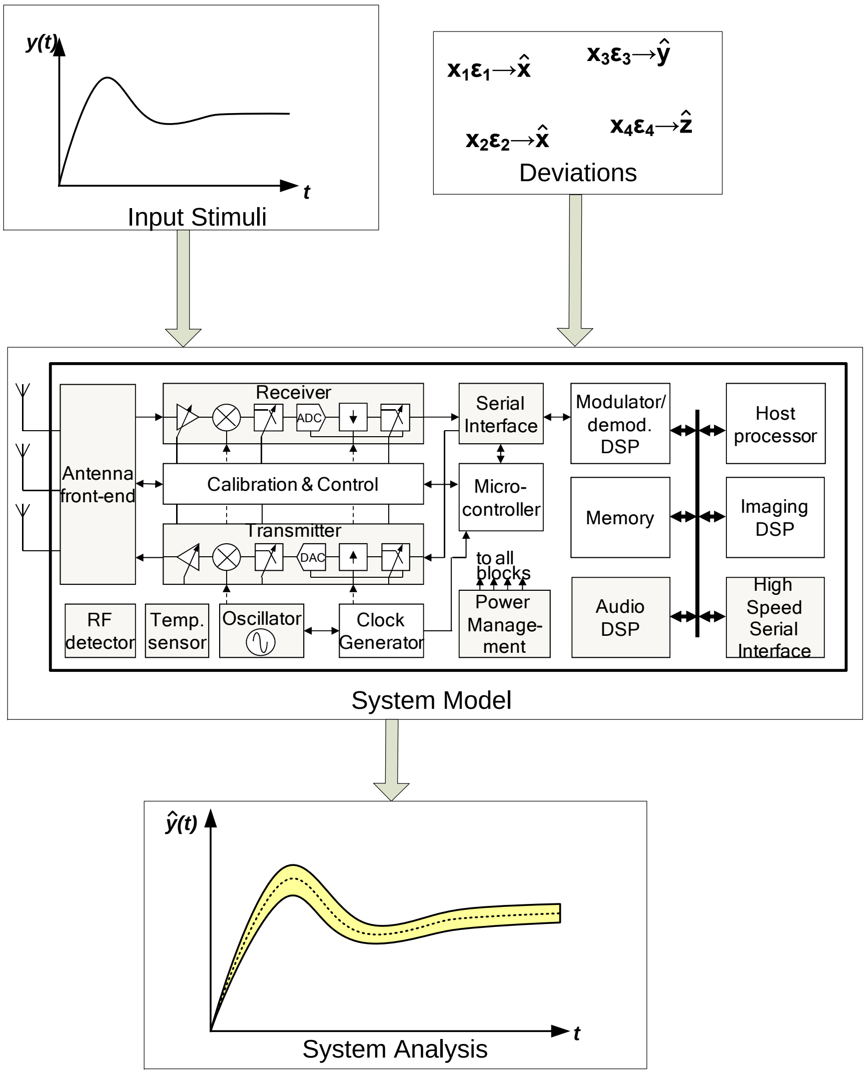

Figure 3.3: Semi-symbolic simulation

Figure 3.3 shows the concept of semi-symbolic simulations on the OSCI motivation example, the universal transceiver. The system model is created as presented by [8], which follows in the nominal model. As semi-symbolic simulations support the combined modeling of nominal parameters and deviating quantities the variational parameter part is described as Affine Arithmetic range. Every parameter deviation adds the system model a sub-range to its description. In the example a total of four ranges are added the system. Two of them (ϵ_1, ϵ_2) contribute to the same parameter $\tilde{x}$ while the others belong to different. The extension of the nominal system model with the deviating parameters result in the definition of the range based system model. All varying parameters which should be considered are added the model and will contribute to the system quantities in a subsequent simulation step. The SystemC AMS framework integrates a simulation engine which is used to simulate the range based system model. An Affine Arithmetic C++ library is linked into the framework and is used to overload all mathematical operations with their range based counterparts. The resulting range based system response is used for analyzes purpose and to identify parameter deviations which have the major impact on the system behavior.

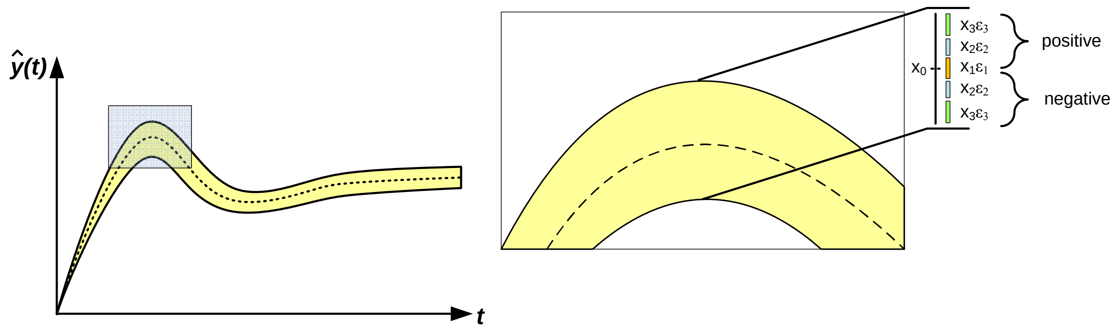

Figure 3.4: Range signal assembled by sub-ranges

The exemplary system response of figure 3.4 illustrates the range signal constitution of individual sub-ranges. The system response is the result from a stimulating input signal feed into the range based system model and the system characteristic affecting it during the simulation. The range based signal assembles of a central value and a set of superimposed ranges. The dashed line in figure 3.4 shows the central value which forms the first part of an AAF $\tilde{y} = x_0$. The set of ranges which contribute to the signal (in this example three $x_1\epsilon_1, x_2\epsilon_2, x_3\epsilon_3$) add up to the full signal range and can be considered as worst case bound of the underlying signal. The epsilon (ϵ) symbols are defined as interval ranges with both sign values $[-1, 1]$ which spans the range also symmetric about the central value. $\tilde{y} = \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i \epsilon_i$ finally defines the set of deviation ranges associated to the range signal $\tilde{y}$. The three deviations of this example equally add positive and negative to the central value. By doing so they span the range within the signal guaranteed lies in for all possible variations of the parameter values that were modeled.

3.2.1 Static and dynamic deviations

Deviations of system parameters appear in differing kinds. Almost every aspect of an electronic system can be source of uncertain value or generally deviation. Is it the most natural form of deviation, part tolerances or more abstract forms like source variations or general inaccuracies. Although the deviations emerge from heterogeneous sources they can be basically divided into two classes. Static deviations that are created at the initializing simulation phase and dynamic effects that are created and added to the system model every time an uncertainty or deviation producing operation is performed. [GHW04] introduced the classification into static and dynamic deviations which will be illustrated in the following enumeration.

When considering $\tilde{y}$ as general Affine Form, deviation models can be considered but are not restricted to:

Static Deviation
Part tolerances: Are considered as deviations $\pm e$ added to the nominal part value y_0 and representing variations in actual part values distributed within the range borders.

$$\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} e \tag{3.1}$$

Voltage Offsets: Adds a deviation symbol $\pm v$ to the system model. Voltage offsets are capable of describing ground variations, potential ground bounces or more general voltage deviations.

$$\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} v \tag{3.2}$$

y_0 specifies the nature of the voltage offset(for instance $y_0 = 0$ for ground variations). *Signal inaccuracies*: Adds a deviation $\pm i$ to the system model which reflects inaccuracies which arise during signal generation

$$\tilde{y} = y_0 * \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} i \tag{3.3}$$

where y_0 gives a nominal signal characteristic (for instance sine function) and the inaccuracy models deviations from this ideal signal shape.

Gain variations: Adds a deviation $\pm g$ to the system model. A gain deviation frequently arises in amplifying or filtering processes when the actual implementation does not follow the idealistic characteristic

$$\tilde{y} = y_0 * \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g \tag{3.4}$$

where y_0 represents the idealistic gain factor.

Additive deviation: More generally deviations can be modeled to influence signal characteristics as additive or multiplicative influence. Additive deviations are considered as deviation $\pm a$ added to the central value of to the Affine Form

$$\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} a \tag{3.5}$$

Multiplicative Deviation: Multiplies the undeviated signal y_0 of unrestricted signal characteristic with a deviation $\pm m$ to span a range around the original signal.

$$\tilde{y} = y_0 * \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} m \tag{3.6}$$

Dynamic Deviation
Dynamic deviations add a deviation symbol to the system model dynamically during the simulation runtime. Hence, the system model is not added one single deviation symbol. One symbol is appended whenever the deviation producing operation is executed. The symbols appearance is also not fixed in simulation time, they appear during simulation runtime.

Quantization error: The quantization operation adds an uncertainty range, representing the maximum possible conversion error to the sampled signal. The quantization error is created every time the conversion is performed, thus adding a range every time of conversion operation. As maximum error margin one half $\pm Q$ of the quantization step is considered. The time dependency of the quantization operation is respected by adding a new deviation symbol every conversion time point n.

$$\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+n)} Q \tag{3.7}$$

Rounding, truncation error: Rounding and truncation uncertainties are closely related to quantization errors. They introduce a representation error of $\pm Q$ and $\pm \frac{1}{2} Q$ for truncation and rounding to the nearest value, respectively.

$$\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+n)} Q \tag{3.8}$$

Above deviation models outline just a few effects which occur in practical systems and are presented for modeling demonstration. For instance, statistic noise contributions could be added the system model [GHW04] which would on the other hand dissolve the formal character of the simulation result. Manifold other deviations can be considered but the modeling properties follow the above examples.

3.2.2 Library implementation

A C++ based Affine Arithmetic library was implemented by [5] under a GNU Lesser General Public License (LGPL). It implemented an Affine Arithmetic Form class and various operations on these classes to allow handling of range based data types. The library is based on the original Affine Arithmetic work [FS97]. The library implementation has been extended in the last years in forms of functionality and implemented operations and has been also published under the (LGPL) license [11].

3.2.2.1 Approximation functions

Non-affine operations demand for an approximation function to compute the resulting affine signal. [FS97] introduced a Chebyshev approximation and min-range approximation to solve non-affine operations. [RSRG12] introduced a combination of Chebyshev and min-range approximation, an interval exact approximation function which avoids over approximations at the range borders but looses sub interval correlations within the resulting ranges.

3.2.3 Range based transistor level solver

[GGB06b, GOB08] introduced a transistor level solver being used to solve and simulate range based system equations. Transistor models are source of strong deviation effects which significantly influence the circuit behavior in real systems. A system description of an differential equation form

3.9).

$$F(\underline{x}(t), \underline{\dot{x}}(t), \underline{u(t)}, t) = \underline{0} \tag{3.9}$$

is hereby solved considering range based signal properties and solving the system equations numerically but still with a guaranteed inclusion of the result. The residual of the numerical approximation is added the system description as supplementary deviation symbol ensuring a formal inclusion of the potential system quantities.

3.2.4 Range based mixed level simulation

Semi-symbolic simulations are applicable on different levels of abstraction. System level simulations model deviations on an abstract level and are especially used for a data flow simulation approach [HGW05]. The desire to add deviation effects on the transistor level to the range based simulation model led to the numerical differential algebraic equation (DAE) solver introduced in [GGB06b]. A comprehensive simulation and analysis framework requires both, the efficient modeling functionality on the system level but as well in depth descriptions of transistor level circuits and their deviations. [ASG12] presented such a mixed level simulation on a basic receiver example. The mixer block in the simulation model was replaced by a corresponding Gilbert cell transistor circuit to improve the analysis accuracy in this selected point of interest.

3.2.5 Garbage collection

The number of deviations in a semi-symbolic simulation model is the main obstacle in achieving fast semi-symbolic simulation runs. [HGW05] introduced the idea of a garbage collection to boost the simulation performance. The idea is to repeatedly evaluate the deviation symbols during the simulation run and if their range is under a pre specified threshold merge them to a single deviation. When starting from a system theory view the system function of a stable system eventually attenuates system quantities in feedback loops. Hence, deviations are also attenuated even if they circle in feedback loops and eventually reach small range values which would be summarized through the garbage process. The garbage collection allows to keep the number of deviation symbols low, even in systems with feedback loops.

3.3 Enhanced modeling and analysis

3.3.1 Improved visualization and tracing properties

The visualization and tracing of range based signals is a fundamental functionality of the presented MARC design framework. The simulation environment SystemC AMS still offers tracing capabilities which can be used generated "‘**.vcd*"’ files or dump signal values in a tabular text file. These integrated tracing capabilities do not work for multiple simultaneous signals like range based signals are. They form the range by a central value and a summation of the sub intervals combined in the overall range of the interval. Hence, they form a tuple $(c_0, +rad, -rad)$ which specify the range quantity by its range boundaries. Johann Glaser implemented a generic multi-trace method [9] in SystemC AMS which allows to trace Affine Arithmetic signals. The visualization of range based signals sometimes require a more complex approach. For instance displaying all sub intervals or showing range based frequency spectra demands for an enhanced visualization tool. Within this work dumping the desired range based signal quantities into a comma separated file "'*.csv"' have solved this issue, as this file can easily be imported into a visualization program like "'sigmaplot"' or can be used for further processing in MATLAB.

3.3.2 Range based Fourier transformation

Frequency domain interpretation of signals provide an important means of analyzing the behavior of simulated systems. Range based system simulations originally concentrated on the time domain. The transition from the time to the frequency domain is not straight forward as the signal quantities not only consist of numerical values but also of symbolic ranges. [SKG+11, SKG+10] introduced a Fourier transformation algorithm applicable also on range based signals. The following sections recapitulates and repeats the fundamentals towards such a range based Fourier transformation originally presented in [SKG+11]. Semi-symbolic simulations provide an important and expressive methodology to analyze the behavior and parameter sensitivity of conservative and non-conservative systems. A behavior analysis can be performed by examining the transient simulation which not only shows the behavior for one parameter realization but provides the system response as signal range caused by all parameters and their variations. However, the analysis is currently restricted to a time domain view of the system behavior. For a basic, elementary class of systems a time domain simulation behavior analysis may suffice. When moving towards more complex systems for a wider field of applications, an analysis gap emerges. Systems with frequency translation structures for instance are hard to be evaluated solely in the time domain. The currently available semi-symbolic simulation methodology does not support a frequency domain signal representation and spectrum analysis. This is the focus of this work. As first, but certainly important intermediate step, the Discrete Fourier Transform has been enhanced to be processable on range based Affine Arithmetic forms. The used semi-symbolic simulation uses the timed TDF model of computation. Thus, the discrete DFT has been chosen because it represents the discrete variant of the widely used Fourier Transform. The Discrete Fourier Transform allows the transition from time to frequency domain and consequently enables a broad field of frequency domain analysis techniques to be applicable on deviated system models.

3.3.2.1 Traditional Fourier Transform

Basically, a Fourier Transform is an operation that transforms complex valued time representatives into their frequency domain counterparts. It is defined for time continuous signals as

$$F(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{3.10}$$

resulting in a complex valued frequency domain quantity. We concentrate on the time discrete transformation, the Discrete Fourier Transform, which is more suitable for the simulation technique used. The environment consists of the timed synchronous data flow models for SystemC AMS and variable discrete time-steps in the SPICE-like transistor-level-solver. The following considerations would also identically apply to the time continuous transformation which indicates no loss of generalization when using the discrete operation.

$$F[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi k}{N}n} \tag{3.11}$$

3.3.2.2 Range based Fourier transformation

To allow the transformation operation to be applicable on Affine Forms, we expand the Discrete Fourier Transform to handle Affine Arithmetic symbols as defined in equation 3.12. As Affine Forms can be considered as superposition of a nominal value with a set of $\mathcal{N}_{\tilde{x}}$ ranges, the Fourier transformation can accordingly also be applied in separate operations. The DFT is a linear operation, therefor the transformation of the Affine Arithmetic symbols simply splits up in a frequency domain superposition of the transformed nominal and partial deviation parts. Using this linear nature simplifies the calculation to a sum of two Fourier transforms, one giving the transformation operation of the nominal value and the second one giving the frequency domain representation of the partial deviation [SKG+10].

$$\tilde{F}[k] = \sum_{n=0}^{N-1} \left(x_0[n] + \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i[n]\epsilon_i[n] \right) e^{-j\frac{2\pi k}{N}n} \tag{3.12}$$

$$\tilde{F}[k] = \sum_{n=0}^{N-1} x_0[n] e^{-j\frac{2\pi k}{N}n} + \sum_{n=0}^{N-1} \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i[n]\epsilon_i[n] e^{-j\frac{2\pi k}{N}n} \tag{3.13}$$

Equation 3.12 shows the structure of the implicit Discrete Fourier Transform operation. The input symbol split up into their nominal values x_0 and the appertaining number of $\mathcal{N}_{\tilde{x}}$ partial deviations $x_i\epsilon_i$. Calculating equation 3.13 results in the generalized $Npoint$ frequency representation of the range based signal $\tilde{x}$. The generalization refers to the $\epsilon_i[n]$ symbols. In this consideration they are time dependent, which means they can represent every value within this interval, independently from its predecessor or successors in time. This behavior perfectly corresponds with the idea of Affine Arithmetic where the partial deviations represent a range of allowed values. The ranges are considered to model the area in where the possible resulting signal values are expected to reside in. The ϵ_i symbols are considered as time dependent, expressed by $\epsilon_i[n]$ which reflects an uncorrelated symbol characteristic. Thus, all general range based Fourier transforms loose the range dependencies which are inherent to Affine Arithmetic. The equation 3.13 has to be calculated using traditional Interval Arithmetic mathematics resulting in a considerable over-approximation and additionally effecting directly the over-approximation with the number of transformation points. One measure to allow a meaningful frequency domain representation of range based signals is to restrict the transformation operation to time independent ϵ_i symbols. The partial deviation envelopes the signal anyway, but the imagined realization which would be a specific value inside the range, stays constant over time. Restricting the transformation to the time independent ϵ_i case, results in the simplified Discrete Fourier Transform given by:

$$\tilde{F}[k] = \sum_{n=0}^{N-1} x_0[n] e^{-j\frac{2\pi k}{N}n} + \epsilon_1 \sum_{n=0}^{N-1} x_1[n] e^{-j\frac{2\pi k}{N}n} \tag{3.14}$$

All ϵ_i symbols in time can be treated as correlated and therefore the partial deviation contribution reduces to a sum over the deviation, transformed by the scalar multiplication of the exponential function. For simplification the ϵ_i is moved in front of the sum, which illustrates the remaining computation. As a result of such a range based transformation we get the Fourier Transform of the nominal signal, superimposed by the frequency representation of the partial deviation. Deviated systems are usually modeled by creating partial deviations for every source of uncertainty.

Whenever a range symbol is created it is labeled by a symbol which identifies the range for the further operations. These partial deviations are basically divided into two groups, static and dynamic deviations. Static uncertainties represent time independent parameters like production tolerances and are modeled in the simulation environment to add a constant deviation to a signal. In contrast, dynamic deviations model time dependent behavior like a quantization error. The uncertainty is different at every simulation time point and is introduced by creating a new deviation symbol at every simulation step. This modeling strategy preserves the source correlation of the symbols as for instance a quantization operation adds uncertainty to the system every time it is performed.
Modeling strategies in semi-symbolic simulations suggest a time independent realization of system ranges. Even, when dynamic deviations are considered, they are added the system model as new ranges with a new symbolic identifier every time the creating operation occurs. Obviously, the number of subranges increase therefore steadily but the time independence is strictly kept. The simplified Fourier transformation equation 3.14 can be used therefore for a wide range of semi-symbolic models respecting the time independence of deviation terms.

3.3.2.3 Amplitude Frequency Spectrum

Calculating a Fourier spectrum of range based signals following equation 3.14 and the according argumentations solve to a straightforward task. The crucial process shifts to the identifying of the worst case bounds for the transformed range based signal. This identification requires careful considerations and is divided herein into two steps. The calculating of the amplitude and the phase spectrum. Three characteristics have to be determined. The nominal range center point and the minimal and maximum boundaries for both the amplitude and the phase spectrum. A superposition of these three characteristics construct the range based frequency spectrum. A calculation of these spectrum properties have to be performed for every frequency sample to obtain the full frequency behavior.

The amplitude spectrum properties are determined by simple trigonometric calculations where the minimal and maximum amplitudes are the diagonal corner points of the deviated signal area in the complex plane. Figure 3.5 shows the relationship between the complex real and imaginary parts and the corresponding absolute value of the polar form representation. The amplitude spectrum properties are derived by correlating the minimum, nominal and maximum complex parts, respectively and thus determining the according absolute quantity. An exception occurs when the deviated area embeds the complex point of origin. In such cases the distance to all four quadratic corners are calculated and a simple minimum-maximum identification is performed.

3.3.2.4 Phase Frequency Spectrum

Figure 3.6 finally shows the construction of the phase spectrum from a Fourier transformed range signal.

Two different range signals denoted by $\tilde{F_1}$ and $\tilde{F_2}$ are presented to illustrate the phase determining operation more clearly. The identifier k selects the single frequency samples whereas the two range signals are completely independent. $\tilde{F_1}$ represent a range based signal in the first quadrant of the complex plane. The deviations cause a quadratic area superimposed the nominal mid-value and representing the maximum deflection. The minimum phase margin is found by correlating

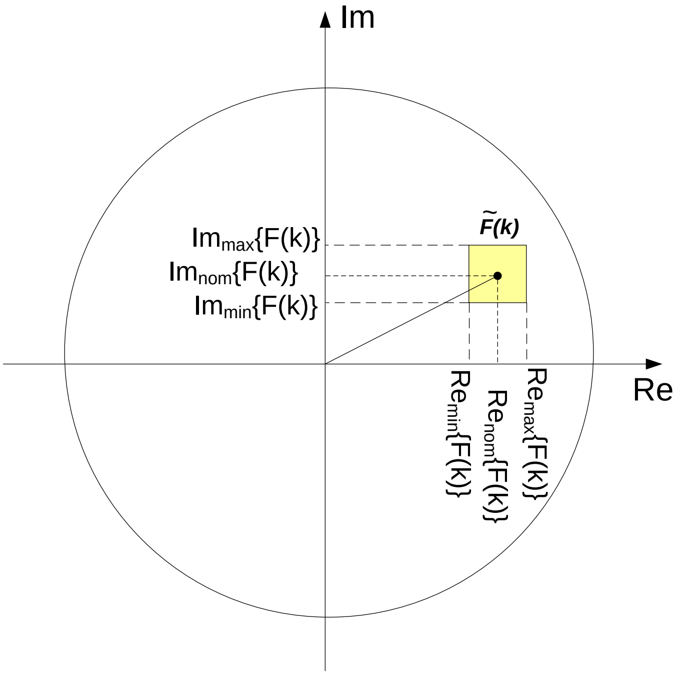

Figure 3.5: Range signal in the frequency domain and its magnitudes

the maximum real part with the minimum imaginary part and the maximum margin by coupling the minimum real part with the maximum imaginary. When analyzing the deviation area of $\tilde{F}_2$ the minimum and maximum values are determined in an inverse operation. The minimum margin is the phase of the minimum real part and maximum imaginary vector and the maximum results from the maximum real part and minimum imaginary vector. Accordingly, the position of the range in the complex plane has to be considered for the phase properties calculation. The relevant range corners switch between the first and second plane and the calculation is mirrored for the third and fourth quadrant. All combinations, as the mixed location in two quadrants or the embedding of the point of origin require a specific determination with deriving all range corner phases and deciding the minimal and maximum values.

3.3.2.5 Applied range based Fourier Transform

Technical systems are usually modeled by creating partial deviations for every source of uncertainty. Whenever a range symbol is created it is labeled by a symbol which identifies the range for the further operations. These partial deviations are basically divided into two groups, static and dynamic deviations. Static uncertainties represent time independent parameters like production tolerances and are modeled in the simulation environment to add a constant deviation to a signal. In contrast dynamic deviations model time dependent behavior, like a quantization error. The uncertainty is different at every simulation time point and is introduced by creating a new deviation symbol at every point in time. This modeling strategy preserves the source correlation of the symbols as for instance a quantization operation adds uncertainty to the system every time it is performed. In the last section it was defined that a Discrete Fourier Transform is applicable in its simplified form, when the single ϵ_i symbols represent time independent deviations. For most

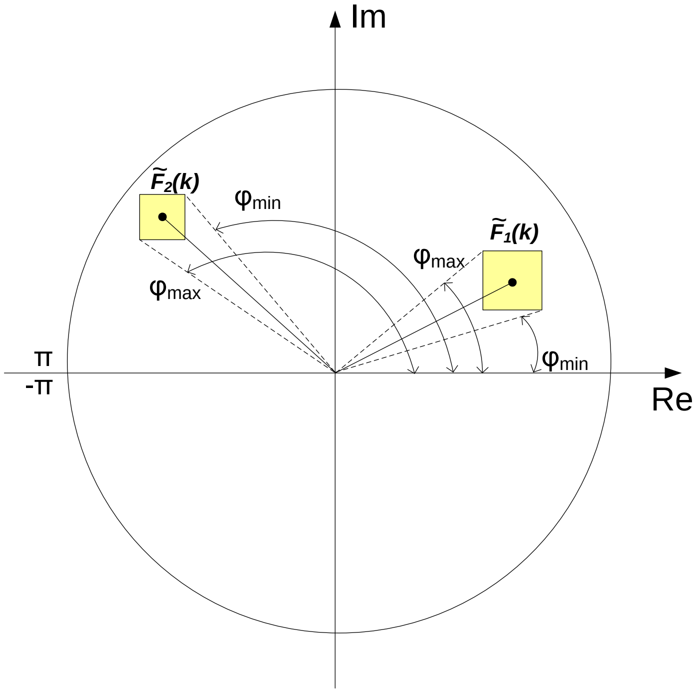

Figure 3.6: Two range signals in the frequency domain and their phase values

of our modeled systems this assumption can be proved. A time independence is the source of our technical modeling strategy to allow operations on correlated intervals. Thus, identical intervals are correlated even when delayed in time by the system model.

3.3.2.6 Fourier Analysis Demonstration

For demonstrating the applicability of the Discrete Fourier Transform deduced in section 3.3.2.2 one semi-symbolic simulation examples is presented in this section. As a semi-symbolic simulation can be applied on different levels of abstraction its usage is shown on an abstracted system level.

A Fourier spectrum on system level, for instance, is particularly helpful for analyzing the system behavior of more complex communication systems when mixer structures in combination with suppressing filter components are used. Hereby, the range based Fourier spectrum is used to analyze the influence of single deviations on the spectral components of the system. The resulting frequency spectrum shown in figure 3.8 gives the spectral components remaining in the system model output signal. The spectrum does not only provide the frequency behavior of the nominal system model but it also delivers an envelope which forms the boundaries of the range defined system behavior.

Figure 3.7 shows the particular demonstration system. The receiver structure is modeled as range based system model respecting the deviation sources. A corresponding semi-symbolic simulation produces a range based system output which reflects the impact of the deviating parameters on the ideal system behavior and which is subsequently analyzed in frequency domain by calculating the Fourier transformation. The resulting complex valued frequency spectrum is finally constructed in a corresponding amplitude spectrum. As deviation sources, gain variations of the filter and

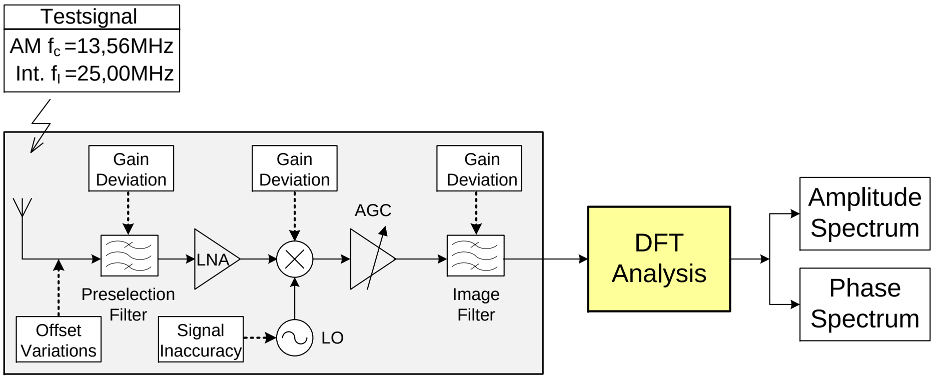

Figure 3.7: Range based frequency analysis

the mixer components, a signal inaccuracy of the local oscillator sine wave and an abstract offset deviation has been added. The receiving signal is constructed nominally without deviations as amplitude modulated (AM) test signal operating on a 13,56 MHz carrier, affected by a 25MHz interferer. The single deviations, associated to each receiver component create a range based signal representing all potential variations from the nominal, ideal value. This range signal, considered as range based system response is passed on to a *DFT Analysis* block which calculates the Fourier transformation following equation 3.14. The resulting complex valued frequency domain representation is finally used to construct a range based amplitude frequency spectrum.

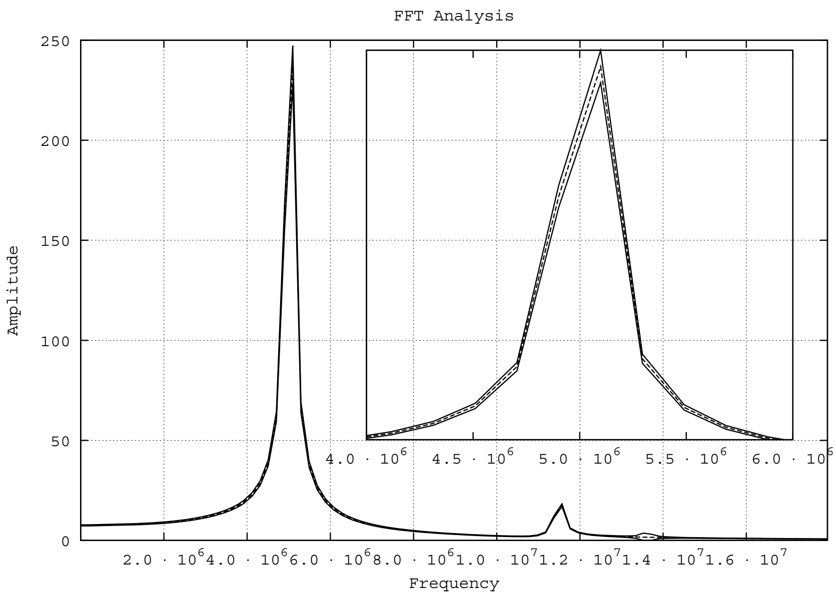

Figure 3.8: Range based amplitude spectrum

Figure 3.8 shows the simulated amplitude spectrum, giving the nominal values as dashed line and the deviation bounds as summation of the single ranges in solid. In the region of 12 MHz an influence of the HF interferer still remains and the 5 MHz range based amplitude peak shows the original baseband test signal affected by the non-ideal receiver structure. The in-figure shows a zoomed portion of the signal peak area showing again the composition of the amplitude spectrum by the minimum and maximum bound and the inner nominal quantity.

3.3.3 Range based Laplace description

Describing systems in the Laplace domain significantly reduces the modeling effort at system level. [OSG11b] introduced a range based Laplace method which allows the direct description of transfer functions in the Laplace domain. Basically SystemC AMS supports entering of transfer functions in the Laplace domain. Unfortunately the integrated Laplace transfer function (LTF) does not support range based quantities. Hence, an improved Laplace function capable of handling range quantities was implemented. The following considerations were presented in [OSG11b]. The function bases on the finite difference approximation where the time step h of the simulation is considered sufficient small compared to the system frequencies. The derivation of quantities simplifies to

$$y'(t) = \lim_{h \to 0} \frac{y(t) - y(t-h)}{h} \approx \frac{y(t) - y(t-h)}{h} \tag{3.15}$$

A Laplace transfer function originally specified in the Laplace domain by

$$H(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + ... + b_0)}{a_n s^n + a_{n-1} s^{n-1} + ... + a_0} \tag{3.16}$$

is describable in time domain by the following equation

$$\sum_{i=0}^{n} b_i x^{(i)}(t) = \sum_{i=0}^{n} a_i y^{(i)}(t) \tag{3.17}$$

which can be easily solved by the approximation presented in equation 3.15 [OSG11b]. As long as the simulation time step is sufficient small the result of the Laplace transfer function can be considered exact. Filter models on system level reduce to a simple description of their transfer function which significantly improves the modeling capabilities.

3.3.4 Range based bit width determination

Range based uncertainty modeling is also used to determine the optimal bit width in digital systems. Quantization, rounding and truncation errors negatively influence the signal to noise ratio of a signal processing system. Finding an optimal bit width in every stage of the system is a challenging task. The implicit error propagation capabilities of Affine Arithmetic supports the analysis of correlated error effects. [LAOW05] introduced a complete framework called "‘MiniBit"’ to estimate the optimal bit widths in a digital system. [LCNT07] enhanced this idea and boosted the computation performance by neglecting the higher order effects in approximation operations. This approach looses the pessimistic character of Affine Arithmetic and is therefor not qualified for formal considerations. [OSG11a] presents a technique for an error bound estimation of fix-point as well as floating-point systems on the basis of SystemC AMS. The main focus is on the range based modeling in SystemC AMS.

3.3.5 Range based jitter model

Jitter of signals and especially phase jitter is a well suited candidate for being modeled by ranges. Jitter is typically considered as additional noise or in a stochastic way. Semi-symbolic simulations provide now the possibility to define it with its worst case bounds. The jitter variance will be defined to stay within the range boundaries and thus providing a formal description for the jitter effect. Equation 3.18 defines the jitter variation as range by specifying it in an Affine Form

$$\tilde{\phi} = \phi_0 + \epsilon_1 \phi_1 \tag{3.18}$$

where ϕ_0 gives the potential phase jitters mid-value and $\epsilon_1 \phi_1$ scales the jitter deviation and completes the Affine Form. The range based phase jitter can be specified either static or dynamic where the dynamic modeling would allow a varying jitter shape throughout the simulation run.

For a first simplification the usage of the jitter model is restricted to sine valued signals which can be specified by

$$y(t) = Asin(2\pi f_0 t + \tilde{\phi}) \tag{3.19}$$

where $y(t)$ represents any general sine wave with magnitude A and the argument $2\pi f_0 t + \tilde{\phi}$ which specifies a sine function of specific frequency f_0 with a range phase jitter of $\tilde{\phi}$. Using the trigonometric identity

$$Asin(x + y) = Asin(x)cos(y) + Asin(y)cos(x) \tag{3.20}$$

the formula divides to

$$y(t) = Asin(2\pi f_0 t)cos(\tilde{\phi}) + Acos(2\pi f_0 t)sin(\tilde{\phi}) \tag{3.21}$$

Let us now assume that phase jitter can be considered as a weak deviation. The jitter stays sufficient small and can be approximated as $\tilde{\phi} << 1$. Under this assumption the formula can be reduced to

$$y(t) = Asin(2\pi f_0 t) + A\tilde{\phi}cos(2\pi f_0 t) \tag{3.22}$$

The small noise approximation has converted the phase noise into a multiplicative noise portion which is now usable for range based modeling. The small noise assumption is a rather good estimate as higher levels of signal jitters would cause the system to fail in its function.

4 MARC Design Environment

The MARC design environment is designed to support a deterministic design refinement process of systems affected by parameter deviations and uncertainties. The idea is to use range based system models, simulate them in a semi-symbolic approach and determine the best suited refinement candidates to refine the system and hence improve the system quality. The advantage of range based system simulations in this scope is twofold. The number of necessary simulation runs is reduced to one by the semi-symbolic approach and the symbolic nature of the deviation symbols allow an efficient even deterministic backtracking of output signal portions to their causing system parameters. This flexibility strongly supports the idea of a deterministic design refinement. The design environment consists of a semi-symbolic simulation framework based on Affine Arithmetic and analysis and tracking functionalities capable of handling range based signals. The simulation framework bases on SystemC AMS which is extended by the Affine Arithmetic library mentioned in section 3.2.2, available at [11] and a new implementation of the library presented in [RSRG12]. Range based system models which are used for semi-symbolic simulations are created within this work in three phases.

- Numeric SystemC AMS model
- Specification of the deviation effects
- Combined range based system model

As a first step the traditional numerical SystemC AMS mixed-signal model is created. This model represents the nominal, intended realization of the application system. All realizations of a system introduce deviations and uncertainties in several parameters of the system. These deviations are identified and added as Affine Forms the initial numerical model to create the complete range based system model. SystemC AMS supports the easy integration of application specific add-on libraries and functionalities. As the core language is C++ extensions to the SystemC AMS functionality are easy integrable if they are also written in C++. The specification of SystemC AMS is flexible enough to allow the integration of an extended function set. Figure 4.1 presents the previously mentioned general SystemC AMS layered approach. When using SystemC AMS for semi-symbolic simulations the SystemC AMS simulation framework is extended by an Affine Arithmetic library "‘*aaflib*"’ which declares the AAF objects and defines mathematical operations on them for operator overloading. The "‘*aaflib*"’ implementation is written in pure C++ and is therefor independent from the SystemC AMS framework. The library is simply included in the SystemC AMS mixed-signal models and extends the functionality to Affine Arithmetic based range signals. The Model of Computation (MoC) which is used for the model description is principle not predefined but limits to the ones for which operators are implemented for overloading. The mixed-signal applications presented in this work are specified in the *Timed Data Flow (TDF)* MoC which are described at system level. The linear DAE solver integrated in SystemC AMS is not capable of handling range based signals, hence LSF and ELN description models are not suitable for the semi-symbolic simulation technique.

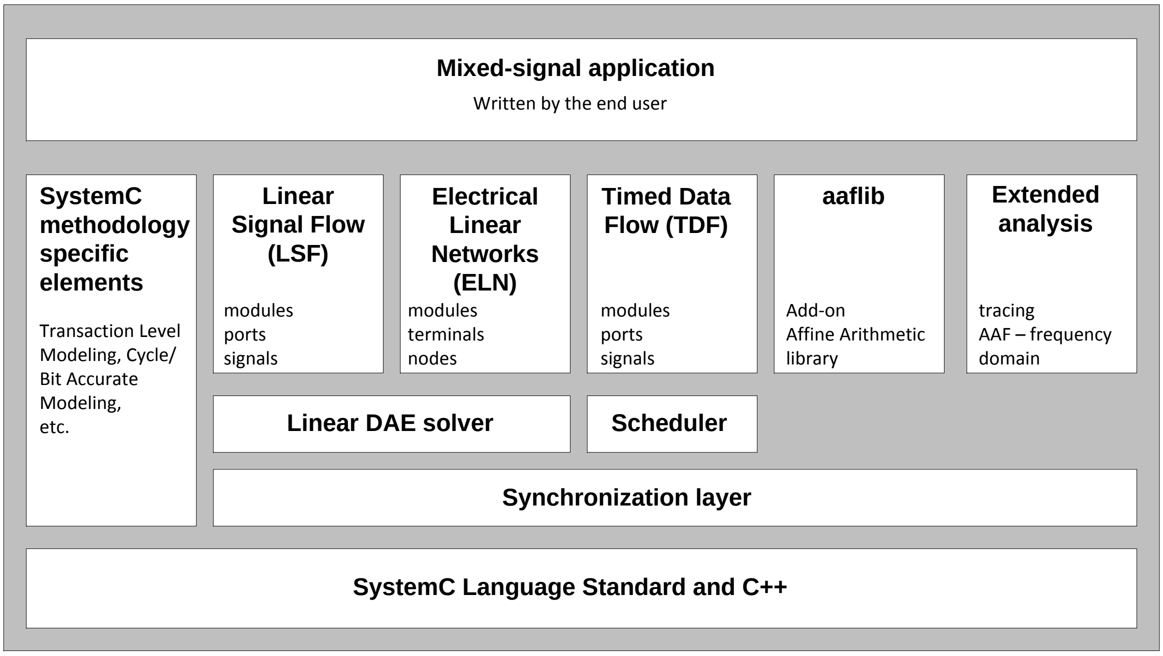

Figure 4.1: SystemC AMS layers with extensions

The SystemC AMS environment is additionally extended by tracing and analysis functionality which broadens the scope for range based quantities. The analysis capability within the semi-symbolic simulation is crucial for the MARC design environment. SystemC AMS already supports various analysis methods for data types specified in the "'*OSCI SystemC AMS extensions Standard 1.0*"' [7]. Range based signals and quantities prohibit the usage of most of this predefined functions. Hence, additional methods for range based signals have to be implemented and integrated into the SystemC AMS environment. Figure 4.1 shows how the extended functionality integrates into the SystemC AMS layer structure and environment.

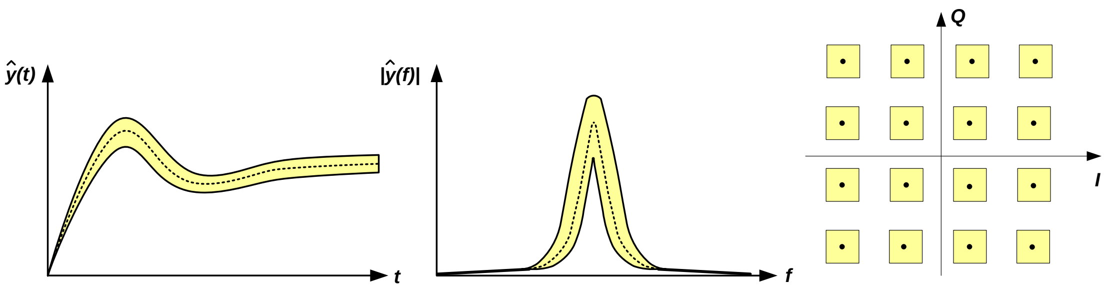

Figure 4.2: Sample analysis methods: a) time domain b) frequency domain, c) constellation diagram

The extensions are coded in C++ and integrate into the SystemC AMS environment parallel to the implemented *Models of Computation*. The mixed-signal models are coded in the desired MoC and the extended functionality is used for the range based quantities. Figure 4.2 gives a schematic overview of advanced analysis techniques and how they support the system analysis for range based quantities. Sub figure a) shows a schematic range based system response in the time domain obtained by a semi-symbolic simulation with at least one source of parameter deviation. The summation of the sub ranges results in the outer bound of the range signal which is identical with the worst case bound in which the potential signal guaranteed lies in. Sub figure b) represents the frequency domain characteristic of a potential signal. The frequency domain analysis especially allows the analysis of spectral components and provides an additional powerful technique to analyze communication systems in particular. The third analysis technique presented in c) is a range based constellation diagram. Parameter variations and uncertainties in communication systems deviate the received signal from the ideal constellation point. The range based deviation quantities span an area around the constellation points, revealing the potential signal deviations. As long as these deviations areas do not overlap the receiver theoretically is still able to correctly detect the coded symbol information.

4.1 Range based system models

Semi-symbolic simulations use range based system models for their simulation and perform the computations by over loading the according mathematical operations with their range based counterparts. The system models are created using the SystemC AMS description syntax and use the "'*Timed Data Flow*"' MoC within this work. A major modeling question is which system parameters are sources of variations, by which value they deviate and which uncertainties arise when considering the intended system. This identified deviations are then added to the system model as range descriptions and accordingly contribute to the system quantities in the subsequent semi-symbolic simulation.

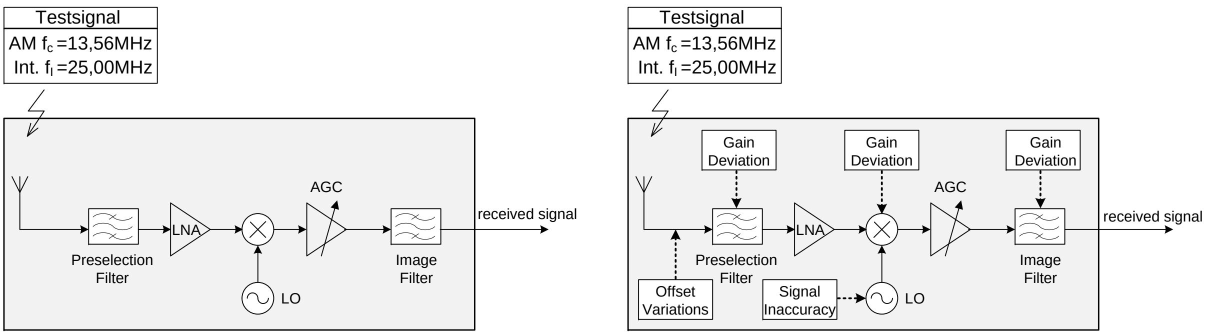

Figure 4.3: a)SystemC AMS model b) Range based model

The deviations can be caused by various sources. They potentially arise when moving from theoretical system descriptions to real implementations. Bit widths in electronic systems introduce quantization errors which can be handled as deviations from the discretized value. Part tolerances and inaccuracies in mathematical functions contribute to the system deviations as well as uncertainties in the model building but also unspecified properties of the actual implementation. Uncertainties in the model cover all aspects that are not considered and contribute as abstract summarized deviation to the range based system model. Uncertainties of the implementation can be seen as system properties that are not fully specified during the implementation phase but should still be considered in the analysis process.

4.1.1 Deviation effects in electronic systems

Electronic systems compose of diverse components which all are described by parameters that potentially vary in their value. The variation is initially specified as a worst deviation from the nominal, intended value. In the scope of semi-symbolic simulations and range based system models these single worst case definitions are translated into range descriptions characterized by Affine Arithmetic. Uncertainties in this field represent just a different form of deviation from a nominal parameter and will also be described as deviation range by Affine Arithmetic. The sources of deviation in electronic systems are manifold, all parameters potentially vary in their respective value and multiple uncertainties exist for a modeled system. In the following an incomplete enumeration of practical deviation effects and their range specification will be presented. $\tilde{y}$ represent in all cases the range signal, y_0 gives the nominal value and $\epsilon_{(\mathcal{N}_{\tilde{y}}+1)}$ indicates the deviation symbol appended the set of already existing deviation effects. The generic symbol g is used in all specifications to hold the particular variation quantity.

- **component tolerances:** $\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g$ describe the general deviation effects caused by component tolerances. The tolerance is added to the nominal part quantity as maximum deviation and is added the set of previously defined deviation symbols.

- **gain deviation:** $\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g$ specifies a generic gain deviation which shows a variation in the actual amplification factor given originally by its nominal factor y_0. This nominal factor is deviated by the variational factor $\epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g$. This gain formula specifies only the deviated gain factor and must additionally be applied on the appropriate operand.

- **opamp gain variation:** $signal * \tilde{y}$ models the gain variation of an opamp using the previously specified gain deviation. The gain factor is divided into two parts, the nominal gain factor and the deviation term which defines the potential variation. *signal* represents the actual quantity which should be amplified.

- **mixer gain variation:** $(signal1 * signal2) * \tilde{y}$. A mixer gain variation translates into a multiplicative deviation effect. $signal1, signal2$ are the two input signals to the mixer operation. The multiplicative range based $\tilde{y}$ factor models the gain variation of the resulting mixer output.

- **voltage variation:** $\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g$ A general voltage variation is specified by an intended voltage quantity and a deviation term giving the variation from this intended value.

- **supply voltage variation:** $\tilde{y} = Vcc + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g$ describes a general supply voltage variation in range based form. The intended voltage is the supply voltage Vcc and the variation can be either a stationary effect like a voltage ripple or even stochastic effects that appear within the specified range.

- **ground variation:** $\tilde{y} = 0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)} g$ Ground variations are very similar to supply voltage variations. Ground variations are additive deviations which model ground line effects like ground bounce or distortions.

- **signal inaccuracies:** $signal * \tilde{y}$. Signals in electronic systems are always non-ideal. If they are constructed they always introduce a divergence from the ideal characteristic (i.e. quantized sampling points for a constructed sine wave in digital systems). This divergence from the ideal characteristic is summarized as inaccuracy in signal representation and is for instance range based modeled by a multiplicative deviation of the intended signal.

- **model uncertainties:** $\tilde{y} = 0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)}g$ describes a very general "‘model uncertainty"’. The idea is to add an extra deviation to the range based system model which respects unconsidered uncertainties that add to the system behavior. These uncertainties could be higher order effects which are neglected in the system model set-up or other minor deviations which are abstracted in this deviation term.

- **quantization error:** $\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+n)}g$. Mixed-signal systems comprise of analog as well as digital sub systems. The digital parts operate on quantized signals which introduce an error at the conversion stage. The worst case quantization error is plus/minus one half of the quantization step $\pm\frac{1}{2}q$ that is represented by the g symbol. The quantization error is added the range based system model every time a conversion operation from analog to digital is performed.

- **rounding and truncation:** $\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+n)}g$. Rounding and truncation are closely related to the quantization error. They do not necessarily appear in conversion operations they also appear during calculations in the digital domain with bit width restricted resources. A classical rounding exhibits a worst case error of one half of the rounding step $\pm\frac{1}{2}r$, the same quantity as the previously quantization error. The truncation error in contrast introduces a possible error quantity of one full truncation step $\pm t$.

- **sine phase jitter:** $sin(\omega t) + \tilde{y}cos(\omega t)$. The phase jitter of a general sine wave can be characterized as additive cosine shaped range quantity as specified in 3.3.5. The phase variation is considered as small and modeled again as worst case jitter value, representing the maximum appearing jitter value.

- **signal noise:** $\tilde{y} = y_0 + \epsilon_{(\mathcal{N}_{\tilde{y}}+1)}g$. Noise is a very common factor in communication systems. Noise can be defined as range quantity where a boundary is defined in which the noise signal most probably resides in. Noise is a stochastic quantity, hence no formal inclusion of all signals is guaranteed, instead a probabilistic one is considered.

Basically the deviation effects are construct able by range descriptions presented in section 3.2.1. Restrictions arise only from Affine Arithmetic and its mathematical description. The mathematical functions that are applicable are the affine operations and a set of non-affine operations which are formally estimated by approximations as specified in section 2.3. The descriptions of deviation effects are not restricted to single descriptions, even combined range effects and their description by complex formulas are possible. A main requirement in the field of semi-symbolic simulations remains for the model designer, to decide which deviation effects to consider in the range based system model and which effects could be ignored. The hierarchical structure of SystemC AMS supports this requirement. The relevant deviation sources are individually considered in every sub-block of the range based system model but commonly contribute to the overall system behavior. Neglected minor effects and possible uncertainties in model creation are finally accounted for by a generic uncertainty deviation term on system level.

4.1.2 Complex deviation effects

Deviations of parameters in heterogeneous systems emerge in various ways. Affine Arithmetic provides basic mathematical functions to construct range representations of deviation effects for use in a range based system model. Deviations are even describe able as complex formulas as they often approach the desired characteristic more accurate. Important to note here is that inaccuracies in modeling the deviation effects always reflects in an over-approximation of the real characteristic. The formal inclusion paradigm is always satisfied. In the following, several "'real life"' deviation descriptions are presented and discussed in respect of their suitability for system simulations.

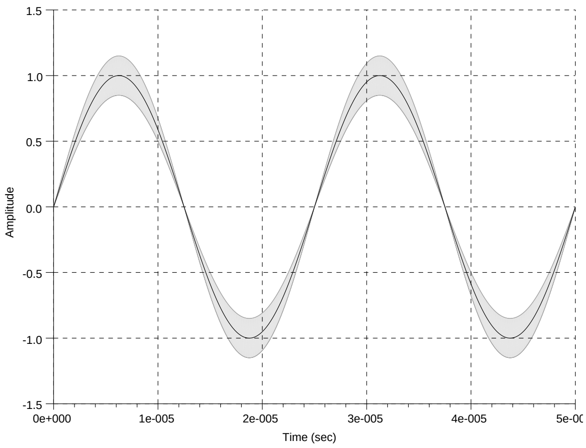

Figure 4.4: sinusoidal function with multiplicative deviation

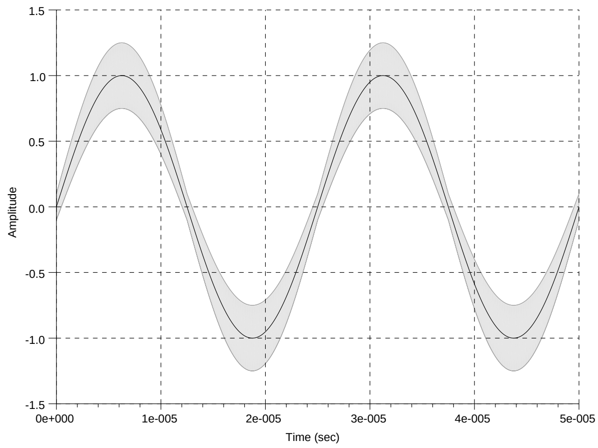

Figure 4.5: sinusoidal function with multiplicative and additive deviation

Figure 4.4 shows a simple multiplicative deviation of a sinusoidal quantity. Sine waves are common quantities in embedded systems as they are for instance commonly used as "'local oscillator"' in mixing stages of typical communication systems. Equation 4.1 gives the deviation model for figure 4.4 with the according deviation quantity.

$$\tilde{z} = sin(\omega t) * \tilde{x} \qquad \tilde{x} = 1.0 + 0.15\epsilon_1 \tag{4.1}$$

This deviation is typical for inaccuracies in analog signal generation. The multiplicative nature assumes a higher inaccuracy at higher signal values but expects no deviation at all at indefinite small quantities. When assuming a signal generation in the digital domain, quantization steps and limited bit length will most prominent contribute to the signal inaccuracy.

$$\tilde{z} = sin(\omega t) * \tilde{x} + \tilde{y} \qquad \begin{aligned} \tilde{x} &= 1.0 + 0.15\epsilon_1 \\ \tilde{y} &= 0.0 + 0.15\epsilon_2 \end{aligned} \tag{4.2}$$

Inaccuracies in the digital domain arise as step wise uncertainties that originate from quantization, truncation and rounding in signal processing operations. The step wise characteristic will be modeled as additive deviation that considers fixed uncertainties also at small system quantities. Equation 4.2 shows a combined multiplicative and additive deviation superimposed an ideal sine wave. This deviation model represents inaccuracies in signal generation in an improved manner and is given in figure 4.5.

A diode characteristic is often approximated by an exponential function based on the "'Shockley diode equation"'. All approximations and considerations of non-ideal effects, uncertainties and deviations influence the actual signal characteristic of the diode model. A range based exponential function integrates these effects into an implicit function description.

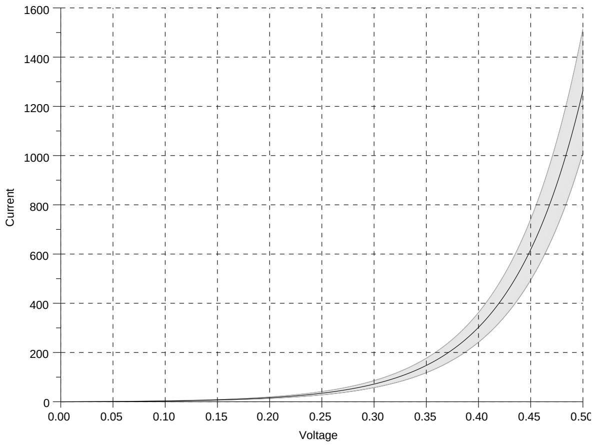

Figure 4.6: diode function with multiplicative deviation

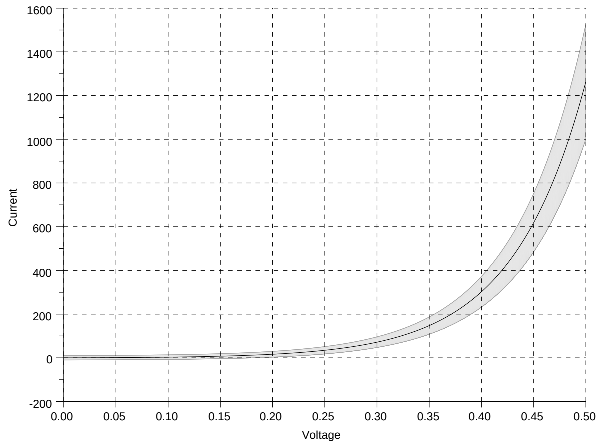

Figure 4.7: diode function with multiplicative and additive deviation

Figure 4.6 shows a diode characteristic extended by a multiplicative deviation model. The current shows a relatively flat increase until the diode voltage passes the thermal voltage and succeedingly rises following and exponential function. The multiplicative deviation, again models rising deviation figures with rising current values.

$$\tilde{z} = e^{\frac{t}{U_T}} * \tilde{x} \qquad \begin{aligned} \tilde{x} &= 1.0 + 0.2\epsilon_1 \\ U_T &= 0.2V \end{aligned} \tag{4.3}$$

This would for instance approximate variations in the signal slope caused by temperature drifts or by variations of component characteristics. Figure 4.7 extends the deviation model by an additive part. This deviation model considers a combination of additive and multiplicative deviation effects contributing to the diode characteristic. The range nature of the deviation model allows an abstracted modeling of variations in diode characteristic with however covering complex and higher order effects in the range area.

$$\tilde{z} = e^{\frac{t}{U_T}} * \tilde{x} + \tilde{y} \qquad \begin{aligned} \tilde{x} &= 1.0 + 0.2\epsilon_1 \\ \tilde{y} &= 0.0 + 10.0\epsilon_1 \\ U_T &= 0.2V \end{aligned} \tag{4.4}$$

Equation 4.4 specifies the range based exponential function used to describe the diode characteristic.

Many components in system simulations show a non-linear characteristic. The diode characteristic for instance has a non-linear(exponential) relation between the voltage and current values. Semi-symbolic simulations approximate non-linear operations on ranges by adding approximation deviations to the system. Vast usage of non-linear operations slow down the simulation performance and obfuscate system behavior by over-approximation. The desire in semi-symbolic simulations is to reduce the number of non-affine operations to a minimum. The requirement to model component properties as accurate and detailed as possible opposes the wish to reduce the complexity of simulation models. Range based descriptions introduce the possibility to abstract and by this simplify simulation models without loosing accuracy of the considered effects. Component characteristics are simplified (linearized) and all neglected effects (non-linearity, higher order effects) are covered by a superimposed approximation range.

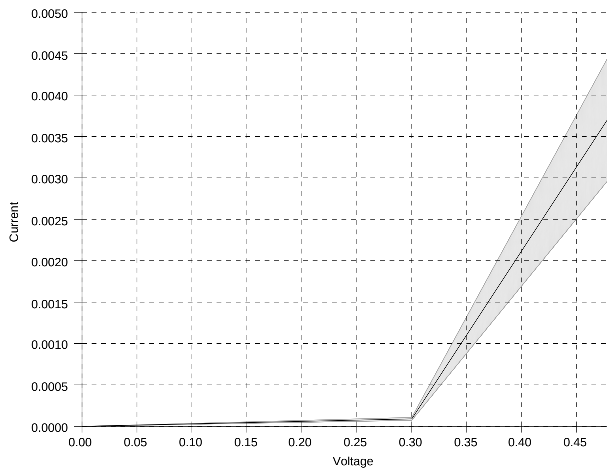

Figure 4.8: abstracted diode function with multiplicative deviation

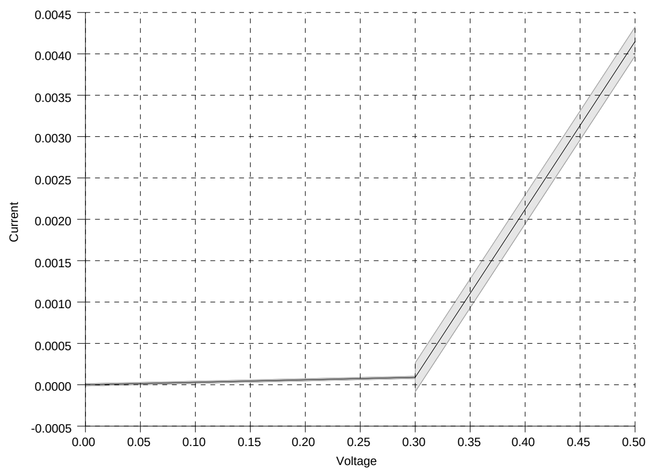

Figure 4.9: abstracted diode function with additive deviation

Figure 4.8 illustrates such an abstracted function of the original presented exponential diode characteristic. The function is approximated by piecewise linear (two in this example) functions which change at the thermal voltage and considers the simplified properties as multiplicative deviation. The plot is rescaled in its axis to visualize the typical diode characteristic.

$$\begin{aligned} \tilde{z} &= 3 * U * \tilde{x} \qquad & U &\leq U_T \\ \tilde{z} &= 200 * U * \tilde{x} \qquad & U &> U_T \\ & & \tilde{x} &= 1.0 + 0.2\epsilon_1 \\ & & U_T &= 0.3V \end{aligned} \tag{4.5}$$

A multiplicative deviation in this special case does not accurately cover the neglected model behavior. A additive deviation is better suited which is shown in figure 4.9 and described in equation 4.6.

$$
\begin{aligned}
\tilde{z} &= 3 * U + \tilde{x} & & U \leq U_T \\
& & & \tilde{x} = 0.0 + 2.0\epsilon_1 \\
\tilde{z} &= 200 * U + \tilde{x} & & U > U_T \\
& & & \tilde{x} = 0.0 + 25.0\epsilon_1 \\
& & & U_T = 0.3V
\end{aligned}
\tag{4.6}
$$

Deviations in semi-symbolic simulations appear in diverse forms. They range from simple additive or multiplicative deviations to complex combined functions. The system designers objective is it to model deviation properties as accurately as possible and with the least over-approximation as achievable. Complex deviation descriptions support these requirements with not restricting the description capability within the mathematical possibilities of Affine Arithmetic.

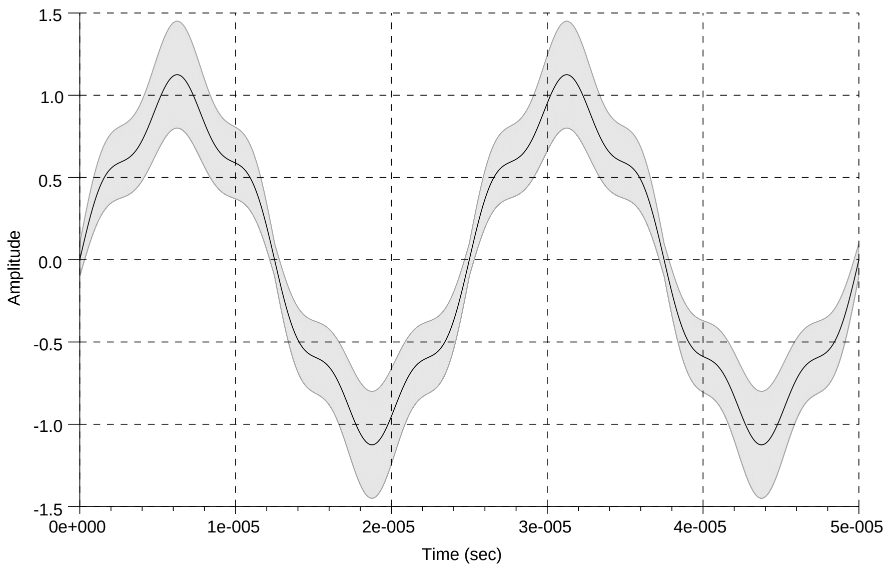

Figure 4.10: sinusoidal function with superimposed complex deviation

Figure 4.10 shows one constructed possibility for a sine signal superimposed by a complex (again sine shaped) deviation model.

$$
\begin{aligned}
\tilde{z} = sin(\omega t) * \tilde{x} + sin(5\omega t) * \tfrac{\tilde{x}}{8} + \tilde{y} \\
\tilde{x} = 1.0 + 0.2\epsilon_1 \\
\tilde{y} = 0.0 + 0.1\epsilon_2
\end{aligned}
\tag{4.7}
$$

Equation 4.7 specifies the modeled deviation property. A sine wave is superimposed by a sine wave shaped deviation of different frequency with an additional multiplicative deviation. This deviation characteristic is intended only to illustrate complex deviation models and show the easy concatenation of deviation properties to complex formulas. The limitations lie in the mathematical definitions of Affine Arithmetic and the implemented simulation library, but also influences the simulation performance of the semi-symbolic simulation run.

4.1.3 Range based model creation

The range based system model, which is the basis for a semi-symbolic simulation is created in a double step. The nominal parameters of the system are used to design a numerical system model that is following added the symbolic deviation models

- creation of nominal model
- integration of deviation models

to finally result in the complete range based model that allows a semi-symbolic simulation of the system under investigation. The deviation effects which are added the range based system model depend on the scope of the system analysis. When a bit width optimization is the refinement objective of the simulation run then mainly the quantization, rounding and truncation errors will be considered for the model creation. Minor contributors, like offset variations or even model uncertainties will not significantly influence the refinement objective (the bit widths in the system).

4.1.4 Mixed-level semi-symbolic simulations

The idea of semi-symbolic simulations on more than one abstraction levels were presented in section 3.2.4. Different systems demand for different optimization and analysis strategies. A hierarchical approach supports this desire to scale the modeling and simulation effort. A mixed level simulation support introduces the possibility to evaluate deviation effects on differing abstraction levels. Deviations are modeled on levels where they can be specified most accurately. For instance a mixers non-ideal behavior is specified on transistor level, while the rest of the communication system is described at system level. The overall behavior of the system can be evaluated at system level and the transistor level properties contribute to the full system.

4.2 Backtracking properties of Affine Arithmetic

Semi-symbolic simulations based on Affine Arithmetic combine numerical simulations with symbolic terms that specify deviation quantities as ranges. Every deviation term is associated with a distinct symbolic identifier (ϵ) that allows the identification of the corresponding source throughout the whole simulation process and time. Deviations in semi-symbolic simulations are basically divided into two groups. "'*User Deviations*"' that are intentionally created to model system deviations in the actual system model and "'*System Deviations*"' that are added by the simulation framework to cover approximation errors originated by non-affine operations. While user deviations are assigned an identifying symbol during creation, system deviations are associated with the causing operation and sources. A string based description specifies the sources of every deviation which allows a comprehensive identification of every deviation and its associated source. [RSRG12] provides a comprehensive discussion on how to efficiently store and track deviations in semi-symbolic simulations and also provides library structures to allow backward and forward tracking of deviations within simulations.
The single deviations of a range based signal superimpose to a joint range signal giving the worst case behavior reachable by the deviated quantities and the system function. But still the range

based signal comprises of sub ranges that summarize to the overall worst case behavior. For the identification of "'bad"' influencing quantities on the system behavior and the tracing of them back to their sources, the range based system response has to be decomposed to the contributing sub ranges. This is where semi-symbolic simulations unfold their power. The symbolic nature of partial deviations allow the backtracking of single deviations to their source. Additionally it allows a decomposition of the Affine Form (AAF) of a quantity for the identification of the worst affecting partial deviation on the range based system quantities. Following this possibility the impact a single deviation has on the system may be estimated and the most influencing/interfering deviation parameter can be identified for possible improvement. Consider a simple system with low pass characteristic which is stimulated by a simple step function. The step response is modeled with two deviations which results in a range based signal.

$$\begin{aligned} \tilde{z} &= 0 * \tilde{x} + \tilde{y} && t < 30\mu s \\ \tilde{z} &= 1.3 * \tilde{x} + \tilde{y} && t = 30\mu s \\ \tilde{z} &= 1 * \tilde{x} + \tilde{y} && t > 30\mu s \\ & && \tilde{x} = 1.0 + 0.03\epsilon_1 \\ & && \tilde{y} = 0.0 + 0.04\epsilon_2 \end{aligned} \tag{4.8}$$

Equation 4.8 gives the formulation of the range based step response with a multiplicative and an additive deviation component. Note the overshoot component at time stamp *30 μs*. This range signal is simulated in a semi-symbolic simulation run by filtering the signal with a low pass filter. The filtered step response is dumped into a trace file during simulation to support an offline signal analysis and illustration.

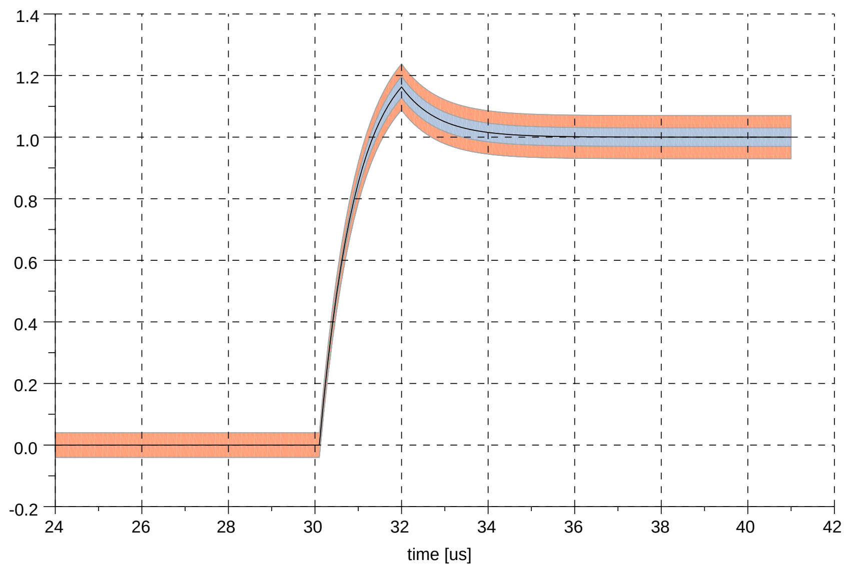

Figure 4.11: Range based step response with overshoot

Figure 4.11 shows the system result obtained by the semi-symbolic simulation run. For a better visibility the two distinct partial deviations are colored in differing colors. The "'blue"' area represents the multiplicative deviation $\tilde{x}$ and the "'salmon"' displays the additive deviation $\tilde{y}$. Both deviations have approximately the same impact on the step response in the saturated region and near the transition point. Before 30 μs in simulation time at the zero level of the step function, just the additive deviation parameter has an influence on the step response. If a design constraint would be to have accurate signal characteristic at small signal values then deviation $\tilde{y}$ would represent the perfect candidate for a design refinement.

4.2.1 Back tracking in comparison to forward tracking

The increasing demand for improving system simulations from pure numerical simulations with its drawback of loosing the dependency relation and impact of system quantities on the numerical output signal, resulted in the combination of numerical with symbolic techniques. The symbolic identifiers introduce now the possibility to track quantities throughout the simulation process up to identifying their single contribution on the output signal. This tracking process can be implemented either as back tracking or forward tracking process. Whilst a back tracking method starts at the designers primary goal, the system simulation output a forward tracking technique goes the other way round and starts at a single deviation source and its partial deviation. For a back tracking process the range based output quantity has to be disassembled and back tracking candidates identified. The back tracking library implementation succeedingly provides the deviation sources of every sub range chosen for back tracking, whether they are "‘user deviations"’ or self-added "‘system deviations"’. The forward tracking method picks a partial deviation and determines in which range signals this particular deviation is used and by that to which range signals it contributes. [RSRG12] provides a more in depth consideration of the tracking processes and explains the chosen implementation in an Affine Arithmetic library.

4.2.2 Implicit sensitivity consideration

Contrary to pure numerical simulations semi-symbolic simulations do not dissolve the dependency between deviation terms and the range based simulation result. In addition to the numerical result the range based signal also holds partitions originated by the deviation source, described as sub range and labeled by a symbolic identifier. This semi-symbolic simulation partly solves the dependency problem (for the range modeled deviation terms) with keeping the simulation costs at a reasonable level. The symbolic nature of this simulation technique allows the reverse identification of contributions to range based quantities. Affine Forms (AAF) themselves hold the dependency/correlation information in an implicit form. A backward tracking of deviation symbols results in the identification of the impact a deviation source has on a certain output quantity. The forward tracking provides information about the sensitivity of output signals on a certain deviation and more important about the correlation of system quantities to a particular deviation factor.

- backward tracking → impact
- forward tracking → sensitivity, correlations

The correlation of different range based system quantities to single or even more deviations is an important factor in a sensitivity analysis of the investigated system. Cross correlations within systems are typically hard to detect and semi-symbolic simulations provide information about that without further costs.

$$\begin{aligned}\tilde{x} &= 2.0 + 0.1\epsilon_1 + 0.2\epsilon_2 + 0.3\epsilon_3 \\ \tilde{y} &= 3.0 + 0.7\epsilon_3 + 0.4\epsilon_4 \\ \tilde{z} &= \tilde{x} + \tilde{y} = 5.0 + 0.1\epsilon_1 + 0.2\epsilon_2 + 1.0\epsilon_3 + 0.4\epsilon_4\end{aligned} \tag{4.9}$$

Equation 4.9 illustrates a correlation of various range based quantities. Quantity $\tilde{x}$ and $\tilde{y}$ share a common deviation symbol that indicates a correlation in deviation term ϵ_3. The $\tilde{y}$ dependency on ϵ_3 is much more prominent as it contributes by a much higher factor. A correlation in user created range signals is mostly evident as the model designer had to chose the deviation terms and to model them into the system model. The correlations resulting from the system function and represented in results of operations are far less obvious. $\tilde{z}$ results from an addition of range signals $\tilde{x}$ and $\tilde{y}$. It obtains all deviation terms of the initial operands and evidently is correlated to them.

A sensitivity analysis is well supported by semi-symbolic simulations. The implicit sensitivity in the symbolic identifiers allows a fast, easy and well defined identification of sensitivities and correlations between system quantities at any simulation time. Certainly this analysis is restricted to the modeled deviations as the remaining system functionality is covered by the numerical simulation result.

The SystemC AMS simulation framework, which is used in the MARC design environment uses a model based approach and supports hierarchical model descriptions. Range based models on the other hand provide an implicit dependency information by the deviation symbols itself. Combining this two properties results in the powerful and easy to extend sensitivity analysis capability a semi-symbolic simulation provides. The system models are easy to extend, with no need to take extra care of correlations and their analysis. The added range based signals simply enter the system model and are manipulated by the model functionality. The new arising sensitivities and correlations are available instantaneously as the deviation symbols hold the information at any time.

4.2.3 Dissolving range quantities in single contributions

The "'MARC design framework"' relies on library implementations that support the use of range based signals for system simulations. The libraries implement mathematical operations for signal manipulation but also methods for debugging, deviation term identification and access to single deviations. Section 3.2.2 describes one of the used library implementations but does not address the disadvantages of the chosen implementation. [11] which bases on [5] shifts its main focus on simulation efficiency and the computation of the worst case quantity. Deviations are solely identified by their index in a *static vector*. *User Deviations* are not differently marked to *System Deviations* which results in a difficult identification of the deviations in a range quantity. All this drawbacks of the library resulted in a new implementation of the Affine Arithmetic library with a special focus on an easy and powerful visibility of the system models deviations. The implementation was done at Vienna University of Technology, Institute of Technology by Michael Rathmair. The differences and enhancements to the older libraries is described in [RSRG12] and especially implemented a new approximation function for non-affine operations as well as an advanced symbol identification and naming service. This advanced symbol tracking methods allow the MARC design environment an efficient system quantity analysis and consequently an easier deviation symbol tracking. As further enhancement it implements various debugging functionalities to support the analysis of simulated system models.

- old (conventional) AAF library → hard deviation identification
  efficient for worst case analysis
- new (symbol centered) AAF library → clear distinction between *User* and *System* deviation
  methods for symbol tracking (backward/forward)

The new library introduces an unambiguous identification of deviations on the costs of an increased overhead and hence simulation effort. The computational effort is about 10 times higher which is caused by the added identification structures to support deviation visibility. The huge advantage of having an explicit deviation tracking technique exceeds this drawback. The library is based on C++ and uses object orientated constructs to implement the library. Methods for tracking of deviations, the introduction of descriptions for each deviation and various dumping methods for debugging purposes extend the functionality. A cleanup method, which preserves the dependencies in the range based quantities helps in using the simulation framework for complex systems. Analysis methods that return dependencies and correlations of evaluated deviations finally allow an efficient and convenient analysis of the simulated system.

4.3 System quality metrics

In order to use a systematic refinement design flow the quality of systems under investigation has to be rated. A refinement decision has to be judged on a deterministic system analysis and evaluation. The rating of system qualities is a non trivial task as the unambiguous "'quality"' property for electronic systems does not exist. The quality always also depends on the application a system is designed for. For instance is a signal to noise ratio (SNR) of received signals an important property in communication systems. On the other hand is the signal magnitude in consumer circuits typically high, which reduces the importance of this particular metric in this field. As a matter of fact several metrics which indicate the quality of a simulated system are available. A very bold and simple metric is the mentioned signal to noise ratio (SNR). This evaluation factor is commonly used in the communication technology field and is a measure on how difficult it is to demodulate and decode transmitted messages without errors.
When analyzing system models by using semi-symbolic simulations, the question is at which point in the simulation the SNR metric will apply. Usually the SNR property is calculated in a statistically or symbolically calculated way where the signal is averaged over a time interval. Various related SNR variants (power ratios, voltage ratios, root mean square amplitude ratios, channel signal to noise ratio,...) exist which are applicable to rate the system.
In the application field of semi-symbolic simulations SNR figures can be used as efficient single rating criterion. Independent from the straightforward use of SNR criterions, the time variation of the SNR according to the momentary simulation quantities complicates the use. An averaged metric would allow a single rating criterion but hides potential wild shots. A permanent momentary SNR calculation would consider the fully system behavior but introduces a set of SNR values that violates the single decision metric paradigm. Single metrics are most often not convenient as it restricts the rating process too strong. A more convenient and practical way is to combine several metrics and introduce some kind of cost functions to scale the importance of single properties to the system quality rating. The semi-symbolic nature of the simulation also supports the refinement of several system properties at once. Thus, the analysis of the system quality is not restricted to one metric evaluation it also allows the identification of several interesting metrics. A succeeding refinement of the major contributing system quantities influencing these metrics

allows a deterministic improvement of the system guided by quality ratings. Following this multi property refinement, an efficient system refinement and rapid system quality improvement can be achieved.

4.3.1 Discussion of approaches

As mentioned in the previous section "'system quality"' is a vague definition. There exits quite a lot of techniques to assess the behavior of the system. Especially in simulation based environments many metric functions can be implemented without any cost. The SNR figure is one possibility of a quality metric, but also a signal to deviation ratio (SDR) would be a natural rating criterion especially in semi-symbolic simulations. Basically the criterion for a system quality determination can be divided into three groups.

- quality metric
- combination of metrics
- cost function

The simple quality metric translates the system quality into a single quantity. The metric can be obtained by assessing the system in different domains (time, frequency) or by applying various evaluation techniques (SNR, SDR, min ripple, ...). The common property is that the system behavior is translated into one metric number that is used to rate the system quality. The combination of metrics uses multiple system properties to rate the system. The single properties can be analyzed in correlation to each other and offers improved analysis capabilities for complex issues. Cost functions support the consideration of quality metrics impact factors on the overall system quality figure. Not all metrics are equally important for the system quality in the evaluated application domain. A cost function considers this difference and scales the single metrics according to their importance. The resulting cost metric efficiently rates the system quality according to a selection of chosen and scaled quality metrics.

4.3.2 Numeric quality metric

Numeric quality metrics translate a rating of a system quality into a numeric quantity. Obviously this goes along with a loss of dependencies but allows an easy and efficient system judgment. The quality metrics are modeled in simulation blocks which are added the range based simulation model for system evaluation. After the quality criterion is violated the behavior of the simulated system is analyzed and the influencing properties are identified. Deviations which most prominent cause negative impacts are marked as critical and subsequently chosen for refinements.

- Signal to noise ratio (SNR), Error vector magnitude (EVM), Bit error rate (BER)
- Signal to deviation ratio (SDR)
- constellation point deviation
- overshoot/ undershoot

- ground variations
- single deviation
- aggregated error/uncertainty/deviation

The common performance metrics SNR, EVM and BER of communication systems are a strait forward criterion to rate a communication system. The BER is the most logical quality metric as it specifies the bit errors of the received and decoded signal caused by the deviations, uncertainties and noise in the receiver stage of the communication system. The error vector magnitude specifies the deviation a received symbol has from its expected position. All three quality metrics are closely related and are state of the art means to evaluate the quality of a communication system.
In semi-symbolic simulations, deviations are added the simulation model and contribute to the simulation result as non-ideal effects. The signal to deviation ratio (SDR) evaluates this feature and gives a measure on which share has the deviation range on the overall signal quantity. This ratio is very important in semi-symbolic simulations as it also specifies the significance of the simulated quantity and the share/cover the deviation has on the quantity. A high ratio would indicate a problem in the visibility of the numerical value of the quantity, possibly caused by excessive over-approximations.
The constellation point deviation metric especially addresses the evaluation of communication systems. This metric can be used as criterion on how deviated single constellation points are from their intended position. This metric is similar to a scatter plot, typically used when measuring constellation points in communication systems but uses the possible deviation effects for a deterministic approach.
Overshoots and undershoots are common effects in analog circuits. The metric on over-/undershoots measures this effects and uses them for the rating of the analog system quality.
Ground variations specify possible ground bounce effects contributing to the system behavior. If the ground reference is shifted towards signal values, effects on every part of the circuit are possible. Immunity against such never avoidable circuit effects is eminent and can be tested and rated by this quality metric. The ground variation can be influenced by more than one deviation source as it summarizes as additive deviation effect around the zero level.
A single deviation is assessed if the causing source is of special importance to the system behavior. For instance a quantization operation introduces a quantization error which subsequently influences the quality in the following digital sub system. Special attention would be given on the impact this single deviation has on the system quality.
Aggregated deviations (error, uncertainty, deviation) are used to assess subclasses of the deviation effects. For instance all labeled uncertainties and the deviations derived from them could be used to determine the impact of this group of deviations on the overall system behavior. Certainly all sub group combinations are valid and applicable.

4.3.3 Temporal quality metric

Previously introduced quality metrics consider system behavior at an explicit point in time. The semi-symbolic simulation provides a time characteristic of the system quantities and behavior. To fully utilize the capabilities of quality metrics they must be enhanced to timed behavior. A minimum and maximum figure allows an easy and powerful expansion of the previously mentioned quality metrics. The check if the system behavior resides under or over a certain threshold would be easy to evaluate with such a metric.

- minimum and maximum value of SDR
- minimum and maximum ripple values
- ripple in frequency bands

The min/max metric in general allows a powerful analysis of the system behavior in respect of violated or satisfied properties. The combination of the numerical metrics with the temporal min/max assessment provides a rich class of evaluation metrics to be applicable on semi-symbolic simulations. The evaluation in the frequency domain also offers an analysis of the system behavior over a period in time. For instance the checking of the present ripple in certain frequency bands (passband, stopband,...) allows a convenient analysis capability which covers the temporal behavior of the analyzed system.

4.3.4 Frequency domain metrics

The analysis of systems purely in the time domain lacks a wide range of analyzes capabilities available in the complex frequency domain. Periodic events of quantities in the time domain translate to single spectral components in the frequency domain. Thus, a rating of periodic system behavior is by far more efficient in the frequency domain and for particular properties available solely in the frequency domain. The evaluation of spectral components in a system quantity is an important feature in this analysis domain. The introduction of a range based "'Discrete Fourier Transform"' allows the spectral analysis of semi-symbolic simulation quantities and fundamentally enhances the analysis possibilities.

- ripple in frequency bands
- inter symbol interference (ISI)
- inter channel interference (ICI)
- SDR in frequency domain
- signal to interference ratio (SIR)
- carrier suppression
- adjacent-carrier interference (ACI)

The analysis capabilities in the frequency domain include but are not restricted to the evaluation of the ripple in dedicated frequency bands. Filtering of system quantities is a common task in communication systems and typically introduces bands (passband, stopband, transition,...) with an associated ripple of the attenuation factor. The system deviations contribute to the filter functionality and therefor influence the ripple characteristic which summarizes to the possible attenuation deviation. The deviations from the intended frequency behavior is an important measure on the spectral behavior of the analyzed system.

The spectral behavior of communication systems is the focus the two analysis metrics "‘inter symbol interference"’ and "‘inter channel interference"’ have. The first determines how much adjacent symbols interfere with each other. The range based system quantities form also a range area around the spectral components of the system. Overlaps of range based symbol components would cause possible system distortions and are assessed by this method. The later analyzes if overlaps in channels occur and gives a metric rating this issue.
The signal to deviation metric in the frequency domain is quite similar to the same metric in the time domain. Others than in the time domain the spectral analysis evaluates periodic quantities and is able to rate system behavior not just for a single point in time but for a particular frequency. The remaining three frequency domain system analysis metrics focus on the influence interferer quantities have on the system behavior. The interferers are real interferers or remaining carrier components that influence the spectral image. The enumeration of frequency domain quality metrics is just a sample of available evaluation methods and can be expanded as desired.

4.3.5 Internal, overall/joint and combined metrics

A semi-symbolic simulation allows the efficient analysis of internal system quantities due to its symbolic enhancement. The result of the simulation is not just a numeric value it additionally provides symbolic sub ranges that specify the impact a deviation source has on this quantity. The symbolic nature of the simulation approach allows an implicit error/impact propagation throughout the simulation model. The assessment of single system quantities allows a detailed analysis granularity. On the other hand overall or joint range based system quantities combine the deviation effects added the system model at various stages to an overall range based system quantity. The analysis of this combined range allows a coarse grained analysis which covers all contributing deviations altogether.
The combination of different quality metrics is an important mean to improve the analysis capabilities of single quality metrics. In particular the construction of complex property descriptions which can be used as rating criterion provides rich advantages. Usually not just one system property defines the quality of a system, more often a combination of several properties constitute to a satisfying behavior of the evaluated system. The combination of quality metrics supports this fact and hence improves the system analysis and rating process to better match the system analysts demands.

4.3.6 Signal to deviation metric (SDR)

The signal to deviation ratio (SDR) is by far the most important rating metric for semi-symbolic simulations. As previously stated, deviation effects are regularly dependent on the actual signal characteristic as they appear as multiplicative or more complex deviation model. In contrast to SNR figures which relate the signal characteristic to an independent noise value, SDR ratios relate the nominal signal quantity to its corresponding deviation term. As the signal quantities vary over time in semi-symbolic simulations, also the SDR figures change over time. Following this fact the SDR metric is typically extended to the timed behavior analysis. The SDR value is tracked over time and the maximum/minimum value is kept as system quality metric. $SDR(t)_{max}$ specifies this particular quality metric for maximum allowed deviation ratio and rates the range based system quality by means of a single, powerful metric.

4.3.7 Sorted findings

The system quality metrics used to analyze and rate the behavior of the evaluated system provide important and meaningful statements on how the system behavior compares to a given objective. The system metrics provide a possibility to create deterministic analysis figures but also results in a set of rated properties. The amount of available information demands for an ordering of the analysis results according to their importance on the refinement/optimization goal.
A sorting of the quality metrics should allow an efficient analysis of the quality metrics and the associated refinement candidates. The system designer finally is able to pick the most promising deviation sources to be chosen for an refinement/improvement to most effectively and deterministicly improve the overall system quality.

4.4 MARC refinement design flow

One of the major problems in designing and evaluating electronic circuits is the analysis of system behavior and a guided identification of system parameters that would improve the overall system behavior when modified. Numerical simulations provide just restricted information about the reasons for unsatisfying system behavior. The simulation results indicate solely problematic behavior they do not provide further information on the causes of the poor performance. Semi-symbolic simulations address this drawback. The symbolic enhancement allows conclusions on critical system parameters and their influence on the system behavior. The simulation performance gain of semi-symbolic simulations in comparison to traditional multi-run simulations additionally promotes the use of them for deviated system quantities. The influence of deviated system quantities on the system quality is typically hard to determine. The symbolic back tracking capabilities of semi-symbolic approaches enables a rich variety of analysis techniques to evaluate deviated system models. The "'embedding"' of semi-symbolic simulations into the powerful SystemC AMS simulation framework supports the easy extension of the simulation framework and on the other hand allows an efficient simulation by utilizing the built in simulation structures. The specification of quality metrics for a deterministic rating of the simulated systems quality expends the evaluation capabilities of the presented methodology and introduces an important factor for a guided refinement design flow. All previously presented properties, enhancements and analysis methodologies provided within semi-symbolic simulations are combined into a system refinement design flow, named "'**MARC refinement design flow**"' hereafter. The particular properties of semi-symbolic simulations with their range based deviation consideration and the powerful back tracking capabilities are used to create a refinement design flow. This design flow not only relies on a heuristic simulation approaches but deterministically identifies refinement candidates as result of the system analysis. The refinement process will be implemented in a reiterated simulation process where the system quality is repeatedly evaluated and compared against a defined system quality metric.

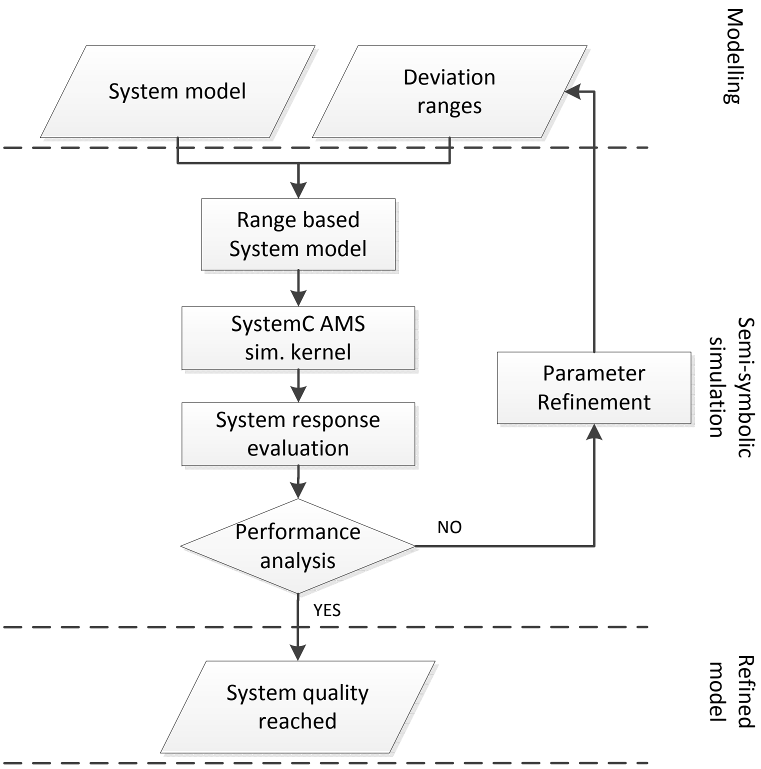

Figure 4.12: MARC refinement design flow

The "'MARC refinement design flow"' given in figure 4.12 summarizes the design steps proposed for a deterministic system quality improvement. All features that were presented and discussed in previous sections are used in implicit or explicit forms to simulate and refine the range based simulation model. An initial step in the design flow is the creation of the range based system model. In an intermediate step the system model is designed for its nominal system parameters using SystemC AMS. The main objective of the thesis refinement flow is to improve the system quality of deviation affected system models. Hence, the nominal system model is added a set of deviation terms that model parameter deviations in a range based way. The so created range based system model is used for a semi-symbolic simulation of the system model. The combined range based system model is integrated in the SystemC AMS simulation environment and is simulated using the "'*Timed Data Flow (TDF)*"' model of computation. The so obtained range based system response is analyzed for its performance/behavior by using predefined quality metrics to rate the system model. If the systems quality is considered as not sufficient an iterative refinement process is started. The system analysis results are disassembled to identify the worst influencing deviation effects and to back track them to their origin. In a refinement step the initial range based system model is updated in the identified refinement candidates and analyzed in the improved behavior in a further simulation run. The iterative refinement process is executed either until a maximum number of predefined iteration steps is reached or if the system quality is rated as sufficiently high. The use of explicit quality metrics facilitates the rating of the simulated models quality in a deterministic way. The combination of the refinement design flow with the symbolic capabilities of semi-symbolic simulations allows a well guided refinement candidate selection and an impact directed modification of the range model.

4.4.1 Iterative refinement process

The iterations in the refinement design flow allow a successively modification of the most important contributors to the range based system response. The primary refinement target is to improve the quality of systems under the influence of parameter deviations. Hence, the refinement candidates also are chosen from the set of deviation effects modeled in the range based system. The number of iterations depend primarily on the achieved system quality and as a second restriction on an upper bound of performed iteration steps. The convenient analysis capabilities of semi-symbolic simulations reduce the demand for a high number of simulation runs as it guides towards an efficient system refinement which significantly improves the system quality. The refinement of system parameters is not restricted to single deviations. The selection of refinement candidates is rather driven by the desire to control the modification impact on the system behavior.

4.4.2 Multiple analysis options

The analysis of system behavior is the most crucial step in the refinement design flow. Possible flaws and abnormalities that are not identified in the analysis section will not be tracked in subsequent stages and will never be scope for a system refinement. Hence, the potential system quality improvement strongly depends on the analysis methodologies available and used. The field of application defines the potential analysis methodology which is best to be used. Communication systems for instance typically require an evaluation of the spectral properties. Following this demand for multiple analysis options a selection of available techniques have been implemented. Semi-symbolic simulations have been extended especially in analysis in recent years [SKG+11, GOGB07, OSG11a] to broaden the available analysis methodologies. A major extension was the introduction of a range based "'Discrete Fourier Transform"'. Especially in the scope of communication systems the spectral analysis is required and allows a wide range of additional methodologies to be used.

The analysis methodology is not restricted to already available techniques. The system designer and verification engineer is free to create additional analysis options. The C++ based simulation environment supports the extension of the simulation framework. New methodologies are easy to integrate and even overloaded methods are free to be used. A selection of already implemented analysis functions is collected in a C++ based add-on library to the simulation framework. The support functions include ("'Discrete Fourier Analysis"', AAF tracing capabilities, AAF "'.csv"' dumping functionality, constellation diagram, and various illustration functions). The tracing or dumping of selected range based system quantities allows an offline quantity evaluation in abstracted analysis environments. MATLAB is frequently used in this work to manipulate, analyze and display range based system quantities. However, the choice of analysis is not restricted whatsoever. The transfer of the system quantities is performed via "'file transfers"'. The most common "'*.csv"' file format simply holds the system quantity at certain time points as comma separated list. The abstract analysis tool imports the vector and uses it for a following analysis. Building blocks for SystemC AMS are additionally implemented and added the simulation framework to simplify the modeling and analysis process. Various repeatedly occurring system functions (Laplace transform, filtering, sine wave jitter,...) are implemented in a reusable add-on library which provides building blocks to simplify the modeling process for the range based system model.

4.4.3 Discussion of quality rating using analysis methods

The analysis methods available for semi-symbolic simulations offer just one part to assess the quality of a system. The quality metrics used for the deterministic evaluation process are derived from the analysis results. The deviated system is modeled in a range based system model and simulated in a semi-symbolic approach. The obtained range based system response is used as basis for an analysis and evaluation of the system quality. This is achieved by a translation from the system behavior to a rating figure that classifies the system characteristics. The system quantity used for the evaluation of the system quality is not pre specified. The overall system response provides essential information about the performance of the simulated model. Further more individual quantities inside the system model provide viable statements on the system properties. The quality rating is deduced from single quality metrics or a combination of various metrics. A clear translation from the system quantities to metric figures allow an accurately defined and repeatable rating of the system. Cost functions introduce a significant extension in behavior evaluation. The system properties are not just evaluated by their characteristic, the rating process is further more extended also to the influence and cost every deviation term has on the overall system. The refinement process is advanced to not just considering impacts of deviations but also the costs of a refinement has on the system costs. Generally, the difficulty the refinement of a particular deviation parameter causes the system designer is added the reasoning about refinement candidates. Cost functions allow an efficient union of different quality metrics. Cost functions deterministically describe the transformation of single quality metrics to one or a set of rating figures.

4.4.4 Deviation refinement step

The basis of the refinement loop is the deviation refinement step. After the analysis of the system behavior and the evaluation using the quality metrics, indicated problematic system behavior is identified and the corresponding deviation term marked for refinement. The semi-symbolic simulation allows a decomposition of simulated system quantities and by that supports the identification of problematic properties. One or more refinement candidates are usually identified and modified following the cost function and the designers appraisal. Every refinement iteration requires a complete range based simulation run. The simulation runs contain all the effects the deviated parameters have on the system behavior. A modification of single or more deviation terms require a complete new simulation and calculation of the system model. The refinement in the current work is restricted to deviation terms. The nominal system behavior is considered as functional adequate, the refinement and optimization is concentrated on the system parameter deviations and their influence on the overall system behavior.

The refinement of the identified parameters is conducted in the range based system model. The single deviation terms are modified/updated in their range scaling and subsequently used for the following system simulation. Either the number of iteration steps or a reached system quality terminates the refinement process. The achieved increase in system quality will hopefully end the iteration loop. A just gradually increase in system behavior introduces the threat of a large number of necessary simulation runs to reach a certain system quality. For this reason the number of iteration loops is restricted to a maximum number. The refinement is usually associated with an increase in component costs, thus the increase in system quality is acquired by an increase in system costs. A trade off between quality and costs have to be found. The application usually specifies the refinement constraints and goals (minimum cost, highest quality, balanced quality,...).

4.4.5 Grouping of deviation symbols

The range based system response gives the influence every deviation has on the system behavior. The deviation terms are consequently ordered according to their impact on the system response and the deviation terms are sorted or grouped according to the system optimization criterion. A simple ordering over their numerical impact is not viable. The impact the single deviations have on the chosen quality metric (the "'target of optimization"') is the important factor that specifies the refinement order. If a particular system variation has to be minimized, deviations contributing to the chosen quantity must be improved.
Basically the deviation whose subrange forms the largest area within the worst case range is chosen as highest contributor. The ordering of the deviation symbols specifies the most valuable refinement candidates. A simple cost-benefit analysis helps to identify the actual refinement targets. Other than indicated by the ordering not just one deviation term is selectable for a refinement during one iteration step. A set of refinement candidates is usually chosen for improvement as the range based system quantities deterministically identify a set of potential deviation sources that influence the system behavior in a not desired way. The selection of the refinement candidates is more driven by the previously mentioned cost function than by a simple ordering of the contributors. Anyway, a grouping of deviation symbols according to their impact on the overall system behavior strongly helps in illustrating of potential refinement candidates within the range based system quantities.

4.4.6 Defining "'worst"' behavior

The worst case range in the field of semi-symbolic simulations is always a superposition of the nominal system behavior with the set of sub ranges caused by the modeled deviation effects. The worst case is not specified by a corner case summation of the system parameters but by a union of the sub ranges contributing to the overall worst case range. The pessimistic calculation of the range area guarantees the inclusion of all possible signal values in the overall range. This introduces the thread of severe over-approximations but allows a formal assessment of the system behavior which outperforms the drawbacks of property masking. The formal character of semi-symbolic simulations introduce a new class of analysis methodologies to be applicable on the simulation result as it allows a formal reasoning about the behavior of the system with respect to the modeled deviations. Deviations of system quantities cause (for every combination of modeled deviation) simulation results that are not specified in their actual value but that are specified in the area they lie in. This area of potential signal characteristics guarantees an inclusion of all possible resulting signals and defines the worst case range of the simulated system.
As previously specified, the worst case behavior is not necessary the worst case range of the system. The worst behavior can also be a combination of several not so "'bad"' effects which reside inside the worst case range. The worst behavior is more important specified by the field of application and optimization objective and not just by the absolute value of the worst case range. Based on the optimization goal the system behavior is analyzed for its range based quantities. The overall range quantity causing the worst behavior is identified and disassembled to obtain its contributing sub ranges. By this repeated segmentation of the causing effects the range and its contributing deviation sources are determined. By this process of in depth worst behavior analysis the "'worst"' contributing deviations are identified and marked for a potential refinement iteration.

4.4.7 Identified refinement objects

The basis of the identification of refinement objects is the disassembling of the range based system response into its involved sub ranges. The single sub ranges are individually evaluated for their influence on the system response. For the set of potential refinement candidates the sub ranges are back tracked to their original deviation origins. The original system parameters indicate the refinement costs associated with an improvement of this deviation. For instance a problematic noise floor would indicate an improvement in ADC conversion precision. An increase in ADC resolution results in an disproportional increase in converter costs and thus increase in overall system costs. Hence, the decision to refine an object should carefully be taken and always influences the system constraints, most often the costs of the system to be manufactured.
The identification process comprises of all the previously defined behavior evaluation methodologies. The grouping of disassembled deviation terms, the identifying of "'worst behavior"', the assessing of the impact of individual deviation terms on the analyzed system quantity and the influence on abstract quality metrics is considered to identify valuable refinement candidates. Cost functions additionally introduce a measure of the costs a specific improvement of a deviation parameter causes for the refinement. The identification of refinement candidates considers all the presented methodologies and individually selects the most appropriate sets of parameters.

4.4.8 Manual optimization strategies

The optimization process is realized manually in the refinement design flow. The refinement loop branches at the "'*performance analysis*"' block in the design flow. The decision whether a specific quality metric is satisfied is met at this point and if not the branch to the parameter refinement is entered. Especially this part of the refinement design flow is realized as manual step. The quality metric, dependent on the field of application and the improvement criterion is chosen by the system designer. This choosing of quality metric is experience guided by the designer and reflects the considered most crucial design parameter. Also the actual evaluation of the system quality following the chosen metric is a manual process. The semi-symbolic simulation framework helps in analyzing the system behavior by supporting analysis options but does not rate the system quality automatically.
The system analysis in particular is a very empirical process. The designer starts with an initial analysis and shifts the scope if necessary to problematic system behavior. The quality metric specifies a measure to rate the system performance. The analysis of the system behavior on the other hand tries to identify the reasons for a system behavior not meeting the quality standards. The scope of the analysis process is to identify system quantities that cause erratic or problematic system behavior in general.
The system response disassembly and parameter identification is also achieved manually. The modification and hence improvement of the refinement candidates finally is executed individually in the initial or latest valid range based system model. The chosen refinement deviations are improved by simply updating the respective deviation term in the original Affine Form and using this improved values for the next simulation iteration.
The manual optimization process is suboptimal but efficient enough as the symbolic guidance to refinement candidates allows a significant system quality improvement in just a few simulation iterations. The manual operator is typically the system designer which assures a certain efficiency in the choosing of quality metrics, adopting the system analysis techniques to identify worst behavior and to perform a guided refinement of the most promising deviation terms.

4.5 Optimization and refinement process

The overall objective of the presented design flow is an optimization, or more detailed a refinement of the initial system model. The optimization aims in improving the robustness and system quality of the verified system. The field of application that were chosen to demonstrate the introduced design flow are "‘communication systems"’. The optimization does not follow a formal methodology. It rather uses a system simulation to reason about the system quality and utilizes the symbolic nature of the underlying semi-symbolic simulation to identify refinement parameters which influence the robustness of the system. The refinement of system parameters usually is associated with certain costs. A special case would be where the analysis of the system behavior would indicate that certain system parameters do not influence the intended behavior at all. In this special case this parameter could be marked as "‘don't care"’ where the parameter could also be relaxed. However, the refinement of system parameters usually are associated with costs. Whether they are real monetary costs or just indications on the importance of the factor on the system quality. This costs are specified by implicit cost functions which are used in the design refinement process. These cost functions guide to a deterministic refinement of system properties and parameters. As a consequence these cost functions lead to a deterministic identification of refinement candidates in the system parameters.
A semi-automatic design refinement have already been achieved in the proposed framework. The semi-symbolic simulation provides analysis capabilities and sensitivity information without any extra costs. The simulation environment is capable of complex analysis methodologies, an efficient simulation engine and of advanced visualization options. The dependencies of deviation effects are kept during the system simulation in an implicit form and allows the consideration of sensitivities during the process of refinement candidate identification. A transition from the semi-automated approach to a fully automated design flow is still missing but is considered as important extension of the proposed framework and design refinement flow.

4.5.1 Optimization versus refinement

Optimization in this context is considered as "‘automated"’ process which modifies the system parameters with the objective to improve the system quality. The optimization process must follow a specific quality criterion and uses automatic techniques to update the system parameters and reason about the influence on the quality criterion. Refinement on the other hand indicates that just a few modification and iteration steps are used to "‘improve"’ the system quality. This process is a semi-automatic approach where parts of the refinement process are tool supported but the important system assessment and modification steps are executed manually. The heterogeneous nature of electronic systems complicate the automated refinement process as system analysis and the selection of refinement candidates is not straightforward. The differences in common quality criterions hamper a unified optimization design flow.
Quality metrics lead to an automated design flow as they allow a deterministic rating of system quality. Optimization is much more powerful than a simple refinement but lacks a methodology for range based systems and semi-symbolic simulations. From a classification perspective, optimization is on top of the refinement design flow. It uses the same design steps but automates the analysis and refinement step and allows an automated re-iteration until a sufficient system quality is reached or the improve in system quality falls under a pre-specified threshold. Optimization uses the same design steps but automizes the deviation identification, refinement candidate selection and modification of deviation effects

4.5.2 Automatic optimization strategy

An automatic optimization approach consists of two major obstacles that have to be solved. Firstly, the analysis process has to be automated. The introduction of quality metrics facilitates a deterministic system validation and by that the integration of the analysis process into an automated design flow. The use of quality metrics does not solve the issue of automatic analysis results but supports techniques for the assessment of systems.

- simulated annealing
- evolutionary algorithms for parameter optimization
- genetic algorithms
- particle swarm optimization
- gradient descent

The second and in this scope more important unsolved issue is the optimization process itself. The system parameters (deviated in their value/ source of variation) are the objective of the optimization process. The parameters should be automatically refined/modified to optimize the system quality to improve the robustness of the implemented system or more generally to increase the system quality.
Many optimization techniques exist with several dedicated especially for parameter optimization. To name and discuss a couple of the available techniques the enumeration lists a selection. Simulated annealing addresses the optimization problem by localizing a global optimum for a given optimization function in a large search space. The parameter deviations span a large design space which form the search space of this optimization approach. Evolutionary optimization algorithms and genetical algorithms use evolutionary strategies to optimize a given parameter space and find an optimal system implementation. It can be classified as metaheuristic optimization technique and perfectly integrates into the proposed, iterative refinement design flow. A particle swarm optimization even is more closely related to the proposed design flow. It iteratively improves the properties of a given optimization candidate by measuring the quality of the candidate. A general optimization methodology which allows the identification of local minima/maxima is the gradient descent. Here the gradient of the optimization function is followed to approach the local minimas/maximas of the function.

5 Case Studies, Results

The previous chapters specified the process of design refinement and the use of semi-symbolic simulations for a deterministic refinement candidate selection. This chapter uses all of the presented techniques and proves them on demonstration examples to show the applicability of the proposed methodology. The field of possible demonstration examples is vast. This thesis concentrates on the refinement of communication systems and hence the demonstration examples will also be chosen from this particular field of application. The field of analysis methods that are used for the evaluation of the modeled systems and their behavior is also wide. This work uses a selection of available methodologies, from a time domain signal to deviation (SDR) metric to a frequency domain analysis to utilize also one of the extended analysis techniques previously presented.
The simulated examples are kept at a basic level to show the motivation for a design refinement flow and to demonstrate the convenient handling of a deterministic refinement approach. The examples are all modeled and simulated in the MARC design environment. This means that SystemC AMS is used to model and simulate the system. As model of computation (MoC) the timed data flow (TFD) is chosen and the Affine Arithmetic library mentioned in section 4.2 is used to compute the system quantities. The library implementation is based on C++ and integrates mathematical operators on range based signals as well as advanced tracing and tracking functionalities important for the use in the refinement design flow. External computation and visualization tools are used to support the quality metric determination and to increase the visibility of the deviation effects on the system behavior. For exchanging simulation quantities between the MARC design environment embedded in the SystemC AMS framework and the external tools tracing modules write (dump) range based system quantities into files for a subsequent offline signal processing.
The refinement iteration steps are highlighted individually. The single refinement operations show the idea of a deterministic system improvement and will be recapitulated in a common table at the end of each demonstration example.

5.1 Refinement targeting SDR quality metrics

As previously mentioned, the SDR metric is the most important quality criterion for range based systems. It specifies the relation between the nominal system quantity and its associated deviation range. It can be considered as measure for the significance of the simulation result, as a high proportion of the deviation on the range quantity indicates a strong abstraction of system properties on a deviation area. A high deviation of system parameters also indicates a high variability of system behavior and thus a wide range of possibly problematic system behaviors originating from the ideal realization. A refinement of the SDR metric improves the variance of the available realizations in general but also improves the robustness of the system by reducing the amount of distorting effects on the system behavior.

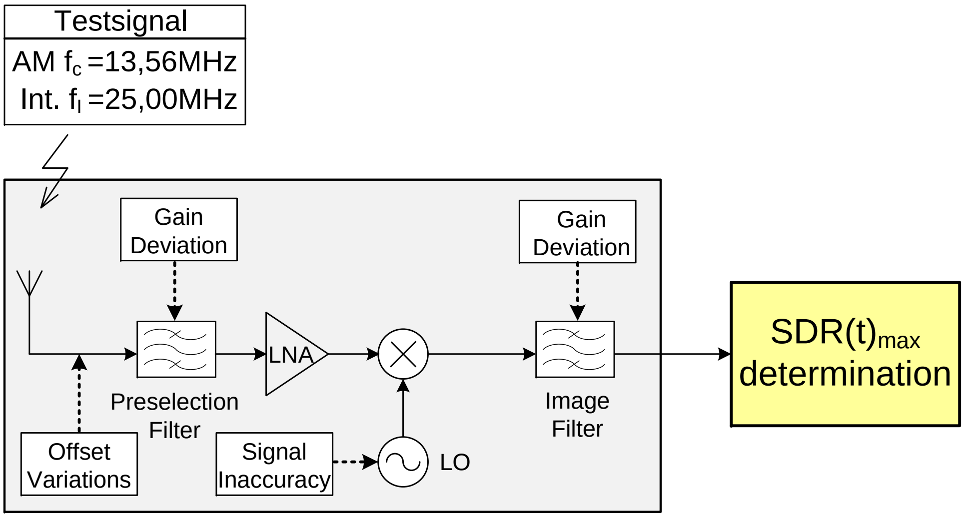

Figure 5.1: Receiver structure SDR refinement

Figure 5.1 shows a demonstration example for a complete semi-symbolic simulation of an amplitude modulation (AM) receiver structure with an attached system analysis unit. The system comprises of a range based system model with four deviations, representing sources which are common to the example structure. A very basic "‘offset variation"’ models deviations causing the ideal ground level to shift. "‘Gain deviations"’ of the two implemented filter structures consider non-ideal gain factors of the used filtering blocks. A "‘signal inaccuracy"’ finally models imperfections in the signal generation of the local oscillators’s (LO) sine wave. This inaccuracy is modeled using a multiplicative deviation effect.
The system model is sourced by an AM test signal which consists of a single sideband signal on a 13,56 MHz carrier, distorted by a 25 MHz interferer signal irradiating on the air interface. The test signal will be received and converted by the deviated system model. The deviations on parameters of the single stages will add deviation effects on the receiving signal and results in a range based signal representation.
The receiving signal is firstly filtered to attenuate the interferer signal in a pre-selection filtering stage. The interferer signal is an unwanted spectral component which distorts the receiving signal and is if possible removed by the input stage of the receiving system. The filtered and at this point already range based receiving signal is following mixed by a 13,56 MHz LO signal, resulting in the baseband representation of the original modulated signal and interfering higher frequency mirror images. The local oscillators sine wave component is generated introducing signal inaccuracies. This inaccuracy also contribute to the range based mixer result which results in an uncorrelated set of deviation effects superimposing to the overall range signal. The at the mixing operation emerging mirror images are removed by a typical medium quality filter. The filter characteristic is non-critical as the spectral distance between the image components and the desired baseband signal is large.
SystemC AMS is used as modeling and simulation framework and an Affine Arithmetic library overloads the required mathematical operations throughout the simulation to compute the range

based receiving signal. The SystemC AMS simulation engine calculates the simulation results using a "'Timed Data Flow (TDF)"' model of computation and dumps signals of interest into an either SystemC AMS supported *VCD* file or user defined proprietary trace files. The available trace files are imported into the external mathematical environment MATLAB which is used to calculate the single SDR figures and its maximum value.
For a performance analysis operation a simple SDR metric has been chosen. Range based system quantities obviously change their value during the simulation process. As the values of the range based quantities change, also the relation between the nominal and the deviation value changes in time. As a result the SDR metric is analyzed not just in its absolute value but also by its timed behavior. This approach allows to evaluate the system quality by straightforward deciding on a single numerical number. The SDR is calculated on the boundary from the analog to the digital domain. In the actual system realization the signal quality at this point is an important measure as the remaining digital sub-system will usually not add additional noise contributions to the signal. Thus, evaluating the signal quality at this point directly determines the quality of the RF front end itself.
The SDR metric is defined as:

$$SDR(t) = 20 * log\left(\frac{A_{nominal}(t)}{A_{deviation}(t)}\right) \qquad [dB] \tag{5.1}$$

where $A_{nominal}(t)$ is the time dependent nominal value of the receiving signal and $A_{deviation}(t)$ represents the corresponding timed deviation or when treated in a communication sense, uncertain distortion term. The result of the SDR(t) metric calculation gives a variation of the metric in time. The system quality decision process expects a single numerical value for its operation, thus the maximum value of the SDR(t) metric is defined as basis for the rating. As the performance metric is measured as SDR the peak SDR value of the demonstrating example receiver signal is calculated in a post simulation process. Based on this SDR figure the quality of the RF sub system is analyzed and refinement decisions are met accordingly, when necessary.

$x_1\epsilon_1$	$x_2\epsilon_2$	$x_3\epsilon_3$	$x_4\epsilon_4$	$SDR(t)_{max}$ dB
0.05	0.015	0.037	0.002	47.43

Table 5.1: Initial system simulation

Table 5.1 lists the internal quantities and corresponding SDR metric of the system refinement process. The range based model is simulated with initially predefined parameter deviation values, estimating a barely sufficient system quality. Quality typically manifest in increased system costs what explains the desire to design systems at the lowest acceptable quality. As previously mentioned four system deviations are included in the range based model. Deviation 1 which is denoted by deviation term $x_1\epsilon_1$ defines the additive offset variation and is initially set to a value of 0.05 which corresponds to about 2.5 % of the signal amplitude. Correspondingly, represented by its deviation term deviation 2 specifies the variations in filter gain of the communication receiver pre-selection filter. Deviation 3 models the local oscillator signal inaccuracy and deviation 4 finally describes the filter gain variation of the image filter structure. The initial semi-symbolic simulation results in a receiving signal to deviation ratio of 47.43 dB maximum. This SDR value is considered to be too low, therefore a refinement process is performed.

$x_1\epsilon_1$	$x_2\epsilon_2$	$x_3\epsilon_3$	$x_4\epsilon_4$	$SDR(t)_{max}$ dB
0.05	0.015	0.017	0.002	54.10

Table 5.2: First iteration in the system refinement

The dominant deviation term is identified by the system analysis and refined to increase the overall system quality. In a first iteration the LO signal inaccuracy is improved to reduce the sine wave signal deviations. The inaccuracy associated deviation term $x_3\epsilon_3$ is reduced to a variation value of 0.017. This almost halves the variability and in particular increases the exactness of the mixing operation. This system improvement results in a considerable increase of the SDR metric to a value of 54.10 dB.
The refinement objective of the current example is the robustness of the system. The quality metric which reflects this refinement objective is the signal to deviation SDR metric. The less influence parameter deviations have on the system behavior the more tolerant is the system on potential distortions. Hence, a minimum SDR ratio of 58 dB has been specified for the system to achieve at the analog-digital edge. As the demanded minimum SDR value is set to 58 dB two additional refinement iterations are necessary to perform.

Iteration	$x_1\epsilon_1$	$x_2\epsilon_2$	$x_3\epsilon_3$	$x_4\epsilon_4$	$SDR(t)_{max}$ dB
1	0.05	0.015	0.037	0.002	47.43
2	0.05	0.015	0.017	0.002	54.10
3	0.05	0.010	0.017	0.002	56.19
4	0.035	0.010	0.017	0.002	58.74

Table 5.3: Refinement progress

The first one improves the gain deviation of the pre-selection filter and the second one reduces the offset variation. Finally, an SDR higher as the pre-specified 58 dB is found and the corresponding system realization satisfies the specification and is ready for implementation. Table 5.3 lists the full refinement procedure with the initial system deviations and the corresponding SDR(t) figure. In total a number of four semi-symbolic simulation runs have to be executed. The first one gives the basic system behavior and its range based quantities. The original deviation values cause a SDR value of 47.43 dB which is clearly below the requested quality metric of 58 dB. A subsequent refinement of deviation term $x_3\epsilon_3$ clearly improves the quality metric but still does not satisfy the specification. Two additional refinement steps finally increase the SDR metric over 58 dB and ends the refinement loop with a sufficient system quality.

5.2 Refinement of phase jitter properties

The first demonstration example implemented a communication receiver with deviations on selected components of the system and was refined to optimize the overall SDR metric of the system. The local oscillators inaccuracies were modeled by a multiplicative range and all deviations were assessed in their combination, by calculating the SDR metric. The second example introduces a phase jitter instead of the multiplicative inaccuracy model. A phase jitter deviation was introduced in section 3.3.5 and uses an additive deviation model to describe the phase jitter. This assumption is only valid if the phase jitter is considered as sufficient small in comparison to the full signal period. This phase jitter contributes directly to the mixer component and its resulting quantity via the local oscillator source. Inaccuracies and distortions in the local oscillators signal hence seriously contribute to the resulting base band signal and influence the signal characteristic both, in the time domain as well as in the frequency domain.

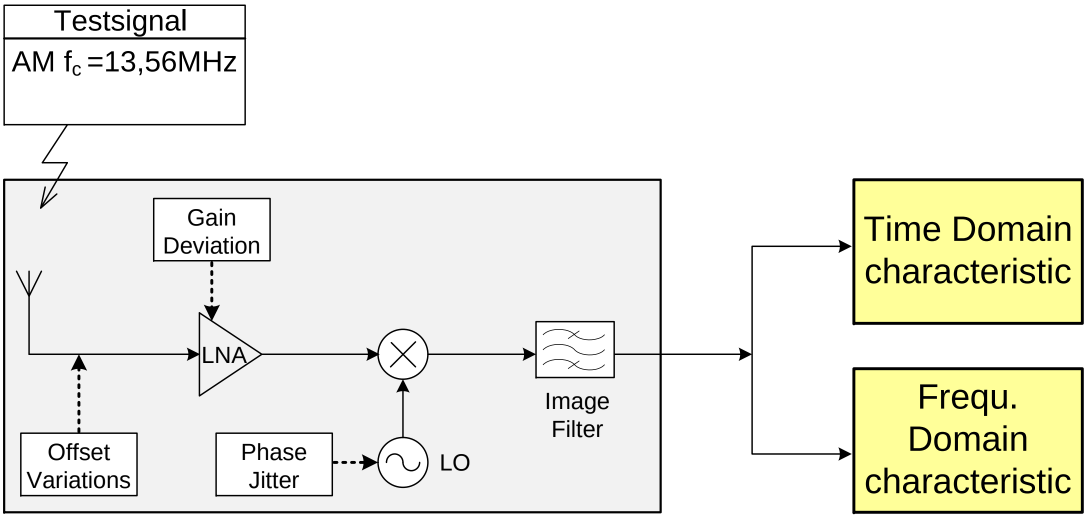

Figure 5.2: Receiver with phase jitter

Figure 5.2 shows the considered communication system with its contributing variations. The example is similar to the first implemented system, but in contrast neglects an interferer component on the receiving test signal and uses a different number and models of deviation effects on the system. The main objective in the given example is to analyze and refine the phase jitter contribution on the system behavior. Hence, the spectral efficiency and composition of the received signal is not as important as in the last example. As mentioned, the phase jitter model follows the assumptions and definitions presented in section 3.3.5. The system analysis step is divided into two sub steps:

- Analysis in the time domain
- Analysis in the frequency domain

In both analysis steps the received baseband signal quantity is decomposed to obtain the main contributors to the range quantity. The range symbolizes the potential signal variation and is intended to be kept small.
The receiver system is stimulated by a simple amplitude modulated test signal which constitutes of a simple 5 MHz baseband sine wave, modulated on a 13,56 MHz carrier available as single side band modulated AM signal. The received test signal is amplified by an LNA and mixed down into the baseband by a mixing block. The mixer is sourced by a 13,56 MHz local oscillator (LO) component and the resulting mixing product is adjusted by the following mirror image removing filter. The resulting baseband signal is analyzed for its range based components and if necessary a refinement step follows to improve the system behavior.

Table 5.4 summarizes the considered deviation effects and their initial ranges. Deviation term $x_1\epsilon_1$ models the additive phase jitter deviation. It contributes to the local oscillators sine wave

$x_1\epsilon_1$ jitter	$x_2\epsilon_2$ lna	$x_3\epsilon_3$ offset	$x_4\epsilon_4$ approx.1	$x_5\epsilon_5$ approx.2
0.1	0.03	0.05	x	x

Table 5.4: Initial phase jitter example

generation and adds a phase jitter to the created sine wave signal. The second deviation term is $x_2\epsilon_2$ which introduces a variation of the gain parameter on the low noise amplifier (LNA). The gain of the amplifier is considered to be not just one singular value, it is considered to be in a range of possible gain values and hence amplify the incoming signal also with all of the possible gain factors. The last modeled user deviation is $x_3\epsilon_3$. It covers offset variations on the ground line modeled as additive deviation with a central value of 0 for the range. The two additional deviation terms emerge during simulation. They both hold approximation related scaling factors and are initially undefined as they depend on the actual values of the range quantities which cause the approximation. The first approximation term originates at the LNA gain multiplication. The gain of the LNA is described as range based quantity. When the arriving range based receiving signal is multiplied by the also range based gain variation a multiplication of two range based quantities occurs. As the multiplication is a non-affine mathematical operation the resulting range quantity is derived by an approximation of the resulting Affine Form. The approximation causes an extra system deviation to be added the simulation model which assures that no higher order effects are missed when modeling the mathematical result by a first order approximation. The second system deviation appears at the mixing operation. Again the range based input signal is multiplied by the range based sine wave, generated in the local oscillator block.
The initial range quantities specify a first approach for system implementation with mild system parameter variations. Deviation $x_1\epsilon_1$, the phase jitter deviation is defined by a phase variation of 0.1 complying to the small signal approximation but still having a significant effect on the signal characteristic. The LNA gain deviation is specified as 0.03 and contributes with a 3 % variation to the gain quantity. The model offset deviation finally is specified by 0.05 adding a small deviation to the nominal ground level of the system.
The "‘MARC"’ design refinement flow defines the following operations

- Initial system simulation (range based)
- System analysis
- System modification

to find a robust and reliable system. The initial simulation obtains a first estimate about the system behavior and creates the range based system quantities that are analyzed to identify system refinement candidates. The system modification finally updates the refinement candidates and determines the improved system quantities in a recursive iteration. The second demonstration example implements two analysis techniques in the analysis process. The system result is analyzed for its time as well as frequency domain characteristic. The range quantities are decomposed and the influence of every deviation term is determined.

Figure 5.3 shows the composition of the received baseband signal in the time domain. The single contributors to the range signal are given in different colors where the phase jitter range present the major part and is given in "‘steelblue"’. It is recognizable that the phase jitters influence is the main deviation contributor to the range based receiver signal. All 5 deviation terms are

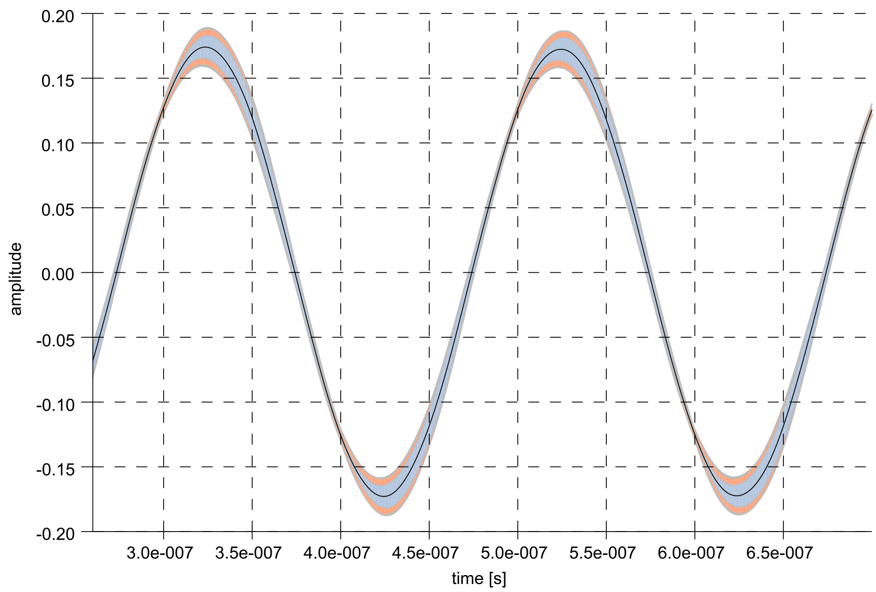

Figure 5.3: Time domain representation of jitter and remaining ranges

superimposed the nominal quantity and form a range area specifying the potential signal forms in their combination. The worst case bound is given by a summation of all ranges and is calculated by pessimistic approximations of the corresponding sub ranges. The overall deviation range summarize to about 15% of the signal amplitude which can be considered as significant variation of the received signal.

Figure 5.4 gives a detailed view on the single contributors to the output quantity. All 5 deviation terms combine to the overall range of the quantity. The single contributors are divided by their color and individually superimpose to the central value of the Affine Form, the nominal quantity. The solid line in the middle represents the central value of the Affine Form. The first deviation that superimposes the nominal value is the phase jitter variation. It is drawn in "‘steelblue"’ and intuitively is identifiable as main area of the range based signal. The second deviation that adds to the Affine Form is the LNA gain deviation. It is given in "‘salmon"’ and adds also a significant variation range to the output signal. The remaining three deviation effects that contribute to the range of the received signal contribute just with a very small share. The remaining user deviation, (modeling the ground offset) effects the output signal just slightly. The two system deviations considering the approximation factors also remain very small. Their influence on the output signal also can be considered as negligible.

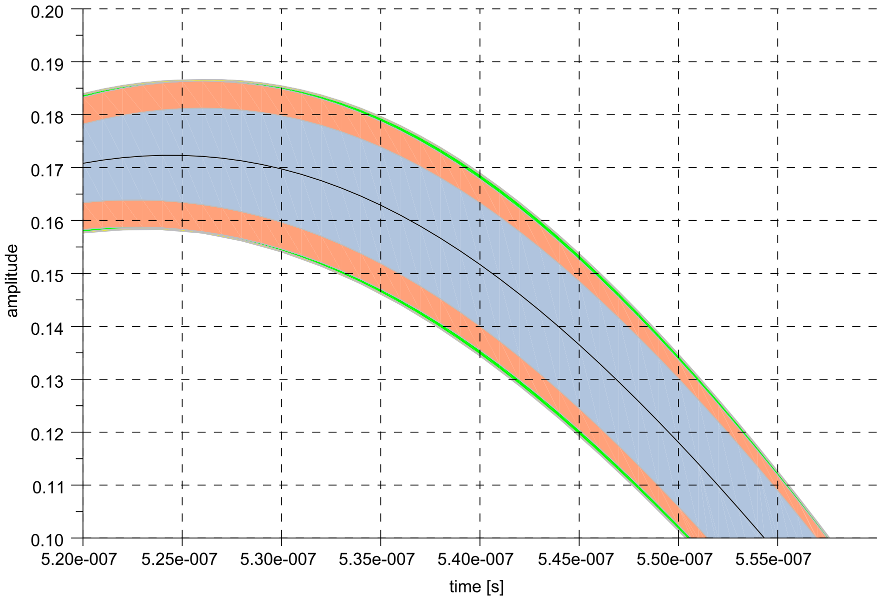

Figure 5.4: Detailed view on single deviation contributors

The analysis of the system behavior is not just restricted on the time domain. As given in figure 5.2 additional to the analysis in the time domain also an analysis step in the frequency domain is executed. A Discrete Fourier Transform (DFT) has been presented in section 3.3.2 and allows in this example the analysis of system quantities also in their spectral behavior. The DFT analysis is implemented in a module within the "'MARC"' framework and allows a very flexible calculation of a quantities spectral components during the simulation run. The frequency spectrum is stored as stream in a "'.csv"' file which is imported into the "'SCILAB"' environment for visualization and post processing.
Figure 5.5 shows a range based frequency spectrum of the systems received and processed baseband signal. The baseband signal is of range nature so the frequency spectrum also comprises of range based spectral components. The spectrum gives the frequency range from 0 Hz to 50 MHz where the DFT was computed for a sampling frequency of 100 MHz calculated on 512 DFT points. The simulation was conducted within the "'MARC"' framework and utilizes the functionality of SystemC AMS to simulate the system. The received signal is analyzed after passing through the systems processing stages and represents the baseband signal. The dominant spectral component is at 5 MHz which is the baseband frequency of the transmitted test signal. The mirror images generated in the mixing operation are attenuated well so that the test signal is the only relevant component in the spectrum. Contrary to traditional spectral components the test signals frequency representation comprises of a nominal mid value and a superimposed frequency representation of the contributing deviations. The deviation from the central value reach significant values. As already recognized after the analysis in the time domain, the variation in magnitude is a significant factor of the test signal.

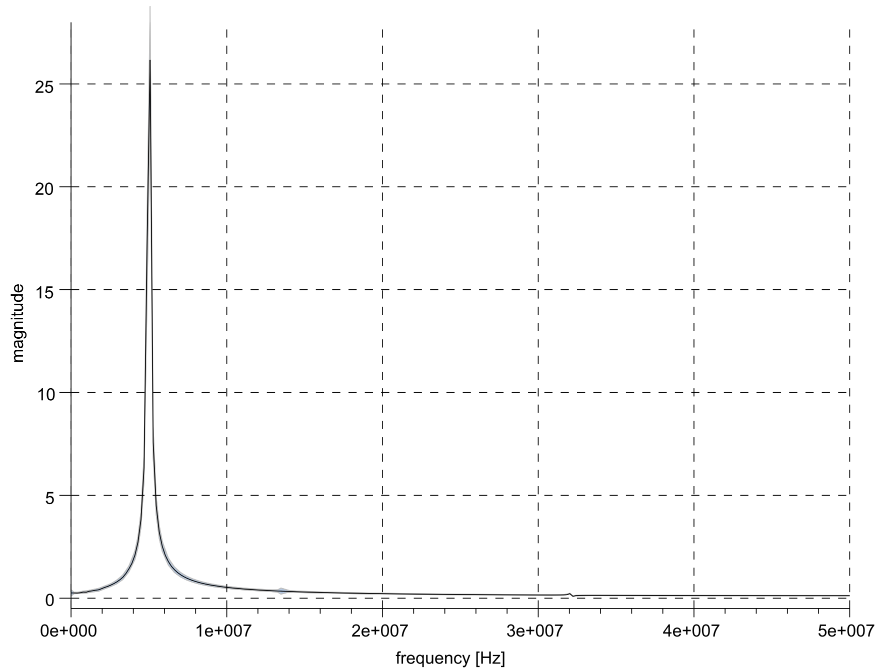

Figure 5.5: Range based Fourier spectra

Figure 5.6 shows a more detailed view on the systems spectral component. The signal is the transmitted 5 MHz test signal, superimposed by all subranges caused by deviation effects during signal processing in the modeled system. The range is given as worst case bound and colored in "‘steelblue"’. The range based system holds a total of 5 deviation effects and all of them contribute to the spectral component in different strengths. Again, the variation in magnitude reach significant levels and a refinement of the deviation parameters seems beneficial to improve the system quality. The worst case bound does not provide additional information about the impact the single deviations have on the spectral component. The Fourier spectrum has additionally be divided into its contributing sub ranges. The DFT module allows to trace single deviation terms to provide information about sub ranges of the range based quantity.

Figure 5.7 finally shows a detailed view of the range based spectral component. The range is divided into two sub ranges. The "‘steelblue"’ portion which represents the influence of the phase jitter deviation and the "‘salmon"’ colored remaining deviations which effect the system output. Obviously to see, the phase jitter deviation is the major influence on the spectral component in respect to magnitude variations. All the remaining deviation terms, whether user deviation or the systems approximation factors do not effect the spectral behavior comparable strong. In fact also the summarized deviation effects do not contribute to the system behavior as strong as the phase jitter variation. Interesting in the detailed view is that the deviations not only add a positive deviation to the magnitude of the analyzed quantity. The range based deviation adds an area of potential values above but also below the nominal quantity. This is modeled by the Affine Form which superimposes a range or interval, center symmetric around the central value.

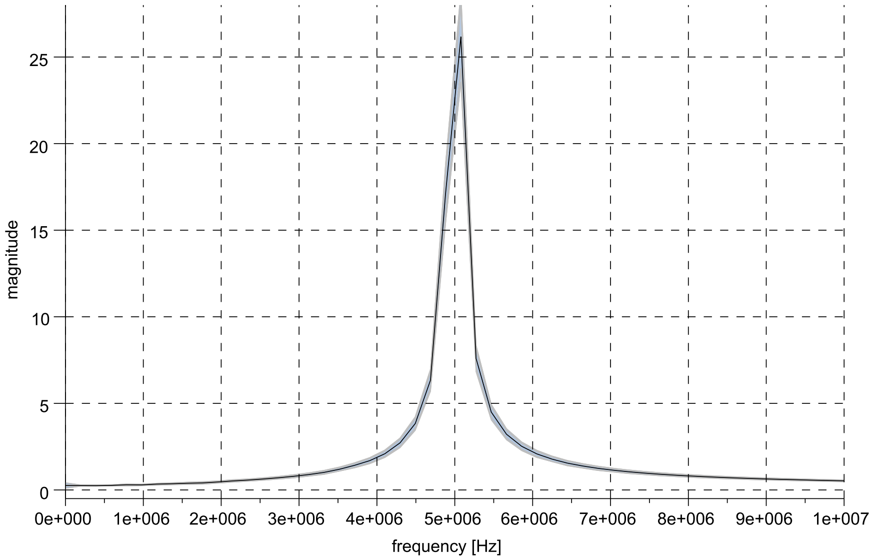

Figure 5.6: Range based spectral component

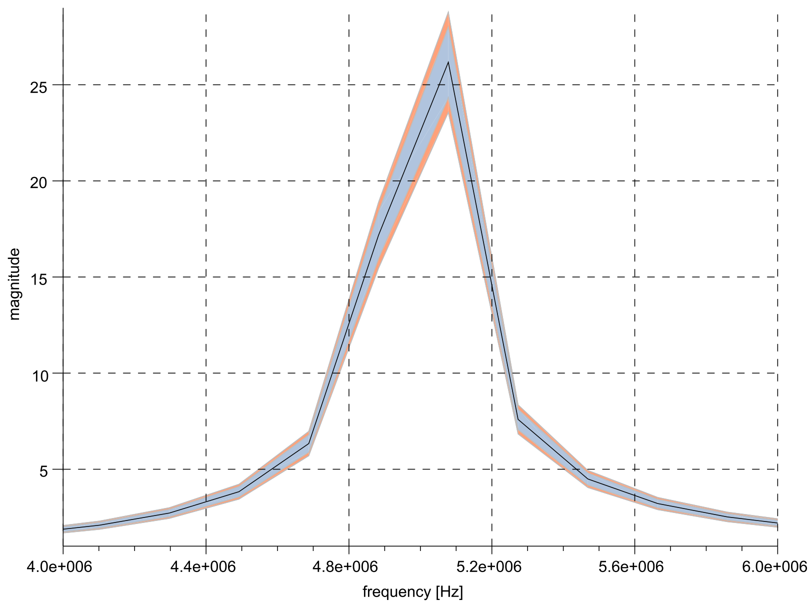

Figure 5.7: Contributors to spectral component

The analysis of the simulated range based system model shows a significant dependency of the system behavior from the local oscillators phase jitter. This dependency is evident in the time domain as well as when performing an analysis step in the frequency domain. The "'MARC refinement design flow"' specifies the block "'analysis"' in which the system performance is evaluated. The previous simulation result gives the initial simulation phase. In this phase the system should be analyzed for possible erratic or reliability influencing system properties. The analysis clearly shows a huge impact of the systems phase jitter deviation on the system behavior. This analysis result strongly indicates that the phase jitter should be refined/improved. This decision is met during the "'parameter refinement"'. The refinement of a parameter always influences the costs of a system. In this example the phase jitter could be reduced by using a higher quality oscillator component. The refinement of this component should not produce significant extra costs in the system design, so the refinement of this system parameter is chosen.

$x_1\epsilon_1$ jitter	$x_2\epsilon_2$ lna	$x_3\epsilon_3$ offset	$x_4\epsilon_4$ approx.1	$x_5\epsilon_5$ approx.2
0.01	0.03	0.05	x	x

Table 5.5: Phase jitter refinement

Table 5.5 gives an overview over the system deviations after a refinement step is applied on the phase jitter deviation. This deviation is reduced by a factor of 10 and has for the next iteration of the refinement process a value of $x_1\epsilon_1 = 0.01$. The remaining deviation terms are not selected for refinement as their influence on the system quantities is smaller and their improvements come with higher costs.

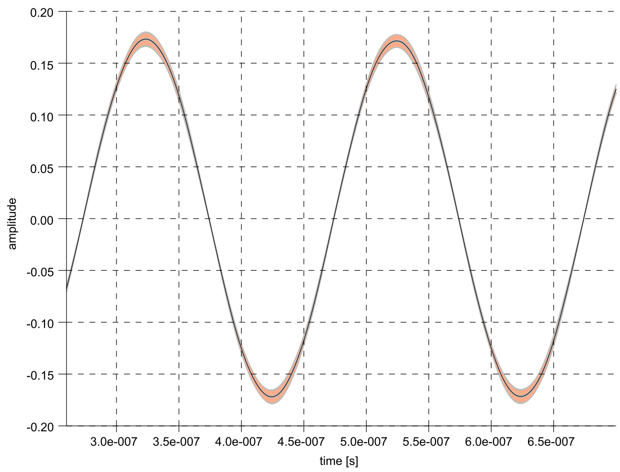

Figure 5.8: Refined phase jitter deviation

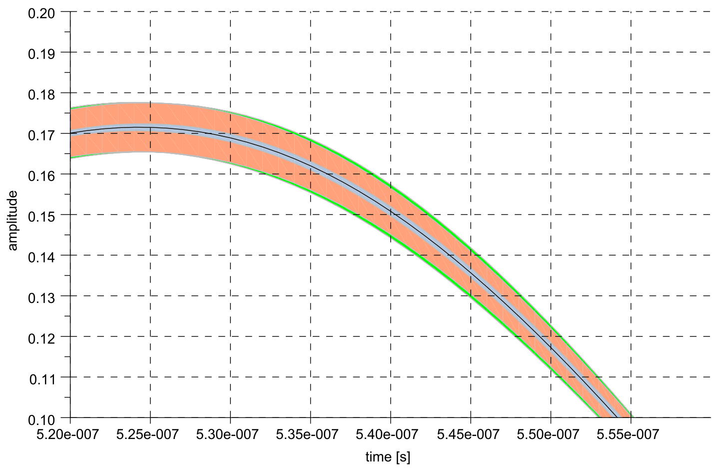

Figure 5.9: Subranges of the refined signal

Figure 5.8 and 5.9 show the time domain signal characteristic after the refinement iteration. The "‘steelblue"’ colored phase jitter deviation is apparently smaller and does not form the main impact factor any more. From the timed behavior the LNA deviation stays as next main contributor to the signal quantities variability and could be modified in a subsequent iteration if necessary. The overall variability of the received signal has reduced to about 6 % of the signal amplitude where the phase jitters influence has changed to almost insignificant.

Figure 5.10 and 5.11 show the refined signal characteristic in the frequency domain. The deviation of the spectral components from the nominal signal characteristic is significantly reduced. The behavior of the system can be estimated much more accurate as the potential variations are minimized.

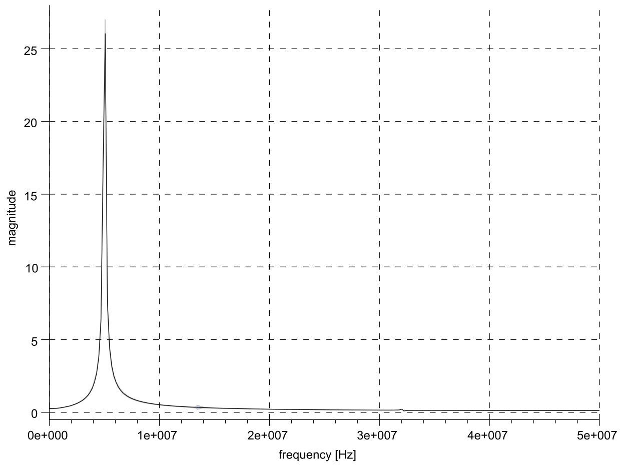

Figure 5.10: Spectra of the refined system behavior

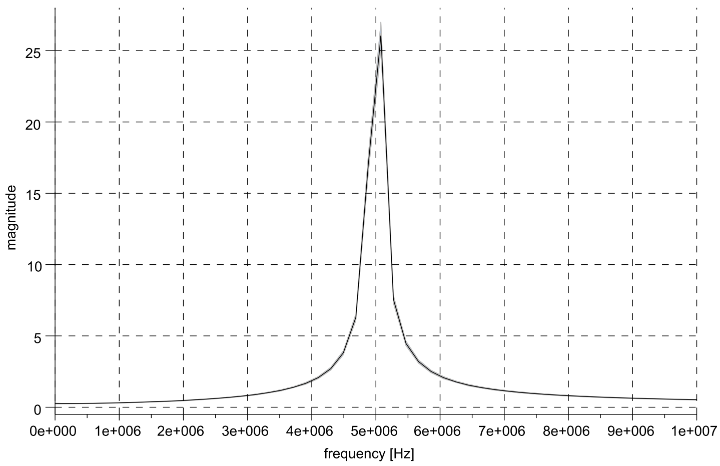

Figure 5.11: Range area of the spectral signal

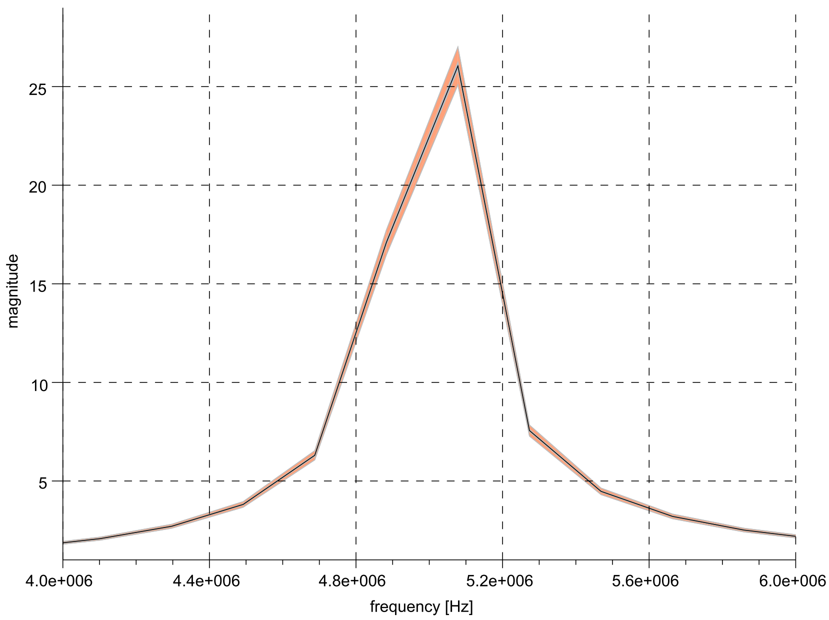

Figure 5.12: Phase jitter contribution to the refined frequency spectra

Figure 5.12 finally shows the influence of the phase jitter on the detailed spectral component in "‘steelblue"’ and in comparison also the impact of all the other user and system deviations. Equally to the analysis in the time domain, the frequency domain evaluation shows a significant reduction of the contribution the phase jitter has on the overall system deviation. The system behavior does not spread so widely any more. The behavior can be expected to be more robust against external distortions and is considered to be more reliable due to minimized internal deviation factors.

Iteration	$x_1\epsilon_1$ jitter	$x_2\epsilon_2$ lna	$x_3\epsilon_3$ offset	$x_4\epsilon_4$ approx.1	$x_5\epsilon_5$ approx.2
1	0.1	0.03	0.05	x	x
2	0.01	0.03	0.05	x	x

Table 5.6: Jitter refinement

Table 5.6 summarizes the refinement in system parameters that lead to a satisfiable system quality. The reduction of the phase jitter resulted in a significant reduction in signal variability and increased the behavior predictability. The analysis of the range based system quantities in the time and the frequency domain showed a strong influence of the phase jitter model on the output signal behavior. The refinement candidate identification (separate block in the "'MARC design refinement flow"') recommended a modification/improvement of the phase jitter deviation. The phase jitter variation was hence reduced and the range based system behavior determined again in a second iteration. The refined system quantities showed much better behavior. A final analysis confirmed a strongly reduced signal deviation and hence and increase in system quality and reliability.

5.3 Refinement in the frequency domain

The third and last demonstration example implements a more complex receiver structure and in particular a higher number of deviation effects. The set-up is compare able to the first example, but integrates more deviation models. The local oscillators signal is also added a phase jitter instead of a simple inaccuracy model and the analysis is performed completely in the frequency domain. The test signal is again a sine wave with frequency of 5 MHz, amplitude modulated on a 13,56 MHz carrier signal. A 25 MHz interferer signal distorts the transmitted signal and appears in the received signal spectrum. The received signal is filtered in a pre-selection filtering stage, amplified by an LNA component, mixed down in the baseband via the mixer stage and finally filtered to remove emerging mirror images.

The system analysis is performed in the frequency domain. The received signal is evaluated after passing through the HF front end structure and reaching the virtual analog to digital domain boundary. The signal is considered to be converted into a digital representation at this point and forwarded to a digital signal post processing unit which finally demodulates the signal. The quantity at this point summarizes all the analog distortions and variations of the receiver structure and could be regarded as quality metric of the analog input stage. A communication receiver is in particular dependent on the spectral behavior of its functionality. The analysis of the spectral behavior at this point allows to determine a benchmark on the system quality especially for the application of communication systems. The analysis does not aim to simply improve the overall spectral behavior or reduce the variability of the spectral components. This example intends to evaluate the correlations within the range based spectrum and identify problematic effects for minimizing them.

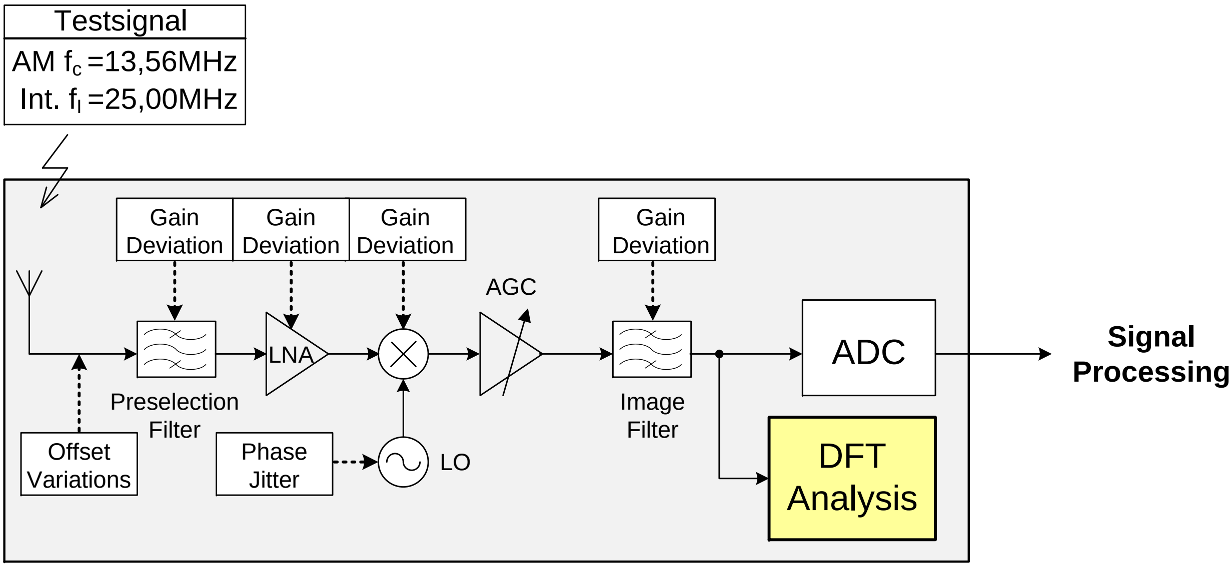

Figure 5.13: Superhederodyne receiver refined in the frequency domain

All analog components in the system add a deviation effect to the range based system model. The analysis covers all these deviation terms in the frequency domain and allows a detailed view on the sensitivities from the analyzed quantity to the originating deviation effects.

$x_1\epsilon_1$ mirror LP	$x_2\epsilon_2$ mixer	$x_3\epsilon_3$ jitter	$x_4\epsilon_4$ lna	$x_5\epsilon_5$ preselection LP	$x_6\epsilon_6$ offset	$x_{7-11}\epsilon_{7-11}$ approx.1-5
0.02	0.04	0.05	0.03	0.01	0.05	x

Table 5.7: Initial system deviations

Table 5.7 gives the single deviation factors of the initial range based system model. The number of deviation terms increased to 11 for this example. The number of user deviations increased to 6 but the increase of non affine mathematical operations result in 5 additional system deviations to handle approximation uncertainties. The first user deviation $x_1\epsilon_1$ models a multiplicative gain deviation of the mirror image removing filter at the end of the receiver stage. $x_2\epsilon_2$ adds a multiplicative gain deviation for the mixer component and $x_3\epsilon_3$ includes the local oscillators phase jitter. The gain deviations of the LNA amplifier $x_4\epsilon_4$ and the pre-selection lowpass filter $x_5\epsilon_5$ finalizes the deviations of the implemented receiver components. $x_6\epsilon_6$ adds an additive deviation on the ground level to model offset variations and potential ground bounces.
The semi-symbolic simulation is again realized in the "'MARC"' framework which uses SystemC AMS as simulation engine and extends the functionality by an Affine Arithmetic library together with additional add on functionality. The Fourier transformation is computed directly in the "'DFT analysis"' block during simulation but the results are exported into the "'SCILAB"' environment for visualization. Figure 5.14 gives the time domain representation of the signal quantity at the analysis point at the end of the analog input stage.

The signal which is analyzed is the filtered baseband signal of the transmitted test signal, superimposed by 11 deviations which are given in different colors. The single contributions to the received signal are easily recognizable. The superposition of all 11 sub ranges on the nominal

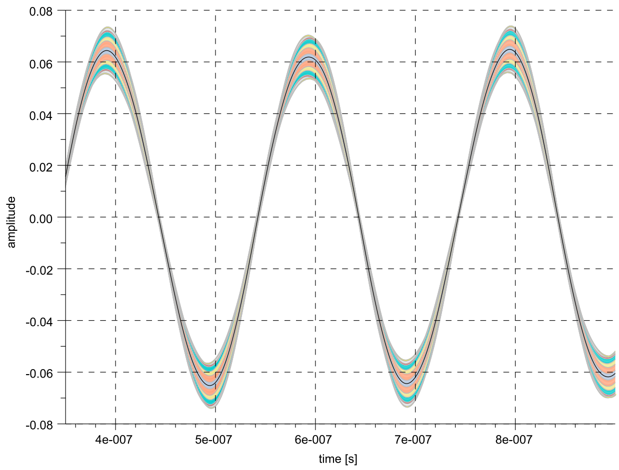

Figure 5.14: Frequency spectra of the refined system behavior

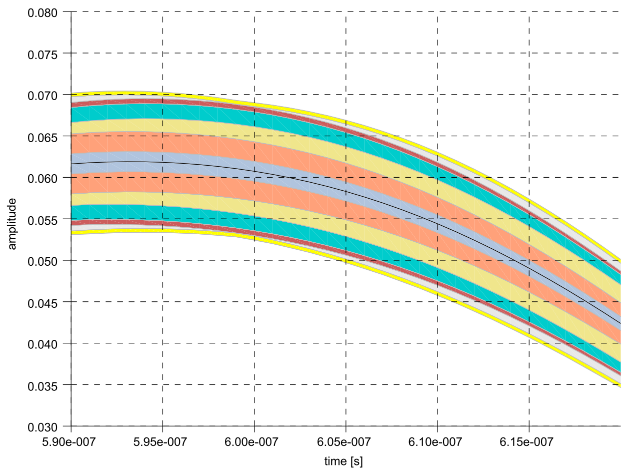

Figure 5.15: Range area of the spectral signal

central value of the Affine Form gives the worst case bound of the received range based signal. In the time domain a significant deviation from the nominal signal characteristic is evident. Closer information on problematic system behavior however is not recognizable within this analysis method.

Figure 5.16 in comparison shows the frequency spectrum of the received signal with its main spectral components. The single deviation contributors are again separated and displayed in different colors. A detailed analysis of the frequency spectrum allows conclusions on problematic behavior. The dominant spectral component at 5 MHz shows the intended baseband signal that was received and processed. The signal shows a deviation from the nominal behavior which gives a measure on how strong the deviation effects influence the received and processed signal. Several further spectral components appear in the spectrum. One and probably the most astonishing is a range area at DC frequency and ultra low frequencies, respectively. This range area appears in "‘yellow"’ which was chosen to color the system deviations that emerge during the processing operations. This deviation can be tracked back to the multiplication of the model offset deviation with the local oscillators sine wave. The multiplication causes an approximation factor to be added the system model which causes a coupling of the additive model offset to the resulting mixer quantity.
At a frequency position of around 11 MHz, portions of the distorting interferer signal appear which are caused by the down mixing operation. This distortion interferes with the intended signal portion at 5 MHz and can be classified as unwanted artifacts of the mixing operation. At frequency position 13,56 MHz finally a significant range area appears which emerges from multiplying the model offset with the local oscillators 13,56 MHz signal. This signal part is also not intended and is given in "‘gray"’ to identify the model offset deviation as its originating source. The frequency spectrum basically shows the spectral behavior of the analyzed system quantity and possible problematic effects. The semi-symbolic simulation extends this basic information about the behavior with the contributions of every deviation factor on the combined spectrum. Identified problematic portions can be distinguished for the impact of every deviation on them and hence a deterministic refinement of system parameters is much easier to apply.

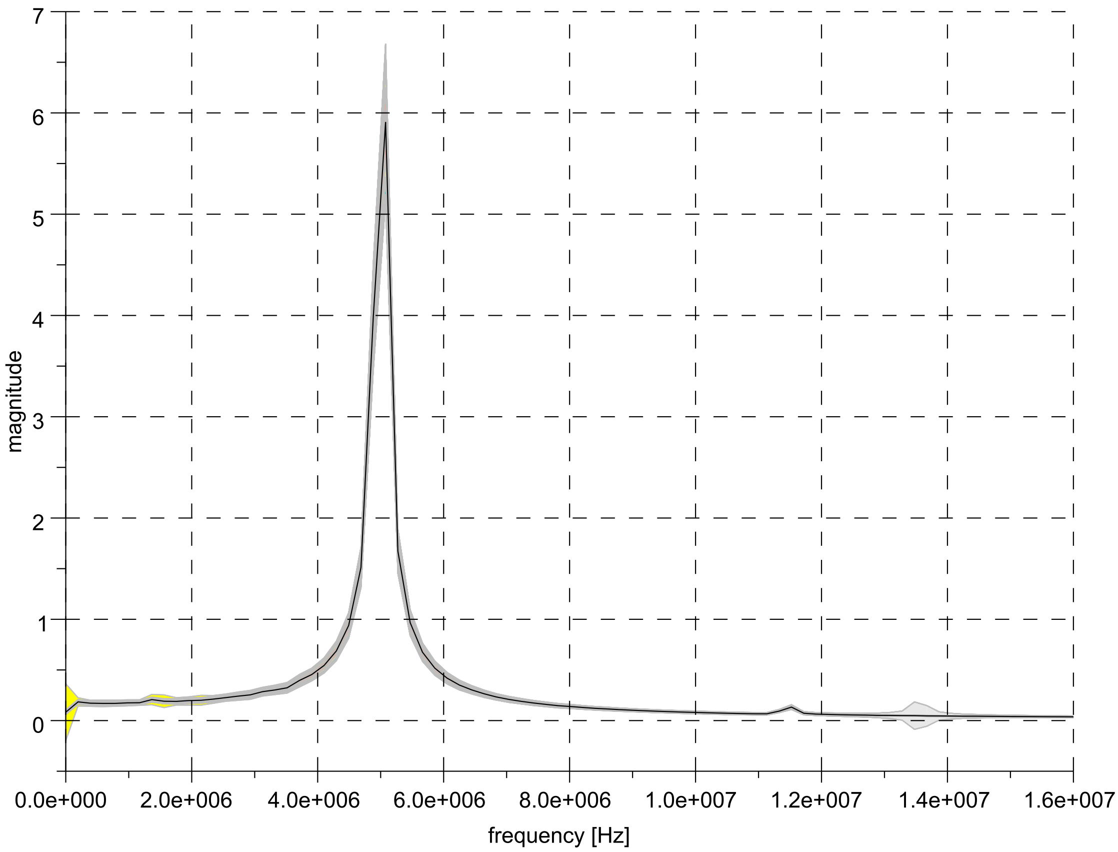

Figure 5.16: Composition of the spectral signal

A detailed view on the main spectral components should help to identify refinement candidates. Figure 5.17 shows a detailed view on the signals main spectral component. The deviations form subranges that add up to the overall range spectrum. The single deviations contribute to the variation area roughly equally and no dominant deviation factor is evident. Figure 5.18 shows a detailed view on the interferer's spectral composition and shows a particular negative influence of the deviation term in "'gray"' which can be identified as model offset deviation.
To summarize the analysis of the range based signal spectrum, several unwanted spectral components were identified. The DC signal portion that originated from the range based multiplication of the model offset deviation in the mixing stage caused a distortion at low frequencies. The interferer signal showed a main contributor in the model offset deviation that increased the variability in the interferer magnitude. Finally, the model offset mixing creates a 13,56 MHz deviation which causes a significant deviation from the nominal signal spectra. All these unwanted signal portions have a source in common, the model offset deviation. The "'refinement candidate identification"' as a consequence picks the model offset to be refined, to improve the overall spectral behavior. A reduction in ground variations is fairly easy to achieve, the additional costs in improving this system parameter can be considered as cheap. Thus, the model offset deviation $x_6\epsilon_6$ is reduced from 0.05 to 0.01. All the other deviation factors are kept identical as they did not show a dominant contribution to the analyzed spectrum.

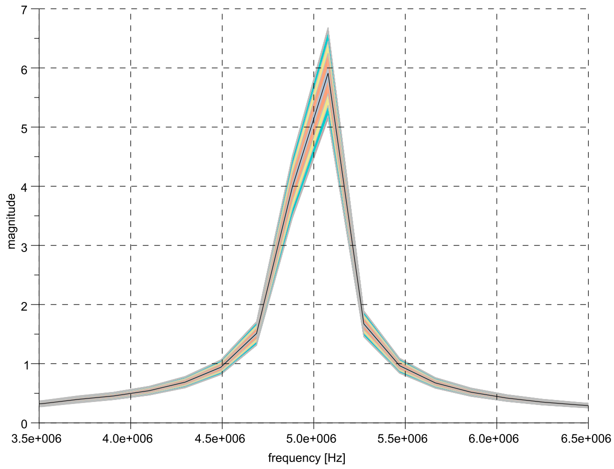

Figure 5.17: Detailed view on the spectral component

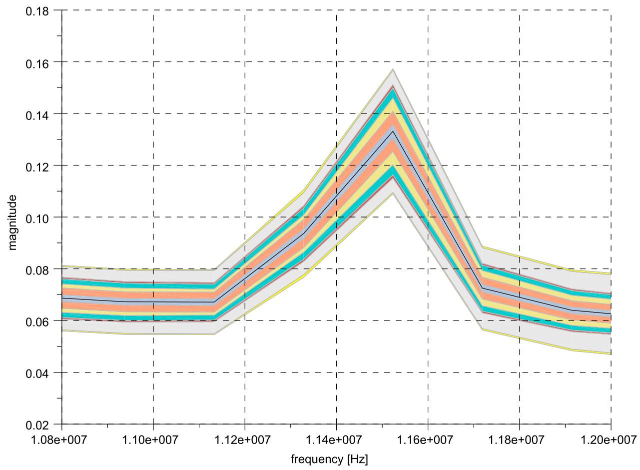

Figure 5.18: Detailed view of the interferer signal

$x_1\epsilon_1$ mirror LP	$x_2\epsilon_2$ mixer	$x_3\epsilon_3$ jitter	$x_4\epsilon_4$ lna	$x_5\epsilon_5$ preselection LP	$x_6\epsilon_6$ offset	$x_{7-11}\epsilon_{7-11}$ approx.1-5
0.02	0.04	0.05	0.03	0.01	0.01	x

Table 5.8: First iteration

Table 5.8 summarizes the refined deviations and shows the system parameters that are used for the first iteration simulation step. The refined system model is again simulated and the received signal after the front end is chosen for a spectral analysis.
Figure 5.19 shows the improved spectral behavior of the refined system. The low frequency distortion almost disappeared as the model offset deviation has been reduced. The coupling effect of the model offset to the 13,56 MHz spectral deviation also almost disappeared. The overall variation of the signals spectrum did not reduce significantly. The appearance of disturbing and completely unwanted spectral effects could be reduced by the improvement of one, deterministically identified deviation factor.
Figure 5.20 and figure 5.21 show again details of the signal's spectral component and the interferer appearance. The modification and improvement of the model offset deviation shows a slight improvement in the deviation from the nominal quantity but does not significantly reduce the variability. The variability directly influences the robustness and reliability of the implemented system. Hence, one of the major refinement objectives is to reduce the overall deviations of the system quantities that are caused by the single deviations, affecting the system parameters. The reduction in overall variation can be achieved by simply reducing all deviation effects without differentiation. This approach is the simplest and does not take into consideration the different costs for parameter refinement or the real impact a deviation has on the overall system. The objective within the "'MARC refinement framework"' is to allow a deterministic refinement candidate identification. Thus, the system parameters which are chosen for refinement will be deduced from the analysis result.

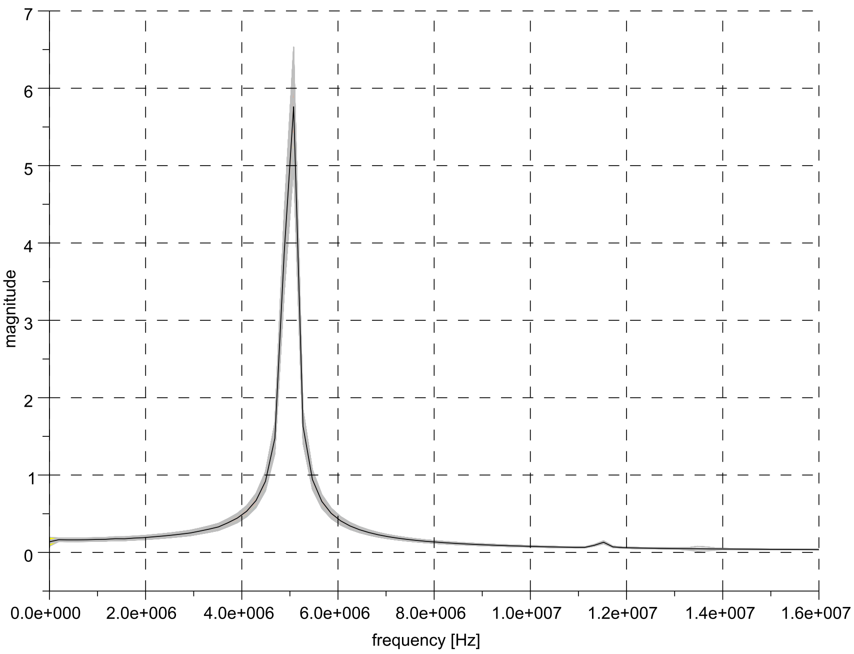

Figure 5.19: First iteration spectral result

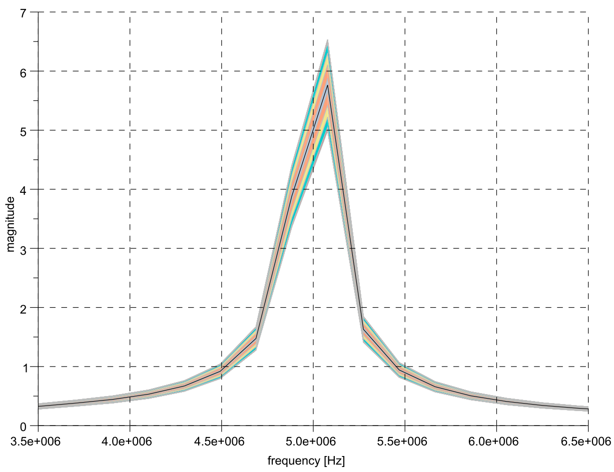

Figure 5.20: Detailed view on the spectral component

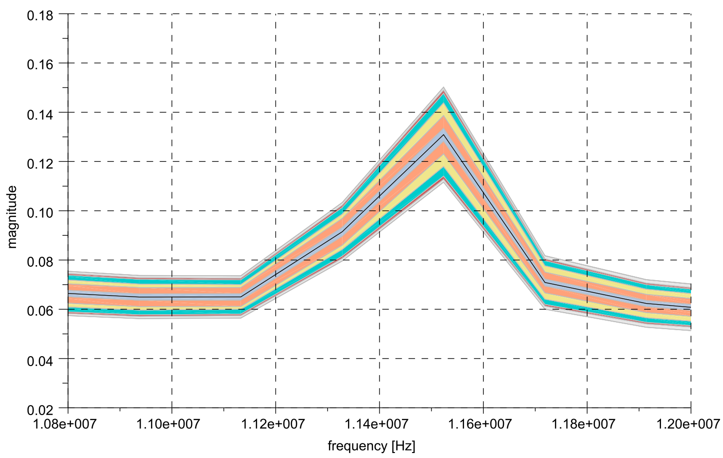

Figure 5.21: Detailed view of the interferer signal

The detailed view on the received signal and the remaining interferer component shows that several deviation factors contribute more to the remaining variability. The model offset's influence on the deviation of the spectral components has become small but especially in the interferer spectrum the mixer deviation in "‘salmon"’ and the phase jitter deviation in "‘khaki"’ show a considerable impact. Although the model offset's influence on the variation of the main spectral components is reduced, it still causes minor spectral deviations at low frequencies and very weak at the LO frequency. Considering all this observations three deviation terms are chosen for a second refinement operation. Table 5.9 summarizes this refinement and shows the modified parameter for the next simulation iteration. To finally remove unwanted spectral components the model offset's deviation is again reduced to the half. The modification of the mixer's gain deviation $x_2\epsilon_2$ from 0.04 to 0.01 and the phase jitter deviation $x_3\epsilon_3$ from 0.05 to 0.01 targets a reduction of the overall deviation in the frequency domain.

$x_1\epsilon_1$ mirror LP	$x_2\epsilon_2$ mixer	$x_3\epsilon_3$ jitter	$x_4\epsilon_4$ lna	$x_5\epsilon_5$ preselection LP	$x_6\epsilon_6$ offset	$x_{7-11}\epsilon_{7-11}$ approx.1-5
0.02	0.01	0.01	0.03	0.01	0.005	x

Table 5.9: Second iteration

Figure 5.22 and figure 5.23 finally show the spectrum of the signal quantity after the second refinement iteration. The distorting spectral components that were caused by the model offset deviation have been successfully reduced to a negligible magnitude and can be considered as unimportant for the system behavior. The deviation of the remaining spectral components have also been significantly reduced by the second refinement step. The overall variability also reduced to a tolerate able quantity and can be considered as sufficient small.
The overall system behavior is evaluated as sufficient and the signal quantities's spectrum has been improved in respect of unwanted spectral components and overall variation of the received signal.

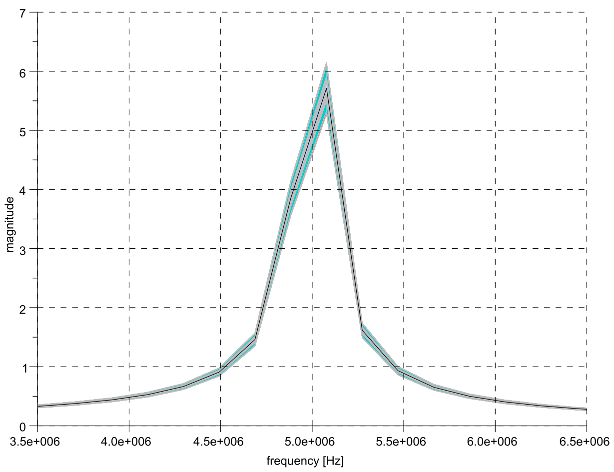

Figure 5.22: Detailed view on the spectral component

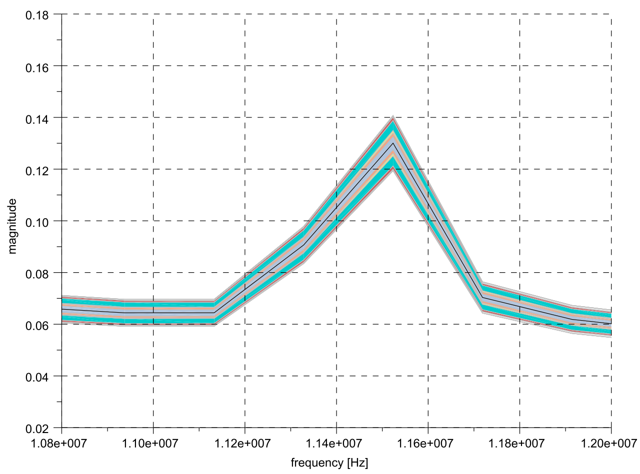

Figure 5.23: Detailed view of the interferer signal

The third demonstration example implemented a communication receiver system with deviations on parameters in every implemented component. Most of the deviations modeled a multiplicative gain deviation but deviation effects for a sine wave's phase jitter and an additive offset variation (ground variation) were also integrated. The range based system model was simulated and the frequency spectrum of the received signal was analyzed for its spectral behavior and potential problems. Two consecutive refinement iterations improved the system quality enough that it could be considered as sufficient and the refinement iteration could be stopped. The optimization of the system behavior just considers the effects the deviation terms have on the system behavior. Basic system design weaknesses, like the not successful removing of the interferer signal from the received signal in the pre-selection filtering are not target of the system refinement in this scope.

5.4 Summary

The "'MARC design refinement flow"' implements a framework for the simulation and analysis of systems affected by parameter deviations. The field of applications this particular design flow is presented for are communication systems. The simulation results strongly depend on the modeled deviation effects. A wide range of possibilities exist to accurately model deviations by range based descriptions. Affine Arithmetic allows mathematical descriptions, hence the models are construct able by mathematical definitions.
Three different optimization approaches were chosen and demonstrated on appropriate example simulations. The first example utilizes a single quality metric to assess the system performance of the simulated system. The semi-symbolic simulation results in range based system quantities that are evaluated for an optimal signal to deviation ratio (SDR). The second example analyzes the system behavior in the time domain as well as in the frequency domain. It evaluates the behavior and picks the main contributing factor and refines it for a system improvement. The third example evaluates the system behavior mainly in the frequency domain. The sensitivity of single deviation effects is utilized to identify negative influencing parameters.
The first example showed the usage of numeric metrics to rate the quality of a system. The second example used the analysis capabilities in the time and the frequency domain to identify one, in particular bad effecting deviation factor. The third example showed the disassembling capabilities of semi-symbolic simulations to analyze the influence of a set of deviation terms on the system behavior. No single analyses option exist which solves all the behavior estimating problems. In contrast a bundle of several valuable methodologies are available and help in classifying the system behavior. The refinement candidate identification is a crucial part in the proposed refinement design flow. The semi-symbolic nature of the simulation helps in analyzing the sensitivity a system quantity has on its contributing deviation factors.

6 Conclusion and Discussion

Robustness and reliability of today's electronic systems is a crucial property to satisfy customers demands and to persist in the market. Erratic or even sensitive behavior of systems that are shipped to customers cause huge damage to the reputation of technology companies. The traditional approach to evaluate the functionality of implemented systems rely on multi-run simulations. The influence of parameter variations on the functionality of the system is analyzed by repeated simulations with a modification of single system parameters in every simulation run. These simulation approaches rely on the simulation performance of modern computer systems to keep the overall run time of the numerous simulation runs within acceptable ranges. Unfortunately, the necessary simulation effort increases much faster than the performance increase reachable by new computer generations. Today's available electronic systems experience a steady increase in its complexity and number of integrated components. Additionally to that rise in complexity the number of model parameters in system simulations is steadily increasing and the ongoing success in device integration causes a significant increase in parameter variations. All these effects multiply the number of necessary simulation runs for a multi-run analysis to reach an acceptable simulation coverage. This intensified simulation effort causes a simulation performance gap that pose the possibility of hampering future system simulation methodologies and by that system analysis approaches.

Semi-symbolic simulations avoid this drawbacks. A semi-symbolic simulation uses range descriptions to model deviations of system parameters and computes range based system quantities as simulation results. The consideration of parameter deviations in range descriptions allows the calculation of all potential signal characteristics reachable from the modeled system parameters and their deviation. This fact reduces the number of simulation runs to obtain a useful simulation coverage from numerous to one. The range based semi-symbolic simulation run determines the range based system quantities that are used to evaluate the system behavior. The simulation results not only consist of a numerical quantity that defines the system behavior at a given simulation time point. The semi-symbolic simulation provides a range of possible signal characteristics that reflect the deviated system quantities. The resulting range is not specified by only one range parameter it rather comprises of a set of sub ranges contributing to the selected system quantity. The single sub ranges are specified by symbolic identifiers and resulting range quantities are calculated if necessary by conservative approximations. The resulting range quantity guarantees to cover all areas that are reachable by the original range descriptions and rather accepts an over approximation instead of an under approximation that would result in missed system behavior.

The name semi-symbolic simulation indicates the combination of numerical results with additional symbolic descriptions. The symbolic nature of this approach allows the identification of single contributors to the calculated system quantities. By utilizing this possibility a deterministic identification of deviation effects that unfavorable affect the system behavior is feasible. The symbolic identifiers in the range based system quantities point to the sources of sub ranges and allows the deterministic refinement of system parameters to improve the overall system quality. The main objective of the presented thesis is to create a refinement design flow that is used to improve the quality of deviated system implementations in a deterministic way. As described, the quality of a system directly correlates with the robustness and reliability of the system under investigation. The semi-symbolic approach especially supports this needs. The symbolic nature of the simulation result allows the identification of advantageous deviation parameters for refinement and in this way to deterministically choose refinement objects. The semi-symbolic simulation is embedded into the "'MARC framework"' which basically consists of a general SystemC AMS environment extended by an Affine Arithmetic library and add-on functionality. The SystemC AMS environment provides an efficient simulation engine and the Affine Arithmetic library overloaded mathematical operations for the simulation of range based system models.

Range based system models which are used for semi-symbolic simulations are created in three steps.

- Numeric SystemC AMS model
- Specification of the deviation effects
- Combined range based system model

The initial numeric SystemC AMS model integrates the core functionality of the designed system. The parameters are specified by their nominal value and the system is implemented with the intended system behavior. In a second step the system is analyzed for potential parameter deviations. The parameter deviations that are identified in this step are modeled as range terms by using Affine Arithmetic and its Affine Forms. Affine Arithmetic defines a selection of mathematical operations on Affine Forms. This mathematical operations may be used to specify the deviation functions that apply best for the identified deviation effect. The restriction for the deviation models only exist in the mathematical description of their characteristic and they are describe able also in complex equational descriptions. The identification of deviation effects and the modeling of them for the range based system model is a crucial process in the design flow. Deviations that are not considered in the simulation model do not contribute to the analyzed system behavior and hence may hide problematic behavior in the actual implemented system. In other words, the analysis and refinement operation is just as good as the system model considers all essential deviation effects in its simulation model. The last step to create the range based system model (which is the basis for the semi-symbolic simulation) is the combination of the numeric (nominal) system model with the range modeled deviations. The combined model is implemented in the SystemC AMS environment and is basically the nominal model with its functionality extended by the deviation models. A simulation run of the combined range based model results in range based system quantities which could be used for a following system analysis.
The objective of the "'MARC refinement design flow"' is not just to simulate the system model on a range based basis. The intended objective is to find system parameters for refinement to improve the system quality and reliability. To find refinement candidates an analysis of the system behavior is an essential step. The analysis should allow a well-directed refinement of particular system parameters to efficiently improve the system behavior. The analysis relies on the symbolic properties of the range based system quantities which allow a back tracking of sub ranges to their

originating sources. Additionally to the symbolic analysis also a class of system rating criterions are used.

These so called quality metrics allow the measuring of system behavior.

- quality metric
- combination of metrics
- cost function

The quality metrics can be defined in different complexity. The first and easiest approach is to use a single quality metric to rate the system. This quality metric can be expressed as numerical value and is used to guide the refinement process to achieve an improvement in system robustness. Single numerical quality metrics are very convenient for decision processes but rarely cover all properties that are considered as main contributing factor to the system behavior. Following this restriction a combination of quality metrics can be used to measure a multi dependent system behavior. Here, a selection of quality metrics is used and the rating of the system quality is found by combining the single factors to a comprehensive measure. Again, the selection of the "'important"' quality metrics is not obvious and a careful selection has to be found. One step further is the definition of cost functions to find a process of automated quality assessment. The refinement of system parameters is always associated with costs, either in a monetary or more general in engineering costs. A cost function regards this efforts and defines a function where the single metrics are scaled by different cost factors. Using cost functions allows to concentrate the refinement process on parameters which represent refinement candidates that show a good trade off between the impact they have on the system behavior and the costs that are associated with an improvement of the respective deviation.

Based on this quality measures and cost functions as well as an analysis of the simulated system quantities the "'MARC refinement design flow"' decides which system parameters should be modified to improve the system quality in a repeated refinement iteration. The system analysis uses the semi-symbolic simulation quantities to determine the influence single or also a set of deviation effects have on the system behavior in different domains. The analysis is applicable in the time as well as in the frequency domain. The time domain allows the efficient analysis of rare events and non periodic signal characteristics. The frequency domain in contrast allows the examination of periodic quantities and offers a wide range of analysis methodologies. In both the time and the frequency domain, the system quantities appear as range based quantities that consist of a central value which is superimposed by a set of deviation caused sub ranges. The analysis of system behavior in the time and the frequency domain allows most of the modern system analysis techniques to be applicable. Transient simulations allow the evaluation of arbitrary input stimuli on the system behavior, a frequency representation allows the analysis of spectral components in the system signals and even range based constellation diagrams allow the evaluation of communication receiver robustness.

The "'MARC refinement design flow"' combines all presented simulation and analysis methodologies into one framework. The semi-symbolic simulation is implemented as SystemC AMS simulation environment which is extended by an Affine Arithmetic library. The used library was implemented new by Michael Rathmair of the Institute of Computer Technology. The new implementation [RSRG12] is especially useful for tracking single deviation terms throughout the simulation process. This extended tracking functionality was not available in earlier implementations and is crucial for a guided refinement process. Along with the newly implemented tracking functionality also some inconsistencies in older implementations were corrected and an additional non-affine approximation variant is introduced. The analysis functionality is bundled in an add-on package that extends the basic SystemC AMS functionality in particular for range based quantities. Additionally to analysis options, also tracing functionality has been implemented to allow off-line processing (i.e. MATLAB post processing, SCILAB visualization,...)

The "'MARC refinement design flow"' performs the following operations

- Initial system simulation (range based)
- System analysis
- System modification

The "'MARC refinement design flow"' can be divided into three main processes. The refinement process must always start with a system model that should be analyzed for improvements. This system model is extended by a set of effective deviation models that expand the system model. The resulting range based system model considers deviations of the chosen system parameters. The initial system model implements the designed system parameters with their variations and uncertainties that appear in real systems. The initial system model is subsequently simulated and the range based system quantities are analyzed for their properties. This system analysis uses the possibilities of semi-symbolic parameter identification and all the implemented analysis methodologies to evaluate the behavior of the deviation affected system. The system analyses with its advanced evaluation methodologies or a combination of analysis with the quality metric approach determines system parameters that are chosen to be refined. These refinement candidates are updated in the third step and the modified system model is simulated again to compute the refined and hopefully improved system quantities. The system modification can be restricted to the update of single system parameters or if the preceding analysis suggests a set of parameters.

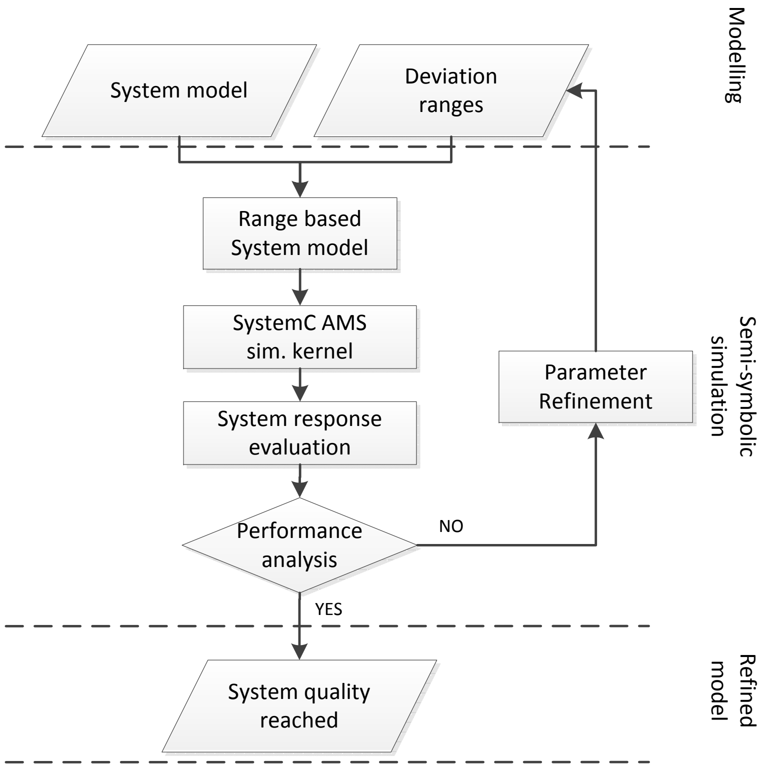

Figure 6.1: MARC design flow

Figure 6.1 shows the "'MARC refinement design flow"' in flow chart representation. The design flow starts in a modeling phase, where the nominal system model but also the identified deviation effects are created. The combination of this two models create the range based system model which is implemented by a SystemC AMS description using a *TDF* model of computation. The system model is simulated by using the SystemC AMS simulation kernel and the previously mentioned Affine Arithmetic library. Input stimuli signals are applied to the system model and a range based system response is computed by the simulation kernel. The deviation range descriptions cause numerical input vectors to be translated to range quantities as the system functionality also is deviated in its behavior. The following step is the system evaluation. The range based system quantities are analyzed for their compliance with the intended system properties and are evaluated for a possible improvement of the system parameters. The so called performance analysis evaluates the system quality with a given target quality. The blocks from the range based system model to the performance analysis can be grouped to the phase semi-symbolic simulation. Within this phase the range based model is simulated and the resulting range quantities analyzed for their behavior.

If the system performance is not sufficient then a refinement iteration is started. In this iteration step the deviated system parameters which contribute worst to the system quality are identified and chosen for refinement. This refinement happens in the range based system model. The identified system parameters are modified in their variation. A repeated simulation run generates the system quantities of the refined system model which can be used for a new performance analysis. The iteration between the system quantity analysis and the parameter refinement is repeated until a sufficient system quality is reached and the system can be considered as robust and reliable.

6.1 Discussion

Semi-symbolic simulations show their potential in simulating systems with parameter deviations. The implicit deviation consideration that is achieved by the superimposed range descriptions allows an efficient simulation process. The increase in simulation performance is primarily bought by reducing the number of simulation runs for given deviations to one. This reduction in runs is achieved on the costs of increased computation complexity which is for an academic implementation of semi-symbolic simulations negligible. The current research focuses on creating new methodologies for semi-symbolic simulations and not in optimizing the simulation library itself. This results in regular simulation performance problems when simulating more complex systems with an increased number of deviation terms. The implemented system analysis techniques that operate on the range based system quantities have to iterate through the existing deviation terms to compute range dependent analysis results. This iteration also restricts the efficiency of the proposed simulation framework. For instance the calculation of a 512 point DFT in the third implemented example lasted 4 hours on a single core Pentium processor. The integration of a FFT algorithm would significantly improve the computation performance but the limitations of increased computation complexity still exists.
The use of a garbage collection as proposed by [HGW05] could help to reduce the number of deviation terms during a simulation run. However, the new implementation [RSRG12] of the Affine Arithmetic library considers the approximation terms as function over time which reduces the number of system generated deviation terms to the number of non-affine operations. A garbage collection would not significantly improve simulations using this library.
The new Affine Arithmetic library extends the functionality from a plain mathematical operation definition to an enhanced deviation term labeling functionality. The main problem when using semi-symbolic simulations for a system refinement is to identify sub ranges of the analyzed system quantity and identify its corresponding source. In the previous implementations, only the numeric deviation ranges were available. The index position in the implemented static deviation vector corresponded to the originating source. System deviations created during non-affine operations appeared at arbitrary positions, certain mathematical operations caused a reordering of certain ranges and the use of the garbage collection fully reordered this static vector and its deviations. The previously used Affine Arithmetic libraries were intended for a worst case analysis of the modeled and simulated system. This worst case estimation does not require information about the inner constitution of the range based quantities. The summation of the single range quantities provide all the information that was intended. This reduction of the semi-symbolic quantities to the worst case bounds is not sufficient for refinement decisions. This corresponds to the lack of information when performing purely numerical simulation and obtaining just statements if the system behavior complies to specifications but without any information which part of the system violated the requirements.
Especially the fact that the position of the deviation terms inside the implemented vector was not fixed is problematic. Hence, an extended functionality for the tracking of deviation terms and the identification of its originating sources had to be found. The division of the deviation terms into "'user deviation"' and "'system deviation"' provided the first approach in finding a new data structure for Affine Forms. The implementation of a second identification structure and an object orientated generation of deviation terms instead of a static list/vector extended the functionality that the deviation terms are trackable at every simulation stage. Also "'system deviations"' are labeled additionally to their ϵ label with a string literal that specifies the originating mathematical operations and the contributing deviation terms. Obviously, the overhead in deviation specification (string identification structures and tracking functionality) and the object

oriented implementation of the deviation list causes a reduction in simulation performance. This performance decrease is tolerated because of the valuable functional extension that is necessary for a deterministic deviation identification used in the refinement design flow. The new library implementation tried to stabilize the library implementation. The data structures that were chosen should be sufficient for future applications and are intended to be kept for the future library versions.
The semi-symbolic simulation environment is also not fixed in its functionality. The Affine Arithmetic library provides mathematical operations to be used for range based quantities but does not specify how deviations are to be modeled. This work tried to present a selection of already existing deviation models and summarized system/circuit effects in range description notation. The "'MARC refinement framework"' is a work in progress where the functionalities and modules are permanently enhanced and improved.
The ICT working group on range based simulations tried to increase the modeling and simulation capabilities for semi-symbolic simulations. Several enhancements were presented within this thesis (phase jitter model, range based Discrete Fourier Transform, range based Laplace transform) and have been bundled in an add-on library to the simulation framework. A building block library has been implemented to simplify the model creation for range based system models. System designers should be supported with often used functional units that are already described by range descriptions where the model can easy be specified by parametrization. An additional add-on package provides the system designer with tracing, analysis and visualization methods. The analysis and visualization is a major part in the refinement design flow. A selection of methodologies have already been implemented but numerous additional are available. The most important analysis extension within this work is the translation of range based quantities into the frequency domain. The most powerful extension to the tracing functionality of SystemC AMS is the trace file creation of range quantities with their single deviations for an offline post processing. The complex analysis tools that were chosen within this work are MATLAB with its powerful signal processing engine and SCILAB for a combined post processing with its handy visualization capabilities.
The demonstration examples chosen in this work aimed to show the analysis options that were presented in earlier sections. The communication system examples were chosen to demonstrate the refinement design flow possibilities for this class of applications. Communication systems are typically sensitive to distortions which directly influence the error rate of received messages. Hence, the robustness of a communication system is an important design criterion and additional design steps are encouraged to increase its robustness and the correlated system reliability.
The first demonstration example showed how a single quality metric can be used to measure but also improve the system behavior. The quality metric used was the signal to deviation ratio (SDR) which was measured as maximum value in this example. The assessment of this particular metric decided if additional refinement iterations were necessary or if the system quality satisfied the requirements. The second example used both, a time domain analysis and a frequency domain analysis. The phase jitter deviation model contributed most to the system behavior in the time domain as well as in the frequency domain. The improvement of this single deviation seriously improved the system behavior of the overall system. An improvement in phase jitter can be achieved by the use of a higher quality oscillator component and is usually easy to implement. The example showed the impact of the phase jitter on the time characteristic but also on the frequency domain characteristic of the received signal. The third example finally used the tracking possibilities to decompose the resulting range quantity in all of its contributing deviations and identified the worst contributing factors. The frequency domain spectral analysis obtained several interfering deviation effects that were refined in several iteration steps to improve the spectral

behavior of the modeled system. The main objective in the third example was to show how a semi-symbolic simulation supports the decomposition of the range quantities into the constituting sub ranges. The size of a sub range at problematic system behavior directly gives the influence that the particular deviation has on this behavior. By analyzing the influence of the single sub ranges a well guided refinement decision is easier to obtain.
Besides the identification and impact estimation of deviation effects on the system behavior the definition of "‘good"’ quality metrics to measure the system quality is an important refinement flow task. Quality metrics simplify the task of system rating and supports the automation of the proposed design flow.

6.2 Outlook

A semi-symbolic simulation environment offers manifold advantages over traditional simulation approaches. A worst case estimate is much more efficient to obtain, as there is just one simulation run to be computed. Also the worst case bounds are pessimistic approximations which allows to guarantee the full coverage of potential signals. The current implementation of the Affine Arithmetic library does not fully take advantage of the efficiency gain over multi-run simulations. The introduced data structures for deviation identification and symbolic labeling in general restrict the efficiency of semi-symbolic simulations heavily. One of the most important improvements for semi-symbolic simulations would be to increase the simulation performance. The C++ based library implementation has to be reviewed for computation bottle necks. The mathematical operations have a potential for improvement which would increase the simulation performance significantly, as they are used frequently in a simulation process. The complete simulation environment has to be evaluated for its performance and interactions of SystemC AMS and the Affine Arithmetic library analyzed for improvements.
The semi-symbolic simulation reaches a good coverage of system properties if the underlying range based system model is complete (every significant deviation effect is considered). The modeling of deviation effects is still an issue for future work. Every particular deviation term has to be carefully described. Hence, the deviation models should be expanded to allow the specification of further deviation terms with an increased specification complexity. The sources for deviations are manifold in real implementations. New integration techniques cause novel deviation effects, increased awareness of variations and the numerous possibilities for deviations in "‘real"’ systems demand for pre-defined and easy to use deviation models. The ICT implemented a building block library for a simplification of range based model creation. These building blocks introduce the first step in supporting a user friendly simulation framework. The number of range based models provided by the building block library is still too small. Additional work have to be put into extending the functionality and number of the building block add-on library. The introduction of the building blocks was intended also to simplify the creation of example simulations for interested designers.
The semi-symbolic simulation framework presented in this thesis was also extended for a mixed-level simulation by a research partner. The system model on system level is here extended by a transistor level description for certain blocks in the model. Both levels of abstraction (system level, transistor level) operate on range based quantities and compute a range based system response. This feature is particular useful as it allows the analysis of every block on the granularity best suited for the intended scope. The integration of the transistor level solver into the "‘MARC simulation framework"’ exists but is still restricted. The coupling of the internal transistor level solver time base with the SystemC AMS simulation time exists but the creation of the block

equations and the need for a pre-specified symbolic input signal prevents the efficient use of this extension.
The refinement design flow itself represents the biggest potential for improvement. In the current implementation the analysis and modification operation is performed by a "'well-guided"' user interaction. The range based system model is simulated and the range based system quantities are analyzed for possible improvements. The identified improvement parameters are evaluated by a designer and refinement candidates chosen that are finally modified by updating the initial range based system model. An automated optimization process would significantly improve this design flow.
The first steps towards an automated design flow have already been taken. The introduction of quality metrics and the formulation of cost functions allow the automatic reasoning about the system quality. The identification of refinement candidates is still too fuzzy and has to be improved for an automated selection. The dynamic modification of range parameters is yet not supported but could be introduced with just medium complexity. One of the missing links towards an automated optimization is a suitable optimization algorithm. Careful consideration should be put into this decision as the selection would also influence the simulation performance.
A further possible enhancement for semi-symbolic simulations would be the introduction of additional physical domains. All technical systems (whether mechanical, constructional, chemical...) contain variations of certain system parameters. These variated systems can also be described by a range based system model and simulated in a semi-symbolic simulation. All the introduced analysis methodologies are applicable and possible problematic behavior is identifiable. The introduction of multiple domains also allows the description of cross domain system properties. Certain system properties affect component behaviors in more than one physical domain. For instance, the current through a component heats a section of a system which indirectly influences the behavior/functionality of a nearby component.

Literature

[ASG12] Adhikari, Sumit ; Schupfer, Florian ; Grimm, Christoph: Co-Simulation Framework for Variation Analysis of Radio Frequency Transceivers. In: *System, Software, SoC and Silicon Debug Conference (S4D)*, 2012, S. 1–6

[BCM+90] Burch, J.R. ; Clarke, E.M. ; McMillan, K.L. ; Dill, D.L. ; Hwang, L.J.: Symbolic model checking: 1020 States and beyond. In: *LICS* (1990), Nr. 2

[BVGE10] Barnasconi, Martin ; Vachoux, Alain ; Grimm, Christoph ; Einwich, Karsten: *Standard SystemC AMS Language Reference Manual*. OSCI, 2010

[CGJ+03] Clarke, Edmund M. ; Grumberg, Orna ; Jha, Somesh ; Lu, Yuan ; Veith, Helmut: Counterexample-guided abstraction refinement for symbolic model checking. In: *Journal of the ACM (JACM)* 50 (2003), Nr. 5, S. 752–794

[CH02] Chan, M. ; Hu, C.: Engineering BSIM for the Nano-Technology Era and Beyond. In: *Proceedings of the 5th International Conference on Modeling and Simulation of Microsystems*. San Juan, Puerto Rico, 2002, S. 662–665

[Coo71] Cook, Stephen A.: The complexity of theorem-proving procedures. In: *Proceedings of the third annual ACM symposium on Theory of computing*, 1971 (STOC '71 8), S. 151–158

[CR09] Claessen, Koen ; Roorda, Jan-Willem: A Faithful Semantics for Generalised Symbolic Trajectory Evaluation. In: *Logical Methods in Computer Science* 5 (2009), Nr. 2

[FCR03] Fang, Claire F. ; Chen, Tsuhan ; Rutenbar, Rob A.: Floating-Point Error Analysis Based On Affine Arithmetic. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, 2003, S. 561–564

[FRV96] Fernandez, Francisco V. ; Rodriguez-Vazquez, Angel: Symbolic Analysis Tools - The State of the Art. In: *Circuits ans Systems, 1996. ISCAS'96, Connecting the World, 1996 IEEE International Symposium on* Bd. 4, 1996, S. 798–801

[FS97] de Figueiredo, Luiz H. ; Stolffi, Forge: *Self-Validated Numerical Methods and Applications*. IMPA/CNPq, Rio de Janeiro, Brazil, 1997 (Brazilian Mathematics Colloquium monographs)

[GGB06a] Grabowski, Darius ; Grimm, Christoph ; Barke, Erich: Ein Verfahren zur Effizienten Analyse von Schaltungen mit Parametervariationen. In: *Tagungsband GI/ITG/GMM - Workshop Modellierung und Verifikation '03*. Dresden, Mar 2006

[GGB06b] Grabowski, Darius ; Grimm, Christoph ; Barke, Erich: Semi-Symbolic Modeling and Simulation of Circuits and Systems. In: *International Symposium on Circuits and Systems 2006 (ISCAS 2006)*. Kos, Greece : IEEE Press, May 2006

[GHW04] Grimm, Christoph ; Heupke, Wilhelm ; Waldschmidt, Klaus: Refinement of

Mixed-Signal Systems with Affine Arithmetic. In: *Design, Automation and Test in Europe 2004 (DATE 04)*, IEEE Press, Feb 2004

[GHW05] Grimm, Christoph ; Heupke, Wilhelm ; Waldschmidt, Klaus: Analysis of Mixed-Signal Systems with Affine Arithmetic. In: *IEEE Transactions on Computer Aided Design of Circuits and Systems* 24 (2005), Nr. 1, S. 118– 123

[GOB08] Grabowski, Darius ; Olbrich, Markus ; Barke, Erich: Analog circuit simulation using range arithmetics. In: *ASP-DAC '08: Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. Seoul, Korea : IEEE Computer Society Press, 2008, S. 762–767

[God96] Godefroid, Patrice ; van Leeuwen, J. (Hrsg.) ; Hartmanis, J. (Hrsg.) ; Goos, G. (Hrsg.): *Partial-Order Methods for the Verification of Concurrent Systems: An Approach to the State-Explosion Problem*. NJ, USA : Springer-Verlag New York, Inc., 1996

[GOGB07] Grabowski, Darius ; Olbrich, Markus ; Grimm, Christoph ; Barke, Erich: Range Arithmetics to Speed up Reachability of Analog Systems. In: *Forum on specification and Design Languages, FDL 2007*, 2007, S. 38–43

[GPHB06] Grabowski, Darius ; Platte, Daniel ; Hedrich, Lars ; Barke, Erich: Time Constrained Verification of Analog VCircuits using Model-Checking Algorithms. In: *Electronic Notes in Theoretical Computer Science (ENTCS)* 153 (2006), Nr. 3, S. 37–52

[Gra09] Grabowski, Darius: *Gebietsarithmetische Verfahren zur Simulation analoger Schaltungen mit Parameterunsicherheiten*. Verlag Dr.Hut, 2009

[GRB+06] Gielen, G. ; Rutenbar, R. ; Borkar, S. ; Brodersen, R. ; Jue-Hsien, C. ; Naviasky, E. ; Saias, D. ; Sodini, C.: Tomorrow's analog: just dead or just different? In: *Design Automation Conference, 2006 43rd ACM/IEEE*, 2006, S. 709–710

[GRGR+99] Guerra, O. ; Rodriguez-Garcia, J. D. ; Roca, E. ; Fernandez, F. V. ; Rodriguez-Vazquez, A.: An Accurate Error Control Mechanism for Simplification Before Generation Algorithms. In: *Design, Automation and Test in Europe Conference and Exhibition 1999. Proceedings*, 1999, S. 412–416

[HC98] Huang, Shi-Yu ; Cheng, Kwant-Ting: *Formal Equivalence Checking and Design Debugging*. Kluwer Academic Publishers, 1998

[HGW05] Heupke, Wilhelm ; Grimm, Christoph ; Waldschmidt, Klaus: Semi-symbolic simulation of nonlinear systems. In: *Forum on Specification and Design Languages 2005 (FDL05)*. Lausanne, Sep 2005

[HJSS06] Helton, J.C. ; Johnson, J.D. ; Sallaberry, C.J. ; Storlie, C.B.: Survey of sampling-based methods for uncertainty and sensitivity analysis. In: *Reliability Engineering & System Safety* 91 (2006), Nr. 10-11, S. 1175–1209

[HKH04] Kap. Formal Verification for Nonlinear Analog Systems: Approaches to Model and Equivalence Checking In: Hartong, Walter ; Klausen, Ralf ; Hedrich, Lars: *Advanced Formal Verification*. Springer US, 2004, S. 205–245

[HSB95] johan H. Seger, Carl ; Bryant, Randal E.: Formal Verification by Symbolic Evaluation of Partially-Ordered Trajectories. In: *Formal Methods in System Design*, 1995, S. 147–190

[JH08] Jesser, Alexander ; Hedrich, Lars: A symbolic approach for mixed-signal model checking. In: *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, 2008 (ASP-DAC '08), S. 404–409

[Kah74] Kahn, Gilles: The semantics of a simple language for parallel programming. In:

Proceedings of IFIP Congress 74, 1974, S. 471–475

[Kle95] Kleijnen, Jack P.: Sensitivity Analysis and Optimization in Simulation: Design of Experiments and Case Studies. In: *Proceedings of the 27th conference on Winter simulation*, 1995, S. 133–140

[Kle08] Kleijnen, Jack P.: Design of experiments: overview. In: *Proceedings of the 40th Conference on Winter Simulation*, 2008 (WSC '08), S. 479–488

[KLM05] Koelbl, Alfred ; Lu, Yuan ; Mathur, Anmol: Embedded tutorial: formal equivalence checking between system-level models and RTL. In: *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, 2005 (ICCAD '05), S. 965–971

[LAOW05] Lee, Dong-U ; A.A., Gaffar ; O., Mencer ; W., Luk: MiniBit: bit-width optimization via affine arithmetic. In: *Proceedings of the 42nd Design Automation Conference*, 2005, S. 837–840

[LCNT07] Lopez, Juan A. ; Carreras, Carlos ; Nieto-Taladriz, Octavio: Improved Interval-Based Characterization of Fixed-Point LTI Systems With Feedback Loops. In: *IEEE Transactions on Computer Aided Design of Circuits and Systems* 26 (2007), Nr. 11, S. 1923–1933

[LM87] Lee, Edward A. ; Messerschmitt, David G.: Synchronous Data Flow. In: *Proceedings of the IEEE* Bd. 75, 1987, S. 1235–1245

[Loz83] Lozier, David W.: The Use of Floating-Point and Interval Arithmetic in the Computation of Error Bounds. In: *IEEE Transactions on Computers* Bd. C-32, 1983, S. 411–417

[MGP99] M.Clark, Edmund ; Grumberg, Orna ; Peled, Doron A.: *Model Checking*. MIT Press, 1999

[Mh02] Ming-hua, Hsieh: Adaptive monte carlo methods for rare event simulation. In: *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, 2002 (WSC '02), S. 108–115

[Moo66] Moore, R. E.: *Interval Analysis*. Eaglewood Cliffs, NJ : Prentice-Hall, 1966

[MT06] Messine, Frederic ; Touhami, Ahmad: A General Reliable Quadratic Form: An Extension of Affine Arithmetic. In: *Reliable Computing* 12 (2006), Nr. 3, S. 171–192

[OH05] Olukotun, Kunle ; Hammond, Lance: The Future of Microprocessors. In: *Queue* 3 (2005), Nr. 7, S. 26–29

[OSG11a] Ou, Jiong ; Schupfer, Florian ; Grimm, Christoph: Modeling Quantization Error of DSP Systems using SystemC AMS. In: *Proceedings of the VW FEDA 2011*. Southampton UK, 2011

[OSG11b] Ou, Jiong ; Schupfer, Florian ; Grimm, Christoph: System Level Communication System Design using Extended SystemC AMS Building Block Library. In: *Tagungsband Austrochip 2011*, 2011, S. 39–43

[RGDP10] Rafaila, M. ; Grimm, Ch. ; Decker, Ch. ; Pelz, G.: Sequential design of experiments for effective model-based validation of electronic control units. In: *e & i Elektrotechnik und Informationstechnik* 127 (2010), S. 164–170. – 10.1007/s00502-010-0738-x. – ISSN 0932–383X

[RSRG12] Rathmair, Michael ; Schupfer, Florian ; Radojicic, Carna ; Grimm, Christoph: Extended Framework for System Simulation with Affine Arithmetic. In: *Proceedings of the 2012 Forum on specification & Design Languages*, 2012, S. 161–168

[Rub81] Rubinstein, R.Y.: *Simulation and the Monte Carlo Method*. New York, NY, USA : John Wiley&Sons, Inc., 1981

[SG10] Schupfer, Florian ; Grimm, Christoph: Towards more Dependable Verification of Mixed-Signal Systems. In: Becker, Bernd (Hrsg.) ; Cardelli, Luca (Hrsg.) ; Hermanns, Holger (Hrsg.) ; Tahar, Sofiene (Hrsg.): *Verification over discrete-continuous boundaries*. Dagstuhl, Germany, 2010 (Dagstuhl Seminar Proceedings 10271)

[SKG+10] Schupfer, Florian ; Kärgel, Michael ; Grimm, Christoph ; Olbrich, Markus ; Barke, Erich: Towards Abstract Analysis Techniques for Range Based System Simulations. In: *Proceedings of the 2010 Forum on specification & Design Languages, FDL 2010*, 2010, S. 159–164

[SKG+11] Kap. Towards Abstract Analysis Techniques for Range Based System Simulations In: Schupfer, Florian ; Krgel, Michael ; Grimm, Christoph ; Olbrich, Markus ; Barke, Erich: *System Specification and Design Languages: Selected Contributions from Fdl 2012*. Springer New York, 2011, S. 105–121

[SLMW03] Shou, Huahao ; Lin, Hongwei ; Martin, Ralph ; Wang, Guojin: Modified Affine Arithmetic Is More Accurate than Centered Interval Arithmetic or Affine Arithmetic. In: *Martin(Eds.), Lecture Notes in Computer Science 2768, Mathematics of Surfaces, Springer-Verlag*, Springer, 2003, S. 355–365

[SR07] Singhee, A. ; Rutenbar, Rob A.: Statistical Blockade: A Novel Method for Very Fast Monte Carlo Simulation of Rare Circuit Eevents, and its Application. In: *Design, Automation Test in Europe Conference Exhibition, 2007. DATE‘07*, 2007, S. 1–6

[SRWG11] Schupfer, Florian ; Radojicic, Carna ; Wenninger, Joseph G. ; Grimm, Christoph: System Refinement Design Flow based on Semi-Symbolic Simulations. In: *Proceedings of the 10th IEEE AFRICON (2011)*, 2011

[SSRG11] Schupfer, Florian ; Svarc, Markus ; Radojicic, Carna ; Grimm, Christoph: A Range Based System Simulation and Refinement Design Flow. In: *Industry Adoption of the SystemC AMS standard*, 2011

[TD06] Tasiran, Serdar ; Demir, Alper: Smart Monte Carlo for Yield Estimation. In: *Proceedings of the ACM/IEEE TAU*, 2006

[Val98] Kap. The Stae Explosion Problem In: Valmari, Antti: *Lectures on Petri Nets: Basic Models*. Springer Verlag, 1998 (Lecture Notes in Computer Science 1491), S. 429–528

[Vla94] Vladimirescu, Andre: *The SPICE Book*. New York, NY, USA : John Wiley & Sons, Inc, 1994

[WGS90] Wambacq, Piet ; Gielen, Georges ; Sansen, Willy: Symbolic simulation of harmonic distortion in analog integrated circuits with weak nonlinearities. In: *Circuits and Systems, 1990., IEEE International Symposium on* Bd. 1, 1990, S. 536–539

[YS03] Yang, Jin ; Seger, Carl-Johan H.: Introduction to generalized symbolic trajectory evaluation. In: *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 11 (2003), Nr. 3, S. 345–353

Internet References

[1] Cyber-physical systems executive summary. http://precise.seas.upenn.edu/events/iccps11/_doc/CPS-Executive-Summary.pdf. Accessed: 08/29/2012.

[2] Frauenhofer gesellschaft: Systemc ams demonstrator. http://systemc-ams.eas.iis.fraunhofer.de/. Accessed: 05/14/2012.

[3] International technology roadmap for semiconductors: System drivers. /www.itrs.net/Links/2011ITRS/2011Chapters/2011SysDrivers.pdf. Accessed: 02/23/2012.

[4] Isscc 2011 trends report. http://isscc.org/doc/2011/2011_Trends.pdf. Accessed: 02/22/2012.

[5] Libaffa - c++ affine arithmetic library for gnu/linux. www.nongnu.org/libaffa/. Accessed: 07/02/2012.

[6] Linear technology. www.linear.com/designtools/software/. Accessed: 23/02/2012.

[7] Osci. www.accellera.org/downloads/standards/systemc. Accessed: 05/14/2012.

[8] Osci whitepaper. http://publik.tuwien.ac.at/files/PubDat_171466.pdf. Accessed: 05/14/2012.

[9] Systemc ams multi-trace implementation. www.accellera.org/Discussion_Forums/ams_forum/archive/msg?list_name=ams_forum&monthdir=201112&msg=msg00005.html. Accessed: 07/04/2012.

[10] Systemc ams working group. www.systemc-ams.org. Accessed: 05/14/2012.

[11] Updarted affine arithmetic library. http://sourceforge.net/projects/aaflib/. Accessed: 07/02/2012.

Florian Schupfer

Address: Universumstrasse 24/06
1200 Vienna, Austria

Date of birth: 10/13/1974
E-mail: florian.schupfer@gmail.com

Education

PhD 2007/10 - 2013/01
Vienna University of Technology
Faculty of Electrical Engineering, Institute of Computer Technology: Coordinator Univ.Prof.Dr. Christoph Grimm The focus of the work was on extending range based system simulations and the application of formal methods on hardware verification. Project lead of two reserach projects, contribution to lectures and research proposals.

Dipl.-Ing.(FH) 1996/10 - 1999/06
University of applied sciences Technikum Wien
Study in electronics with specialization in telecommunications and medical engineering. Diploma thesis on **"'Simulation of an ADSL transmission system using MATLAB 5.2"'**.

Secondary school for technical studies 1990/09 - 1996/07
Technical school on electronics and informatics

Publications - conferences

- F.Schupfer and C.Grimm. Towards more Dependable Verification of Mixed-Signal Systems. In: *Dagstuhl Seminar Proceedings:Verification over discrete-continuous boundaries*, 2010.
- F.Schupfer, M.Svarc, C.Radojicic and C.Grimm. A Range Based System Simulation and Refinement Design Flow. In:*Industry Adoption of the SystemC AMS standard*, Dresden, 2011.
- F.Schupfer, C.Radojicic, J.G.O.Wenninger and C.Grimm. System Refinement Design Flow based on Semi-Symbolic Simulations. In:*Proceedings of the 10th IEEE Africon (2011)*, Zambia, 2011.
- J.Ou, F.Schupfer and C.Grimm. System Level Communication System Design using Extended SystemC AMS Building Block Library. In:*Tagungsband Austrochip 2011*, pp39-43, 2011.
- J.Ou, F.Schupfer and C.Grimm. Modeling Quantization Error of DSP Systems using SystemC AMS. In:*Proceedings of the VW FEDA 2011*, Southampton,UK,2011.
- C. Radojicic, F. Schupfer, M. Rathmair, C. Grimm, Assertion-based Verification of Signal Processing Systems with Affine Arithmetic, In: *Proceedings of the 2012 forum on specification and design languages (FDL 2012)*, pp. 20 - 26, 2012.
- C. Radojicic, F. Schupfer, M. Rathmair, C. Grimm, Checking Property Specifications of Analog Systems with Affine Arithmetic, In: *Informationstagung Mikroelektronik 2012*, pp. 184-188, 2012.

- M. Rathmair, F. Schupfer, C. Radojicic, C. Grimm, Extended Framework for System Simulation with Affine Arithmetic, In: *Proceedings of the 2012 Forum on specification & Design Languages (FDL 2012)*, pp. 161-168, 2012.
- W. Clifford, J. Glaser, F. Schupfer, J. Haase, C. Grimm, Example-driven Interconnect Synthesis for Heterogeneous Coarse-Grain Reconfigurable Logic, In: *Proceedings of the 2012 Forum on specification & Design Languages*, pp. 170-177, 2012.
- S. Adhikari, F. Schupfer, C. Grimm, Parametric Synthesis and Extraction of Analog-ESL Designs, In: *Proceedings of VW-FEDA*, Southampton, United Kingdom, 2011.
- J.G.O. Wenninger, J. Haase, F. Schupfer, K. Gravogl, C. Grimm, SYCYPHOS: Ein Framework zum Entwurf von Cyber Physical Systems, In: *Zuverlssigkeit und Entwurf / ITG-Fachbericht 231 5.GI/GMM/ITG-Fachtagung*, pp. 140-141, 2011.
- F. Schupfer, J.G.O. Wenninger, C. Grimm: SYCYPHOS-A Framework for Designing Cyber Physical Systems. In: *Industry Adoption of the SystemC AMS Standard (invited)*, Dresden, 2011.
- F. Schupfer, J. Ou, P. Brunmayr, C. Grimm, Performance Guided High Level Algorithm Partitioning, In: *2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Qingdao, 2010.
- F. Schupfer, M. Krgel, C. Grimm, M. Olbrich, E. Barke, Towards Abstract Analysis Techniques for Range Based System Simulations, In: *Proceedings of the 2010 Forum on specification & Design Languages*, pp. 159-164, 2010.
- K. Lee, F. Schupfer, C. Grimm, A range based method for Noise Analysis of mixed A/D Communication Systems, In: *Tagungsband zur Informationstagung Mikroelektronik 10*, pp. 261-266, 2010.
- J. Ou, F. Schupfer, J. Haase, A SystemC Based Methodology for HW/SW Partitioning of an Embedded Computer Vision System, In: *Proceedings of the IASTED International Conference on Modelling, Simulation and Identification*, pp. 133-141, 2009.
- P. Brunmayr, J. Haase, F. Schupfer, Late Hardware/Software Partitioning by using SystemC Functional Models, In: *Proceedings of the 3rd Asia International Conference on Modelling and Simulation (AMS 2009)*, pp. 194-199, 2009.
- C. Grimm, K. Gravogl, F. Schupfer, I. Neumann, The AutoSUN Verification Environment, In: *Proceedings of Specification, Verification and Design Languages (FDL 2008)*, pp. 249-250, 2008.
- C. Radojicic, F. Schupfer, C. Grimm, Semi-symbolic Analysis of Analog and Signal Processing Systems, In: *Presentation: FAC Workshop*, Snowbird, Utah, USA.

Publications - book chapter

- F.Schupfer, M.Kärgel, C.Grimm, M.Olbrich and E.Barke. Towards Abstract Analysis Techniques for Range Based System Simulations. In:*System Specification and Design Languages: Selected Contributions from Fdl 2012.* pp105-121, Springer New York,2011.

Patents

- G.Humer, R.Kloibhofer, R.Lieger, F.Schupfer, G.Steinboeck,"Channel Simulation And Development Platform And Use Thereof", 2006-04-16, Publication number:WO2006026799

Employment

Instiute of Computer Technology 2007/11 - present
Employed as university assistent at Vienna University of Technology

Austrian Research Centers GmbH 2002/12 - 2007/11
FPGA design engineer contributing to several research and industrial projects

VA Tech Elin EBG Traction GmbH 2002/02 - 2002/12
FPGA design and verification engineer contributing to several industrial projects

Siemens AG Austria 2000/07 - 2001/06
ASIC design engineer

Ericsson Austria AG 2000/02 - 2000/06
Project engineer

Austrian Research Centers GmbH 1998/09 - 1999/04
Practical training and diploma thesis

Trainings

2006/09	Flexray workshop	2 day
2006/05	Embedded systems development	1 day
2004/03	Project management - basics	5 day
2004/06	Embedded systems(Power PC)	1 day
2003/12	Processor seminar	1 day
2003/04	Xilinx Xtreme DSP	3 day
2003/03	FPGA solutions	1 day

Interests

Sailing, Mountaineering, Climbing