Jitter Analysis Methods for the Design and Test of High-Speed Serial Links

Distribution Tail Fitting Based on Gaussian
Quantile Normalization

Stefan Erb

Jitter Analysis Methods for the Design and Test of High-Speed Serial Links

Distribution Tail Fitting Based on Gaussian Quantile Normalization

Dipl.-Ing. Stefan Erb

Submitted as thesis to attain the academic degree "Dr. techn."
at the

Graz University of Technology

Institute of Electronics

Supervisor: ....
Univ.-Prof. Dipl.-Ing. Dr. techn. Wolfgang Pribyl

Co-Supervisor: ....
Univ.-Prof. Dipl.-Ing. Dr. techn. Ernst Stadlober

Graz, May 2011

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

. . . .
Date

. . . .
Signature

Abstract

In this thesis analysis methods for serial high-speed interfaces are presented to investigate characteristics and impact of timing uncertainty or jitter. Such methods are widely used for estimating the bit error rate (BER) of serial links, and for signal integrity measurements and compliance testing.

First, an algorithm is developed which determines the Gaussian tail behavior of measured jitter distributions and separates them into random and deterministic components using an efficient optimization scheme. The resulting timing budget allows to accurately quantify the total jitter of clock signals and phase locked loop (PLL) systems. The fast analysis technique allows for an implementation as embedded system and thus, supports a broad variety of design-for-test (DFT) applications such as production testing, on-chip diagnostics and on-line monitoring. One chapter of the thesis is dedicated to these hardware design aspects.

The underlying mathematical principle is based on the Gaussian quantile normalization, which has already been discussed and utilized in previous approaches. A detailed and thorough performance comparison is thus carried out to highlight their different properties. Further, it is demonstrated how the proposed method can easily be generalized for use with arbitrary non-Gaussian tails, as is the case for optical high-speed interconnects.

In a first case study, a fast system level model of a serial high-speed PLL is developed. The system has already been realized as test structure and thus, allows for a direct comparison with measurements. These include the analysis of closed loop phase noise, jitter transfer characteristics as well as the jitter tolerance behavior. Especially the latter two also comprise the use of the developed method. A second case study realizes a BER tester with a 3Gb/s serial high-speed interface on an FPGA board. It demonstrates the practical aspects of jitter measurement, diagnosis and optimization.

Kurzfassung

In dieser Arbeit werden Methoden zur Analyse von Zeit-Jitter vorgestellt, die vor allem in digitalen seriellen Schnittstellen Anwendung finden. Solche Methoden erlauben die Abschätzung der Bitfehlerrate (BER) und werden deshalb häufig für die Beschreibung der Signal-Qualität und zum Nachweis der Einhaltung von Standards eingesetzt.

Im Rahmen der Arbeit wird zunächst eine Methode entwickelt, welche das Gauß-Verhalten an den beiden Enden einer gemessenen Jitter-Verteilung feststellt und die Verteilung daraufhin extrapoliert. Die Methode zeichnet sich vor allem durch ihre Genauigkeit und Effizienz aus. Sie ist daher besonders für die quantitative Analyse der Qualität von Taktsignalen und Phasenregelschleifen (PLLs) geeignet. Aufgrund des einfachen Algorithmus kann sie auch sehr gut zusammen mit Teststrukturen, Diagnosewerkzeugen oder für die Echtzeit-Überwachung solcher Systeme eingesetzt werden.

Das zugrundeliegende mathematische Prinzip basiert auf der Quantil-Normalisierung von Gauß-Verteilungen, welche bereits in ähnlichen Ansätzen verwendet wurde. Deshalb umfasst die Arbeit auch einen detaillierten und umfangreichen Vergleich verschiedener Methoden, der die jeweiligen Eigenschaften aufzeigt. Ein weiterer Abschnitt ergänzt die vorgestellte Methode um ein generalisiertes Prinzip, mit dem sie auf einfache Weise auf beliebige nicht-Gauß Verteilungen angewandt werden kann.

In einer ersten Fallstudie wird ein sehr schnelles Modell für eine PLL erstellt. Dieses System existiert bereits als Teststruktur, und ermöglicht daher einen direkten Vergleich der erhaltenen Simulationsergebnisse mit Messdaten. Der Vergleich umfasst eine Rauschanalyse der PLL, die Simulation der Jitter-Transfer-Funktion sowie des Jittertoleranz-Verhaltens. Die Simulation der letzten beiden Eigenschaften erfolgt dabei mit Hilfe der neuen Analysemethode. Eine zweite Fallstudie realisiert einen BER-Tester zusammen mit einer 3Gb/s S-ATA Schnittstelle auf einem FPGA Board. Sie dient vor allem der Darstellung praktischer Aspekte bei der Diagnose und Optimierung von Jitter.

Danksagung

Ich möchte mich bei allen bedanken die mich in den letzten dreieinhalb Jahren bei der Umsetzung dieser Dissertation unterstüzt haben.

Allen voran, mein Betreuer und Mentor, Prof. Wolfgang Pribyl, der mir bei meinem Forschungsthema großen Freiraum gelassen, und mich in meinen Interessen und Anliegen sehr stark unterstützt hat. Diese Rahmenbedingungen haben das erfolgreiche Abschließen der Dissertation erst möglich gemacht und mitunter ganz wesentlich zur Qualität der vorliegenden Arbeit beigetragen. Ich empfinde den erfolgreichen Abschluss daher als großes Geschenk, das uns hoffentlich noch viele Jahre über diese Arbeit hinaus verbindet.

Ein weiterer großer Dank gebührt meinem Zweitbegutachter, Prof. Ernst Stadlober, der mir in zahlreichen Diskussionen die wesentlichen Aspekte der Extremwerttheorie für mein Thema bereitwillig nähergebracht hat. Ausgehend von der Modellierung elektronischer Systeme hat meine Arbeit dadurch einen Streifzug durch die Welt der Statistik unternehmen können, den ich für mich persönlich als sehr bereichernd empfinde.

Für die fachliche Unterstützung danke ich vor allem den Ingenieuren des ‘Analog Design Support’ Teams der Infineon Villach, die mir bei der Erstellung des Modells für serielle high-speed Schnittstellen geholfen und die Messdaten der Teststruktur zur Verfügung gestellt haben. Reinhard Steiner hat mir immer wieder gern Auskunft erteilt sowie den Zugang zu den Hochleistungsrechnern über die gesamte Dauer der Dissertation ermöglicht.

Nicht zuletzt verdanke ich auch meinen Kollegen aus dem Dissertantenlabor zahlreiche nützliche Hinweise und Anregungen. Allen voran Christoph Böhm, der mir immer wieder gerne und manchmal auch sehr kurzfristig meine Publikationen verbessert hat.

Ich möchte an dieser Stelle auch noch Siegfried Rainer und Jan Ranglack erwähnen, für deren Freundschaft ich sehr dankbar bin. Danke William Robinson, für die Englisch-Korrekturen.

STEFAN ERB
GRAZ, MAI 2011

The real voyage of discovery consists not in seeking new landscapes
but in having new eyes.

MARCEL PROUST

There is nothing more practical than a good theory.

KURT LEWIN

Das Runde muss ins Eckige.

SEPP HERBERGER

Contents

List of Figures

List of Tables

List of Abbreviations

BB-PD	Bang-Bang Phase Detector
BER	Bit Error Rate
BERT	Bit Error Rate Tester
BIJM	Built-In Jitter Measurement
BUJ	Bounded Uncorrelated Jitter
CDF	Cumulative Density Function
CDR	Clock and Data Recovery
CP	Charge-Pump
CPLL	Charge-Pump Phase-Locked Loop
DDJ	Data Dependent Jitter
DFE	Decision Feedback Equalizer
DJ	Deterministic Jitter
DNL	Differential nonlinearity
EQ	Equalizer
FPGA	Field Programmable Gate Array
GGD	Generalized Gaussian Distribution
Gb/s	Gigabit per second
INL	Integral nonlinearity
IQR	Interquartile Range
ISI	Inter-Symbol Interference
JTOL	Jitter Tolerance
LF	Loop Filter
MP	Microprocessor
PD	Phase Detector
PDF	Probability Density Function
PLL	Phase-Locked Loop
PRBS	Pseudo Random Binary Sequence
PSD	Power Spectral Density
Q	Quantile
QN	Quantile Normalization
RJ	Random Jitter
RMS	Root Mean Square
RX	Receiver
S-ATA	Serial Advanced Technology Attachment (an interface standard)
SJ	Sinusoidal/Periodic Jitter
sQN	Scaled Quantile Normalization
SUT	System Under Test
TIA	Time Interval Analyzer
TJ	Total Jitter
TX	Transmitter
UI	Unit Interval
VCO	Voltage Controlled Oscillator

Nomenclature

α	Shape parameter of generalized Gaussian distributions, see figure 7.3
α_{RJ}	Shape parameter of generalized Gaussian random jitter source, see figure 7.6
Δk	Distance factor for logarithmic search grid, see figure 4.8
ΔP_t	Minimum probability interval for tail fitting, see figure 4.21
ΔT_t	Minimum time interval for tail fitting, see figure 4.21
ϵ_{conf}	Confidence threshold for convergence of jitter tolerance algorithm, see figure 9.2
$\hat{\sigma}_{err}$	Standard error of linear regression, see equation (4.5)
$\hat{c}_{1.2}$	Algorithmic scenario for tail fitting based on ΔP_t parameter, see section 4.3.2
$\hat{n}$	Fitness measure based on regression length, see equation (4.14)
$\hat{n}_T$	Fitness measure based on regression length of $\hat{T}$, see equation (4.16)
$\hat{T}$	Fitness measure based on regression error and slope, see equation (4.15)
κ	Kurtosis
IQR	Interquartile range of extrapolation error or error spread, see equation (3.17)
BER_{spec}	Target bit error rate level, specification requirement
DJ_{pp}	Deterministic jitter peak-to-peak value after tail fitting, see equation (3.13)
RJ_{rms}	Standard deviation of Gaussian tails after tail fitting, see equation (3.13)
$\mu_{L,R}$	Mean value of left/right fitted Gaussian model, see figure 2.11
ν	Learning rate parameter for adaptive recursion, see equation (9.1)
σ_e	Standard deviation of extrapolation error, see equation (3.16)
$\sigma_{abs/per/lt/max}$	RMS value of absolute/period/long term/maximum jitter, see equation (8.15)
σ_{acc}	RMS value of accumulated jitter, see equation (8.14)
σ_{DNL}	Standard deviation to model differential nonlinearity error, see section 5.4.3
$\sigma_{L,R}$	Standard deviation of left/right fitted Gaussian model, see figure 2.11
σ_{RJ}	Standard deviation of random jitter component of test distributions, see figure 3.5
TJ_{pp}	Total jitter peak-to-peak value or timing budget, see equation (3.12)
ξ	Skewness
a	Statistical confidence level, see figure 3.4
A_{DJ}	Amplitude of deterministic jitter component of test distributions, see figure 3.5
$A_{L,R}$	Amplitude of left/right fitted Gaussian model, see figure 2.11
A_{SJ}	Sinusoidal jitter amplitude
E_L	Estimation loss, see equation (3.18)
E_m	Mean value of extrapolation error, see equation (3.16)
$E_{med,\alpha}$	Median value of tail shape error, see equation (7.10)
E_{med}	Median value of extrapolation error, or error bias, see equation (3.17)
f_{SJ}	Sinusoidal jitter frequency
I_{max}	Maximum number of iterations of jitter tolerance algorithm
K	Number of evaluations
k	Amplitude scaling factor, see figure 4.3
N	Number of jitter samples in a distribution, sample size
n	Fitted tail length, see equation (4.5)
$N_{min/max}$	Minimum/Maximum sample size of jitter tolerance algorithm, see figure 9.2
o	Offset of linear function

p	Probability
$P_{t,min}$	Minimum probability threshold for tail fitting, see figure 4.21
q	Quantile
Q_{min}	Minimum Q-domain threshold for tail fitting, see figure 4.27
$Q_{th,c}$	Algorithmic scenario for tail fitting based on Q_{min} threshold, see section 4.4.2
R	Number of bins per unit interval
s	Slope of linear function
T	Bit period, if normalized to the unit interval T=1 UI
$T(f_{SJ})$	Jitter transfer function
t_{eye}	Eye opening
$t_{L,R}$	Left/Right timing budget of jitter, see figure 2.10
x	Timing jitter amplitude

1. Introduction

A brief overview to the topic of jitter and bit error rate (BER) analysis in high-speed communications is given. The problem domain is presented and an outline to the overall document is given, together with a list of scientific contributions which have been elaborated throughout the work.

1.1. Motivation and Problem Domain

With the increasing demand on higher clock frequencies for synchronization and data transmission, timing uncertainty, or jitter has become a major limiting factor for today's high-speed communications. This is especially true for serial data transmission which utilizes phase locked loop (PLL) based circuits for synchronization. In this case, limited jitter tolerance and inherent phase noise can lead to erroneous data recovery, and hence, force the need for an accurate quantification of timing jitter and associated effects.

Figure 1.1 shows the typical block diagram of a serial high-speed interface, composed of the three main blocks, transmitter, channel and receiver [49,114]. The transmit (TX) buffer is triggered by a high-speed PLL, running at the desired transmission rate of several Gigabit per second (Gb/s). The channel or transmission path is the major contributor to signal degradation. It limits data throughput by introducing noise and signal distortions, both leading to timing jitter. The receiver (RX) also consists of a high-speed PLL, realized as a clock and data recovery (CDR) unit to synchronize with the received bit stream. The decision block can be a simple data latch using the recovered clock and the analog input signal.

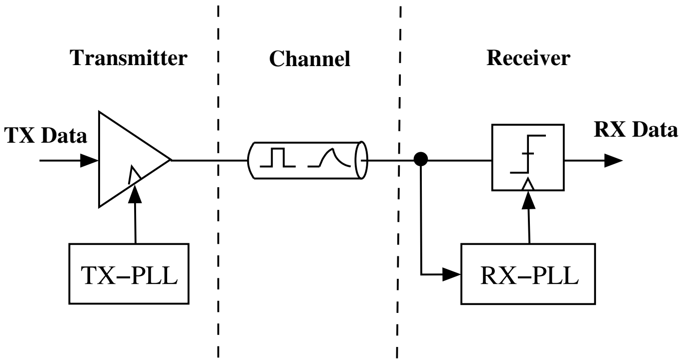


FIGURE 1.1.: Block scheme of a serial high-speed transceiver.

Serial high-speed interfaces usually transmit data without a dedicated clock signal. A special encoding guarantees for sufficient bit transitions inside the data stream in order to correctly synchronize the receiver clock with the input data. The RX-PLL as a synchronization circuit has to cope with input jitter and provide a certain robustness against timing variations. If jitter exceeds a critical amplitude, the PLL will not be able to track timing variations correctly, and hence, lead to erroneous signal recovery or misinterpretation of the received data.

Serial interface standards, such as S-ATA [118], specify stringent requirements on the error probability, or bit error rate (BER) of a recovered data stream. The BER directly reflects the influence of timing jitter on system performance, and is thus the best suited figure of merit to indicate the quality of a digital communication interface. In fact, one likes to describe a relation between BER and measured jitter to thoroughly investigate its influences and to afford identification of possible root causes.

Since jitter is basically a random process, its analysis involves the use of statistical analysis methods. These are applied to a set of collected jitter samples and try to accurately determine the impact of timing jitter on the investigated system. In this thesis such jitter analysis methods are developed, analyzed, compared and applied to practical simulations and measurement cases. The subsequent section describes the key contributions and results which have been achieved throughout this work. It is followed by a brief overview of the overall document.

1.2. Contributions

Several key contributions and results of this work extend the state of knowledge, and are summarized below. A more detailed description of the key results can also be found in the conclusions section at the end of the thesis, together with a list of own publications [C1-C9].

First, a novel method (here denoted as scaled Q-normalization, sQN) for jitter and BER analysis is developed. It is based on the Gaussian quantile normalization principle, where the three parameters amplitude, mean and standard deviation of Gaussian model functions are identified and fitted into the tails of a jitter distribution. This allows them to be extrapolated down to any desired probability level. The method is realized with a flexible and efficient optimization scheme, and allows for fast tail fitting combined with accurate extrapolation results. Its performance is analyzed with respect to the extrapolation error, which is shown to highly depend on the sample size and the shape of test distributions. From the basic concept, two algorithmic approaches with conservative fitting parameters are derived and optimized, in order to improve the error behavior with respect to accuracy and outlier suppression. For typical test distributions (uniform combined with Gaussian) and a number of jitter samples $N{=}10^6$ (tail extrapolations range over six orders of magnitude, down to the 10^{-12} level), the estimated jitter budget has an error bias $<2\%$ and an overall error $<3\%$ in more than 97.5% of the cases. The method is partly published in [C1,C8,C9].

Another contribution of this thesis focuses on hardware design aspects, to utilize the proposed sQN method together with real jitter measurement systems. Their limited precision causes quantization effects and introduces additional extrapolation error which must be considered and dealt with. First, requirements with respect to minimum tail amplitude and time resolution of measured distributions are investigated. Corresponding equations are derived to guarantee these requirements. Then, the combined influence of limited sample size and time resolution on the accuracy of the sQN method is described in terms of empirical relations that quantify error bias, statistical spread and the combination of both. These relations aid a system designer in finding an optimum performance trade-off between desired accuracy of analysis and hardware expense. Finally, also the effect of process variations in measurement systems is added to these empirical relations, in order to quantify their influence. Two typical design examples act as design guidelines to realize hardware jitter measurements with a certain target accuracy, when using the proposed sQN method. The described contributions are also published in [C4,C8].

Further, a comprehensive performance comparison of different jitter analysis methods based on the quantile normalization principle is provided. Therefore, the sQN method is compared with the known conventional quantile normalization (QN) method, as well as higher order polynomial methods (QP2, QP3, QP4). The idea is to provide a detailed performance reference for comparison with future jitter analysis methods. As a fundamental result, in simulators the QN method highlights the same beneficial property of a strictly positive error bias as the sQN method. This is a clear advantage compared to higher order polynomial methods (QP2, QP3, QP4), which achieve acceptable accuracy only for certain test distributions. A comprehensive comparison with the sQN method is carried out, and clearly shows that sQN achieves the best performance. However, this is at the cost of a larger computational demand. Although the QN method is less accurate, it offers a significant speed-up (≈ 35 times compared to sQN) for tail fitting. This complementary property demands an additional error analysis with QN when used together with hardware measurements. This allows a system designer to choose between the better suited algorithmic alternative. Equivalent to the sQN method, an error analysis is thus carried out for the QN method, which quantifies its extrapolation error in terms of the two key parameters sample size and time resolution, as required for hardware jitter measurements. In addition, the previously described design examples for hardware measurements are extended for use with the conventional QN method. Therefore, the design equations for the sQN method can simply be reused. Obtained results highlight that also for hardware measurements the QN method is generally less accurate than sQN, but is also less affected by differential non-linearity error, as caused by process variations. These results are partly published in [C3,C8].

A dedicated part of the thesis also focuses on the generalization of the sQN principle for use with arbitrary non-Gaussian tails. Such tails may for example appear with amplitude distributions of high-speed optical links. As a function class, the generalized Gaussian distribution (GGD) is used for tail fitting. The presented principle is fully consistent with the existing jitter decomposition model and thus, forms a logical extension. In simulators, it is able to identify the exponential tail characteristic of a distribution and to extrapolate tails with acceptable accuracy. However, the estimated tail characteristic suffers from large errors if tails decay very fast or in hardware systems with very limited time resolution. Further, with its large computational demand the primary use cases are simulations. The generalized principle is also presented in [C5].

Several case studies describe typical application fields for the proposed sQN method. First, a fast behavioral model for charge-pump PLLs is implemented, which is based on an exact solution for the 2nd order loop filter, and includes a parasitic gain regulator pole as well as an oscillator noise model [C2]. It is able to simulate approximately 10^6 bit periods within one minute on an Intel 2.2GHz laptop and thus, allows for in-depth system exploration as well as statistical jitter analysis. Jitter transfer functions and phase noise spectra of the modeled PLL are compared with measurements from an existing hardware structure and show excellent agreement. The sQN method is here used to simulate and verify the measured jitter transfer functions. Therefore it extracts the deterministic jitter component from collected distributions. For additional comparison, also a spectral analysis method is used.

As a second application, the sQN method is used for identifying jitter tolerance curves of the previous high-speed PLL model. Therefore, external jitter is injected to the PLL and adjusted until a desired error probability is obtained. In order to solve this inverse problem, an adaptive search algorithm is developed, which highly reduces the number of required jitter samples [C6]. Comparative results show, that the included sample size adaptation makes the recursive search 2-3 times faster. Results also highlight, that the proposed algorithm can be used for both simulations and hardware measurements.

In a final case study, a practical jitter measurement and analysis system for the diagnosis and optimization of transmission lines, PLLs and transceiver structures is developed [C7]. The target architecture is a high-speed FPGA, and as an example, various jitter measurements are carried out with RG-58 coaxial cables as well as a standard 1m S-ATA cable. Optimizations are performed with the FPGA internal transceiver settings and equalizer structures, which allow to reduce the total jitter of a 5m test cable significantly (up to 28%). In concluding analyses, also the extrapolation error of sQN and QN fitting methods is investigated and compared against theoretical worst case errors from previous numerical analyses. In this context the different DJ shapes are experimentally confirmed to be well suited for estimating the error of pure PLL jitter, ISI, and ISI plus additional noise affected channels.

1.3. Thesis Overview

An introduction to jitter analysis in high-speed serial links is given in the subsequent chapter 2. Fundamentals of jitter, noise and BER analysis in serial interfaces are described, together with an overview to the state-of-the-art in the field. Further, high-speed PLLs are briefly discussed as fundamental circuits for data transmission and recovery, and a thorough survey of jitter analysis methods is included.

Chapter 3 provides the mathematical basics involved with Gaussian quantile normalization, which forms the underlying key principle for jitter analysis methods investigated throughout this thesis. Measures for statistical performance evaluation and a set of suitable test distributions are defined as well.

In chapter 4 the scaled Q-normalization method is realized as an accurate and efficient method for jitter analysis. Subsequent evaluations demonstrate excellent performance, which is further improved by optimizing the algorithmic behavior with conservative fitting parameters. From the fundamental optimization scheme, two algorithmic principles are derived which achieve similar performance in a simulator environment.

In chapter 5 hardware design aspects are investigated, in order to utilize the developed method together with jitter measurement devices or built-in test structures. Since these hardware systems suffer from limited precision and additional error effects, a design trade-off between hardware expense, measurement accuracy and analysis speed is required. In this context, several design equations and empirical relations are derived to characterize the key parameters for a hardware measurement system.

The performance of different jitter analysis methods are compared in chapter 6. First, a unifying optimization scheme is derived to highlight the conceptual difference between the proposed method from chapter 4 and others which are also based on the Gaussian quantile normalization. A thorough performance analysis and comparison of each method is then carried out.

Chapter 7 generalizes the quantile normalization principle for use with arbitrary non-Gaussian tails. This allows the proposed method to be applied to generic analysis scenarios where the Gaussian tail assumption does not hold anymore, as can be the case with amplitude noise in optical fiber interconnects.

In chapter 8 a fast system level model of a serial high-speed PLL is developed, which has been realized as test structure for a 3Gb/s S-ATA interface. The event-driven model represents an enhanced version of a prior approach. It serves as a case study where the jitter analysis method from chapter 4 affords derivation and comparison of various jitter transfer functions.

A fast method for identifying the jitter tolerance curve of high-speed PLLs is introduced in chapter 9. A recursive algorithm determines the tolerance behavior, and adaptively adjusts the sample size of collected jitter distributions to minimize the required test time. The method is applied to the PLL model from the previous chapter and simulation results are compared against measured tolerance curves. Such simulations are particularly useful for indicating the robustness of a PLL against input jitter.

Chapter 10 describes a diagnostic tool for jitter analysis, which is implemented on an FPGA. It estimates the total jitter caused by a system under test and thus, can be used for testing the quality of transmission channels and for optimizing the parameter configuration of interface structures. Further it experimentally confirms the error behavior of the developed fitting method.

Finally, chapter 11 summarizes the contents of this thesis and concludes with a brief outlook to future research directions.

2. Fundamentals of Jitter, PLLs and BER Analysis

In this chapter an introduction to the basics of jitter and PLLs in high-speed serial links is given. After a brief overview to the sources of timing jitter in communication systems, different types and definitions for clock jitter and phase noise are discussed. The fundamental principle of PLLs for high-speed data transmission is explained, followed by a comprehensive overview of the state-of-the-art of jitter analysis methods. The main focus is on analysis techniques that relate jitter with the bit error rate (BER), as required for testing high-speed serial links.

2.1. Jitter in High-Speed Serial Links

Dealing with jitter plays a crucial role in the design of digital high-speed interfaces. Since timing uncertainty is the major cause for erroneous data recovery, a robust receiver architecture is one of the most challenging design criteria. Serial high-speed standards impose heavy requirements on the allowed bit error rate (BER), and specify typical target BERs of 10^{-12} or even less [43, 118].

Figure 2.1 shows different jitter sources in a serial interface which contribute to the overall accumulated jitter at the receiver. A certain amount of jitter is already introduced by the non-ideal clock synthesizer inside the transmitter structure. Depending on the quality of the channel, inter symbol interference (ISI) as well as reflections and crosstalk may strongly degrade the signal integrity along the transmission path. Finally also at the receiver side, a non-ideal equalizer (EQ) and PLL inherent phase noise of the clock and data recovery (CDR) unit will additionally provide their own timing jitter [3, 6, 8, 11, 15, 45, 53, 82, 83, 127].

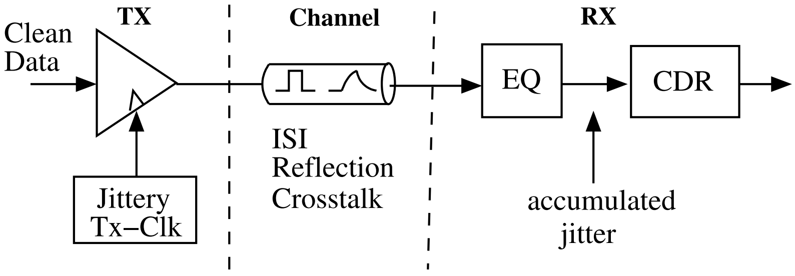


FIGURE 2.1.: Jitter sources in a serial high-speed interface [5].

A common way to highlight the problem of signal recovery and presence of jitter is the eye diagram, as depicted in figure 2.2. It shows the received analog data signal folded in time, with the bit period referred to as unit interval (UI). The untreated received eye is often almost closed and has to be reopened with equalization techniques or signaling schemes that try to compensate the channel influence. Timing jitter especially degrades system performance when causing a large horizontal eye closure, since signal transitions spread over the entire bit interval and thus, impede the recovery circuit to synchronize with the data. Inside the plotted waveform one may thus define an eye mask [43, 47] which must not be violated in order to meet signal quality requirements.

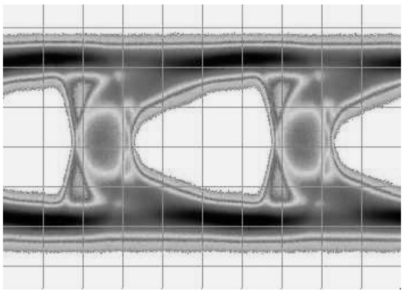


Figure 2.2.: Typical receiver eye diagram affected by jitter.

Considering the timing budget or horizontal eye closure at the optimum decision threshold in figure 2.2, simple probability distributions may be used for jitter analysis. Timing jitter is then described as a statistical signal in terms of different components [82,104]. As shown in the scheme in figure 2.3, an observed total jitter (TJ) distribution can basically be decomposed into a bounded deterministic (DJ) and an unbounded random (RJ) part. Both components relate to independent time-domain random processes and thus, appear as convolved in histogram domain.

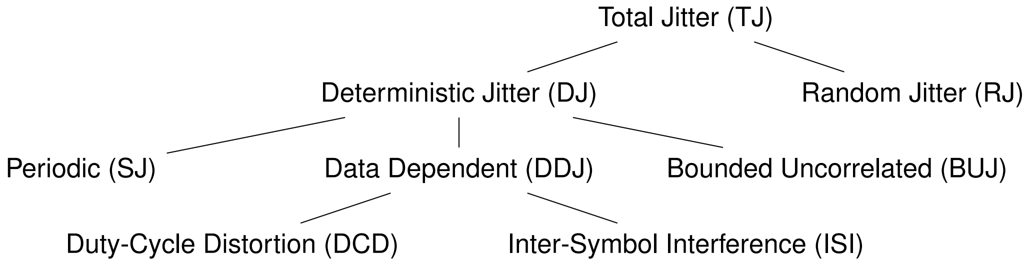

Total Jitter (TJ)
Deterministic Jitter (DJ)
Random Jitter (RJ)
Periodic (SJ)
Data Dependent (DDJ)
Bounded Uncorrelated (BUJ)
Duty-Cycle Distortion (DCD)
Inter-Symbol Interference (ISI)

Figure 2.3.: Jitter components according to [82, 104].

Random jitter is usually considered as Gaussian, but can basically follow any unbounded probability behavior. It is observed at both distribution tails, extending them toward infinity. Bounded deterministic jitter can be of arbitrary shape, and is expressed in terms of various subcomponents in order to investigate and distinguish various root causes. DJ is further subdivided into sinusoidal or periodic (SJ), bounded uncorrelated (BUJ), and data-dependent jitter (DDJ). Use of generated SJ is especially important for jitter tolerance testing [43] and for the measurement of jitter transfer functions [46, 121] in PLLs. Sometimes SJ also appears as an effect of parasitic spurs or power supply noise. BUJ is mainly caused by couplings, such as crosstalk from adjacent transmission lines, digital switching logic or ground bounce effects. BUJ is always considered bounded because of the limited coupling strength. Exact models are difficult to derive for this component since both coupling signals and mechanisms are highly variable. Finally, DDJ is a jitter component which can be related to the transmitted data pattern. Duty-cycle distortion (DCD) is caused by a difference in the pulse width between logical high and low levels and ranges from voltage offsets or different rise and fall times at signal transitions. Inter symbol interference (ISI), appears when the channel impulse response extends over several bit periods. As shown in figure 2.4, a single transmitted pulse is spread in time along the channel, and thus overlaps and influences adjacent symbols. This causes a significant error in timing recovery. Fortunately, DDJ influences can be fully compensated with an equalizer (EQ) if the channel is characterized by its impulse response, or with an adaptive decision feedback equalizer (DFE) if it is unknown [5]. The use of EQs is restricted on compensating DDJ, other jitter components which propagate through the transmission path are then dealt by the receiver PLL.

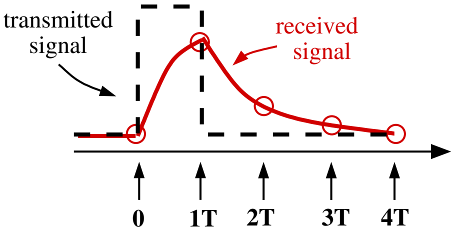


FIGURE 2.4.: Inter-Symbol-Interference caused by a transmission channel [5].

In order to provide a fundamental understanding of the underlying research field, first the mathematical description of timing jitter and phase noise will be reviewed. In this context, jitter is seen as time domain representation of phase noise, which describes the spectral purity of an oscillator. Further, as a random process, phase noise has to be described in terms of statistical measures such as variance and power spectral density (PSD). In the following sections thus, mathematical relations for these measures will be derived. These definitions and derivations are very common in PLL literature and can for example be found in [30, 36, 42, 121].

2.1.1. Phase Noise Definition

We start the mathematical analysis of phase noise with the output signal $v(t)$ of a non-ideal oscillator, which is influenced by the phase noise $\phi(t)$ as time-domain random process [30, 42].

$$v(t) = A\cos(\omega_0 t + \phi(t)) \tag{2.1}$$

This phase modulated signal can be decomposed in terms of Bessel functions. If the phase variation of the noise signal is small compared to the reference period of the ideal oscillator $|\phi(t)| \ll 1\text{rad}$, equation (2.1) can be rewritten as:

$$v(t) \approx A\cos(\omega_0 t) - A\phi(t)\sin(\omega_0 t) \tag{2.2}$$

The output spectrum of this signal consists of two Dirac impulses located at the carrier frequency $\omega = \pm\omega_0$ together with the frequency translated spectrum of $\phi(t)$. If $\phi(t)$ is considered as a stationary random process, its phase noise spectrum $S_\phi(\Delta\omega)$ can be calculated by the Fourier transform of the auto-correlation function. Here the variable $\Delta\omega$ is used to denote the frequency offset from the carrier ω_0, as shown in figure 2.5, and $S_\phi(\Delta\omega)$ is referred to as double-sideband PSD, which contains the spectral power of both sidebands of the oscillator spectrum. The phase noise is often quantified in terms of a single-sided spectral noise density $L\{\Delta\omega\}$. This is the noise density

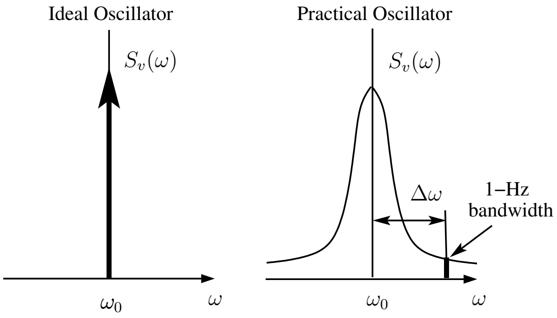


FIGURE 2.5.: Ideal and real oscillator spectrum [42].

measured at a frequency offset $\Delta\omega$ from the carrier and is therefore one-half of $S_\phi(\Delta\omega)$. $L\{\Delta\omega\}$ additionally refers to the carrier power and is given in units of $\mathrm{dBc/Hz}$:

$$L\{\Delta\omega\} = \frac{S_\phi(\Delta\omega)}{2} = 10\log\left(\frac{\text{noise power in 1 Hz BW at } \omega_0 + \Delta\omega}{\text{carrier power}}\right) \; [\mathrm{dBc/Hz}] \tag{2.3}$$

For a detailed description of the phase noise PSD also refer to [42, 78] or [30, chapter 11].

A typical phase noise spectrum for a voltage controlled oscillator (VCO) in high-speed PLLs is given in figure 2.6. This spectrum is also known as Leeson process and consists of three distinct noise regions. The $1/\omega^3$ and $1/\omega^2$ terms represent flicker and thermal noise of electronic components inside the VCO. These noise regions are integrated due to the frequency-to-phase conversion, which corresponds to a multiplication of $1/\omega^2$ in spectral domain. The $1/\omega^0$ phase noise floor is caused by external components and is not affected by the integration process.

The VCO noise model is usually specified using four parameters: the flicker noise corner frequency f_{fl}, a measured phase amplitude A_1 with corresponding frequency f_1 located in the $1/\omega^2$ region, and the noise floor amplitude A_{PhN}. This phase noise model can also be realized in time domain [122].

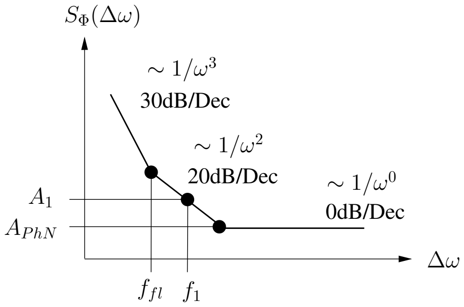


Figure 2.6.: Typical oscillator phase noise spectrum [42].

2.1.2. Clock Jitter Definition

Clock jitter is considered as the phase noise behavior in time domain, which causes the timing displacement of a digital clock signal. This means, as a random process it can only be observed at the edges of a clock signal. According to figure 2.7 we distinguish between various definitions of clock jitter [15, 22, 23, 42, 74, 119, 142].

Absolute Jitter $j_{abs,k}$ is the time difference of the k-th clock edge measured between an ideal ($t_{id,k}$) and a non-ideal (t_k) clock signal:

$$j_{abs,k} = t_k - t_{id,k} \tag{2.4}$$

Period Jitter $j_{per,k}$ is defined as the time variation of the clock period. It is the time difference between k-th clock period T_k and the ideal period $\overline{T}$:

$$\begin{aligned} j_{per,k} &= T_k - \overline{T} \\ &= (t_{id,k} - j_{abs,k}) - (t_{id,k-1} - j_{abs,k-1}) - \overline{T} \\ &= j_{abs,k-1} - j_{abs,k} + t_{id,k} - t_{id,k-1} - \overline{T} \\ \Rightarrow j_{per,k} &= j_{abs,k-1} - j_{abs,k} \end{aligned} \tag{2.5}$$

Accumulated Jitter $j_{acc,k}^{(m)}$ is defined similar to period jitter, besides that the time displacement of a non-ideal clock is measured m periods after the reference clock edge. According to this definition we have $j_{per,k}{=}j_{acc,k}^{(1)}$.

$$\begin{aligned} j_{acc,k}^{(m)} &= T_k^{(m)} - \overline{T^{(m)}} \\ \Rightarrow j_{acc,k}^{(m)} &= j_{abs,k-m} - j_{abs,k} \end{aligned} \tag{2.6}$$

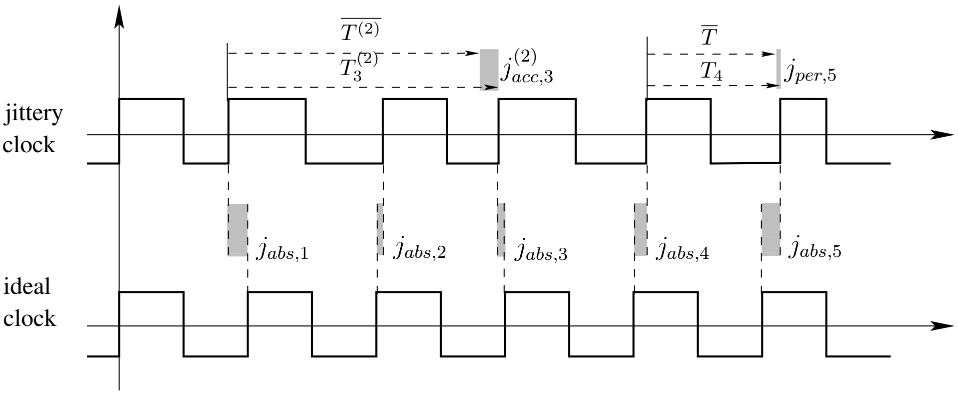


Figure 2.7.: Definitions of absolute, period and accumulated jitter [99].

Input-Output Jitter The most common method for analyzing the performance of a PLL is to measure the time difference between input reference frequency and the PLL output clock. This allows for directly quantifying the time domain misalignment of the PLL, which yields a qualitative description for synchronization performance.

If the PLL is used for clock and data recovery (CDR) as with serial high-speed receivers, the reference frequency is replaced by the analog input data and jitter values are measured between bit transitions of the input signal and the PLL output clock (see figure 2.8). In order to correctly quantify IO jitter, thus, an exact time interval measurement has to be performed.

IO jitter measurements are very important for practical use in high-speed communications and required by a broad variety of applications, such as production tests, clock synchronization, or signal quality specification [82].

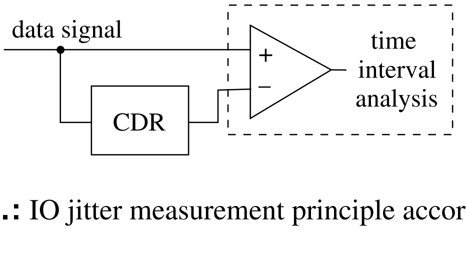


Figure 2.8.: IO jitter measurement principle according to [82].

2.1.3. Relation Between Phase Noise and Jitter

We are still missing a relation between the phase noise $\phi(t)$ as continuous random process, and jitter for digital clock signals in time domain. The phase modulated signal in equation (2.1) can also be seen as a sampling clock, where the zero crossings correspond to the edges of a digital clock signal. For the ideal case we have $\phi(t){=}0$, and the sampling instants correspond to $\{0, T_0, 2T_0, \ldots, kT_0\}$. The non-ideal sampling instants are affected by absolute jitter, and thus

$\{j_{abs,0}, T_0 + j_{abs,1}, 2T_0 + j_{abs,2}, \ldots, kT_0 + j_{abs,k}\}$. Therefore, the k-th phase deviation caused by jitter appears at time instant $t_k{=}kT_0 + j_{abs,k}$, so that

$$j_{abs,k} \cdot \omega_0 = \phi(kT_0 + j_{abs,k}). \tag{2.7}$$

In addition, if the absolute jitter is significantly smaller than one sampling period ($j_{abs}{\ll}T_0$), we have $\phi(kT_0 + j_{abs,k}){\approx}\phi(kT_0)$, and can rewrite equation (2.7) as

$$j_{abs,k} \approx \frac{\phi(kT_0)}{\omega_0}, \tag{2.8}$$

where $j_{abs,k}$ is now a discrete time random process, which simply corresponds to a sampled and scaled version of the continuous phase noise process $\phi(t)$.

2.2. PLLs for Serial High-Speed Communications

In this section a brief introduction to high-speed PLLs is given, as required for serial communication interfaces. PLLs are non-linear synchronization systems that have been investigated and described thoroughly in literature [7,30,36,92,121]. In high-speed serial links they are commonly used as clock synthesizers at the transmitter, and for clock and data recovery (CDR) at the receiver side [11, 45, 46, 134]. A classical digital PLL is composed of three basic components: a phase detector (PD), a loop filter (LF) and a voltage controlled oscillator (VCO). The charge-pump PLL (CPLL) architecture also omits the divider along the feedback path and includes an additional charge-pump (CP), in order to achieve a simple high-speed design with low phase noise. The basic block scheme is depicted in figure 2.9.

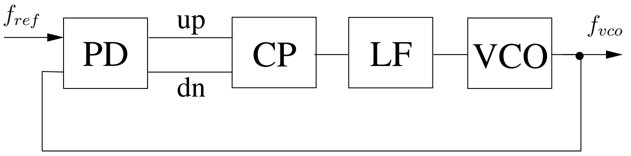


Figure 2.9.: Basic block scheme of a CPLL [121].

The behavior can be summarized as follows: The VCO generates an output clock with frequency f_{vco}, which depends on the given input voltage. The phase detector compares the phase of this clock against a reference frequency f_{ref}, and decides whether f_{ref} is preceded (early) or pursued (late) by f_{vco}. According to this decision, logical down (early) or up (late) pulses are generated. Both signals drive a charge-pump which injects or unloads current into the loop filter, and thus provides the control voltage for the oscillator. If the oscillator clock is late, several up pulses are generated by the PD which increases the loop filter voltage and thus, moves the oscillator toward higher frequency where both phases are again aligned. For an early oscillator clock the reverse behavior is observed. This way a non-linear control loop acts as synchronization system.

In the past years, charge-pump PLL architectures have dominated the field of high-speed transceivers due to a low phase noise. Although all-digital PLLs [25, 110] are becoming increasingly important with technology scaling and for design cost reduction, CPLLs are still widely used. They offer two major advantages compared to pure analog architectures. First, a flexible design can be achieved with decoupled parameters such as the loop bandwidth, damping factor and lock range. Second, the included charge-pump allows for a zero static phase offset [46, 133].

In serial high-speed links both transmitter and receiver are characterized by PLLs as clock synthesizers. The transmitter often uses an additional clock divider in the PLL feedback path to multiply the reference frequency in order to yield the desired high-speed data rate. Conversely, the receiver is characterized by a clock and data recovery (CDR) circuit, where the serial input data is used as reference frequency and the VCO output is the synchronized clock. The PD can determine a phase mismatch only at input signal changes, which requires a sufficient amount of bit transitions inside the data stream. Therefore, typically the 8b10b encoding scheme [43] is used, which converts 8 bit of original data into 10 bit for transmission. This encoding guarantees for sufficient bit transitions with a maximum of four consecutive equal bits, and a DC-balanced signal.

CPLLs have been analyzed and described thoroughly in literature, where the theory has been extended from the linear model of analog PLLs [36]. A valid continuous-time approximation is obtained, if the loop bandwidth is considered significantly smaller (at least 1/10) than the update frequency of the phase detector. Since high-frequency signals are suppressed by the loop filter, digital pulses of the phase detector are averaged and thus, a linear s-domain model can be used for a CPLL design. At higher frequencies where the PD update rate is comparable to the loop bandwidth, the feedback delay will induce an excessive phase shift and hence, lead to instability. In order to account for this effect, discrete-time z-domain equations have been derived as well [46]. Unfortunately, these analytical equations still do not provide an accurate description of the non-linear phase noise behavior inside a CPLL. Only behavioral time domain models that are able to cope with the non-linear loop dynamics of a CPLL thus correctly reflect the true phase noise of high-speed transceivers. Such a model will be implemented in chapter 8 to demonstrate the application of a proposed jitter analysis method, and to guarantee that specification requirements such as the target BER and jitter tolerance are met.

2.3. Jitter and BER Analysis Methods

With the PLL as fundamental system for clock synchronization we are highly interested in specifying its synchronization performance. Especially in digital high-speed interfaces where jitter can lead to erroneous data recovery, jitter analysis, jitter tolerance and robust design become important issues. In this context the probability of data misinterpretation in terms of bit error rate (BER) is an important quality criterion for receivers. This measure is further supported by interface standards specifying target BER levels of 10^{-12} or even less [118]. Therefore, as figure of merit one ideally likes to give a relation between measured timing jitter and the BER.

The direct verification of a target BER=10^{-12} is very time consuming and impractical, it can quickly take several minutes [72,96] to perform a single BER test. In [89,96] the trade-off between test time and BER confidence level is examined, and the following equation derived for the amount of data bits N, needed to guarantee a desired target BER:

$$N = \frac{1}{B}\left[-\ln(1-a) + \ln\left(\sum_{k=0}^{E}\frac{(N \times B)^k}{k!}\right)\right], \tag{2.9}$$

where B=10^{-12} is the desired BER level, a specifies the statistical probability or confidence level that the true BER value is less than B, and E is the number of detected errors during measurement. When no bit errors occur (E=0), the second term of the equation is zero and the solution to equation (2.9) is simplified. For example, with a=0.95 it is necessary to transmit N=$3.0/B$=$3{\cdot}10^{12}$ bits without errors in order to meet the imposed specification requirement. In a 3Gb/s transceiver this would require an analysis time of T=$N/3{\cdot}10^9$=$1000\mathrm{s}$=$16\mathrm{m}, 40\mathrm{s}$. Such a huge test time cannot be tolerated for high volume production tests, where all specification requirements of the transceiver have to be verified within several hundred milliseconds.

Therefore, test engineers have to rely on analysis methods which allow for accurate BER estimation using a number of jitter samples which is several orders of magnitude smaller. Thus, corresponding mathematical models and equations must be provided in order to correctly verify the desired target BER. Jitter values can usually be obtained easily from a model simulation, however this process is often more complex in practice when carried out on hardware. High precision equipment is required to perform accurate off-chip jitter measurements, including the use of high-speed sampling scopes, time interval analyzers (TIAs) or bit error rate testers (BERTs) [12–14, 86, 130]. A detailed documentation of methodologies for jitter and signal quality measurements can also be found in [43].

External noise sources can easily affect off-chip measurements at multiple Gb/s rates. Thus, a broad variety of built-in jitter measurement (BIJM) systems [16–18, 35, 48, 57, 60, 65, 66, 100, 129, 131] has been developed in recent years as well. Such systems require a large amount of die area if the jitter histograms have to be collected in real-time [16]. This is especially the case if for example frequency domain analyses have to be realized and thus, all jitter values are needed. In cases where the measurement time is uncritical, BIJM circuits also become very small but then, they can only be used for histogram based jitter analysis.

Nevertheless, histogram based methods represent the most important class of analysis principles, since they directly relate jitter with the BER. This is not directly the case for time or frequency domain based methods. In the following sections these three analysis domains will be explained in more detail in order to give a comprehensive overview to the state-of-the-art in the topic.

2.3.1. Histogram Based Analysis

Histogram or statistical domain based methods estimate jitter influence using probability distributions of collected jitter values. Starting with the observed eye diagram in figure 2.10, a jitter distribution is obtained from the horizontal cross section at a desired signal level. For simplicity, here only the optimum decision level or zero crossing line is considered.

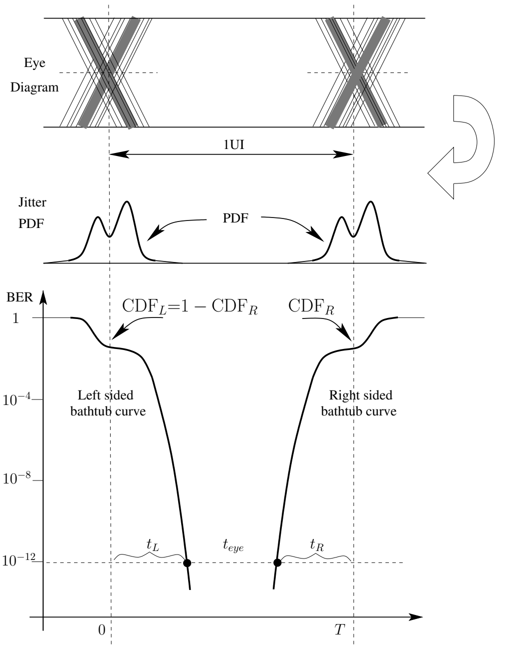


Figure 2.10.: Bathtub function example.

The collected distribution corresponds to the histogram or probability density function (PDF) of jitter samples, assuming that timing jitter is a stationary random process. A measured distribution is often represented by the so called bathtub plot as shown in the bottom part of figure 2.10. Therefore, the integral of the density or cumulative density function (CDF), is calculated for both distribution tails and put into logarithmic scale. Additionally, $\mathrm{CDF}_R(x)$ is defined as right sided bathtub curve, while the reverse function $\mathrm{CDF}_L(x)$=$1-\mathrm{CDF}_R(x)$ is denoted as left sided bathtub, according to the respective tail. The CDF directly describes the BER as a function of sampling time and thus, can be used to identify the jitter extent at any desired BER level of interest.

The goal now is to design a clock recovery system in a way that both tails are separated sufficiently far from each other at the required target BER. This target level is chosen according to the specification requirement for high-speed serial links, which is usually BER_{spec}=10^{-12}. The eye opening at this level is given by the distance between the corresponding points on left and right BER curves:

$$t_{eye} = T - t_L - t_R, \tag{2.10}$$

where T is the bit period, and t_L and t_R the resulting distances on the bathtub curve at 10^{-12}. The total jitter peak-to-peak value TJ_{pp} or timing budget can thus directly be determined with

$$\mathrm{TJ}_{pp} = T - t_{eye} = t_L + t_R. \tag{2.11}$$

If normalized by the unit interval (UI) so that $T{\equiv}1$, t_L and t_R equal the portion of eye closure. These are the parts of the UI not accessible for sampling if the target BER has to be fulfilled.

The bathtub curve representation offers a simple way to verify whether a measured jitter distribution achieves the specification. Unfortunately, in order to determine the correct eye aperture at very low probability levels a huge amount of jitter samples must be collected. For BER levels of 10^{-12} and lower, a direct measurement of the histogram is not feasible. Especially in simulations bathtub curves are only tracked down to probability levels that are orders of magnitude higher than the target BER.

Therefore, an extrapolation of the bathtub curves is required. This extrapolation can be huge, in the case of $N{=}10^8$ jitter samples it still ranges over four orders of magnitude, and can thus only be done correctly if valid model assumptions are made for the underlying jitter distributions. Common model assumptions are aligned to the popular Gaussian tail model [123] and can be characterized as follows:

1. Jitter is a stationary random process.
2. The measured total jitter (TJ) distribution can be separated into two components, random (RJ) and deterministic jitter (DJ).
3. RJ is observed at the outer tails of a TJ distribution, and follows an unbounded Gaussian which can be fully described by its mean μ, standard deviation σ and amplitude A.
4. DJ follows a finite, bounded distribution.

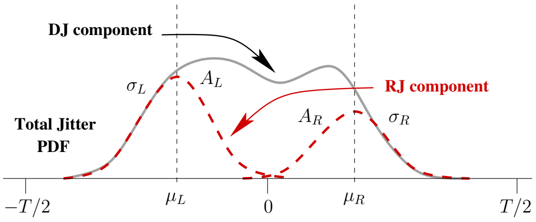


Figure 2.11.: RJ and DJ components of a jitter PDF. Definitions of right(R) and left(L) tails correspond to the bathtub curves from figure 2.10.

According to these assumptions a TJ distribution can always be decomposed into two Gaussian tails together with an arbitrary shaped bounded DJ component, as shown in figure 2.11. In order to correctly extrapolate a measured distribution, analysis methods have to identify the three model parameters μ, σ and A for both tails. This means, one is basically trying to fit a Gaussian function into the measured distribution tails. Jitter analysis methods are thus also referred to as tail fitting algorithms or jitter decomposition methods, while in mathematical statistics this problem domain is also known as tail extrapolation and treated by extreme value theory. Once the model parameters have been identified, the TJ timing budget can easily be calculated for arbitrary probabilities and thus used for BER analysis. A mathematical description of the timing budget is provided later on in section 3.

Various methods were developed to separate the random and deterministic jitter components with tail fitting algorithms [51, 54, 58, 84, 95, 124, 136]. In this section, existing techniques are reviewed in order to provide a comprehensive overview to the state-of-the-art.

Methods Based on Chi-Square Statistics

These are quite popular methods for distribution tail fitting, and have been widely used for jitter and BER analysis in high-speed serial links [52,84,90]. The Tailfit algorithm by Li et al. [85] uses the chi-square test as goodness-of-fit measure for fitting a Gaussian function into distribution tails.

The model includes the parameters mean, standard deviation and amplitude values, and is fitted into the tails by minimizing the difference between measured tail data and model prediction. This minimization process corresponds to a three dimensional optimization.

Methods based on the described principle suffer from a few drawbacks. First, the tail part of the distribution has to be identified before starting the optimization. This includes use of a tail identification algorithm with conservative parameters, which behave suboptimal for a broad range of distribution shapes. Second, the tail fitting algorithm itself suffers from a high complexity due to the three dimensional optimization. A successful minimum search thus requires a robust convergence behavior.

Methods Based on Quantile Normalization

The quantile normalization principle [19, 106, 115] is based on a linearizing transform which greatly simplifies the analysis of distribution tails. The idea is to transform a distribution into a domain where the bathtub tails, if Gaussian, are represented as straight lines. A simple linear extrapolation is then carried out to estimate the TJ timing budget at the target BER. This normalizing transform is realized using the so called quantile function [106], which is commonly used for QQ-plots [19, 20, 116] in statistics. The transformed quantile domain is also referred to as Q-scale [53,95,123] or Q-space [82] in jitter and BER analysis.

Besides the general extreme value theory, Gaussian Q-normalization was first used by Popovici in [111] for BER analysis of digital links. Although the BER is only described as a function of the signal to noise ratio, the underlying mathematical concept can also be mapped onto the jitter decomposition problem. Popovici provides a thorough description of the quantile function to normalize Gaussian distributions, together with rational approximation coefficients and a Gaussian regression algorithm.

Hänsel et al. [51] used the Q-function for decomposing jitter distributions into RJ and DJ. Lines were fitted to the normalized distribution in Q-domain, where the line slope and offset allowed for the reconstruction of Gaussian model mean μ and standard deviation σ (see figure 2.11). This principle was subsequently also described by Stephens [123] and Kizer [70]. However, the method is not able to recover the Gaussian amplitude A, and is thus in some sense incomplete.

A general drawback of this conventional Q-normalization technique is that estimation accuracy is very sensitive to the selected fitting region. The fit should be performed only at the tail parts of the distribution that truly follow the underlying linearized Gaussian. Unfortunately, this line behavior is only approached asymptotically, and it is thus difficult to determine where the asymptote begins.

Hong and Cheng [53, 54] tried to improve the Q-normalization method by fitting higher order polynomials to the Q-normalized bathtub instead of linear functions, and achieved an acceptable accuracy for estimated total jitter values. However, influence of statistical random data fluctuation has not been considered, and the approach investigates only a few special test cases.

Finally, Miller [93–95] proposed the “normalized Q-scale” analysis, where the Q-normalization method also includes the Gaussian amplitude A as third model parameter. The underlying mathematical principle was already described by Popovici [111], and includes an additional pre-scaling factor to normalize the Gaussian amplitude before transforming the distribution into Q-domain.

This thesis will also put the main focus on jitter analysis methods based on quantile normalization. As will be shown, the linearizing property allows for an accurate and efficient extrapolation of distribution tails, and can also be used to derive a unifying optimization scheme for tail fitting which covers all of the above described references [51, 54, 95, 111, 123]. Further, it can be generalized for use with arbitrary non-Gaussian tails.

Other Methods

Other less popular jitter decomposition methods are based on techniques for deconvolution [124, 125, 128], Gaussian mixture models [98] and the wavelet transform [136]. Deconvolution methods rely on the idea that in histogram based analysis a total jitter PDF is given as convolution result of the RJ and DJ components. If one of these two components is approximately known or estimated, a deconvolution algorithm can be used to determine the other component, and thus to retrieve the Gaussian model parameters. A major drawback of these methods is that they suffer from accuracy, since either the DJ or RJ component has to be estimated prior to the deconvolution.

Another method is based on the wavelet transform [136] and uses derivatives of Gaussian wavelets to detect the locations (mean values) of the Gaussian model functions. The variances are determined from a transformed log-likelihood function, while Gaussian amplitudes are not considered. Due to the applied wavelet transform, this approach also suffers from a high computational demand.

2.3.2. Time-Domain Based Analysis

Jitter analysis techniques based on time-domain [27, 89, 143] rely on jitter measurements carried out in real-time. This is only feasible for dedicated real-time measurement systems, such as high-speed sampling scopes or time interval analyzers (TIAs). In time-domain, jitter is then treated as a statistical random signal which can be analyzed in terms of correlation and statistical moments.

Correlation analysis was introduced by Dou and Abraham [27, 28], and considers the evolution of jitter samples in time by calculating the autocorrelation function. Unlike histogram based analysis it allows for the extraction of different DJ components (see figure 2.3), such as duty cycle distortion (DCD), sinusoidal (SJ) and even data dependent jitter (DDJ). With only a few thousand samples, estimates can already be obtained with an acceptable accuracy. Decomposition of DJ into these subcomponents affords identification of the root causes of jitter. Unfortunately, the approach still misses a relation between extracted DJ subcomponents and the total jitter, which impedes derivation of the BER.

In [89] a method for the measurement time reduction based on signal to noise ratio (SNR) decrease is presented. This technique captures the amount of bit errors over a certain number N of transmitted bits. The SNR of the system is intentionally reduced by a known quantity until errors are captured, which results in a quicker measurement of the degraded BER. The relationship between SNR and BER can be derived from Gaussian statistics and is documented in many communications text books, such as [49, 114]. However, the described approach is valid only if Gaussian RJ is the dominant cause for bit errors. It cannot be applied to arbitrary jitter distributions and usually requires an amount of test samples which is too high for simulation applications.

2.3.3. Frequency-Domain Based Analysis

The time domain series of jitter can also be represented and analyzed in frequency domain using the Fourier transform [82, 105, 139]. The PSD is then used to represent the jitter spectrum, by applying averaging techniques such as the periodogram method. Peaks in the spectrum can be interpreted as SJ or DDJ, while the average noise floor denotes the power of RJ. Unfortunately, bounded uncorrelated jitter (BUJ) cannot be distinguished in a PSD, and a relation to the BER is thus not explicitly given.

In [55, 56] four spectral regions of the jitter transfer function of CDR circuits are defined to allow for BER analysis. The approach is restricted to Gaussian RJ combined with SJ, where the sinusoidal jitter frequency is extracted from the spectral information. The obtained jitter transfer characteristic of CDRs is subsequently [101, 102] also used to derive an analytical approximation for the maximum phase error, which can be adapted for BER calculations.

In his book [82] Li thoroughly describes frequency domain principles for jitter separation including DDJ, SJ and RJ types. BUJ can generally not be separated from the RJ noise floor, unless it can be measured independently or controlled in some way.

3. Mathematical Background

This chapter deals with the mathematical basics involved with the developed jitter analysis methods and optimization schemes. First, the quantile normalization is reviewed as fundamental mathematical principle for a powerful class of tail fitting methods which is going to be analyzed and optimized throughout subsequent chapters. Then performance metrics and test distributions are introduced for the qualitative analysis of tail fitting methods.

3.1. Quantile Normalization

Fitting methods investigated in this thesis are all based on quantile normalization. It forms a powerful technique for linearizing the tails of a jitter distribution and thus, allows simple linear functions to be fitted via regression analysis. These lines then become the medium for tail extrapolation. In order to fully understand the underlying concept, first the generic mathematics involved with quantiles are described before focusing on the Gaussian distribution as special case.

3.1.1. Quantile Function

The derivation of quantiles [19,20,106,115,116] starts by defining a set of data samples $x_1, \ldots, x_N$, drawn from an unknown distribution function $F(x)$. The data is ordered, so that $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(N)}$, and the empirical distribution function of observed random samples $\tilde{F}(x)$ is defined as

$$\tilde{F}(x) = p = \frac{i}{N+1} \quad \text{for} \quad x_{(i)} \leq x < x_{(i+1)}\,, \quad i = 1, \ldots, N \tag{3.1}$$

where the use of $N{+}1$ (instead of N) avoids $\tilde{F}(x_{(N)}){=}1$. $x_{(i)}$ is an empirical estimate for the $p_i = i/(N+1)$ quantile of the distribution $F(x)$. One can define the quantile function $Q(p)$ to describe a relation between ordered quantiles and the original amplitude of data, which is equal to the inverse probability function [106]:

$$Q(p) = F^{-1}(p) = x \tag{3.2}$$

This function allows to represent a distribution by the quantile plot (or QQ-plot) [19, 20, 116]:

$$\left\{ \left(F^{-1}\left(\frac{i}{N+1}\right), x_{(i)} \right) : i = 1, \ldots, N \right\} \tag{3.3}$$

where observed amplitudes $x_{(i)}$ are represented in terms of amplitudes of the theoretical model quantiles $F^{-1}(p_i)$. For a large sample size N, the sample quantiles $x_{(i)}$ approximate a shifted and scaled version of the theoretical ones [115, 116]. This offers a linearized perspective on distributions, which is especially useful for tail fitting. As an example, in figure 3.1 an empirical distribution $\tilde{F}(x)$ is shown, obtained from $N{=}100$ random samples of a normal distribution with zero mean and unit variance $F(x){=}\mathcal{N}(0,1)$ (solid line). The quantile function $F^{-1}(p)$ at the right transforms the empirical distribution into an easy-to-fit linear function.

Thus, if a gathered distribution exactly matches the expected probability function as described by the inverse F^{-1}, the result is a perfect line along the unit diagonal. If for example only the tail part follows an expected behavior, as is the case for the RJ-DJ model with Gaussian tails

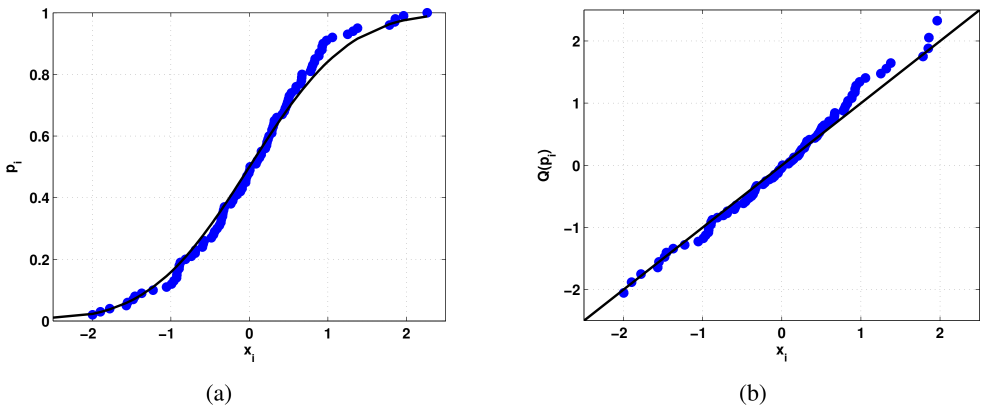


Figure 3.1.: Example for a) an empirical distribution function $\mathrm{CDF}(x)=\tilde{F}(x)$ with $N=100$ random samples of $\mathcal{N}(0,1)$ and b) the corresponding quantile plot.

(section 2.3.1), this line is still observed at the tail parts. This way, the tail fitting problem can basically be simplified to a linear regression analysis.

Such a linearizing transform offers a great simplification for the tail fitting procedure, especially in terms of computational demand. As we will see in chapter 4, the method of least squares can be implemented very efficiently for this purpose, as it uses only summing terms and recursions. Further, the residual error after transform becomes approximately constant and normally distributed over a large probability region, which makes the least squares method an ideal candidate for the maximum likelihood estimation of tail parameters.

3.1.2. Gaussian Quantile Normalization

With the Gaussian tail assumption of jitter distributions in section 2.3.1, one is fundamentally interested in linearizing tails with respect to the normal function. However, the Gaussian tail of a total distribution can be of arbitrary mean, variance and amplitude. Thus a standardized form of the quantile function $Q(p)$ must be provided, which allows to recover the model parameters from fitted lines in quantile domain. Once these parameters have been determined, tail extrapolations down to any desired probability level become very simple.

Subsequently, the Gaussian quantile function is derived and embedded into a simple optimization scheme for tail fitting. This also shows how to use the quantile function together with linear regression analysis. Similar derivations can also be found in [82, section 5.3] or [51, 123] for the Gaussian case, while [19, 115] describe statistical tail modeling in general. As was already explained, the bathtub curve of a total jitter distribution describes the BER as a function of the jitter amplitude x. Initially, a pure Gaussian jitter distribution is assumed where the error rate of the right tail $\mathrm{BER}_R(x)$ can be expressed by the well known Gaussian integral:

$$\mathrm{BER}_R(x) = \rho_T \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\left[\frac{(x'-\mu_R)^2}{2\sigma_R^2}\right]} dx' \tag{3.4}$$

where μ_R is the mean value and σ_R the standard deviation of the Gaussian. The parameter ρ_T is the transition density and reflects the probability of bit transitions in the transmitted data signal. In a clock-like ‘1010…’ pattern for example we have $\rho_T=1$, while for pseudo random binary sequences (PRBS) $\rho_T=0.5$. In our case the BER definition describes the probability course p of the right bathtub curve with negative jitter values and thus, represents the right sided cumulative density function $\mathrm{CDF}_R(x)$:

$$F(x) = p = \mathrm{CDF}_R(x) = \frac{\mathrm{BER}_R(x)}{\rho_T} \tag{3.5}$$

In order to obtain a standardized representation of the integral in equation (3.4) the variable q normalizes a Gaussian function with respect to mean μ and standard deviation σ:

$$q = \frac{x - \mu}{\sigma} \tag{3.6}$$

With the standardized variable we yield

$$\mathrm{CDF}_R(q) = p = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{q} e^{-\left(\frac{q'^2}{2}\right)} dq' \tag{3.7}$$

The complementary error function is defined as

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-x'^2} dx' = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{-x} e^{-x'^2} dx' \tag{3.8}$$

and hence, equation (3.7) can be simplified to

$$\mathrm{CDF}_R(q) = p = \frac{1}{2} \cdot \mathrm{erfc}\left(\frac{-q}{\sqrt{2}}\right), \quad p \in [0, 1] \tag{3.9}$$

For the left bathtub curve $\mathrm{CDF}_L(x)$=1−$\mathrm{CDF}_R(x)$ the same result is obtained, when using the negative BER integral from x to ∞. The inverse Gaussian probability function or quantile normalization is thus given by

$$Q_{gauss}(p) = q = F_{gauss}^{-1}(p) = -\sqrt{2} \cdot \mathrm{erfc}^{-1}(2 \cdot p) \tag{3.10}$$

In jitter analysis $Q_{gauss}(p)$ is briefly known as the Q-function [54, 82, 123]. It is commonly used to transform measured probability functions into Q-domain, where tails appear as straight lines. Often $Q_{gauss}(p)$ is defined using a positive sign. Here, the negative sign is used explicitly to maintain symmetry between probability domain and Q-domain.

An example for a Q-normalized bathtub is given in figure 3.2. The linearizing effect on Gaussian tails yields curves which can easily be fitted and extrapolated by simple linear functions. The standardized variable q as defined in equation (3.6), makes the quantile normalization independent from mean and standard deviation of the original Gaussian model. As a direct consequence, the mean value μ is mapped onto a line offset, while the standard deviation σ is mapped onto a line slope in Q-domain. After the transform, both parameters can easily be retrieved from the linear regression, as also shown in figure 3.2. The zero crossings of the lines correspond to the Gaussian means μ_L and μ_R, while the standard deviations σ_L and σ_R are given by the respective tail slope. A coefficient comparison between standardization and obtained linear function yields:

$$q = (x - \mu)/\sigma \iff q = o + s \cdot x$$

$$\Rightarrow \sigma = 1/s \quad \mu = -o/s \tag{3.11}$$

A simple optimization scheme can be derived from the Gaussian quantile normalization. The scheme in figure 3.3 consists of two consecutive blocks, with the measured CDF data as input and the regression error or an equivalent criterion as goodness-of-fit measure. An optimization procedure identifies the best suited Q-tail region for linear regression analysis and returns fitted

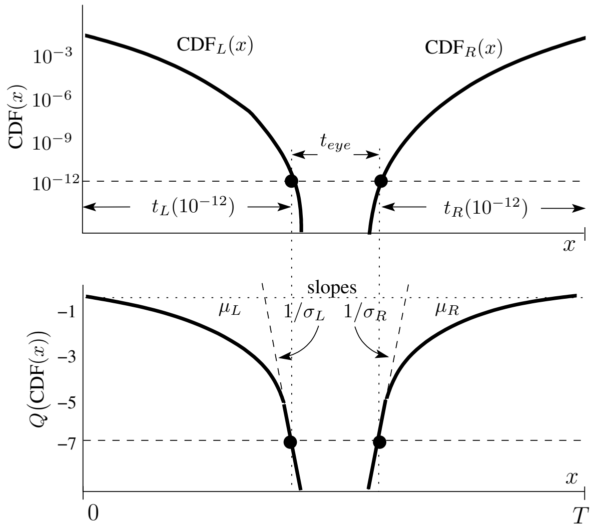


Figure 3.2.: Q-normalization principle demonstrated with a bathtub function.

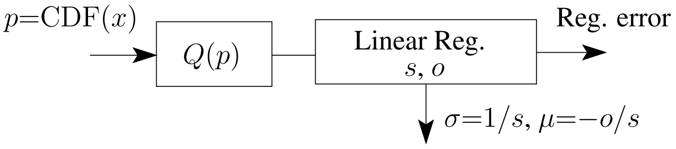


Figure 3.3.: Simple optimization scheme for Gaussian tail fitting based on Q-normalization.

line offset o and slope s. These values are then used to retrieve the Gaussian tail parameters μ and σ. This optimization has to be carried out for both distribution tails independently.

Note, that the presented optimization scheme does not consider the Gaussian tail amplitude A as a third model parameter. This forms a missing gap for many tail fitting methods based on conventional Q-normalization [51, 54, 82, 123]. Therefore, Miller [95] introduced an additional variable for amplitude normalization. In chapter 4 a way to include this variable into the present scheme is shown, which closes the missing gap and significantly improves fitting performance.

Returning to the recovered Gaussian model parameters μ and σ, the TJ timing budget as important quality measure for high-speed interfaces can now be easily retrieved. According to the bathtub function in figure 3.2, the peak-to-peak value of total jitter TJ_{pp} is decomposed into RJ and DJ:

$$\mathrm{TJ}_{pp} = \mathrm{DJ}_{pp} + \mathrm{RJ}_{pp} \tag{3.12}$$

where each of these components is described in terms of the tail parameters:

$$\mathrm{DJ}_{pp} = \mu_L - \mu_R \tag{3.13a}$$

$$\mathrm{RJ}_{rms} = \frac{\sigma_L + \sigma_R}{2} \quad \text{with} \quad \mathrm{RJ}_{pp} = \mathrm{RJ}_{rms} \cdot \big(-2 \cdot Q(\mathrm{BER}_{spec})\big) \tag{3.13b}$$

Here, RJ_{rms} denotes the root-mean-square value and is calculated as the mean of the two Gaussian standard deviations. The probability dependent Q-factor from equation (3.10) denotes the units of Gaussian standard deviations we have to move away from the mean value, in order to reach the desired target BER level. With BER_{spec}=10^{-12}, equation (3.12) is rewritten in the commonly used form:

$$\mathrm{TJ}_{pp} = \mathrm{DJ}_{pp} + 14.07 \cdot \mathrm{RJ}_{rms} \tag{3.14}$$

In order to highlight the influence of RJ_{rms} on the total amount of jitter, in table 3.1 important probability levels together with their Q-factors are given.

$\mathbf{BER}_{spec}$	$-\mathbf{2} \cdot \mathbf{Q}_{\mathrm{BER}}$
10^{-6}	9.51
10^{-9}	12.00
10^{-12}	14.07
10^{-15}	15.88

TABLE 3.1.: Multiplicative constant to specify a target BER for TJ_{pp} values.

A general problem appears with fitted regression lines in Q-domain. The Q-tails often approach the linear behavior asymptotically, which can end up in misleading TJ_{pp} estimates if the fit is not performed in a suitable probability region. Due to the asymptote it is not possible to detect an exact probability level where the linear behavior begins. With the proposed fitting method in chapter 4 this effect is also visible and has to be investigated carefully in order to afford accurate tail extrapolations. In sections 4.3.2 and 4.4 this problem domain will especially be addressed by focusing on performance optimizations with additional fitting parameters.

3.2. Performance Analysis of Algorithms

In order to analyze and compare the performance of tail fitting methods, a broad variety of test distributions must be generated. The basic idea is to evaluate the fitting quality of an algorithm, which is directly reflected by the estimation accuracy of TJ values. This performance primarily depends on the test distribution shape as well as the amount of collected jitter samples. In this section a brief look shows how to investigate these influences.

3.2.1. Performance Metrics

The performance of a fitting algorithm can basically be measured by its ability to correctly identify Gaussian tail parameters. When test distributions are built of RJ and DJ components, unfortunately it is not possible to express the true parameter values as closed form equations. One can thus only use a numerical approximation of the TJ distribution to identify the true timing budget at the target BER=10^{-12}. If the tail fitting algorithm is able to perfectly decompose the test distribution into the parameterized Gaussian tail, there will be no difference between the estimated $\mathrm{TJ}_{pp,est}$ and the true timing budget $\mathrm{TJ}_{pp,true}$. For real fitting methods the difference, or relative error E can be used as a measure for estimation performance. In our analyses, test distributions will be generated with random jitter values, and thus suffer from statistical variations. Therefore, also the estimation error has to be treated as a random variable with its statistical mean and standard deviation:

$$E_k = \frac{\mathrm{TJ}_{pp,est,k} - \mathrm{TJ}_{pp,true}}{\mathrm{TJ}_{pp,true}}, \quad k = 1, \dots, K \tag{3.15}$$

$$E_m = \mathrm{mean}\{E_k\}\ , \quad \sigma_e = \mathrm{std}\{E_k\} \tag{3.16}$$

Here, K is the number of evaluation runs and should at least equal a few hundred, in order to construct empirical relations. Sometimes a tail fitting method might also produce misleading outliers due to convergence problems. In this case it is better to use the median value E_{med} together with the interquartile range IQR (interval between upper q_{up} and lower q_{lo} quartile) to specify statistical spread. These measures are less prone to outlier degradation [29]:

$$E_{med} = \mathrm{median}\{E_k\}\ , \quad IQR = q_{up}\{E_k\} - q_{lo}\{E_k\} \tag{3.17}$$

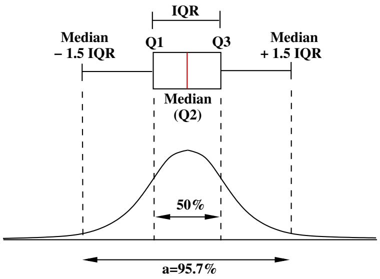


Figure 3.4.: IQR definition for the Gaussian distribution.

Various fitting methods may produce TJ estimates with different error behavior. Therefore, a common performance indicator is preferred, which considers both biased and dispersive error influences. Unfortunately, there exists no optimal combination, and hence, a suitable confidence interval has to be specified. For a Gaussian distributed error as shown in figure 3.4, a symmetric dispersion around the median value is obtained where the data range is typically defined by $3\,IQR$. The Gaussian standard deviation for $\pm 1.5\,IQR$ equals $\pm 2.02\,\sigma_e$, which corresponds to a confidence value of a=95.7%. If the error is not Gaussian distributed as is the case with misleading outliers, E_{med} and IQR form robust estimates, which are better suited to describe distributions of unknown shape.

With the given confidence level a=95.7% the overall error, or estimation loss E_L can be defined as a measure for combined biased and dispersive error influence:

$$E_L = |E_{med}| + 1.5 \cdot IQR \tag{3.18}$$

Only for a normal error distribution we have the equivalent form:

$$E_L = |E_m| + 2.02 \cdot \sigma_e \tag{3.19}$$

According to this definition, E_L defines a positive error threshold, which is exceeded by less than $(1-a)/2 \approx 2.2\%$ of estimates.

Sometimes a tail fitting algorithm may also produce misleading outliers, especially when convergence failures occur. Such failures yield error distributions that are quite different from the ideal Gaussian as shown in figure 3.4, and are typically characterized by slowly decaying heavy tails. Outliers have to be avoided as far as possible. In order to measure their presence, the fourth standardized moment or kurtosis κ can be used. This statistical moment describes the “peakedness” of a distribution and yields a value of κ=3 for an ideal Gaussian. If a distribution is outlier-prone, κ will be significantly larger, while for the bounded uniform case it is κ=1.8.

3.2.2. Test Distributions

With the statistical performance measures, various distribution shapes may be defined to investigate the estimation performance of a fitting algorithm. In literature, different distribution types have commonly been used [54, 69, 82, 104, 120]. In these documents, TJ distributions are usually composed of Gaussian RJ and bounded DJ, both characterized by the standard deviation σ_{RJ} and amplitude A_{DJ}, together with the selected DJ type. In subsequent analyses sinusoidal, uniform, triangular and quadratic curve shaped DJ types will be used for performance evaluation, as depicted at the left of figure 3.5. Together with Gaussian RJ they yield the TJ test distributions at the right.

Sinusoidal DJ is observed as periodic variation of edge positions and thus, also referred to as sinusoidal jitter (SJ). Its root causes can be PLL spurs or power supply noise [104]. Uniform DJ especially appears with inter-symbol interference (ISI), while triangular jitter is caused by crosstalk dominated noise. Quadratic curve shaped DJ is finally obtained as combination of ISI and crosstalk [54], and approximates a bounded Gaussian distribution which may also result from bounded uncorrelated jitter (BUJ) [76, 120].

Combined test distributions can easily be generated with jitter samples of independent time domain random processes. The RJ distribution is realized with a Gaussian normal process J_{RJ} of

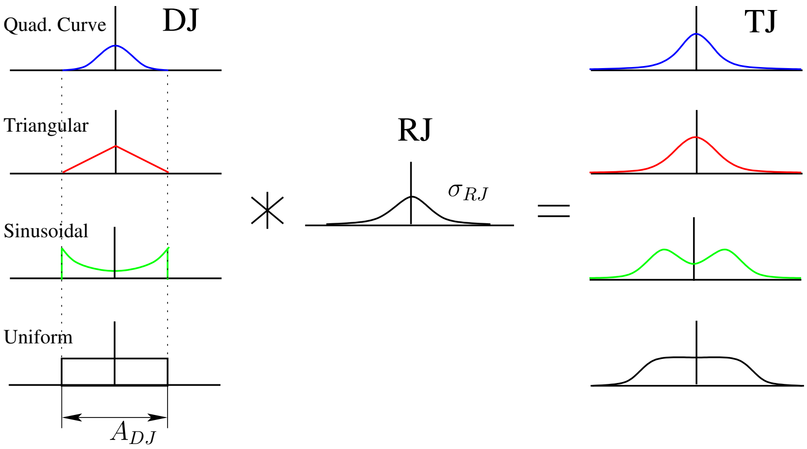


Figure 3.5.: Test distribution types, constructed with parameters σ_{RJ} and A_{DJ}

Random Process	Characterization
J_{RJ}	$\mathcal{N}(\mu{=}0, \sigma{=}\sigma_{RJ})$
$J_{DJ,sin}$	${}^{A_{DJ}}/_{2} \cdot \sin(2\pi \cdot f_{SJ}/f_D \cdot k + \varphi)$
$J_{DJ,uni}$	$\mathcal{U}(-{}^{A_{DJ}}/_{2}, +{}^{A_{DJ}}/_{2})$
$J_{DJ,tri}$	$\sum_1^2 \mathcal{U}(-{}^{A_{DJ}}/_{2}, +{}^{A_{DJ}}/_{2})/2$
$J_{DJ,qua}$	$\sum_1^3 \mathcal{U}(-{}^{A_{DJ}}/_{2}, +{}^{A_{DJ}}/_{2})/3$

Table 3.2.: Definitions for time domain random processes.

zero mean and standard deviation equal σ_{RJ} as developed in [10], while DJ PDFs are constructed with bounded random processes according to the respective DJ shape. Sinusoidal jitter additionally uses a sinus function of random phase φ and frequency f_{SJ}. The uniform process generates a random number out of the bounded interval $[-A_{DJ}/2, +A_{DJ}/2]$, while triangular and quadratic curve shaped jitter can be realized as two or three superimposed uniform processes. The total jitter random process J_{TJ} is simply obtained by the addition of RJ and DJ components:

$$J_{TJ} = J_{RJ} + J_{DJ} \tag{3.20}$$

where the random processes correspond to the shape characterizations from table 3.2.

Note that the distribution synthesis with the two components J_{RJ} and J_{DJ} in time domain yields a convolution in histogram domain. Due to the addition of independent random variables, the resulting TJ distribution will be decomposed into DJ$_{pp}$ and RJ$_{rms}$, which differ from the original σ_{RJ} and A_{DJ} parameters. According to the central limit theorem, the combination of an arbitrary bounded random process with an unbounded Gaussian process always leads to a distribution which is more Gaussian-like than the prior bounded component. Thus, an increase of the RJ component (RJ$_{rms} \geq \sigma_{RJ}$) as well as a decrease of the DJ component (DJ$_{pp} \leq A_{DJ}$) will be observed [123]. For sinusoidal DJ type, this topic has also been investigated in the appendix of [52].

The resulting TJ shapes can be characterized in a representative way using the variable ratio σ_{RJ}/A_{DJ}. The TJ shape then depends only on the relative difference between the two variables, while the distribution size can be described by just one of them.

The $\mathrm{TJ}_{pp,true}$ values at the target BER are determined using numerical approximations. Therefore, closed form equations of the independent RJ and DJ components are convolved, which allows for a direct approximation of the complete TJ shape. Then, the TJ value closest to the 10^{-12} level is extracted and an additional Newton step carried out. The relative numerical error is guaranteed to be smaller than 10^{-4}.

Throughout subsequent analyses, the specified test distributions will always relate to the independent parameters σ_{RJ} and A_{DJ} prior to convolution. This is to provide reproducible simulation results. The uniform DJ shape will especially be utilized as a reference, since it represents a good compromise between easily decomposable sinusoidal shape, and hardly separable triangular or quadratic curve shapes.

4. A Fast and Accurate Jitter Analysis Method

With the mathematical background from the previous chapter a novel, fast and accurate jitter analysis method is developed. This method puts the Q-normalization into the context of a complete three-dimensional Gaussian model optimization, where the unknown tail parameters mean μ, standard deviation σ and amplitude A are identified for both distribution tails. The optimization scheme is realized with simple recursions that allow for a very fast exploration of the search space. As will be demonstrated in this chapter, the proposed method yields very accurate fitting results combined with low computational demand and a flexible design architecture. It automatically determines the best suited tail part for distribution tail fitting, and thus represents a clear improvement to existing tail fitting methods.

The three-dimensional approach is based on an additional scaling factor, included with the optimization scheme from figure 3.3. Thus, it also allows for tail amplitude search. Although the proposed principle has been developed independently, this idea is not novel. Popovici [111] already described the mathematical principle for Gaussian quantile normalization with respect to unknown amplitude A and standard deviation σ. There was no need to include the mean value μ, since the signal to noise ratio was used for BER analysis and the application focus was not on jitter decomposition. Miller [95] was the first to suggest the amplitude scaling factor which finally allowed for three-dimensional Gaussian tail fitting. Unfortunately, both the determination of the tail part as well as the optimization scheme for model parameter search were not described.

In this chapter a complete approach to three-dimensional tail fitting based on Gaussian quantile normalization is provided. A simple optimization scheme is first derived, where the search algorithm simultaneously searches for the Gaussian tail part while identifying the best suited model parameters. A detailed and thorough description of the algorithmic characteristics and associated mathematical fundamentals outlines an excellent fitting quality. A comprehensive performance analysis then gives an impression on the potential of the proposed method. It involves the combination of different test distributions and sample sizes. Subsequent performance optimizations further improve estimation accuracy as well as the robustness of the algorithm. Therefore two conservative concepts based on probability domain parameters as well as a Q-domain threshold are proposed. After respective performance analyses with optimized parameters, the chapter concludes with a summary of the novel jitter analysis method.

Due to the excellent tail fitting performance combined with a fast implementation, flexible architecture and a minimum of conservative fitting parameters, the developed method is also meant to act as a reference for future designs. For this purpose, in chapter 6 a comprehensive performance comparison is carried out against other methods based on quantile normalization. Since the tail fitting method utilizes the quantile normalization in a scaled sense, throughout this thesis it is referred to as “scaled Q-normalization” (sQN) method.

4.1. Scaled Q-Normalization

The derivation of the proposed method starts with the generic Gaussian tail assumption, where a fitting algorithm has to determine the Gaussian model parameters mean μ, standard deviation

σ, and amplitude A, best matching the measured distribution tails. As was already shown in section 3.1.2, the quantile normalization of Gaussian tails yields linearized curves, which can be used for regression analysis. Along these tails in Q-domain the two parameters μ and σ can easily be retrieved, but not the amplitude A. Now the same principle is extended in order to determine all three Gaussian model parameters.

Considering a typical jitter distribution with the PDF as shown in figure 4.1 (solid curve), the probability function or CDF covers the complete probability range from zero to one. Thus, the PDF area equals A=1. Similar, a pure Gaussian function $\mathcal{N}(x)$ (dashed curve at the left) with same area A=1 can be fitted into the distribution tail. For the moment it is assumed that the left Gaussian function $\mathcal{N}(x)$ represents the ideal fitting result with known tail parameters μ and σ.

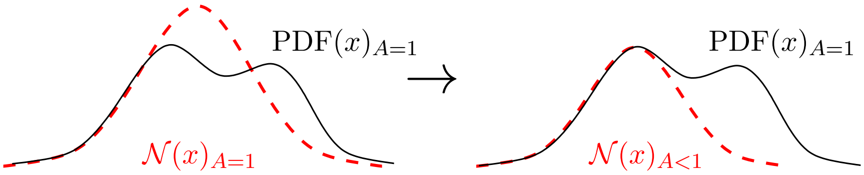


Figure 4.1.: Amplitude matching with adapted Q-normalization function.

At the left of figure 4.1 a comparably small part of the Gaussian function overlaps the outer left tail of the jitter PDF. This means, the Gaussian function with A=1 cannot be fitted nicely into the distribution tail, even with known parameters μ and σ. A smaller Gaussian amplitude instead, would allow for an optimized fit with respect to both tail length and fitting error, as depicted in the right figure. This means, an adapted Q-normalization function must be found to optimize the fit, so that the Gaussian model best matches the distribution tail. With different jitter PDFs, it would theoretically be necessary to derive an adapted Q-normalization function for each of the possible tail areas $A<1$, which is not feasible.

Instead one can think of a reverse approach where the probability of distribution samples is scaled by a multiplication factor k, and the Q-normalization remains constant. This principle is demonstrated in figure 4.2, where the original PDF is blown up by the scaling factor k. Although the obtained probabilities will obviously increase and thus, span an area which is larger than one, the tail fitting principle now has to be seen from the perspective of the constant Q-normalization stage. This means, the normalization is narrowed down toward a smaller probability region, because of the scaling. In fact, the k-scaled distribution is Q-normalized in an original probability region from zero to $1/k$. Thus, the scaled distribution corresponds to a Gaussian tail search with a smaller area of A=$1/k$.

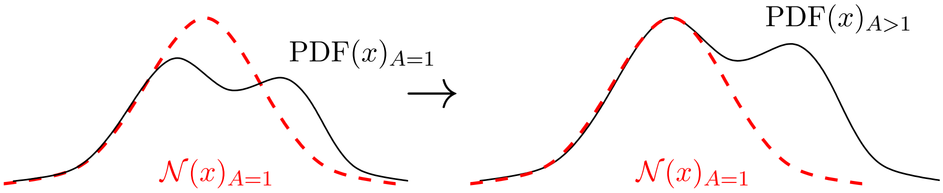


Figure 4.2.: Amplitude matching with scaled distribution.

The three-dimensional tail model can be fully parameterized by

$$\mathrm{PDF}(x) = \frac{A}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.1}$$

to characterize the Gaussian tail shape of jitter distributions. The probability function $\mathrm{CDF}(x)$ will subsequently range from zero to amplitude A, while the quantile normalization is simply achieved

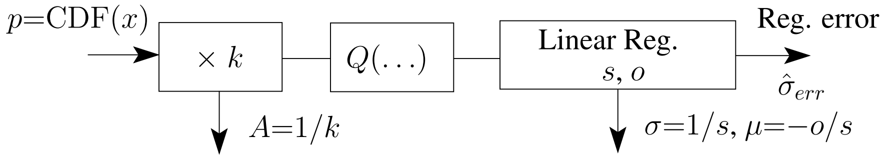


Figure 4.3.: Optimization scheme of scaled Q-normalization (sQN) method.

by probability scaling. In order to construct an enhanced optimization scheme equivalent to the one from figure 3.3, thus, only the scaling factor k must be added prior to Q-normalization. This directly leads to the scaled Q-normalization principle presented in figure 4.3.

With this scheme all three Gaussian model parameters μ, σ, and A can be identified. The parameter $k \geq 1$ scales the tail probabilities of the input CDF. The subsequent Q-normalization stage expects a CDF with values in the complete probability range from zero to one, and thus, narrows the normalized region down toward the tail part. This means, the Q-normalization is carried out only on original tail probabilities from zero to $1/k$, and the resulting Q-tail is linearized with respect to amplitude $A{=}1/k$. The two remaining model parameters standard deviation σ and mean μ are identified by linear regression analysis the same way as already described in section 3.1.2.

Note, that for the moment the optimization scheme is only applied to the negative (right) bathtub tail (see figure 2.10), because the scaling factor k requires the input CDF to start from zero. If used with positive (left) tails, $\mathrm{CDF}_R(x)$ must be replaced with the reverse probability function $\mathrm{CDF}_L = 1 - \mathrm{CDF}_R$. The two obtained Gaussian tail models can finally be used to determine the overall TJ timing budget. Therefore equations (3.12) and (3.13) are rewritten to include the Gaussian tail amplitude A as third model parameter:

$$\mathrm{TJ}_{pp} = \mathrm{DJ}_{pp} + \mathrm{RJ}_{pp} \tag{4.2a}$$

$$\mathrm{DJ}_{pp} = \mu_L - \mu_R \tag{4.2b}$$

$$\mathrm{RJ}_{pp} = -\sigma_L \cdot Q(\mathrm{BER}_{spec}/A_L) - \sigma_R \cdot Q(\mathrm{BER}_{spec}/A_R) \tag{4.2c}$$

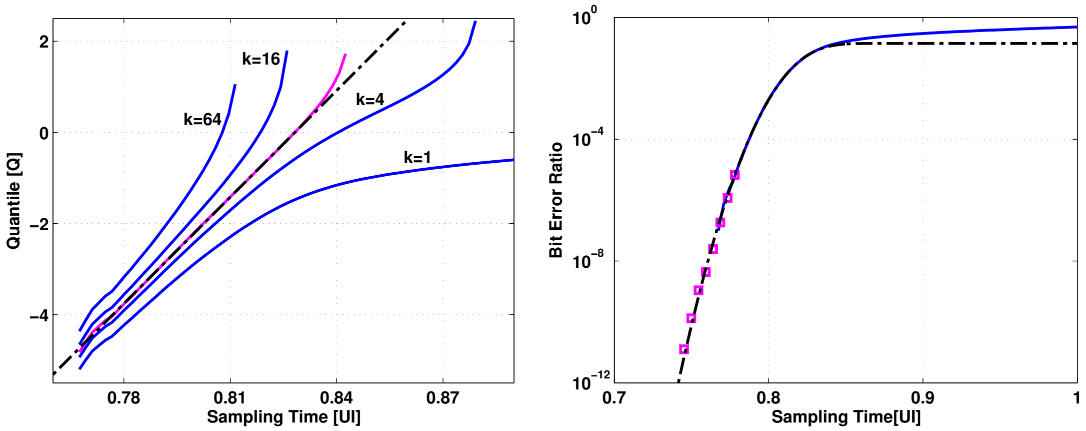
From the third expression we can see, that RJ_{pp} now also depends on the amplitudes of the Gaussian models. For the special case $A_L{=}A_R{=}1$, the Q-function yields $Q(\mathrm{BER}_{spec})$ and the model reduces to the same equation as in (3.13).

In the following subsections the proposed scheme is described in more detail. First, the focus is put on an efficient realization of the optimization procedure and the involved search algorithm. Then, the generalization property of the scheme is highlighted as an additional feature. An error analysis is carried out to justify the quantile normalization with associated linear regression as a fundamental mathematical principle, which is close to the optimum solution for the tail fitting problem. Finally, also the algorithmic details for an implementation in C++ are provided.

4.1.1. Optimization Procedure

An important goal for tail fitting methods is to realize an optimization process with fast tail parameter search. Generally, this optimization corresponds to a minimum search of the regression error or an equivalent fitness measure, with the three unknown model parameters as search dimensions. As will be shown, the proposed scheme is able to solve this optimization problem very quickly.

The scheme in figure 4.3 can also be seen as a twofold approach with two consecutive stages. The first stage weights the measured CDF with the scaling factor k and performs the Q-normaliza-



(a) k-scaled Q-functions for k=$\{1, 4, 16, 64\}$, the maximum linearity is obtained at k=7.15

(b) Left bathtub distribution with fitted Gaussian tail.

Figure 4.4.: Scaled Q-normalization principle demonstrated with N=10^7 jitter samples of a 1 Gb/s signal (measured with Agilent Infiniium 40 GS/s). The jittery data is generated with a SyntheSys BERT-Scope 7500A (sinusoidal A_{DJ}=0.3 UI, σ_{RJ}=0.2/14.07 UI), which also provides the measured magenta bathtub samples at lowest BER level in (b). The required calibration delay is chosen to match the highest BERT point with the measured Agilent bathtub.

tion. Including k into equation (3.10), thus, yields a k-scaled Q-function:

$$Q_k(x) = -\sqrt{2} \cdot \text{erfc}^{-1}\big(2 \cdot \text{CDF}(x) \cdot k\big) \tag{4.3}$$

This function may be plotted for various values of k as shown in the example of figure 4.4, where a typical jitter distribution is analyzed. The effect on the original CDF is observed as bent Q-functions, which achieve best linearity for a certain scaling factor. Here, the resulting analysis domain is also referred to as scaled Q-domain, where the optimum scaling factor yields a linear function which is best described by a Gaussian. The obtained fitting result already shows the potential of this approach, when performing tail extrapolations over several orders of magnitude.

After Q-normalization, the second optimization stage fits a regression line into the tail region of the k-scaled Q-function, using the method of least squares. It yields fitted slope s, offset o and regression error $\hat{\sigma}_{err}$, with the error as fitness measure for the optimization procedure. An essential speed-up is achieved by choosing a representation with the fitting length n as variable.

This variable denotes the n outermost points on a bathtub tail which are used for regression analysis. A finite time resolution for jitter values is used, so that distributions become discretized and consist of a limited amount of R bins per UI. One can also think of dividing the bit period into equally sized steps. This leads to an adjustable time resolution $1/R$, which greatly reduces computational demand. Instead of each single jitter value, the linear regression is now carried out only along the reduced number of bins. When chosen too small, R will obviously degrade the fitting performance. In simulators this can be avoided by selecting R sufficiently large, but hardware systems will usually suffer from coarse resolutions. This problem domain is especially addressed in chapter 5, when focusing on hardware design aspects.

Each collected jitter value is assigned to a bin of the discretized distribution function. The regression error $\hat{\sigma}_{err}$ can thus be represented as a function of the number of fitted bins n, and subsequently used for optimization. In figure 4.5, $\hat{\sigma}_{err}$ is plotted as a two dimensional function of scaling factor k and tail length n, demonstrating its usability as goodness-of-fit measure. The global

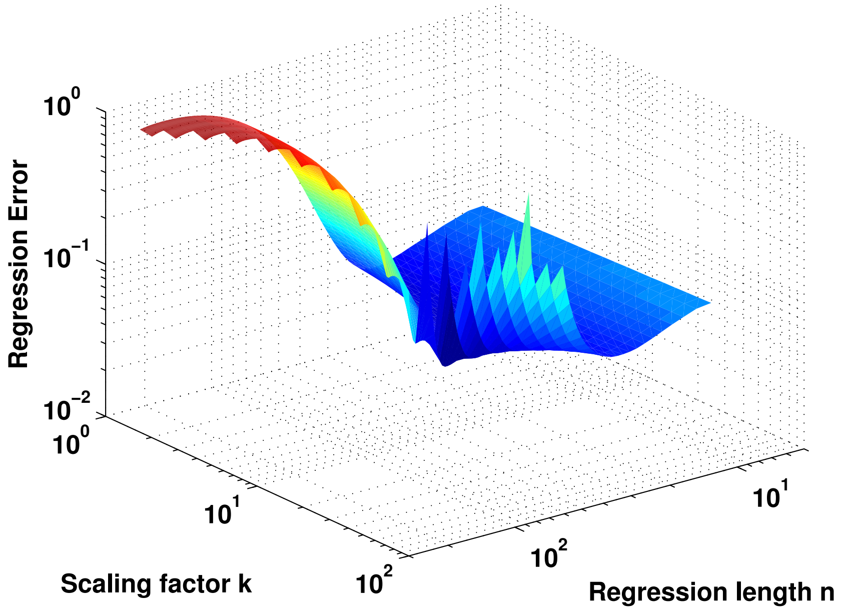


FIGURE 4.5.: Regression error $\hat{\sigma}_{err}$ depending on fitting length n and scaling factor k. The example is the same as in figure 4.4 with a time resolution of $1.83\,\mathrm{ps}$ ($1\,\mathrm{UI} = 1000\,\mathrm{ps}$).

error minimum is obtained at optimized k and n, and can be used to retrieve the three Gaussian model parameters. Although the error is given as a function of only two variables $\hat{\sigma}_{err}{=}f(k,n)$, Gaussian tail fitting remains a three dimensional optimization problem in a strict mathematical sense. The variable n only hides the linear regression, which deals with the two parameters line slope s and offset o.

The linear regression stage detects the error minimum by recursively incrementing n over the Q-tails. That is, the search algorithm starts with a few outermost tail samples or data pairs (x_i, q_i) in Q-domain, and moves toward higher probability levels by recursively adding samples from the bins. Thus, for each additional sample the investigated Q-region becomes larger, while qualitatively described by the corresponding regression error $\hat{\sigma}_{err}(n)$. This procedure has two major advantages. First, all the outermost samples are included, which is very important for tail fitting as they belong to the Gaussian tail part. Second, the linear regression offers very simple recursions for adding tail samples, and hence, the desired error minimum is detected very efficiently.

With given data pairs (x_i, q_i), regression analysis assumes the linear relation:

$$q_i = o + s \cdot x_i \ , \quad i = \{1, \ldots, n\} \tag{4.4}$$

The regression coefficients and the error are calculated using least squares equations [20, p. 393]:

$$s = \frac{n \cdot \sum x_i q_i - \sum x_i \cdot \sum q_i}{n \cdot \sum x_i^2 - \left(\sum x_i\right)^2} \tag{4.5a}$$

$$o = \frac{\sum q_i - \sum x_i \cdot s}{n} \tag{4.5b}$$

$$\hat{\sigma}_{err} = \sqrt{\frac{\sum (q_i - o - s \cdot x_i)^2}{n-2}} \tag{4.5c}$$

where s and o are the estimated parameters for line slope and offset, and $\hat{\sigma}_{err}$ is the standard error to be minimized. Since n is constantly incremented during optimization, the present summing terms can be implemented very efficiently as recursions.

In order to determine a global error minimum, the linear regression stage must be applied to every value of k. To avoid high computational load, a logarithmically scaled search grid is utilized for initial estimation of k. In a refined minimization an accurate estimate is then obtained after a few more iterations. Once the optimization process is completed, the Gaussian tail model is simply given by the fitted parameters:

$$A = 1/k, \quad \sigma = 1/s, \quad \mu = -o/s \tag{4.6}$$

The factor k forms the reciprocal of the Gaussian amplitude as already described previously. The parameter σ is the reciprocal of the gradient or slope of the linearized Q-tail. This is due to the inherent property of the Q-function, to normalize a given distribution in units of Gaussian standard deviation (see equation (3.6)). Finally, μ is the jitter magnitude where the regression line crosses the zero value in Q-domain and decomposes the jitter distribution into bounded DJ and unbounded RJ components. Thus, with

$$q \overset{!}{=} 0 = \mu \cdot s + o \Rightarrow \mu = -o/s \tag{4.7}$$

we yield the third expression in equation (4.6) and hence, confirm the result obtained with the comparison of coefficients in equation (3.11).

The proposed scaled Q-normalization method must be applied to both distribution tails separately. For negative jitter values the right sided distribution function $\mathrm{CDF}_R(x)$ is used as input to the optimization scheme in figure 4.3, while for positive jitter values the reverse function $\mathrm{CDF}_L = 1 - \mathrm{CDF}_R$ is utilized. The obtained left and right Gaussian tail parameters are finally able to decompose a measured jitter distribution into RJ and DJ, as described by equation (4.2).

As will be demonstrated in subsequent performance analyses (section 4.2), the presented approach achieves excellent accuracy, even if a comparable small amount of jitter samples forms the distribution. Due to the linearization of a Gaussian function in Q-domain, the three-dimensional optimization problem (μ, σ and A) is basically simplified to a linear least squares regression with preceding data normalization. A key advantage is the efficient application of recursions inside the regression stage, making the optimization process very fast. Other approaches, such as fitting algorithms based on chi-squared tests [52,84,90] have to face a non-linear three-dimensional optimization. Hence, they are complex and suffer from a high computational demand.

4.1.2. Generalized Optimization Scheme

Another key advantage of the proposed fitting method relates to its flexible architecture. The described analysis principle can be seen in a very generic context where the quantile normalization stage is replaced by a normalization function that linearizes distribution tails according to an expected shape. This way the proposed optimization scheme can be reused for arbitrary non-Gaussian tails, that are given in terms of the three model parameters amplitude A, dispersion σ and location μ. Therefore, the scheme only has to define an adequate normalization function which is able to transform the expected tail behavior into a linear function. Given an expected tail distribution with probability function $p{=}F(x)$ and the quantile function from equation (3.2), this

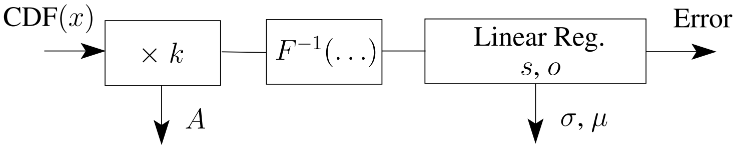


Figure 4.6.: Generalized optimization scheme for tail fitting.

transform is realized with:

$$Q(p) = q = F^{-1}(p,\ \mu{=}0, \sigma{=}1, A{=}1) \tag{4.8}$$

which corresponds to the inverse CDF of unit amplitude, unit standard deviation and zero mean. The corresponding scheme is given in figure 4.6.

Possible candidates for tail fitting other than the Gaussian model are distributions that exhibit power-law behavior at the tails, such as the generalized Gaussian, generalized extreme value or generalized Pareto distributions. These functions include large classes of tail shapes and introduce additional degrees of freedom to the optimization scheme. In chapter 7 this generalization principle will be further discussed.

The ability to substitute the Gaussian quantile normalization with any desired tail shape makes the proposed scheme very flexible and thus, a powerful approach to tail extrapolation. In addtition, if the amplitude pre-scaling factor is omitted, it becomes fully consistent with prior jitter decomposition methods based on conventional Q-normalization as described in [51,54,82,123].

4.1.3. Residual Analysis

In this subsection the mathematical background is provided to justify the proposed normalization scheme from a qualitative perspective. So far, the quantile normalization has been described as a linearizing transform, where the tail fitting problem can be solved very efficiently using the least squares method (equation (4.5)), and the lines obtained serve as a simple medium for tail extrapolation. However, a qualitative analysis of this principle is still missing.

Quantile normalization transforms the measured tail into a linear model where regression can easily be carried out in the desired tail region. For such a linear model one likes to identify the values of slope and offset that make the given data most likely. That is, we are searching for the maximum likelihood, which is given by the method of least squares if the following conditions are met [20, p. 389]:

1. The variance of the response variable q is constant.
2. Residuals are normally distributed.
3. The explanatory variable x is measured without error.

As will be shown, methods based on quantile normalization fulfill these optimality conditions to a reasonable degree and hence, provide an excellent basis for distribution tail fitting. To verify this, the behavior of residuals r must be analyzed, they are defined as the distances between response variable q and model prediction $\hat{q}$:

$$r_i = q_i - \hat{q}_i = q_i - o - s \cdot x_i\ , \quad i = 1, \ldots, N \tag{4.9}$$

A residual plot depicts r_i against fitted values $\hat{q}_i$ and thus, visualizes trends or non-constant error in the scatter behavior of residuals which is also denoted as heteroscedasticity [20, p. 340]. Ideally, r_i should be randomly scattered over the whole plot. This guarantees for a constant error variance which is independent from the fitted model value and thus, fulfills the first of the conditions above.

However, for quantile normalization this is not the case in the outermost tail region, as shown in the example of figure 4.7. Here, the residuals of 25 normal distributions with standard deviation $\sigma_{RJ} = 0.1$ UI, zero mean as well as a sample size of $N{=}10^4$ are represented in a scatter plot. Due to the known parameters, fitted values $\hat{q}$ were replaced by the sample amplitude x, to highlight the error structure along the unit bit period.

The reason for the observed heteroscedasticity is the limited sample size with probability granularity $1/N$. This effect can be visualized by calculating the confidence bounds for quantiles. The

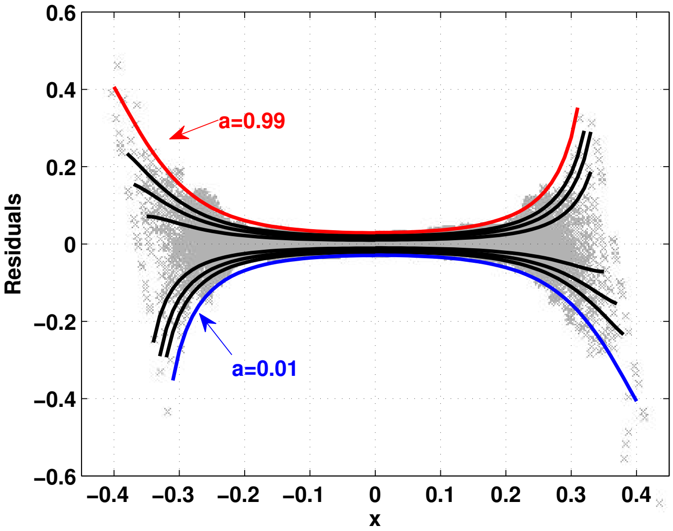


Figure 4.7.: Scatter plot of residuals for 25 realizations of a normal distribution with sample size $N = 10^4$ and $\sigma_{RJ} = 0.1\,\mathrm{UI}$. The curves correspond to the 1%, 5%, 10% and 20% upper and lower confidence bounds when transformed into Q-domain, according to equation (4.12).

probability of an observed quantile to be smaller than the theoretical one corresponds to a binomial random variable with parameters (N, i) [71, sec. 3.1] [117]. Using the binomial distribution this yields:

$$a = P(X_{(i)} \le x_p) = \sum_{j=0}^{i} \binom{N}{j} p^j (1-p)^{N-j} = I_{1-p}(N-i, i+1) \qquad (4.10)$$

where $X_{(i)}$ is the i-th order statistics, x_p the theoretical quantile at probability p, and $a \in [0,1]$ the confidence level. Further,

$$I_x(u,v) = \frac{\Gamma(u+v)}{\Gamma(u)\Gamma(v)} \int_0^x t^{u-1}(1-t)^{v-1} dt \qquad (4.11)$$

is the regularized incomplete beta function [31], which is used for numerical calculations. The confidence level a, sample size N and probability p of the target quantile $x_p = Q(p)$ are fixed, and we are searching for $i = f(a, p, N)$ such that equation (4.10) is fulfilled. The residuals $r_{i,a}$ for the Gaussian distribution example are finally given by

$$r_{i,a} = Q(i/N) - Q(p) \qquad (4.12)$$

where Q is the quantile function, and the resulting $r_{i,a}(p)$ are plotted in figure 4.7 for different a values. At a=0.5 the residual function follows the zero line.

A possible way to compensate existing heteroscedasticity is to use a generalized least squares approach, where observed quantiles are weighted according to their variance. However, even with known variance this is difficult, since quantiles are highly correlated. Thus, the complete covariance matrix Σ must be described. According to [71, sec. 4.8] the covariance elements σ_{ij} can be determined as:

$$\sigma_{ij} = \frac{\min(p_i, p_j) - p_i \cdot p_j}{f(Q(p_i)) \cdot f(Q(p_j))} \qquad (4.13)$$

where p_i are the quantiles as defined in equation (3.1), $f = F'$ denotes the probability density function and $f(Q(p))$ is known as sparsity function [106]. An approximation of Σ is for example given in [117] for the case of generalized extreme value distributions. The presented approach also highlights the matrix computations involved with generalized least squares, and yields a regression model with uncorrelated, constant errors. However, the inverse Σ^{-1} must be calculated, which is only feasible for small sample sizes. This is an essential drawback which impedes utilizing least squares equations (4.5) as efficient recursions. Nevertheless, heteroscedasticity of Gaussian quantiles only influences the outermost tail region. With the huge sample sizes given in jitter analysis scenarios, its influence on fitted tails becomes sufficiently small for a majority of test cases, as will also be demonstrated later on.

Note, that other tail fitting methods that are not based on the quantile normalization as linearizing transform, have to face a non-linear regression where the error structure may highly degrade the quality of fit. The Q-normalization technique instead, can at least always guarantee for an approximately constant error in a higher probability region.

The third optimality condition as listed previously, demands x to be observed or measured without error, which can also be guaranteed only for simulations but not for hardware measurements. In section 4.1.1, already a time resolution variable was introduced, to speed-up the optimization process using distributions with a discrete number of bins R. Thus, collected jitter values additionally suffer from a rounding effect or quantization into integer multiples of $1/R$. This causes an error in x, which also degrades the quality of quantile normalization. Its effect on fitting performance will be investigated thoroughly in chapter 5.

In order to maintain efficiency of the scaled Q-normalization method, the analysis focus is only on performance optimizations where the proposed scheme can utilize the fast least squares recursions. Thus in sections 4.3 and 4.4, optimization will especially be carried out by introducing conservative tail fitting parameters or by selecting suitable goodness-of-fit measures for the search routine. In this context for example, an important question to be answered is, whether outermost tail samples should be discarded or not.

4.1.4. Implementation of the Fitting Algorithm

The presented optimization scheme for the scaled Q-normalization method has been implemented with C/C++ programming language. The goal was to achieve a fast implementation that allows for an in-depth analysis of the proposed method. This problem was addressed by a twofold approach. First, the method was embedded into an easy-to-configure stimuli testbench, which allows for a quick configuration and specification of all relevant parameters of the algorithm. Second, the method itself uses a numerical approximation of the Q-normalization function, which reduces computational cost remarkably. In the following both testbench and fitting algorithm are described in detail.

Testbench for Performance Analysis

A flow diagram of the implemented testbench is given in the left part of figure 4.8. The algorithm starts by parsing all relevant analysis parameters and configuration settings from an input options file before starting the analysis loop. This allows for multiple analyses with different parameter settings to be executed in parallel. On a cluster of simultaneous calculators this principle greatly increases exploration capabilities for the method. The analysis loop requires two key parameters. The sample size N specifies the amount of collected jitter values for each test distribution, while the number of independent evaluation runs K is required for statistical analysis of the extrapolation error. Since jitter is a statistical random process, collected distributions suffer from random tail variations, and thus have to be analyzed using multiple evaluations.

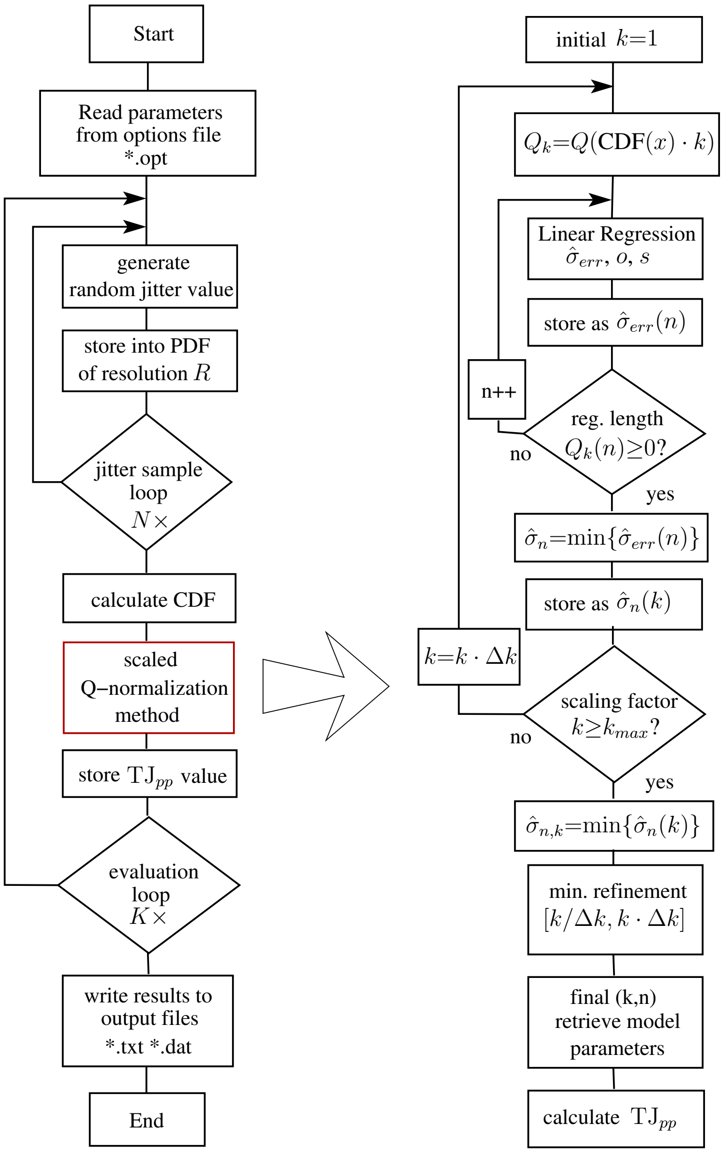

Start
Read parameters from options file *.opt
generate random jitter value
store into PDF of resolution R
jitter sample loop $N\times$
calculate CDF
scaled Q–normalization method
store TJ_{pp} value
evaluation loop $K\times$
write results to output files *.txt *.dat
End
initial k=1
Q_k=Q(CDF(x) $\cdot$ k)
Linear Regression $\hat{\sigma}_{err}$, o, s
store as $\hat{\sigma}_{err}(n)$
n++
reg. length $Q_k(n)\geq 0$?
no
yes
$\hat{\sigma}_n$=min$\{\hat{\sigma}_{err}(n)\}$
store as $\hat{\sigma}_n(k)$
k=$k \cdot \Delta k$
scaling factor $k\geq k_{max}$?
no
yes
$\hat{\sigma}_{n,k}$=min$\{\hat{\sigma}_n(k)\}$
min. refinement $[k/\Delta k, k \cdot \Delta k]$
final (k,n) retrieve model parameters
calculate TJ_{pp}

Figure 4.8.: Flow graphs of implemented testbench (left) and tail fitting algorithm (right).

Jitter samples are generated according to the involved time domain random processes as already described in section 3.2. These random processes are realized as C functions from [34], as an improved alternative to the built-in standard C library. A Mersenne twister is used for generating uniformly distributed samples and the method from [10] transforms them into a normal distribution.

Generated jitter values are assigned to a PDF vector which uses a discrete amount of bins R, specified in the options file. The PDF is thus represented by an integer vector where each bin represents a discrete time interval. When a jitter sample falls into a certain time interval, the corresponding vector entry is incremented, equal to a counter variable. To obtain the simulated probability values, counter values only have to be normalized by the sample size N. This principle limits data memory when gathering large amounts of jitter samples, and allows for modeling the limited time resolution of hardware systems. Later on, with PLL behavioral simulations the resolution of the simulator will be selected as 1fs. For the investigated 3Gb/s serial interface

Parameter Description	Symbol	Default Value
number of jitter samples	N	10^7
number of evaluation runs	K	250
number of bins per unit interval	R=R_{sim}	$3.33 \cdot 10^5$
target BER	BER_{spec}	10^{-12}
default DJ type		uniform

Table 4.1.: Default algorithm configuration and important key parameters.

this yields R_{sim}=$3.33{\cdot}10^5$ bins per bit period, which guarantees for a sufficiently detailed timing resolution so that quantization effects can be neglected.

Once N jitter samples have been gathered, the CDF is calculated as integral of the jitter distribution. The scaled Q-normalization method then fits Gaussian functions into the distribution tails, and returns the estimated timing budget $\text{TJ}_{pp,est}$ at the desired target level of BER_{spec}=10^{-12}. In order to allow for a statistical analysis of this timing budget, K evaluations are carried out. The described key analysis parameters are also summarized in table 4.1, together with their default values.

After the evaluation loop, simulation results are stored in two separate output files. A logfile (*.txt) contains recorded information about simulation progress, successful termination, start and stop time, as well as a copy of the input options file. The output data file (*.dat) contains all estimation results, such as the timing budgets over multiple evaluation runs. This file is meant to be used by MATLAB scripts for post-processing and representation of results.

Fitting Algorithm

The implementation of the tail fitting algorithm is depicted in the right flow graph of figure 4.8. It represents the realization of the scaled Q-normalization block at the left. As already described in section 4.1.1 the basic algorithm is implemented as an optimization procedure which minimizes the fitting error $\hat{\sigma}_{err}$ as function of regression length n and scaling factor k. This concept is realized with a nested loop for the regression length and an outer loop for the scaling factor.

The algorithm uses an initial search grid to identify the best suited scaling factor k, and hence, starts with the first scaling value at k=1. With the CDF as input, the k-scaled Q-function is calculated and the linear regression analysis performed. The nested loop collects error values $\hat{\sigma}_{err}(n)$ over increasing regression length and continues until a maximum value of $Q_k{\geq}0$ in Q-domain is reached. This limit corresponds to a scaled CDF probability of 0.5 or half of the Gaussian tail model. It aids in excluding CDF samples which hardly belong to the measured Gaussian tail, and avoids negative influence of unfavorable DJ shapes.

The minimum error value of the nested loop is the optimum along the first search dimension n, and is stored in a vector according to the different scaling factors of the outer loop. The second search dimension k uses a logarithmically scaled search grid with the grid distance Δk=1.2 as default value. The parameter k_{max}=$1/A_{min}$ must be chosen to include the minimum expected tail amplitude, and thus, also affects computational demand of the search algorithm.

The coarse minimum along the second search dimension k is given by the initial search grid, which must be further refined. This refinement process is carried out with a C implementation of the MATLAB function `fminbnd()` from the Optimization Toolbox [88]. The required upper and lower search bounds are the adjacent grid values of the selected scaling factor Δk, with the final result located inside the interval $[k/\Delta k, k \cdot \Delta k]$.

After the refinement, the optimization of $\hat{\sigma}_{err}$ is concluded, and the Gaussian model parameters can be retrieved from the resulting k_{opt} and n_{opt} values together with the regression coefficients in equation (4.6). Note that the tail fitting algorithm at the right hand side of figure 4.8 has to be applied to both tails of a jitter distribution. The total jitter estimate $\mathrm{TJ}_{pp,est}$ is finally calculated with both Gaussian model parameters according to equation (4.2).

In the C++ environment where the fitting algorithm has been implemented, the computational effort depends on various factors. First, the Q-normalization function utilizes the inverse complementary error function $\mathrm{erfc}^{-1}(x)$, which can only be solved numerically and is computationally expensive when using iterative approaches. As a solution, the MATLAB `erfcinv()` function offers a very fast polynomial minimax approximation, with a relative error $\leq 1.13 \cdot 10^{-9}$. This function has been transferred to the C++ environment. Second, the linear regression analysis has to be carried out over the complete distribution tail inside the nested loop. This evidences the importance of equations (4.5) which use a recursive solution for calculating line offset, line slope and regression error.

Finally, the time discretization into R number of bins per UI highly influences computational effort as it also defines the number of nested loop iterations for a given jitter distribution. In figure 4.9 the average calculation time t_c is determined for a typical test distribution which occupies approximately half of the unit interval. With a larger R, test distributions contain more bins and thus, the computational demand is linearly increased. Simulations are carried out with an Intel Core Duo 2.2GHz laptop, where the scaled Q-normalization (sQN) method with a search grid interval $k{=}[10^1, 10^3]$ and $\Delta k{=}1.2$ is typically 35 times slower than the simplified Q-normalization (QN) method without scaling factor (see section 3.1.2). Note, that the sQN method offers the advantage of a significantly higher accuracy compared to QN, as will especially be demonstrated in chapter 6.

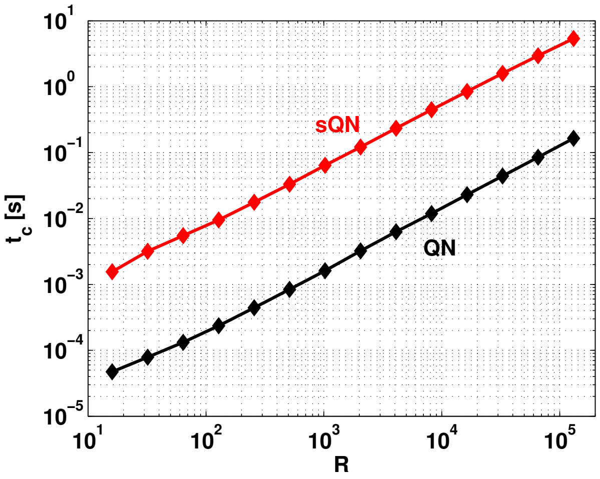


Figure 4.9.: Calculation time t_c of QN and sQN algorithms depending on the number of bins R. Test distribution (section 3.2.2): $\sigma_{RJ}{=}0.05\,\mathrm{UI}$, $A_{DJ}{=}0.2\,\mathrm{UI}$ (uniform DJ), $N{=}10^7$, $K{=}50$ evaluations.

With a bathtub curve that covers half of the unit interval (UI) and thus, consists of $\approx$150 k bathtub samples at $R_{sim}{=}3.33 \cdot 10^5$, sQN optimization of both tails takes several seconds. This may be acceptable for system behavioral simulations where usually minutes or hours are spent to collect a sufficient amount of data samples. With hardware jitter measurements or in production testing this would be too time consuming, but here the bathtub is obtained using phase interpolators of coarse time resolutions, that divide the UI into larger sized intervals. Assuming typically $R{=}128$ bins as given with a 7 Bit phase interpolator, the optimization process is also several orders of magnitude faster compared to the simulation.

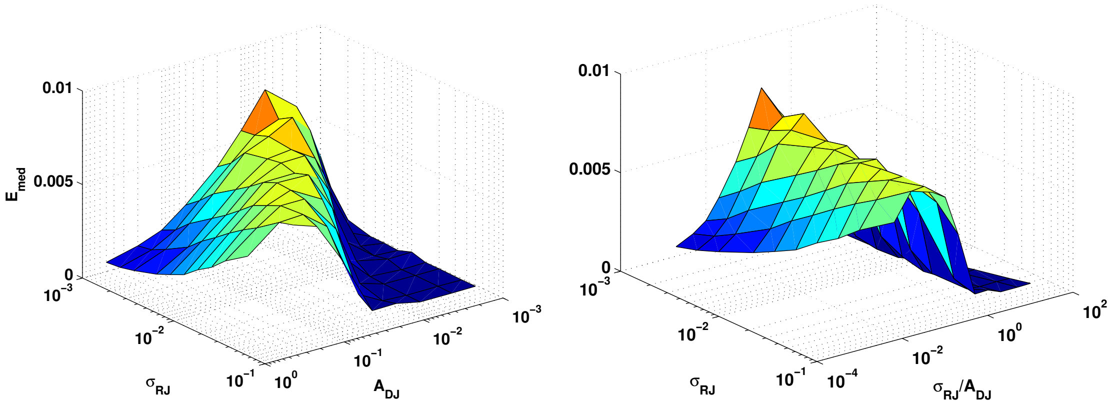
4.2. Performance Analysis

In this section the tail fitting performance of the proposed sQN method is analyzed. Therefore, test distributions and error metrics from section 3.2 are used. According to the way of synthesizing these distributions, the estimation error can be investigated with respect to varying distribution shape (σ_{RJ}, A_{DJ}, DJ type) and sample size N.

Accuracy of the extrapolated timing budget $\mathrm{TJ}_{pp,est}$ is evaluated by observing statistical spread and bias of the estimation error over multiple evaluation runs. At least several hundred evaluations are thus necessary to reliably judge the error behavior. Since it is not possible to predict whether an implemented algorithm reveals convergence problems, statistical measures must be especially robust against outliers. Therefore, median value E_{med} and interquartile range IQR are used, according to the definitions from equation (3.17).

An initial error analysis is carried out by creating test distributions with different values of σ_{RJ} and A_{DJ} (uniform), where the median error bias of the tail fitting method is investigated. Results are shown in figure 4.10(a). The obtained figure is symmetric and can thus be simplified by using only the ratio σ_{RJ}/A_{DJ} instead of both variables. In figure 4.10(b) a rotated view is obtained with σ_{RJ}/A_{DJ} as single dependent variable, which allows to discard σ_{RJ}. From a mathematical perspective the shape of a jitter distribution is fully described by this ratio, as long as the estimation error is not influenced by the timing quantization of distributions. This is always the case for simulations where $R\gg$, so that the estimation error E_{med} does not vary. In the example of figure 4.10 the default analysis configuration from table 4.1 has been used to construct the surfaces. Each point on the surface is the median error value of K=250 evaluations where the estimated $\mathrm{TJ}_{pp,est}$ values are obtained from fitted and extrapolated distribution tails. The true $\mathrm{TJ}_{pp,true}$ values for error calculation are given by numerical approximations as described in section 3.2.2. Each of the distributions uses N=10^8 samples with a resolution of R_{sim}=$3.33{\cdot}10^5$ time divisions per UI.

In the following subsections an initial analysis investigates the influence of varying sample size N. Then, the focus is put on the ratio σ_{RJ}/A_{DJ} as distribution shape variable as well as on different DJ types. Therefore, always the default values from table 4.1 are used for performance evaluation, unless otherwise specified.



(a) (b)

Figure 4.10.: Influence of both σ_{RJ} and A_{DJ} on median estimation error E_{med}. With the variable ratio σ_{RJ}/A_{DJ} a rotated view is obtained where the symmetric representation can be simplified and one of the two variables discarded.

4.2.1. Influence of Sample Size

The performance of the implemented tail fitting method is first investigated with respect to a varying number of jitter samples N. Figure 4.11 shows the statistical spread of K=250 estimates of TJ$_{pp,est}$ at BER=10^{-12} over varying sample size N. The test distribution consists of uniform DJ ($A_{DJ,uni}$=0.2 UI) and Gaussian RJ (σ_{RJ}=0.025 UI). In the graph, blue boxes delimit upper and lower quartiles of the evaluation results, while red lines mark the median value. Black whiskers show the extent of data scattering, outliers are marked with a plus sign, and the dashed magenta line finally denotes the true value TJ$_{pp,true}$=0.523 UI. Figure 4.11 is intended to give a first impression on the statistical behavior of TJ$_{pp,est}$ estimates and on the way these values have to be analyzed in order to provide a qualitative description of estimation performance. Note, that the obtained boxplots also highlight a large amount of outliers, which have to be suppressed using additional tail fitting parameters as will be introduced in section 4.3.

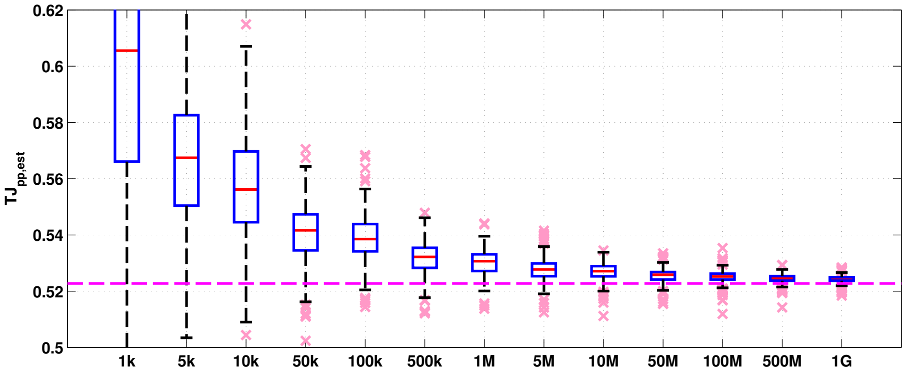
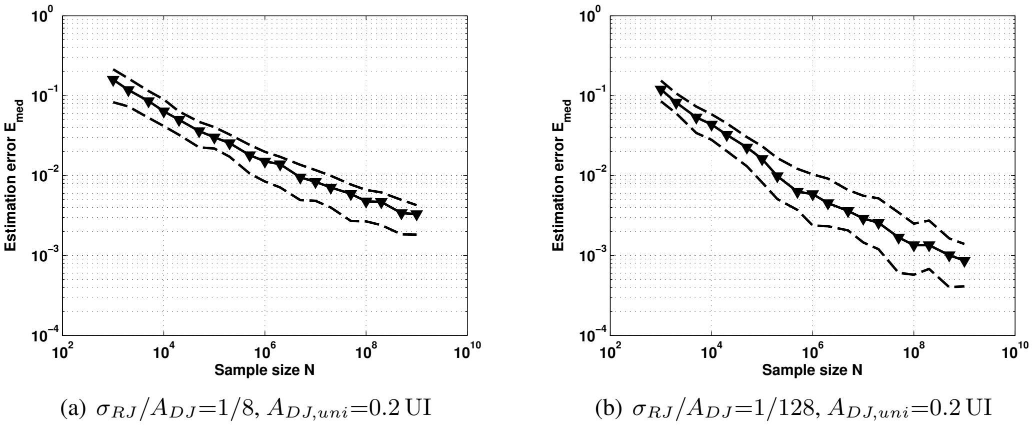


Figure 4.11.: Example for extrapolation error over varying sample size N. TJ$_{pp,true}$=0.523 UI, $A_{DJ,uni}$=0.2 UI, σ_{RJ}=0.025 UI, K=250.

In figures 4.12(a) and 4.12(b) the influence of varying sample size is demonstrated with respect to two different jitter ratios (σ_{RJ}/A_{DJ}=1/8 and 1/128). The markers show median error values obtained with two hundred realizations, while the dashed lines denote upper and lower quartiles as statistical spread. With increasing sample size, both error bias as well as statistical spread decrease toward zero, which empirically proves consistency of the scaled Q-normalization (sQN) approach.



(a) σ_{RJ}/A_{DJ}=1/8, $A_{DJ,uni}$=0.2 UI

(b) σ_{RJ}/A_{DJ}=1/128, $A_{DJ,uni}$=0.2 UI

Figure 4.12.: Influence of varying sample size N on estimation error E_{med}, with two different test distributions.

The positive error bias is an effect caused by the Q-normalization principle. Bathtub functions are transformed into Q-domain where they are represented in a linearized form. This linear behavior of Q-tails is approached asymptotically, which introduces error bias for extrapolated tails. As figure 4.13 demonstrates for a left Q-tail, fitted lines tend to overestimate the true timing budget, especially at small sample size N. With a large number of jitter samples, bathtub curves can be tracked down to deep probability levels. Simultaneously, also the asymptotic behavior of a Q-tail is significantly reduced. Overestimated TJ values yield positive errors with pessimistic estimates, which is a beneficial property of fitting methods based on Q-normalization. This general property is valid for the sQN method as well. Figure 4.12 also highlights an additional influence of the jitter ratio σ_{RJ}/A_{DJ} or distribution shape on estimation performance. This effect is investigated subsequently.

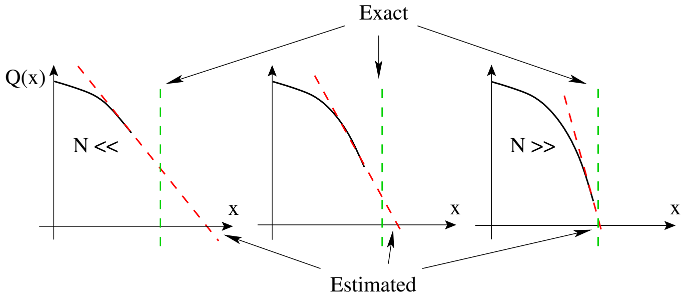
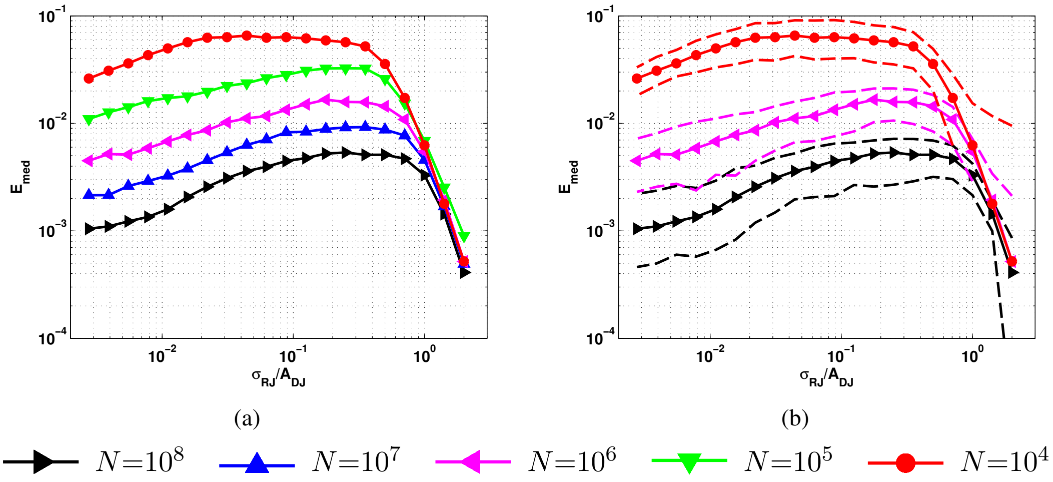


Figure 4.13.: Asymptotic linearity of Q-tails as fundamental cause for error bias.

4.2.2. Influence of Test Distribution Shape

Figures 4.14(a) and 4.14(b) investigate the estimation error over test distribution shape by varying the ratio σ_{RJ}/A_{DJ}. Different performance curves are constructed with $N{=}\{10^4,\ldots,10^8\}$ and uniform type DJ. As expected, best results are obtained with the largest sample size. Note, that with a target $\mathrm{BER}{=}10^{-12}$ the estimation of TJ_{pp} values with $N{=}10^6$ for example, corresponds to a bathtub extrapolation over six orders of magnitude. In 4.14(b) the same curves for $N{=}\{10^4, 10^6, 10^8\}$ are plotted again, together with upper and lower quartiles as dashed lines to demonstrate the influence of statistical spread. Both plots highlight an inferior fitting performance



(a)

(b)

Figure 4.14.: Influence of jitter ratio $\sigma_{RJ}/A_{DJ,uni}$ and sample size N on error.

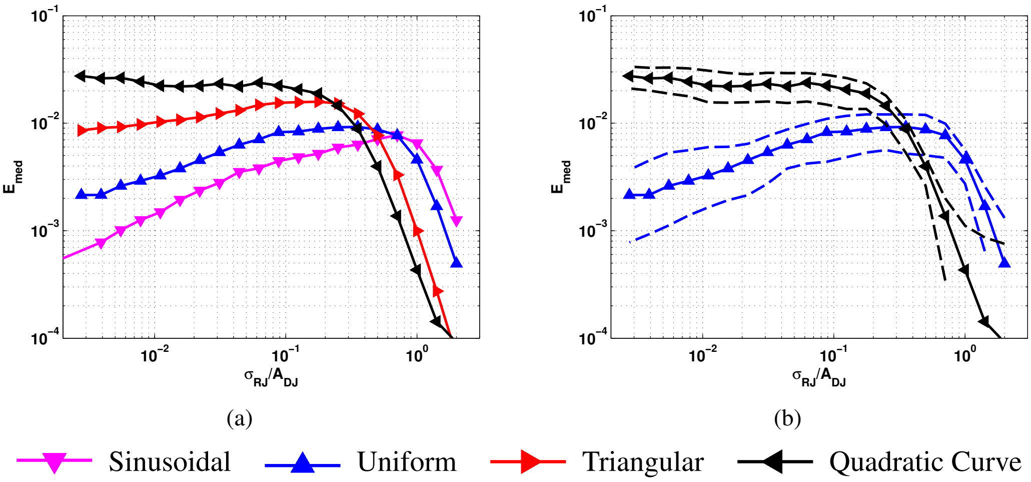


Figure 4.15.: Influence of jitter ratio σ_{RJ}/A_{DJ} and DJ type on error. N=10^7.

for the RJ dominant case, with a maximum around $\sigma_{RJ}/A_{DJ,uni}{\approx}1/4$. This distribution shape will later on also be utilized for worst case analysis.

So far, only uniform DJ has been considered for performance evaluation. Similar to the previous plots, figures 4.15(a) and 4.15(b) thus demonstrate the estimation error and evaluation spread for sinusoidal, uniform, triangular and quadratic curve shaped DJ. If the DJ type is changed to more Gaussian-like shapes such as a triangle or a quadratic curve, estimates also degrade since the algorithm tends to detect a single Gaussian-like peak instead of the steep tails at the distribution edges. Only a very small percentage of the collected samples belongs to the true Gaussian tail, making a correct tail detection very difficult. Therefore the estimation error becomes large, especially when quadratic curve shaped DJ is combined with a small RJ component. Although this combination is rather theoretical and unlikely to appear in real measurements, a possible way to handle the problem is to use a specific parameter configuration optimized with respect to the desired working region.

4.3. Performance Optimization with Different Fitness Measures

In this section performance optimizations of the scaled Q-normalization (sQN) method are conducted to improve estimation accuracy. First an alternative fitness measure based on the regression length of fitted tails is introduced and combined with the conventional regression error to achieve an optimized fitness criterion for Gaussian model parameter search. To improve the convergence behavior and outlier suppression of the method, several conservative fitting parameters are introduced and analyzed. As already mentioned, due to random variations of the outer Q-tail part, the linear regression can produce misleading results which cause outliers in TJ_{pp} estimates. The additional fitting parameters can suppress such outliers when selected appropriately, but they have to be understood and used carefully in order to avoid any performance degradation.

4.3.1. Performance Analysis of Different Fitness Measures

So far, only the regression error $\hat{\sigma}_{err}$ has been considered as goodness-of-fit measure for Gaussian model parameter search. In this section other fitness measures and their combinations are investigated as well, in order to increase the quality of the proposed fitting method.

Figure	Description	Fitness Value
4.17(a)	Regression error	$\hat{\sigma}_{err}$
4.17(a)	Regression length	$\hat{n}$
4.17(b)	Error/slope	$\hat{T}$=$\hat{\sigma}_{err}/s$
4.17(b)	Reg. length Error/slope	$\hat{n}_T$

Table 4.2.: List of investigated fitness measures.

As was shown with the optimization scheme in section 4.1.1, the proposed sQN method yields k-scaled Q-functions after the transform (see equation (4.3), figure 4.4(a)), and identifies scaling factor k and fitting length n with optimum linearity. For this case the regression error becomes a minimum, but the length of the linearized tail also approaches a maximum. Thus one may consider a fitness measure based on the regression length of Q-tails as well. The criterion $\hat{n}$ can for example be defined as

$$\hat{n} = 1 - \frac{n_{err,min}}{R}, \quad n_{err,\min} = n\Big|_{\hat{\sigma}_{err}=min\{\hat{\sigma}_{err}(n)\}} \tag{4.14}$$

where R is the number of bins per UI, here needed for normalization, and $n_{err,min}$ is the length which corresponds to the regression error minimum along a transformed Q-tail. $\hat{n}$ maximizes the regression length n or number of fitted tail samples over varying scaling factor k. That is, the regression error $\hat{\sigma}_{err}$ is still used for minimization along the first search dimension n, while $\hat{n}$ is applied along the second search dimension k.

In [117] Scholz provides a linearizing transform for generalized extreme value distributions, and suggests a fitness criterion based on the ratio of regression error $\hat{\sigma}_{err}$ and tail slope s:

$$\hat{T} = \hat{\sigma}_{err}/s \tag{4.15}$$

Scholz analyzes the covariance structure of the regression error after performing the transform and demonstrates, that the variance of the error depends on the slope s of the fitted line. Thus, $\hat{T}$ can be used to indicate the appropriateness of a fitted tail and to compensate the slope influence.

Equivalent to the regression length $\hat{n}$, a length measure based on $\hat{T}$ instead of $\hat{\sigma}_{err}$ may be defined. This measure also maximizes the regression length of fitted tails, and is denoted as $\hat{n}_T$.

$$\hat{n}_T = 1 - \frac{n_{T,min}}{R}, \quad n_{T,\min} = n\Big|_{\hat{T}=min\{\hat{\sigma}_{err}(n)/s(n)\}} \tag{4.16}$$

$\hat{n}_T$ also searches for the best suited scaling factor k and acts only along the second search dimension.

With the different fitness measures, now also the algorithmic implementation from figure 4.8 must be adapted. Therefore, in figure 4.16 the flow graphs of the fitting algorithms based on $\hat{\sigma}_{err}$ (left) and $\hat{T}$ (right) optimization are plotted again. Depending on the selected type of algorithm, either the regression length ($\hat{n}$, $\hat{n}_T$) or the regression error ($\hat{\sigma}_{err}$, $\hat{T}$) is used for the outer loop with scaling factor k. The nested loop remains the same for both cases. All four optimization criteria are also summarized in table 4.2.

Estimation performance of the fitness measures in figure 4.17 is tested by carrying out K=250 evaluation runs over varying jitter ratio σ_{RJ}/A_{DJ}. From figure 4.17(a) we notice that $\hat{\sigma}_{err}$ provides a smaller error bias (expressed with markers as median values), while $\hat{n}$ achieves a better accuracy, or less statistical spread (expressed by the dashed lines as upper and lower quartiles). An equivalent effect is observed with $\hat{T}$ and $\hat{n}_T$. Although $\hat{T}$ provides the smallest bias out of the four fitness criteria, the lower quartiles evidence a large distance to the median values.

In order to determine an optimum trade-off between these two conflicting interests, a common performance indicator must consider both error bias and dispersion. When recalling the performance metrics from section 3.2, the estimation loss definition in equation (3.18) E_L=$|E_{med}|$ +

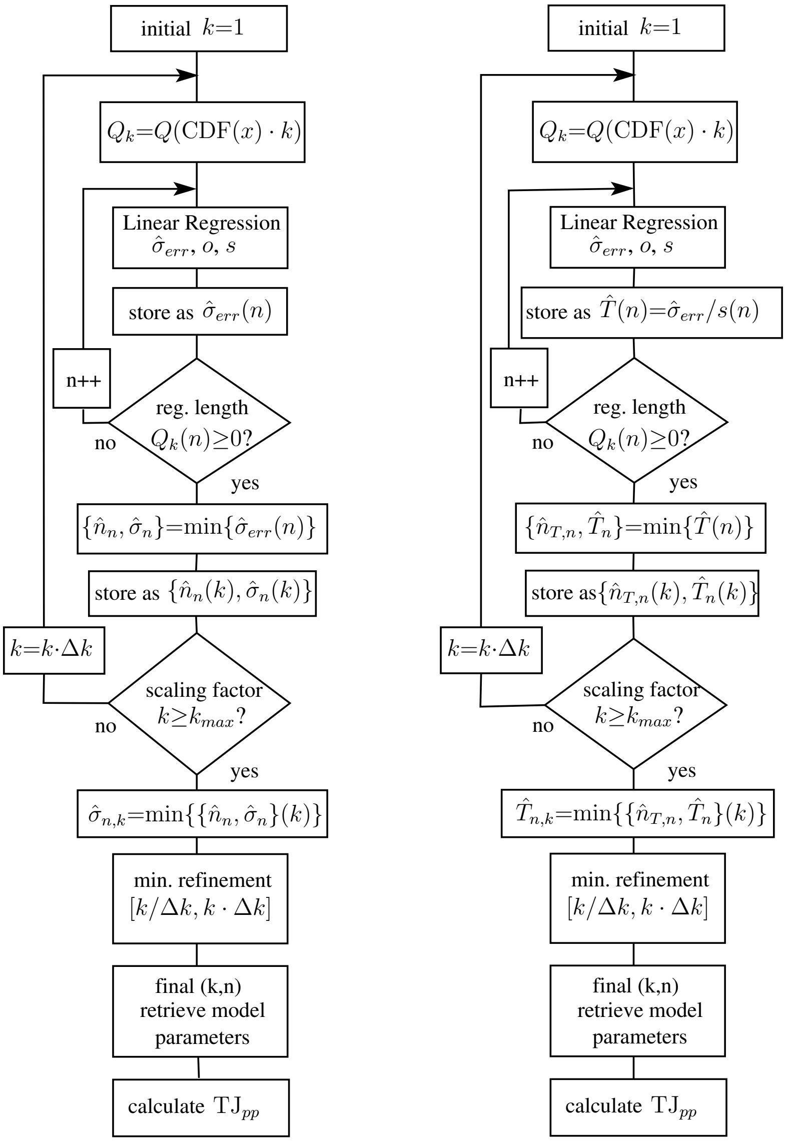
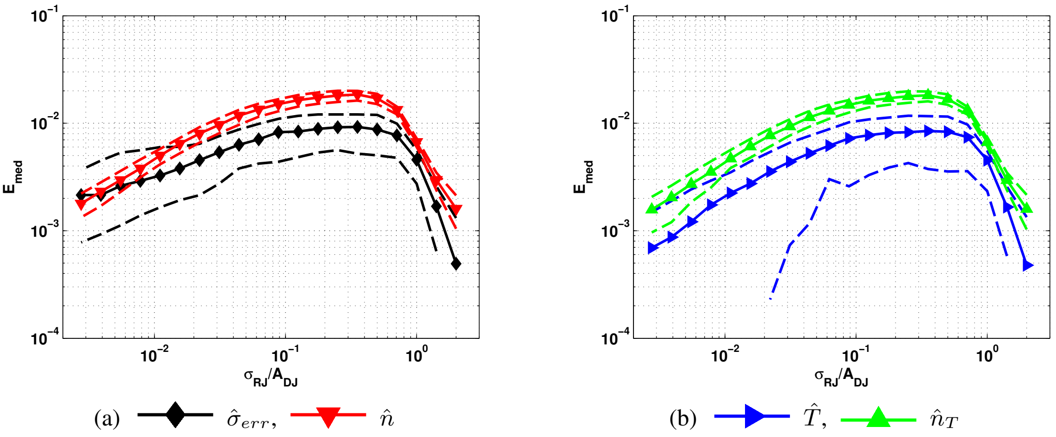

initial k=1
Q_k=Q(CDF(x) · k)
Linear Regression
$\hat{\sigma}_{err}$, o, s
store as $\hat{\sigma}_{err}(n)$
n++
reg. length
$Q_k(n)\geq 0$?
no
yes
$\{\hat{n}_n, \hat{\sigma}_n\}$=min$\{\hat{\sigma}_{err}(n)\}$
store as $\{\hat{n}_n(k), \hat{\sigma}_n(k)\}$
k=k·Δk
scaling factor
$k\geq k_{max}$?
no
yes
$\hat{\sigma}_{n,k}$=min$\{\{\hat{n}_n, \hat{\sigma}_n\}(k)\}$
min. refinement
$[k/\Delta k, k \cdot \Delta k]$
final (k,n)
retrieve model
parameters
calculate TJ$_{pp}$
initial k=1
Q_k=Q(CDF(x) · k)
Linear Regression
$\hat{\sigma}_{err}$, o, s
store as $\hat{T}(n)$=$\hat{\sigma}_{err}/s(n)$
n++
reg. length
$Q_k(n)\geq 0$?
no
yes
$\{\hat{n}_{T,n}, \hat{T}_n\}$=min$\{\hat{T}(n)\}$
store as$\{\hat{n}_{T,n}(k), \hat{T}_n(k)\}$
k=k·Δk
scaling factor
$k\geq k_{max}$?
no
yes
$\hat{T}_{n,k}$=min$\{\{\hat{n}_{T,n}, \hat{T}_n\}(k)\}$
min. refinement
$[k/\Delta k, k \cdot \Delta k]$
final (k,n)
retrieve model
parameters
calculate TJ$_{pp}$

Figure 4.16.: Flow graphs of $\hat{\sigma}_{err}$ based (left) and $\hat{T}$ based (right) algorithmic principles.

$1.5{\cdot}IQR$ can be used. E_L gives a simple confidence interval which is exceeded by less than 2.2% of the evaluations for a normal error distribution.

With E_L the proposed fitness measures can now be compared against each other as shown in figure 4.18. From this representation we notice that the performance of pure regression error $\hat{\sigma}_{err}$ is not outperformed by $\hat{T}$ due to its large statistical spread. Both $\hat{n}$ as well as $\hat{n}_T$ perform slightly worse than $\hat{\sigma}_{err}$ over a broad shape region. The reason why $\hat{T}$ is outperformed by $\hat{\sigma}_{err}$, is the large distance of lower quartiles from the median value. In fact, TJ estimates demonstrate skewed distributions with heavy negative tails. A possible cause to this effect might be correlated errors inside the regression model. In [117] Scholz de-correlates the error of regression parameters using the inverse of the covariance matrix, but unfortunately this highly increases complexity of the fitting algorithm and is thus impractical for computations.

Since the asymptotic tail behavior always guarantees for a positive error bias, one could argue that the estimation loss can also be calculated using only the upper quartile distance. Negative errors would thus not affect the worst case error, even if distributions are heavy tailed. Neverthe-



(a) $\hat{\sigma}_{err}$, $\hat{n}$ (b) $\hat{T}$, $\hat{n}_T$

Figure 4.17.: Estimation performance of different fitness measures from table 4.2. Test distributions are generated with the default analysis configuration from table 4.1.

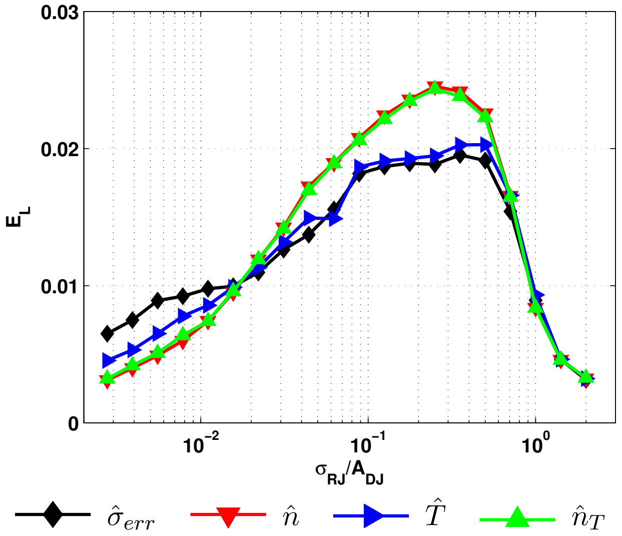


Figure 4.18.: Estimation loss indicator E_L of different fitness measures.

less, in order to avoid outliers and heavy tails as far as possible and to ensure a certain algorithmic robustness, a Gaussian-like error behavior is preferred.

A veritable improvement of estimation performance can be achieved by combining fitness measures during the optimization procedure. So far, the whole optimization has been performed using only one fitness measure out of the candidates from table 4.2. When recalling the flow graphs in figure 4.16, the optimization process first starts with an initial search grid and then refines the obtained optimum grid value down to the desired accuracy. This principle can also be used to combine two different fitness measures. Out of the numerous possibilities two combinations are investigated and presented subsequently.

In figure 4.19 the search grid resolution Δk is varied using both $\hat{\sigma}_{err}$ and $\hat{n}$ as fitness measures. For the logarithmically scaled search points, $\hat{n}$ is first used to determine the initial rough grid estimate, and the refinement is then continued with $\hat{\sigma}_{err}$ in a local search environment. From the resulting E_L curves, the best results are obtained with a grid distance Δk=1.2. A comparison with the performance in figure 4.18 clearly shows the improvement.

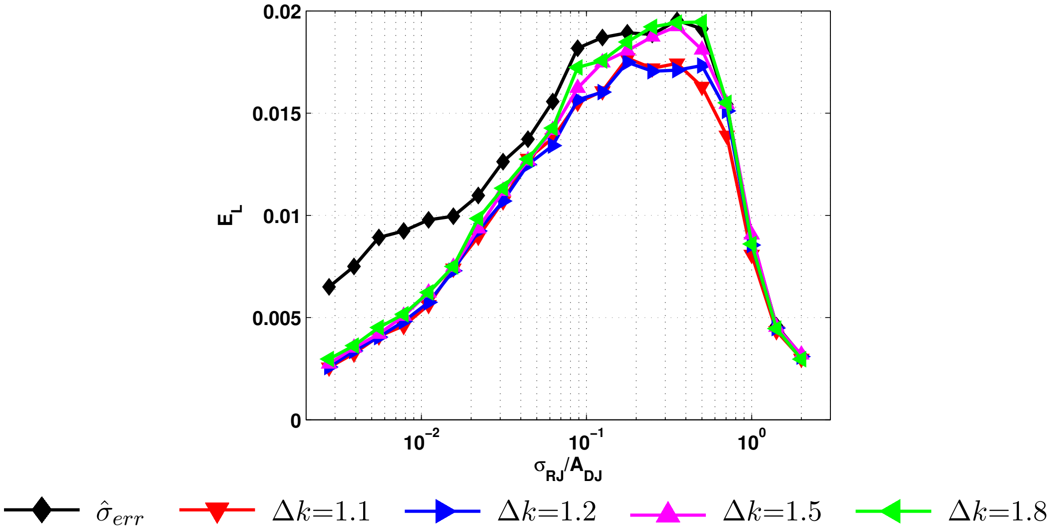


Figure 4.19.: Performance E_L over varying search grid resolution Δk=$\{1.1, 1.2, 1.5, 1.8\}$. The black curve is the reference from figure 4.18.

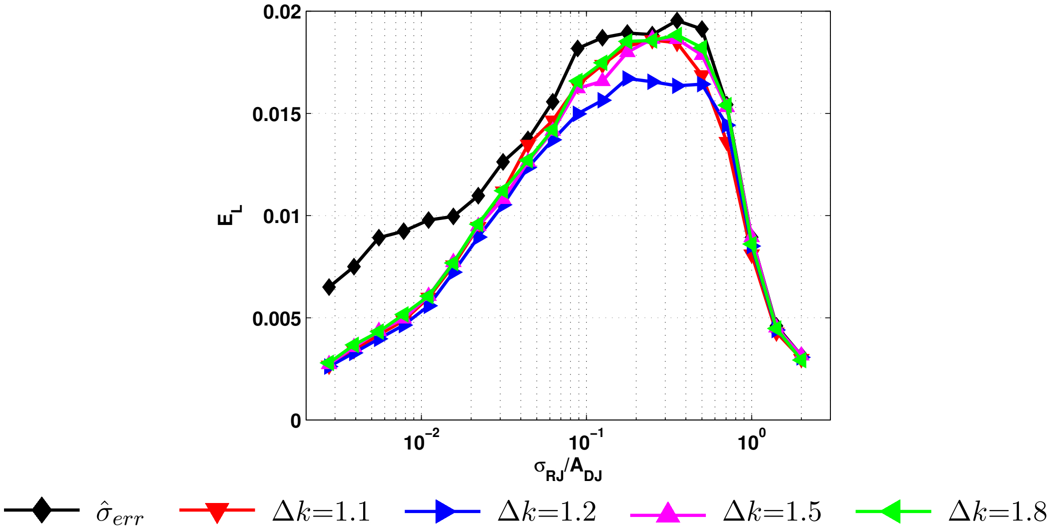


Figure 4.20.: Performance E_L with secondary search refinement Δk=$\{1.1, 1.2, 1.5, 1.8\}$. The black curve is the reference from figure 4.18

In figure 4.20, after complete optimization with $\hat{n}$, a second refinement step with $\hat{\sigma}_{err}$ is additionally performed in a local environment around the first estimate. The environment is specified with the same scaling factor Δk. The performance results are similar to those before and thus, again better than simple $\hat{\sigma}_{err}$ optimization. Considering the increased optimization effort for this scenario the first one is the more convenient choice.

Summarizing these results, a well suited combination of $\hat{n}$ and $\hat{\sigma}_{err}$ measures is obtained, when a search grid resolution Δk=1.2 first provides an initial estimate with $\hat{n}$, which is then refined by $\hat{\sigma}_{err}$. Subsequently this combined optimization scenario is referred to as $\hat{c}_{1.2}$.

Note that the error maxima in figures 4.18 to 4.20 are always located at the same jitter ratio σ_{RJ}/A_{DJ}≈$1/4$. This means, the worst case shape is constant and independent from the investigated fitness measures, and can thus be used for simplified worst case analysis. The presented analyses used only test distributions of uniform type DJ and thus, the optimized $\hat{c}_{1.2}$ scenario is theoretically only valid for this single DJ shape. In other cases one would thus also have to search for other combinations as well. However, the uniform distribution is here considered as a good trade-off between easily separable sinusoidal and difficult Gaussian-like shapes. This allows to use $\hat{c}_{1.2}$ as algorithmic scenario for all DJ shapes.

4.3.2. Improvement of Convergence Behavior

The previous subsection has focused on optimum fitness measures for determining the Gaussian model parameters. In this subsection strategies are investigated to avoid convergence failures of the fitting algorithm. Such failures are noticed as misleading outliers in the estimated timing budget, and can only be suppressed by introducing additional conservative parameters. The goal is to provide suited parameters that afford a maximum flexibility on investigated test distributions without affecting the performance.

When recalling the regression stage in figure 4.3, a linear function is fitted to the tails of the scaled, Q-normalized bathtub curve. The fitting algorithm starts the minimum search with an initial number of outermost tail samples. The fitted bathtub part, or investigated tail region is expressed in terms of the regression length n and is successively increased, so that the regression error minimum together with corresponding fitting length can be determined along the tail.

Due to the limited quantity of jitter values N for a distribution, in Q-domain an increase of the tail variance toward lowest probabilities is observed. This results from the discretization with probability granularity $1/N$. In other words, the variance of regression error is not constant as desired (also see section 4.1.3), but depends on the measured tail probability, which especially affects lowest probabilities near $1/N$. If the fitted lines are constructed with only few probability values of the outermost tail part, random data variations may thus cause highly misleading outliers.

Conservative Parameters for Tail Fitting

The described fitting problem can be handled by introducing several conservative parameters that specify a constant bathtub region at the tail edges which must be used for initial tail fitting. If the fit starts with a sufficient minimum amount of initial tail samples n_{min}, wrong convergences caused by the granularity noise can be fully suppressed. Subsequently, three conservative fitting parameters as defined in figure 4.21 are investigated.

A simple way to overcome random data fluctuation is to introduce a minimum probability threshold $P_{t,min}$. This parameter simply discards bathtub nodes with probability level less than the specified threshold. The second parameter ΔT_t specifies a time interval along the tail where initial bathtub samples are selected for regression analysis. In a simulator with adjustable time resolution one can always guarantee for sufficient data points on the bathtub curve, even for distributions with a very steep tail. Thus ΔT_t may be used down to the selected time resolution $1/R$. If combined with $P_{t,min}$, the tail fitting algorithm only uses nodes with probability higher than the threshold.

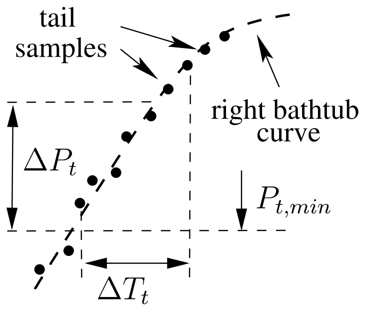


Figure 4.21.: Parameter definitions

Equivalent, a third parameter ΔP_t may define an initial probability interval. The amount of selected tail samples thus changes with the slope of a Q-tail. This means, less bathtub samples are selected with steep tails, which automatically considers the influence of varying RJ tails.

In the subsequent paragraphs the influence of the three parameters with respect to estimation performance and statistical dispersion is investigated. The idea is to give selection criteria and to see how an adequate choice improves fitting performance. The definition of estimation loss E_L from equation (3.18) is therefore reused as performance indicator. To investigate the influence of outliers caused by the algorithm, further the kurtosis κ is utilized as a presence-of-outlier indicator which has to be minimized. Subsequent analyses are carried out with the default parameters from table 4.1.

Influence of $\mathbf{\Delta T_t}$ and $\mathbf{P_{t,min}}$

The parameter ΔT_t specifies a constant part of the unit interval (UI), and hence, it defines a minimum number of tail samples n_{min} which must be included with the linear regression analysis. For a simulator with pre-specified time resolution R_{sim}, n_{min} is given by:

$$n_{min} = R_{sim} \cdot \Delta T_t \tag{4.17}$$

ΔT_t has to be chosen carefully in order to afford correct TJ estimation. If selected too large, bathtub samples that do not belong to the Gaussian tail are used for tail fitting, which results in a high estimation error. If selected too small, outliers may be caused by wrong convergences of the linear regression stage, simply because the initial tail region is only supported by a few bathtub samples. Especially RJ dominant distributions consist of highly varying tail endings and thus, suffer from this effect.

Median error E_{med}, interquartile range IQR, performance E_L and kurtosis κ, depending on both ΔT_t and $P_{t,min}$ are investigated in the rows of figure 4.22. The jitter ratio $\sigma_{RJ}/A_{DJ,uni}$=$1/4$ corresponds to the worst case shape from the performance analysis in figure 4.18, where the estimation loss is maximum. The other analysis parameters are set to the default configuration from table 4.1. Three different optimization criteria with $\hat{\sigma}_{err}$, $\hat{n}$ and $\hat{c}_{1.2}$ from section 4.3.1 are investigated in the different columns.

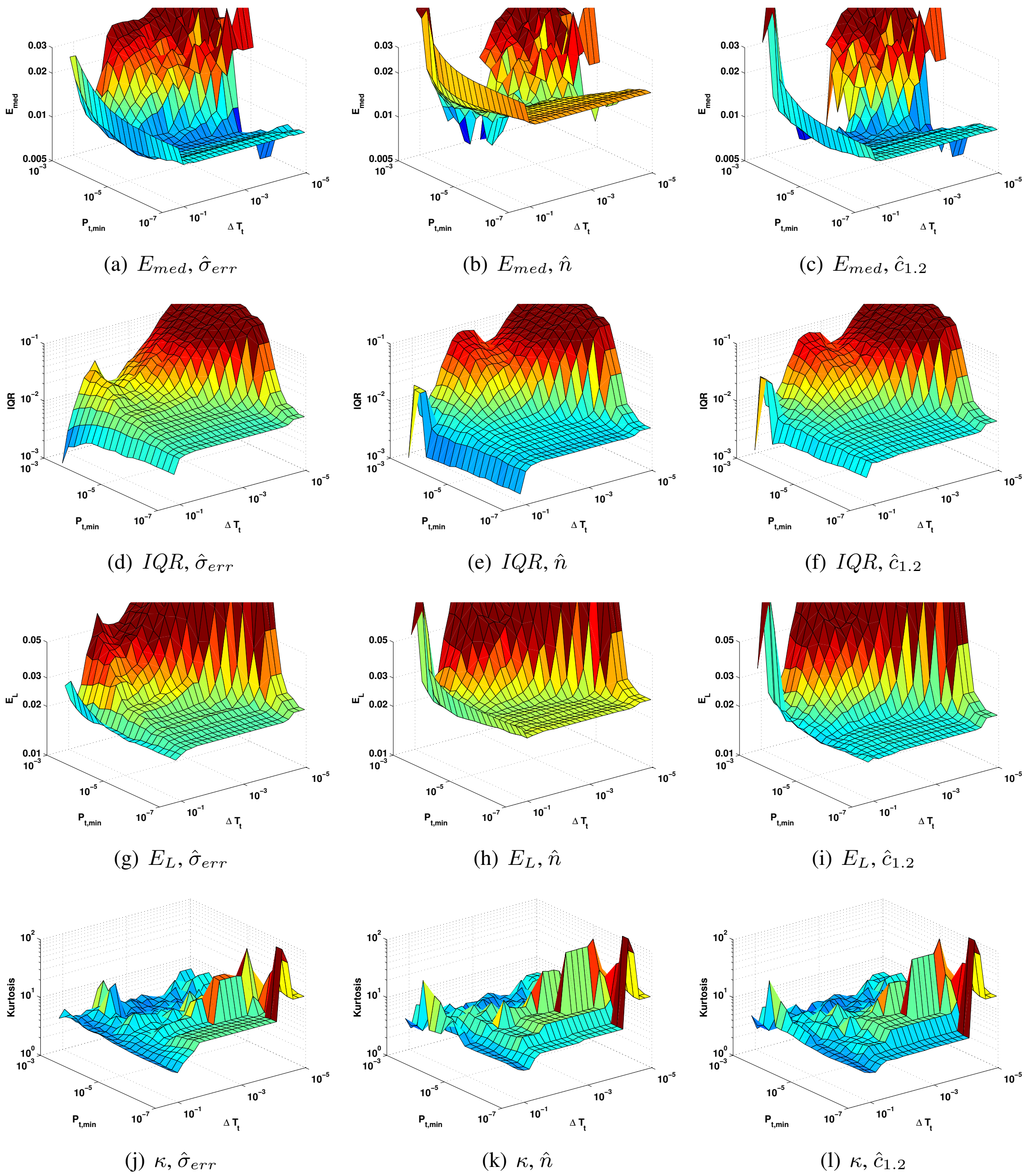
The plots in the first row show the median error E_{med}, which is approximately constant for a broad range of ΔT_t values and decreases toward higher values of $P_{t,min}$, until a critical $P_{t,min}$ value is reached. Here the results become highly unstable unless the selected ΔT_t is close to the optimum Gaussian tail length of the given test distribution, which is ΔT_t≈0.1UI. For larger values of ΔT_t the median error again increases.

The interquartile range IQR in the subfigures of the second row remains constant for a large range of ΔT_t and reaches a minimum at the correct Gaussian tail length. The IQR increases toward higher $P_{t,min}$ values, which confirms the importance of bathtub samples at lowest probability levels. In fact, the tail fitting method looses accuracy when these samples are cut off by the probability threshold $P_{t,min}$. The interquartile range also highlights a large influence on the estimation loss E_L, which has been calculated according to the definition from equation (3.18), and is depicted in the third row of figure 4.22.

In the bottom row, the kurtosis surfaces display the outlier behavior of the scaled Q-normalization method. Again, optimum kurtosis κ is only achieved with a known tail length. As soon as the probability threshold discards parts of the measured tail ($P_{t,min}$>0), κ increases significantly, and thus, outliers appear.

Considering that both E_L and kurtosis have to be minimized, the best performance for all three fitness measures is achieved with $P_{t,min}$=0 and ΔT_t≈0.1 UI. Note that this parameter configuration can only be used for the given test distribution but not for other shapes. This means, a priori knowledge of the ideal tail length is needed in order to obtain optimum fitting results.

When comparing the three fitness measures $\hat{\sigma}_{err}$, $\hat{n}$ and $\hat{c}_{1.2}$, the least statistical spread is achieved with $\hat{n}$ at $P_{t,min}$=0 (figure 4.22(e)). However, the overall estimation loss E_L behaves significantly better with $\hat{c}_{1.2}$ and thus, achieves best overall performance with $\hat{c}_{1.2}$ optimization.



(a) E_{med}, $\hat{\sigma}_{err}$

(b) E_{med}, $\hat{n}$

(c) E_{med}, $\hat{c}_{1.2}$

(d) IQR, $\hat{\sigma}_{err}$

(e) IQR, $\hat{n}$

(f) IQR, $\hat{c}_{1.2}$

(g) E_L, $\hat{\sigma}_{err}$

(h) E_L, $\hat{n}$

(i) E_L, $\hat{c}_{1.2}$

(j) κ, $\hat{\sigma}_{err}$

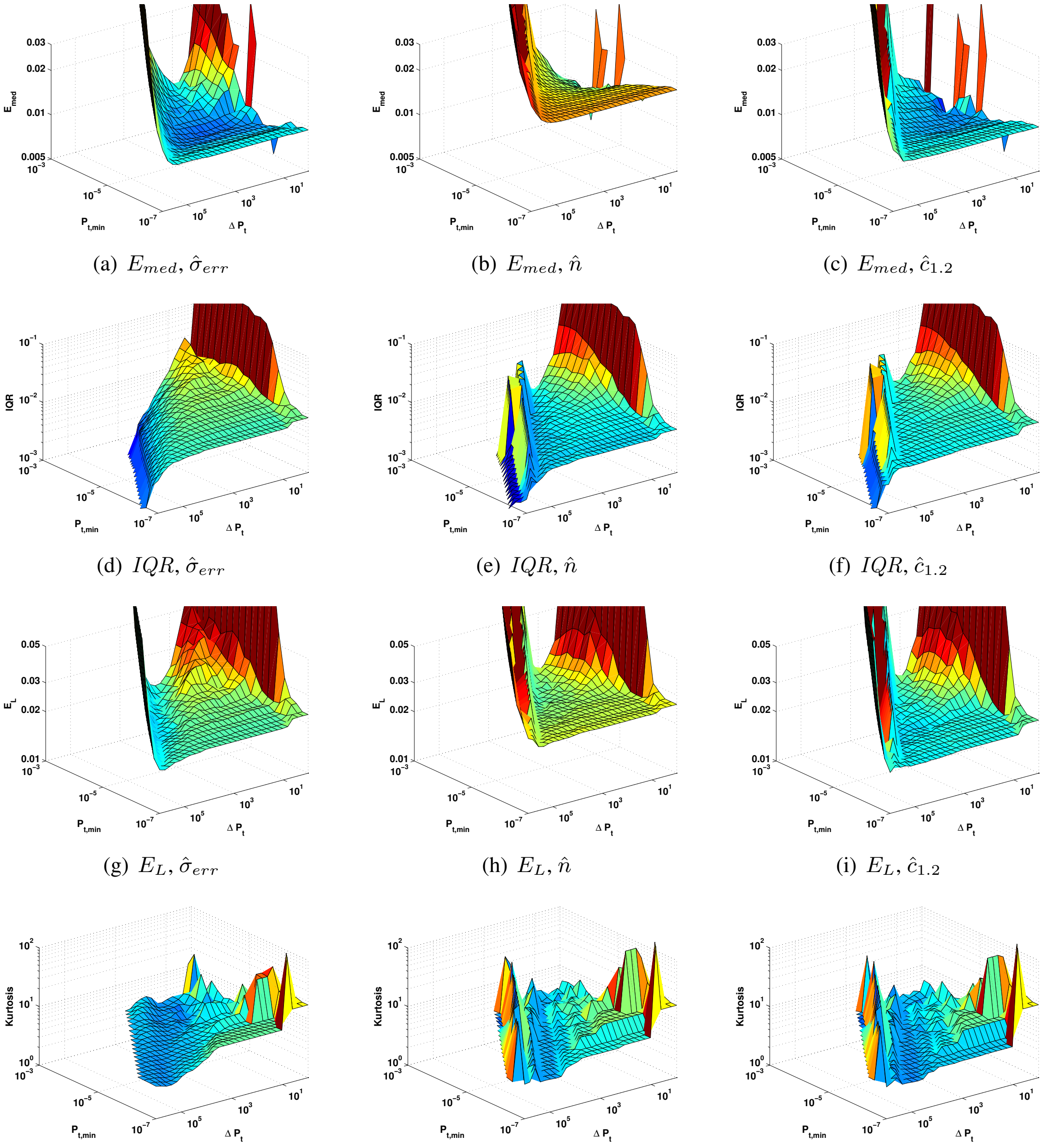
(k) κ, $\hat{n}$

(l) κ, $\hat{c}_{1.2}$

Figure 4.22.: E_{med} (top), IQR, E_L, and kurtosis κ (bottom) for ΔT_t versus $P_{t,min}$ parameter surfaces. Three optimization criteria $\hat{\sigma}_{err}$ (left), $\hat{n}$ (middle) and $\hat{c}_{1.2}$ (right) are investigated. Test distribution: σ_{RJ}=0.05 UI, $A_{DJ,uni}$=0.2 UI, N=10^7, K=250.

Influence of $\Delta \mathrm{P_t}$ and $\mathrm{P_{t,min}}$

ΔP_t defines a probability region or interval where bathtub samples at lowest probability levels must be included for regression analysis. Distributions with a small RJ component yield steep bathtub tails which are tracked by few tail samples, while for the RJ dominant case many tail samples are needed. However, for both cases the Gaussian probability region is approximately constant. The parameter ΔP_t instead of ΔT_t thus, offers a much more robust way of initial tail selection, without prior knowledge of the distribution shape. Further, random variations at the



(a) E_{med}, $\hat{\sigma}_{err}$ (b) E_{med}, $\hat{n}$ (c) E_{med}, $\hat{c}_{1.2}$

(d) IQR, $\hat{\sigma}_{err}$ (e) IQR, $\hat{n}$ (f) IQR, $\hat{c}_{1.2}$

(g) E_L, $\hat{\sigma}_{err}$ (h) E_L, $\hat{n}$ (i) E_L, $\hat{c}_{1.2}$

(j) κ, $\hat{\sigma}_{err}$ (k) κ, $\hat{n}$ (l) κ, $\hat{c}_{1.2}$

Figure 4.23.: E_{med} (top), IQR, E_L and κ (bottom) for ΔP_t versus $P_{t,min}$ parameter surfaces. Three optimization criteria $\hat{\sigma}_{err}$ (left), $\hat{n}$ (middle) and $\hat{c}_{1.2}$ (right) are investigated. Test distribution: σ_{RJ}=0.05 UI, $A_{DJ,uni}$=0.2 UI, N=10^7, K=250.

outermost tail endings, which are caused by the probability granularity $1/N$, can directly be faced by choosing ΔP_t sufficiently large.

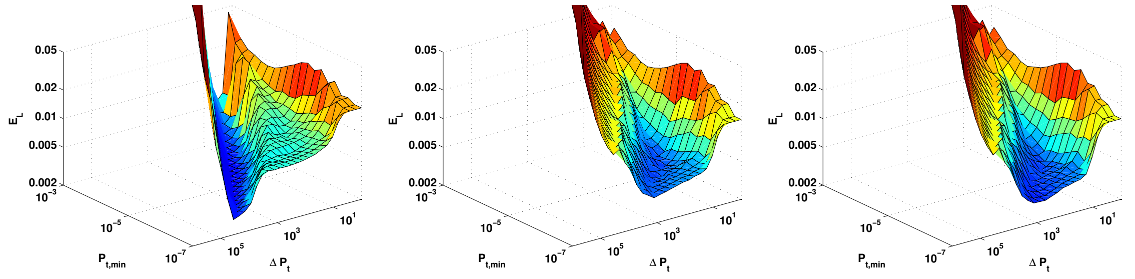
In figure 4.23 again the fitting behavior of the scaled Q-normalization method is analyzed, this time as a function of the parameters ΔP_t and $P_{t,min}$. The selected test distribution and performance measures are the same as in figure 4.22.

For all three optimization criteria $\hat{\sigma}_{err}$, $\hat{n}$ and $\hat{c}_{1.2}$ a convergence limit for the variable product $\Delta P_t{\cdot}P_{t,min}$=const is noticed. It is given by the Gaussian tail part of the test distribution. An upper probability level P_{up} for tail selection can be defined according to this product:

$$P_{up} = \Delta P_t \cdot P_{t,min} \ll A_{t,min} \tag{4.18}$$

P_{up} must belong to the Gaussian part of the distribution tail in order to guarantee for correct fitting, and hence, it must be significantly smaller than the minimum Gaussian tail amplitude $A_{t,min}$. Since a Q-tail approaches the linear behavior asymptotically, such an exact probability level cannot be determined but has to be approximated, as will be demonstrated later in section 5.2. When P_{up} is located in the convergence region well below the tail amplitude $A_{t,min}$, a similar behavior of E_{med}, IRQ and E_L is noticed as with the analysis of ΔT_t before. E_{med} basically decreases toward higher values of $P_{t,min}$, but the interquartile range IQR and also the overall estimation loss E_L increase. For all three optimization criteria $\hat{\sigma}_{err}$, $\hat{n}$ and $\hat{c}_{1.2}$ a minimum estimation loss E_L and smallest kurtosis κ is obtained, if also the threshold parameter $P_{t,min}$ is at its minimum.

Estimation performance depending on the parameters ΔP_t and $P_{t,min}$ must also be investigated with respect to varying distribution shape σ_{RJ}/A_{DJ}. The resulting estimation loss surfaces for σ_{RJ}/A_{DJ}=$1/256$ are plotted in figure 4.24 and demonstrate a behavior which is similar to the prior figures. Besides less estimation error in the convergence region which is due to a better suited distribution shape, the convergence limit has now also moved toward a lower $\Delta P_t{\cdot}P_{t,min}$ variable product because of the smaller Gaussian tail amplitude.



(a) (b) (c)

Figure 4.24.: E_L for parameter surfaces ΔP_t versus $P_{t,min}$, with $\hat{\sigma}_{err}$ (a), $\hat{n}$ (b) and $\hat{c}_{1.2}$ (c) optimization criterion. Test distribution: σ_{RJ}/A_{DJ}=$1/256$, $A_{DJ,uni}$=0.2, σ_{RJ}=0.00078125, N=10^7, K=250.

This shows, that it is not possible to specify an ideal parameter configuration which yields best fitting performance for the general case of arbitrary distribution shapes. Instead, one has to rely on a suboptimal configuration, which guarantees for correct algorithmic convergence and minimizes the influence on estimation performance. From the E_L surfaces in figures 4.23 and 4.24 at least $\Delta P_t{\geq}10^2$ is recommended to afford correct tail fitting. Further, without $P_{t,min}$ the kurtosis of estimates demonstrates a minimum outlier presence, combined with a maximum selection range for ΔP_t. Thus $P_{t,min}$ can be omitted and replaced with the minimum probability granularity $P_{t,min}$=$1/N$ instead. Equation (4.18) is thus reformulated with the condition:

$$A_{t,min} \gg P_{up} = \Delta P_t / N \tag{4.19}$$

As long as the upper probability level for tail selection P_{up} is located significantly below the minimum Gaussian tail amplitude $A_{t,min}$, jitter distributions will be fitted correctly. If ΔP_t is chosen too large, fitted tails are forced to include samples from the DJ component, which misleads the extrapolation result. With a given minimum amplitude $A_{t,min}$ it is thus possible to define a suitable parameter range for ΔP_t. As a general selection criterion, ΔP_t should be chosen as large as possible, but without violating equation (4.19). In section 5.2 design aspects for jitter analysis systems will be investigated, and analyses will focus on ideal tail selection with more detail, especially on a relation between $A_{t,min}$ and the test distribution shape.

To summarize these results, best tail fitting behavior for the scaled Q-normalization (sQN) method is obtained, when using the fitness criterion $\hat{c}_{1.2}$, together with ΔP_t as conservative parameter for tail selection. The other two parameters ΔT_t and $P_{t,min}$ can be discarded. A suited selection range for ΔP_t is given with:

$$A_{t,min} \cdot N \gg \Delta P_t \geq 10^2 \tag{4.20}$$

Further, ΔP_t should always be as large as possible, which yields the default values

$$\Delta P_t = \begin{cases} 10^3 & \text{if} \quad N \geq 10^6 \\ N/10^3 & \text{if} \quad N \leq 10^6 \end{cases} \tag{4.21}$$

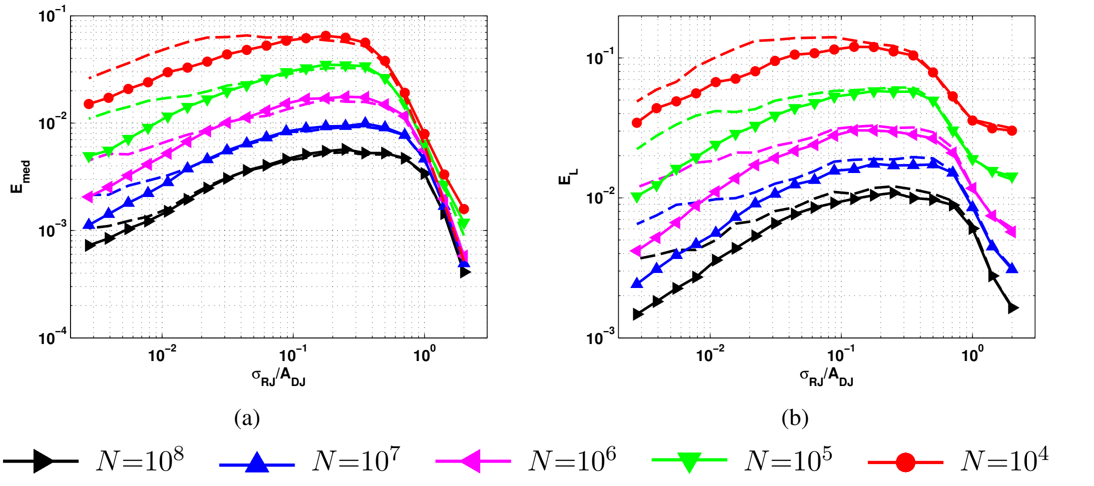
These default values for ΔP_t are chosen such that the upper tail selection bound $\Delta P_t/N$ is located significantly far above the probability granularity $1/N$ of collected jitter distributions and thus, avoids misleading outliers. Even if distribution tails are very flat (RJ dominant case) and suffer from statistical noise, ΔP_t=10^3 guarantees for correct convergence if N≥10^6. However at small sample sizes, ΔP_t must also be selected smaller, in order to fulfill condition (4.19). The minimum Gaussian amplitude $A_{t,min}$ which can be fitted correctly by the sQN method, thus, must be significantly larger than

$$A_{t,min} \gg 10^{-3} \tag{4.22}$$

The investigated test distributions from section 3.2.2 violate this amplitude minimum only for extreme cases with triangular and quadratic curve shaped DJ. Further, for N=10^4 we only have ΔP_t=10^1 and thus, partly accept outliers. The default ΔP_t configuration is applied subsequently.

4.3.3. Performance Analysis with Optimized Parameters

With the optimized tail search criterion $\hat{c}_{1.2}$ and the fitting parameter ΔP_t estimation performance of the sQN method can be re-evaluated and compared with the simple $\hat{\sigma}_{err}$ optimization from section 4.2. Note, that this prior analysis was already carried out with ΔP_t=10^1, in order to



(a) (b)

Figure 4.25.: E_{med} and E_L performance of $\hat{c}_{1.2}$ based optimization over varying $\sigma_{RJ}/A_{DJ,uni}$ and sample size N. N=$\{10^4, 10^5, 10^6, 10^7, 10^8\}$.

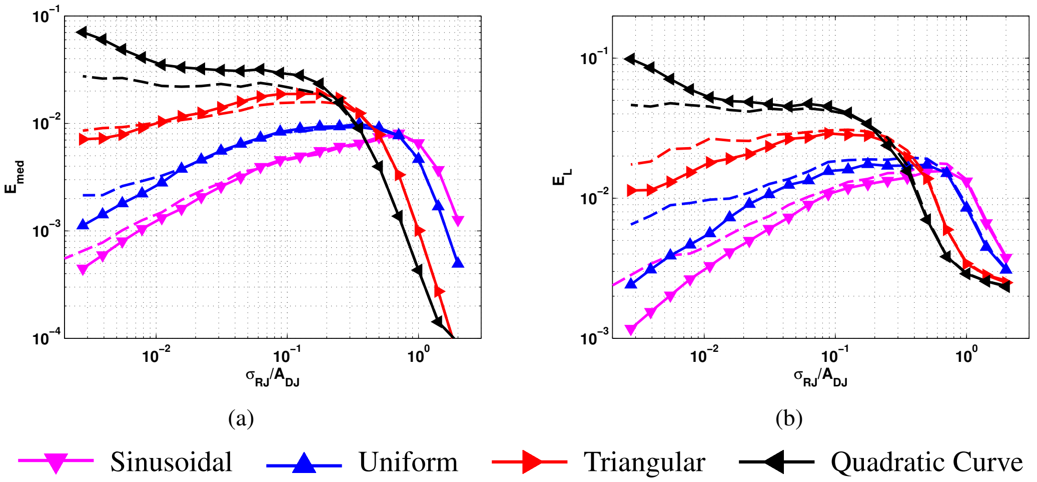


Figure 4.26.: E_{med} and E_L performance of $\hat{c}_{1.2}$ based optimization over varying σ_{RJ}/A_{DJ} and DJ shape: sinusoidal, uniform, triangular and quadratic curve. N=10^7.

afford evaluations without causing too many outliers. As figure 4.25 shows, the median estimation error E_{med} with optimized parameters usually performs slightly worse than with prior tail search (expressed by the dotted lines), but the estimation loss E_L is improved due to less statistical spread combined with complete outlier suppression. In fact, the kurtosis (not shown in the figures) of the optimized search criterion has become significantly smaller.

If again the DJ shape is varied, as depicted in figure 4.26, a similar performance improvement for E_L is noticed. With quadratic curve shaped DJ an error peaking in the lowest σ_{RJ}/A_{DJ} region is additionally noticed, which is due to wrong tail fits. In this region, bathtub tails become extremely steep and $A_{t,min}$ reaches below the bound given by (4.19). The fitting algorithm is thus not able to detect the correct tail amplitude with the given bathtub samples anymore, and rather converges toward the Gaussian-like DJ shape instead of the steep RJ component.

4.4. Performance Optimization with Q-Domain Threshold

The previous section focused on optimizing fitness measures and convergence behavior with different fitting parameters. These parameters defined a lower probability region for tail fitting and thus, directly operated on the measured probability function or bathtub. In this section fitting parameters are investigated in scaled Q-domain. Therefore a Q-domain threshold parameter is introduced and its fitting performance analyzed over a suited parameter range. Equivalent to the previous section, a comparison with the simple $\hat{\sigma}_{err}$ based optimization from section 4.2 highlights performance improvements.

4.4.1. Minimum Q-Domain Threshold Definition

As described in section 4.1.1, in scaled Q-domain a bathtub is represented by bent Q-tails, obtained from the weighted Q-function in equation (4.3). Each Q-tail corresponds to the Gaussian quantile normalization for a specific amplitude, represented by the scaling factor k. In figure 4.27 the Q-tails of a right bathtub curve are plotted for different scaling factors. As a conservative fitting parameter, the minimum Q-domain threshold Q_{min} may be defined to denote an outer tail part for tail fitting. One can easily see, that for a specific scaling factor the Q-tail achieves best linearity in the lower interval defined by Q_{min}. Together with goodness-of-fit measures, Q_{min} can thus also be used to assist in Gaussian model parameter search.

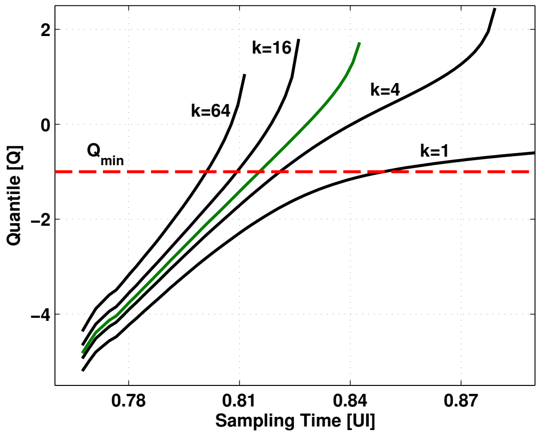


Figure 4.27.: Q_{min} threshold parameter definition.

In Q-domain, functions are described in terms of the standardized variable $q{=}(x-\mu)/\sigma$ (3.6), which leads to a simple regression model for Gaussian tails represented as straight line. The variable q thus also indicates the number of standard deviations one moves away from the mean value of the Gaussian model. This means, the Q_{min} threshold defines the tail part as distance from the Gaussian model mean in terms of its standard deviation. As a direct consequence, the fitting region is automatically adjusted according to the expected Gaussian model.

The example in figure 4.27 uses a Q-domain threshold of $Q_{min}{=}{-1}$ equal to one standard deviation. This value corresponds to the lower inflection point of a Gaussian PDF, located at $x{=}\mu{-}1\sigma$. When searching for the best suited regression line over varying k, it is thus assumed that measured jitter distributions follow a Gaussian function at least up to the inflection point. In scaled Q-domain the search can thus be carried out without knowing the Gaussian model parameters. For $k{=}1$ the measured Q-tail already starts to bend before reaching the $q{=}{-1}$ level, indicating that the tail probabilities increase faster than the expected Gaussian with amplitude $A{=}1/k{=}1$. Equivalently, also for $k{=}64$ the Q-tail does not follow a linear behavior. The optimized tail amplitude is found at $k{=}7.15$ where the linear part even reaches beyond the $q{=}0$ level, which means that more than half of the Gaussian function can be fitted into the measured tail.

Obviously, different values for the threshold parameter Q_{min} are possible. An analysis of the suited parameter range is thus needed and carried out subsequently. Different optimization scenarios and fitness measures also require a performance comparison, in order to identify a best suited configuration. Similar to the $\hat{c}_{1.2}$ criterion with the parameter ΔP_t (section 4.3.2), the Q_{min} threshold also defines a minimum amplitude $A_{t,min}$ and thus, limits the Gaussian tail model search. An according functional relation will be derived in section 5.2. As a beneficial property, Q_{min} allows for the derivation of an exact equation compared to the rough approximation for ΔP_t in equation (4.19).

4.4.2. Parameter Optimization

In this subsection the influence of Q_{min} on estimation performance is investigated. Therefore, the threshold is applied together with two different optimization scenarios. The first one uses Q_{min} as a constant limit and thus, simply defines the lower tail part for linear regression. That is, the fit is only performed in the lowest probability region from the tail end up to Q_{min}. Goodness-of-fit measures such as $\hat{\sigma}_{err}$ or $\hat{T}$ thus only aid in searching for the best suited scaling factor on a predefined tail length. The second scenario uses Q_{min} to define a minimum tail interval, which

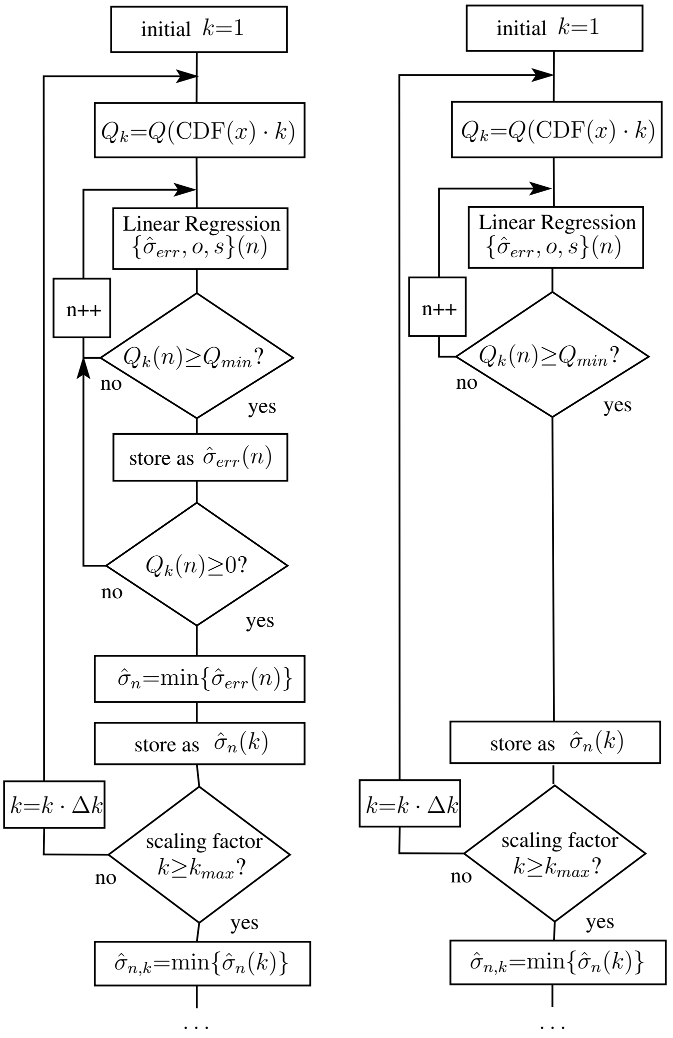
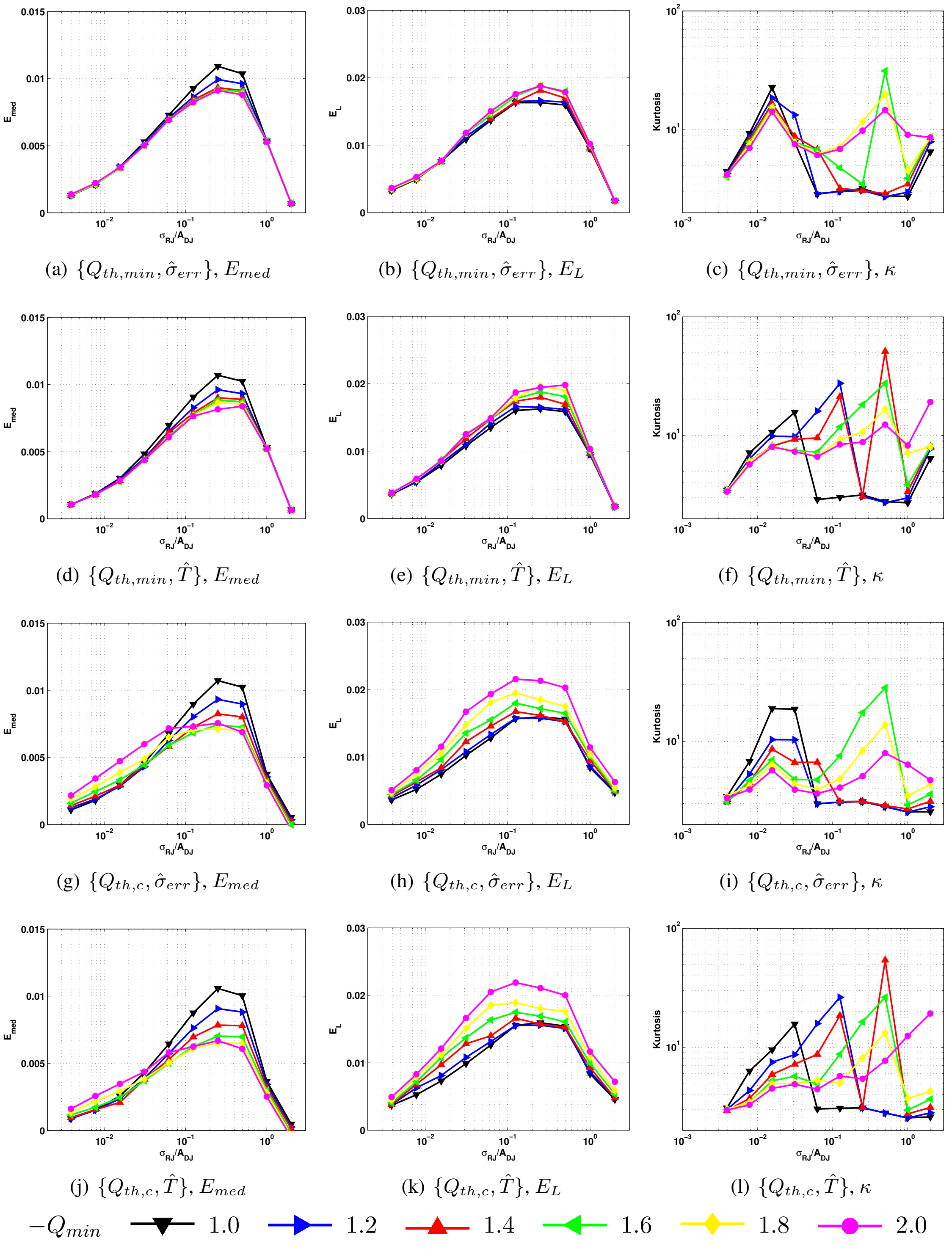

initial k=1
Q_k=Q(CDF(x) · k)
Linear Regression $\{\hat{\sigma}_{err}, o, s\}(n)$
n++
$Q_k(n)$≥Q_{min}?
no
yes
store as $\hat{\sigma}_{err}(n)$
$Q_k(n)$≥0?
no
yes
$\hat{\sigma}_n$=min$\{\hat{\sigma}_{err}(n)\}$
store as $\hat{\sigma}_n(k)$
k=k · Δk
scaling factor k≥k_{max}?
no
yes
$\hat{\sigma}_{n,k}$=min$\{\hat{\sigma}_n(k)\}$
...
initial k=1
Q_k=Q(CDF(x) · k)
Linear Regression $\{\hat{\sigma}_{err}, o, s\}(n)$
n++
$Q_k(n)$≥Q_{min}?
no
yes
store as $\hat{\sigma}_n(k)$
k=k · Δk
scaling factor k≥k_{max}?
no
yes
$\hat{\sigma}_{n,k}$=min$\{\hat{\sigma}_n(k)\}$
...

FIGURE 4.28.: Flow graphs for algorithmic principles based on Q-domain threshold, with minimum $Q_{th,min}$ (left) and constant $Q_{th,c}$ (right) threshold scenario.

also allows for any larger tail lengths. Equivalent to the optimization principles from section 4.3 the best suited tail part is then identified according to the smallest fitness value along the tail.

The flow graphs for these two algorithmic principles are plotted in figure 4.28 and subsequently denoted as $Q_{th,c}$ for constant and $Q_{th,min}$ for minimum threshold analysis. The last blocks have been omitted since they are equivalent to prior algorithms (see figure 4.16). The left graph describes the $Q_{th,min}$ based fitting principle, where Q_{min} is now included into the algorithmic structure. As long as the Q-tail values $Q_k(n)$ are located in the tail region, and thus $Q_k(n)<Q_{min}$ (also see figure 4.27), they will recursively be added by the algorithm but not used for minimum search. As soon as the Q-domain threshold is reached, the minimum search finds the best suited tail length inside the interval $Q_{min}<Q_k(n)<0$. This minimum value has to be identified for every scaling factor k in order to allow for Gaussian amplitude search.

The right graph in figure 4.28 describes the simple $Q_{th,c}$ scenario, where only the initial tail part up to Q_{min} is used for regression analysis. Other tail lengths are not allowed. Note, that both flow graphs only describe the search with an initial coarse search grid, and thus have to be followed by a refinement stage in order to yield the final results. The refinement stages operate with the same optimization criterion as for the respective search grid.



(a) $\{Q_{th,min}, \hat{\sigma}_{err}\}$, E_{med}

(b) $\{Q_{th,min}, \hat{\sigma}_{err}\}$, E_L

(c) $\{Q_{th,min}, \hat{\sigma}_{err}\}$, κ

(d) $\{Q_{th,min}, \hat{T}\}$, E_{med}

(e) $\{Q_{th,min}, \hat{T}\}$, E_L

(f) $\{Q_{th,min}, \hat{T}\}$, κ

(g) $\{Q_{th,c}, \hat{\sigma}_{err}\}$, E_{med}

(h) $\{Q_{th,c}, \hat{\sigma}_{err}\}$, E_L

(i) $\{Q_{th,c}, \hat{\sigma}_{err}\}$, κ

(j) $\{Q_{th,c}, \hat{T}\}$, E_{med}

(k) $\{Q_{th,c}, \hat{T}\}$, E_L

(l) $\{Q_{th,c}, \hat{T}\}$, κ

Figure 4.29.: E_{med} (left), E_L (middle) and kurtosis κ (right) over varying distribution shape σ_{RJ}/A_{DJ}. In the different rows four algorithmic principles with combinations of $\{Q_{th,min}, Q_{th,c}\}$ and $\{\hat{\sigma}_{err}, \hat{T}\}$ are investigated. The Q_{min} threshold is chosen as $-Q_{min}$=$\{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, uniform DJ type, N=10^7, K=250.

Both flow graphs can either use $\hat{\sigma}_{err}$ or $\hat{T}{=}\hat{\sigma}_{err}/s$ as fitness measures to drive the minimum search. Together with the two algorithmic principles $Q_{th,c}$ and $Q_{th,min}$, thus four combinations are obtained which can be compared against each other. Other fitness measures such as $\hat{n}$ or $\hat{n}_T$ cannot be applied, since Q_{min} also affects the regression length over varying scaling factor k.

In figure 4.29 the estimation performance for each of the four algorithm combinations is depicted. The different subfigure rows correspond to these combinations, while the columns describe the respective behavior of median error E_{med}, estimation loss E_L and kurtosis κ over varying distribution shape σ_{RJ}/A_{DJ}. While E_L describes the estimation performance, κ indicates the presence of outliers, as already described in section 4.3.2. Positive peaks in the course of κ indicate strong outliers and have to be avoided as far as possible. Further, each subfigure also investigates the influence of Q_{min} parameter variations by showing different curves for $-Q_{min}{=}\{1.0, \ldots, 2.0\}$.

From the E_{med} curves in the left column of figures 4.29 (a,d,g,j) we notice that the $Q_{th,c}$ scenario has a better influence on error bias compared to $Q_{th,min}$. With decreasing threshold Q_{min}, error bias can be reduced significantly. Unfortunately, this benefit is payed with a higher statistical spread, as can be seen from the E_L plots in 4.29 (b,e,h,k) where the common performance is changed only marginally. Considering the worst case distribution shape at $\sigma_{RJ}/A_{DJ}{=}1/4$, in fact the optimum compromise between error bias and spread can be found in the interval $Q_{min}{\in}[-1.2, -1.0]$.

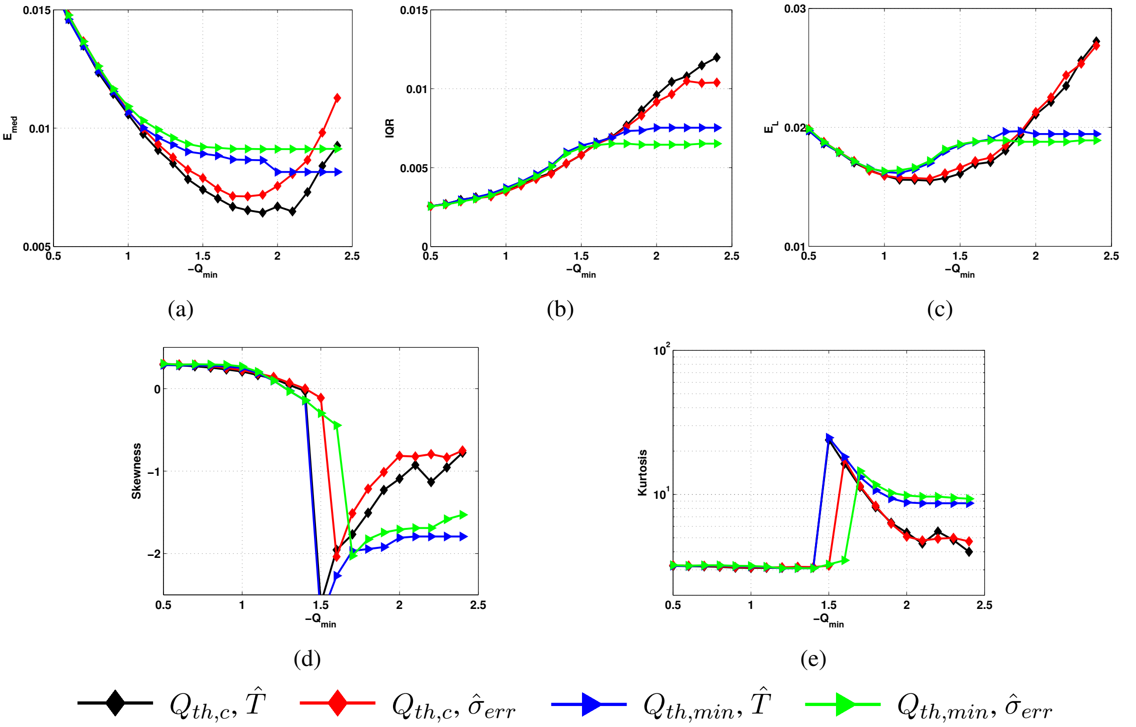
Additionally an increase of kurtosis is observed at largest shape values for $Q_{min}{\leq}-1.4$. This indicates a high uncertainty of the algorithm when choosing a small outermost fitting region. In fact, with RJ dominant shapes the Q-tails become linear over a wide probability range, and thus, the algorithm suffers from random noise at the outer tails.

Comparing the two fitness measures $\hat{\sigma}_{err}$ and $\hat{T}$, the first one tends to produce estimates that are less affected by outliers. This can especially be observed when comparing the kurtosis of figures 4.29(c) and 4.29(i), with 4.29(f) and 4.29(l). Hence, the $Q_{th,c}$ optimization scenario in combination with $\hat{\sigma}_{err}$ as fitness measure offers the best choice out of the four candidates.

In order to get a better impression on how Q_{min} affects estimation performance, in figure 4.30 the four algorithm candidates are compared with respect to varying $-Q_{min}{=}\{0.5, \ldots, 2.4\}$. The selected test distribution is $\sigma_{RJ}/A_{DJ}{=}1/4$ to realize the worst case shape. A visible difference in performance appears as soon as $Q_{min}{\leq}-1.0$. The $Q_{th,c}$ optimization estimates the true TJ value slightly better than $Q_{th,min}$, which can be observed in both E_{med} and E_L figures. The kurtosis of all four algorithms follows a flat course, until at some point a high peak is observed. This peak is negative skewed, indicating that large negative outliers are present in the distribution of TJ estimates.

A closer analysis of these outliers highlights, that the regression stage has converged to the outermost tail part by fitting a Gaussian model of very small amplitude. Due to the small Q_{min} threshold parameter, the algorithm was in fact forced to carry out the regression analysis at the outermost tail part of distributions, which is highly affected by statistical random variations. The further Q_{min} is moved toward the outer tail, the higher the risk that outliers will occur. The only way to avoid such a high risk is thus to increase Q_{min}.

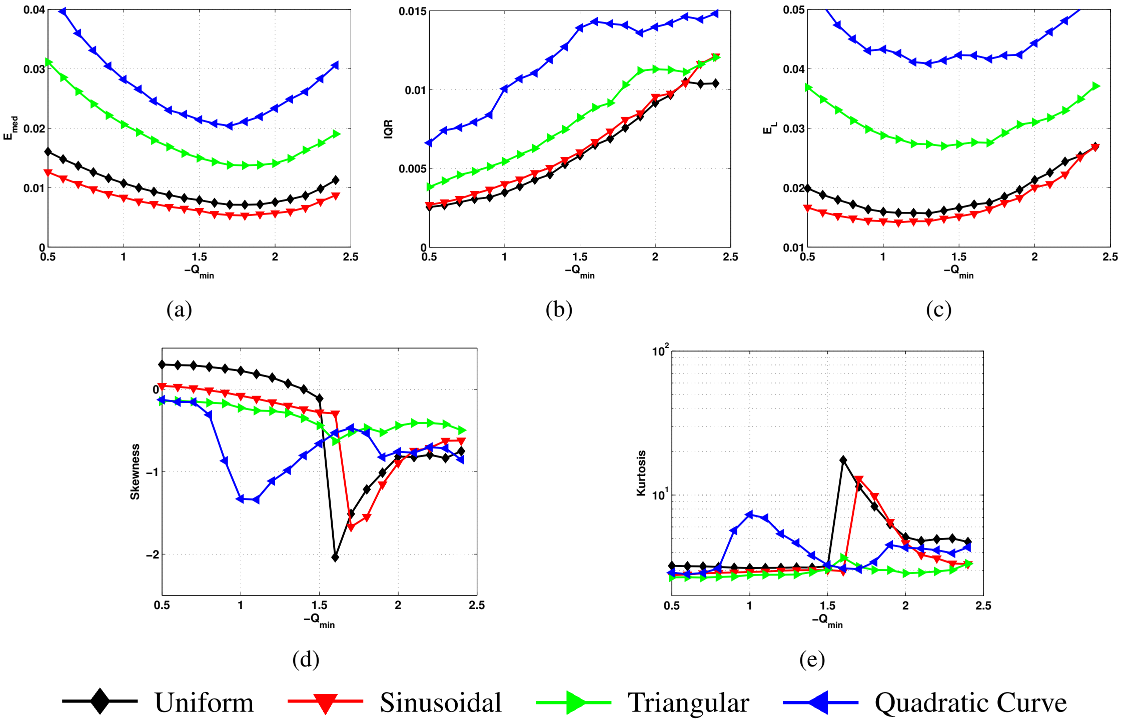
In order to determine an optimum value of Q_{min} which can be used for various distribution shapes, a similar analysis with respect to different DJ types is carried out. Figure 4.31 shows the performance behavior of $Q_{th,c}$ scenario with $\hat{\sigma}_{err}$ fitness measure and different DJ types. A good compromise is found at $Q_{min}{=}-1.2$. This value is located in the minimum region of estimation loss curves 4.31(c), and provides correct tail fitting results at an acceptable risk, as can be seen from the kurtosis in 4.31(e). However, the quadratic curve DJ produces misleading outliers even for $Q_{min}{\geq}-1.2$. The reason is the same as described before. The Q-tails of test distributions have become very steep, so that random variations again easily cause outliers.



(a) (b) (c)

(d) (e)

Figure 4.30.: E_{med} (a), IQR (b), E_L (c), skewness (d) and kurtosis (e) of the four algorithmic principles from 4.29 plotted over $-Q_{min}$=$\{0.5, \ldots 2.4\}$. Test distribution (worst case shape): σ_{RJ}/A_{DJ}=$1/4$, uniform DJ, N=10^7, K=250.



(a) (b) (c)

(d) (e)

Figure 4.31.: E_{med} (a), IQR (b), E_L (c), skewness (d) and kurtosis (e) of DJ types over $-Q_{min}$=$\{0.5, \ldots, 2.4\}$, $Q_{th,c}$, $\hat{\sigma}_{err}$ scenario. Test distributions (worst case): σ_{RJ}/A_{DJ}=$1/4$ (uni.), $1/2$ (sin.), $1/8$ (tri.), $1/16$ (quad.). N=10^7, K=250.

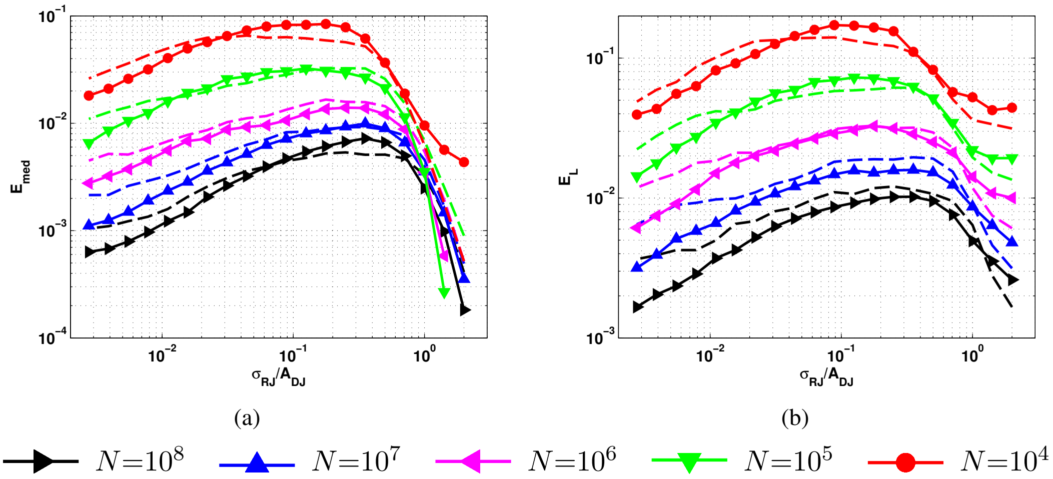


Figure 4.32.: E_{med} and E_L performance for $Q_{th,c}$ criterion with Q_{min}=-1.2 over varying $\sigma_{RJ}/A_{DJ,uni}$ and sample size N. N=$\{10^4, 10^5, 10^6, 10^7, 10^8\}$.

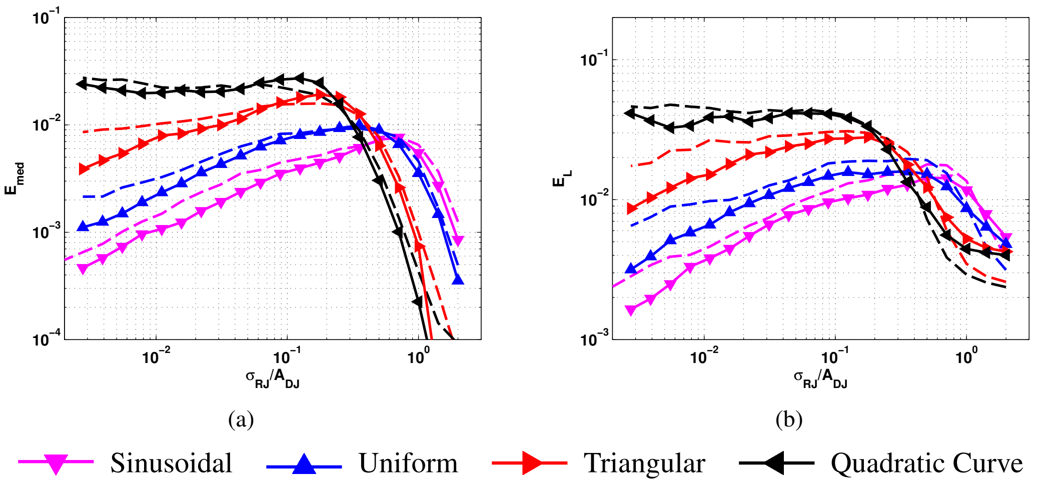


Figure 4.33.: E_{med} and E_L performance for $Q_{th,c}$ criterion with Q_{min}=-1.2 over varying σ_{RJ}/A_{DJ} and DJ type: sin., uni., tri. and quad. curve. N=10^7.

4.4.3. Performance Analysis with Optimized Parameters

Similar to section 4.3, the optimized Q-domain threshold method can also be compared with the original simple $\hat{\sigma}_{err}$ optimization from section 4.2. According to the previous results, the constant Q-domain criterion $Q_{th,c}$ uses a threshold of Q_{min}=-1.2. Figure 4.32 shows, that the median estimation error E_{med} of the threshold method usually performs slightly worse when compared with the original tail fitting method (dotted lines). However, the overall estimation loss E_L is better for $N{\geq}10^7$, and comparable to the $\hat{c}_{1.2}$ based algorithm from section 4.3. For $N{\leq}10^6$ the estimation loss E_L is larger. The Q-domain threshold also improves the performance of all investigated DJ types, as can be seen in figure 4.33.

The error peaking in the lowest σ_{RJ}/A_{DJ} region, which could be observed with the optimized $\hat{c}_{1.2}$ method at quadratic curve shaped DJ (figure 4.26), is not present anymore. The observable minimum tail amplitude $A_{t,min}$ is now given by the search range limit for k_{max}=$1/A_{t,min}$. However, the Q_{min} threshold method is also less robust against outliers.

4.5. Summary

A complete approach to Gaussian tail fitting, referred to as scaled Q-normalization (sQN) was proposed and realized. Therefore, the quantile normalization principle was embedded into a three dimensional optimization scheme for Gaussian model parameter search with unknown tail amplitude A, mean μ and standard deviation σ. Due to efficient recursions (equations (4.5)), this optimization can be performed very fast. The calculation time linearly depends on the selected number of bins R for a jitter distribution, and was investigated in figure 4.9. The algorithm was implemented with C/C++ and yields very accurate TJ_{pp} estimates as the performance analysis in section 4.2 demonstrates. The resulting estimation error bias is always positive and thus pessimistic, which is a further beneficial property of algorithms based on quantile normalization.

Causes for performance degradation of the method can be summarized as follows:

- A small sample size N of collected distributions, as shown in figures 4.14, 4.25 and 4.32.
- The fitting algorithm operates on distribution shapes in the worst case region, for example $\sigma_{RJ}/A_{DJ,uni}{\approx}1/4$ at uniform type DJ.
- The DJ shape is Gaussian-like but bounded, as shown in figures 4.15, 4.26 and 4.33.

Especially the last point leads to jitter distributions where only a very small percentage of the collected samples belongs to the Gaussian tail part. The only way to deal with this problem is to increase the sample size N and thus, to prolongate the Gaussian tail, but as tail samples become rarely with larger jitter amplitude, acquisition time also increases exponentially.

In order to improve the estimation performance as well as the robustness of the sQN method, two algorithmic approaches have been investigated. The first one, $\hat{c}_{1.2}$ in section 4.3, uses conservative fitting parameters in probability domain. It suggests an initial fitting region $\Delta P_t{\geq}10^2$ which covers at least two decades, in order to avoid outliers caused by statistical tail variations. From this principle, a combination which utilizes both regression length and error, has been identified as best suited optimization criterion. According to equation (4.19), the minimum tail amplitude $A_{t,min}$ must be located significantly above the minimum tail fitting region given by $\Delta P_t/N$.

The second approach, $Q_{th,c}$ in section 4.4, uses a constant threshold Q_{min} in scaled Q-domain, which defines the Gaussian tail region in terms of standard deviations beside the model mean. This representation form allows for a flexible choice of the tail part. With a smaller parameter Q_{min}, estimation accuracy is improved, but the risk for outlier occurrence is also increased. As an acceptable compromise $Q_{min}{=}{-}1.2$ has been identified.

Both approaches improve the estimation performance compared to the simple algorithm based on regression error $\hat{\sigma}_{err}$ from section 4.2. For uniform DJ, $N{=}10^6$ and worst case test distributions, the error bias is $<2\%$ and the overall error is generally $<3\%$ in more than 97.5% of the cases. Although the Q-domain threshold scenario seems to slightly outperform the $\hat{c}_{1.2}$ scenario with different DJ types, it clearly performs worse at smaller sample sizes $N{\leq}10^6$ (see figures 4.25(b) and 4.32(b)). In the subsequent chapter this effect is shown to originate from a larger error variation of $Q_{th,c}$ if only the outermost tail part can be used for a fit. Further, a relation between sample size N, minimum tail amplitude $A_{t,min}$ and threshold Q_{min} was derived. This also includes a more detailed comparison of both algorithms.

So far, the proposed sQN method with its two algorithmic versions has only been applied to simulated jitter distributions, where the simulator timing resolution R_{sim} is chosen sufficiently high, so that an influence on estimation performance can be neglected. A coarse time quantization of distributions introduces a limited, discrete amount of bins. This is an inherent property of hardware systems and has to be investigated thoroughly to allow for a hardware implementation of the proposed method. This issue will especially be addressed in the subsequent chapter. Summarized parts of this chapter have also been published in [C1,C8,C9].

5. Hardware Design Aspects

This chapter focuses on hardware related design aspects for the scaled Q-normalization (sQN) method. So far, the proposed approach has only been considered as a pure mathematical optimization scheme, which is applied to simulated distributions or behavioral models to investigate the impact of timing jitter on system performance. As an important application, one also likes to use the method together with real jitter data collected in measurements. However, due to the limited precision, such devices introduce certain quantization effects and include additional error sources the analysis method has to cope with.

The subsequent sections investigate these effects, and try to derive empirical relations or describe the resulting error behavior and limitations. The idea is to give the designer useful hints on how to select key parameters for a jitter measurement system, in order to guarantee a certain target accuracy and robustness. Thus, a complete design guideline for the sQN method from chapter 4 is provided.

A key advantage of jitter distributions is that they can be collected on-chip using built-in jitter measurement (BIJM) systems. With data frequencies in the GHz range, off-chip instrumentation may include significant noise caused by interconnect wires, which can easily affect measurements. In such cases, the impact of external noise can be eliminated with a direct on-chip measurement. Hence, in recent years a broad variety of BIJM topologies and implementations [16–18, 35, 48, 57, 60, 65, 66, 79, 100, 129, 131] has been realized as design-for-test (DFT) structures to support production tests as well as on-chip diagnostics [12–14, 86, 130]. Although subsequent analyses for the scaled Q-normalization method are basically valid for both instrumentation devices and BIJM systems, the focus of this chapter is more on future oriented on-chip applications.

The timing jitter of a PLL is analyzed by measuring IO jitter, defined as the time difference between reference frequency and PLL output clock (section 2.1.2). In the case of serial high-speed interfaces with a clock and data recovery (CDR) unit, the reference frequency is given by the analog input data, and jitter values are measured between bit transitions of the input signal and the PLL output clock. In order to correctly quantify IO jitter, thus, an exact time interval measurement has to be performed. Such measurements are typically realized with an adjustable delay element inside the PLL under test, as depicted in figure 5.1. A binary phase detector (PD) compares the delayed output clock against the transition edges of the serial data stream. If the clock precedes the data edge, a logical one is created at the output, otherwise a zero. After N clock cycles, the counter state reflects the jitter probability for the selected delay value. One can sequentially step the delay over the whole bit period and thus, measure the complete probability function of IO jitter. The smallest step size defines the achievable number of bins R in a unit interval (UI), or time resolution $1/R$.

A significant speed-up of the measurement is achieved if up to R PDs and counters are used in parallel, together with a delay line [16, 66]. Further, the PD can be replaced by a simple D flip-flop or data latch if the input data is known. That is, the recovered data from the flip-flop is compared against the expected original data, which again allows for detecting errors. However, the simple BIJM principle in figure 5.1 measures only one bathtub point at a time. With N received bits the probability value for the selected delay is tracked down to the BER level $1/N$, which also forms the probability granularity of the measurement. A complete bathtub measurement must be

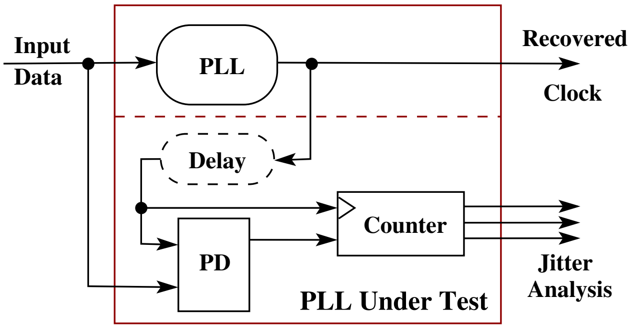


Figure 5.1.: BIJM based IO jitter measurement for PLLs.

performed over R time steps, which yields the test time

$$t_t[\mathrm{UI}] = N \cdot R \tag{5.1}$$

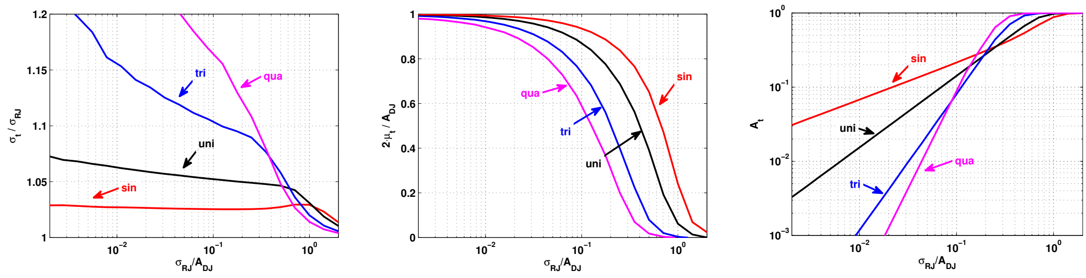
and also highlights the key problem for time critical measurements. While R can be reduced with parallel circuit structures, the sample size N introduces a fundamental time limit for the reachable probability depth. Further, mismatches in the delay line, as caused by process variations and the non-ideal PD structure are additional sources for timing error. They can significantly affect the accuracy of a fitting method and are described by a differential nonlinearity (DNL) error term. Thus, the estimation error is basically influenced by the three parameters N, R and DNL error.

This problem domain is addressed in the subsequent sections. First, the tail parameters of test distributions from section 3.2.2 are characterized, in order to relate them with a certain distribution shape. This allows to specify minimum requirements for jitter analysis. Subsequently, empirical relations for minimum sample size N and minimum time resolution R are derived. Since both parameters have a significant influence on the extrapolation error of a fitting method, the focus is also put on empirical relations for error prediction. In addition, a model is provided to consider the DNL error caused by process variations. Finally, two comprehensive design examples are given to demonstrate applicability of the empirical relations and to provide design guidelines for jitter measurement systems. The chapter concludes with a brief summary.

5.1. Tail Parameters of Test Distributions

As was shown in section 3.2.2, test distributions are synthesized using RJ and DJ components, and can be fully described in terms of the jitter ratio σ_{RJ}/A_{DJ}. However, in a practical design one is also interested in the tail characteristic of a total distribution. Often, typical tail parameters are well known from measurements, and the goal of a jitter analysis is to fulfill certain minimum requirements. Thus, one likes to provide relations between the shape variables σ_{RJ} and A_{DJ}, and fitted tail parameters A_t, σ_t and μ_t, obtained by the fitting method.

With the convolution of RJ and DJ components in histogram domain (figure 3.5), generally an increase of the RJ variance ($\sigma_t \geq \sigma_{RJ}$) as well as a decrease of the DJ amplitude ($\mu_t \leq A_{DJ}/2$) [52, 123] is observed. Additionally, the tail amplitude A_t is noticed with the sQN method. Unfortunately, tail parameters cannot be determined exactly, not even with ideal bathtub curves where $\{N,R\} \rightarrow \infty$. The problem is that a Q-tail always approaches the Gaussian line asymptotically, even with optimized scaling factor k (see figure 4.4). An optimum k only guarantees for best linearity of the obtained Q-tail, but still suffers from this asymptotic effect. In fact, tail fitting algorithms have to deal with a badly situated optimization problem, since it is not possible to specify where the asymptote or Gaussian tail part exactly begins. In other words, the fitted tail parameters always depend on the selected sample size N and number of bins R.



(a) (b) (c)

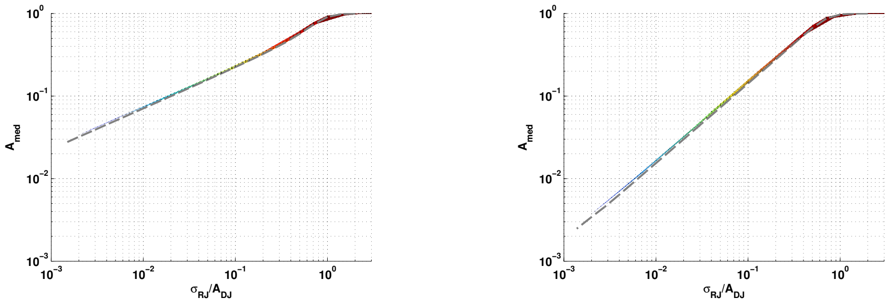
FIGURE 5.2.: Fitted test distributions: tail parameters standard deviation σ_t (a), mean μ_t (b), and amplitude A_t (c) over varying shape σ_{RJ}/A_{DJ}, N=10^8, R=R_{sim}=3.3·10^5.

A possible way to deal with this problem is a fast numerical tail approximation that reflects the realistic behavior of the sQN method. Therefore, ideal test distributions are created by convolving RJ and DJ components, and bathtub tails are simulated down to the target BER=10^{-12}, so that the noise of random variations is suppressed. Then quantization effects of N and R are re-introduced. The applied fitting method thus yields the same tail parameters which otherwise have to be estimated from median values of time consuming, statistical evaluations. This allows to quickly characterize the average fitting behavior of the method. By applying this principle, in the subsequent sections thus, empirical relations for tail parameters are derived.

5.1.1. Relation Between Distribution Shape and $\mathrm{A_t}$

In order to correctly describe the tail amplitude as a function of distribution shape, one has to deal with the asymptotic tail behavior. For the moment the influence of time resolution is suppressed by choosing R=R_{sim}. As an example, in figure 5.2 fitting results for the three tail parameters σ_t, μ_t and A_t are presented at N=10^8. As expected, the obtained RJ standard deviation σ_t (figure5.2(a)) is always larger than σ_{RJ}, while μ_t (figure5.2(b)) is smaller than $A_{DJ}/2$. Unfortunately, both parameters highly depend on the selected N and R, and follow a non-linear course which impedes derivation of simple empirical relations. For the estimated σ_t values in figure 5.2(a) and triangular or quadratic DJ types even numerical inaccuracies are observed, they result from the probability granularity $1/N$ at the outermost tail region. However, the fitted tail amplitude A_t (figure 5.2(c)) has a linear characteristic over a large range of shape values. Thus, one can use linear regressions that aid in specifying minimum requirements for amplitudes. As will be shown in the subsequent section 5.2, a limited sample size N also imposes a minimum analyzable tail amplitude $A_{t,min}$. Therefore, the curves in figure 5.2(c) can be used to relate $A_{t,min}$ with a corresponding distribution shape or vice versa. It is important to note that these curves represent pessimistic results for an arbitrary time resolution R. In fact, the practical case with a limited R always leads to larger amplitudes than the ones given in figure 5.2(c). Thus, the curves approximate the R→∞ case, and only depend on the sample size N.

For comparison, the numerical tail approximations are empirically verified, as shown in figure 5.3. Therefore, median tail amplitudes of fitted test distributions are determined with respect to varying σ_{RJ} and A_{DJ}, as well as sinusoidal (5.3(a)) and uniform (5.3(b)) DJ types. The dashed lines mark the results obtained by fast numerical approximations, and correctly match the observed median amplitudes. This means, the approximations truly reflect to a large extend the average behavior of the fitting method. Note, that the observed amplitude curves are in fact planes. Median amplitudes are always constant for a certain jitter ratio σ_{RJ}/A_{DJ} and thus, allow for a simplified one-dimensional representation.



(a) Sinusoidal DJ

(b) Uniform DJ

Figure 5.3.: Comparing the median of tail amplitude A_t over varying jitter ratio σ_{RJ}/A_{DJ} with fast numerical approximations (dashed lines). N=10^8, K=250, ΔP_t=10^5.

DJ	N = 10^5		N = 10^6		N = 10^7		N = 10^8		N = 10^9	
	a	**b**	**a**	**b**	**a**	**b**	**a**	**b**	**a**	**b**
Sin.	0.766	0.492	0.729	0.495	0.700	0.497	0.673	0.497	0.652	0.498
Uni.	1.489	0.926	1.389	0.945	1.292	0.955	1.209	0.962	1.136	0.966
Tri.	5.566	1.619	5.462	1.723	5.251	1.793	4.826	1.832	4.420	1.863
Qua.	19.789	2.022	23.758	2.280	26.467	2.469	31.949	2.639	22.187	2.639

Table 5.1.: Coefficients for equations (5.3) and (5.4), with sQN $\hat{c}_{1.2}$ algorithm, R=R_{sim}.

The four curves in figure 5.2(c) follow a linear behavior over a large range of jitter ratios, until the tail amplitude is close to the maximum value A_t=1. An empirical relation between tail amplitude and jitter ratio can thus be obtained using simple linear functions. For the resulting regression coefficients a logarithmic scaling of both axes must additionally be considered, thus:

$$\begin{aligned} \ln(y) &= a_L + b\cdot\ln(x), && \text{with} \quad y{=}A_t,\ x{=}\sigma_{RJ}/A_{DJ} \\ &= \ln(e^{a_L}\cdot x^b), && \text{with} \quad a = e^{a_L} \\ \Rightarrow y &= a\cdot x^b \end{aligned} \tag{5.2}$$

When re-inserting the original variables we yield

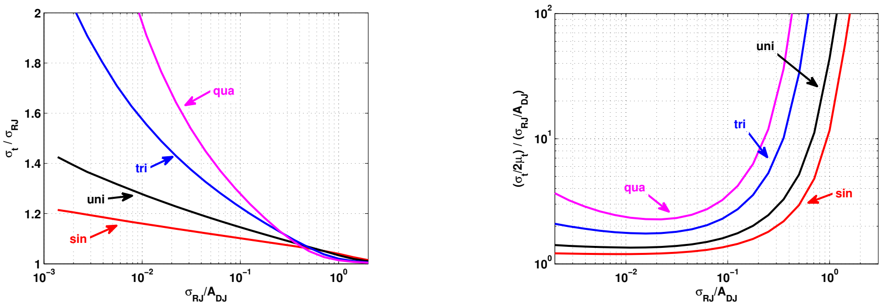
$$A_t = a\cdot\left(\frac{\sigma_{RJ}}{A_{DJ}}\right)^b \tag{5.3}$$

with the inverse

$$\frac{\sigma_{RJ}}{A_{DJ}} = \left(\frac{A_t}{a}\right)^{1/b} \tag{5.4}$$

The regression coefficients for each of the four DJ types as well as N=$\{10^5,\dots,10^9\}$ are given in table 5.1. These are valid for the sQN method with $\hat{c}_{1.2}$ fitting criterion, which achieves best performance, as will be demonstrated later in section 5.4. Regressions are carried out with fifty shape values in an equidistant grid interval σ_{RJ}/A_{DJ}=$[10^{-3}, 10^0]$, where only the obtained amplitudes in the range A_t=$[10^{-3}, 3{\cdot}10^{-1}]$ are used, in order to avoid non-linearities of the probability limit $1/N$ and the uppermost amplitude region. The given sample sizes represent important candidates for a typical BIJM system design.

With a given test distribution (σ_{RJ}, A_{DJ}), the coefficients allow to specify the tail amplitude A_t obtained with the sQN method. In section 5.2 this will especially aid in determining minimum requirements for amplitudes. Note, that if N is unknown or variable, one can still use the largest



(a)

(b)

Figure 5.4.: Normalized standard deviation $\frac{\sigma_t}{\sigma_{RJ}}$ (a) and jitter ratio $\frac{\sigma_t/2\mu_t}{\sigma_{RJ}/A_{DJ}}$ (b) of fitted test distributions over varying shape σ_{RJ}/A_{DJ}, $A_t{\equiv}1$, $N{=}10^8$, $R{=}R_{sim}{=}3.3{\cdot}10^5$.

possible value of N to obtain a pessimistic amplitude. That is, fitted tail amplitudes become smaller with increasing N, which is also an effect of the asymptotic tail behavior in Q-domain.

5.1.2. Relation Between Distribution Shape and σ_t, μ_t

As was already mentioned in the previous analysis, fitted standard deviation σ_t and mean μ_t behave highly non-linear over varying N and R. Nevertheless, it is also possible to describe empirical relations for these tail parameters. Section 5.3 will show, that the sQN method behaves equivalent to the conventional Q-normalization (QN) method without scaling ($A_t{\overset{!}{=}}1$) for the extreme case of a very coarse time resolution R. That means, worst case fitting parameters can be provided which are generally valid for both methods. Therefore only the tail amplitude must be discarded. In figure 5.4 the standard deviation σ_t and the jitter ratio $\sigma_t/(2\mu_t)$ of fitted tails with $A_t{\overset{!}{=}}1$ are represented with respect to the four different DJ types. These curves now allow to specify minimum requirements of fitted tail parameters, which can be related to the original variables σ_{RJ} and σ_{RJ}/A_{DJ} prior to convolution. This is especially useful when specifying a minimum standard deviation $\sigma_{t,min}$ as done in section 5.3 later on.

With higher order polynomials for regression analysis, we yield the following two relations:

$$y = a_0 + a_1 \cdot x + \ldots + a_p \cdot x^p, \quad x = \ln(\sigma_{RJ}/A_{DJ}) \tag{5.5a}$$

$$y = \begin{cases} \ln(\sigma_t/\sigma_{RJ}) & \text{(figure 5.4(a))} \quad (5.5\text{b}) \\ \ln\left(\dfrac{\sigma_t/(2\mu_t)}{\sigma_{RJ}/A_{DJ}}\right) & \text{(figure 5.4(b))} \quad (5.5\text{c}) \end{cases}$$

where the inverse can be determined iteratively using a simple Newton iteration. Equivalent to tail amplitudes, regression coefficients are given in table 5.2, for three important sample size candidates $N{=}\{10^5, \ldots, 10^9\}$. Regressions are carried out in the interval $\sigma_{RJ}/A_{DJ}{=}[2{\cdot}10^{-3}, 0.5]$, where equation (5.5c) additionally uses $y{\leq}\ln(10^1)$ as upper bound. This is to guarantee for sufficient accuracy of fitted polynomials (r-squared statistic $1{-}r^2{<}10^{-3}$, defined in section 5.4) and does not restrict the analysis, since one is only interested in finding minimum parameter values.

Note, that instead of the empirical equations (5.3) and (5.5) it is always possible to rely on fast numerical approximations as described initially. They allow to quickly investigate the average fitting behavior for arbitrary N and R. In the subsequent sections the numerical approximations will be used together with additional equations to guarantee for correct behavior of fitting methods. A corresponding design example is also given in section 5.5.

DJ	**N**	**Eq.** (5.5b)				**Eq.** (5.5c)					
		$\mathbf{a_0 \cdot 10^3}$	$\mathbf{a_1 \cdot 10^3}$	$\mathbf{a_2 \cdot 10^3}$	$\mathbf{a_3 \cdot 10^3}$	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4 \cdot 10^3}$	$\mathbf{a_5 \cdot 10^3}$
Sin.	10^5	56.9	−42.1	0	0	2.99	3.08	1.51	0.385	49.0	2.46
	10^6	50.2	−32.5	0	0	2.62	2.71	1.35	0.350	45.3	2.30
	10^7	44.9	−26.2	0	0	2.36	2.48	1.26	0.332	43.8	2.26
	10^8	40.5	−21.9	0	0	2.16	2.31	1.20	0.324	43.5	2.28
	10^9	36.8	−18.8	0	0	1.96	2.05	1.04	0.278	37.2	1.94
Uni.	10^5	51.3	−56.1	5.77	0	4.23	4.06	1.86	0.444	53.8	2.58
	10^6	42.7	−49.9	3.08	0	3.72	3.60	1.67	0.407	50.2	2.46
	10^7	38.3	−43.2	1.91	0	3.37	3.29	1.55	0.383	48.0	2.38
	10^8	34.8	−38.0	1.24	0	3.02	2.88	1.33	0.324	40.3	1.99
	10^9	32.7	−33.2	0.94	0	2.83	2.74	1.29	0.322	40.8	2.04
Tri.	10^5	−86.5	−231.0	−44.45	−8.61	7.81	7.00	2.90	0.618	67.1	2.84
	10^6	−33.2	−144.6	−16.13	−3.79	6.85	6.20	2.60	0.566	62.9	2.78
	10^7	−9.3	−103.3	−5.17	−1.78	6.22	5.70	2.43	0.539	61.3	2.78
	10^8	−0.1	−84.8	−2.42	−1.07	5.55	5.00	2.10	0.461	52.0	2.35
	10^9	6.1	−71.3	−0.78	−0.64	5.21	4.77	2.04	0.456	52.3	2.40
Qua.	10^5	−61.4	−157.3	−13.00	−9.91	13.24	12.90	5.77	1.342	161.2	7.69
	10^6	−120.7	−245.7	−51.85	−11.36	11.14	10.65	4.66	1.052	121.3	5.49
	10^7	−92.5	−201.1	−36.50	−7.63	9.87	9.41	4.10	0.922	105.3	4.73
	10^8	−57.9	−148.3	−18.77	−4.46	9.06	8.73	3.86	0.880	102.0	4.67
	10^9	−35.5	−113.7	−8.33	−2.57	8.08	7.62	3.30	0.743	85.1	3.88

Table 5.2.: Coefficients for equations (5.5b) and (5.5c), valid for QN and sQN methods.

5.2. Minimum Sample Size

As a fundamental limitation, a real jitter measurement system can only collect a limited number of jitter samples N within a certain time interval. Even if jitter values are gathered at every bit transition, a high-speed interface cannot afford histogram measurements down to the target BER of 10^{-12} in a feasible amount of time. Thus, the sample size N is strictly limited and mostly predefined by the maximum test time of a measurement system.

High-speed sampling scopes and time interval analyzers (TIAs) as mentioned in [43, 91] afford jitter analysis in real-time, that is, an IO-jitter value is obtained at every bit transition of the received data stream. For a measurement device this corresponds to the theoretical maximum. BIJMs support this real-time TIA feature only in special cases, because the circuit in figure 5.1 must be realized as a parallel structure with R elements to support each delay value [16]. Thus, silicon demand might be too large. Nevertheless, if acquisition time is uncritical, BIJMs can also be realized with very simple, area saving circuits such as time-to-digital converters [17, 57] or Vernier ring oscillators [18].

With the sample size N as a measure for the effort of collecting jitter distributions, one is basically interested in characterizing the minimum tail amplitude $A_{t,min}$, which can be fitted correctly by the algorithm. This allows to specify a worst case distribution when using equation (5.3) from the previous section, and hence, to describe minimum requirements for N in terms of the test distribution shape.

Since collected probability values are always integer multiples of the granularity $1/N$, two fundamental problems have to be dealt. First, fitted tail data suffers from random variations and may cause outliers if only a small, outermost tail region is used for fitting. Second, the fitting methods should ideally include all of the visible RJ tail, but none of the DJ component.

The first problem has already been addressed in section 4.3, where the design parameter ΔP_t was introduced. It defines a region where tail samples at lowest probability levels are always included for regression analysis, and must at least range over two decades. Results also showed, that ΔP_t should be selected as large as possible in order to include the complete visible RJ tail. This second problem is also depicted in figure 5.5 for a right bathtub curve with minimum tail amplitude $A_{t,min}$. Since only part of the Gaussian RJ is observed at the TJ distribution ending, $A_{t,min}$ must significantly exceed the upper probability level P_{up} imposed by ΔP_t. Therefore, as a rule of thumb equation (4.20) provides the following constraints:

$$\Delta P_t \geq 10^2, \text{as large as possible} \tag{5.6a}$$

$$A_{t,min} \gg P_{up} = \Delta P_t / N \tag{5.6b}$$

where P_{up} is the product of multiplication factor ΔP_t and probability granularity $1/N$. The latter condition is rather vague and does not indicate the minimum amplitude as precise as desired. Since ΔP_t is a free design parameter, we like to include all of the observable RJ tail for the worst case situation in figure 5.5. However, the shape of a measured distribution is basically unknown and thus, there is no information available on how deep P_{up} must be located below $A_{t,min}$.

In a simple conservative approach a pessimistic threshold can be selected to define the observable tail part of a Gaussian function. One can for example assume that the Gaussian function is at least visible up to the lower inflection point, which is located at one standard deviation from the mean value. In order to determine the tail part, thus the tail probability p at the inflection point must be calculated. Therefore we recall the Gaussian probability function or CDF from equation (3.9)

$$\text{CDF}(q) = Q^{-1}(q) = p = \frac{1}{2} \cdot \text{erfc}\left(\frac{-q}{\sqrt{2}}\right) \tag{5.7}$$

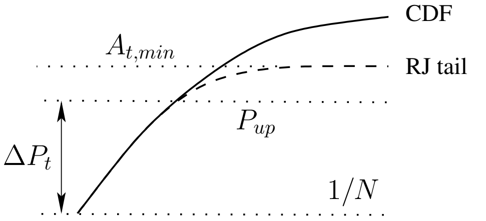


Figure 5.5.: Right bathtub curve (solid) with fitted Gaussian tail (dashed).

which is also the inverse of the Q-function (3.10). The inflection point of a Gaussian is located at $x{=}\mu - \sigma$. In Q-domain, the Gaussian is a linear function with the standardized variable q:

$$q = \frac{x-\mu}{\sigma} \xrightarrow{x=\mu-\sigma} q = -1 \tag{5.8}$$

The corresponding tail probability is thus:

$$Q^{-1}(q{=}-1) = p_{\sigma,1} = 0.159 \tag{5.9}$$

It defines the probability level where the observable Gaussian part of a distribution tail starts. If referred to the minimum amplitude $A_{t,min}$, one can specify a conservative threshold for P_{up}:

$$\begin{aligned} A_{t,min} \cdot p_{\sigma,1} &\geq P_{up} = \Delta P_t / N \\ A_{t,min} &\geq \frac{\Delta P_t}{p_{\sigma,1} \cdot N} \\ A_{t,min} &\geq 6.3 \cdot \Delta P_t / N \end{aligned} \tag{5.10}$$

This final result can be used together with equation (5.3), which allows to identify A_t for a given distribution shape. Thus, the distribution shape can now also be related with a minimum sample size requirement.

Note that the obtained condition is only valid if fitted tails truly follow a Gaussian below the inflection point. According to the central limit theorem, the combination of Gaussian RJ with a bounded DJ component (section 3.2.2) always leads to a TJ distribution which is more Gaussian-like. Therefore, the synthesis principle already indicates validity, and in fact, empirical analyses with the scaled Q-normalization method confirm this assumption. As long as the algorithm is able to correctly identify the tail part, the fitted region also includes the inflection point.

However, this cannot be generalized for arbitrary distribution shapes and thus, the conservative threshold may have to be re-specified with more pessimistic tail assumptions. Unfortunately, no matter how p_σ is selected, a certain risk to choose it inappropriately must always be faced. Although rather theoretical, this is still an inherent disadvantage of the $\hat{c}_{1.2}$ fitting algorithm from section 4.3.

In section 4.4 the constant threshold scenario $Q_{th,c}$ was introduced as an alternative to define the Gaussian tail part in Q-domain (figure 4.27). The worst case analysis situation is here given by the shortest Q-tail, where the distance between lower tail end and the threshold Q_{min} becomes minimum. In this case, the fitted tail suffers from the highest amount of statistical random variation.

In mathematical terms one can say that Q_{min} must be located above the required minimum interval for tail fitting, which is given by $\Delta P_t/N$ when transformed into scaled Q-domain. The worst case situation is given with a maximum scaling factor k_{max}, and thus

$$Q\left(k_{max} \cdot \frac{\Delta P_t}{N}\right) \leq Q_{min} \tag{5.11}$$

where the left hand side describes the $\Delta P_t/N$ interval in scaled Q-domain. Thus, by applying Q^{-1} on both sides and by inserting the minimum amplitude $A_{t,min}{=}1/k_{max}$ we have

$$\frac{\Delta P_t}{N \cdot A_{t,min}} \leq Q^{-1}(Q_{min}) \tag{5.12}$$

which can be rewritten as

$$A_{t,min} \geq \frac{\Delta P_t}{Q^{-1}(Q_{min}) \cdot N} \tag{5.13}$$

This final result is very useful as it directly relates the sample size N with the minimum tail amplitude $A_{t,min}$. Since N describes the measurement effort, the minimum amplitude $A_{t,min}$ returns the resulting benefit. The two parameters ΔP_t and Q_{min} influence the relation and are now *both* included with the Q_{min} based optimization scenario. That means, equation (5.13) was originally derived for the optimization scenario with Q-domain threshold Q_{min} from section 4.4. However, now it also shows the missing link for the conservative probability threshold ΔP_t in equation (5.10).

This threshold was needed to derive an amplitude relation for the $\hat{c}_{1.2}$ optimization scenario in section 4.3, using only the fitting parameter ΔP_t. Now with the same structure of the formula we can easily see, that the probability factor p_σ corresponds to

$$p_\sigma \equiv Q^{-1}(Q_{min}) \tag{5.14}$$

This means, the additional Q_{min} parameter transforms the conservative approach of equation (5.10) into the exact relation (5.13). We recognize this as a beneficial property of the $Q_{th,c}$ algorithmic scenario from section 4.4.

In order to optimize equation (5.13) with respect to minimum tail amplitude, we choose ΔP_t as small as possible, using the minimum $\Delta P_t{=}10^2$ (equation (5.6)). The threshold parameter Q_{min} is back-transformed into probability domain, where its reverse value also influences $A_{t,min}$. In figure 5.6 the Q^{-1} behavior is displayed as a function of Q_{min}. According to this plot one would like to minimize Q_{min}, but this comes along with a degraded performance since an optimum value was already identified at $Q_{min}{=}{-}1.2$ (see figure 4.31). Hence, as long as $A_{t,min}$ is not a critical specification requirement, changes of ΔP_t or Q_{min} should preferably be avoided. Thus, only a larger sample size N is truly able to decrease $A_{t,min}$.

As an example, in section 4.4.3 distributions were analyzed using $N{=}10^7$ and $Q_{min}{=}{-1.2}$ as threshold. Now we also like to guarantee a minimum tail interval to avoid outliers caused by statistical tail variations. With the minimum $\Delta P_t{=}10^2$ and equation (5.13):

$$\begin{aligned} &\{N=10^7,\ Q_{min}=-1.2,\ \Delta P_t=10^2\} \\ &\quad\Rightarrow A_{t,min} = 8.7\cdot 10^{-5} \end{aligned} \tag{5.15}$$

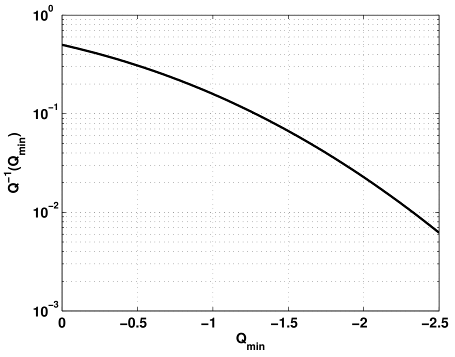


Figure 5.6.: Inverse quantile function Q^{-1} applied to Q_{min}.

The result allows to correctly fit tail amplitudes down to $A_{t,min}{=}8.7{\cdot}10^{-5}$, without causing large outliers. For the worst case with quadratic curve shaped DJ and equation (5.4) we yield a smallest jitter ratio of $\sigma_{RJ}/A_{DJ}{=}6.0{\cdot}10^{-3}$, which covers nearly all of the test distributions in figures 4.26 and 4.33. Thus, none of the investigated distribution shapes is strongly affected by outliers. However, ΔP_t should always be selected as large as possible to guarantee for an optimum outlier suppression. Further, the obtained result is only valid for simulations where $R{=}R_{sim}$ and thus, when time resolution does not affect the performance. This problem domain is addressed subsequently.

5.3. Minimum Time Resolution

Equivalent to the sample size, also the time resolution R of a jitter measurement system causes limitations. Collected jitter distributions are represented by a discrete number of bins or nodes R, which divide the unit interval (UI) into equidistant time intervals. A coarse resolution leads to a small number of probability samples along the bathtub tails. This quantization effect limits the applicability of jitter analysis methods and is thus addressed in this section.

Ideally, one likes to specify the minimum random jitter component $\sigma_{t,min}$ of a test distribution, which can be fitted correctly by the algorithm. A convergence failure is basically noticed as a limiting effect of steep tails, where TJ estimates become highly biased. It is caused when a bathtub function is not supported by a sufficient number of tail samples.

Due to the quantile normalization, the standard deviation σ_t of a fitted Gaussian is given by the reciprocal of the tail slope (equation (4.6)). According to the scheme in figure 5.7, the linearized Gaussian part of a tail in Q-domain must be supported by at least three nodes or bins. These bins guarantee for correct linear regression. If the Q-tail becomes too steep, the upper bin will be located in the bent curve region which belongs to the DJ component, thus leading to wrong extrapolations. The maximum slope s_{max} ca be found by defining the extreme case with Δt_{min} and Δq.

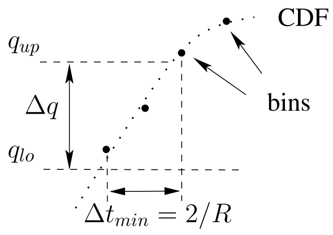


Figure 5.7.: Definition of maximum slope s_{max}.

$$\sigma_{RJ,min} = 1/s_{max} = \Delta t_{min}/\Delta q = \Delta t_{min}/|q_{lo} - q_{up}| \tag{5.16}$$

Along the time axis, at least three bins of the Q-tail are needed to calculate the regression error. Together with the given time resolution $1/R$ this corresponds to a time interval $\Delta t_{min}{=}2/R$. The value for Δq is the Q-tail region where the fitting algorithm forces tail samples to be included for regression analysis. Thus, again the two algorithmic approaches $\hat{c}_{1.2}$ and $Q_{th,c}$ with ΔP_t interval and Q_{min} threshold must be considered separately.

Minimum Time Resolution with Fitting Parameter $\mathbf{\Delta P_t}$

The optimized algorithm from section 4.3 uses the parameter ΔP_t to define a probability region where measured tail bins are included. The lowest probability level p_{lo} is thus given by the granularity $1/N$, and the upper probability level p_{up} by the fitting parameter as factor $\Delta P_t/N$. Additionally these probability values are transformed into scaled Q-domain, where the optimum scaling factor k_t is determined, and thus

$$q_{lo} = Q(k_t \cdot p_{lo}) = Q(k_t/N) \tag{5.17a}$$

$$q_{up} = Q(k_t \cdot p_{up}) = Q(k_t \cdot \Delta P_t/N) \tag{5.17b}$$

In scaled Q-domain we yield

$$\begin{aligned} \Delta q &= |Q(k_t/N) - Q(k_t \cdot \Delta P_t/N)| \\ &= \left| Q\left(\frac{1}{A_t \cdot N}\right) - Q\left(\frac{\Delta P_t}{A_t \cdot N}\right) \right| \end{aligned} \tag{5.18}$$

and can derive a relation for minimum standard deviation:

$$\sigma_{t,min} = \frac{2}{R} \cdot \left(\frac{1}{|Q\left(\frac{1}{A_t \cdot N}\right) - Q\left(\frac{\Delta P_t}{A_t \cdot N}\right)|} \right) \tag{5.19}$$

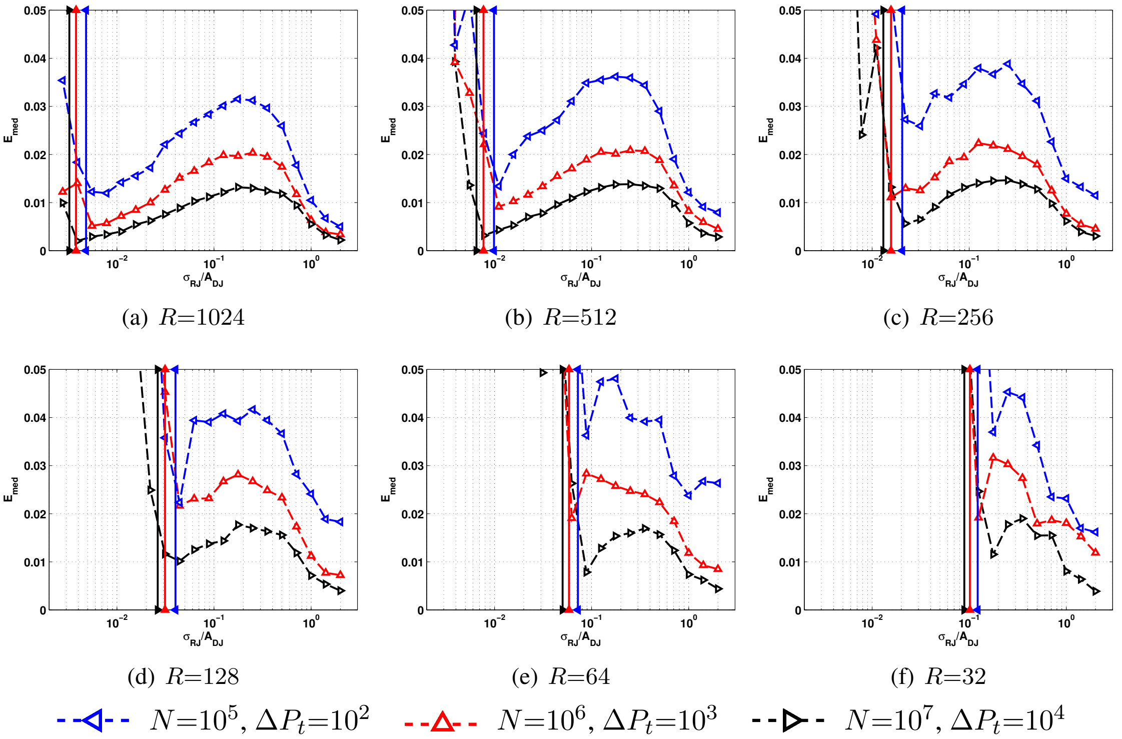
This relation is only valid if the upper boundary $q_{up}{=}Q\left(\frac{\Delta P_t}{A_t \cdot N}\right)$ is located in the linearized Gaussian part of the Q-tail. In the previous section corresponding equations (5.6) and (5.10) were derived to ensure that this condition is met.

Unfortunately, the tail amplitude A_t is difficult to specify as it depends on the shape of a jitter distribution. The algorithm from section 4.3 with $\hat{c}_{1.2}$ fitness criterion maximizes the tail length of an initial search grid, before carrying out a locally bounded minimum search. The tail length is thus essential for initial tail amplitude search. At very coarse time resolutions, the algorithm cannot identify any amplitudes where the regression length is maximum, simply because there are not sufficient bins supporting the bathtub tails anymore. In this case, an amplitude of $A_t{=}1$ is assumed and thus, the algorithm behaves equivalent to the conventional Q-normalization (QN) method. This allows to omit A_t, and as a final result

$$\sigma_{t,min} \approx \frac{2}{R} \cdot \left(\frac{1}{|Q(1/N) - Q(\Delta P_t/N)|} \right) \tag{5.20}$$

This equation correctly tracks the convergence limit of the sQN method, and is empirically verified in figure 5.8. The vertical lines mark calculated values for $\sigma_{t,min}$, according to equation (5.20) at three different sample sizes $N{=}\{10^5,\ 10^6,\ 10^7\}$. For each of the sample sizes, the corresponding curve over varying distribution shape shows where the fitting algorithm truly starts to fail. Over different sample size, $\Delta P_t{=}\{10^2, 10^3, 10^4\}$ is also varied, in order to obtain a constant ratio $\Delta P_t/N{=}10^{-3}$. Thus, the upper probability bound q_{up} is kept constant and the benefit of increasing sample size N can be demonstrated.

Note, that the $\sigma_{t,min}$ values in equation (5.20) refer to the fitted tails of a total distribution. They cannot directly be related with the curves in figure 5.8, since σ_{RJ} and A_{DJ} correspond to parameters prior to TJ composition. Therefore, these parameters were identified using the inverse of equation (5.5b). With known and constant A_{DJ}, a Newton approach finds the $\sigma_{RJ,min}$ value where the correct $\sigma_{t,min}$ is obtained.



(a) R=1024

(b) R=512

(c) R=256

(d) R=128

(e) R=64

(f) R=32

Figure 5.8.: Smallest analyzable RJ component $\sigma_{t,min}$. Empirical relation (5.20) is verified with respect to varying time resolution R=$\{1024, \ldots, 32\}$ and sample size N=$\{10^5, 10^6, 10^7\}$. ΔP_t=$\{10^2, 10^3, 10^4\}$, K=250, uniform DJ type, $\hat{c}_{1.2}$ optimization scenario.

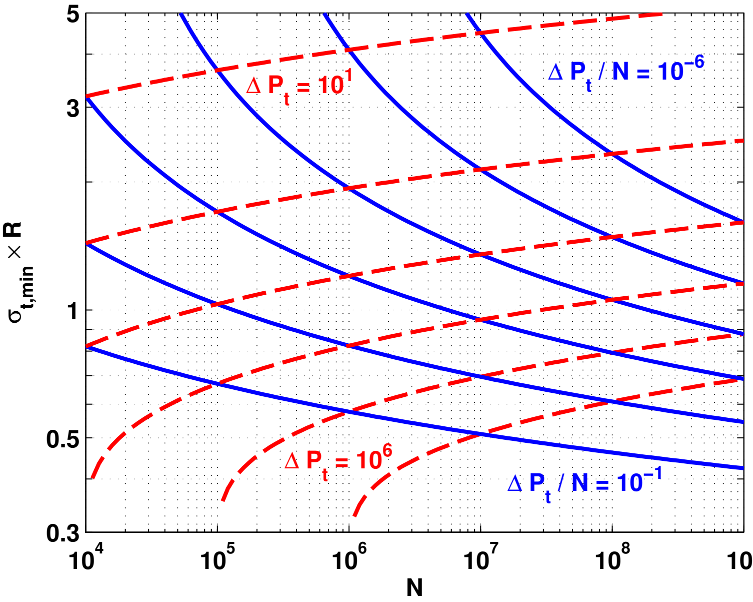


Figure 5.9.: ΔP_t selection chart for identifying $\sigma_{t,min}{\cdot}R$, as described by equation (5.20). ΔP_t=$\{10^1, \ldots, 10^6\}$ (dashed), $\Delta P_t/N$=$\{10^{-6}, \ldots, 10^{-1}\}$ (solid).

In figure 5.9 additionally, a chart for (5.20) is provided to help selecting the free design parameter ΔP_t, and to verify whether a certain minimum RJ tail can be fitted correctly. For example, knowing N and R one can easily verify whether a desired $\sigma_{t,min}$ is guaranteed for a certain ΔP_t. $\sigma_{t,min}$ decreases linearly with the number of bins, and is thus represented by the normalized, dimensionless variable $\sigma_{t,min} \cdot R$. The chart is constructed using two different types of curves, where either ΔP_t or the ratio $\Delta P_t/N$ is constant. This is to guarantee the two requirements with respect to outlier suppression (equation (5.6)) and minimum tail amplitude (equation (5.10)) independently. Both curve types allow for the analysis of a smaller RJ standard deviation, if ΔP_t is increased while N is constant. The maximum ΔP_t is only restricted by the minimum tail amplitude. Note that in equation (5.20) the value $\Delta P_t/N$ forms the upper probability level for tail selection. If constant, the sample size N can be used to increase the fitting region and thus, to identify a smaller RJ standard deviation.

An interesting effect is further noticed with constant ΔP_t. If *more* jitter samples are used for collecting distributions, $\sigma_{t,min}$ becomes *larger*. This seems contradictory, but is a result of the nonlinear Q-function behavior when only the lowest probability region is used for tail fitting. In fact, the benefit in this case lies in a smaller minimum amplitude $A_{t,min}$. To summarize these observations, both ΔP_t and N should be chosen as large as possible, without violating equation (5.10).

Minimum Time Resolution with $\mathrm{Q_{min}}$ Threshold

Equivalent to the ΔP_t based algorithm, $\sigma_{t,min}$ can also be determined for the Q_{min} based optimization scenario from section 4.4. Here, the probability region where tail samples are included for regression analysis is given by the Q_{min} threshold as upper bound, and the probability granularity $1/N$ as lower bound. According to figure 5.7, in scaled Q-domain we yield

$$q_{lo} = Q(k_t/N) \tag{5.21a}$$

$$q_{up} = Q_{min} \tag{5.21b}$$

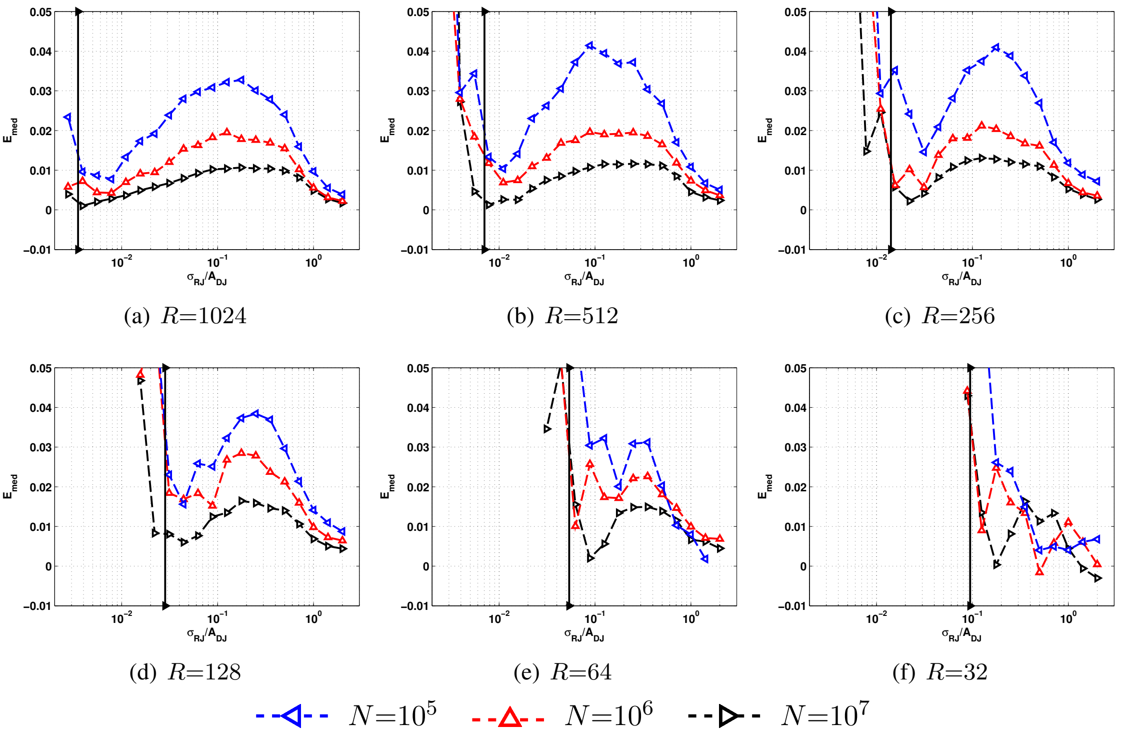
This leads to the equation

$$\sigma_{t,min} = \frac{2}{R} \cdot \left(\frac{1}{\left| Q\left(\frac{1}{A_t \cdot N}\right) - Q_{min} \right|} \right) \tag{5.22}$$

The upper bound Q_{min} must belong to the linearized Q-tail part, which can easily be verified by equation (5.13). Again, it is not possible to specify a tail amplitude A_t because it depends on the distribution shape, but the minimum amplitude $A_{t,min}$ from equation (5.13) can be used instead. This is because correct convergence for the fitting method is assumed, and $A_{t,min}$ forms a worst case scenario which allows for pessimistic $\sigma_{t,min}$ estimation. Thus, when inserting equation (5.13) into (5.22) :

$$\sigma_{t,min} \leq \frac{2}{R} \cdot \left(\frac{1}{\left| Q\left(\frac{Q^{-1}(Q_{min})}{\Delta P_t}\right) - Q_{min} \right|} \right) \tag{5.23}$$

where the $\sigma_{t,min}$ function is now reduced to a simplified form without depending on the sample size N. This expression cannot be reduced further, due to the non-linearity of the Q-function. Equivalent to the previous analysis, applicability of this equation is demonstrated in figure 5.10. Vertical solid lines mark the calculated values from equation (5.23). The course of median error E_{med} over varying jitter ratio shows where the fitting algorithm starts to fail at each of the three sample sizes $N{=}\{10^5, 10^6, 10^7\}$. The Q-domain threshold $Q_{min}{=}{-1.0}$ is here used for



(a) R=1024

(b) R=512

(c) R=256

(d) R=128

(e) R=64

(f) R=32

Figure 5.10.: Smallest analyzable RJ component $\sigma_{t,min}$. Empirical relation (5.22) is verified with respect to varying time resolution R=$\{1024, \ldots, 32\}$ and sample size N=$\{10^5, 10^6, 10^7\}$. Q_{min}=-1.0, ΔP_t=10^2, K=250, uniform DJ type, $Q_{th,c}$ optimization scenario.

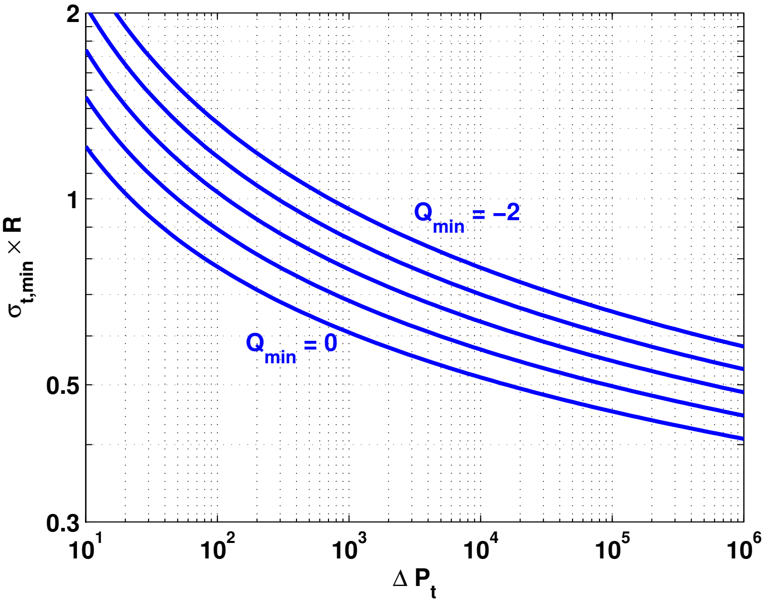


Figure 5.11.: ΔP_t and Q_{min} selection chart for identifying the normalized variable $\sigma_{RJ,min}{\cdot}R$, as described by equation (5.23). Q_{min}=$\{0, -0.5, -1.0, -1.5, -2.0\}$.

calculations, which is different from the optimum value in section 4.4.2. This is due to the coarse time resolutions $R{\ll}R_{sim}$, and allows to include more tail samples with the $Q_{th,c}$ optimization scenario.

Again, $\sigma_{t,min}$ values in equation (5.23) refer to the fitted tail parameter of a TJ distribution, and can only be plotted in figure 5.10 when using the inverse of equation (5.5b). This yields different results for the three sample sizes, which are located very close to each other. Here only the $N{=}10^7$ case is shown, due to more pessimistic values.

If ΔP_t is increased, $\sigma_{t,min}$ estimates become smaller, which allows to search for a suited parameter value without depending on the sample size N. The normalized standard deviation $\sigma_{t,min}{\cdot}R$ is plotted in a chart for ΔP_t selection over different Q_{min} values (figure 5.11). The chart suggests ΔP_t to be selected as large as possible, but note that equation (5.13) must always be fulfilled. Also, $\Delta P_t{\geq}10^2$ is highly recommended to guarantee for sufficient outlier suppression.

5.4. Estimation Error Analysis

The estimation performance of the scaled Q-normalization method is highly affected by the sample size N, time quantization R and differential non-linearity (DNL) error of a jitter measurement system. In this section, first, empirical relations are derived that quantify estimation error in terms of analysis parameters (N, R) and the distribution shape (σ_{RJ}, A_{DJ}). Then an error ripple effect is investigated, which appears with a coarse time discretization of distribution tails. Finally, a DNL error model is provided to include the effect of process variations into the empirical relations. The presented sections are meant to assist the designer in finding an optimum trade-off between fitting accuracy and hardware expense.

5.4.1. Empirical Error Analysis

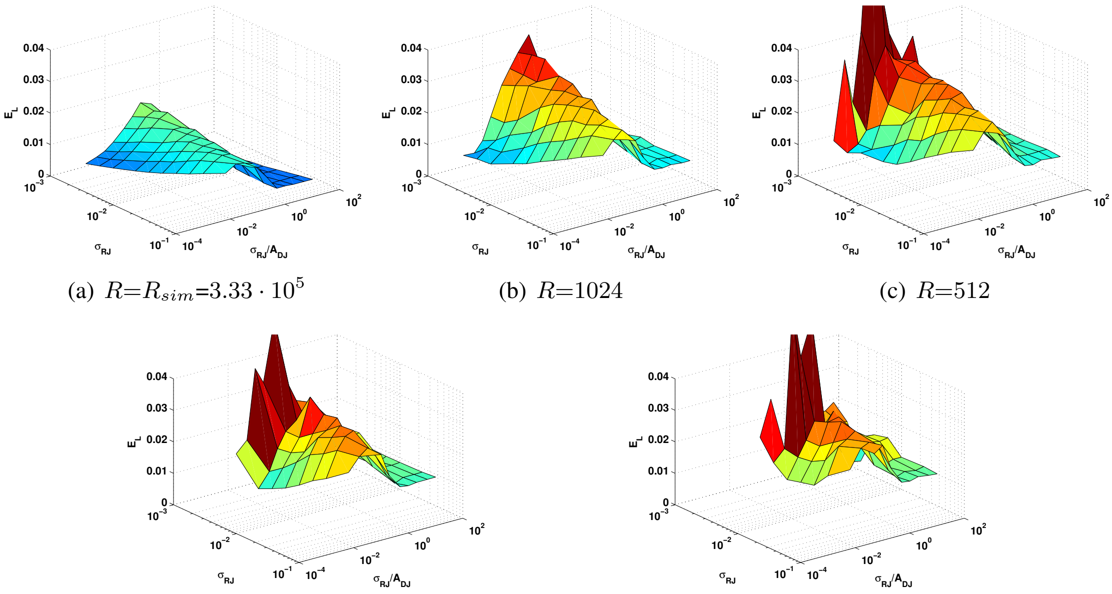
The estimation error E as defined in equations (3.17) and (3.18) is a statistical variable, which can basically be expressed as a function of four variables:

$$\{E_{med}, IQR, E_L\} = f(\sigma_{RJ}, A_{DJ}, N, R) \tag{5.24}$$

In order to derive empirical relations, first the complexity of this four-dimensional function must be reduced. As an introductory example, in figure 5.12 the estimation loss E_L over varying distribution shape is depicted, by combining RJ (σ_{RJ}) with uniform type DJ (A_{DJ}). TJ_{pp} estimates are obtained from fitted bathtub tails using different time resolution R. In the example, the sQN algorithm uses the optimized parameter configuration from section 4.3.2, with $\hat{c}_{1.2}$ and $\Delta P_t{=}10^5$.

Basically two effects are noticed. First, the algorithm is highly biased if random jitter falls below a certain minimum $\sigma_{RJ,min}$. This limiting effect is caused by the discrete time resolution and has already been dealt in the previous section. Second, E_L increases when either σ_{RJ} or R are reduced at constant jitter ratio σ_{RJ}/A_{DJ}. That means, the surfaces in figure 5.12 are symmetric and thus, the ratio σ_{RJ}/A_{DJ} can be reused as simplifying shape variable. However, an additional dependency on σ_{RJ} and R is observed, combined with a high error ripple if either one of the two variables is varied.

In order to quantify the estimation error, equation (5.24) must be simplified. Considering again the jitter ratio σ_{RJ}/A_{DJ} as single variable, only the shapes of largest error from the analysis in figure 4.26 may be investigated within the scope of a worst case analysis. This yields a single shape value per DJ type. Additionally, with constant ratio both σ_{RJ} and A_{DJ} vary simultaneously, so that each parameter also represents the total distribution size. The estimation error thus reduces to



(a) R=R_{sim}=$3.33 \cdot 10^5$

(b) R=1024

(c) R=512

(d) R=256

(e) R=128

Figure 5.12.: Estimation loss E_L for different values of σ_{RJ}, A_{DJ}, and R. R_{sim} in figure 5.12(a) is obtained with a simulator time resolution of 1 fs in a 3Gb/s interface. N=10^8, ΔP_t=10^5, K=250, $\hat{c}_{1.2}$ optimization criterion.

DJ type	**Worst case** $\boldsymbol{\sigma_{RJ}/A_{DJ}}$	**Amplitude** $\mathbf{A_{t,min}}$ (eq. (5.3), tab. 5.1)
None, only RJ	—	1.000
Sinusoidal	1/2	0.477
Uniform	1/4	0.319
Triangular	1/8	0.107
Quadratic	1/16	0.021

Table 5.3.: Selected worst case shape values $\sigma_{RJ,min}/A_{DJ}$ and corresponding worst case tail amplitudes $A_{t,min}$ (N=10^8) for different DJ types.

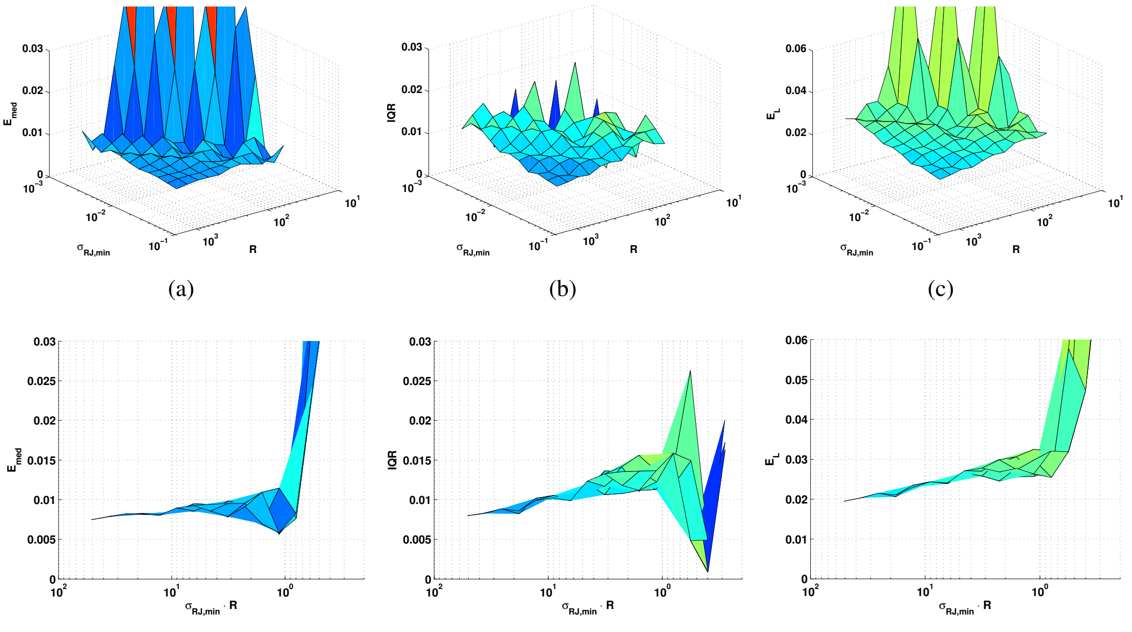
a simplified function of three variables

$$\begin{aligned} \max\{E_{med}, IQR, E_L\} &= f(\sigma_{RJ}, N, R) \\ \text{if} \quad \sigma_{RJ}/A_{DJ} &= \text{const.} \end{aligned} \tag{5.25}$$

The variable σ_{RJ} now describes the overall distribution size, while A_{DJ} is discarded. Remember, that A_{DJ} does not fully disappear as parameter, it has only been transformed into a dependent variable. Varying the parameter σ_{RJ} now also means to vary A_{DJ} according to the selected worst case ratio.

This jitter ratio depends on the selected DJ type and thus, has to be determined for each of the investigated shapes. From the performance results in sections 4.3.3 and 4.4.3 we can easily determine these ratios for subsequent analysis. They are given in table 5.3 together with worst case tail amplitudes, obtained by equation (5.3) at N=10^8. The shape values have been selected as power-of-two to simplify the performance analysis, so that the numerical approximations of true total jitter values $\mathrm{TJ}_{pp,true}$ can be reused.

The next step is to search for a linear or logarithmic dependency between two of the independent



(a) (b) (c)

(d) (e) (f)

Figure 5.13.: Median estimation error E_{med} (a,d), interquartile range IQR (b,e) and estimation loss E_L (c,f) over varying R and $\sigma_{RJ,min}$. N=10^8, ΔP_t=10^5, K=250, $\sigma_{RJ,min}/A_{DJ}$=const.=$1/4$, uniform DJ type.

variables from equation (5.25), while keeping the third one constant. As depicted in figure 5.13, such a relation is found for a constant sample size N. If the parameters R or σ_{RJ} are varied, all three performance indicators E_{med}, IQR and E_L yield plane surfaces that consist of a convergence region, which is to some degree affected by ripple. Inside the convergence region, bathtub tails are supported by sufficient bins, thus allowing the fitting algorithm to correctly extrapolate tails. When either σ_{RJ} or R become too small, the algorithm shows a large error bias. The ripple increases when moving toward the convergence limit, which is an effect caused by the coarse time discretization and will also be investigated later on in this section.

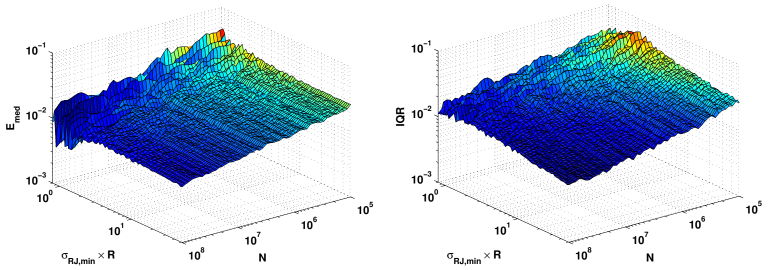
The regression analysis can be simplified by changing the variable representation into the product form $\sigma_{RJ}{\cdot}R$, as demonstrated in the bottom row of figure 5.13. This is due to the constant ratio σ_{RJ}/A_{DJ}, which changes the whole distribution size when only σ_{RJ} is varied. The fitting algorithm cannot distinguish between a variation of the distribution size or the time resolution. In fact, the extrapolation error is only influenced by the number of bins which form the bathtub tail. An increase in performance of the algorithm can therefore be achieved with a larger number of bins R as well as a larger distribution size given in terms of σ_{RJ}.

As a result, another dependency between the variables R and σ_{RJ} has been identified. One of them can be discarded if only the product $\sigma_{RJ,min}{\cdot}R$ is considered, which further reduces the estimated error in equation (5.25) to a function of two variables:

$$\max\{E_{med}, IQR, E_L\} = f(\sigma_{RJ}{\cdot}R, N)\ , \quad \sigma_{RJ,min}/A_{DJ} = \text{const.} \tag{5.26}$$

Consequently, for a constant sample size N, estimation performance in figures 5.13 (d,e,f) can also be approximated with a simple regression line.

The variable product $\sigma_{RJ}{\cdot}R$ is here referred to as node or bin density of a distribution. When comparing this density with parameter selection charts from figures 5.9 and 5.11, we notice that it also corresponds to the normalized standard deviation. Depending on the selected algorithmic type, equations (5.20) and (5.23) can thus be used to identify the expected convergence limit.



(a) (b)

FIGURE 5.14.: Surfaces for empirical analysis of E_{med} (a) and IQR (b). The worst case shape σ_{RJ}/A_{DJ}=1/4 (uniform type DJ) is here investigated in the range N=$\{10^5, \ldots, 10^8\}$ and $\sigma_{RJ}{\cdot}R$=$\{0.8, \ldots, 51.2\}$, K=250.

In figure 5.13 the $\hat{c}_{1.2}$ based algorithm was used with N=10^8 and ΔP_t=10^5 as parameters. The selection chart in figure 5.9 and equation (5.20) yield $\sigma_t{\cdot}R$=0.79 as limit. With σ_{RJ}/A_{DJ}=1/4 and equation (5.5b) we get $\sigma_{RJ}{\cdot}R$=0.72, which is consistent with the observed surfaces in figure 5.13.

Using the sample size N as second independent variable, the estimation error can be represented by two-dimensional surfaces. If regression planes achieve acceptable accuracy, an empirical description of the extrapolation error thus becomes possible. Subsequently, the two algorithmic candidates from sections 4.3 and 4.4 are investigated separately.

Empirical Error with Parameter $\Delta \mathrm{P_t}$

The sQN method with optimized $\hat{c}_{1.2}$ criterion (section 4.3) is investigated with respect to varying sample size N and bin density $\sigma_{RJ}{\cdot}R$. Therefore, the design parameter ΔP_t must be selected appropriately. In figure 5.9 a selection chart was given for either ΔP_t or the probability $\Delta P_t/N$, depending on whether the minimum tail amplitude $A_{t,min}$ is known. Here, the distribution shapes with worst case characteristics from table 5.3 are analyzed. With a constant $\Delta P_t/N$=10^{-3}≪$A_{t,min}$ all the given tail amplitudes are included and condition (5.10) is fulfilled. Further, the analysis benefits from a larger tail interval if N is increased.

In figure 5.14, as an example, the two-dimensional surfaces of median error E_{med} and interquartile range IQR are plotted. Both surfaces can be approximated very well using regression planes where the resulting regression coefficients define the empirical relations. At smallest bin densities $\sigma_{RJ}{\cdot}R$≤2 the median error starts to oscillate, because of a scarce discretization or limited number of bins on the bathtub tails. This effect will be investigated later on in section 5.4.2.

When deriving empirical coefficients for the regression planes, a logarithmic scaling of all three axes must be considered. Thus, regressions are described with

$$\begin{aligned} \ln(z) &= c_1 + c_2 \cdot \ln(x) + c_3 \cdot \ln(y) \\ &= \ln(e^{c_1} \cdot x^{c_2} \cdot y^{c_3}) \\ \Rightarrow z &= e^{c_1} \cdot x^{c_2} \cdot y^{c_3} \end{aligned} \tag{5.27}$$

When mapped onto the original variables we thus have

$$x = \sigma_{RJ}{\cdot}R, \quad y = N, \quad z = \{E_{med}, IQR, E_L\} \tag{5.28}$$

By choosing the regression coefficients a_0, a_1 and a_2

$$a_0 = e^{c_1}, \quad a_1 = -c_2, \quad a_2 = -c_3 \tag{5.29}$$

$\mathbf{DJ}$, $\sigma_{\mathbf{RJ}}/\mathbf{A_{DJ}}$		$\mathbf{N} = [\mathbf{5 \cdot 10^5}, \mathbf{10^8}]$ $\sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$				$\mathbf{N} = [\mathbf{10^4}, \mathbf{10^6}]$ $\sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$			
		$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{r^2}$	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{r^2}$
Sinusoidal $1/2$	E_{med}	0.248	0.121	0.161	0.977	0.463	0.107	0.211	0.961
	IQR	0.471	0.198	0.188	0.965	0.668	0.168	0.230	0.967
	E_L	0.919	0.168	0.177	0.982	1.444	0.144	0.223	0.976
Uniform $1/4$	E_{med}	0.256	0.103	0.157	0.977	0.494	0.099	0.206	0.968
	IQR	0.313	0.190	0.163	0.960	1.019	0.161	0.260	0.973
	E_L	0.720	0.154	0.161	0.981	1.898	0.136	0.238	0.981
Triangular $1/8$	E_{med}	0.447	0.070	0.170	0.984	1.089	0.035	0.239	0.994
	IQR	0.365	0.163	0.168	0.950	0.727	0.124	0.235	0.949
	E_L	0.994	0.117	0.169	0.984	2.179	0.076	0.237	0.988
Quadratic $1/16$	E_{med}	0.983	0.044	0.201	0.992	1.724	-0.003	0.243	0.992
	IQR	0.681	0.170	0.201	0.960	1.105	0.101	0.263	0.955
	E_L	1.987	0.098	0.201	0.986	3.266	0.036	0.250	0.990
Only RJ	E_{med}	0.091	0.408	0.137	0.925	0.552	0.457	0.272	0.936
	IQR	1.471	0.342	0.252	0.961	4.129	0.313	0.365	0.971
	E_L	2.002	0.355	0.231	0.969	6.583	0.335	0.350	0.973

Table 5.4.: E_{med}, IQR and E_L regression coefficients for equation (5.30). $\Delta P_t/N{=}10^{-3}$ (planes for large N) and $\Delta P_t/N{=}10^{-2}$ (planes for small N), $\hat{c}_{1.2}$ algorithm, parameter intervals specified above, $K{=}250$.

and re-substituting these variables one gets

$$\{E_{med}, IQR, E_L\} = a_0 \cdot (\sigma_{RJ}{\cdot}R)^{-a_1} \cdot N^{-a_2} \tag{5.30}$$

The three regression coefficients have been determined for each of the investigated DJ types as well as the pure Gaussian RJ case, and are listed in table 5.4 for two different sample size intervals. The surfaces use a constant time resolution $R{=}2048$, while varying only the RJ standard deviation in the range $\sigma_{RJ}{=}[1/1024, 1/40]$ UI. This way the bin density $\sigma_{RJ}{\cdot}R$ supports arbitrary floating point values, and is not restricted to integer values of R. Note that the given parameter range for $\sigma_{RJ}{\cdot}R$ is smaller than the one depicted in figure 5.14, in order to exclude error oscillations. In fact, the intervals have been selected from original equidistant grids of $75{\times}51$ nodes with $\sigma_{RJ}{\cdot}R{=}\{0.8, \ldots, 51.2\}$ UI as well as $N{=}\{10^5, \ldots, 10^8\}$ and $N{=}\{10^4, \ldots, 10^6\}$ respectively.

The r-squared statistic [20] in the last columns describes the quality of regression planes with $r^2{\in}[0, 1]$. The obtained results always highlight a very high degree of correlation with planes for both large and small N. The latter ones yield large IQR values, due to a strong influence of random tail variations. This influence was reduced to some degree by choosing a larger $\Delta P_t/N{=}10^{-2}$.

A direct error estimation with E_L offers the advantage of slightly improved precision, but is less flexible. That means, E_{med} and IQR are not restricted to the definition from equation (3.18), and can always be adapted as linear combinations to describe arbitrary statistical confidence levels.

Empirical Error with $\mathrm{Q_{min}}$ Threshold

Equivalent to the previous analysis, also the error of the $Q_{th,c}$ algorithm based on the threshold Q_{min} from section 4.4 can be investigated. As a major difference to simulations with $R{=}R_{sim}$, the threshold is now reduced to $Q_{min}{=}{-}1.0$ (instead of $Q_{min}{=}{-}1.2$). This adaptation allows the $Q_{th,c}$ algorithm to include more tail samples, especially at coarse time resolutions.

Two-dimensional error surfaces are again plotted as functions of bin density $\sigma_{RJ}{\cdot}R$ and sample size N. Equivalent to the previous analysis, regression planes describe the empirical relation

$\mathbf{DJ}, \sigma_{\mathbf{RJ}}/\mathbf{A_{DJ}}$		$\mathbf{N} = [\mathbf{5} \cdot \mathbf{10^5}, \mathbf{10^8}]$ $\sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$				$\mathbf{N} = [\mathbf{10^4}, \mathbf{10^6}]$ $\sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{5.0}, \mathbf{51.2}]$			
		$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{r^2}$	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{r^2}$
Sinusoidal $1/2$	E_{med}	0.179	0.101	0.150	0.797	0.423	0.075	0.217	0.932
	IQR	0.912	0.227	0.208	0.934	0.671	0.176	0.202	0.935
	E_L	1.453	0.196	0.193	0.967	1.454	0.145	0.208	0.968
Uniform $1/4$	E_{med}	0.185	0.098	0.146	0.826	0.773	0.086	0.250	0.952
	IQR	0.927	0.250	0.202	0.960	0.889	0.161	0.215	0.950
	E_L	1.440	0.207	0.186	0.978	2.121	0.137	0.227	0.969
Triangular $1/8$	E_{med}	0.259	0.069	0.150	0.929	1.818	0.040	0.299	0.982
	IQR	0.985	0.191	0.208	0.962	0.843	0.137	0.208	0.947
	E_L	1.533	0.148	0.188	0.981	2.883	0.098	0.244	0.986
Quadratic $1/16$	E_{med}	0.570	0.039	0.186	0.979	2.751	-0.006	0.308	0.989
	IQR	1.478	0.160	0.231	0.975	0.948	0.124	0.211	0.962
	E_L	2.505	0.111	0.212	0.985	3.911	0.069	0.257	0.988
Only RJ	E_{med}	0.045	0.432	0.095	0.856	0.174	0.418	0.188	0.679
	IQR	0.482	0.235	0.190	0.958	1.032	0.259	0.248	0.942
	E_L	0.728	0.264	0.175	0.968	1.751	0.280	0.242	0.957

Table 5.5.: E_{med}, IQR and E_L coefficients for equation (5.30). $Q_{min}{=}{-1.0}$, $\Delta P_t{=}10^2$, $Q_{th,c}$ algorithm, parameter intervals specified above, $K{=}250$.

of these two variables. At very low densities, $\Delta P_t{=}10^2$ guarantees for sufficient bins to be included with regression analysis. This avoids outliers, but unfortunately the pessimistic estimation property with positive error bias for E_{med} is not guaranteed anymore. In fact, E_{med} now also returns negative values, as can also be seen in figure 5.10. This is an effect caused by statistical tail variations, which leads to highly overestimated scaling factors and optimistic TJ values.

The same original grids are used for the regression planes as with the previous analysis. Table 5.5 contains the regression coefficients a_0, a_1 and a_2 for each of the investigated DJ types, as well as the r-squared statistic to indicate the quality of fitted planes. The empirical relation is again given by equation (5.30). This time, E_{med} planes highlight significantly smaller r^2 values, because of the partial influence of negative errors at lowest bin densities. This problem is overcome by selecting a smaller parameter range of $\sigma_{RJ}{\cdot}R$ for planes with small N. However, besides the pure Gaussian case, E_L is significantly larger compared to the $\hat{c}_{1.2}$ based algorithm.

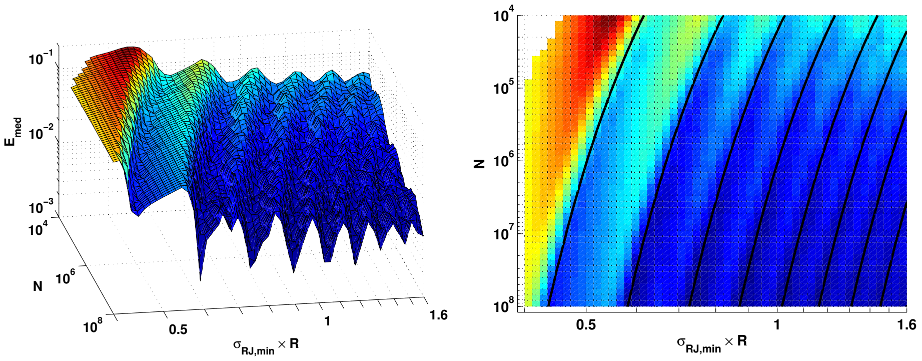
To highlight this difference in a brief example, we assume a jitter measurement system with the parameters $R{=}128$ and $N{=}10^8$. For the sQN method with $\hat{c}_{1.2}$ based tail fitting $\Delta P_t{=}10^5$ is selected, and a minimum analyzable standard deviation $\sigma_{t,min}{=}6.2{\cdot}10^{-3}$ obtained (equation (5.20)). According to equation (5.5b), at a worst case jitter ratio of $\sigma_{RJ}/A_{DJ}{=}1/4$ (uniform DJ) we get $\sigma_{RJ,min}{=}5.7{\cdot}10^{-3}$. The worst case error is determined using equation (5.30) and yields

$$E_{med} = 1.46\%, IQR = 1.64\%, E_L = 3.90\% \tag{5.31}$$

For the $Q_{th,c}$ based tail fitting with $Q_{min}{=}{-1.0}$ instead

$$E_{med} = 1.31\%, IQR = 2.42\%, E_L = 4.98\% \tag{5.32}$$

This result shows that the $Q_{th,c}$ algorithm, although less biased, suffers from a significantly larger statistical spread which is especially noticed with E_L. A similar error characteristic is observed for all four DJ types. Also for the pure RJ case, the error of the $Q_{th,c}$ algorithm becomes excessively large as soon as $\sigma_{RJ}{\cdot}R{\leq}5$ and $N{\leq}10^6$. Generally, this makes the $\hat{c}_{1.2}$ algorithm a better suited choice for tail fitting with hardware based jitter measurements. Therefore, only the $\hat{c}_{1.2}$ algorithm will be further utilized subsequently.



(a) (b)

Figure 5.15.: Error ripple effect: simulated E_{med} surface (a) and expected "error valleys" according to equation (5.35) (b) with a test distribution $\sigma_{RJ,min}/A_{DJ}$=1/4 (uniform DJ). The investigated parameter ranges are N=$\{10^4,\ldots,10^8\}$ and $\sigma_{RJ,min}{\cdot}R$=$\{0.4,\ldots,1.6\}$. K=250.

5.4.2. Error Ripple Effect

As was already shown in figure 5.14, error oscillations can be observed when the bin density $\sigma_{RJ,min}{\cdot}R{\leq}2$ and thus, reaches toward the convergence limit. This ripple effect is caused by the coarse time discretization of bathtub tails. If tails are described by only few bins, fitting results highly depend on their locations along the bathtub curve and are thus highly scattered. In such cases a suitable location of bins can significantly improve fitting performance.

In figure 5.15(a) the resulting ripple of a test distribution with $\sigma_{RJ,min}/A_{DJ}$=1/4 and uniform DJ type is shown. At smallest bin densities this ripple is still visible even though E_{med} becomes very large. For tail fitting the $\hat{c}_{1.2}$ based algorithm is used with a minimum tail interval of $\Delta P_t/N$=10^{-3}. If this interval does not include sufficient bins, the algorithm always selects at least three so that a result is always obtained. Obviously, the median error E_{med} is then highly biased.

The observed ripple effect may be described in terms of mathematical equations. The fitted Gaussian tail is given by the parameters A_t, σ_t and μ_t which can be used to calculate the timing budget or jitter extend J_t of the left or right bathtub tail.

$$J_t = \mu_t + \sigma_t \cdot Q\left(\frac{1}{A_t \cdot N}\right) \tag{5.33}$$

The error ripple reaches a minimum if the last sample is located just at the tail edge of a distribution, which is the case when J_t is an exact integer multiple of time intervals given by the resolution $2/R$. We can thus write the remainder equation

$$\mathrm{rem}(J_t, 2/R) = \mathrm{rem}(J_t{\cdot}R/2, 1) \overset{!}{=} 0 \tag{5.34}$$

to indicate that the variable $J_t{\cdot}R/2$ should be an integer multiple. Thus the equation

$$\mu_t + \sigma_t \cdot Q\left(\frac{1}{A_t \cdot N}\right) \overset{!}{=} \frac{2}{R} \cdot n \tag{5.35}$$

is obtained, where n is an arbitrary integer value. In figure 5.15(b) the first eight traces are depicted for the given test distribution to prove correct behavior of the equation. If the sample size N can be adjusted, it is highly recommended to fulfill

$$0 < \mathrm{rem}(J_t{\cdot}2/R, 1) < 0.5 \tag{5.36}$$

in order to guarantee that the fitting algorithm operates in a minimum error region.

As a certain drawback, the Gaussian model parameters (A_t, μ_t, σ_t) must be known. In a simulation they can be identified using a numerical approximation of the ideal bathtub function, equivalent to the calculation of tail parameters in section 5.1. At N=10^6 thus, we yield A_t=0.380, μ_t=0.0653 and σ_t=0.0531. Note, that these values form a compromise, since $\{A_t,\mu_t,\sigma_t\}=f(N)$, due to the asymptotic tail behavior in Q-domain.

In a real measurement scenario Gaussian model parameters can be approximated using the median values of multiple tail fits. The remainder function (5.36) is then an indicator on how far the tail edge is located from the outermost distribution sample, and hence, tells whether the fitting result lies within an error maximum or minimum. Equation (5.35) can additionally be inverted to identify suited values for the sample size N.

$$N = \frac{1}{A_t \cdot Q^{-1}\left(\frac{2n/R-\mu_t}{\sigma_t}\right)} \tag{5.37}$$

where n defines the trace number to be located on. If N is adjustable, one is thus able to move from an error maximum to an error minimum by simply changing the sample size. However, this analysis is only valid for a single distribution shape, and cannot be applied to a broad range of distributions. Further the measurement system must not be affected by differential non-linearity (DNL) error, which is rarely the case. DNL is caused by timing mismatches or process variations, and leads to a smoothing of the rippled surface from figure 5.15 as well as a large statistical spread of fitting results. This effect is investigated subsequently.

5.4.3. Error Analysis with Modeled Process Variations

Process variations can significantly affect the accuracy of a BIJM system by causing DNL error. Hence, its influence on the performance of fitting algorithms must be investigated. The major reasons for DNL error are timing mismatches of the delay line and a non-ideal PD structure, which are always present in a real BIJM system. These effects can be modeled using normally distributed random steps with mean $1/R$ and standard deviation σ_{DNL}. The steps start at the synchronization time instant or center of a jitter distribution as shown in figure 5.16, and define the bins where random jitter samples are assigned. Real bin locations thus differ from the equidistant time steps of an ideal measurement system. As an additional problem, the DNL error is summed up over the delay line and yields an integral non-linearity (INL) which significantly exceeds the DNL values. This effect can only be reduced if the PLL output clock is very clean and directly provides multiple phases.

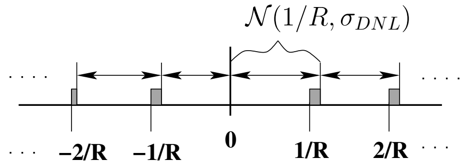


Figure 5.16.: DNL error model to describe the effect of process variations.

Similar to the empirical error analysis, the DNL error term is included as third variable and yields a hyperplane where multiple linear regression can be performed. In order to deal with the large computational demand, a reduced grid resolution is chosen with 25×25 nodes for bin density $\sigma_{RJ}{\cdot}R$=$\{0.8, \ldots, 51.2\}$ UI, as well as N=$\{5{\cdot}10^5, \ldots, 10^8\}$ and N=$\{10^4, \ldots, 10^6\}$ respectively. Each plane is simulated with respect to varying DNL error in the range σ_{DNL}=$\{0.0, \ldots, 0.19\}$ using an equally spaced distance of 0.01.

$\mathbf{DJ}, \sigma_{\mathbf{RJ}}/\mathbf{A}_{\mathbf{DJ}}$		$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{r^2}$
		$\mathbf{N} = [\mathbf{5} \cdot \mathbf{10^5}, \mathbf{10^8}], \quad \sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$					
Sinusoidal $1/2$	E_{med}	1.683	0.144	0.116	-1.859	0.922	0.936
	IQR	1.594	0.138	0.197	-8.185	1.309	0.960
	E_L	0.764	0.138	0.165	-6.582	1.303	0.967
Uniform $1/4$	E_{med}	1.735	0.133	0.102	-1.656	0.761	0.934
	IQR	1.652	0.133	0.190	-8.261	1.410	0.957
	E_L	0.812	0.131	0.152	-6.424	1.326	0.965
Triangular $1/8$	E_{med}	0.470	0.189	0.068	-0.733	0.344	0.970
	IQR	1.578	0.136	0.181	-7.064	1.216	0.948
	E_L	0.227	0.158	0.118	-4.775	1.014	0.964
Quadratic $1/16$	E_{med}	-0.040	0.207	0.040	0.439	-0.066	0.979
	IQR	0.975	0.171	0.148	-5.514	1.093	0.952
	E_L	-0.388	0.188	0.082	-2.997	0.682	0.971
Only RJ	E_{med}	0.948	0.226	0.469	-10.767	3.841	0.807
	IQR	1.097	0.156	0.356	-7.935	0.481	0.958
	E_L	0.407	0.159	0.379	-7.777	0.621	0.964
		$\mathbf{N} = [\mathbf{10^4}, \mathbf{10^6}], \quad \sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$					
Sinusoidal $1/2$	E_{med}	0.736	0.213	0.111	-0.828	0.178	0.963
	IQR	0.924	0.192	0.161	-5.469	1.107	0.951
	E_L	-0.036	0.199	0.138	-3.977	0.835	0.969
Uniform $1/4$	E_{med}	0.807	0.199	0.093	-0.998	0.205	0.968
	IQR	0.488	0.223	0.158	-5.172	1.060	0.955
	E_L	0.278	0.213	0.128	-3.769	0.802	0.972
Triangular $1/8$	E_{med}	0.119	0.221	0.035	-0.246	0.067	0.984
	IQR	0.399	0.230	0.128	-4.520	0.995	0.917
	E_L	-0.606	0.224	0.074	-2.640	0.624	0.959
Quadratic $1/16$	E_{med}	-0.557	0.245	-0.003	-0.148	0.076	0.984
	IQR	0.074	0.259	0.082	-3.264	0.732	0.932
	E_L	-1.117	0.250	0.025	-1.534	0.392	0.971
Only RJ	E_{med}	0.670	0.252	0.489	-2.599	-0.230	0.934
	IQR	-0.424	0.274	0.340	-4.425	0.329	0.954
	E_L	-1.033	0.270	0.363	-4.134	0.244	0.965

Table 5.6.: Coefficients for E_{med}, IQR and E_L with included DNL error, equation (5.38). $\Delta P_t/N$=10^{-3} (large N) and 10^{-2} (small N), $\hat{c}_{1.2}$ algorithm, K=200.

For multiple linear regression a suitable model description was identified with the relation

$$y = -a_0 - a_1 x_1 - a_2 x_2 - a_3 x_3 - a_4 x_2 x_3 \tag{5.38a}$$

$$x_1 = \ln(N),\ x_2 = \ln(\sigma_{RJ} \cdot R),\ x_3 = \ln(1 + \sigma_{DNL}) \tag{5.38b}$$

$$\{E_{med}, IQR, E_L\} = e^y \tag{5.38c}$$

where the last coefficient a_4 also considers a correlation between bin density and DNL error, which proved to significantly increase the quality of fitted hyperplanes. The coefficients are given in table 5.6. Here, smaller r-squared values for the fitted hyperplanes of E_{med} are especially noticed with the pure RJ shape. This is, because the DNL error highly affects statistical tail variations and pure Gaussian distributions suffer from the strongest influence of random noise. Thus, fitted hyperplanes also become inaccurate.

5.5. Design Examples

In order to highlight the practical aspect of the derived equations and empirical relations, two typical design examples are given. The first one is intended for jitter diagnosis, where the measurement time is not crucial. The second one focuses on production testing with stringent requirements on the test time and a small sample size N, it further assumes an undersampling technique as described in [80, 129] for jitter measurements.

5.5.1. Example for Jitter Diagnostics

A BIJM system is assumed to be designed for a 3Gb/s interface with a maximum number of bins R=128. The system shall be able to carry out on-chip diagnostics, where the complete bathtub curve must be measured and fitted within a few hundred milliseconds. The simple measurement scheme in figure 5.1 with an adjustable delay element, one phase detector and a counter, requires a sequential bathtub measurement with N jitter samples at each of R delay steps. With N=10^7 a maximum test time of

$$t_{t,max} = (N \cdot R)/3{\cdot}10^9\,\mathrm{s} = 427\,\mathrm{ms} \tag{5.39}$$

is obtained. This value can be reduced to some degree, because the bins with high error rates quickly collect bit errors and thus, accurate BER values are obtained very fast. Furthermore, tails are assumed to follow a monotonic behavior. One can thus start the measurement at the center of a jitter distribution, and already stop after the first bin without errors. For the given resolution, calculation time of the fitting method affects measurement speed only marginally when carried out off-chip.

In the example, worst case distributions with $\sigma_{t,min}$=0.01UI are assumed, where the major contributor to jitter is inter-symbol interference (ISI) over the transmission channel and thus, DJ is approximately uniform [54]. With both a pessimistic DJ amplitude covering the whole UI and a symmetric TJ distribution, we yield a maximum DJ value (equation (4.2)) of:

$$2 \cdot \mu_{t,max} + 2 \cdot \sigma_{t,min} \cdot Q(10^{-12}) \overset{!}{=} 1\,\mathrm{UI} \tag{5.40}$$

$$\Rightarrow \mu_{t,max} = 0.36\,\mathrm{UI} \tag{5.41}$$

From the inverse of equation (5.5c) we have the jitter ratio $\sigma_{RJ,min}/A_{DJ,uni}$=$9.84{\cdot}10^{-3}$ which is necessary to synthesize this TJ shape. With equation (5.3) the minimum amplitude can now be determined:

$$A_{t,min} = 1.292 \cdot (9.84 \cdot 10^{-3})^{0.955} = 15.6 \cdot 10^{-3} \tag{5.42}$$

Equation (5.10) allows to specify a maximum value for the design parameter ΔP_t, so that the minimum amplitude $A_{t,min}$ can still be fitted correctly.

$$\Delta P_t \leq A_{t,min} \cdot N/6.3 \approx 2.5 \cdot 10^4 \tag{5.43}$$

From the selection chart in figure 5.9 (or by equation (5.20)) ΔP_t=10^4 is chosen equivalent to $\Delta P_t/N$=10^{-3}, which yields a minimum bin density of 0.95. This result is smaller than the minimum imposed by our assumptions:

$$0.95 < \sigma_{t,min} \cdot R = 0.01 \cdot 128 = 1.2 \tag{5.44}$$

Therefore, the selected ΔP_t is applicable and $\sigma_{t,min}$=0.01 UI can be fitted correctly.

The empirical relation (5.5b) can be used to determine $\sigma_{RJ,min}$=$7.57{\cdot}10^{-3}$, since both the minimum ratio $\sigma_{RJ,min}/A_{DJ,uni}$ and $\sigma_{t,min}$ are known. With equation (5.30) and the coefficients from table 5.4 the maximum fitting errors for the sQN method with $\hat{c}_{1.2}$ can finally be determined.

$$E_{med} = 2.0\%,\ IQR = 2.3\%,\ E_L = 5.4\%$$

For the selected minimum variance $\sigma_{RJ,min}$, the resulting coarse bin density is the major contributor to the overall error. This influence becomes even more evident if additionally a DNL error of σ_{DNL}=0.05 UI is assumed and the empirical equation (5.38) with table 5.6 applied:

$$E_{med} = 2.3\%,\ IQR = 3.4\%,\ E_L = 7.4\%$$

which demonstrates that the statistical spread of the fitting method can be highly affected by DNL error. However, this is also an effect of low bin densities, since DNL has been modeled as a random step with time interval $1/R$ (section 5.4.3), and can thus also be reduced using a higher time resolution. This design example will also be continued in section 6.5 to compare the extrapolation error of the sQN method with the conventional Q-normalization (QN) method.

5.5.2. Example for Production Tests

The jitter of a high-speed PLL is assumed to be measured by a second PLL, using an undersampling technique as described in [53, 80, 129]. The total jitter of the PLL should be verified within a maximum of 50ms, in order to guarantee for a sufficiently fast production test of multiple PLLs running at f_s=771.4MHz each [81]. The given architecture moves the sampling position over one delay step $1/R$ after each bit period. Therefore, $N{\cdot}R$ periods are required in order to measure a BER value down to the probability level $1/N$. With a minimum of one counter used in the test structure, this measurement must be repeated R times for each sampling position. A speed-up can also be achieved with C counters in parallel, which yields a test time of

$$t_t = \frac{(N \cdot R) \cdot R}{C \cdot f_s} = \frac{N \cdot R^2}{C \cdot f_s} \tag{5.45}$$

The sQN fitting method shall operate appropriately without being affected by large error oscillations and thus, $\sigma_{RJ}{\cdot}R{\geq}2$ is required (also see figure 5.15). With a selected number of bins R=91, this can be guaranteed if the minimum standard deviation $\sigma_{t,min}{\geq}\sigma_{RJ}$ of the fitted random jitter component is greater or equal to

$$\sigma_{t,min}[s] \geq 2/R \cdot 1/f_s = 28.5\,\text{ps} \tag{5.46}$$

Note that the system is still able to operate down to the minimum value given by equation (5.20), but additionally suffers from error oscillations if $\sigma_{RJ}{\cdot}R{<}2$.

As shown in equation (5.45), the test time t_t can be linearly decreased with a larger number of parallel counters C. Equivalent, the number of samples N can also be increased to improve accuracy of the jitter analysis method, if t_t is kept constant. Thus, the expected worst case extrapolation error E_L of the sQN method can be plotted against the number of implemented parallel counters C, which is a direct measure for the hardware expense. Therefore, empirical relation (5.30) and the table of coefficients 5.4 are applied, together with the regression planes for small sample sizes. In the calculations, we assume the pure Gaussian RJ case as well as sinusoidal DJ, where the latter is typically observed with high-speed PLLs that are affected by spectral spurs.

As a result, the extrapolation error E_L in figure 5.17 is depicted over increasing number of counters C, which also defines the sample size N. N is determined from the inverse of equation (5.45), while a minimum bin density of $\sigma_{RJ}{\cdot}R$=2 is assumed. For the pure RJ case with $C{\geq}14$ and t_t=20ms ($\Rightarrow N{\approx}$26k), the given measurement system is for example able to estimate the TJ of the PLL under test with $<15\%$ error. Note that this result reflects the combined influence of worst case error bias and spread, and includes approximately 97.5% of estimates. The presented design example will also be continued in section 6.5 for comparison with the QN method.

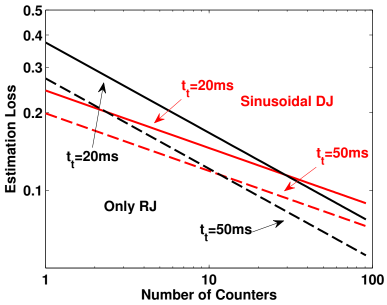


Figure 5.17.: Estimation loss E_L of sQN method over varying number of counters C, with t_t=$\{20, 50\}$ ms, and worst case sinusoidal DJ (red) or pure RJ (black) case.

5.6. Summary

Hardware related design aspects were investigated to utilize the scaled Q-normalization method for on-chip jitter diagnosis or together with built-in jitter measurement (BIJM) system. Influences of limited sample size N as well as number of bins R on the algorithmic performance were investigated. For each analysis, the two algorithmic scenarios $\hat{c}_{1.2}$ (section 4.3) and $Q_{th,c}$ (section 4.4) were investigated independently.

In order to characterize the tail parameters of fitted test distributions, first, the polynomial equations (5.3) and (5.5) were derived. These allow to change between the variable representation prior (σ_{RJ}, A_{DJ}) and after (A_t, σ_t, μ_t) distribution synthesis. The coefficients in table 5.1 are used together with equation (5.3) and specify the tail amplitude A_t obtained with the sQN fitting method. The coefficients in table 5.2 and equation (5.5) specify minimum requirements of the sQN method with respect to the tail parameters σ_t and μ_t. The obtained results are also valid for the conventional Q-normalization (QN) method without scaling (chapter 6) and thus, allow for parameter specification of both methods.

With the $\hat{c}_{1.2}$ algorithm, the minimum tail amplitude $A_{t,min}$ is estimated by defining a conservative threshold, as shown with equation (5.10). The $Q_{th,c}$ fitting algorithm instead allows for the derivation of an exact equation (5.13). This result also highlights the missing link to the first algorithm.

The time resolution variable divides the unit interval into a discrete number of bins R, and causes a limiting effect for maximum tail slope, which can be expressed as minimum Gaussian standard deviation $\sigma_{t,min}$. With $\hat{c}_{1.2}$, equation (5.20) has been derived to identify the $\sigma_{t,min}$ value which can be fitted correctly by the algorithm. Validity of this equation has been demonstrated empirically, and a selection chart for ΔP_t has been given in figure 5.9, in order to simplify a suitable choice. The two conditions in equations (5.6) and (5.10) further guarantee for sufficient outlier suppression as well as a robust algorithmic behavior. With the $Q_{th,c}$ algorithm, equation (5.23) has been derived to give a pessimistic estimate for $\sigma_{t,min}$. A selection chart has been provided in figure 5.11 as well. As a clear advantage of this second algorithm, Q_{min} and ΔP_t can be adjusted independently from the sample size N. Correct convergence of the algorithm is again guaranteed with ΔP_t as large as possible and condition (5.13) fulfilled.

Considering the combined influence of sample size and time resolution on the extrapolation error of fitting algorithms, empirical relations were derived to approximate error bias and spread as a function of sample size N and bin density $\sigma_{RJ,min}{\cdot}R$ (see equation (5.30)). These empirical relations investigate the worst case distribution shapes of the four important DJ types defined in section 3.2.2, as well as the pure Gaussian RJ case. They are meant to aid the designer in finding an optimum performance trade-off. The corresponding empirical coefficients can be found in table 5.4 for $\hat{c}_{1.2}$, and in table 5.5 for the $Q_{th,c}$ algorithm. According to obtained results, the $\hat{c}_{1.2}$ algorithm from section 4.3 clearly highlights a better performance. This is mainly due to the unfavorable behavior of $Q_{th,c}$ at lowest bin densities, which leads to large error oscillations and may even cause negative errors. This highly degrades the extrapolation performance, and also the quality of fitted regression planes.

The observed error oscillations, or error ripple effect at very small bin densities $\sigma_{RJ}{\cdot}R{<}2$ has been investigated as well in section 5.4.2. As a fundamental result, equation (5.35) describes the observable oscillations, and can be used to identify suitable working regions in order to avoid error maximums. However, results behave optimal only for a single distribution shape, which is rather impractical.

In the case of differential non-linearity (DNL) error, as caused by timing mismatches of the jitter measurement system, the statistical spread of TJ_{pp} estimates increases significantly and becomes the major contributor to overall estimation loss. Thus a DNL error model together with empirical equation (5.38) and the table of coefficients 5.6 has been derived. It describes the influence of DNL error on estimation performance in terms of the standard deviation σ_{DNL}.

In a first design example, applicability of the derived equations with respect to jitter diagnosis has been demonstrated. Starting with a worst case RJ of $\sigma_{t,min}{=}0.01\mathrm{UI}$ and a time discretization $R{=}128$, the minimum amplitude $A_{t,min}$ was determined by assuming ISI dominated jitter (uniform DJ). This allowed to correctly specify the design parameter ΔP_t, in order to guarantee for correct convergence of the algorithm. With equation (5.30) and the table of coefficients 5.4 a worst case error bias $E_{med}{\leq}2.0\%$ as well as an estimation loss $E_L{\leq}5.4\%$ were guaranteed if the measurement system was not affected by DNL. Otherwise, with an assumed DNL standard deviation of $\sigma_{DNL}{=}0.05\mathrm{UI}$ and equation (5.38) together with the table of coefficients 5.6, these values increased to $E_{med}{\leq}2.3\%$ and $E_L{\leq}7.4\%$ respectively.

Finally in a second example, the derived empirical relations were applied to a jitter measurement system for fast production testing of high-speed PLLs. The sQN method was able to estimate the true TJ budget in a given measurement time of $t_t{=}20\mathrm{ms}$ with less than 14.9% error (pure RJ case) as well as 13.5% error (sinusoidal DJ), if only the number of counters was increased to $C{=}14$ in order to achieve a larger sample size N.

Parts of this chapter have also been published in [C4,C8].

6. Comparison of Gaussian Tail Fitting Methods Based on Q-Normalization

In this chapter the performance of the scaled Q-normalization (sQN) method is compared with various other tail fitting principles based on the Gaussian quantile normalization. This analysis is meant to give useful insight to the performance and stability of fitting algorithms, and to highlight their advantages and drawbacks. So far, literature is still missing on such comparisons and generally lacking from a detailed performance description. This is partly also due to the high computational demand associated with statistical evaluations. In this work, this problem is dealt by a powerful cluster of up to fifty parallel workstations using 3GHz Intel Xeon processors.

From the variety of histogram based fitting techniques [51,54,84,95,124,136] this chapter only focuses on methods related to the Gaussian Q-normalization principle. Obviously, a chi-squared test as for example used in [52,84,90] would be a prominent candidate, but is omitted here because of the highly individual structure of such algorithms. In fact the optimization process is quite complex and typically includes histogram smoothing, outlier removal, tail part identification and a Gaussian model search over several optimization stages. Implementing a chi-squared test with acceptable accuracy and robust behavior is thus time consuming and hard to achieve. In contrast, the presented fitting methods do not require any data preprocessing, and can directly be applied onto raw jitter distributions.

The first type of fitting algorithm being compared against the sQN method is the conventional Q-normalization (QN) method without scaling. The algorithm is simply obtained by omitting the pre-scaling factor k and directly performs a linear regression analysis in Q-domain, as already mentioned in section 3.1.2. This principle was proposed in [51, 111] and subsequently also described in [82, 123]. The second class of algorithms uses higher order polynomials for tail fitting in Q-domain, and was first suggested by Hong in [54]. The idea is to replace the linear regression stage by a higher order polynomial regression, which fits polynomial functions into the measured Q-tails. The obtained polynomial coefficients thus describe a parameterized bathtub function, which can easily be used to extrapolate distributions down to the BER level of interest and thus, to recover the TJ timing budget. This principle allows for a whole class of polynomial regression methods (QP2, QP3, . . .) to be compared against the sQN method.

In figure 6.1 the optimization scheme for Gaussian Q-normalization, when combined with polynomial regression, is depicted. Polynomials of arbitrary order can be fitted to the distribution tails in Q-domain, where the resulting regression coefficients $\{a_0, \ldots, a_p\}$ denote the functional relation between jitter amplitude x and quantile q. Similar to the sQN method, the regression error $\hat{\sigma}_{err}(n)$ can again be interpreted as a function of tail length n, when starting the regression with

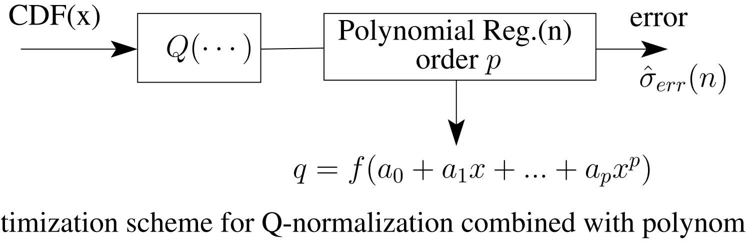


FIGURE 6.1.: Optimization scheme for Q-normalization combined with polynomial regression.

outermost tail samples and recursively moving toward higher probabilities. Therefore, the basic implementation principle of the fitting algorithm remains the same.

Note, that for the special case of order $p{=}1$, the regression function reduces to a line with offset $o{=}a_0$ and slope $s{=}a_1$. Since the scaling factor k is not included, this corresponds to the conventional method without scaling (QN), as described before. In other words, the QN method is equal to the first order polynomial regression (QP1) in Q-domain.

When trying to combine the scaling factor k with higher order polynomials, the optimization becomes unstable and diverges toward meaningless values of $k{<}1$. Thus, only for the linear case with $p{=}1$, the scaling factor k can be part of the optimization scheme. This also confirms the sQN method as three dimensional approach to Gaussian tail parameter search. The QN method instead, with $k{=}A{=}1$ always fits a Gaussian function of maximum amplitude, and can thus only be used to retrieve mean μ and standard deviation σ of the Gaussian model. Further, higher order polynomials (QP2, QP3, ...) cannot be used at all for retrieving Gaussian model parameters.

In the subsequent sections first, an efficient algorithmic implementation as required for the polynomial regression of Q-tails, equivalent to the realization of sQN in section 4.1.4 is described. Then optimum selection criteria for conservative fitting parameters $P_{t,min}$ and ΔP_t are again discussed. This is meant to improve the robustness of polynomial fitting methods by selecting an appropriate tail region. Starting with a performance evaluation of the different methods, polynomials up to the fourth order are investigated, which is sufficient as will also be shown. Then the polynomial methods are compared with sQN, and as a result it is shown that the conventional QN method is also suitable for tail fitting with hardware measurements of coarse time resolutions. Thus, coefficients for the empirical error analysis with QN are derived equivalent to section 5.4. The chapter concludes with a brief summary.

6.1. Implementation of Algorithms

When recalling the algorithmic implementation of the sQN method in section 4.1.4, the testbench for performance analysis can be completely reused since only the algorithm block must be replaced with the respective polynomial fitting algorithm (left flow graph in figure 4.8). The scaling factor k is not utilized, and hence, the algorithm flow graph is greatly simplified.

Figure 6.2 depicts the analysis procedure which corresponds to a simple minimum search of the regression error $\hat{\sigma}_{err}$ over varying tail length n. First, a measured jitter distribution is transformed into Q-domain. The minimum regression error $\hat{\sigma}_{err,min}$ is then determined along the tail length up to a maximum value of $Q{=}0$, which corresponds to half of the Gaussian model. As soon as the optimum tail length is identified, the polynomial coefficients can be retrieved. They are directly used for tail extrapolation and thus, to determine TJ_{pp} values which are again used for error analysis.

The simplified implementation structure is a key advantage compared to the sQN method. The scaling factor k is not present in the optimization scheme anymore and thus, one search dimension is eliminated, which leads to a simplified one-dimensional search space with only the regression length n as unknown variable.

This means a great speed-up for the optimization process, but at the same time the regression error $\hat{\sigma}_{err}$ also becomes the only applicable fitness value. Other measures based on the fitted regression length such as $\hat{n}$ (see section 4.3.1), or combinations of different measures cannot be applied anymore.

With higher order polynomials, the computational demand for calculating the regression coefficients increases as well. Thus, an efficient implementation of the polynomial regression stage is required. The goal is to keep a recursive description of the coefficients, equivalent to equation (4.5) for the linear case, so that the results can easily be obtained from summing terms.

We can describe a polynomial regression of order p as an approximation to a set of tail data pairs (x_i, q_i):

$$q_i = a_0 + a_1 x_i + \ldots + a_p x_i^p, \qquad i = \{1, \ldots, n\} \tag{6.1}$$

The number of n pairs can be arranged according to the Vandermonde [39] system of equations:

$$\begin{bmatrix} 1 & x_1 & \ldots & x_1^p \\ 1 & x_2 & \ldots & x_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^p \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \tag{6.2}$$

When multiplied at the left by the transposed matrix we get:

$$\begin{bmatrix} n & \sum x_i & \ldots & \sum x_i^p \\ \sum x_i & \sum x_i^2 & \ldots & \sum x_i^{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^p & \sum x_i^{p+1} & \ldots & \sum x_i^{2p} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sum q_i \\ \sum x_i q_i \\ \vdots \\ \sum x_i^p q_i \end{bmatrix} \tag{6.3}$$

This system of equations contains only summing terms which can easily be updated using recursions, as required by the polynomial fitting stage. To obtain the regression coefficients $\{a_0, \ldots, a_p\}$ the matrix inverse must be calculated. The symmetric arrangement of summing terms corresponds to a Hankel matrix, which can be inverted very efficiently using the Levinson-Durbin algorithm. Here, an implementation from [112] is used, which requires only $3p^2 + 9p + 3$ multiply and divide operations, and is thus sufficiently fast for analyses up to the required polynomial of order p=4. The regression error is calculated as standard deviation of a fitted polynomial with the tail data:

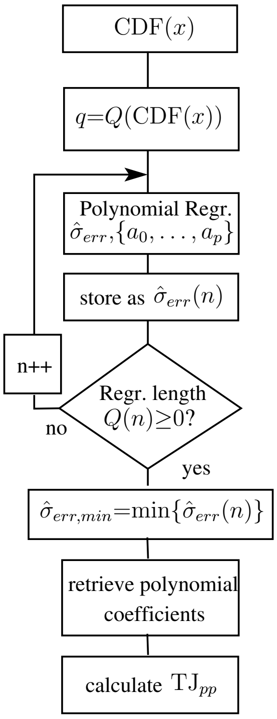

CDF(x)
q=Q(CDF(x))
Polynomial Regr.
$\hat{\sigma}_{err}$,$\{a_0, \ldots, a_p\}$
store as $\hat{\sigma}_{err}(n)$
n++
Regr. length
$Q(n) \geq 0$?
no
yes
$\hat{\sigma}_{err,min}$=min$\{\hat{\sigma}_{err}(n)\}$
retrieve polynomial coefficients
calculate TJ$_{pp}$

Figure 6.2.: $\hat{\sigma}_{err}$ based polynomial fitting.

$$\hat{\sigma}_{err}(n) = \sqrt{\frac{\sum_{i=1}^{n} (q_i - a_0 - a_1 x_i - \ldots - a_p x_i^p)^2}{n - (p+1)}} \tag{6.4}$$

The Levinson-Durbin algorithm is included with the C/C++ simulation environment, where fast statistical simulations are carried out. As with the sQN implementation, this allows for an in-depth analysis of the algorithmic performance. Further, MATLAB is again used for post processing of fitting results as well as for data representation.

In section 4.1.4 the computational effort of sQN and QN methods was already compared. With the very efficient Levinson-Durbin recursion, the difference between QN and higher order polynomials is marginal and thus, analyses of QP2, QP3 and QP4 methods are omitted here.

6.2. Performance Optimization

Equivalent to the sQN method, analyses can be carried out to optimize the polynomial fitting algorithms with respect to estimation error and robustness. Thus optimum parameter regions are derived for every polynomial regression order individually, in order to obtain best estimation results for each.

As already demonstrated with the simplified flow graph in figure 6.2, the minimum error can be determined very quickly without amplitude scaling factor k. The minimum search is simply carried out along the tail length n, where the error $\hat{\sigma}_{err}$ is the goodness-of-fit measure.

When optimizing the performance, one likes to determine a best suited parameter configuration for each of the polynomial fitting methods, equivalent to section 4.3.2. For the sQN method the default configuration in equation (4.21) was derived as suitable interval for initial tail selection. This configuration avoids outliers caused by statistical tail variations, and thus supports a robust fitting behavior. For the polynomial methods the conservative parameters now have to be re-specified.

Subsequent optimizations again use the median error E_{med} and interquartile range IQR of TJ_{pp} estimates for performance analysis, since they are less influenced by outliers compared to mean and standard deviation. In addition, the estimation loss E_L was defined in equation (3.18) to consider both error bias and spread. Further, higher order moments such as skewness ξ and kurtosis κ are used for investigating the outlier behavior.

From the three conservative tail fitting parameters ΔP_t, ΔT_t and $P_{t,min}$, as defined and described in section 4.3.2, ΔT_t exhibited poor performance, unless the Gaussian tail length was known, which is usually not the case. Therefore, we only focus on the analysis of threshold $P_{t,min}$ and probability interval ΔP_t instead. Subsequently, performance optimizations are carried out for each of the polynomial orders, starting with the first order or conventional Q-normalization method.

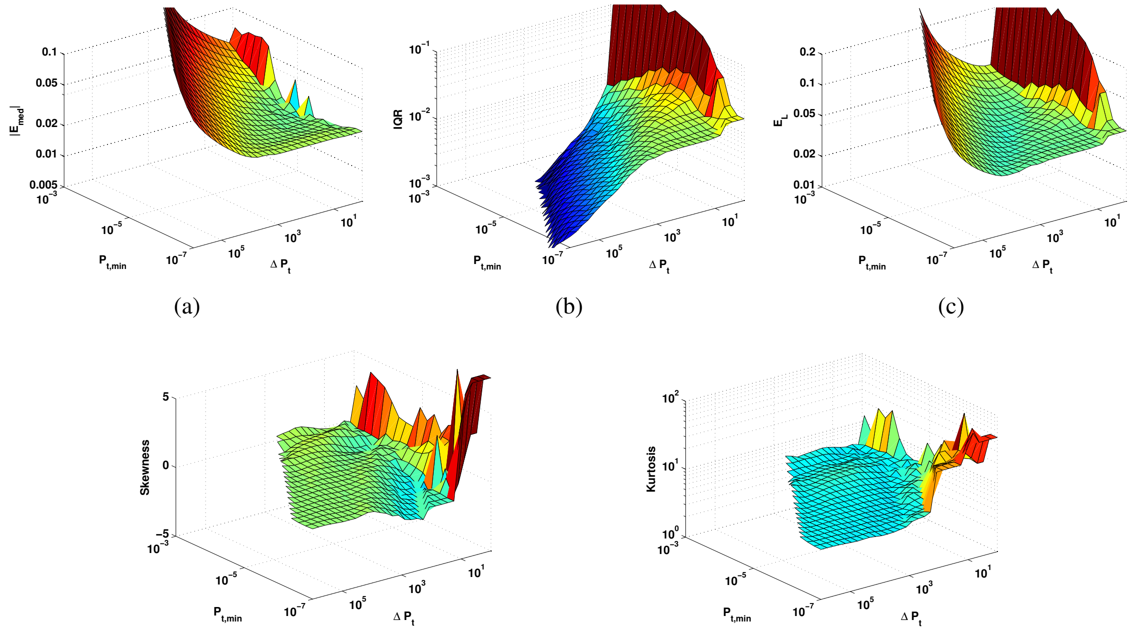
First Order Polynomials

In figure 6.3 the fitting behavior of the conventional Q-normalization (QN) method with first order (linear) regression is analyzed. Statistical performance measures are plotted as surfaces of varying parameters ΔP_t and $P_{t,min}$, while the selected test distribution has a worst case jitter ratio of $\sigma_{RJ}/A_{DJ,uni}{=}1/8$ and uniform DJ type. Similar to prior analyses with sQN (see figure 4.23), the median error E_{med}, interquartile range IQR, estimation loss E_L, skewness ξ and kurtosis κ (third and fourth order statistical moments) are reported in each of the subfigures. The kurtosis has already been utilized as a measure for outlier presence. The skewness now additionally describes the asymmetry of estimates, and thus denotes whether distribution data is centered at the left ($\xi{<}0$) or right ($\xi{>}0$) from the mean value.

The resulting surfaces behave quite different from the ones obtained with the sQN method. The median error E_{med} in figure 6.3(a) approaches the convergence limit for the variable product $\Delta P_t{\cdot}P_{t,min}$ very slowly and thus, rather forms a smooth transition than a clear cut off. As expected, the error bias is generally larger compared to sQN, since the fitting algorithm is based on the simplified two dimensional tail model without Gaussian amplitude search.

The statistical spread, expressed by the interquartile range IQR in figure 6.3(b), indicates a significant decrease toward higher values of the variable product $\Delta P_t{\cdot}P_{t,min}$, which is in clear contrast to the median estimation error. This causes the overall error E_L in figure 6.3(c) to remain constant over a large parameter range. Optimum performance can for example be achieved with $\Delta P_t{\approx}10^4$ and $P_{t,min}{=}1/N{=}10^{-7}$, which is however only valid for the given distribution case. Other test distributions with smaller tail amplitudes also yield surfaces where the convergence region is narrowed down, and given by a smaller $\Delta P_t{\cdot}P_{t,min}$ product.

Skewness ξ and kurtosis κ in figures 6.3(d) and 6.3(e) indicate a large amount of outliers if the variable product $\Delta P_t{\cdot}P_{t,min}$ is too small. A negative ξ means heavy tails directing toward smaller TJ budgets. From the observed plane, at least $\Delta P_t{\cdot}P_{t,min}{\geq}10^3$ is thus recommended to successfully suppress outliers. This is in clear contrast to the sQN method (see figure 4.23), where a smaller tail region was already sufficient. The observed effect can be explained by a stronger asymptote in Q-domain compared to sQN, since the amplitude scaling factor is omitted. However,



(a) (b) (c)

(d) (e)

FIGURE 6.3.: First order polynomial regression (QN): median error E_{med}, interquartile range IQR, estimation loss E_L, skewness ξ and kurtosis κ surfaces over varying ΔP_t and $P_{t,min}$. Test distribution: σ_{RJ}/A_{DJ}=1/8, $A_{DJ,uni}$=0.2 UI, σ_{RJ}=0.025 UI, N=10^7, K=250.

with a negative ξ, outliers are mostly located closer to the true timing budget, and are therefore uncritical.

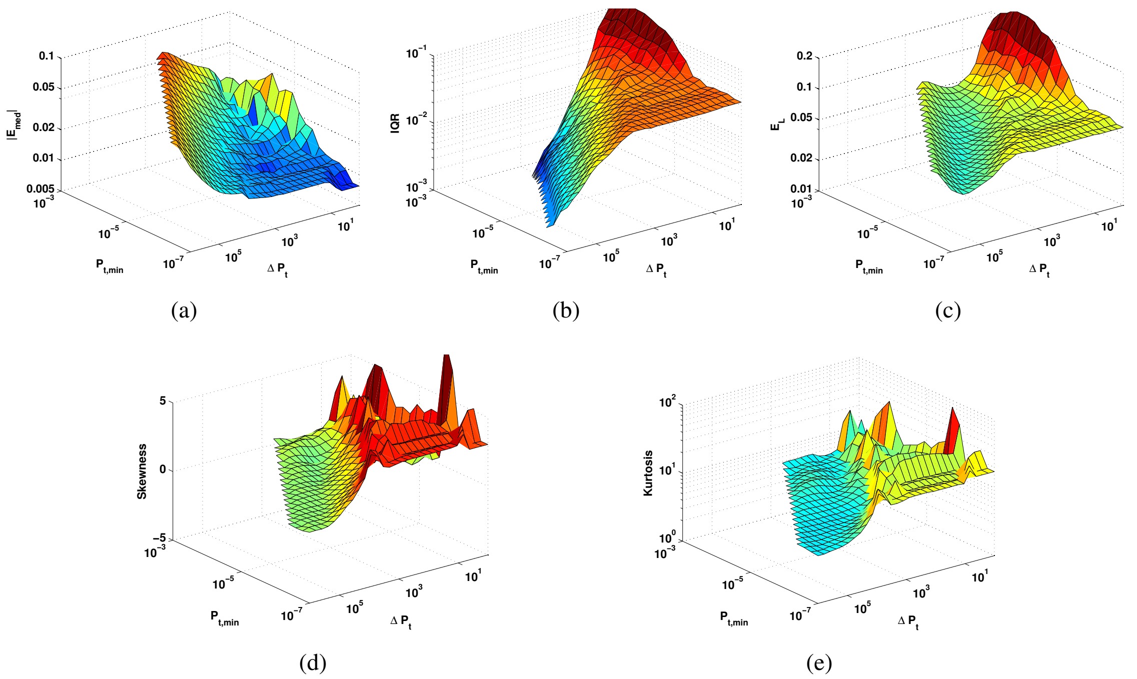
To summarize the properties of the QN method, best performance is achieved with the parameter product $\Delta P_t{\cdot}P_{t,min}$ chosen as large as possible. This avoids convergence failures as well as negative skewed error distributions. In the subsequent performance comparison the same test distributions will be used as for the analysis of the sQN method (see section 4.3.3). If the outlier presence at small sample size N is neglected, the same ΔP_t values can thus be utilized.

Second Order Polynomials

Figure 6.4 shows the fitting performance of the second order polynomial regression (QP2), where the worst case test distribution has again been identified at a jitter ratio $\sigma_{RJ}/A_{DJ}{\approx}1/8$. The performance metrics are the same as in figure 6.3.

Median error E_{med}, interquartile range IQR and estimation loss E_L in the upper subfigures behave quite similar to first order regression analysis. The magnitude of E_{med} in figure 6.4(a) is significantly smaller than with first order polynomials, but this benefit is lost due to an increased statistical spread. The overall estimation loss E_L in figure 6.4(c) highlights an optimum parameter region for the variable product $\Delta P_t{\cdot}P_{t,min}$, forming a distinct valley. This is due to the IQR course with a falling edge along the surface in figure 6.4(b), which already starts before the rising edge of E_{med} in figure 6.4(a) influences the overall E_L. Inside this region, best estimates are obtained.

Skewness and kurtosis evidence a critical drawback for TJ estimates obtained from second order polynomials. If only a small part of the distribution tail is selected for polynomial regression, the resulting estimates will be highly scattered and contain a large amount of outliers. In fact,



(a)

(b)

(c)

(d)

(e)

Figure 6.4.: Second order polynomial regression (QP2): E_{med}, IQR, E_L, ξ and κ surfaces over varying ΔP_t and $P_{t,min}$. Test distribution: $\sigma_{RJ}/A_{DJ}{=}1/8$, $A_{DJ,uni}{=}0.2\,$UI, $\sigma_{RJ}{=}0.025\,$UI, $N{=}10^7$, $K{=}250$.

the positive skewness clearly indicates a heavy tail directing toward higher TJ estimates, and even reaches an undesired peak for a worst case configuration. As will be shown later on with performance evaluations, the median error E_{med} is negative for the selected test distribution. Thus, to some degree one may tolerate positive skewed tails, but the magnitudes in figure 6.4(d) are by far too large.

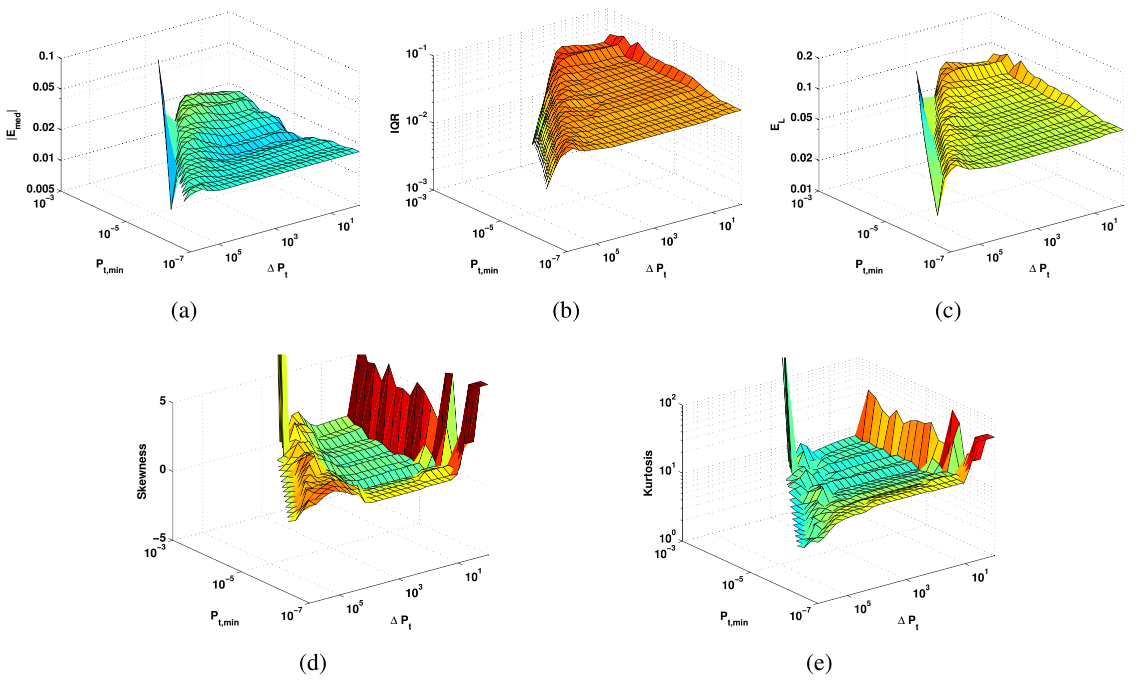
In order to ensure that second order polynomials achieve correct fitting results without outliers and high skewness, a sufficiently large parameter product $\Delta P_t{\cdot}P_{t,min}$ must be selected. If the threshold parameter $P_{t,min}$ is not in use, the minimum probability corresponds to the granularity of $P_{t,min}{=}1/N{=}10^{-7}$. The probability region for best tail fitting is then located at $\Delta P_t{=}5{\cdot}10^5$, which is here used as default value for second order polynomials. It must still be adapted accordingly for use with different sample sizes. Fortunately, the fitted tail part is not anymore restricted to the linearized Q-tail, and may also include parts of the DJ component in order to achieve better extrapolations. However, it becomes impossible to find an optimum parameter configuration which is well suited for all test distribution shapes. Here we simply choose $\Delta P_t{=}5{\cdot}10^{-3}{\cdot}N$ to guarantee at least for a sufficiently large fitting region.

Third and Fourth Order Polynomials

The same analysis from the previous paragraphs can also be carried out to investigate the estimation performance of third and fourth order polynomials (QP3, QP4). Worst case test distributions with uniform type DJ have again been identified at the jitter ratio $\sigma_{RJ}/A_{DJ,uni}{\approx}1/8$.

With both polynomial orders, the regression error described by E_{med}, IQR and E_L cannot be influenced by the fitting parameter ΔP_t. Also the threshold parameter $P_{t,min}$ provides best performance only if reduced to its minimum $P_{t,min}{=}1/N$.

Unfortunately, the statistical random variation of tails has a highly misleading effect on higher



(a)

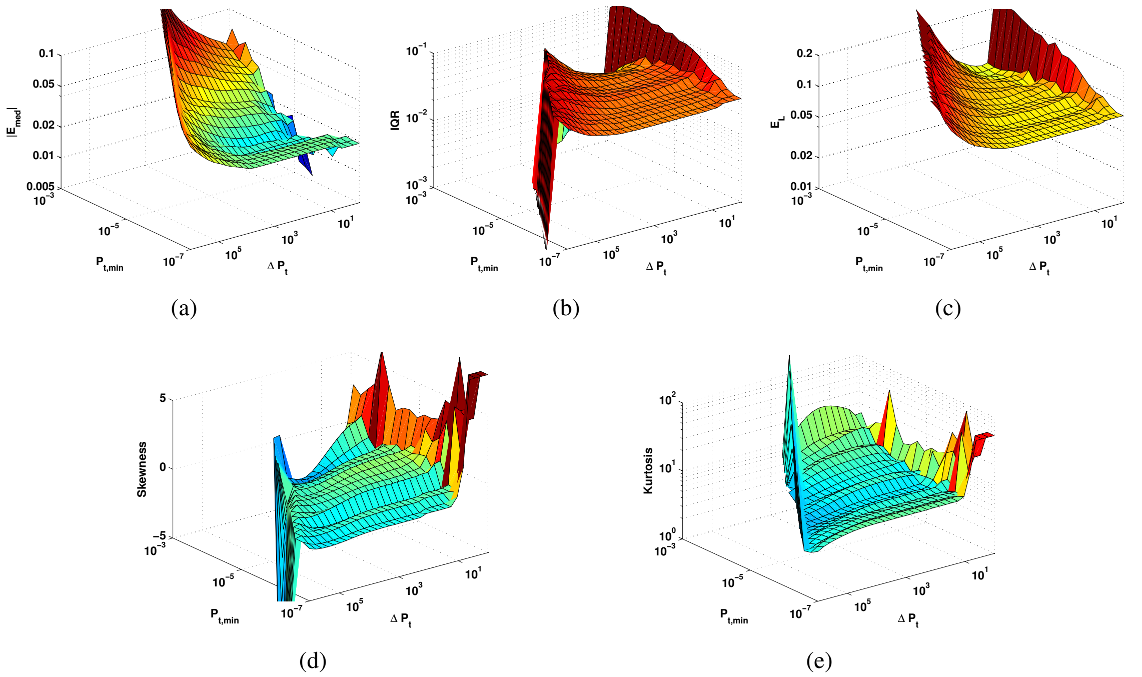
(b)

(c)

(d)

(e)

Figure 6.5.: Third order polynomial regression (QP3): E_{med}, IQR, E_L, ξ and κ plots for ΔP_t versus $P_{t,min}$. σ_{RJ}/A_{DJ}=1/8, $A_{DJ,uni}$=0.2 UI, σ_{RJ}=0.025 UI, N=10^7, K=250.



(a)

(b)

(c)

(d)

(e)

Figure 6.6.: Fourth order polynomial regression (QP4): E_{med}, IQR, E_L, ξ and κ plots for ΔP_t versus $P_{t,min}$. σ_{RJ}/A_{DJ}=1/8, $A_{DJ,uni}$=0.2 UI, σ_{RJ}=0.025 UI, N=10^7, K=250.

order polynomials, which generally impedes them to achieve accurate fitting results. In the case of third order polynomials, skewness and kurtosis show heavy tails toward both sides (skewness is close to zero while kurtosis is large). These heavy tails are caused by convergence failures of the regression stage, where the fitted polynomial does not extrapolate the bathtub correctly down to the target BER. This can for example be the result of a non-monotonic course of the fitted polynomial, which contains a local minimum and/or maximum. Such a failure can be identified and dealt by fitting a simple, strictly monotonic linear function into the distribution tail. However, this corresponds to a reduction of the regression order and thus, introduces a different error characteristic, which finally leads to the heavy tails observed.

Fourth order polynomials highly suffer from random variations of measured bathtub tails. Although skewness is almost zero and kurtosis reaches small values close to the normal distribution case, the IQR plane shows an extremely large statistical spread. This highlights a general problem of tail fitting algorithms with higher order polynomials. Instead of converging toward a simple regression function, they tend to follow random data variations which mislead the extrapolation result. In regression analysis this effect is commonly known as overfitting problem.

The ΔP_t parameter selection for third and fourth order polynomials is uncritical as it does in fact not influence the estimation performance. We thus select a default value of $\Delta P_t{=}10^3$ for both polynomial orders.

The performance optimization of polynomial fitting methods concludes with a brief summary. Each of the fitting methods has been investigated individually, in order to find an adapted, well suited parameter configuration which drives the regression stage. These configurations are summarized in table 6.1. Note that the parameters were selected using only uniform DJ type with the worst case test distributions identified at $\sigma_{RJ}/A_{DJ}{=}1/8$. They do not consider triangular or quadratic curve shaped DJ. Thus, parameter configurations are primarily meant as compromise solutions, where especially the higher order polynomials cannot be utilized over a broad variety of test shapes. If for example, specific shapes are known, other configurations may be more suitable.

Polynomial	$\mathbf{\Delta P_t}$				
Order	$\mathbf{N = 10^4}$	$\mathbf{N = 10^5}$	$\mathbf{N = 10^6}$	$\mathbf{N = 10^7}$	$\mathbf{N = 10^8}$
1^{st}	10^1	10^2	10^3	10^3	10^3
2^{nd}	$5 \cdot 10^1$	$5 \cdot 10^2$	$5 \cdot 10^3$	$5 \cdot 10^4$	$5 \cdot 10^5$
3^{rd}	10^1	10^2	10^3	10^3	10^3
4^{th}	10^1	10^2	10^3	10^3	10^3

Table 6.1.: Default parameter configuration of ΔP_t for polynomial tail fitting methods, derived with worst case test distributions $\sigma_{RJ}/A_{DJ}{=}1/8$ of uniform DJ type.

For $N{\geq}10^6$ the parameter ΔP_t is always constant besides second order polynomials, where it must be chosen sufficiently large to avoid misleading outliers. By selecting $\Delta P_t{=}5{\cdot}10^{-3}{\cdot}N$, the optimum region from figure 6.4 is guaranteed, which has also been verified for different parameter surfaces over varying N. ΔP_t does not influence the performance of third and fourth order polynomials, and was thus set to the default sQN values from equation (4.21). With first order polynomials, parameter selection is more critical since the product $\Delta P_t{\cdot}P_{min}$ must not exceed the minimum tail amplitude of test distributions. To guarantee this also for a small sample size $N{=}\{10^4, 10^5\}$, the default sQN configuration is used as well. However, in this case the QN method is also affected by outliers.

6.3. Performance Analysis of Polynomial Fitting Methods

In this section a performance analysis for each of the described polynomial fitting methods is carried out, using the parameter configurations from table 6.1. Equivalent to the sQN method in section 4.2 the extrapolation error is investigated over varying sample size N and the four different DJ shapes.

The influence of N is thus shown in figures 6.7-6.10, where E_{med} and E_L are plotted for each of the different polynomial regression orders. The median error of first order polynomials (linear case) is always positive, which is due to the asymptote in Q-domain as already described together with the scaled Q-normalization (sQN) method in section 4.2.1. Estimated TJ_{pp} values therefore tend to be larger than the true timing budget, and are thus pessimistic. As can be seen in the lower E_{med} subfigures, this beneficial property is not maintained with higher order polynomials (QP2, QP3, QP4).

With decreasing sample size N, both median error and statistical spread increase, which leads to a higher overall estimation loss E_L as depicted in the right column of subfigures. Especially with third and fourth order polynomials, this effect is additionally supported by a significant statistical spread at larger jitter ratios σ_{RJ}/A_{DJ} (RJ dominant case), which impedes accurate tail fitting. Note, when varying the number of jitter samples N, also the conservative tail fitting parameter ΔP_t is adapted in order to keep the fitting algorithms working in their optimized parameter region (see table 6.1).

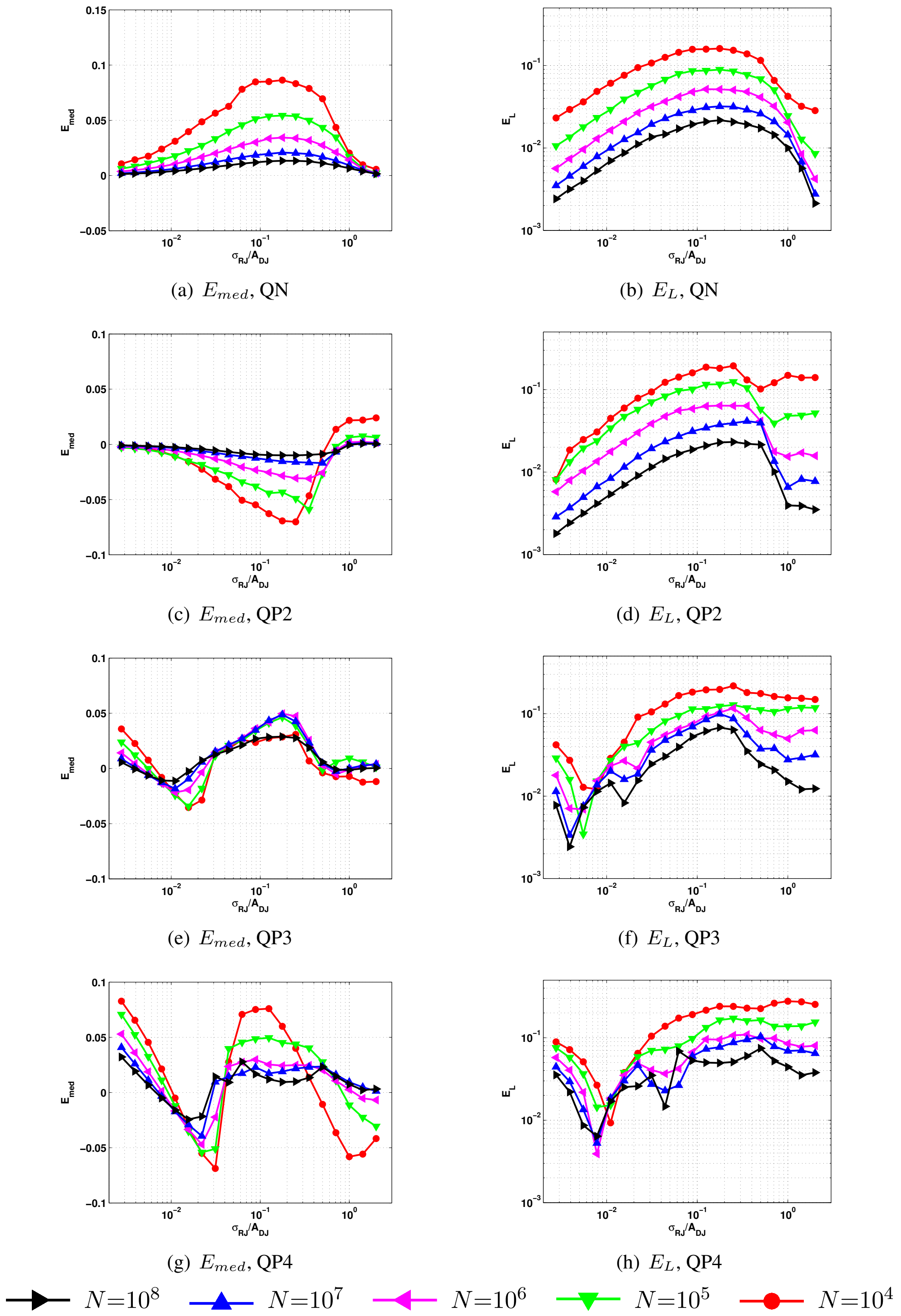
When comparing the fitting performance over different DJ shapes (figures 6.7-6.10), we notice that the QN method with first order polynomials still maintains its pessimistic extrapolation property. However, triangular and quadratic curve shaped DJ are highly biased at smallest σ_{RJ}/A_{DJ} and for $N{\leq}10^6$, because the tail amplitudes fall below the analyzable minimum given by equation (5.10):

$$A_{t,min} = 6.3 \cdot \Delta P_t/N = 6.3 \cdot 10^{-3} \tag{6.5}$$

Hence, the ΔP_t parameter forces the fitting algorithm to include parts of the DJ component and thus, misleads the TJ estimation. This effect can only be overcome by further reducing ΔP_t, which would however highly increase the presence of outliers. A large error bias is also observed with higher order polynomials at smallest jitter ratios, which is also due to very steep tails. Since the Gaussian component is barely visible at the distribution tails, these algorithms here rather converge toward the bounded, Gaussian-like DJ shape.

Another interesting effect appears with QP2 at $N{\geq}10^6$ and triangular or quadratic curve shaped DJ. If a certain minimum ratio can be guaranteed, equivalent to a lower bound for the Gaussian tail slope in Q-domain, second order polynomials yield excellent fitting results down to $\sigma_{RJ}/A_{DJ}{\approx}2{\cdot}10^{-2}$. They clearly outperform the QN method, and can compete with it for the complete range of sinusoidal and uniform DJ shapes.

Third and fourth order polynomials only produce estimates with acceptable accuracy over a small range of jitter ratios. Further, they are not able to track uniform and sinusoidal DJ at large jitter ratios, where the tails of Q-normalized distributions correspond to simple linear functions. While the median error E_{med} in figures 6.7(e) and 6.7(g) for example, approximately converges toward zero at highest jitter ratios (pure Gaussian case), the estimation loss in figures 6.7(f) and 6.7(h) reaches a large constant. This effect is further magnified toward a decreasing sample size N. It clearly indicates an over-fitting problem of higher order polynomials, caused by the statistical random variations of tails.



(a) E_{med}, QN

(b) E_L, QN

(c) E_{med}, QP2

(d) E_L, QP2

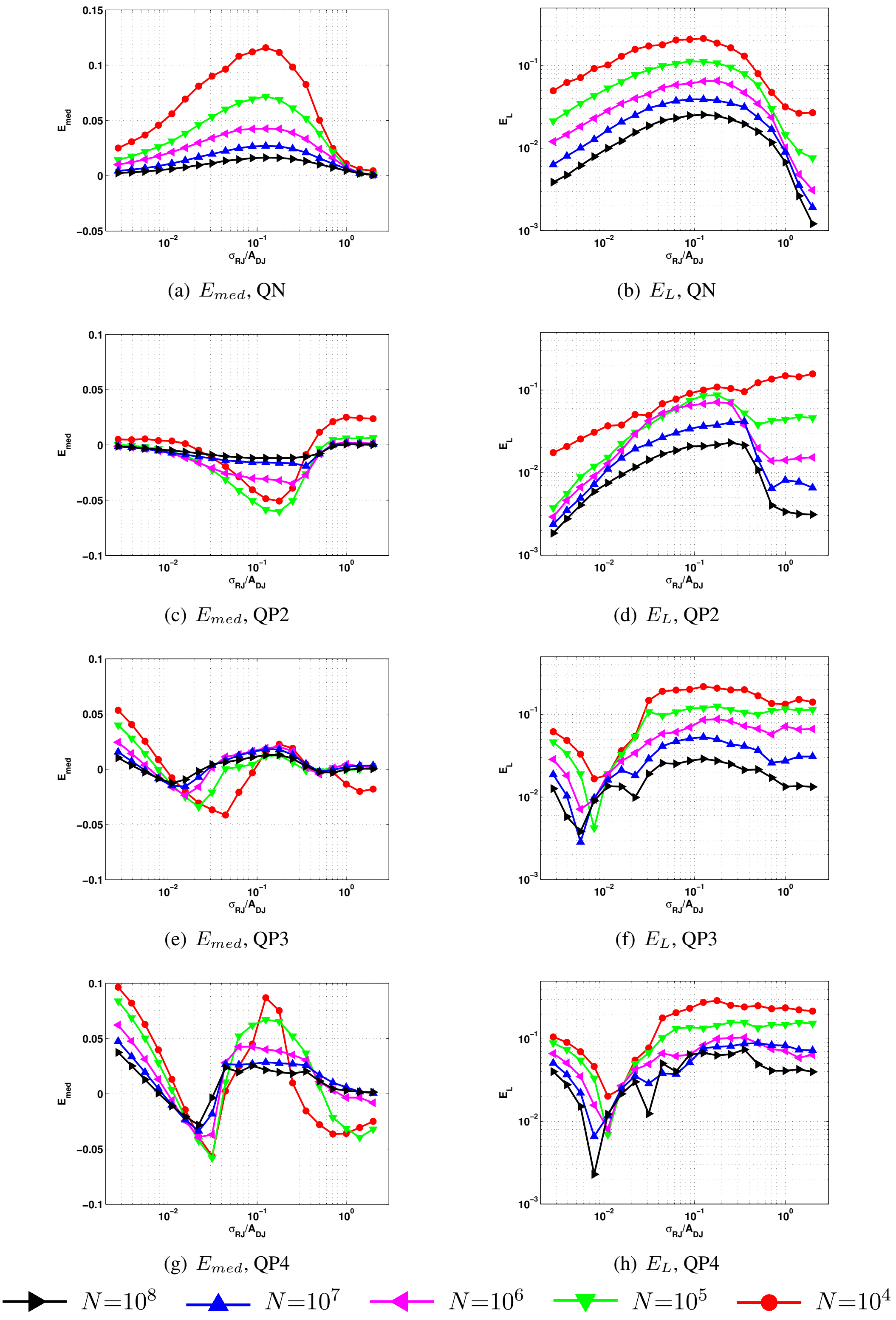
(e) E_{med}, QP3

(f) E_L, QP3

(g) E_{med}, QP4

(h) E_L, QP4

Figure 6.7.: Sinusoidal type DJ: median error E_{med} (left) and estimation loss E_L (right) for polynomial tail fitting methods QN (a,b), QP2 (c,d), QP3 (e,f) and QP4 (g,h). N=$\{10^4, 10^5, 10^6, 10^7, 10^8\}$, K=250.



(a) E_{med}, QN

(b) E_L, QN

(c) E_{med}, QP2

(d) E_L, QP2

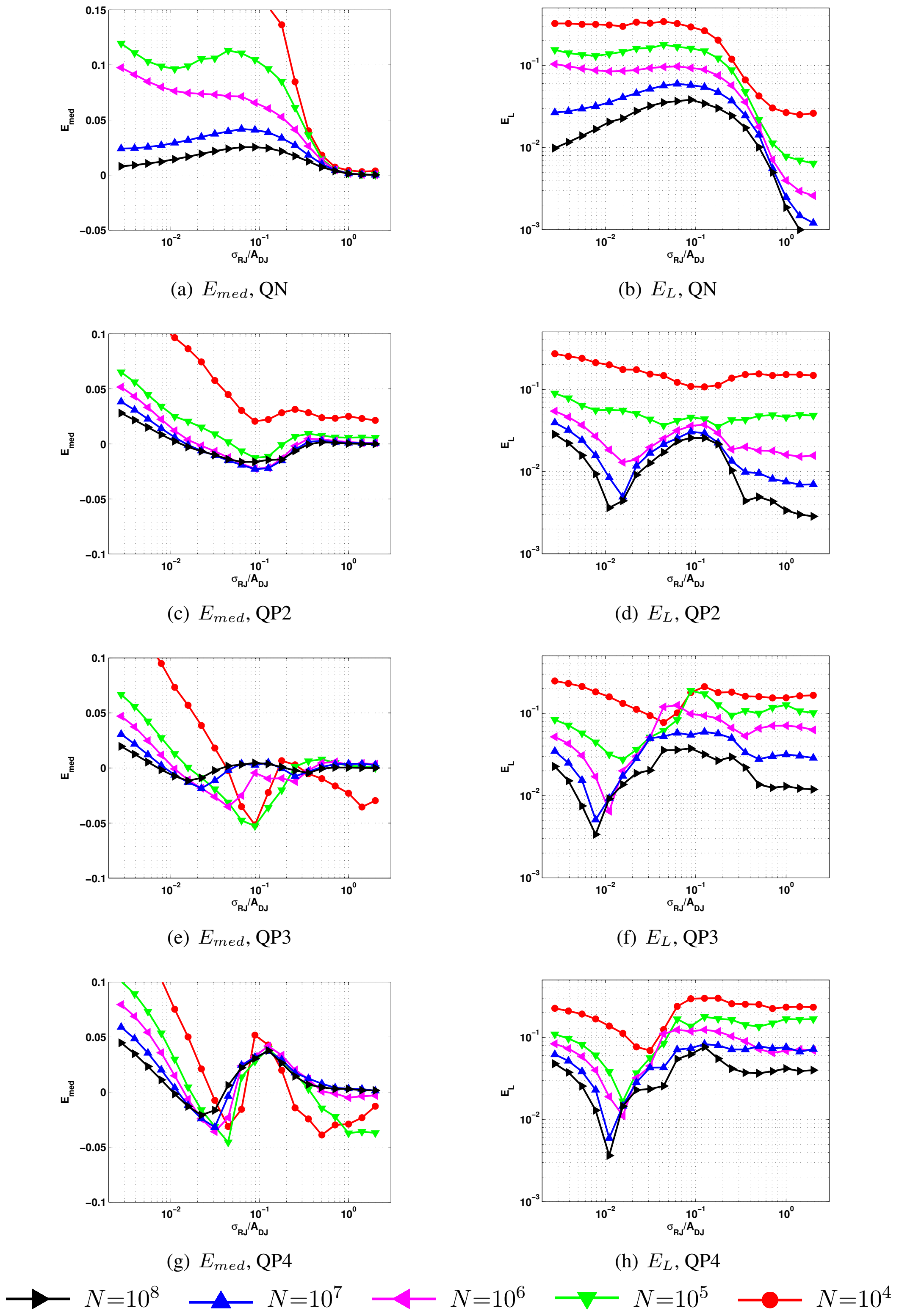
(e) E_{med}, QP3

(f) E_L, QP3

(g) E_{med}, QP4

(h) E_L, QP4

Figure 6.8.: Uniform type DJ: median error E_{med} (left) and estimation loss E_L (right) for polynomial tail fitting methods QN (a,b), QP2 (c,d), QP3 (e,f) and QP4 (g,h). N=$\{10^4, 10^5, 10^6, 10^7, 10^8\}$, K=250.



(a) E_{med}, QN

(b) E_L, QN

(c) E_{med}, QP2

(d) E_L, QP2

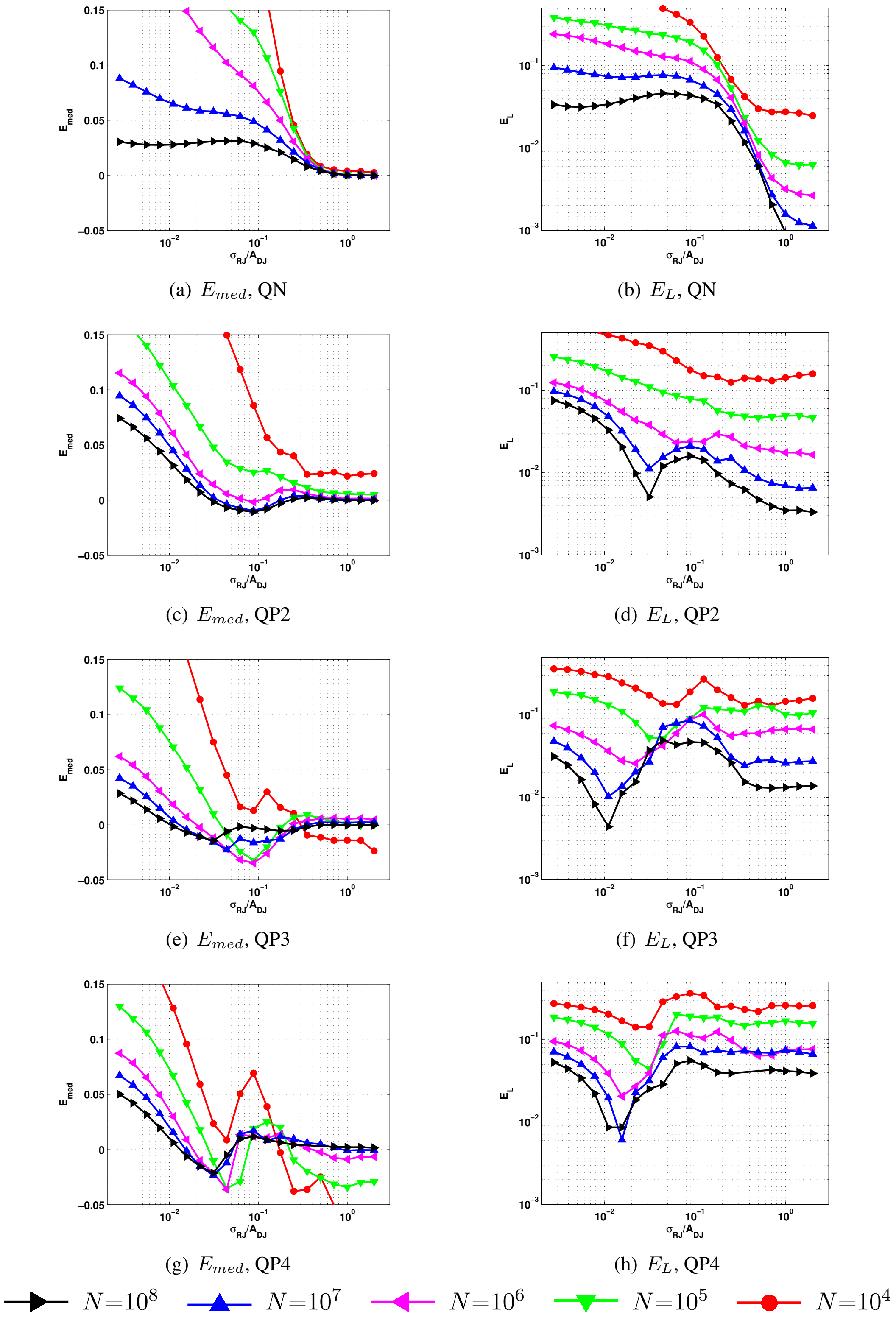
(e) E_{med}, QP3

(f) E_L, QP3

(g) E_{med}, QP4

(h) E_L, QP4

Figure 6.9.: Triangular type DJ: median error E_{med} (left) and estimation loss E_L (right) for polynomial tail fitting methods QN (a,b), QP2 (c,d), QP3 (e,f) and QP4 (g,h). N=$\{10^4, 10^5, 10^6, 10^7, 10^8\}$, K=250.



(a) E_{med}, QN

(b) E_L, QN

(c) E_{med}, QP2

(d) E_L, QP2

(e) E_{med}, QP3

(f) E_L, QP3

(g) E_{med}, QP4

(h) E_L, QP4

Figure 6.10.: Quadratic curve type DJ: median error E_{med} (left) and estimation loss E_L (right) for polynomial tail fitting methods QN (a,b), QP2 (c,d), QP3 (e,f) and QP4 (g,h). N=$\{10^4, 10^5, 10^6, 10^7, 10^8\}$, K=250.

6.4. Comparison with Scaled Q-normalization Method

The polynomial fitting methods (QN, QP2, QP3, QP4) can also be compared against the scaled Q-normalization (sQN). Therefore, the optimized method from section 4.3.3 is used, with $\hat{c}_{1.2}$ based optimization and the ΔP_t settings from equation (4.21). The performance comparison is first carried out over varying sample size N as well as different DJ distribution types. Finally, also the influence of limited time resolution R is investigated.

Influence of Sample Size and Distribution Shape

The comparison of algorithmic performance is carried out in figures 6.11-6.15 where E_{med} and E_L are evaluated over varying distribution shape and DJ type, for each of the five different fitting algorithms sQN, QN, QP2, QP3 and QP4. At the moment, the influence of time quantization R is not considered and thus, R_{sim}=3.3·10^5 is used as equivalent number of bins for simulators.

As can be seen from the different figures, the sQN method achieves best performance. The three-dimensional Gaussian parameter search (A, σ, μ) allows for accurate tail extrapolations, and the TJ timing budget is determined in a robust manner. Nevertheless, due to the additional scaling factor inside the optimization scheme, this also means an additional degree of freedom, which significantly increases computational demand (see figure 4.9).

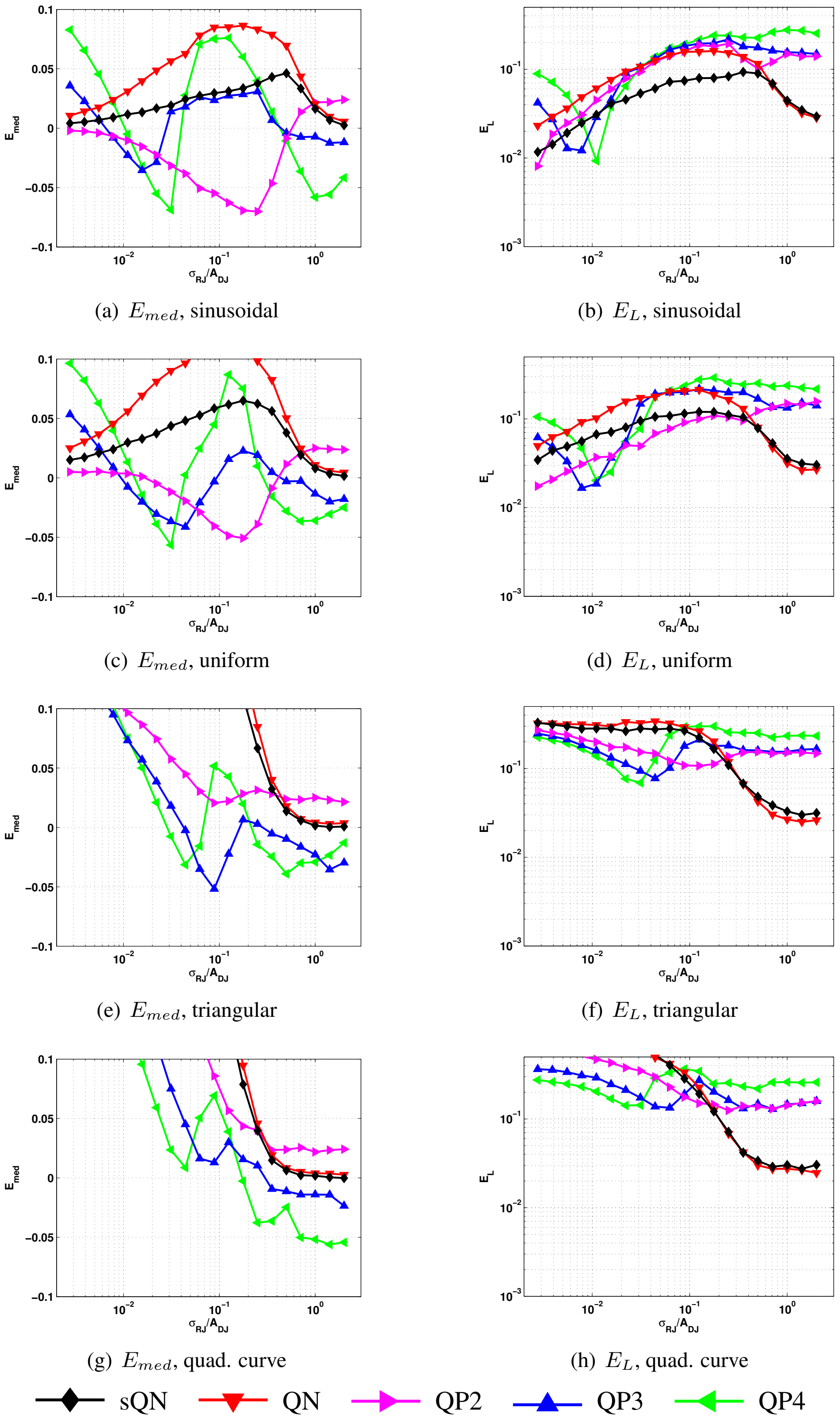
The conventional QN method in contrast, simply assumes a Gaussian tail area of A=1 and thus, only determines the two parameters σ and μ. The method is faster, but does not estimate the TJ budget as accurate as sQN. The magnitudes of E_{med} show that both sQN and QN methods only produce estimates of positive error bias. This is due to the asymptotic tail behavior in Q-domain (see section 4.2.1). Estimated TJ values are therefore always pessimistic, which is a clear advantage of the two methods. As can be seen with higher order polynomials (QP2, QP3, QP4) this beneficial property is not maintained. Additionally, polynomials do not allow for recovering Gaussian model parameters.

With decreasing sample size, both median error and statistical spread increase, which generally leads to a large overall estimation loss E_L. With third and fourth order polynomials, this effect is additionally supported by an especially large statistical spread at higher jitter ratios σ_{RJ}/A_{DJ}. This generally impedes higher order polynomials to be utilized for accurate tail extrapolation, except for certain known shapes.

Influence of DJ Distribution Type

In figures 6.11-6.15 the four different DJ types are investigated as well. Error bias of QN is significantly larger than with sQN, but remains strictly positive. QP2 estimates yield excellent fitting results for Gaussian-like DJ types, which is due to an optimum working region. Thus, if a certain minimum jitter ratio of $\sigma_{RJ}/A_{DJ}{\geq}2 \cdot 10^{-2}$ can be guaranteed (like a lower bound for Gaussian tail slope in Q-domain) and $N{\geq}10^6$, QP2 can also be utilized for tail fitting. At sinusoidal and uniform DJ shapes the algorithm performance is similar to QN, and for quadratic curve DJ it even outperforms the sQN method. Thus, as long as the lower bound is guaranteed, QP2 can theoretically replace the QN method. However, note that the beneficial property of pessimistic TJ estimation is lost with the QP2 algorithm. Further, it cannot be used to characterize Gaussian model parameters.

Regression analysis with third and fourth order polynomials (QP3, QP4) only yields estimates of acceptable accuracy in a small range of jitter ratios. As an interesting effect, they are not able to track the distribution tails correctly at highest jitter ratios, where linear functions are given in Q-domain. Although the error bias E_{med} converges toward zero, the estimation loss of QP3 and QP4 reaches a large constant (for example see E_L subplots in figures 6.13). This behavior clearly



(a) E_{med}, sinusoidal

(b) E_L, sinusoidal

(c) E_{med}, uniform

(d) E_L, uniform

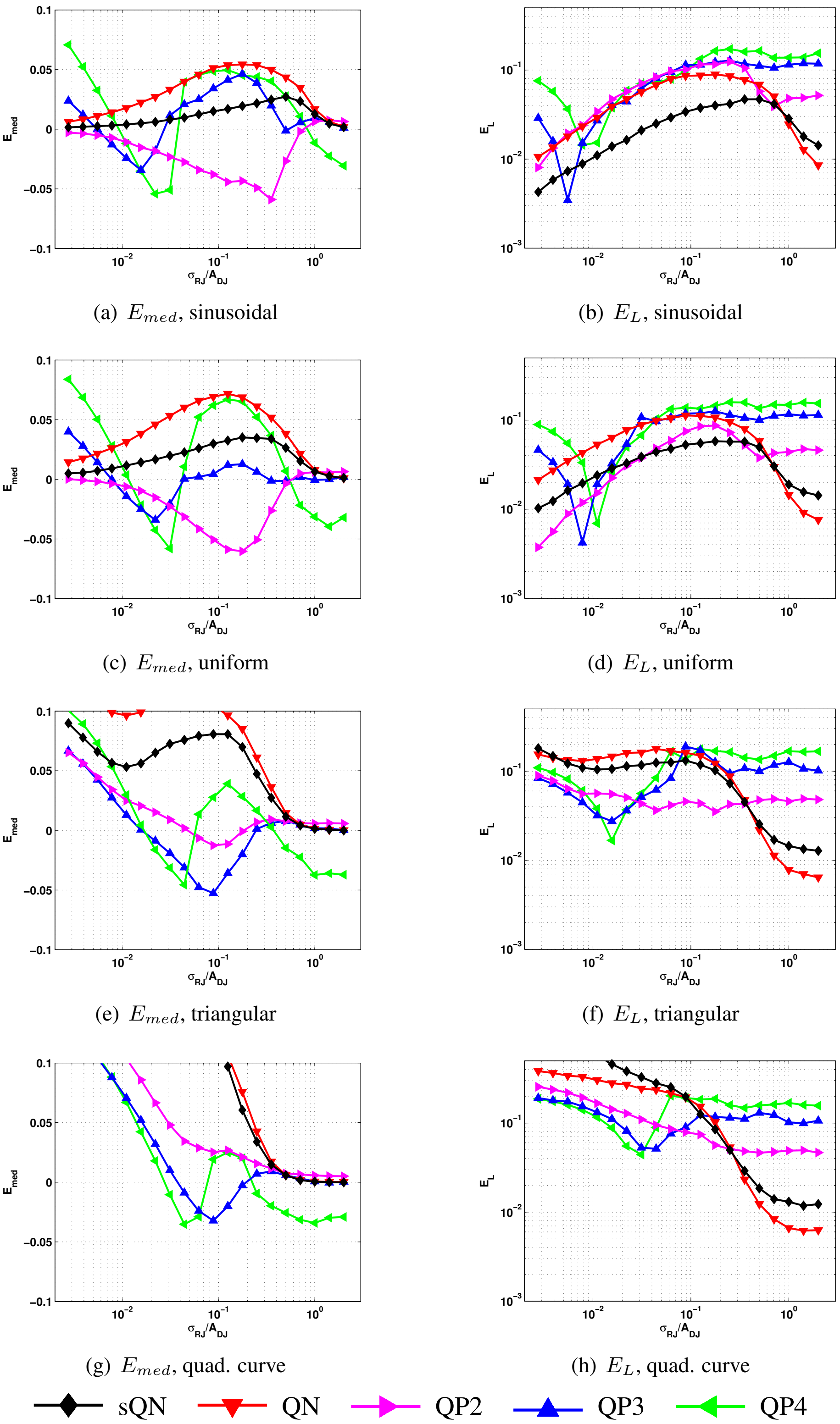
(e) E_{med}, triangular

(f) E_L, triangular

(g) E_{med}, quad. curve

(h) E_L, quad. curve

Figure 6.11.: N=10^4: E_{med} and E_L of compared fitting algorithms (sQN, QN, QP2, QP3, QP4) over varying test distribution shape σ_{RJ}/A_{DJ} and DJ type. K=250, R_{sim}=$3.3 \cdot 10^5$.



(a) E_{med}, sinusoidal

(b) E_L, sinusoidal

(c) E_{med}, uniform

(d) E_L, uniform

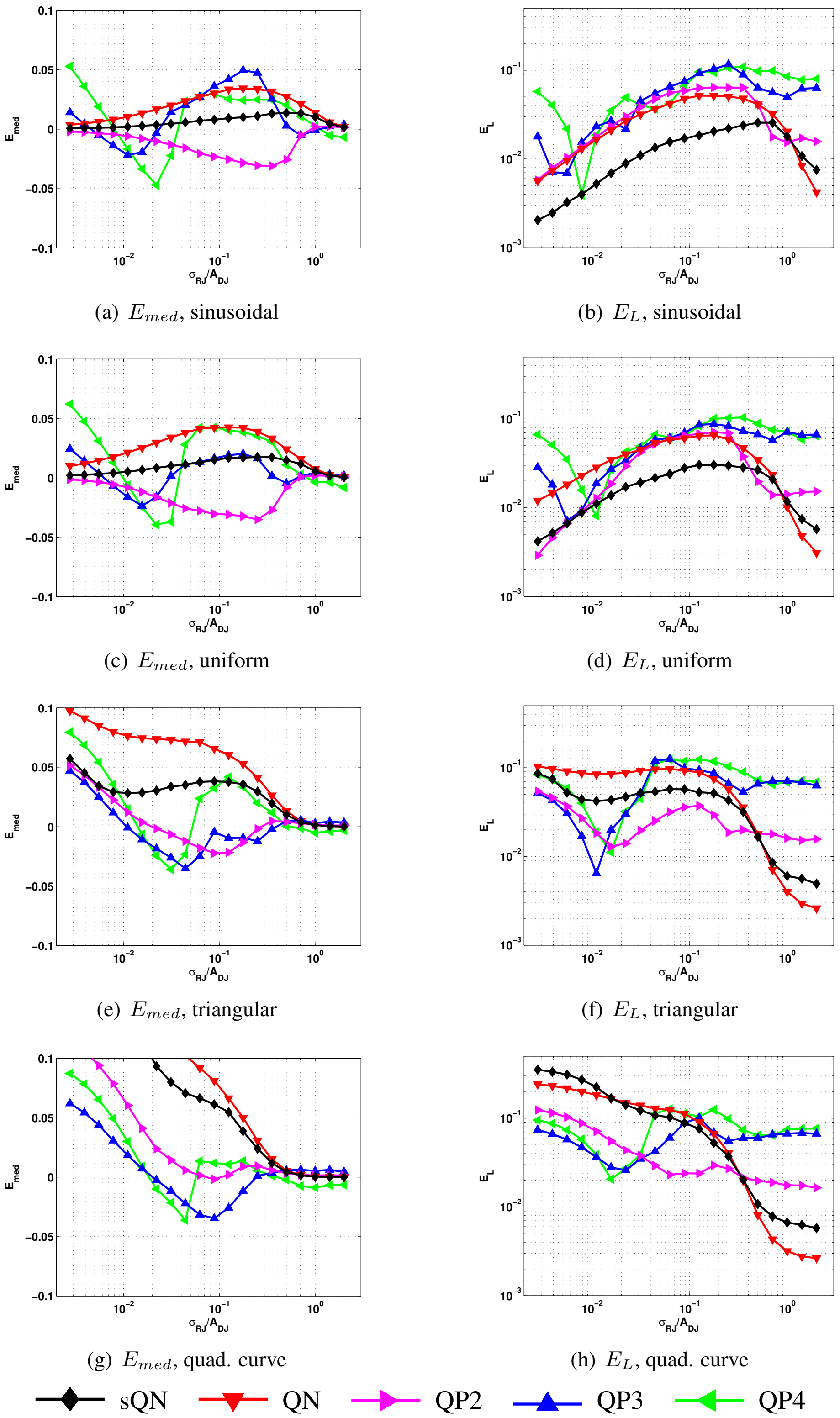
(e) E_{med}, triangular

(f) E_L, triangular

(g) E_{med}, quad. curve

(h) E_L, quad. curve

Figure 6.12.: N=10^5: E_{med} and E_L of compared fitting algorithms (sQN, QN, QP2, QP3, QP4) over varying test distribution shape σ_{RJ}/A_{DJ} and DJ type. K=250, R_{sim}=$3.3 \cdot 10^5$.



(a) E_{med}, sinusoidal

(b) E_L, sinusoidal

(c) E_{med}, uniform

(d) E_L, uniform

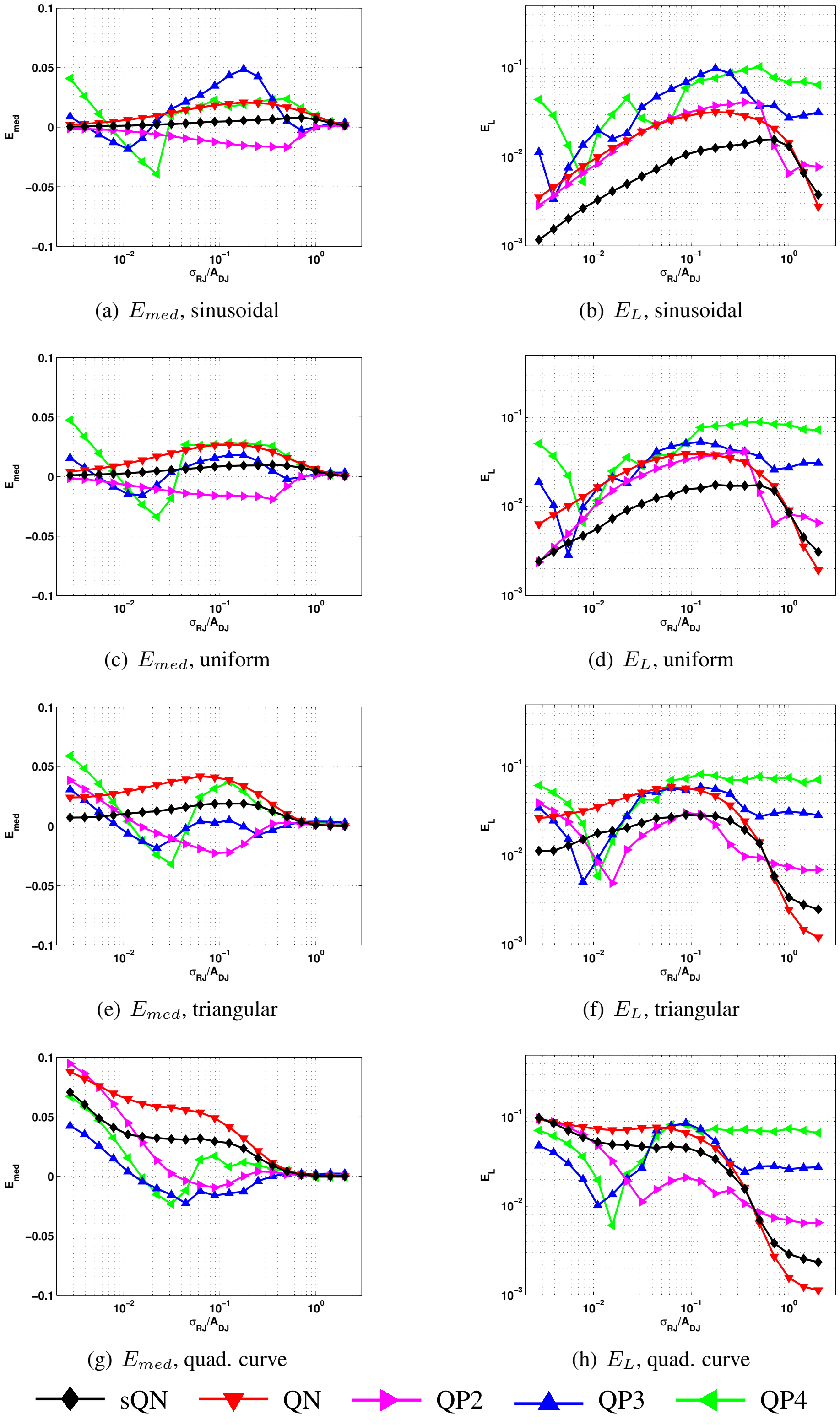
(e) E_{med}, triangular

(f) E_L, triangular

(g) E_{med}, quad. curve

(h) E_L, quad. curve

Figure 6.13.: N=10^6: E_{med} and E_L of compared fitting algorithms (sQN, QN, QP2, QP3, QP4) over varying test distribution shape σ_{RJ}/A_{DJ} and DJ type. K=250, R_{sim}=$3.3 \cdot 10^5$.



(a) E_{med}, sinusoidal

(b) E_L, sinusoidal

(c) E_{med}, uniform

(d) E_L, uniform

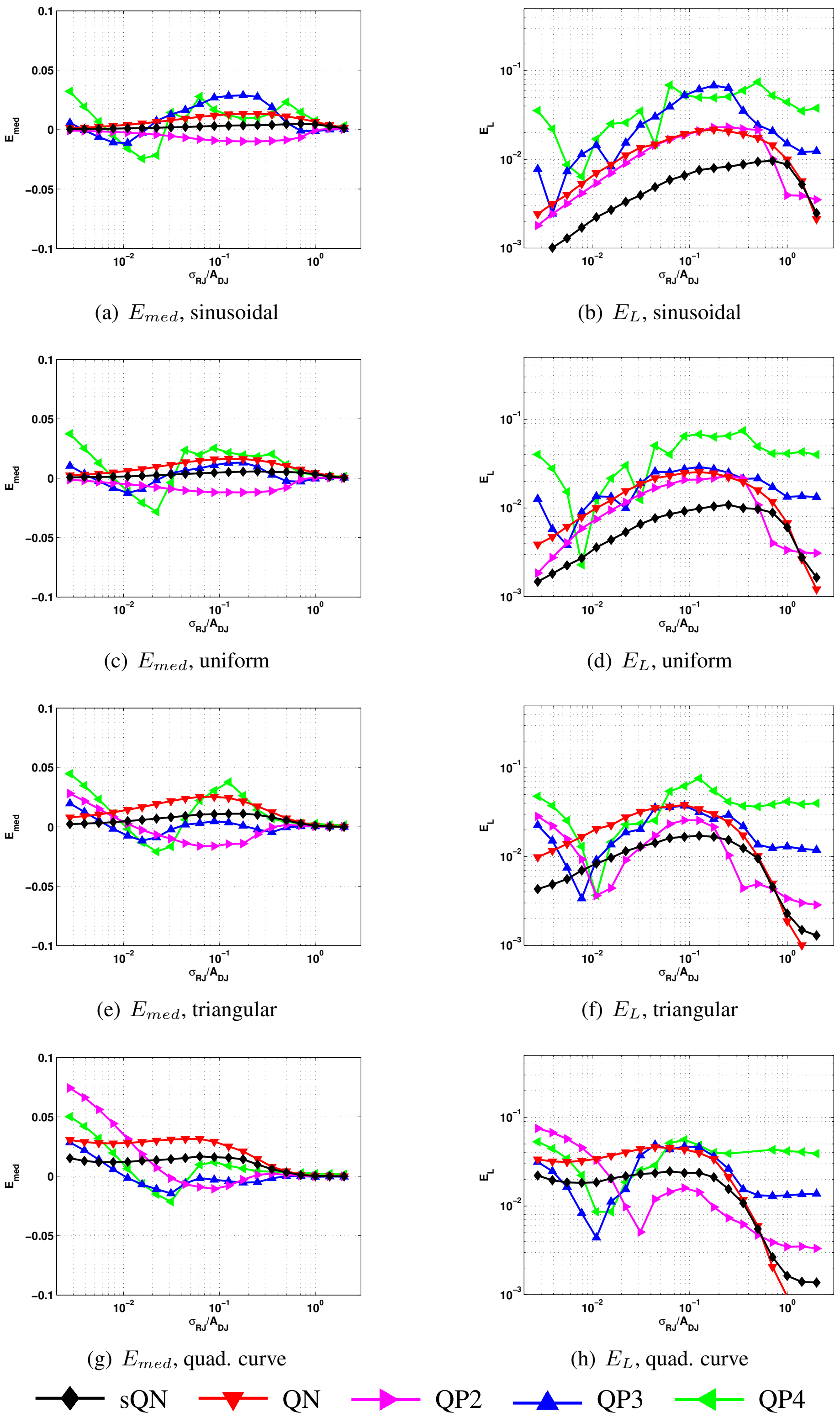
(e) E_{med}, triangular

(f) E_L, triangular

(g) E_{med}, quad. curve

(h) E_L, quad. curve

Figure 6.14.: N=10^7: E_{med} and E_L of compared fitting algorithms (sQN, QN, QP2, QP3, QP4) over varying test distribution shape σ_{RJ}/A_{DJ} and DJ type. K=250, R_{sim}=$3.3 \cdot 10^5$.



(a) E_{med}, sinusoidal

(b) E_L, sinusoidal

(c) E_{med}, uniform

(d) E_L, uniform

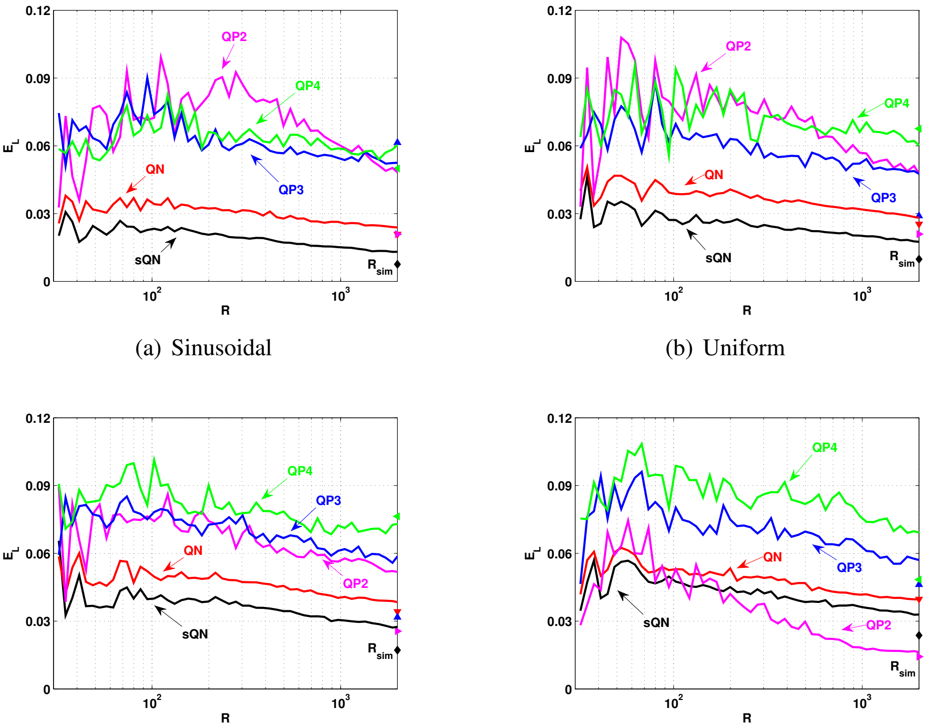
(e) E_{med}, triangular

(f) E_L, triangular

(g) E_{med}, quad. curve

(h) E_L, quad. curve

Figure 6.15.: N=10^8: E_{med} and E_L of compared fitting algorithms (sQN, QN, QP2, QP3, QP4) over varying test distribution shape σ_{RJ}/A_{DJ} and DJ type. K=250, R_{sim}=$3.3 \cdot 10^5$.



(a) Sinusoidal

(b) Uniform

(c) Triangular

(d) Quadratic curve

Figure 6.16.: Estimation loss E_L of sQN, QN, QP2, QP3, QP4 fitting algorithms over varying number of bins in the interval $R{=}[32, 2048]$. The figures evaluate the different fitting algorithms with sinusoidal, uniform, triangular and quadratic curve DJ types. $N{=}10^8$, $K{=}250$, $\sigma_{RJ}/A_{DJ}{=}1/8$, $A_{DJ}{=}0.2\,\text{UI}$.

indicates an over-fitting problem of higher order polynomials, caused by the statistical random variation of tails.

Influence of Time Resolution

The number of bins R has so far not been considered for the comparison of algorithms. In a simulation environment this value can always be chosen such that it does not affect the performance of a fitting method. In real hardware measurement systems, as already demonstrated in chapter 5, R is a critical design parameter and will always be limited. Thus, one likes to investigate its influence on the compared algorithms, in order to identify appropriate candidates for further in-depth analysis.

In figure 6.16 the estimation loss E_L is depicted as a function of time quantization in a typical range of $R{=}[32, 2048]$. Test distributions are created using a constant jitter ratio $\sigma_{RJ}/A_{DJ}{=}1/8$, to guarantee for a sufficient amount of extrapolation error with all methods. As can be seen from the resulting course of E_L, the sQN method achieves best performance with all four DJ types.

With decreasing time resolution, the obtained curves show an increased noise behavior, which is basically caused by the coarse discretization of bathtub tails. In section 5.4.2 this effect was already described as error ripple. Higher order polynomials generally suffer from large oscillations which are additionally intensified with increasing polynomial order. Further, the performance advantage of second order polynomials together with quadratic curve DJ in figure 6.16(d) is lost if the time resolution falls below $R{<}200$.

$\mathbf{DJ}$, $\sigma_{\mathbf{RJ}}/\mathbf{A_{DJ}}$		$\mathbf{N} = [\mathbf{5} \cdot \mathbf{10^5}, \mathbf{10^8}]$ $\sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$				$\mathbf{N} = [\mathbf{10^4}, \mathbf{10^6}]$ $\sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$			
		$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{r^2}$	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{r^2}$
Sinusoidal $1/4$	E_{med}	0.368	0.062	0.167	0.986	0.484	0.047	0.189	0.973
	IQR	0.389	0.153	0.173	0.947	0.843	0.132	0.251	0.967
	E_L	0.940	0.111	0.170	0.979	1.573	0.090	0.220	0.980
Uniform $1/8$	E_{med}	0.356	0.042	0.155	0.983	0.530	0.025	0.180	0.985
	IQR	0.510	0.143	0.191	0.957	1.213	0.137	0.282	0.974
	E_L	1.030	0.091	0.172	0.982	1.827	0.074	0.225	0.988
Triangular $1/16$	E_{med}	0.382	0.005	0.138	0.984	0.707	-0.027	0.166	0.989
	IQR	1.187	0.173	0.248	0.973	2.319	0.155	0.346	0.975
	E_L	1.248	0.063	0.176	0.990	2.047	0.024	0.218	0.988
Quadratic $1/16$	E_{med}	0.436	-0.003	0.129	0.987	0.762	-0.022	0.152	0.982
	IQR	2.355	0.177	0.283	0.977	2.150	0.144	0.331	0.976
	E_L	1.598	0.052	0.178	0.988	1.999	0.021	0.199	0.986
Only RJ	E_{med}	0.113	0.463	0.145	0.950	0.721	0.456	0.292	0.950
	IQR	1.753	0.405	0.262	0.964	5.371	0.350	0.388	0.971
	E_L	2.357	0.417	0.240	0.971	8.431	0.368	0.371	0.973

Table 6.2.: E_{med}, IQR and E_L coefficients for QN method, equation (6.6). $\Delta P_t/N{=}10^{-3}$ (large N) and $\Delta P_t/N{=}10^{-2}$ (small N), intervals specified above, $K{=}250$.

Therefore, only sQN and QN fitting methods are suitable for use with hardware jitter measurements. In fact, higher order polynomials introduce too much error and can generally not guarantee for pessimistic TJ estimation. With its fast algorithm, the QN method forms an alternative to the computationally more expensive sQN method and thus, the error behavior is thoroughly analyzed in the following section.

6.5. Estimation Error Analysis of Conventional Q-Normalization

In section 5.4 empirical relations were derived to describe the extrapolation error of the sQN method. Since the QN method forms a suitable alternative, the same empirical analysis is now also carried out for the QN method. Therefore equation (5.30) is reused, which describes the performance metrics E_{med}, IQR and E_L in terms of three regression coefficients a_0, a_1 and a_2:

$$\{E_{med}, IQR, E_L\} = a_0 \cdot (\sigma_{RJ}{\cdot}R)^{-a_1} \cdot N^{-a_2} \tag{6.6}$$

These coefficients are determined for each of the investigated test distributions and are subsequently listed in table 6.2. Estimation performance is investigated with respect to varying sample size N and bin density $\sigma_{RJ}{\cdot}R$. The intervals have been selected from original equidistant grids of 75×51 nodes with $\sigma_{RJ}{\cdot}R{=}\{0.8,\ldots,51.2\}$ UI as well as $N{=}\{10^5,\ldots,10^8\}$ and $N{=}\{10^4,\ldots,10^6\}$ respectively. Note, that the worst case shapes σ_{RJ}/A_{DJ} for conventional QN have been re-specified using figures 6.7-6.10, as they differ from the respective sQN values. A constant ratio $\Delta P_t/N{=}10^{-3}$ for large N as well as $\Delta P_t/N{=}10^{-2}$ for small N again guarantee for correct convergence of the algorithm. The r-squared statistic in the last column of the table also highlights an excellent correlation.

The DNL model coefficients can be recalculated for the QN method as well, where a suitable

DJ, $\sigma_{\mathbf{RJ}}/\mathbf{A_{DJ}}$		$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{r^2}$
		$\mathbf{N} = [\mathbf{5 \cdot 10^5}, \mathbf{10^8}], \quad \sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$					
Sinusoidal $1/2$	E_{med}	1.403	0.144	0.054	−1.514	0.558	0.961
	IQR	1.550	0.140	0.139	−6.880	1.383	0.941
	E_L	0.619	0.140	0.096	−5.091	1.186	0.958
Uniform $1/4$	E_{med}	1.170	0.146	0.045	−1.416	0.462	0.973
	IQR	1.392	0.151	0.122	−6.419	1.394	0.935
	E_L	0.429	0.146	0.077	−4.453	1.086	0.956
Triangular $1/16$	E_{med}	0.825	0.146	0.002	−0.179	0.026	0.983
	IQR	0.794	0.194	0.149	−6.020	1.233	0.932
	E_L	0.069	0.162	0.043	−3.099	0.728	0.967
Quadratic $1/16$	E_{med}	0.768	0.134	−0.005	−0.221	0.057	0.986
	IQR	0.258	0.218	0.172	−5.489	1.108	0.937
	E_L	−0.126	0.161	0.042	−2.610	0.631	0.972
Only RJ	E_{med}	1.416	0.189	0.521	−8.926	1.936	0.911
	IQR	1.071	0.156	0.421	−7.517	0.053	0.955
	E_L	0.375	0.158	0.442	−7.462	0.198	0.964
		$\mathbf{N} = [\mathbf{10^4}, \mathbf{10^8}], \quad \sigma_{\mathbf{RJ}} \cdot \mathbf{R} = [\mathbf{2.0}, \mathbf{51.2}]$					
Sinusoidal $1/2$	E_{med}	0.701	0.192	0.043	−0.432	0.083	0.977
	IQR	0.797	0.202	0.123	−4.402	1.075	0.948
	E_L	−0.130	0.196	0.081	−2.771	0.685	0.970
Uniform $1/4$	E_{med}	0.707	0.174	0.021	−0.499	0.082	0.984
	IQR	0.423	0.237	0.114	−3.982	0.945	0.936
	E_L	−0.272	0.201	0.058	−2.356	0.568	0.970
Triangular $1/16$	E_{med}	0.323	0.167	−0.023	−0.407	0.116	0.979
	IQR	−0.439	0.319	0.138	−3.734	0.883	0.946
	E_L	−0.614	0.211	0.017	−1.616	0.426	0.976
Quadratic $1/16$	E_{med}	0.249	0.156	−0.029	−0.409	0.120	0.965
	IQR	−0.267	0.299	0.128	−3.535	0.824	0.946
	E_L	−0.566	0.194	0.007	−1.455	0.386	0.963
Only RJ	E_{med}	0.760	0.241	0.491	−3.274	−0.300	0.952
	IQR	−0.516	0.281	0.380	−4.168	0.102	0.949
	E_L	−1.094	0.274	0.400	−4.003	0.032	0.961

Table 6.3.: E_{med}, IQR and E_L coefficients for QN method with included DNL error, equation (5.38). $\Delta P_t/N{=}10^{-3}$ (large N) and $\Delta P_t/N{=}10^{-2}$ (small N), $K{=}200$.

model description has already been given in equation (5.38):

$$\begin{aligned} &y = -a_0 - a_1 x_1 - a_2 x_2 - a_3 x_3 - a_4 x_2 x_3 \\ &x_1 = \ln(N),\ x_2 = \ln(\sigma_{RJ} \cdot R),\ x_3 = \ln(1 + \sigma_{DNL}) \\ &\{E_{med}, IQR, E_L\} = e^y \end{aligned} \tag{6.7}$$

Corresponding QN coefficients are listed in table 6.3. In order to deal with the large computational demand, a reduced grid resolution for bin density $\sigma_{RJ}{\cdot}R$ and sample size N is utilized with $25{\times}25$ nodes, and the same parameter intervals as with previous error analysis. Each plane is simulated with respect to varying DNL error in the range $\sigma_{DNL}{=}\{0.0, 0.01, \ldots, 0.19\}$. The r-squared statistic shows a slightly degraded performance of E_{med} hyperplanes for the pure Gaussian RJ case. The additional DNL error in fact strongly affects random tail variations, which also leads to a larger noise of fitted hyperplanes.

The obtained empirical relations can now be used to design a jitter measurement system which uses the conventional QN method for tail fitting, instead of the sQN method from chapter 5. Therefore, the design equations can simply be reused, as they are also valid for QN. Equation (5.3) and condition (5.10) guarantee for correct tail fitting if a minimum amplitude $A_{t,min}$ is given. Further, equation (5.20) and the selection chart in figure 5.9 specify a minimum RJ standard deviation $\sigma_{t,min}$ with known R, N and ΔP_t. Thus, QN and sQN errors may also be compared by simply extending the design examples from section 5.5.

Example for Jitter Diagnostics The estimation performance resulting from R=128, N=10^7 and $\sigma_{RJ,min}$=$7.57{\cdot}10^{-3}$, when inserted into equation (6.6) yields worst case errors (uniform DJ type):

$$E_{med} = 2.0\%,\ IQR = 2.3\%,\ E_L = 5.4\% \quad \text{(sQN)}$$
$$E_{med} = 2.9\%,\ IQR = 2.4\%,\ E_L = 6.4\% \quad \text{(QN)}$$

For the given $\sigma_{RJ,min}$, the resulting coarse bin density is the major contributor to overall error and thus, a rather small performance difference between QN and sQN is observed. This difference becomes even smaller if additionally a DNL error of σ_{DNL}=0.05 UI is assumed and empirical equation (6.7) is applied:

$$E_{med} = 2.3\%,\ IQR = 3.4\%,\ E_L = 7.4\% \quad \text{(sQN)}$$
$$E_{med} = 3.2\%,\ IQR = 3.0\%,\ E_L = 7.7\% \quad \text{(QN)}$$

which indicates that the statistical spread of the sQN algorithm is generally more affected by DNL error than the QN method. However, this is also an effect caused by low bin densities, since DNL has been modeled as a random step with time resolution $1/R$ (section 5.4.3) and can thus, also be reduced with a higher time resolution.

Example for Production Tests The presented BIJM system can now also be used together with the QN method, which only changes the empirical coefficients of equation (6.6) to the ones specified in table 6.2. The results are depicted in figure 6.17. For pure Gaussian distributions the error difference is marginal, because both QN and sQN methods behave similar. For sinusoidal DJ, a performance difference is instead clearly visible.

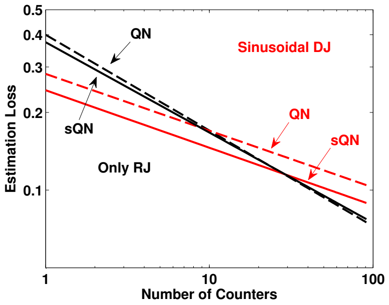


Figure 6.17.: Estimation loss E_L of sQN and QN methods over varying number of counters C, with t_t=20 ms, and worst case sinusoidal DJ (red) or pure RJ (black) case.

6.6. Summary

The performance of various recently proposed, polynomial tail fitting principles based on Gaussian quantile normalization was compared. First a unifying optimization scheme (figure 6.1) equivalent to the one for the sQN method was developed, in order to realize all of the investigated polynomial methods (QN, QP2, QP3, QP4). The required polynomial regression led to a Hankel matrix notation which can be inverted very efficiently by using a Levinson-Durbin recursion from [112]. As a fundamental simplification compared to the sQN method, only the regression error from equation (6.4) was shown to be applicable as fitness measure for tail fitting. Optimum parameter regions for tail selection with ΔP_t were discussed for each of the polynomial orders. This led to the selected parameter configurations in table 6.1.

Performance evaluations were first carried out for simulator environments (R_{sim}=$3.3{\cdot}10^5$) with respect to varying sample size N as well as DJ type (figures 6.7-6.10) for each of the polynomial methods. The median error of QN with linear functions was shown to be strictly positive, which is due to the asymptote in Q-domain, equivalent to the sQN method in section 4.2.1. Extrapolated tails are thus always pessimistic. This beneficial property was generally not maintained with higher order polynomials (QP2, QP3, QP4). Third and fourth order polynomials only produced results with acceptable accuracy for a small range of jitter ratios. For RJ dominant shapes, they were generally not able to correctly extrapolate tails, although forming simple linear functions in Q-domain. This effect was shown to be a general over-fitting problem of higher order polynomials.

A comprehensive performance comparison with the sQN method based on $\hat{c}_{1.2}$ with ΔP_t was carried out in figures 6.11-6.15. The sQN method clearly achieved best performance for the cost of a larger computational demand as already shown previously with figure 4.9. The QN method, although less accurate, offers approximately 35 times faster fitting results. Further, it was shown to possess the same positive error property as already described before and is thus, also well suited for tail fitting.

An interesting effect was observed with second order polynomials for $N{\geq}10^6$. The QP2 method returned excellent results down to $\sigma_{RJ}/A_{DJ}{\approx}2{\cdot}10^{-2}$ with triangular or quadratic curve shaped DJ, where it even outperformed the sQN method. This is due to an optimum working region. Nevertheless, the pessimistic extrapolation property of QN and sQN methods is not guaranteed with the QP2 algorithm. Additionally, polynomials do not allow for recovering Gaussian model parameters.

The influence of a limited time resolution with R bins on the extrapolation error was investigated in figure 6.16. It again highlighted the effect of error oscillations at coarse resolutions, which was also investigated in section 5.4.2. Results showed, that higher order polynomials are generally more affected by such oscillations than sQN and QN methods. Further, the performance advantage of the QP2 method with quadratic curve DJ was lost as soon as $R{<}200$.

Summarizing these results, only the sQN and QN fitting methods are well suited for use with hardware jitter measurements. Higher order polynomials introduce too much error and can generally not guarantee for pessimistic tail extrapolations. Therefore, an error analysis for the QN method was carried out in section 6.5, equivalent to the sQN error analysis in section 5.4. This allowed to compare the extrapolation performance of both methods and thus, to choose the better suited algorithm for a BIJM design. Results with the continued design examples from section 5.5 basically showed, that QN is generally less accurate than sQN, but is also less affected by differential non-linearity errors. Further, for pure Gaussian test distributions the performance difference between QN and sQN methods becomes marginal.

Note that for a system design with the conventional QN method, all the important design equations from chapter 5 can be reused. This especially includes equation (5.3) and condition (5.10) to guarantee for correct tail fitting results with respect to a minimum tail amplitude $A_{t,min}$, as well as equation (5.20) and the selection chart in figure 5.9 to specify a minimum RJ standard deviation $\sigma_{t,min}$. Parts of this chapter were also published in [C3,C7].

7. Jitter Analysis Method for Generalized Gaussian Tail Extrapolation

In this chapter the Gaussian quantile normalization is brought into a generic analysis context. Section 4.1.2 briefly mentioned the idea of generalizing the developed optimization scheme for use with arbitrary tail shapes. As a possible application, especially the amplitude distributions of high-speed signals may sometimes follow non-Gaussian shapes, including both heavy tailed distributions as well as fast decaying tails. In such cases one likes to identify the unknown shape in order to guarantee for correct signal recovery. As an extension to the pure Gaussian Q-normalization thus, classes of tail distributions may be described with additional shape parameters. The generalized principle then allows for correct tail extrapolation and jitter estimation of arbitrary tails as long as they belong to the defined class.

Generalizing the scaled Q-normalization (sQN) method from chapter 4 with additional shape parameters allows the linearizing principle to be extended to a complete class of tail distributions, which includes the Gaussian function as a special case. Here, special attention is given to the generalized Gaussian distribution (GGD) class of probability functions. It provides only one additional shape parameter, and the goal is thus, to realize accurate tail extrapolations for GGDs. The unknown shape parameter increases computational demand significantly, as it introduces an additional degree of freedom to the optimization search space. The GGD class is also expected to be less accurate than the sQN method for the special case of Gaussian tails. This is a natural generalization effect when unknown tail characteristics and thus, less information is given to the analysis system. On the other hand, the generalized method offers an outstanding opportunity, it can identify underlying tail characteristics, and is thus able to tell whether a tail truly behaves Gaussian-like or not.

In the following sections first the generalization principle is justified with a short literature review, and criteria are described to ensure consistency with the existing RJ-DJ model as introduced in section 2.3.1. Then various classes of generalized distributions and the corresponding quantile normalization functions are presented. The generalized Gaussian distribution (GGD) is described and its advantages compared to other function classes, before embedding it into the generalized optimization scheme for tail fitting. This scheme is implemented as efficient C++ routine. Performance evaluations are carried out and compared against the sQN method and other fitting principles. The chapter concludes with a brief summary.

7.1. Introduction to Generalized Tail Fitting

So far, the unbounded random part of a distribution tail has been assumed to be strictly Gaussian, which is also the case for many practical situations. However, literature also indicates scenarios where non-Gaussian tails must be handled [2, 37, 40, 59, 93, 126, 135], and thus the Gaussian assumption does not hold anymore. Such scenarios especially appear in optical high-speed communications where signals suffer from severe distortions, introduced by intersymbol interference (ISI) and noise. Here, signal integrity is rather affected by the amplitude noise instead of timing jitter. As shown in figure 7.1 bit errors can result from both timing jitter and amplitude noise [82], where in optical links especially the latter is the dominant cause to erroneous signal recovery [59, 126].

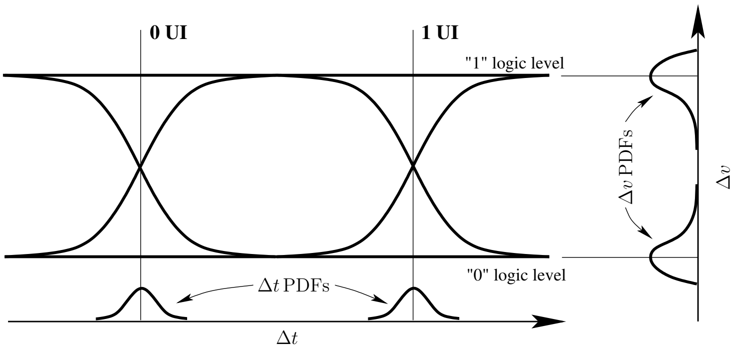


Figure 7.1.: Eye diagram with timing jitter and amplitude noise PDFs [82].

Typical amplitude histograms in optical fiber channels follow a chi-squared distribution where the critical tails decay very slowly and thus, cause bit errors at the receiver side. In order to minimize the BER, approaches based on maximum likelihood estimators and Viterbi-decoders are utilized [2, 59, 126]. These approaches deal with non-linear noise properties of the transmission medium, and require a careful analysis of observed amplitude distributions as well as the use of generalized tail fitting methods.

Weinstein [135] was the first to propose a method for approximating the generalized Gaussian distribution (GGD) class of functions. It extrapolates tails and estimates the BER without knowledge of the underlying tail shape. Stojanovic [126] described an extrapolation method, which also uses a subset of the GGD class. He also notes, that such tails especially appear in long haul optical fibers and channels that suffer from severe signal distortions.

For short-range wireline communications, linear additive noise sources prevail. Thus, with a large number of random processes involved, amplitude noise and timing jitter PDFs mostly follow a Gaussian tail. Nevertheless, in [75, 104] different jitter types are classified and the RJ section also defines an arbitrary non-Gaussian case which cannot be decomposed by conventional fitting methods. However, non-Gaussian timing jitter has so far only been described for soliton transmission, as a result of the Gordon-Haus effect [37, 44].

When focusing on the analysis of non-Gaussian tails thus, amplitude noise clearly dominates the practical use case. Fortunately, with the definition of the unit interval (UI), fitting methods are not restricted to the analysis of timing jitter and can equivalently also be applied to amplitude histograms. In this context, an accurate jitter decomposition method for arbitrary RJ shapes is subsequently developed. It is consistent with the existing RJ-DJ model [104] and hence, forms a logical extension to the commonly accepted modeling approach.

The method can easily be derived from the generalized sQN scheme in figure 4.6. Therefore, only a suitable quantile function for the generic normalization stage must be determined, as depicted in figure 7.2. $Q(p)$ forms the heart piece for tail linearization. It may normalize a specific tail shape, or a whole class of distributions when described by one or more shape parameters (α, β, ...). Note that every additional parameter introduces another degree of freedom for the opti-

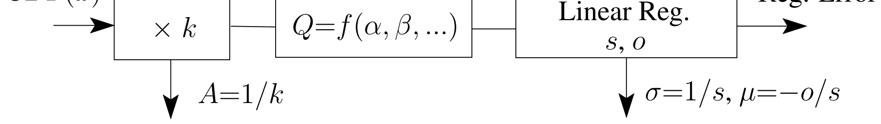


Figure 7.2.: Generalized optimization scheme.

Distribution	**PDF: f(x)**	**Quantile Normalization: Q(p)=F^{-1}(p)**
	$x \in (-\infty, +\infty)$	$p \in [0,1]$
Gaussian	$\frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$-\sqrt{2} \cdot \text{erfc}^{-1}(2p)$
Generalized Gaussian	$\frac{\alpha}{2\cdot\beta\cdot\Gamma(1/\alpha)} e^{-\left\vert\frac{x-\mu}{\beta}\right\vert^{\alpha}}$ $\beta = \sigma\sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}$ $\Gamma(u) = \int_0^\infty e^{-t} t^{u-1} dt$	$\gamma^{-1}\Big(\frac{1}{\alpha}, \vert 2p-1\vert\Big)^{1/\alpha} \cdot \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}} \cdot \text{sgn}(2p-1)$ $\gamma(u,x) = \frac{1}{\Gamma(u)} \int_0^x t^{u-1} e^{-t} dt$ sgn$(\ldots)$ ... sign function
	$x \in [0, \infty)$	$p \in [0,1]$
Exponential	$\frac{1}{\sigma} e^{-\frac{x}{\sigma}}$	$-\ln(1-p)$
Generalized Pareto	$\frac{1}{\sigma}\Big(1 + \alpha \cdot \frac{x-\mu}{\sigma}\Big)^{-1-\frac{1}{\alpha}}$	$\begin{cases} -\ln(1-p) & \text{if } \alpha = 0 \\ -1/\alpha\Big[1-(1-p)^{-\alpha}\Big] & \text{if } \alpha \neq 0 \end{cases}$
Generalized Extreme Value	$\frac{1}{\sigma}(1+\alpha z)^{-1-\frac{1}{\alpha}} \cdot e^{-(1+\alpha z)^{-\frac{1}{\alpha}}}$ $z = (x-\mu)/\sigma$	$\begin{cases} -\ln\big(-\ln(p)\big) & \text{if } \alpha = 0 \\ -1/\alpha\Big[1-\big(-\ln(p)\big)^{-\alpha}\Big] & \text{if } \alpha \neq 0 \end{cases}$

Table 7.1.: Quantile normalization functions for different tail distributions [106].

mization scheme and thus, significantly increases computational demand. Once the tail parameters have been identified, the timing budget TJ$_{pp}$ can easily be determined at the target BER$=10^{-12}$ with:

$$\text{TJ}_{pp} = t_L + t_R \tag{7.1a}$$

$$t_L = \mu_L + \sigma_L \cdot Q(10^{-12}/A_L, \alpha, \beta, \ldots) \tag{7.1b}$$

$$t_R = \mu_R + \sigma_R \cdot Q(10^{-12}/A_R, \alpha, \beta, \ldots) \tag{7.1c}$$

7.1.1. Quantile Normalization Functions

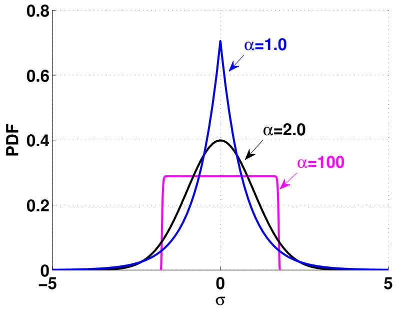
With equation (4.8) a general law has been given to describe the quantile function inside the optimization scheme. With a probability distribution $p{=}F(x)$, the quantile normalization $Q(p)$ is

$$q = Q(p) = F^{-1}(p,\ \mu{=}0, \sigma{=}1, A{=}1) \tag{7.2}$$

which corresponds to the inverse probability function for an expected tail shape of unit amplitude, unit standard deviation and zero mean.

The quantile normalization $Q(p)$ may utilize additional shape variables to include the tail characteristics of entire function classes. In table 7.1 $Q(p)$ is listed for various candidates. The Gaussian and generalized Gaussian functions are defined over the complete real axis. That is, they are symmetric for positive and negative values of x. Other distributions such as the exponential, generalized Pareto or generalized extreme value PDFs are bounded toward negative values of x. The PDF definitions use a positive range for x to denote the right sided normalization function, which is applied to positive bathtub tails with $p{=}F(x)$. If measured tails belong to the domain of attraction of the normalizing function, the optimization scheme from figure 7.2 is thus able to identify a best suited set of tail parameters.

The presented distribution classes give an outline to possible tail characteristics, which may exhibit power-law, exponential or Gaussian-like behavior instead of the pure Gaussian case. The



Shape α	Distribution
$\rightarrow 0$	Dirac impulse
0.5	Gamma
1.0	Laplace
2.0	Gaussian
$\rightarrow \infty$	Uniform

Figure 7.3.: Special GGD shapes.

correct choice for an expected tail shape is crucial as it highly influences extrapolation results. Generalized extreme value (GEV) distributions and generalized Pareto (GP) distributions are commonly used in extreme value theory, and their use for nonparametric tail extrapolation and threshold models in general has been discussed extensively in literature [19, 115, 117]. Nevertheless, here we focus on the generalized Gaussian distribution (GGD), which offers two major advantages compared to other generalizations. First, it represents a simple and direct generalization of the pure Gaussian function. With only one additional shape parameter α, it includes important special cases such as the Gaussian (α=2) or the exponential (α=1) tails. Second, it is fully consistent with the commonly accepted RJ-DJ model from [104], and thus allows for decomposing a total distribution into its non-Gaussian random and bounded deterministic components. In fact, the additional shape parameter α extends the existing model with the ability of tail shape characterization.

7.1.2. Generalized Gaussian Distribution

In this section, properties of the generalized Gaussian distribution (GGD) as a generic function for tail fitting and extrapolation are described. As the name already indicates, it puts the Gaussian distribution into a general context, where an additional shape parameter α defines the exponential rate of decay:

$$\mathrm{PDF}(x) = f(x, \alpha, \beta, \mu) = \frac{\alpha}{2 \cdot \beta \cdot \Gamma(1/\alpha)} e^{-|\frac{x-\mu}{\beta}|^{\alpha}} \tag{7.3a}$$

$$\beta = \sigma \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}, \quad \Gamma(u) = \int_0^{\infty} e^{-t} t^{u-1} dt \tag{7.3b}$$

where $[-\infty < x < \infty]$ and $\alpha > 0$. $\Gamma(u)$ is the gamma function, which is required for amplitude normalization so that α can be varied independently. Note that β is a dependent scale parameter, directly related with σ and α, and hence, it does not influence the tail shape.

The advantages of this representation form are, that only the α parameter defines the tail shape and that the GGD class covers several special function types. With α=2 the normal distribution is obtained, while for α=1 the Laplace distribution with exponential tails, and for α=0.5 the heavy tailed Gamma distribution. A small shape value yields an impulsive function with slowly decaying tails, while a large value leads toward the uniform distribution. This behavior is also depicted in figure 7.3. The α parameter thus offers a flexible way for representing a generic distribution class with exponential tail behavior.

The quantile normalization for GGD functions is derived as inverse of the CDF, which is obtained from the PDF integral:

$$p = \mathrm{CDF}(x) = F(x, \alpha, \sigma, \mu) = \frac{1}{2} \cdot \left[1 + \mathrm{sgn}(x - \mu) \cdot \gamma\left(\frac{1}{\alpha}, \left| \frac{x - \mu}{\beta} \right|^{\alpha} \right) \right] \tag{7.4a}$$

$$\gamma(u, x) = \frac{1}{\Gamma(u)} \int_0^x t^{u-1} e^{-t} dt \tag{7.4b}$$

where $\gamma(u, x)$ is the incomplete gamma function and $\mathrm{sgn}(\ldots)$ the sign function. The inverse CDF can be written as:

$$F^{-1}(p, \alpha, \sigma, \mu) = \gamma^{-1}\Big(1/\alpha, |2p - 1|\Big)^{1/\alpha} \cdot \beta \cdot \mathrm{sgn}(2p - 1) + \mu \tag{7.5}$$

For the quantile normalization $Q(p, \alpha){=}F^{-1}(p,\ \mu{=}0, \sigma{=}1)$ the case with unit standard deviation and zero mean must be considered, and thus:

$$Q(p, \alpha) = \gamma^{-1}\Big(1/\alpha, |2p - 1|\Big)^{1/\alpha} \cdot \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}} \cdot \mathrm{sgn}(2p - 1) \tag{7.6}$$

The resulting equation describes a transform which allows for the linearization of arbitrary GGD tails. When inserted into the optimization scheme from figure 7.2 we have four unknown variables: α, A, σ and μ. This four-dimensional search space has to be dealt by an efficient search algorithm, as described in the subsequent section.

Equivalent to the analysis of test distributions with Gaussian tails in section 3.2.2 GGD, test distributions must also be created in order to analyze and compare the estimation performance of tail fitting algorithms. Therefore, the composition principle with RJ and DJ can be reused by replacing the Gaussian RJ generator with a GGD jitter source. The additional shape parameter is denoted as α_{RJ}, and all the prior performance metrics such as estimation loss E_L or median error E_{med} can be reused.

An important issue relates to correct GGD jitter generation. Since the quantile normalization function describes a probability $p{\in}[0, 1]$ in terms of the amplitude of a normalized random variable, one can use equation (7.5) for random sample generation. Figure 7.4 demonstrates this principle, where a uniform random process J_{uni} generates jitter samples which are used as input to the GGD normalization function. The required standard deviation σ is obtained by scaling, while a non zero mean μ yields additional data offset.

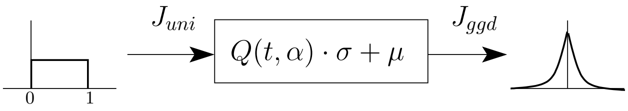


Figure 7.4.: GGD random generator.

7.2. Implementation of Algorithm

An efficient implementation of the quantile normalization for GGD shapes requires a fast realization of the inverse incomplete gamma function γ^{-1}. Equation (7.6) cannot be expressed as a closed form equation. This means, $Q(p, \alpha)$ has to be implemented as an iterative approximation, which requires considerable computational demand. Another problem relates to the large amount of jitter samples required for performance analyses. If test distributions shall include up to several million random samples, they must be generated very quickly. Therefore, this section especially focuses on speed optimization for the GGD jitter source and the optimization scheme in figure 7.2, which faces the four-dimensional search space.

In [26] the γ^{-1} function has been realized very efficiently with a third-order Schröder iteration, supported by the Newton-Raphson method. A target accuracy of $\epsilon_{rel}{=}10^{-6}$ is typically reached after the second or third iteration step. Also the included complete gamma (7.3b) and incomplete gamma (7.4b) functions are realized with minimax rational approximations and a uniform asymptotic expansion respectively. This yields an excellent computational efficiency.

The described functions have been realized as C++ routines, where one million γ^{-1} function calls require approximately three seconds of simulation time on an Intel Core Duo 2.2GHz laptop. This is still not fast enough, neither for tail parameter search with the scheme in figure 7.2, nor for the GGD random generator in figure 7.4. Especially when gathering millions of random samples, test distributions have to be generated significantly faster. Therefore, additional minimax approximations of $Q(p,\alpha)$ have been realized, which support certain discrete α values and achieve an additional speed up of more than one order of magnitude. As selectable grid values, they cover shapes in a range from $\alpha{=}[1,10]$ with logarithmically scaled distance and a maximum relative error of $\epsilon_{rel}{=}10^{-6}$. These minimax approximations allow for speeding up the optimization process with a fast initial search grid and thus, also achieve a quick generation of random values.

An efficient realization of the optimization scheme in figure 7.2 requires a fast identification of the global error minimum. The regression stage fits a simple linear function to the tail part of n outermost tail samples by reusing the recursions (4.5) of the sQN method from section 4.1.1. The regression is thus already optimized with respect to fast line slope and offset recovery. The regression error $\hat{\sigma}_{err}$ is the mean square error of fitted data pairs (q_i, x_i):

$$\hat{\sigma}_{err} = \sqrt{\frac{\sum_{i=1}^{n} \left(q_i - o - s \cdot x_i\right)^2}{n-2}} \tag{7.7}$$

The two remaining parameters scaling factor k and shape α obviously define the quantiles, here given as q_i values. Both parameters have to be identified in the context of an additional outer optimization, characterized by another two dimensional minimum search.

In section 4.3 the ratio $\hat{T}{=}\hat{\sigma}_{err}/s$ was also introduced as a fitness measure, suggested by Scholz [117] for judging the appropriateness of a fitted line. For GGD tail extrapolation, this criterion now drives the outer optimization. The optimization process is composed of two steps. An initial search grid identifies a coarse error minimum, which is then refined by a search routine.

The initial search grid locates the global minimum, where the tail length n is pushed in as far as possible. This corresponds to the same algorithmic principle as already described in section 4.3. For each of the (k,α) grid values the $\hat{T}(n)$ minimum is determined as a function of tail length n, and the grid value with maximum tail length is selected. The search grid is logarithmically scaled with a distance factor of $\Delta k{=}1.2$ for both k and α variables. The investigated intervals are $k{=}[10^1, 10^3]$ and $\alpha{=}[1, 10^1]$, while $\Delta P_t{\geq}10^2$ is used as minimum tail interval for outlier suppression. The resulting two-dimensional surface is thus a function of k and α, and typically forms a narrow valley where the global minimum is hardly distinguishable along the bottom course. A typical example is given in figure 7.5.

The second optimization step is a local refinement of the resulting plane, carried out with a bounded search algorithm. Starting with the initial grid pair (k_{in}, α_{in}), the bounds are:

$$\begin{aligned} &\alpha_{lo} \leq \alpha_{in} \leq \alpha_{up}, \qquad \alpha_{lo} = \max(1, \alpha_{in}/\Delta k), \qquad \alpha_{up} = \min(10, \alpha_{in} \cdot \Delta k) \\ &k_{lo} \leq k_{in} \leq k_{up}, \qquad k_{lo} = \max(1, k_{\alpha_{lo}}), \qquad k_{up} = \min(10^3, k_{\alpha_{up}}) \end{aligned} \tag{7.8}$$

where $k_{\alpha_{lo}}$ and $k_{\alpha_{up}}$ are determined from the search grid as k-minimums which appear at the α_{lo} and α_{up} values respectively. The refinement stage finally determines the minimum values

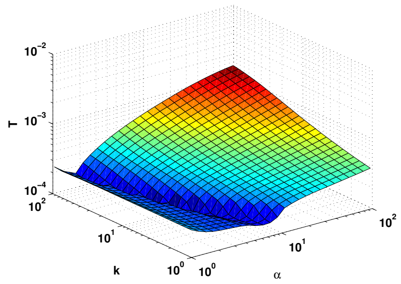


Figure 7.5.: $\hat{T}$ surface for two-dimensional minimum search with initial search grid.

(k_{opt}, α_{opt}) on the parameter surface. It uses the BOBYQA algorithm [113], which showed good performance and a fast convergence compared to other search algorithms from the NLopt [67] nonlinear optimization library. As convergence criteria a minimum relative parameter variation of $x_{tol} \leq 10^{-5}$ or a maximum function count of three hundred iterations are used.

The overall optimization converges sufficiently fast, especially due to the minimax function approximations of the initial search grid. The refinement step typically requires 30-40 function calls. As with the sQN method in section 4.1.4, computational demand again depends on the time resolution R, since the algorithm has to process all distribution bins. For R_{sim}=3.3·10^5 and the same test distribution as for the performance analysis in figure 4.9 (A_{DJ}=0.2 UI, σ_{RJ}=0.05 UI, α_{RJ}=2), the 2.2GHz laptop typically requires approximately one minute, which is acceptable for model simulations. In order to speed up the optimization in subsequent analyses, the simulator time resolution is reduced to some degree (R_{sim}=10^5) without significant influence on the error.

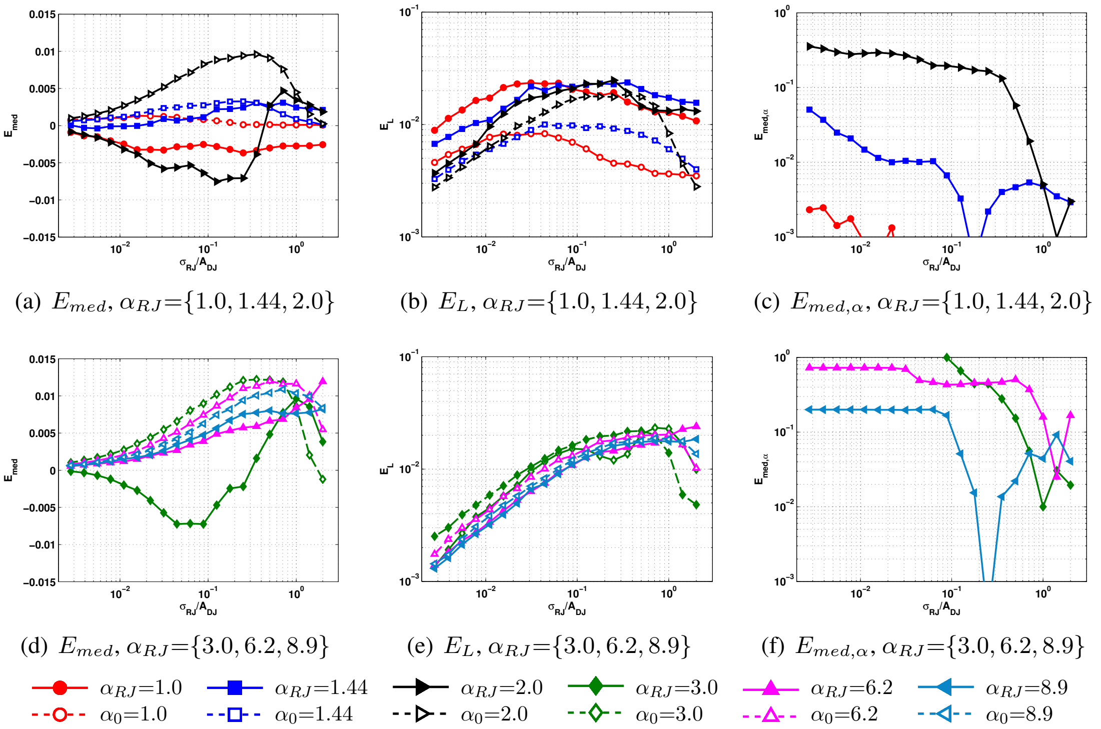
7.3. Performance Analysis

This section focuses on performance evaluations for the GGD fitting method. Therefore, the same analyses are carried out as for the sQN method. The GGD test distributions utilize the parameters defined in section 3.2.2 (A_{DJ}, σ_{RJ}), with the additional shape parameter α_{RJ} for the generalized RJ component. This allows for a simple performance comparison with other methods, while the GGD random source is realized in a consistent manner according to figure 7.4. Note, that the GGD fitting method is not restricted to the analysis of timing jitter. It can also be applied to amplitude histograms, which is also the major use case as was already discussed in the introductory section. The analysis in terms of the unit interval (UI) is thus only used for a consistent comparison of different algorithms.

The GGD performance evaluation starts by describing influences of the additional shape parameter on estimation error. Therefore both software modeling and hardware scenarios with coarse time resolution are considered. A brief performance comparison with different estimation principles and an existing generalized fitting method is also provided. This method was originally suggested by Weinstein [135], and is based on a simple double-log scale.

7.3.1. Software Model Simulations

Since system level simulations play an important role for transceiver designs, first the performance of the GGD method is investigated when used together with software models. In a simulator, the time resolution R_{sim} or number of bins per unit interval can be selected arbitrarily and thus, will not influence the accuracy of a tail fitting method. The same high-speed transceiver as for the sQN method is assumed, running at 3Gb/s and a slightly reduced simulator time resolution of 3.3 fs.



(a) E_{med}, α_{RJ}=$\{1.0, 1.44, 2.0\}$ (b) E_L, α_{RJ}=$\{1.0, 1.44, 2.0\}$ (c) $E_{med,\alpha}$, α_{RJ}=$\{1.0, 1.44, 2.0\}$

(d) E_{med}, α_{RJ}=$\{3.0, 6.2, 8.9\}$ (e) E_L, α_{RJ}=$\{3.0, 6.2, 8.9\}$ (f) $E_{med,\alpha}$, α_{RJ}=$\{3.0, 6.2, 8.9\}$

Figure 7.6.: Median error E_{med}, estimation loss E_L, and shape error $E_{med,\alpha}$ over varying test distribution shape: N=10^7, ΔP_t=10^2, uniform DJ, R_{sim}=10^5, K=250.

With R_{sim}=10^5 and a distribution sample size of N=10^7, tails can be fitted sufficiently accurate and correspond to an extrapolation which ranges over five orders of magnitude.

In figure 7.6 the estimation performance of the GGD tail fitting method is investigated with respect to varying jitter ratio σ_{RJ}/A_{DJ} and α_{RJ}. The solid lines with filled markers denote the proposed GGD method, while dashed lines show the results obtained when α_{RJ} is already known to the search algorithm. In this case the constant α_0=α_{RJ} replaces the shape variable and yields a reduced search space for the fitting algorithm. The special case of α_0=α_{RJ}=2 for example, gives the simplified form of Gaussian tail fitting with scaled Q-normalization from chapter 4.

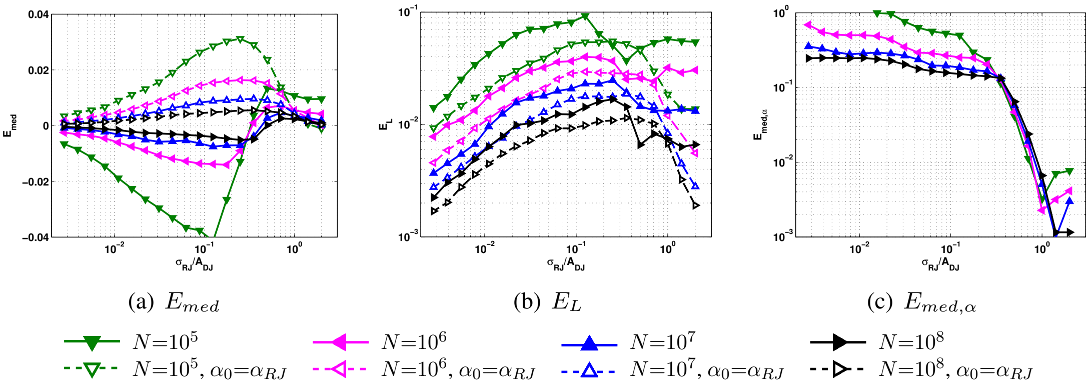
At α_{RJ}=1, the error bias E_{med} tends to be negative and thus, to slightly underestimate the TJ values. This is not the case for known α_0=α_{RJ}, which is again due to the asymptotic tail behavior in quantile domain. For larger α_{RJ} values, the median error of the GGD method also becomes positive, which is an effect caused by very steep tails.

Over all the investigated distribution shapes, E_L yields a worst case error which is less than 2.5%. If α_{RJ}>3, the estimation loss also shows that the generalized method can directly compete with the known α_0 scenario. Unfortunately, this is due to highly overestimated shape values, as can be seen in figures 7.6(c) and 7.6(f). Here, the median shape error $E_{med,\alpha}$ is calculated equivalent to E_{med}, using estimated shape values:

$$E_{\alpha,k} = \alpha_{est,RJ}/\alpha_{true,RJ} - 1, \quad k = 1,\ldots,K \tag{7.9}$$

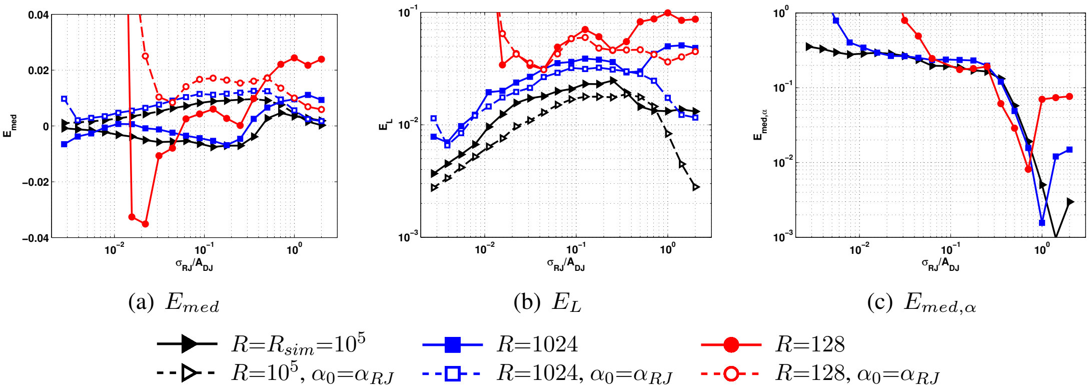
$$E_{med,\alpha} = \text{median}\{E_{\alpha,k}\} \tag{7.10}$$

Especially with steep tails, either realized by small jitter ratios σ_{RJ}/A_{DJ} or a large α_{RJ}, the algorithm rather overestimates the true shape instead of correctly fitting a small tail amplitude. Therefore, an exact tail shape identification of distributions can only be carried out in a very limited sense.



(a) E_{med} (b) E_L (c) $E_{med,\alpha}$

Figure 7.7.: E_{med}, E_L, $E_{med,\alpha}$ for various distribution shapes equivalent to figure 7.6, with constant shape parameter α_{RJ}=2.0 (Gaussian case) and varying sample size N=$\{10^7, 10^8, 10^9\}$, K=250.



(a) E_{med} (b) E_L (c) $E_{med,\alpha}$

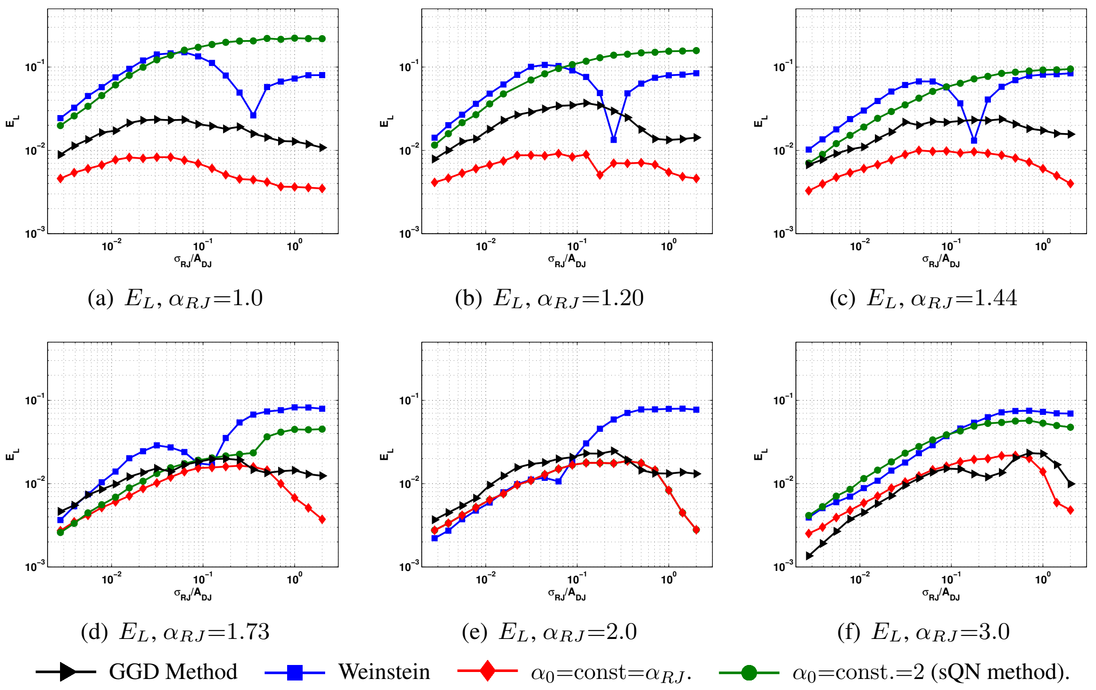
Figure 7.8.: E_{med}, E_L, $E_{med,\alpha}$ for various distribution shapes equivalent to figure 7.6, with constant shape parameter α_{RJ}=2.0 (Gaussian tails) and varying time resolution R=$\{10^5, 1024, 128\}$, N=10^7, K=250.

Acceptable results are obtained with $\alpha_{RJ}\leq 2$ and large jitter ratios $\sigma_{RJ}/A_{DJ}\geq 0.5$ (RJ dominant case). A possible way to extend this application range is to increase the sample size N, which has been kept comparable small so far. Therefore, in figure 7.7 the α_{RJ}=2.0 case is again investigated with respect to varying N. Although a significant improvement can be achieved for both E_{med} and E_L, α_{RJ} still remains highly overestimated. Also note, that an exponential increase of the sample size can soon lead to unacceptable simulation or measurement times.

7.3.2. Hardware Model Simulations

Hardware models for jitter measurement systems additionally have to consider a very limited time resolution $R \ll R_{sim}$ of collected jitter distributions. This limitation can highly affect the extrapolation error. Since the GGD fitting method already performed poor in terms of correctly identifying the tail shape α, a strong performance degradation for the hardware scenario is expected as well.

Figure 7.8 demonstrates the effect on estimation performance with two typical resolutions R=1024 and R=128. Besides an increased E_L, also a lower bound for the minimum jitter ratio is introduced by R, as can be seen at $\sigma_{RJ}/A_{DJ}\approx 3{\cdot}10^{-3}$ (R=1024) and $\approx 2{\cdot}10^{-2}$ (R=128). This is due to the insufficient number of bins located on a measured bathtub tail. Compared to



(a) E_L, α_{RJ}=1.0

(b) E_L, α_{RJ}=1.20

(c) E_L, α_{RJ}=1.44

(d) E_L, α_{RJ}=1.73

(e) E_L, α_{RJ}=2.0

(f) E_L, α_{RJ}=3.0

Figure 7.9.: Estimation loss E_L over varying jitter ratio σ_{RJ}/A_{DJ}. The test distributions are the same as in figure 7.6, with α_{RJ}=$\{1.0, 1.2, 1.44, 1.73.2.0, 3.0\}$, N=10^7, R_{sim}=10^5 and K=250.

the scenario with constant and known α_0 (dashed lines), the generalized fitting method performs significantly worse. Especially with R=128 a large peak toward negative E_{med} is noticed. This behavior is highly undesired as it leads to optimistic TJ estimates. The error peak ranges from a large jump of shape estimates in figure 7.8(c), which also evidences the critical jitter ratio limit where the GGD method starts to fail.

7.3.3. Comparison with Other Methods

The developed GGD fitting method is briefly compared with special scenarios including the known α_0 case, the pure Gaussian tail assumption, as well as Weinstein's method [135] for GGD tail approximation. This method simply transforms measured tails into a double-log scale, where regression lines can be used for extrapolation. Weinstein showed with the asymptotic expansion of the probability function (equation (7.3a)), that the double-log scale is asymptotically linear for GGD tails. The resulting method is thus simple and very fast.

In figure 7.9 the performance of the developed GGD method is compared with I) Weinstein's method, II) known α_0=α_{RJ} shape parameter and III) Gaussian tail assumption with α_0=2 (sQN method). This last scenario demonstrates how sensitive the Gaussian fitting method is with respect to varying tail shapes, and indicates the importance of either correct α_0 choice, or application of the generalized GGD method with unknown tail characteristics.

The scenario with known shape α_0=α_{RJ} achieves best performance over all three shape values α_{RJ}=$\{1.0, 2.0, 3.0\}$. As an additional advantage, it tends to overestimate the true TJ values and thus, always guarantees for a pessimistic tail extrapolation. The GGD tail fitting method nearly achieves the same accuracy, but generally underestimates the true value and thus yields slightly optimistic TJ values.

Weinstein's method generally overestimates the true TJ_{pp} value at large jitter ratios σ_{RJ}/A_{DJ}

and underestimates it at small ones. This makes the method only applicable down to a certain limit. The error E_L is quite large due to the approximation of GGDs in double-log domain, but the method is especially simple and thus, very fast.

The performance comparison with the sQN method highlights a possible application of the generalized method, which is for tail shapes in the interval range α_{RJ}=$[1, 2]$. Here, the sQN method becomes highly optimistic with negative error bias, because it always assumes Gaussian tail behavior. The test distributions instead are heavy tailed and thus, follow a flat course which cannot be tracked correctly by sQN. If σ_{RJ}/A_{DJ} is sufficiently large and the TJ error must be strictly positive, Weinstein's method is better suited, because it always guarantees for pessimistic tail extrapolations.

7.4. Summary

A generic optimization scheme for non-Gaussian tail fitting was presented, which led to a jitter analysis method for generalized Gaussian distributions (GGDs). Therefore, the Gaussian quantile normalization principle from chapter 4 was reused. The method is fully consistent with the existing RJ-DJ model and utilizes only one additional shape parameter to describe different tail characteristics.

In simulators the proposed method correctly fits and extrapolates tails that belong to the GGD class of functions, although clearly outperformed by the case of known shape parameter α. The major drawback of the generalized method results from the optimistic extrapolation property. This is due to a general overestimation of shape values, especially at α=2 and α=3, as shown in figure 7.6. This problem can only be dealt by a significant increase of the sample size N (figure 7.7). In hardware scenarios the limited time resolution R introduces additional error as well as a lower bound for analyzable jitter ratios σ_{RJ}/A_{DJ} (figure 7.8). Both effects further augment the optimistic error nature of the GGD method.

A comparison of different methods was also carried out in figure 7.9. It clearly shows the performance advantage of the generalized method with heavy tailed distributions (α_{RJ}<2) in comparison with the sQN method from chapter 4. If a pessimistic tail extrapolation with α_{RJ}≥1 is the crucial design criterion, Weinstein's method from [135] is the best suited choice.

The proposed method has been implemented using a minimum of constraints in order to facilitate a broad application field. Improvements can possibly be realized by considering only specific shapes or narrow parameter ranges, and by combining them with more suitable optimization criteria and fitness measures then the ones utilized here.

Summarizing these results, the GGD method must be utilized very carefully when characterizing unknown tail shapes. In simulation scenarios it provides accurate extrapolations, although with an undesired optimistic error bias. A reduction of the number of bins R as required for hardware applications is not recommended, since obtained results become highly unreliable. Summarized parts of this chapter have also been published in [C5].

8. An Accurate Behavioral Model for High-Speed PLLs

To highlight the practical aspect of jitter analysis methods, a typical application with system modeling and simulation is presented. Therefore, an accurate behavioral model of a high-speed transceiver is implemented. It has also been realized as test structure [62], and the goal is to analyze its jitter behavior as typically required for system development and verification. The transceiver is modeled as system level model according to the top-down methodology [63], using SystemC as a C++ based library [38, 41, 61]. Therefore, system blocks are first brought into an abstract design, and then successively refined down to the desired accuracy.

The model realizes a charge-pump PLL (CPLL) for high-speed clock and data recovery at 3Gb/s, and is intended for the S-ATA communication standard [118]. It uses accurate transient simulations to analyze and predict the jitter behavior in terms of phase noise PSDs and jitter transfer functions. As an enhanced version of a prior event driven approach [50], it affords accurate analysis of jitter generation and propagation effects inside a CPLL.

In the following, the basic CPLL model is described together with prior modeling approaches before proceeding to an enhanced event-driven model for accurate behavioral simulations. In the analysis section, phase noise spectra, time domain parameters and jitter transfer characteristics are derived and compared with measurements from the test structure, to demonstrate applicability of the model, as well as the tail fitting method from chapter 4.

8.1. Modeling Principle

In this section an accurate model for jitter and phase noise analysis of a typical CPLL structure is developed. It is intended as clock and data recovery (CDR) unit for serial high-speed interfaces. In figure 8.1 the functional block scheme is depicted, running at a serial data rate of 3Gb/s. The CPLL is composed of an Alexander type bang-bang phase detector (BB-PD), a charge-pump, a second order passive loop filter, and a voltage controlled oscillator (VCO) which is preceded by a gain regulator.

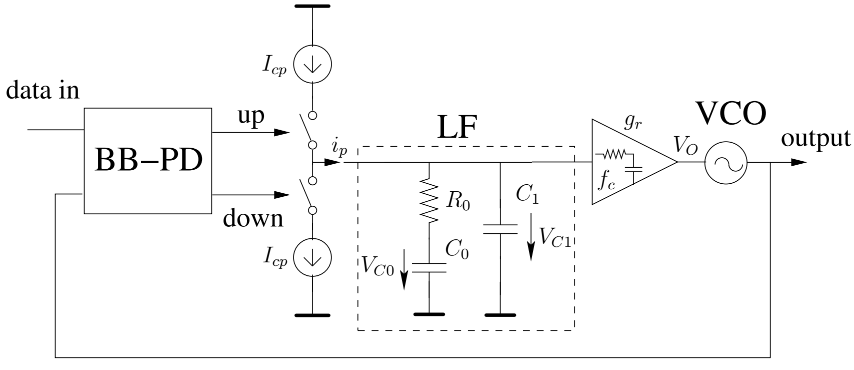


FIGURE 8.1.: Functional block scheme of the CPLL.

Only transient simulations are able to accurately reflect the true time domain behavior and cope with the non-linear loop dynamics. For such simulations, two different modeling principles have been developed in recent years. The first one is based on event-driven concepts [1,21,46,50,77], where the analog part of the CPLL (i.e. the loop filter) is replaced by a set of non-linear recursive equations. These are exact solutions to the loop filter difference equations. Hence, they are able to determine the exact time instant of the subsequent VCO clock, which transforms the analog circuit part into an event-driven block. Since all other circuit components are digital blocks, the complete CPLL can thus be simulated as an event-driven system, which is typically characterized by its non-uniform simulation time steps.

The second principle [107,108] instead, uses uniform time steps for simulation. The analog loop filter is initially converted into discrete-time via impulse-invariance or bilinear transform. The quantization noise of asynchronous events is subsequently considered by varying the amplitude of digital signals according to the location of the transition edge between sampling instances. This yields a highly accurate signal representation in discrete-time, combined with a fast model implementation which has also found use in modern PLL simulators [109].

In this section an enhanced event-driven model is implemented according to the first modeling technique as explained in [46,50]. Although the principle from [107] might afford very fast simulations and thus be an intuitive choice, it does not allow for a dynamic variation of model components. This is an important point, since the present CPLL model is only a part of an overall transceiver structure which requires a careful stability analysis. Especially when changing between different operating modes, such as startup phase and normal lock-to-data operation, a stable PLL behavior must always be guaranteed. The event-driven model uses the actual physical state of analog filter components to recursively determine successive states. Thus, it directly reflects the physical behavior of the PLL loop filter at any calculated time instant, which also allows for dynamic filter variations. This way, one can easily switch between different operating modes. A discrete-time filter instead cannot include dynamic variations, because its state variables do not reflect the actual physical state.

As modeling environment, the SystemC [41,61] programming language affords simple system level simulations, and can directly embed the developed jitter analysis methods. In the following subsections a closer look to the event-driven model is given. The basic model is based on an exact solution for the 3rd order CPLL [50], and is here enhanced with a noise model for the VCO [122], an additional parasitic pole for the gain regulator, and some BB-PD caused non-ideal effects.

8.1.1. Basic Event-Driven Model

In order to construct an event-driven CPLL model, a delay based signal description for each of the blocks in figure 8.1 is required. This is easily achieved with pure digital blocks, where the output only changes when input signal events occur. The BB-PD for example, only generates a logical up or down pulse when triggered by the VCO clock. According to the phase difference between input data and VCO clock, it sets the up or down signals at the output accordingly. The charge-pump then converts these signals into current pulses i_p of defined amplitude.

The critical component is the analog loop filter together with the VCO. A mathematical solution is required to calculate the next time instant where the VCO completes the clock cycle and thus, triggers the BB-PD over the feedback path. Once this time instance is known, the simulation time progress can be fully described in terms of an event list which is handled by the SystemC scheduler. The mathematical description of the analog circuit starts by assuming a linear relation between VCO frequency and the control voltage V_O of the loop filter output:

$$f_{vco}(V_c) = K_v \cdot V_O(t) + f_0 \tag{8.1}$$

where K_v is the linear frequency slope and f_0 the zero voltage frequency. In a practical VCO, this simple relation is hardly valid for the complete tuning curve of the oscillator. However, in lock-to-data mode it is constantly driven at the transmission rate and thus, must only be valid for a local operating point. The oscillator output phase can be expressed as integral of the VCO frequency, and hence:

$$\Theta_{vco}(t) = \Theta_{vco}(t_0) + 2\pi \int_{t_0}^{t_0+t} \left[K_v V_O(\tau) + f_0\right] d\tau \tag{8.2}$$

The charge-pump generates discrete current pulses for a second order loop filter. These current pulses are assumed as constant $i_p \in \{+I_{cp}, 0, -I_{cp}\}$, without depending on the loop filter voltage or other non-linear effects. The behavior can thus be described in time domain with two differential equations, obtained by Kirchhoff's laws:

$$\frac{di_1(t)}{dt} + i_1(t) \cdot \frac{C_0 + C_1}{C_0 R_0 C_1} = \frac{i_p}{C_0 R_0} \tag{8.3a}$$

$$\frac{di_0(t)}{dt} + i_0(t) \cdot \frac{C_0 + C_1}{C_0 R_0 C_1} = \frac{i_p}{C_1 R_0} \tag{8.3b}$$

Solving these differential equations and representing the variables in voltage domain, leads to the recursive equations:

$$V_{C0}(t) = \frac{1}{C_0} \cdot \int i_0(\tau) d\tau = V_{C0}(t_0) + \frac{1}{C_0} \cdot \left[\frac{i_0(t_0) - \beta_2}{\beta_1}\left(1 - e^{-\beta_1 t}\right) + \beta_2 \cdot t\right] \tag{8.4a}$$

$$V_{C1}(t) = \frac{1}{C_1} \cdot \int i_1(\tau) d\tau = V_{C1}(t_0) + \frac{1}{C_1} \cdot \left[\frac{\beta_2 - i_0(t_0)}{\beta_1}\left(1 - e^{-\beta_1 t}\right) + \beta_3 \cdot t\right] \tag{8.4b}$$

with

$$V_O(t) = V_{C1}(t), \quad i_0(t_0) = \frac{V_{C1}(t_0) - V_{C0}(t_0)}{R_0}$$

$$\beta_1 = \frac{C_0 + C_1}{C_0 R_0 C1}, \quad \beta_2 = \frac{C_0 \cdot i_p}{C_0 + C_1}, \quad \beta_3 = \frac{C_1 \cdot i_p}{C_0 + C_1}$$

where the gain regulator with parasitic pole f_c and gain g_r in figure 8.1 is not considered for the moment. We can now insert equation (8.4b) into equation (8.2), and yield the final recursive equation for the VCO phase:

$$\Theta_{vco}(t) = \Theta_{vco}(t_0) + 2\pi \left\{ t\Big(f_0 + K_v V_O(t_0)\Big) + \frac{K_v}{C_1}\left[\frac{i_0(t_0) - \beta_2}{\beta_1^2}\left(1 - e^{-\beta_1 t} - \beta_1 t\right) + \beta_3 \frac{t^2}{2}\right]\right\} \tag{8.5}$$

The goal is to determine the exact time instant t where the VCO phase reaches $\Theta_{vco}(t)=2\pi$ and thus, produces the next clock edge for the BB-PD, which finally closes the feedback loop. Since equation (8.5) cannot be inverted, a Newton iteration is utilized to recursively identify t. In our case the function of interest is $\Theta_{vco}(t)$ which converges toward 2π and thus, we have the iteration:

$$t_{n+1} = t_n - \frac{\Theta_{vco}(t_n) - 2\pi}{\Theta'_{vco}(t_n)} = t_n - \frac{\Theta_{vco}(t_n) - 2\pi}{K_v V_c(t_n) + f_0} \tag{8.6}$$

The Newton method typically converges after the second or third iteration step by reaching the maximum simulator time resolution of 1fs. With this basic model from [50], additional non-ideal and non-linear effects can now be included, in order to significantly improve accuracy of the given CPLL model.

8.1.2. VCO Noise Model

An event-driven phase noise model of the VCO is included with the basic CPLL. The model was developed in [122] and simulates the typical phase noise spectrum $S_\Phi(\Delta\omega)$ of an oscillator as already introduced in section 2.1.1. Such a Leeson process consists of three distinct noise regions, which are uniquely specified using four parameters: the flicker noise corner frequency f_{fl}, the measured phase amplitude A_1 with corresponding frequency f_1 in the $1/f^2$ region, and the phase noise floor amplitude A_{PhN}.

The model is implemented as an independent time domain random process which delivers a noise frequency each time it is called. This frequency value is shaped according to the spectral density of the Leeson process, and combined with the basic noise-free CPLL model:

$$t_{id+jit} = \frac{1}{1/t_{id} + f_{jit}} \tag{8.7}$$

where t_{id} is the ideal time period until the next VCO clock and f_{jit} the current noise frequency.

A Leeson noise generator has been implemented in [9], and consists of a random number generator that supplies different discrete time filters for the three noise regions. The basic concept is depicted in figure 8.2, where the gain factors are calculated as follows:

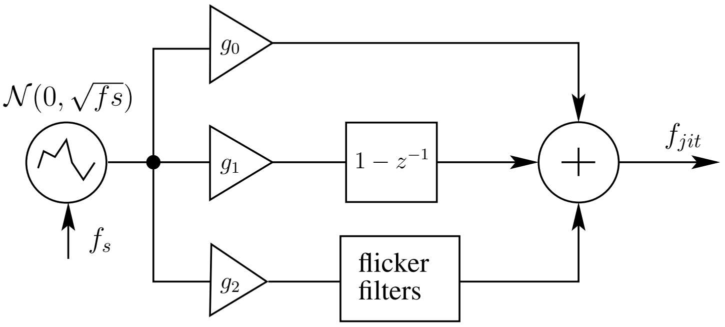


Figure 8.2.: Leeson noise generator [9].

$$g_0 = f_1\sqrt{10^{(A_1/10)}}\,, \quad \sim 1/f^2 \text{ noise} \tag{8.8a}$$

$$g_1 = 10^{(\Delta/20)}\,, \quad \Delta = A_{PhN} - 10\log_{10}(4\pi^2/f_s^2)\,, \quad \sim 1/f^0 \text{ noise} \tag{8.8b}$$

$$g_2 = f_1\sqrt{f_{fl}} \cdot 10^{(A_1/20-1.2704)}\,, \quad \sim 1/f^3 \text{ noise} \tag{8.8c}$$

The flicker noise filter bank according to [122] is realized with eight filters of increasing cut-off frequency, where the first one is located at 100 Hz.

$$y_k[n] = a_k\, y_k[n-1] + b_k\, g_2\, x[n]\,, \quad k = [0,\ldots,7] \tag{8.9a}$$

$$a_k = 1 - 2\pi\frac{10^{(k+2)}}{f_s}\,, \quad b_k = 2\pi\frac{10^{(k/2+2)}}{f_s} \tag{8.9b}$$

$$f_{fl}[n] = \sum y_k[n] \tag{8.9c}$$

where $x[n]$ are the random samples from the noise generator at the rate f_s. The overall jitter frequency f_{jit} is the sum of the three noise components:

$$f_{phn}[n] = g_1(x[n] - x[n-1])\,, \quad f_{f1}[n] = g_0\, x[n] \tag{8.10a}$$

$$f_{jit} = f_{fl} + f_{phn} + f_{f1} \tag{8.10b}$$

The sampling rate f_s must be sufficiently large to guarantee for correct noise generation and thus, simply the maximum VCO frequency is used.

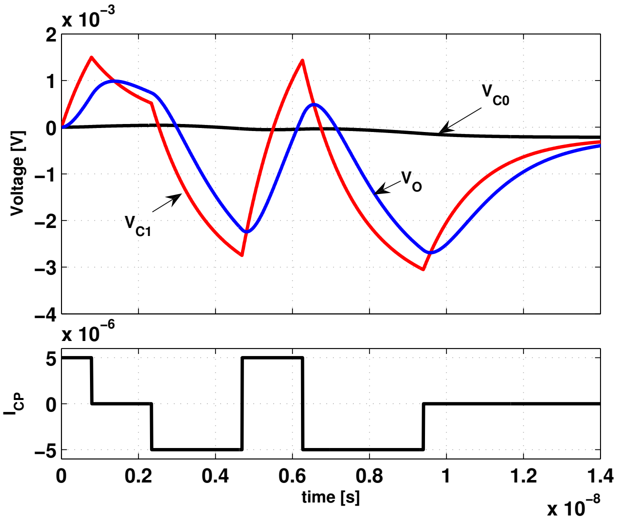


Figure 8.3.: Loop filter voltage behavior depending on input current I_{cp}, the voltages can be found in the block scheme, figure 8.1.

8.1.3. Gain Regulator and BB-PD

To model the gain regulator, an additional parasitic pole with cut-off frequency f_c and gain g_r must be included with the event-driven model. The pole reflects the VCO pre-amplifier influence as depicted in figure 8.1, and significantly increases accuracy of the real hardware behavior.

The derivation of the model equations is equivalent to the prior analysis, and yields the following state equations:

$$\begin{aligned} V_O(t) = V_O(t_0) + \frac{g_r}{C_1}\Bigg[\frac{\beta_2 - i_0(t_0)}{\omega_c - \beta_1}\left(\frac{\omega_c}{\beta_1}\left(1 - e^{-\beta_1 t}\right) - \left(1 - e^{-\omega_c t}\right)\right) - \cdots \\ - \frac{\beta_3}{\omega_c}\left(1 - e^{-\omega_c t}\right) + \beta_3 t\Bigg] + \left(V_{C1}(t_0)\cdot g_r - V_O(t_0)\right)\cdot\left(1 - e^{-\omega_c t}\right) \end{aligned} \tag{8.11}$$

$$\begin{aligned} \Theta_{vco}(t) = \Theta_{vco}(t_0) + 2\pi\Bigg\{ f_0 t + \frac{g_r K_v}{C_1 \omega_c}\Bigg[\frac{\beta_2 - i_0(t_0)}{\omega_c - \beta_1}\cdot\Bigg(1 - e^{-\omega_c t} - \omega_c t - \cdots \\ - \frac{\omega_c^2}{\beta_1^2}\left(1 - e^{-\beta_1 t} - \beta_1 t\right)\Bigg) + \left(\frac{\beta_3}{\omega_c} - V_{C1}(t_0)C_1\right)\left(1 - e^{-\omega_c t} - \omega_c t\right) + \cdots \\ + \frac{\beta_3 \omega_c}{2} t^2\Bigg] + \frac{K_v}{\omega_c} V_O(t_0)\left(1 - e^{-\omega_c t}\right)\Bigg\} \end{aligned} \tag{8.12}$$

The VCO control voltage V_O is now given by the gain regulator output which introduces an additional, third state variable. The subsequent VCO clock period is calculated using equation (8.12) and replaces equation (8.5) of the prior basic model.

Figure 8.3 shows the difference of the voltage behavior between basic and enhanced model, characterized by V_{C1} and V_O respectively. The additional parasitic pole highlights a significant influence on the voltage course, and is thus essential for reflecting the real circuit behavior.

The digital BB-PD block is modeled with a propagation delay t_{del} and the non-ideal behavior of data latches. When the VCO clock triggers close to an input data transition, digital output data suffers from meta-stability, offset voltages, or setup and hold time violations. These effects are considered by introducing a random output value at the logical up or down signals, if one of the criteria is violated.

8.1.4. Default Model Parameters

The complete model can now be utilized for behavioral simulations. An analysis run on an Intel Core Duo $2.2\,\mathrm{GHz}$ laptop is typically able to collect 10–$20\mathrm{k}$ jitter values per second of simulation time, depending on the transition density of the data pattern.

To prove model accuracy and applicability, a set of default parameters as used with the hardware test structure [24] is given in table 8.1. While loop filter, charge-pump and VCO conversion gain K_v are given by their nominal values, the phase noise coefficients were specified from prior measurements. The gain regulator pole f_c, propagation delay of the BB-PD and meta-stability range were pre-estimated and finally chosen to match the phase noise spectra at the different parameter configurations of figure 8.4.

Component	Parameters & Values
Loop filter	$R_0 = 700\,\Omega$, $C_0 = 70\,\mathrm{pF}$, $C_1 = 2\,\mathrm{pF}$
Charge-pump	$I_{cp} = 5\,\mu\mathrm{A}$
VCO	$K_v = 2.7\,\mathrm{GHz/V}$, $f_{fl} = 10\,\mathrm{MHz}$ $A_1 = -120\,\mathrm{dBC}$ @ $f_1 = 10\,\mathrm{MHz}$ $A_{PhN} = -138\ \mathrm{dBC}$
Gain regulator	$g_r = 1.0$, $f_c = 250\,\mathrm{MHz}$
Phase detector	$t_{del} = 150\,\mathrm{ps}$, $V_{meta} = \pm 1\,\mathrm{mV}$
Data pattern	$\mathrm{Pat} = 0101\ldots$

Table 8.1.: Default model parameter settings.

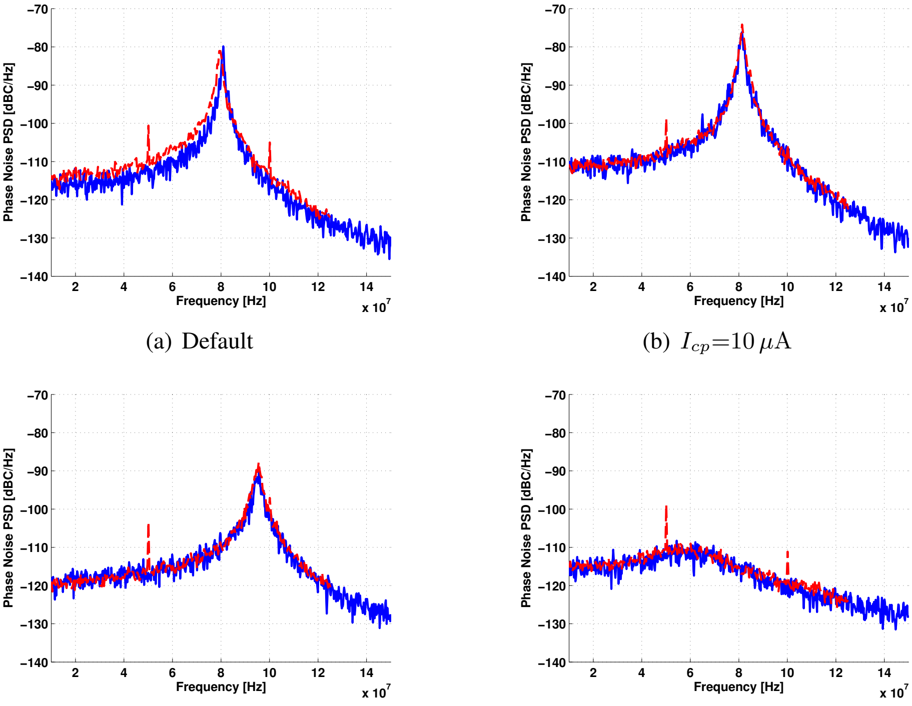
8.2. Jitter and Phase Noise Analysis

In this section the jitter and phase noise behavior of the event-driven model is compared with measurements from the hardware test structure [24]. Using the definitions from section 2.1.2, phase noise spectra are determined as well as key parameters that characterize jitter in time domain. Finally two different methods for deriving the jitter transfer function of a PLL are presented and compared with measured curves.

8.2.1. Closed Loop Phase Noise

A first demonstration of model validity is given by the comparison between measured and simulated power spectral densities (PSD) of the PLL jitter. Therefore, jitter-free signal transitions are assumed at the CDR data input, and absolute jitter values j_{abs} as defined by equation (2.4) are gathered from the VCO output clock. For model simulations, the PSD is calculated using the Welch Periodogram [103] with a Hann window of 50% overlap, and eight averaged periods of N=16384 jitter samples each. For the measurements from [24], an Agilent E4440A spectrum analyzer was utilized.

The phase noise PSD is very sensitive to parameter variations, but the comparison in figure 8.4 shows excellent matching at various parameter configurations. This mainly originates from the accurate behavioral model. The first spectrum is given by the default parameters from table 8.1, while the other plots vary one of the parameters as specified in the caption. In the measured



(a) Default

(b) I_{cp}=10 μA

(c) R_0=400Ω

(d) Pat=11110000...

Figure 8.4.: Measured (dashed red) and simulated (solid blue) phase noise PSD over different parameter settings.

signals, additional spectral spurs are observed. They originate from a 50MHz clock which is used as on-chip reference frequency. Besides these spurs, the model is able to accurately reflect the true phase noise behavior. In combination with fast simulations, this allows for a deep and thorough system exploration.

Several common time-domain parameters for jitter characterization can also be determined from the statistics of absolute jitter. They form simple alternatives to the spectral description of phase noise. Typically, one specifies the standard deviation or RMS value of accumulated jitter:

$$\sigma_{acc}(m) = \text{RMS}\big(j_{acc}^{(m)}\big) = \text{RMS}\big(j_{abs,k} - j_{abs,k+m}\big) \tag{8.13}$$

where $j_{abs,k}$ (section 2.1.2) is the absolute jitter sequence of the VCO clock. It can be calculated from the autocorrelation function r_{jabs} of absolute jitter values [42, 99]:

$$\sigma_{acc}^2(m) = 2 \cdot (r_{jabs}(0) - r_{jabs}(m)) \tag{8.14}$$

Commonly used RMS values include absolute jitter σ_{abs}, long term jitter σ_{lt}, period jitter σ_{per} or maximum jitter σ_{max} [22, 42], and are defined as:

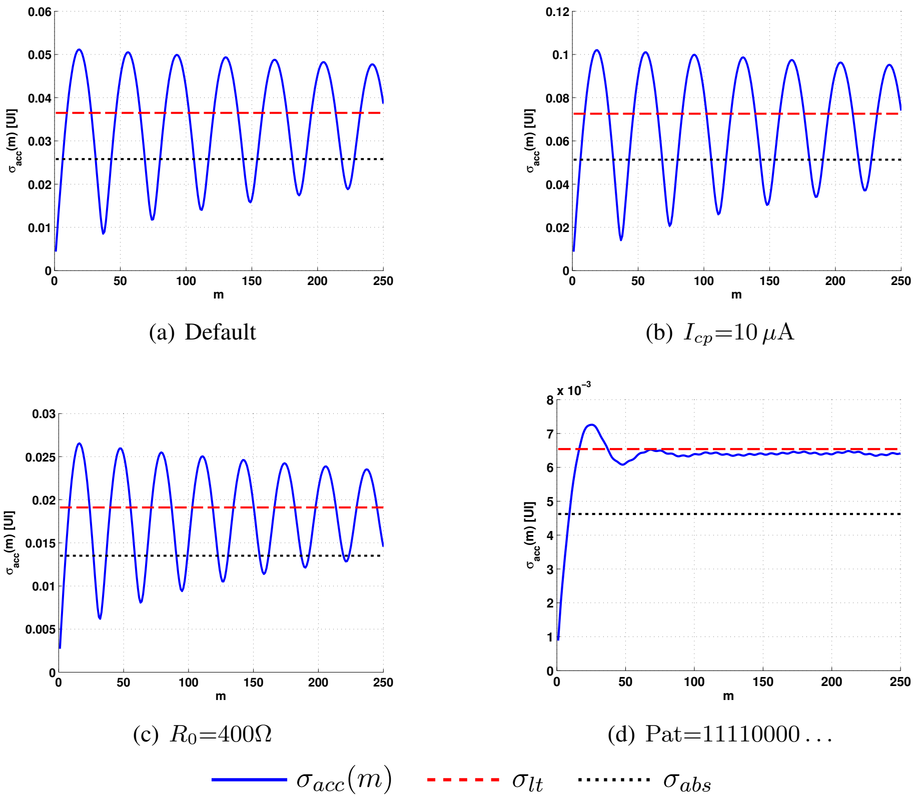
$$\sigma_{abs} = \text{RMS}\big(j_{abs,k}\big) = \sqrt{r_{jabs}(0)} \tag{8.15a}$$

$$\sigma_{lt} = \sigma_{acc}(m)\Big|_{m\to\infty} = \sqrt{2} \cdot \sigma_{abs} \tag{8.15b}$$

$$\sigma_{per} = \sigma_{acc}(1) \tag{8.15c}$$

$$\sigma_{max} = \max\big(\sigma_{acc}(m)\big) \tag{8.15d}$$

where especially σ_{lt} is often used to specify the overall performance of a PLL. In figure 8.5 the RMS value of accumulated jitter $\sigma_{acc}(m)$ is depicted for the same parameter configurations as in



(a) Default

(b) I_{cp}=10 μA

(c) R_0=400Ω

(d) Pat=11110000...

Figure 8.5.: RMS values of accumulated $\sigma_{acc}(m)$, absolute σ_{abs} (dotted line) and long term jitter σ_{lt} (dashed line). The curves are constructed using a sample size of N=10^5.

Configuration	σ_{abs}[UI]	σ_{lt}[UI]	σ_{per}[UI]	σ_{max}[UI]	$\sigma_{abs,psd}$[UI]
Default	0.0258	0.0365	0.00440	0.0512	0.0315
I_{cp}=10 μA	0.0513	0.0726	0.00874	0.1020	0.0611
R_0=400Ω	0.0135	0.0191	0.00273	0.0265	0.0170
Pat=11110000...	0.0046	0.0065	0.00089	0.0073	0.0058

Table 8.2.: RMS jitter values from figure 8.5. 1 UI = 333 ps.

figure 8.4, together with calculated values for σ_{abs} (dotted line) and σ_{lt} (dashed line). In the course of $\sigma_{acc}(m)$ especially a periodicity given by the spectral peak can be noticed.

Table 8.2 lists the RMS values from equation (8.15). The absolute jitter can also be determined from the measured phase noise spectrum using Parseval's Theorem [43, Annex D], and yields $\sigma_{abs,psd}$ in the last column. Measured RMS values are generally larger than simulated ones, which is due to mismatches in the peak region and additional parasitic spurs. As can also be observed, an increase of the charge-pump current leads to a linear increase of absolute jitter and the oscillation magnitude in figure 8.5.

8.2.2. Jitter Transfer Function

The jitter transfer function $T(f_{SJ})$ is a common way of characterizing PLL jitter [7, 46, 121]. Typically, it follows a low pass curve, because low jitter frequencies can easily be followed by the loop filter, while higher frequencies are attenuated.

$T(f_{SJ})$ of the given hardware structure can be measured with a BERT-Scope when using sinusoidal jitter of frequency f_{SJ} and amplitude A_{SJ} at the CPLL input. The BERT-Scope determines the timing budget of jitter distributions at the output, using a built-in tail fitting method. In order to get a reference value, the first point is measured at a sinusoidal jitter frequency of typically f_{SJ}=10kHz, so that the loop filter can easily follow the input jitter and thus, a transfer characteristic of $T(f_{SJ})\overset{!}{=}1$ can be assumed. The corresponding timing budget can subsequently be used as a reference for determining the complete $T(f_{SJ})$ curve over varying frequency f_{SJ}.

In simulations, the same approach requires the use of the tail fitting method from chapter 4. That is, distributions are gathered from IO jitter values of the behavioral model and extrapolated, in order to identify the timing budget at the target BER$=10^{-12}$. The typical sample size of simulated distributions is significantly smaller (typically 4 to 5 orders of magnitude) compared to real-time BERT measurements. This is an essential drawback for simulations, since the timing budget has to be estimated from higher BER levels, where the number of Gaussian tail samples may not be sufficient. Further, the fitting algorithm of the BERT-scope may produce different estimates.

Another problem domain can be highlighted with phase noise spectra, when additional random jitter (RJ) is added to the data input. Figure 8.6 yields a clear amplitude difference between measured and simulated PSDs. This can be due to model inaccuracies, non-ideal behavior of the RJ generator, or additional phase noise as caused by interconnect wires and the analog receiver circuit. Therefore, an additional, alternative analysis method is preferred for model simulations, in order to truly reflect the transfer characteristic. Such a method is here realized using the FFT spectrum of absolute jitter at the PLL output. The characteristics of both tail fitting and spectral methods are briefly summarized:

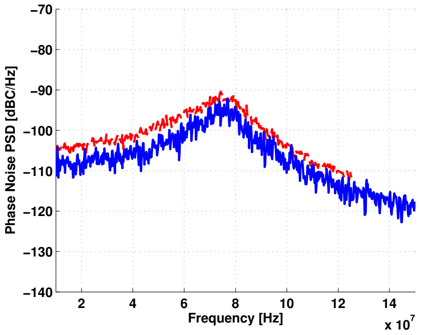


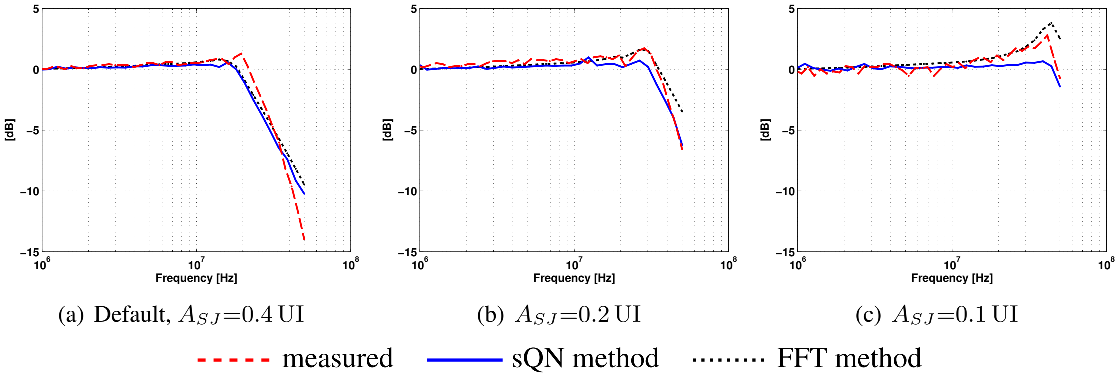
FIGURE 8.6.: Phase noise PSD mismatch with RJ=0.3 UI and default parameters (table 8.1).

Tail Fitting Method For each calculated frequency point, N=10^6 IO-jitter samples are collected and the scaled Q-normalization (sQN) method is applied ($\hat{c}_{1.2}$ optimization, ΔP_t=10^2), while the PLL model is stressed with the jitter frequency f_{SJ}. Only estimated DJ$_{pp}$ values are used for determining the jitter transfer function $T(f_{SJ})$. The first frequency value is assumed to be ideally followed by the loop filter, and thus DJ$_{pp}(f_{SJ,min})$ is used as a reference to calculate

$$T(f_{SJ}) = \mathrm{DJ}_{pp}(f_{SJ})/\mathrm{DJ}_{pp}(f_{SJ,min}). \tag{8.16}$$

Spectral Method Transfer function values are calculated from the frequency bins of the jitter spectrum, which are calculated by the FFT of absolute output jitter. For each frequency point, N samples are collected and FFT transformed, while the PLL model is stressed with the corresponding sinusoidal jitter frequency f_{SJ} and amplitude A_{SJ}. A sample size of $N \geq 10^5$ is chosen to allow for coherent sampling:

$$N_C/N = f_{SJ}/f_s \tag{8.17}$$

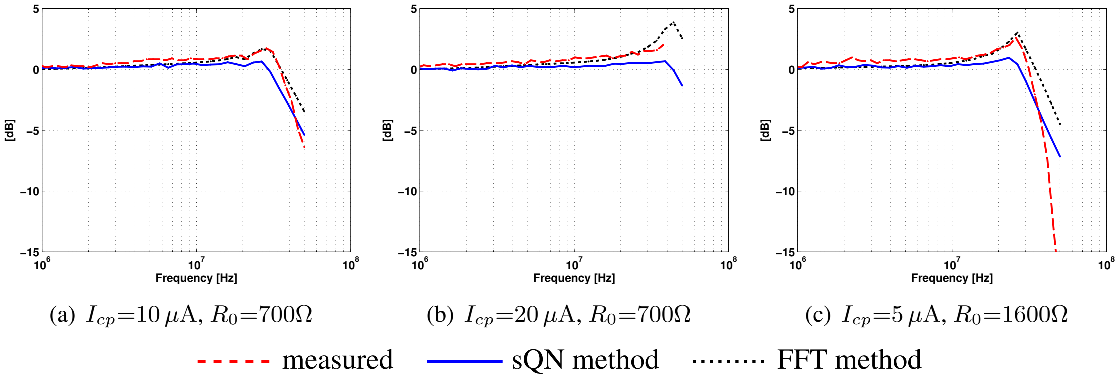


(a) Default, A_{SJ}=0.4 UI

(b) A_{SJ}=0.2 UI

(c) A_{SJ}=0.1 UI

Figure 8.7.: $T(f_{SJ})$ over varying jitter amplitude A_{SJ}.



(a) I_{cp}=10 μA, R_0=700Ω

(b) I_{cp}=20 μA, R_0=700Ω

(c) I_{cp}=5 μA, R_0=1600Ω

Figure 8.8.: $T(f_{SJ})$ over varying loop filter parameters.

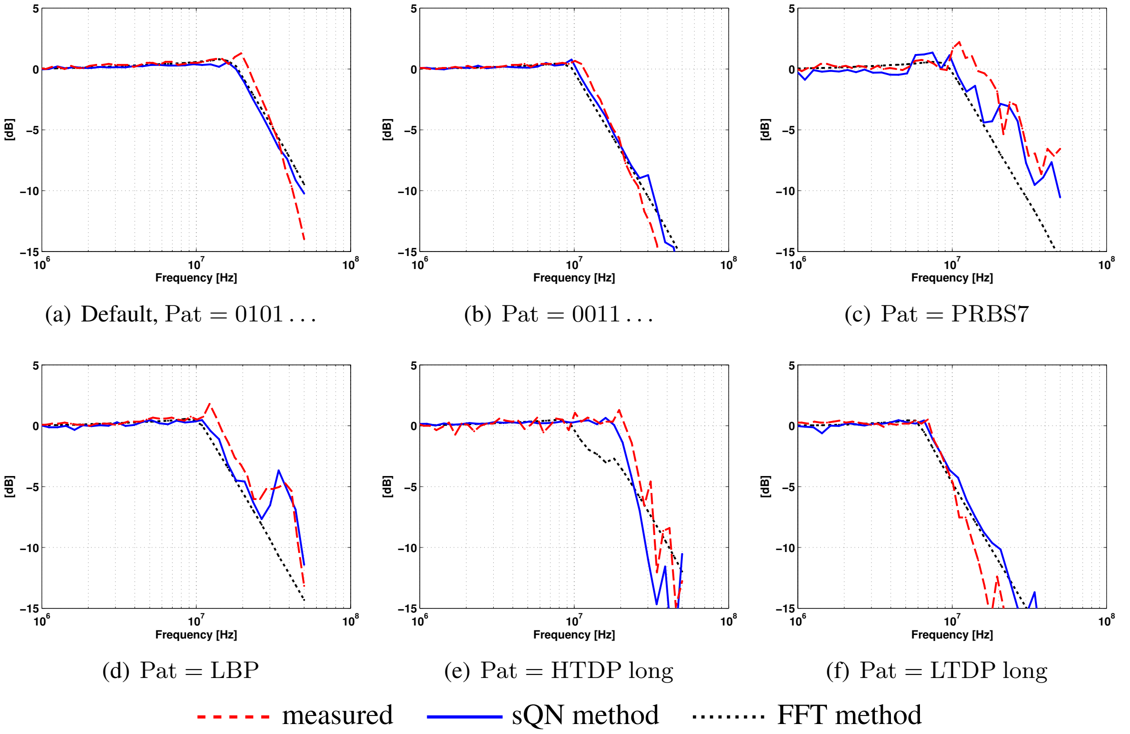
where N_C is the bin where the transfer function value is determined. If for example f_{SJ}=1 MHz, f_s=3 GHz ⇒ N_C=34, N=102000. Further,

$$T(f_{SJ}) = |\mathrm{FFT}_{N_C}(j_{abs,k})|/(A_{SJ}/4) \tag{8.18}$$

The jitter amplitude represents a peak-to-peak value while the FFT produces a double-side spectrum of half amplitude and thus, A_{SJ} must be divided by four.

Both simulation methods use a logarithmic grid f_{SJ}=$[10^5, 5{\cdot}10^7]$ MHz of 50 frequency points. With the Intel 2.2GHz laptop and N=10^5, a full transfer function can be determined in a few minutes. The default model parameters were already given in table 8.1, with an additional default jitter amplitude of A_{SJ}=0.4 UI.

An initial comparison of transfer functions over different jitter amplitudes A_{SJ} is given in figure 8.7, where the typical low pass behavior is observed. Since the CPLL is a non-linear system, the cut-off frequency of $T(f_{SJ})$ varies too. Both simulation methods are able to correctly track the cut-off frequency. Although the simulated curves generally highlight acceptable matching with the measured ones, the sQN method generally underestimates the true jitter transfer behavior. This can especially be observed in the transition region of figures 8.7(c) and 8.8(c). However, the FFT method correctly reflects the measured peaks. In figures 8.7 and 8.8 generally, both simulation methods do not exactly match the slope of $T(f_{SJ})$ in the cut-off region. It is likely that this effect ranges from additional parasitic poles which have not been considered, such as the line termination, input amplifier or equalizer circuit. Figure 8.9 depicts $T(f_{SJ})$ for different test patterns, as



(a) Default, Pat = 0101 . . .

(b) Pat = 0011 . . .

(c) Pat = PRBS7

(d) Pat = LBP

(e) Pat = HTDP long

(f) Pat = LTDP long

FIGURE 8.9.: $T(f_{SJ})$ over varying test pattern.

specified in [118, section 7.2.4]. Unlike the FFT method, the sQN tail fitting method is now able to correctly follow the measured course, due to the same analysis principle.

8.3. Summary

A fast system level model for accurate behavioral simulations of charge-pump PLLs has been developed. Unlike modeling approaches with uniform time steps, an event-driven model uses state variables that reflect the real physical state and thus, allow for dynamic run-time variations. The model forms an enhanced version of a prior approach [50] with included gain regulator pole, VCO noise model and non-ideal behavior of the BB-PD. Depending on the transition density of the selected data pattern, it is typically able to collect 10-20k jitter values per second simulation time on an Intel Core Duo 2.2GHz laptop.

Simulation results of the closed loop phase noise over varying parameter configuration proved excellent agreement with jitter measurements from an existing hardware structure (figure 8.4). In addition to the PSD spectra, also the RMS values of accumulated jitter were determined in figure 8.5. They especially highlight the observed PSD peaks.

Two simulation methods were compared with measured jitter transfer functions $T(f_{SJ})$ in figures 8.7-8.9. One of the two simulation methods is based on the equivalent measurement principle, where a tail fitting method identifies the DJ peak-to-peak characteristics DJ_{pp} over varying jitter frequency. Using a reference value at low frequencies, equation (8.16) allows for determining $T(f_{SJ})$. In measurements, DJ_{pp} was determined with a BERT scope, while for simulations the sQN method from chapter 4 was utilized. As it can be difficult to observe certain phase noise effects with the sQN method in simulations, a second analysis method based on spectral analysis was implemented as well. It calculates the transfer function by identifying the amplitude of absolute jitter at the PLL output.

Both methods correctly identify the cut-off frequency of measured $T(f_{SJ})$ in figures 8.7-8.9. The spectral method is further able to track the peaking behavior of jitter transfer functions, which is generally underestimated by the sQN method. Transfer characteristics of different test pattern are instead only reflected correctly by the sQN method, due to the same analysis principle.

The presented model and simulation results have also been published in [C2].

9. A Method for Fast Jitter Tolerance Analysis

Jitter analysis methods can be used for identifying jitter tolerance (JTOL) curves of high-speed PLLs. This chapter focuses on this application field and realizes an algorithm for the automatic determination of such curves. Due to the influence of timing jitter along the transmission channel, high-speed PLLs and CDR structures have to provide a certain robustness against timing variations. Thus, interface standards often specify tolerance masks [43] to define minimum bounds for jitter tolerance which must be guaranteed by a system design.

A JTOL curve describes the robustness of a PLL against an injected sinusoidal jitter frequency. Therefore, a jitter amplitude must be determined where the TJ budget exactly covers the complete bit period or UI. This corresponds to an inverse problem which is additionally influenced by the statistical variation of collected distributions. Typically, a JTOL measurement scheme as depicted in figure 9.1 uses a modulated clock source with corresponding pattern generator, and is characterized by the injected sinusoidal jitter frequency f_{SJ} and amplitude A_{SJ}. The CDR under test suffers from the jittery signal and produces output data with increased error probability. A bit error rate tester (BERT) may compare the recovered data with the expected original one, and determine the resulting error rate. Equivalently, a time interval analyzer (TIA) can directly measure the time difference between the zero crossing of the analog input signal and the recovered clock edge. The obtained IO jitter values (section 2.1.2) again allow for collecting distributions that represent the error rate.

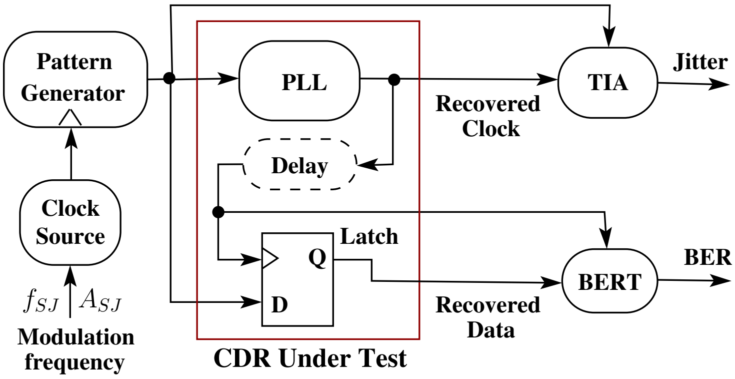


FIGURE 9.1.: JTOL measurement scheme using TIA or BERT.

The BER measurement principle has already been discussed together with built-in jitter measurement (BIJM) systems in chapter 5, which also explains the similarity between figures 9.1 and 5.1. Again, the adjustable delay element introduces a time discretization which yields a limited number of bins R per UI. The BERT measurement is easy to implement since bit errors can be counted. The result is a single probability value, meaning that a complete jitter distribution requires a sequential BER scan over all R delay steps. A TIA measurement instead, collects jitter values at every bit transition of the received data and thus, directly yields probability distributions with maximum speed. It can also be realized using R BERT elements in parallel, which requires a delay line together with a bank of latches and counters. Although area consuming, BIJMs with such a real-time TIA feature have already been realized successfully using high resolution time-to-digital converters [16, 66].

Jitter tolerance measurements are very challenging, because the amplitude of sinusoidal jitter must exactly produce the target BER at the CDR under test. Thus, one is searching for the A_{SJ} value where both distribution tails cross each other at the 10^{-12} level. So far, this JTOL test problem has been addressed either from a measurement or a simulation perspective. Methods for hardware measurements have been proposed in [32, 33, 140, 141], where especially the principle in [33] is very efficient as it is based on the Gaussian Q-normalization with subsequent tail extrapolation. Using several measurement points, a regression line can be constructed where the correct A_{SJ} value is estimated easily via extrapolation. This BERT based approach does not require the delay element in figure 9.1, but needs a considerable amount of measured bit errors, and is thus too time consuming for simulations if the whole jitter tolerance curve has to be identified over varying jitter frequency f_{SJ}. Simulation methods work equivalent to TIA based measurements. They additionally use statistical models [97] or special waveforms [4] to minimize the required number of jitter samples as much as possible, so that JTOL simulations can be carried out in a feasible amount of time. However, they can generally not be used for hardware measurements.

In this chapter an analysis method is proposed, where the jitter tolerance curve of a PLL is determined very quickly using an adaptive algorithm. The method is sufficiently fast for use with both behavioral simulations and TIA based jitter measurements. A minimum measurement time is obtained by automatically adapting the sample size of collected jitter distributions according to the dynamics of the PLL under test. This adaptive recursion utilizes the previously described sQN and QN methods from chapters 4 and 6.

In subsequent sections, first the adaptive principle of the algorithm is described. In a practical use case, then a simulation example reuses the modeled charge-pump PLL from chapter 8. Obtained simulation results show, that JTOL curves can also be determined correctly when using hardware systems of limited accuracy. In a final analysis, simulation results are compared with JTOL measurements from the given hardware structure.

9.1. Adaptive Algorithm for JTOL Analysis

As already mentioned, a JTOL analysis method is a search algorithm which identifies the jitter amplitude A_{SJ} where the error probability of the CDR under test is equal to the target BER=10^{-12}. This is an inverse problem, which must be solved independently for every jitter frequency f_{SJ}. The goal is thus, to cross the two bathtub tails of a jitter distribution at the target BER, so that TJ$_{pp}$=1 UI. As a fundamental limitation one is not able to measure or simulate such a low probability level in a feasible amount of time, which again involves the use of tail fitting methods. For performance comparison the two different methods sQN and QN, as described in chapters 4 and 6, are investigated subsequently.

Both methods suffer from statistical tail variations and thus, lead to an uncertainty of the estimated eye opening. Obviously, this uncertainty highly depends on the sample size N of a collected jitter distribution, and an ideal choice poses a fundamental problem: estimation accuracy and thus, statistical confidence ask for a large N, while fast jitter measurements require N to be as small as possible.

As a solution to this problem a twofold search method is proposed. The primary, basic search of A_{SJ} is carried out with a recursive algorithm which minimizes the extrapolation error of fitted tails. It is described by the recursion

$$A_{SJ}(n+1) = A_{SJ}(n) + \nu \cdot e(n) \tag{9.1}$$

This equation is widely used in adaptive filter theory [49] where least-mean-square algorithms or

Kalman filters are implemented, and offers a high robustness against statistical variations of the error term $e(n)$. Using a block size of N jitter values, a tail fitting algorithm can provide a cost function and is thus able to specify the error $e(n)$. The jitter amplitude for the next iteration step $A_{SJ}(n+1)$ is then determined according to equation (9.1), using the old value $A_{SJ}(n)$ and the error $e(n)$ which is additionally scaled by the learning rate parameter ν.

The second part of the algorithm adaptively adjusts the sample size N for each collected jitter distribution. The method starts with a minimum sample size N_{min} and decides after each iteration, whether N must be increased or not. As soon as a maximum number N_{max} is successfully reached, the search algorithm converges and the JTOL analysis can proceed with the next jitter frequency. This second algorithm allows for a significant speed-up of the A_{SJ} search, because only few jitter samples are needed for initial iterations. Further, the increase of N starts at a point where A_{SJ} is already quite close to the final result.

9.1.1. Cost Function

For the adaptive recursion (9.1) a suited error term $e(n)$ should behave proportional to the injected jitter amplitude A_{SJ}. In [33] it is shown that, besides the injected sinusoidal, all jitter sources in a JTOL measurement environment can be considered as uncorrelated and approximately constant. In Gaussian quantile domain this allows for the construction of a linear relationship, which is valid over a certain amplitude range.

$$Q = a \cdot A_{SJ} + b \tag{9.2}$$

Theoretically, only two measurement points would thus be sufficient to determine the unknown parameters a and b, but unfortunately the tail extrapolations suffer from statistical variations. This impedes a direct calculation of A_{SJ} without multiple evaluations, but one can still benefit from the proportional influence on Q-values and derive a cost function.

The Q-value where the timing budget covers the whole unit interval ($\mathrm{TJ}_{pp}\overset{!}{=}1$) can be determined by rewriting equation (4.2) from chapter 4:

$$\frac{(1-\mu_R-\mu_L)}{\sigma_L+\sigma_R} = \begin{cases} \frac{\sigma_L \cdot Q(p/A_L)+\sigma_R \cdot Q(p/A_R)}{\sigma_L+\sigma_R} & \text{(sQN)} \\ Q(p), \text{ with } A_L{=}A_R{=}1 & \text{(QN)} \end{cases} \tag{9.3}$$

For the QN method $Q(p){=}Q_{est}$ is directly calculated from the left hand side, while for the sQN method it can be determined recursively using a simple Newton iteration. Q_{est} must approach the desired target BER$=10^{-12}$, which gives the normalized error term $e(n)$:

$$e(n) = \frac{Q_{est} - Q(10^{-12})}{Q(10^{-12})} = \frac{Q_{est}}{7.03} - 1 \tag{9.4}$$

This result is used together with the adaptive algorithm in equation (9.1).

9.1.2. Sample Size Adaptation

The automatic adaptation of sample size N is based on the decision, whether the variance of A_{SJ} falls below the expected error variance of the tail fitting method. It forms the important heart piece of the JTOL analysis, and also decides if the search is completed or not. The algorithm uses three adjustable parameters: minimum and maximum sample size (N_{min} and N_{max}) as well as the target deviation ϵ_{conf} for amplitude values. This last parameter specifies the statistical confidence interval for the final A_{SJ} result.

A flow graph of the algorithm is given in figure 9.2, where A_{SJ} is identified for a single frequency f_{SJ}. The algorithm starts with the minimum sample size N_{min} and waits until the first

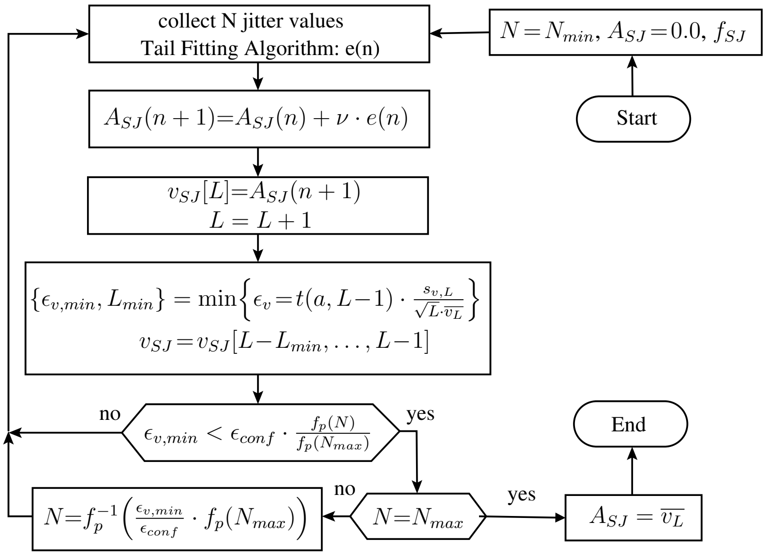

collect N jitter values
Tail Fitting Algorithm: e(n)
$N = N_{min}$, $A_{SJ} = 0.0$, f_{SJ}
Start
$A_{SJ}(n+1) = A_{SJ}(n) + \nu \cdot e(n)$
$v_{SJ}[L] = A_{SJ}(n+1)$
$L = L+1$
$\{\epsilon_{v,min}, L_{min}\} = \min\left\{\epsilon_v = t(a, L-1) \cdot \frac{s_{v,L}}{\sqrt{L \cdot \overline{v_L}}}\right\}$
$v_{SJ} = v_{SJ}[L - L_{min}, \ldots, L-1]$
no
$\epsilon_{v,min} < \epsilon_{conf} \cdot \frac{f_p(N)}{f_p(N_{max})}$
yes
End
no
$N = f_p^{-1}\left(\frac{\epsilon_{v,min}}{\epsilon_{conf}} \cdot f_p(N_{max})\right)$
$N = N_{max}$
yes
$A_{SJ} = \overline{v_L}$

FIGURE 9.2.: Flow graph of JTOL analysis algorithm.

block of jitter samples has been collected. After applying the tail fitting method, the error $e(n)$ from equation (9.4) is determined and used for updating the recursion in equation (9.1). The resulting new $A_{SJ}(n+1)$ value is stored in an array $v_{SJ}[0, \ldots, L-1]$ of variable length, where the statistical variation of amplitudes can be observed over multiple iterations.

With blocks of only N_{min} jitter samples at the beginning, the recursion quickly settles $A_{SJ}(n)$ to a level where it constantly oscillates around its true value and exhibits statistical random walks. A measure for the statistical variation of $A_{SJ}(n)$ can be derived if only the L last recursions are considered. Assuming a normal distribution, the confidence interval of A_{SJ} is specified as t-statistic with

$$\epsilon_v = t(a, L-1) \cdot \frac{s_{v,L}}{\sqrt{L} \cdot \overline{v_L}} \tag{9.5}$$

and

$$\overline{v_L} = \frac{1}{L} \sum A_{SJ,i} \tag{9.6}$$

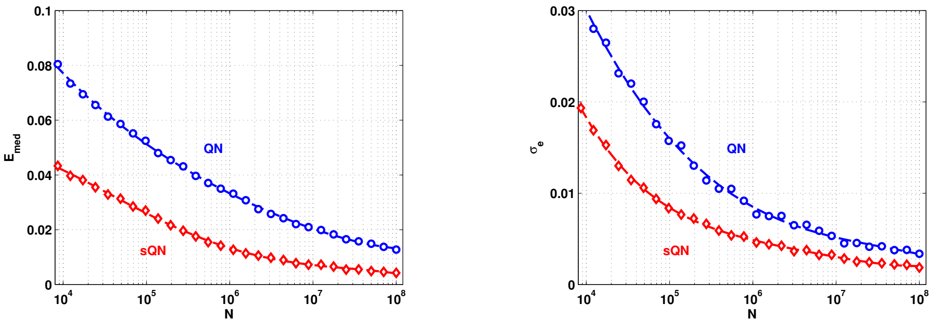
$$s_{v,L} = \sqrt{\frac{\sum A_{SJ,i}^2 - L \cdot \overline{v_L}^2}{L-1}} \tag{9.7}$$

where ϵ_v is the estimated confidence bound of a t-distribution with confidence level a=0.95 and $L-1$ degrees of freedom. It is proportional to the empirical standard deviation $s_{v,L}$ and normalized by the empirical mean $\overline{v_L}$. If ϵ_v falls below the target deviation ϵ_{conf}, the JTOL algorithm has converged.

The length L of the array v_{SJ} is continuously increased. It is incremented at every recursion, but only a subset of its newest elements is used to minimize the observed statistical variation. At each recursion, the minimum epsilon value is searched over all possible lengths:

$$\epsilon_{v,min} = \min\{\epsilon_v,\ v_{SJ}[L-k, \ldots, L-1] \mid k=2, \ldots, L\} \tag{9.8}$$

This minimum search yields an optimistic estimate of the actual statistical confidence of A_{SJ} values. It allows for quickly changing to a higher sample size N as soon as the observed optimistic tolerance $\epsilon_{v,min}$ falls below a known comparison threshold. Hence, the algorithm behavior is optimized with respect to a minimum number of recursions.



(a) Median error E_{med}

(b) Standard deviation σ_e

FIGURE 9.3.: Worst case error behavior of tail fitting methods over varying N. 4^{th} order regression polynomials, K=250, worst case distribution shapes A_{DJ}/σ_{RJ}=$1/2$ (sQN) and $1/4$ (QN).

Alg.	p_0	p_1	p_2	p_3	p_4
QN	0.2036	-0.03269	0.001823	$-3.466\cdot10^{-5}$	0.0
sQN	0.3493	-0.08615	0.008218	$-3.530\cdot10^{-4}$	$5.71\cdot10^{-5}$

TABLE 9.1.: Polynomial regression coefficients for σ_e.

The ideal comparison threshold corresponds to the expected error of the tail fitting method. This error behavior can only be approximated, because extrapolation results depend not only on the sample size N, but also on the underlying distribution shape. The CDR under test is stimulated with sinusoidal jitter and thus, collected jitter distributions are expected to consist of a bounded sinusoidal component combined with Gaussian random jitter. From the performance comparison in figures 6.11-6.15, worst case distribution shapes resulting from combined sinusoidal and Gaussian jitter components can be specified for the two fitting methods. The worst case error for each method can thus be used to obtain a simplified function of sample size N. In figure 9.3 the median error E_{med} (left) and corresponding standard deviation σ_e (right) curves are plotted. K=250 tail fits were carried out for each of the different sample sizes, and 4^{th} order polynomials were fitted to achieve a functional relationship $f_p(N)$ between sample size N and error behavior. Note, that the error bias E_{med} cannot be compensated with the JTOL algorithm since the underlying distribution shape is basically unknown. However, the standard deviation σ_e can be used as a pessimistic indicator for choosing the right sample size N. The logarithmic scaling in figure 9.3(b) leads to a polynomial

$$f_p(N) = p_0 + p_1 \cdot \log(N) + \ldots + p_4 \cdot \log(N)^4 \tag{9.9}$$

with the coefficients in table 9.1, and an analysis range of N=$[10^4, 10^8]$. In order to compare the actual confidence interval $\epsilon_{v,min}$ with the expected error of the fitting method, we can use these error polynomials and formulate the condition

$$\epsilon_{v,min} < \epsilon_{conf} \cdot \frac{f_p(N)}{f_p(N_{max})} \tag{9.10}$$

The error at $f_p(N)$ is normalized by $f_p(N_{max})$ so that the target bound ϵ_{conf} forms the reference. In the flow graph (figure 9.2) this condition decides whether the obtained A_{SJ} estimates are sufficiently accurate, so that jitter distributions of a larger sample size have to be collected in subsequent iterations. If this is the case, a new value for N is determined from the inverse of the actual $\epsilon_{v,min}$, otherwise the JTOL algorithm continues to iterate. The inverse $f_p^{-1}(N)$ is simply realized by a Newton approach.

The overall structure of the algorithm guarantees for a strictly monotonic increase of N until N_{max} is reached. With $\epsilon_{v,min}{<}\epsilon_{conf}$ and $N{=}N_{max}$ the final convergence criterion is met. Note, that in order to identify a complete jitter tolerance curve, the JTOL algorithm from figure 9.2 must be repeated for every desired frequency f_{SJ}. Some additional speed up can thus be achieved when using the amplitude result of the last frequency as initial value for the next one.

An additional reset feature has also been included with the presented JTOL algorithm. Since the jitter amplitude A_{SJ} is operated in a region where the observed jitter extends over the complete UI, the modeled PLL may easily become unstable and produce bit errors. Thus, the JTOL method also requires a well defined reset behavior. This is realized by first returning to a smaller, stable A_{SJ} value. Then the learning rate parameter ν is additionally decreased by a factor of two, so that the search algorithm is more focused on the region of interest.

9.2. Application Example

In this section the proposed JTOL algorithm is applied to the behavioral model for charge-pump PLLs from chapter 8. This model includes analog component values, a VCO noise model (Leeson process), a gain regulator pole, as well as the phase detector delay and metastability behavior.

Suitable parameters are first identified to optimize the behavior of the JTOL algorithm, and examples are given to highlight the varying loop dynamics of the PLL over f_{SJ}. Then the analysis proceeds to jitter tolerance curves where performance comparisons between hardware and software models are carried out.

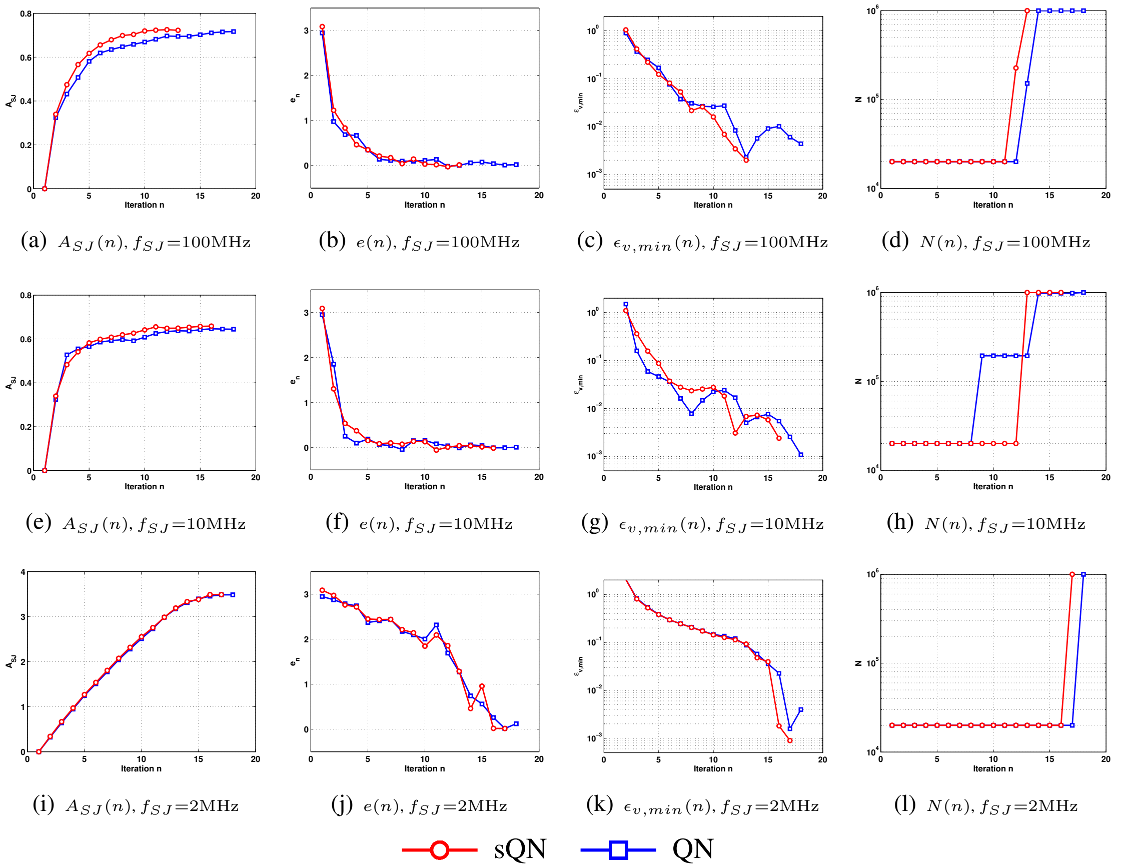
9.2.1. JTOL Parameter Optimization

First, simple simulations shall optimize the algorithmic behavior together with the modeled charge-pump PLL. The default parameters for the PLL have already been listed in table 8.1. Instead of the clock-like signal, this time the lone bit pattern (LBP) is used for compliance testing, as suggested by the 3Gb/s S-ATA standard [118]. Further, 0.18 UI of additional random jitter are superimposed to the generated jittery data signal, in order to verify jitter tolerance with respect to the specified maximum values DJ=0.42 UI and TJ=0.60 UI [118, p. 179, tab. 31].

In figure 9.4 the typical behavior of the JTOL algorithm is demonstrated for three different jitter frequencies $f_{SJ}{=}\{100, 10, 2\}$MHz. The PLL model utilizes the default parameter settings. The QN method (squares) in figure 9.4(d) for example, carries out the first eleven iterations with $N_{min}{=}2{\cdot}10^4$ data samples. Then, a subset of $L{\geq}2$ last A_{SJ} values has reached a confidence level $\epsilon_{v,min}$ which demands a larger sample size N, as determined by equation (9.10). Over successive iterations, the N parameter increases monotonically toward $N_{max}{=}1{\cdot}10^6$ until finally $\epsilon_{v,min}{<}\epsilon_{conf}$ is reached. The sQN method (circles) generally needs less iterations and jitter samples, and thus converges faster. This is due to the more accurate tail fitting principle, which also means less undesired error bias as depicted in figure 9.3(a).

A minimum sample size of $N_{min}{=}2{\cdot}10^4$ generally proved sufficient for initial tail estimates. The given PLL model typically simulates 10^6 bits within 45s on a 3GHz Intel Xeon workstation, and thus, $N_{max}{=}10^6$ forms a good trade-off between fitting accuracy and simulation time. Here, a worst case error bias of 1.3% for the sQN method is accepted (also see figure 9.3(a)). The confidence interval $\epsilon_{conf}{=}0.005$ is chosen small enough so that the final statistical variance of A_{SJ} can be neglected.

The learning rate parameter ν must also be chosen correctly in order to support quick convergence. If it is too large, A_{SJ} values will exhibit large statistical variations and the JTOL algorithm



(a) $A_{SJ}(n)$, f_{SJ}=100MHz (b) $e(n)$, f_{SJ}=100MHz (c) $\epsilon_{v,min}(n)$, f_{SJ}=100MHz (d) $N(n)$, f_{SJ}=100MHz

(e) $A_{SJ}(n)$, f_{SJ}=10MHz (f) $e(n)$, f_{SJ}=10MHz (g) $\epsilon_{v,min}(n)$, f_{SJ}=10MHz (h) $N(n)$, f_{SJ}=10MHz

(i) $A_{SJ}(n)$, f_{SJ}=2MHz (j) $e(n)$, f_{SJ}=2MHz (k) $\epsilon_{v,min}(n)$, f_{SJ}=2MHz (l) $N(n)$, f_{SJ}=2MHz

FIGURE 9.4.: Examples for the adaptive JTOL algorithm converging toward the unknown jitter amplitude A_{SJ}. Parameters are given in table 9.2.

will require more iterations before the obtained statistical confidence is sufficiently accurate. In extreme cases, it may even become unstable and produce errors. Otherwise, if ν is chosen too small the JTOL search will converge before A_{SJ} reaches the correct value, simply because also the amplitude variations are too small. The ideal ν depends on the dynamics of the investigated system and thus, changes with varying jitter frequency f_{SJ} as well as the parameter configuration of the model.

In figure 9.5 the convergence probability of the JTOL algorithm is investigated over varying ν. As model configurations, four different parameter settings as given in table 9.3 were used, each with twenty frequency points in a logarithmically scaled range of f_{SJ}=$[10^6, 10^8]$ MHz. Successful JTOL runs were counted by combining two limiting criteria: number of iterations I and overall sample size $\overline{N}$. In order to yield a valid JTOL run, both criteria must be fulfilled.

The obtained curves in figure 9.5 thus represent an empirical probability for convergence. They clearly highlight an increase in the $0.01 \leq \nu \leq 0.1$ region where the performance settles toward a maximum. When further increasing ν, a larger variation of probabilities is observed because the obtained A_{SJ} values also suffer from a strong statistical fluctuation. At largest ν, the PLL further looses its lock state several times. Thus the reset feature, as described previously, divides ν until a stable convergence of A_{SJ} values is again achieved. For the presented PLL model the learning rate is selected such that maximum performance with the QN method is achieved, which is at ν=0.11. Further, a maximum number of iterations I_{max}=50 aborts simulations if no convergence criterion is reached. All default settings for the JTOL algorithm are again summarized in table 9.2.

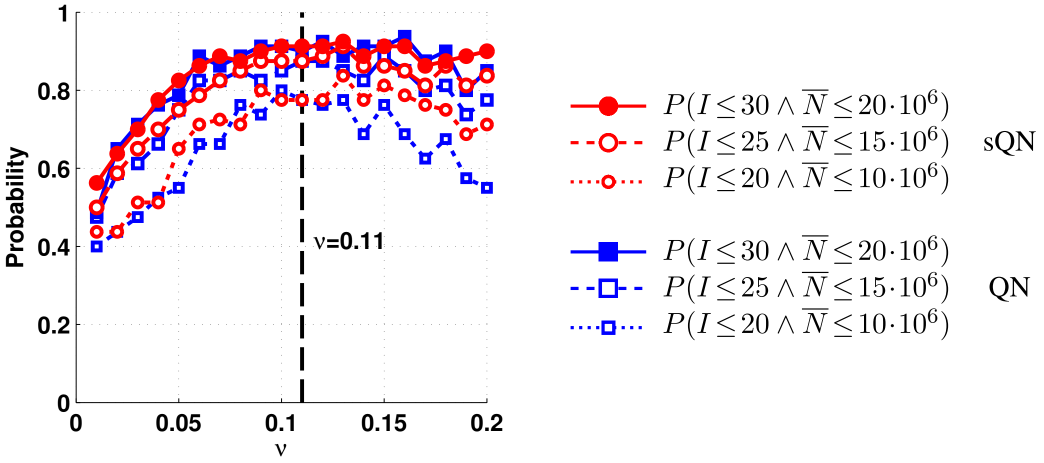


Figure 9.5.: Probability for successful convergence of JTOL algorithm.

N_{min}=$2\cdot 10^4$	N_{max}=10^6
ϵ_{conf}=0.005	I_{max}=50
Pat: LBP, RJ=0.18 UI	ν=0.11

Table 9.2.: Default JTOL algorithm settings.

9.2.2. Simulation Results

For the given PLL model two different JTOL analysis scenarios are investigated. First, a software based scenario uses the default simulator time resolution of 1fs, which yields R_{sim}=$3.3\cdot 10^5$ for the 3Gb/s interface. Second, a hardware based scenario reduces this value to R=512 and R=32, which allows to investigate the effect of limited precision in TIA based jitter measurements.

In tables 9.3 and 9.4 the performance of the JTOL algorithm is compared with respect to four different model parameter settings and the two sQN and QN tail fitting methods. In order to highlight the performance gain of sample size adaptation, both tail fitting methods are applied with a constant sample size N=N_{max} as well. These non-adaptive algorithmic versions are subsequently referred to as c-QN and c-sQN methods.

The overall sample size $\overline{N}$ is the sum of samples needed for JTOL analysis runs with 20 frequency points in an equally spaced logarithmic search grid of f_{SJ}=$[10^6, 10^8]$ MHz. Since most of the simulation time is spent to gather jitter distributions, this value directly reflects the computational effort for identifying a complete JTOL curve. In a hardware scenario it also relates to the overall measurement time. As to be expected, smallest $\overline{N}$ are obtained by the quickly converging sQN method. It is immediately followed by the QN method, which suffers from a larger error bias and spread, and thus, on average needs more samples to converge. If no sample size adaptation is performed (c-QN and c-sQN methods), the overall sample size $\overline{N}$ increases by a factor of 2-3.

$\overline{t_c}$ is the overall computation time on a 3GHz Intel Xeon CPU, consumed with tail fitting methods. It is thus a measure for the complexity of algorithms. In tables 9.3 and 9.4, tail fitting with QN is more than one order of magnitude faster than sQN. Further, as already described in section 4.1 figure 4.9, the computational demand linearly depends on R. Unfortunately the granularity of time measurements is 10ms, which impedes a reliable estimation of computational demand at small R (table 9.4) as well as for QN and c-QN methods.

$\overline{I_f}$ describes the average number of iterations per jitter frequency. In table 9.3 the $\overline{I_f}$ values of

Param.	**Alg.**	**Default**	$\mathbf{I_{cp}}$=10μA	$\mathbf{R_0}$=400Ω	$\mathbf{K_v}$=4GHz/V
$\overline{\mathbf{N}}$ [M]	**QN**	73.2	**77.4**	92.3	79.1
	sQN	**69.9**	80.9	**64.6**	**61.6**
	c-QN	185	316	167	271
	c-sQN	174	298	195	237
$\overline{\mathbf{t_c}}$ [s]	**QN**	**28.2**	**29.2**	**31.1**	**28.4**
	sQN	357	512	332	362
	c-QN	33.6	47.9	30.6	43.0
	c-sQN	544	950	626	696
$\overline{\mathbf{I_f}}$	QN	14.8	19.8	16.3	16.8
	sQN	12.9	21.3	14.0	15.7
	c-QN	9.3	15.8	**8.4**	13.6
	c-sQN	**8.7**	**14.9**	9.8	**11.9**

TABLE 9.3.: JTOL analysis results for R=R_{sim}=3.3·10^5.

Param.	**Alg.**	**R = 32**				**R = 512**			
		Def.	$\mathbf{I_{cp}}$ 10μA	$\mathbf{R_0}$ 400Ω	$\mathbf{K_v}$ 4GHz/V	**Def.**	$\mathbf{I_{cp}}$ 10μA	$\mathbf{R_0}$ 400Ω	$\mathbf{K_v}$ 4GHz/V
$\overline{\mathbf{N}}$ [M]	**QN**	89.2	**102.3**	146.3	86.1	84.4	93.3	**84.8**	84.9
	sQN	**88.9**	105.6	**94.9**	**80.2**	**67.0**	**65.0**	85.2	**65.2**
	c-QN	182.0	294.0	229.0	269.0	175.0	285.0	209.0	256.0
	c-sQN	193.0	301.0	211.0	247.0	187.0	284.0	221.0	240.0
$\overline{\mathbf{t_c}}$ [s]	**QN**	<0.04	<**0.01**	<**0.03**	<**0.01**	0.12	**0.10**	0.18	0.13
	sQN	0.32	0.39	0.39	0.28	2.30	3.11	2.73	2.71
	c-QN	<**0.02**	<0.03	<0.04	<0.02	<**0.05**	0.12	<**0.09**	**0.11**
	c-sQN	0.31	0.37	0.33	0.30	1.83	2.79	2.27	2.37
$\overline{\mathbf{I_f}}$	QN	13.8	21.2	17.7	16.6	13.7	20.5	15.6	17.2
	sQN	15.3	19.2	17.7	16.3	13.9	18.8	15.5	16.9
	c-QN	**9.1**	**14.7**	11.5	13.5	**8.8**	14.3	**10.5**	12.8
	c-sQN	9.7	15.1	**10.6**	**12.4**	9.4	**14.2**	11.1	**12.0**

TABLE 9.4.: JTOL analysis results for R=32 and R=512.

sQN are slightly better than those of QN, due to the faster convergence. Both c-QN and c-sQN methods with constant N yield the best results for $\overline{I_f}$, because also the error bias is constant (refer to figure 9.3(a)).

Table 9.4 shows the behavior of the JTOL algorithm, when carried out with a coarse time resolution or reduced number of bins R, as is the case for hardware measurements. Generally a slight increase of the number of overall samples $\overline{N}$ is observed, especially with sQN and QN methods. This is, because a coarse time resolution R leads to larger statistical variations of TJ estimates, obtained from extrapolated distribution tails. Thus, the JTOL algorithm on average also needs more samples in order to reach the convergence criterion.

Overall sample size $\overline{N}$ and average number of iterations $\overline{I_f}$ of both QN and sQN methods can also be investigated over varying R, as shown in figure 9.6. While $\overline{I_f}$ remains approximately constant over the complete analysis range, $\overline{N}$ in fact highlights a slight increase toward smaller R. This empirically confirms the assumption of a larger statistical variation of TJ estimates. However, at any R the JTOL algorithm correctly converges within the specified maximum number of iterations I_{max}=50, indicating that it can also be utilized for hardware measurements.

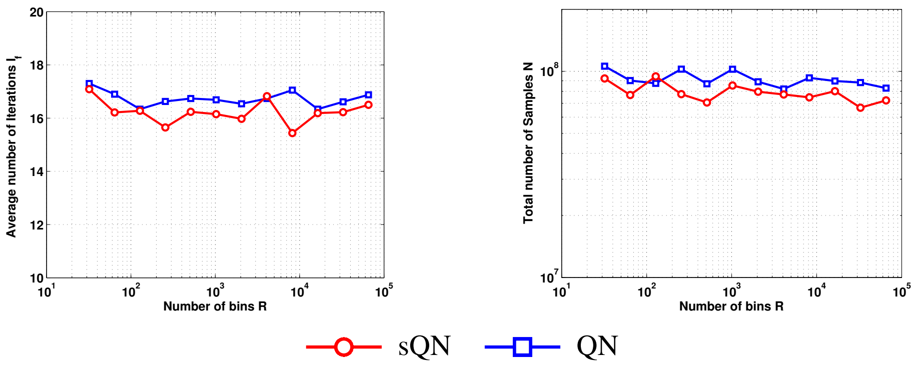
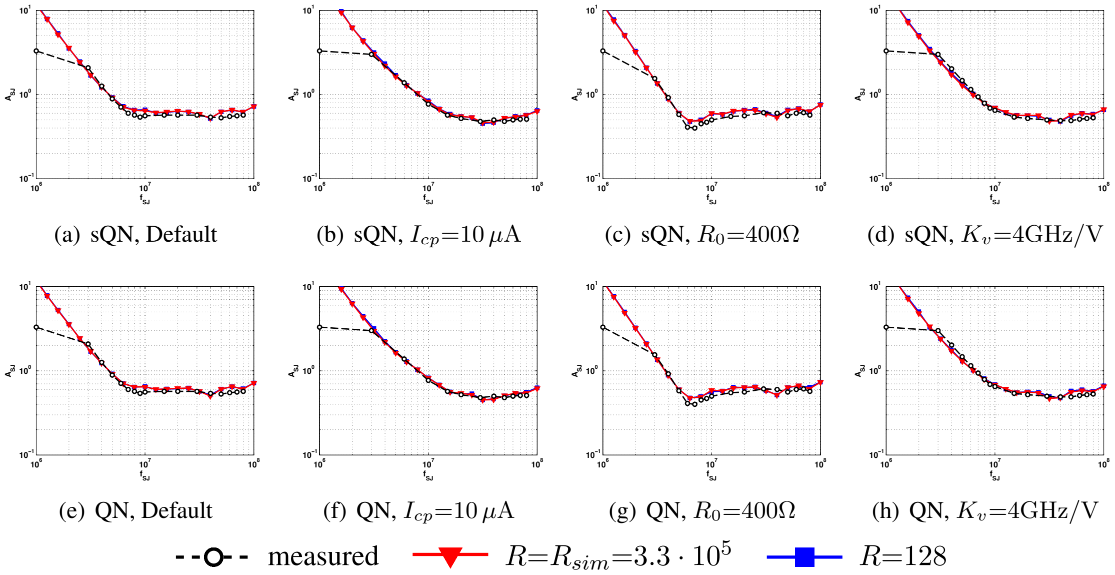


Figure 9.6.: a) Average number of iterations $\overline{I_f}$ and b) total sample size $\overline{N}$ over varying R.



(a) sQN, Default

(b) sQN, I_{cp}=10 μA

(c) sQN, R_0=400Ω

(d) sQN, K_v=4GHz/V

(e) QN, Default

(f) QN, I_{cp}=10 μA

(g) QN, R_0=400Ω

(h) QN, K_v=4GHz/V

Figure 9.7.: Simulated (tables 9.3 and 9.4) and measured JTOL curves at different model parameter configurations.

With the confidence level ϵ_{conf}=0.005, this is further achieved without visibly affecting the quality of obtained JTOL curves. In figure 9.7 the simulated JTOL curves for the sQN and QN methods at both R_{sim} and R=128 are plotted together with manually measured JTOL curves of the same PLL hardware structure. Note that the measurement device (SyntheSys BERT-Scope 7500A) has an amplitude limit of $A_{SJ,max} = 3.3\,\mathrm{UI}$, which is reached at f_{SJ}=$1\,\mathrm{MHz}$. Further, differences with simulated curves are mainly given by model inaccuracies and thus, do not reflect the performance of the JTOL algorithm. The hardware oriented model simulation with only R=128 bins already matches excellent with the R_{sim} high resolution scenario and can still handle the varying loop dynamics over the complete frequency range. The obtained results show, that the best jitter tolerance curve is achieved with the default model parameter settings. The S-ATA mask specification [118] demands a minimum tolerance of $A_{SJ} = 0.42\,\mathrm{UI}$ which is here clearly guaranteed.

If a complete JTOL measurement over twenty frequency points with the QN method typically requires a total of $\overline{N} \approx 100\mathrm{M}$ samples ($\approx 1.2\mathrm{h}$ of simulation time), the time consumed in a TIA based hardware measurement is $\overline{t_N} \approx 33\mathrm{ms}$ for a 3Gb/s interface. Together with $\overline{t_c} \approx 130\mathrm{ms}$ (R=512) for calculations and 1ms additional time buffer per iteration $\overline{t_I} \approx \overline{I_f} \cdot 20 \cdot 1\mathrm{ms} = 320\mathrm{ms}$

($\overline{I_f}\approx 16$), this gives an overall time consumption of

$$\overline{t_N} + \overline{t_c} + \overline{t_I} = 33\,\mathrm{ms} + 130\,\mathrm{ms} + 320\,\mathrm{ms} = 483\,\mathrm{ms}, \tag{9.11}$$

which is needed to identify the complete jitter tolerance curve over twenty frequency points.

9.3. Summary

A fast and accurate method for the identification of jitter tolerance curves of high-speed PLLs has been presented. An adaptive algorithm determines the unknown jitter amplitudes recursively and is optimized with respect to a small number of iterations and sample size. The algorithm realization started with the simple adaptive recursion in equation (9.1) and the associated derivation of a cost function given in equation (9.4). An algorithm for automatic sample size adaptation was realized and described with the flow graph in figure 9.2.

The basic idea is to observe jitter amplitudes over a subset of L last recursions, which allows to describe their statistical confidence interval in terms of a Gaussian t-statistic (equation (9.5)). As soon as this value falls below a predefined threshold, the algorithm adapts the sample size N of collected distributions accordingly. This adaptation process is controlled by the known error behavior of QN and sQN fitting methods. Therefore, the worst case distribution shapes of both methods were used to derive 4^{th} order polynomials, which approximate the extrapolation error as a function of sample size N (figure 9.3). The overall algorithm structure guarantees for a strictly monotonic increase of the adapted sample size until the maximum N_{max} is reached, together with the desired confidence level. The presented algorithm is repeated for every jitter frequency f_{SJ}.

As an application example, the proposed analysis method was applied to the PLL model from chapter 8. First, suitable analysis parameters were identified to optimize the algorithm behavior with respect to the given model. As an example, figure 9.4 highlights the varying loop dynamics of the PLL at different f_{SJ}. Default algorithm settings, such as the optimized learning rate parameter ν=0.11 were specified in table 9.2.

Finally, the analysis proceeded to the determination of jitter tolerance curves. Therefore, four algorithmic versions were compared, including the two fitting methods with (QN and sQN) and without (c-QN and c-sQN) sample size adaptation. Performance results were given in table 9.3 for simulators (R=R_{sim}=3.3·10^5), and in table 9.4 for simulated hardware measurements (R=512 and R=32). These include four different parameter settings with twenty f_{SJ} values each.

Results demonstrated, that the adaptive sample size adaptation of QN and sQN generally decreases the overall number of required jitter samples $\overline{N}$ by a factor of 2-3. This value directly reflects the simulation effort or the measurement time in a hardware system. The overall computation time $\overline{t_c}$ is the smallest with QN and c-QN methods and outperforms sQN and c-sQN methods for more than one order of magnitude. Finally, the average number of recursions per jitter frequency $\overline{I_f}$ is the smallest without sample size adaptation (c-QN and c-sQN methods). This is due to the constant error bias which does not influence the adaptation process, as is the case with QN and sQN methods. The same performance characteristics of $\overline{N}$, $\overline{t_c}$, and $\overline{I_f}$ can also be observed with simulated hardware scenarios, besides a highly reduced computational demand. For each test case in figure 9.6, the algorithm was always able to converge within a maximum number of fifty recursions over varying R, thus indicating that the proposed algorithm can also be applied to hardware measurements.

As a final result, in figure 9.7 the obtained jitter tolerance curves were compared against real jitter measurements from the test structure and showed an excellent matching. The simulation of such jitter tolerance curves is particularly useful to optimize a CPLL design with respect to its robustness against input jitter. It also allows for the verification of imposed specification requirements such as a jitter tolerance mask. Contents of this chapter have partly been published in [C6].

10. An FPGA based Diagnostic Tool for Jitter Measurement and Optimization

In order to highlight the practical aspects of tail fitting methods, in this chapter an embedded jitter measurement system is presented, which acts as a diagnostic tool for serial high-speed interfaces. The underlying idea is to combine a real BIJM system with the previously described sQN and QN tail fitting methods. This is to confirm the theory of hardware design aspects from chapter 5 and to prove correctness of the derived equations and empirical relations. Further, extrapolated tails yield the TJ timing budget, which allows for judging the quality of transmission lines, PLLs or transceiver structures as system under test (SUT). The resulting diagnostic tool is thus able to optimize and configure an SUT without the use of additional instrumentation devices.

Many BIJM topologies and embedded systems [16,18,57,64,65,68,73,79] have been designed for production tests and on-chip diagnostics, and some of them have also been used for jitter optimizations [87, 132]. However, the combined use of BIJMs with tail fitting methods has not been considered so far. This chapter especially points out the benefit of applied tail fitting methods to estimate the TJ budget, which forms a direct quality measure for the impact of timing jitter on system performance. The target platform is a Virtex-5 FPGA [137] on an ML507 high-speed evaluation board [138]. It produces a 3Gb/s serial reference signal and retrieves timing jitter information from the SUT.

In the following sections first the diagnostic principle is introduced. Then the implemented FPGA logic together with the analysis software is described, and parameter optimizations with different diagnostic scenarios and test cases are carried out. Finally the observed tail fitting error is compared with expected worst case errors from the theory in chapter 5. A brief summary is given at the end.

10.1. Measurement Principle and Implementation

The fundamentals of jitter diagnostics are given to solidify the understanding of the developed analysis tool. The FPGA implementation is described subsequently and represents a direct realization of the underlying principle.

10.1.1. Diagnostic Principle

The jitter analysis of serial high-speed signals requires an accurate measurement of IO jitter, which has already been described in the introductory part of chapter 5 (also refer to figure 5.1). The fundamental scheme for jitter diagnostics is very similar to a BIJM system and depicted in figure 10.1. A system under test basically degrades the signal integrity of a generated data stream by producing jitter. It can be a simple transmission line, a PLL/CDR or even a complete transceiver structure. The fundamental difference to BIJMs is the internal reference clock which acts as a common basis for transmit buffer and jitter measurement circuit. In this way, external noise sources are excluded and the quality of jitter measurements is given by the minimum inherent jitter of the reference clock.

The BIJM principle with the adjustable delay element, phase detector (PD) and counter has already been described in chapter 5, and allows for collecting jitter distributions over successive

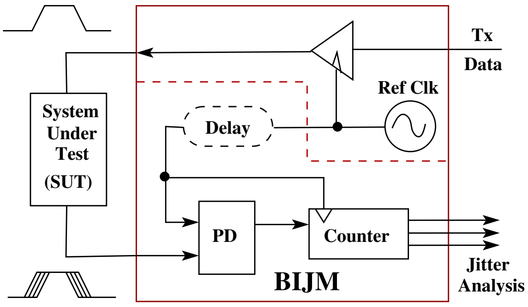


Figure 10.1.: Basic principle for jitter diagnostics.

measurement runs of varying delay. The smallest delay step defines the number of bins R in a UI, while the PD can also be replaced by a simple D flip-flop if the input data stream is known. In this case the recovered data is first compared against the expected transmit data, which then allows for error counting.

If a jitter distribution is measured down to a BER level of $1/N$ over R delay steps, the overall time consumed is

$$t_t[\text{UI}] = N \cdot R. \tag{10.1}$$

Although test time is rather uncritical, also in diagnostic applications the measured BER depth $1/N$ poses a fundamental time limit. In order to estimate the TJ budget at the target BER$=10^{-12}$, again tail fitting methods have to be applied. Using the sQN and QN methods from chapters 4 and 6 allows to parameterize the Gaussian tails (μ, σ, A) of a jitter distribution, and thus, to estimate the extrapolated TJ peak-to-peak value TJ_{pp}

$$\text{TJ}_{pp} = t_R - t_L \tag{10.2}$$

$$t_{L(R)} = \mu_{L(R)} - \sigma_{L(R)} \cdot Q(10^{-12}/A_{L(R)}) \tag{10.3}$$

with the Q-function defined in equation (3.10). The obtained TJ_{pp} estimates are subsequently used for jitter diagnostics and parameter optimization.

10.1.2. Implementation

Subsequently a 3Gb/s jitter diagnosis system is implemented using an FPGA. The measurement is realized using a Xilinx Virtex-5 FX70T on an ML507 high-speed evaluation board [138], which is controlled by a MATLAB program running on a remote desktop computer. The general FPGA block scheme is given in figure 10.2. In order to collect jitter distributions, a dedicated high-speed transceiver (GTX) [137] is combined with a BER test (BERT) logic and controlled by an instantiated microprocessor (MP). Measurement results are then transferred to the remote computer where the tail fitting methods are applied. The three main blocks of the FPGA logic are subsequently described in more detail.

High-Speed Transceiver

The Xilinx GTX is used for the parallel to serial and serial to parallel data conversion, and is needed to lower the data rate for use with the internal, custom FPGA logic. The 3Gb/s data stream is converted from or into a 20bit word along with a 150MHz clock signal to mark the beginning of a new word. The transceiver is driven in lock-to-reference mode, meaning that the PLL for data recovery (Rx-PLL) is locked to the same reference frequency of the transmitter buffer (Tx-PLL),

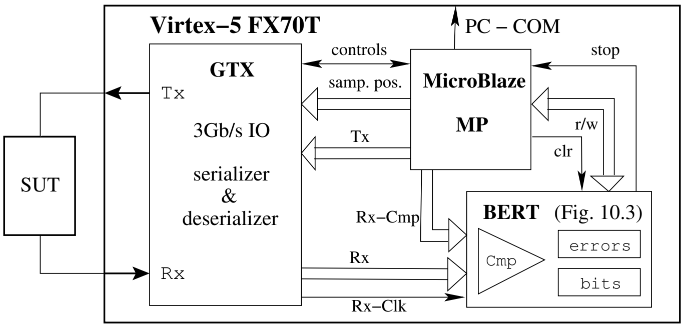


Figure 10.2.: Block scheme for the FPGA based 3Gb/s jitter measurement system.

in order to realize the diagnostic principle from figure 10.1. A complete jitter distribution can be measured by adjusting the data sampling point of the receiver PLL over R=128 time steps. This is done by using the Dynamic Reconfiguration Port (DRP) [137] of the GTX, which is controlled by the MP. The transmit signal is also a 20bit word given by the MP.

BER Tester

The custom made BERT logic consists of two units: a bit counter and an error counter (figure 10.3). The former keeps track of the sum of total bits by counting the number of clock cycles from the Rx clock. The latter compares the Rx data to the data pattern set by the MP. The number of errors is then added every Rx clock cycle. Each of the two counters is compared to a preset maximum value. Once either number is reached, a generated enable signal simultaneously stops both counters. This way the measurement can be stopped by either the number of bits or errors, which is especially advantageous during the synchronization phase (MP). Once a measurement at one sampling position is completed the MP reads the values held by the counters, resets them and increases the delay to start the next measurement run.

Microprocessor

The MicroBlaze Processor controls the measurement sequence and realizes the serial interface (RS232) between the ML507 board and a remote computer. The software flow graph in figure 10.4 shows the steps of a measurement run, needed for collecting a complete jitter distribution. The data pattern to be transmitted along with the number of bits N per sampling position are first

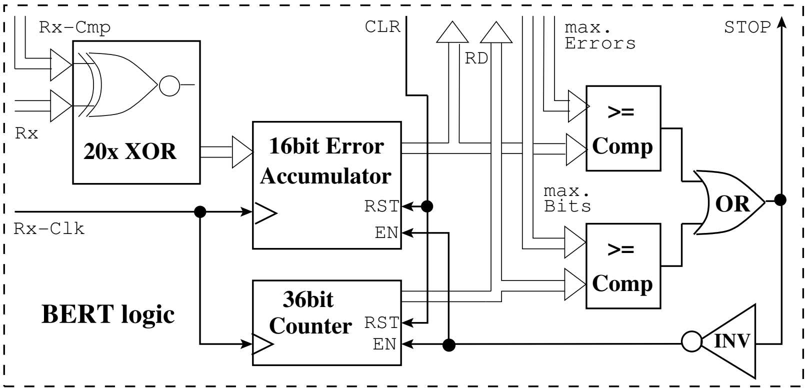


Figure 10.3.: Realization of the BERT logic.

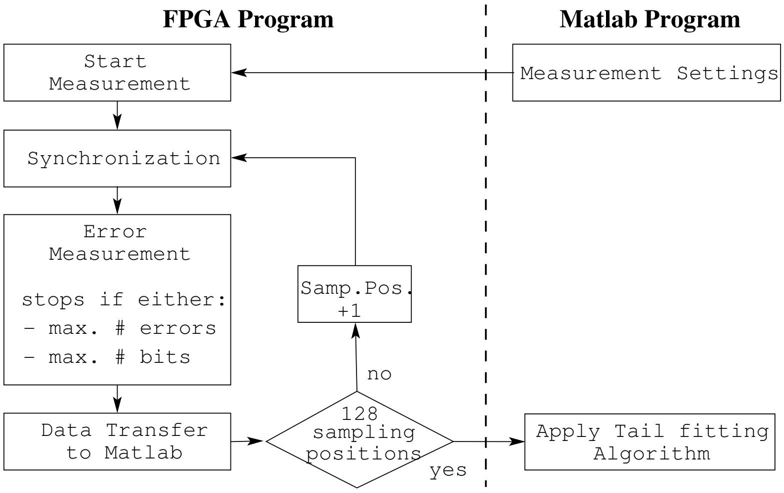


FIGURE 10.4.: Flow graph of BER measurement and analysis.

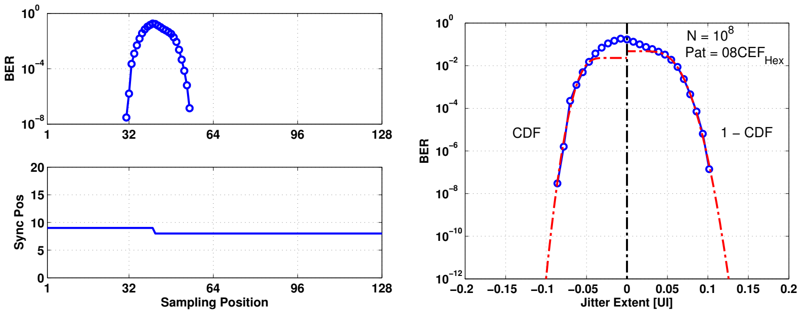
entered using MATLAB. The MP then starts with short BER measurements at the first delay step in synchronization mode. This phase is needed to correctly match the 20bit Rx and Rx-Cmp words, due to an unknown channel delay. During the synchronization, the Rx word is compared with a circular shifted version of the original Tx data pattern, where the BERT collects only 3200 words or 64k bits to evaluate the number of errors. This Rx-Cmp pattern is successively rotated over all 20 positions. The lowest number of errors thus indicates the synchronization pattern to which the Rx data pattern needs to be compared. The synchronization phase is carried out very quickly, consuming $<0.5\text{ms}$, and is repeated at each of the R sampling positions or delays. As a key advantage, this allows for measuring both distribution tails in a single measurement run (also see figure 10.5).

After synchronization, the MP resets both counters, and starts the long BER measurement with sample size N and a maximum number of 32k errors. Once either counter reaches the maximum, the MP reads their values and calculates the BER, passes it on to MATLAB, and increases the sampling delay to restart the measurement process. Once the BER at every delay step has been measured, MATLAB runs the program containing the sQN and QN fitting methods, which finally yields the desired TJ_{pp} value at the target BER=10^{-12}.

10.2. Jitter Measurements and Optimization

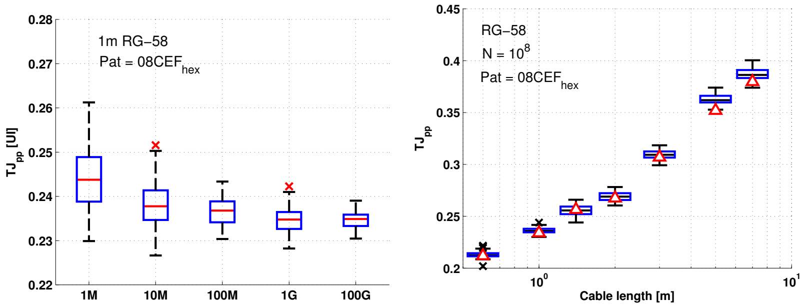
In this section the presented diagnostic tool is applied to various test cases and optimization scenarios. To demonstrate the operating principle, it is first used to measure the jitter distribution of a simple 1m RG-58 coaxial cable. In order to yield substantial inter-symbol interference (ISI) on the transmission line, a PRBS4-like 20bit pattern 08CEF_{hex} is sent. The obtained measurement result is shown in figure 10.5(a). Using the selected data pattern, only one synchronization value out of the 20 positions produces the correct, minimum number of errors. Since it is independently determined at each sampling position prior to the effective BER measurement, both distribution tails are directly obtained from a single measurement run. This is especially observed at the distribution center, where the synchronization value is decremented by one. Here, the measured BER corresponds to the right tail behavior or reverse cumulative distribution function $1{-}\text{CDF}$. Otherwise, with a constant synchronization position BER values would saturate at $8/20{=}0.4$, which is equal to the transition density of the selected pattern.

The applied sQN method accurately extrapolates distributions down to the target BER=10^{-12},



(a) (b)

Figure 10.5.: Example for a measured jitter distribution (left) and sQN fitted tails (right). $N=10^8$, Pat$=$08CEF$_{hex}$, 1m RG-58 coaxial cable.



(a) (b)

Figure 10.6.: K=100 evaluations of TJ$_{pp}$ estimates over a) varying sample size N, and b) RG-58 cable length. Triangles mark medians of exact measurements ($N=10^{11}$).

as can be seen from the example in figure 10.6(a) where K=100 distributions of the 1m RG-58 cable are fitted over varying sample size N. As expected, both bias and statistical spread decrease with larger N because the extrapolation range becomes smaller. Note that the error bias is always positive and thus, yields pessimistic TJ$_{pp}$ estimates, which is due to the beneficial extrapolation property described in section 4.2.1.

In figure 10.6(b) K=100 estimates of TJ$_{pp}$ are statistically evaluated over varying cable length, to estimate the influence of ISI on total jitter. The triangles mark median values of N=10^{11} measurements to approximate the ideal extrapolation result. With 5m and 7m cables, the signal is additionally degraded by noise effects, which cause a larger error bias in figure 10.6(b).

With N=10^8 samples a measurement run takes approximately 2s, which is sufficiently fast for jitter diagnosis. This value is clearly below the expected test time of

$$t_t = (N \cdot R)/3{\cdot}10^9\,\text{s} = 4.27\,\text{s}. \tag{10.4}$$

This is, because bins with high error rates quickly count bit errors and thus, the maximum number is reached very fast. Additionally, the FPGA software first locates a jitter distribution within the 128 sampling positions, so that measurements are only made in the region of interest.

The reference clock of the GTX transceiver includes a certain amount of inherent jitter, which forms the fundamental minimum for the jitter measurement system. It can be identified using the internal loopback mode of the GTX, and yields typical fitting results for right bathtub tails as shown in figure 10.7. Due to the clean data signal, tails become very steep and thus, contain only few data points. Especially at N=$\{10^6, 10^7\}$ a correct tail extrapolation is therefore not possible

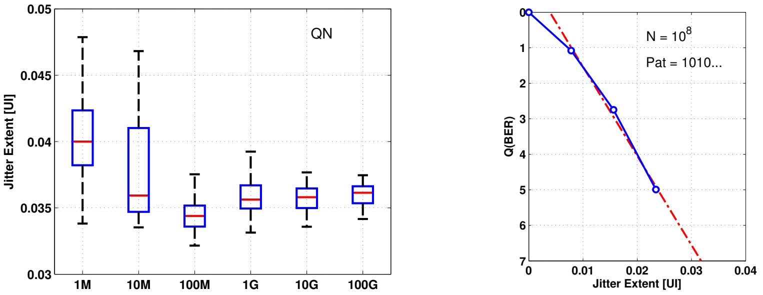


Figure 10.7.: Estimated positive jitter at internal loopback mode, using QN as fitting method. Especially for smallest sample sizes $N=\{10^6, 10^7\}$ a correct tail extrapolation fails because bathtub curves (right) become too steep.

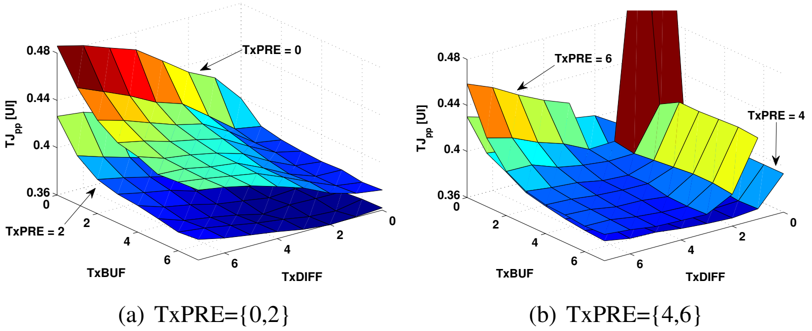
anymore. At $N=10^8$ further, the median is smaller than the approximated true median at $N=10^{11}$, which is due to the error oscillations at lowest bin density, as described in section 5.4.2. Observing multiple measurement runs, the linear course in Q-domain is given until $Q_{up}=Q(\Delta P_t/N)\approx 1$, while $\sigma_{t,10^{-11}}\approx 3.7\cdot 10^{-3}$. These values can be inserted into equation (5.20) for calculating the required minimum sample size, which gives $N_{min}>10^7$. This is also confirmed by the results in figure 10.7. The timing budget of GTX inherent jitter is approximated with

$$\mathrm{TJ}_{pp,\mathrm{loopback},N=10^{11}} = \mathrm{TJ}_{pp,min} \approx 0.149\,\mathrm{UI} \tag{10.5}$$

Instead of testing the quality of transmission channels, the jitter diagnosis system can also optimize the parameter configuration of a complete high-speed interface as SUT. To demonstrate this concept, in an example the GTX internal Rx and Tx structures are optimized together with a $5\mathrm{m}$ RG-58 coaxial cable.

First, only the Tx buffer configuration is optimized. Therefore, the GTX provides three ports which control differential output swing (TxDIFF), pre-driver swing (TxBUF) and pre-emphasis (TxPRE) [137]. Each of these ports has eight different gain settings, thus with a total of 512 parameter combinations. Using the diagnostic tool, the TJ_{pp} value acts as fitness measure for parameter optimization, and yields the surfaces in figure 10.8. The selected sample size for measurements is $N=10^8$. Although median values from a statistical evaluation with $K=25$ measurements are displayed, a single evaluation is already sufficient for optimizations, since the obtained standard deviation of estimates is always $\sigma_{TJ_{pp,est}}<0.008\,\mathrm{UI}$.

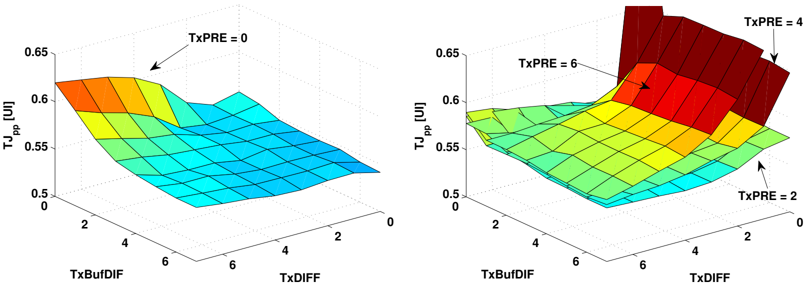
Second, the Rx-PLL is used in normal lock-to-data mode as is the common case for high-speed data recovery. Without the clean reference clock, the Rx-PLL now suffers from a significant



(a) TxPRE={0,2}

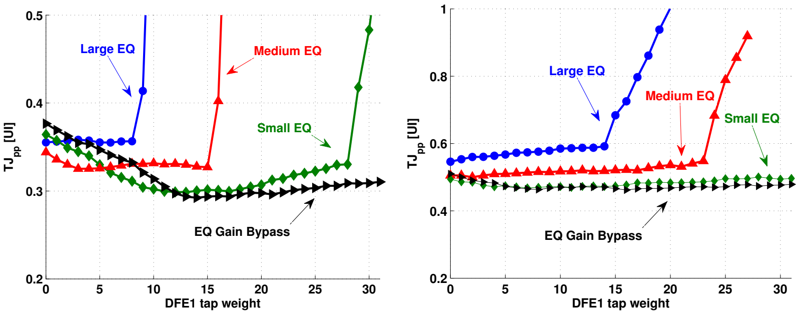
(b) TxPRE={4,6}

Figure 10.8.: TJ_{pp} surfaces for Tx buffer optimization. $N=10^8$, $K=25$.



(a) TxPRE=0 (b) TxPRE={2,4,6}

FIGURE 10.9.: TJ$_{pp}$ surfaces for Tx buffer optimization, lock-to-data mode. N=10^8, K=25.



(a) lock-to-reference (b) lock-to-data

FIGURE 10.10.: TJ$_{pp}$ optimization of Rx structures, including four different EQ settings and a single DFE tap weight. N=10^8, K=25.

amount of additional jitter caused by the recovered signal clock, as can also be seen in figure 10.9.

The two optimized Tx parameter configurations finally yield

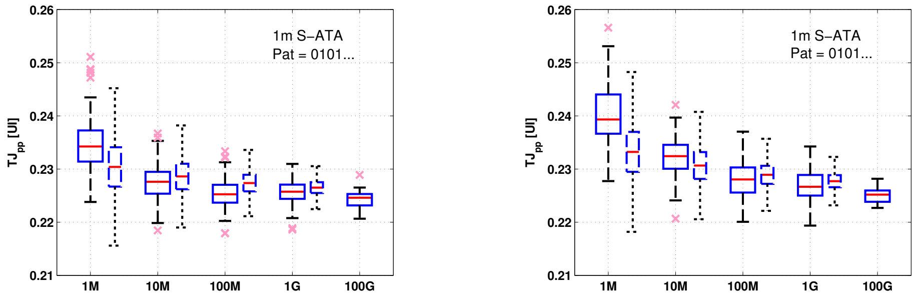
$$\begin{array}{lll} \text{DIFF=0, BUF=5, PRE=1} & \text{TJ}_{pp}\text{=0.36 UI} & \text{(lock-to-reference)} \\ \text{DIFF=0, BUF=5, PRE=0} & \text{TJ}_{pp}\text{=0.55 UI} & \text{(lock-to-data)} \end{array}$$

indicating, that there is no visible benefit when using signal pre-emphasis in lock-to-data mode.

Equivalently, also the Rx structure can be optimized. The GTX includes a built-in gain equalizer (EQ) and a decision feedback equalizer (DFE) with adjustable tap weights. The EQ affords either low, medium or large high-frequency boost or bypass with gain factor. In figure 10.10 these four EQ settings are used together with the first DFE tap DFE1=[0...31] for parameter optimization. The plotted curves indicate the EQ settings over varying DFE1 tap value. With EQ gain bypass and the DFE, a significant reduction of the total jitter down to TJ$_{pp}$ $<$ 0.3 UI is possible in lock-to-reference mode (left). Again, in lock-to-data mode (right) the TJ budget is significantly larger and the DFE also achieves only a slight improvement.

10.3. Analysis of Extrapolation Error

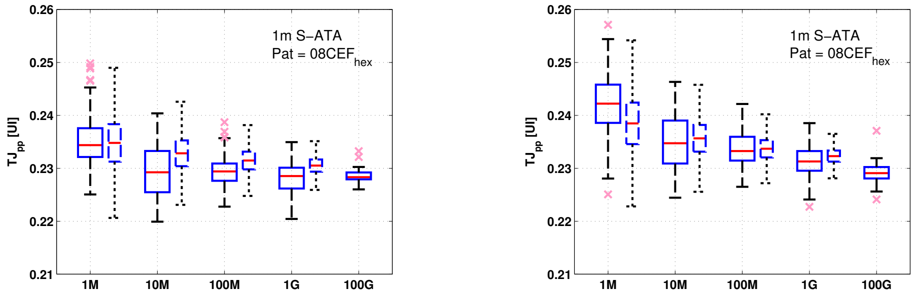
In this section the extrapolation error of both sQN and QN tail fitting methods is investigated when applied to practical jitter measurements. Therefore again different test cables are used as SUTs for the jitter diagnosis system. For both methods a minimum fitting region of ΔP_t=10^3 is selected, since the DJ shape of measured distributions is basically unknown and may change over varying



(a) sQN

(b) QN

Figure 10.11.: Estimated TJ_{pp} values of a $1\mathrm{m}$ S-ATA crossover cable (solid boxes), $N{=}\{10^6,\ldots,10^9\}$. The $N{=}10^{11}$ measurement allows for worst case error estimation (equation (5.30), tables of coefficients 5.4 and 6.2) by assuming sinusoidal (PLL noise dominated) jitter (dashed boxes). $\Delta P_t{=}10^3$.



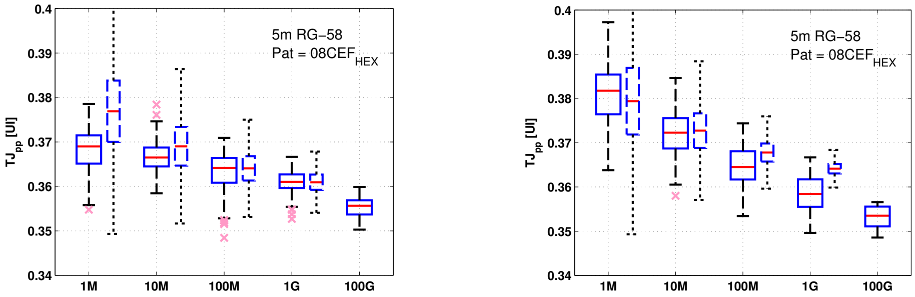
(a) sQN

(b) QN

Figure 10.12.: Estimated TJ_{pp} values of a $1\mathrm{m}$ S-ATA crossover cable (solid boxes), $N{=}\{10^6,\ldots,10^9\}$. Worst case error estimation with assumed uniform (ISI dominated) jitter (dashed boxes). $\Delta P_t{=}10^3$.

cable length. With short cables jitter is expected to be dominated by ISI, while for longer cables additional noise effects such as couplings, reflections or crosstalk will also degrade the signal.

Both QN and sQN fitting methods are first applied to jitter distributions of a standard $1\mathrm{m}$ S-ATA crossover cable and a clock-like data pattern. A statistical evaluation of TJ_{pp} values over $K{=}100$ bathtub measurements and varying sample size N yields the solid boxes in figure 10.11. As expected, with increasing N the obtained medians converge toward the true value while statistical spread becomes smaller. Thus, a good estimate for the true TJ budget is obtained from measurements at $N{=}10^{11}$ ($K_{10^{11}}{=}25$). The observed statistical behavior of TJ_{pp} estimates can be compared with expected worst case errors using the derived empirical relation (5.30). With $\sigma_{RJ} \approx \sigma_{t,10^{-11}} \approx 0.012$, these errors can be calculated by choosing the correct DJ type. For a clock-like pattern, the signal is only affected by the PLL jitter of the reference clock and the transmission channel cannot produce additional ISI. Thus, sinusoidal DJ is assumed (also refer to section 3.2.2). The obtained statistical error (dashed boxes) must be generally larger than the measurements, which is the case and confirms validity of the worst case error estimation. Only at $N{=}10^6$, fitted values indicate a larger error bias jump, meaning that the underlying DJ shape behaves different from the assumed sinusoidal distribution. Thus, the given sample size is not anymore sufficient for correct tail extrapolation.



(a) sQN (b) QN

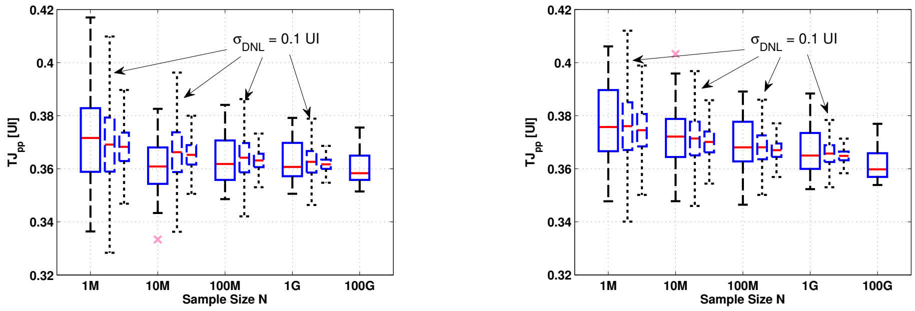
FIGURE 10.13.: Estimated TJ_{pp} of a $5\mathrm{m}$ RG-58 coaxial cable (solid boxes), $N{=}\{10^6,\ldots,10^9\}$. Worst case errors (dashed boxes) assume quadratic curve DJ (combined ISI and unknown noise couplings, equation (5.30), tables 5.4 and 6.2). $\Delta P_t{=}10^3$.

The second example in figure 10.12 uses a PRBS4-like 08CEF$_{hex}$ pattern for ISI dominated jitter, instead of the clock-like signal. The expected worst case error assumes uniform DJ and is again larger than the measured error.

A third example in figure 10.13 shows the influence of additional noise effects, as appear with long transmission channels. Therefore, jitter measurements are now carried out with a $5\mathrm{m}$ RG-58 coaxial cable and statistical evaluations are again performed using sQN and QN methods. Also the PRBS4-like 08CEF$_{hex}$ pattern is used to produce ISI. The worst case error behavior can now be described as a combined effect of ISI and unknown noise couplings, using quadratic curve DJ. Validity of this assumption is confirmed with the obtained extrapolation results, which remain within the expected error limits. At $N{=}10^9$ the worst case error yields somewhat pessimistic medians and rather tends to underestimate the measured statistical spread. Note that this sample size is also located beyond the original range of regression planes $N{=}[5\cdot 10^5, 10^8]$ for the tables of coefficients 5.4 and 6.2, which was considered as more suitable for this use case.

The fourth example investigates the influence of process variations on fitted tails. Therefore the Rx-PLL of the GTX is used in lock-to-data mode which is the common scenario for high-speed signal recovery. This operating mode causes the Rx-PLL to lose lock if the data sampling position is shifted too close to the center of a jitter distribution and thus, only allows for correct BER measurements in the lower tail region of bathtub curves. However, this operating mode also produces a random phase of the recovered clock as soon as the lock state is reached. Hence, a DNL error of the BIJM system also affects the measurement result randomly. Over multiple bathtub measurements it is thus observed as the random effect modeled in section 5.4.3. As test channel, the $1\mathrm{m}$ S-ATA crossover cable is again utilized. Figure 10.14 shows the statistical spread of TJ_{pp} estimates over varying N with the Rx-PLL in lock-to-data mode (solid boxes). Using equation (5.30) (uniform DJ) without the DNL error effect, the right dashed boxes are obtained. Due to the present DNL error, especially the statistical spread is underestimated. This can be solved by using equation (5.38) together with the tables of coefficients 5.6 and 6.3, and assuming an additional DNL error of $\sigma_{DNL}{=}0.1$ UI. The obtained results are the dashed boxes in the center.

As a final analysis, in figure 10.15 the influence of varying channel length (RG-58 coaxial cable) on the measured TJ timing budget is shown. The solid lines correspond to median values of estimated TJ_{pp} values with $N{=}10^7$ samples and the test pattern 08CEF$_{hex}$. Black crosses mark approximated true medians obtained from the $N{=}10^{11}$ measurements, and dotted lines highlight the expected worst case error under a uniform DJ assumption. As expected, the TJ budget generally increases with larger cable length and thus indicates the increase of ISI. For lock-to-reference mode, the uniform DJ assumption is violated at large cable lengths due to additional noise influ-



(a) sQN

(b) QN

Figure 10.14.: Estimated TJ$_{pp}$ of a $1\mathrm{m}$ S-ATA crossover cable with Rx-PLL in lock-to-data mode (solid left boxes). Worst case error estimations assume uniform DJ using (5.30) (right dashed boxes) and (5.38) (center dashed boxes) which additionally includes the effect of process variations σ_{DNL}=0.1UI. ΔP_t=10^2.

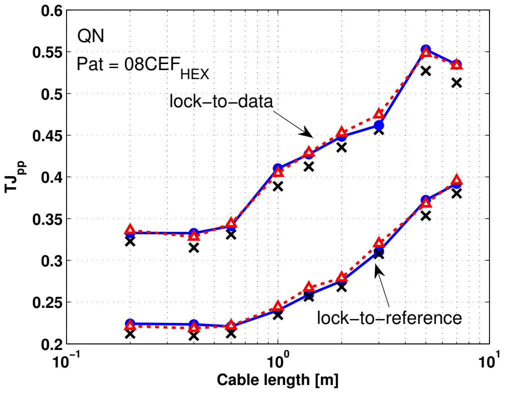


Figure 10.15.: TJ$_{pp}$ medians of tail fitted estimates (N=10^7, circles), exact measurements (N=10^{11}, crosses) and expected worst case uniform DJ (triangles) over varying cable length. ΔP_t=10^3.

ences. The same effect is also observed for very small cable lengths due to steep bathtub tails, as discussed in the previous section with figure 10.7. In lock-to-data mode, overall TJ increases significantly. The included Rx-PLL now superimposes its own jitter with unknown characteristic and thus, violates the uniform DJ assumption as well. Also, measurement results suffer from larger variations, as they are generally more affected by cable induced jitter.

10.4. Summary

An FPGA based diagnostic tool for total jitter estimation in high-speed interfaces has been developed. It allows for quantifying the timing budget caused by a system under test and thus, can be used for testing the quality of transmission channels or optimizing the parameter configuration of interface structures. Jitter measurements at 3Gb/s with a sample size of N=10^8 and R=128 delay steps in a unit interval require approximately 2s of test time.

After an initial demonstration of the jitter measurement and sQN fitting principle in figure 10.5, the TJ$_{pp}$ values of different RG-58 cables were determined in figure 10.6 using the developed diagnostic tool. The inherent jitter of the reference clock (TJ$_{pp}$≈0.15 UI) was also quantified with loopback measurements and represented in figure 10.7. As an example for parameter optimization with the presented diagnostic tool, the built-in Rx equalizers and Tx buffers of the FPGA were configured together with a $5\mathrm{m}$ RG-58 coaxial cable. This allowed to decrease the TJ timing budget down to $\mathrm{TJ}_{pp}<0.3\mathrm{UI}$.

Further, the extrapolation error of sQN and QN fitting methods was investigated and compared against theoretical worst case errors from sections 5.4 and 6.5. In this context the sinusoidal, uniform and quadratic curve DJ shapes were experimentally confirmed to be well suited for pure clock jitter, ISI, and ISI plus external noise affected channels (figures 10.11, 10.12 and 10.13). When the receiver PLL was operated in normal lock-to-data mode, the additional effect of random process variations could be made visible, which also allowed to apply the DNL error model in figure 10.14. As a final analysis, in figure 10.15 the influence of varying RG-58 coaxial cable length on the measured TJ timing budget was shown, together with the predicted worst case errors under an ISI dominated DJ assumption. Results showed, that this condition is not fulfilled for large cable lengths due to additional noise influences, as well as for very small cable lengths due to the system limitations. Also in lock-to-data mode the overall TJ increases and changes the observed DJ characteristic.

Parts of this chapter have also been published in [C7,C8].

11. Conclusion

This work concludes with an overall summary and overview to the key results achieved throughout the thesis. A brief outlook to future directions is given at the end.

11.1. Results Summary

In this thesis, first the scaled Q-normalization (sQN) method for jitter and BER analysis was presented and realized in chapter 4. It is based on the Gaussian quantile function (equation (3.10)), which linearizes the tails of jitter distributions and thus, allows for Gaussian tail fitting and extrapolation. The Q-function was embedded into an efficient optimization scheme where a simple recursion achieves a very fast exploration of the three-dimensional search space. With this recursion, the sQN method automatically determines the best suited tail part for fitting. Thus, it represents a clear improvement to other methods where the tail part is predefined in a conservative manner or must be identified using an additional algorithm.

The extrapolation error of the basic sQN principle has been investigated in section 4.2, where the major causes for a degraded performance have been identified as small sample sizes N, worst case combinations of RJ and DJ, as well as Gaussian-like DJ shapes. However, the resulting extrapolation error is always positive biased and thus pessimistic, which is a further beneficial property of algorithms based on quantile normalization in general.

From the basic sQN principle, two optimized algorithmic scenarios have been derived which allow for improved error behavior and robustness. The $\hat{c}_{1.2}$ scenario in section 4.3 recommends a minimum probability interval $\Delta P_t{\geq}10^2$ for outermost tail selection. This parameter avoids outliers caused by statistical tail variations. Further, the scenario combines fitness measures based on both regression length and error to achieve an optimized error behavior. Another algorithmic scenario, $Q_{th,c}$, is based on a constant threshold Q_{min} in scaled Q-domain and thus, defines the Gaussian tail region in terms of standard deviations beside the model mean. This representation form allows for a flexible tail choice. As a trade-off, $Q_{min}{=}-1.2$ has been identified to achieve acceptable accuracy with reduced risk for outlier occurrence.

Both algorithmic approaches improve the estimation performance compared to the basic sQN principle. For example, with a typical uniform DJ, $N{=}10^6$ and worst case test distributions ($\sigma_{RJ}/A_{DJ}{=}1/4$), the error bias is still $<2\%$ with an overall error $<3\%$ in more than 97.5% of the cases (confidence level $a{>}0.95$). The performance of the $\hat{c}_{1.2}$ scenario is equivalent to $Q_{th,c}$ for $N{\geq}10^6$, and outperforms it at smaller sample sizes. This is due to a larger error variation of $Q_{th,c}$ when a fit is performed at the outermost tail part.

In chapter 5, hardware design aspects for the sQN method were investigated. The basic idea was to highlight properties of the proposed method when used together with test equipment, diagnostic tools or built-in jitter measurement (BIJM) systems. Unlike simulations, these systems introduce a discretization effect, which divides a distribution into R time intervals or bins per UI. As key parameters for a system design, both the sample size N as well as the discrete number of bins R were shown to cause fundamental limiting effects with respect to analyzable test distributions. In order to correctly fit distribution tails, hence, equations were derived that specify minimum requirements for tails.

First the tail parameters of fitted test distributions were characterized in section 5.1 using the polynomial equations (5.3) and (5.5). They allow for changing forth and back between the variables before (σ_{RJ}, A_{DJ}) and the obtained tail parameters after (A_t, σ_t, μ_t) distribution synthesis. The coefficients in table 5.1 and equation (5.3) specify tail amplitudes A_t obtained with the sQN fitting method. The two parameters σ_t and μ_t describe minimum requirements for sQN tail fitting, and are also valid for the conventional Q-normalization (QN) method described in chapter 6.

Requirements with respect to minimum tail amplitudes $A_{t,min}$ were investigated in section 5.2. For the $\hat{c}_{1.2}$ algorithm a conservative threshold with N and the design parameter ΔP_t were given in equation (5.10). For the $Q_{th,c}$ algorithm instead, equation (5.13) led to an exact solution, and further highlighted the missing link to the $\hat{c}_{1.2}$ algorithm based on ΔP_t.

Limitations introduced by the discrete time resolution with R number of bins per UI were investigated in section 5.3. This problem can also be represented in terms of a minimum analyzable standard deviation $\sigma_{t,min}$ of Gaussian tails. With equation (5.20) for $\hat{c}_{1.2}$ and equation (5.23) for the $Q_{th,c}$ algorithmic scenario, the minimum value $\sigma_{t,min}$ is determined which can be fitted correctly by each of the algorithms. Selection charts that aid in identifying the design parameter ΔP_t in figure 5.9 ($\hat{c}_{1.2}$) or both Q_{min} and ΔP_t in figure 5.11 ($Q_{th,c}$) were provided as well. Note that the parameters in both charts must also fulfill the previously mentioned requirements with respect to the minimum tail amplitude (equation (5.10) or (5.13)), and outlier suppression with $\Delta P_t \geq 10^2$ chosen as large as possible.

In section 5.4 the combined influence of sample size and time discretization on the extrapolation error of the sQN method was quantified. Therefore, the empirical relation (5.30) was derived together with the tables of coefficients 5.4 ($\hat{c}_{1.2}$) and 5.5 ($Q_{th,c}$). It describes bias, spread and their combined influence on extrapolation error. The empirical relation is given in terms of a two-dimensional function of sample size N and the variable product $\sigma_{RJ,min} \cdot R$. This product is a measure for the bin density along a distribution tail. The empirical relation aids a designer in finding an optimum performance trade-off between the required accuracy of a BIJM and the hardware expense in terms of key parameters N and R. The obtained results clearly highlight a better performance for the $\hat{c}_{1.2}$ algorithm.

The additional effect of process variations in a jitter measurement system was investigated in section 5.4.3. It is a typical result of timing mismatches inside the measurement system, and was modeled as differential non-linearity (DNL) error with standard deviation σ_{DNL}. As a resulting effect, the statistical spread of extrapolated tails increases significantly and thus, becomes the major contributor to overall error. Thus, a well suited empirical equation (5.38) together with included DNL error effect was derived, and according coefficients specified in table 5.6.

The effect of error oscillations at lowest bin densities was also investigated in section 5.4.2. If a certain distribution shape is known, equation 5.35 allows to determine error maximums and to adjust N and R accordingly. For the general case of unknown distribution shapes, $\sigma_{RJ,min} \cdot R \geq 2$ can avoid such oscillations.

Section 5.5 presented two typical design examples to highlight the calculation steps involved with optimized BIJM designs. The first example assumed a jitter diagnosis scenario with a minimum RJ tail of $\sigma_{t,min}=0.01\text{UI}$ and a realizable number of $R=128$ bins per UI. First, the minimum amplitude $A_{t,min}$ was determined to correctly specify ΔP_t, which guaranteed for correct fitting behavior. With equation (5.30) the sQN method achieved a worst case error bias $E_{med} \leq 2.0\%$ as well as an estimation loss $E_L \leq 5.4\%$ if the BIJM design was not affected by process variations. Otherwise, with $\sigma_{DNL}=0.05\text{UI}$ and equation (5.38) the error increased to $E_{med} \leq 2.3\%$ and $E_L \leq 7.4\%$.

The second design example focused on production testing with very stringent requirements on the test time $t_t \approx 20\text{ms}$. With 14 parallel counters, the assumed system would be able to collect only

$N{\approx}26\text{k}$ jitter values within the given test time, but still achieve an extrapolation error $E_L{<}15\%$.

The performance of the proposed and previously published tail fitting principles based on Gaussian quantile normalization was compared in chapter 6. These include the scaled Q-normalization (sQN) method from chapter 4 and 5, the conventional Q-normalization method (QN), as well as higher order polynomial methods (QP2, QP3 QP4) for tail fitting and extrapolation. The optimization scheme in figure 6.1 was first derived to give a unifying, generalized perspective on the compared methods. An efficient implementation of this scheme included the use of a fast Levinson-Durbin recursion. Optimum parameter regions were derived using the ΔP_t parameter for outer tail part selection and led to the configuration in table 6.1.

A comprehensive performance evaluation was first carried out for typical simulator environments by assuming $R{=}R_{sim}{=}3.3{\cdot}10^5$. With a varying sample size N and DJ type, figures 6.7-6.10 highlighted the different characteristics for each of the investigated methods. As a fundamental result, the QN method showed the same beneficial property of a strictly positive error bias as the sQN method. This is due to the asymptotic behavior of tails in Q-domain as already observed with the sQN method in section 4.2.1. Hence, extrapolation results are always pessimistic. This is a clear advantage compared to higher order polynomial methods (QP2, QP3, QP4), which achieved acceptable accuracy only for certain DJ types or for a small range of distributional shapes. Third and fourth order polynomials suffered from a large statistical variation of results, especially with RJ dominant distributions where the Q-tails approximately follow a linear course. This problem is generally due to an over-fitting effect.

A comprehensive comparison with the proposed sQN method ($\hat{c}_{1.2}$ optimization based on ΔP_t) was carried out in figures 6.11-6.15. The sQN method clearly achieved best performance as long as the fitting condition for tail amplitudes in equation (5.10) was correctly met. Although the QN method is less accurate, it offers the advantage of approximately 35 times faster tail fits, as was also shown previously in figure 4.9. With this complementary property and the pessimistic tail extrapolation, conventional QN is well suited for tail fitting and thus, also becomes a candidate for hardware designs. The influence of such a limited time discretization $R{\ll}R_{sim}$ on the extrapolation error was investigated in figure 6.16. It basically showed a linear increase of the extrapolation error over a large range of R for both QN and sQN methods.

In order to allow hardware designers to choose between the better suited algorithm alternative for a jitter measurement system, the extrapolation performance of the QN method was evaluated in section 6.5, equivalent to the sQN error analysis in section 5.4. Resulting coefficients were given in table 6.2 for the empirical relation (6.6) without additional process variations, and in table 6.3 for the empirical relation (6.7) with included process variations as differential non-linearity error. Hardware performance of the QN method was highlighted with the continued sQN design examples from section 5.5. Results showed that, although the QN method is generally less accurate than sQN, it is also less affected by differential non-linearity error.

For a jitter measurement system using the QN method, the previously derived design equations from chapter 5 can be applied again. Hence, equation (5.3) and condition (5.10) guarantee for a minimum amplitude $A_{t,min}$, while equation (5.20) and the selection chart in figure 5.9 guarantee for a minimum RJ standard deviation $\sigma_{t,min}$ to be fitted correctly.

The flexible architecture of the sQN optimization scheme was highlighted in chapter 7, where a generalized version let to a scheme for arbitrary non-Gaussian tail fitting. In this context, the generalized Gaussian distribution (GGD) function was introduced for tails with arbitrary exponential power law behavior. A GGD uses an additional shape parameter α, and includes the Gaussian distribution as a special case ($\alpha{=}2$). It is thus fully consistent with the existing RJ-DJ model.

For simulations where R_{sim} is sufficiently large, the GGD method achieved acceptable accuracy, although it suffered from a slightly negative or optimistic error bias. The reason is a general overestimation of shape parameters, as was shown in figure 7.6. For hardware scenarios with limited R, additional error further degraded the performance and let to unreliable extrapolations. This use case is thus not recommended. A performance comparison in figure 7.9 clearly highlighted the advantage of the generalized method when used together with heavy tailed test distributions ($\alpha \leq 2$), instead of the sQN method from chapter 4. Since the proposed method was implemented for use with a broad range of test distributions, further improvements may be achieved if specific shapes or test distributions are known and the fitting algorithm only focuses on these special characteristics.

Chapter 8 provided a first application example for the sQN method when used with system behavioral simulations. Therefore, an accurate CPLL model was implemented as an enhanced version of a prior event-driven approach. It allowed for dynamic run-time variations of parameters and included a gain regulator pole as well as a VCO noise model. On an Intel Core Duo 2.2GHz laptop, the model was able to gather up to 20k jitter values per second simulation time.

Initial simulations in figure 8.4 compared the closed loop phase noise PSD with measurement data at different parameter configurations, and showed an excellent agreement. Only spectral spurs at multiples of $50\mathrm{MHz}$, could not be reflected as they were not modeled. Two different simulation methods, one based on the sQN method and the other on a spectral method, determined jitter transfer functions of the modeled PLL. Resulting curves were again compared with measurements in figures 8.7-8.9. Both methods correctly identified the measured cut-off frequency of transfer functions. However, the spectral method better reflected the peaking behavior in the cut-off region, while the sQN method was instead able to correctly track curves obtained with different test patterns. This is due to the same analysis principle as used in measurements.

A second application example for the sQN method was given in chapter 9, together with an adaptive algorithm for jitter tolerance analysis of high-speed PLLs. It consists of an adaptive recursion (equation (9.1)) as well as a mechanism for automatic sample size adaptation (figure 9.2), to efficiently determine jitter tolerance curves.

In the example, the algorithm was applied to the PLL model from chapter 8. After deriving a well suited set of parameters (table 9.2), the performance of four different algorithmic combinations was investigated. These included use of the QN and sQN fitting methods as well as their realizations without sample size adaptation (c-QN and c-sQN).

Results highlighted a general decrease of the total number of required jitter samples $\overline{N}$ by a factor of 2-3, if the automatic sample size adaptation (QN, sQN) is utilized. The smallest computational effort was achieved with QN and c-QN. The smallest number of iterations instead, was given without sample size adaptation (c-QN and c-sQN), because their constant error bias does not influence the adaptation process. Hardware simulations with a reduced number of bins R (table 9.4 and figure 9.6) indicated that the developed algorithm can also be applied to hardware jitter measurements. Finally, in figure 9.7 simulated jitter tolerance curves were also compared against measurements from the according test structure and highlighted an excellent matching.

The third application example for tail fitting methods was given in chapter 10, where an FPGA based diagnostic tool for TJ estimation in high-speed interfaces was implemented. The measurement routine used an efficient algorithm, which realizes jitter measurements at 3Gb/s with a sample size of $N{=}10^8$ and $R{=}128$ in approximately 2s of test time. Using this diagnostic tool, the TJ_{pp} values of different RG-58 cables were determined in figure 10.6. The inherent jitter of the reference clock was quantified with loopback measurements and indicated $\mathrm{TJ}_{pp}{\approx}0.15\,\mathrm{UI}$. As an example for parameter optimization, the built-in Rx equalizers and Tx buffers of the FPGA were configured in figures 10.8-10.10 together with a $5\mathrm{m}$ RG-58 coaxial cable. For the best case with an included DFE, the TJ timing budget was decreased over 28% below $\mathrm{TJ}_{pp}{<}0.3\mathrm{UI}$.

The extrapolation error of sQN and QN fitting methods was investigated and compared against theoretical worst case errors from sections 5.4 and 6.5 respectively. This allowed to experimentally confirm the sinusoidal, uniform and quadratic curve DJ shapes as well suited for clock jitter, ISI, and ISI plus additional noise affected channels (figures 10.11, 10.12 and 10.13). The DNL error model was also successfully applied in figure 10.14 with jitter measurements from the Rx-PLL operating in lock-to-data mode.

11.2. Outlook

The sQN method from chapter 4 was shown to achieve an excellent accuracy of extrapolated tails, combined with a fast and flexible tail fitting procedure. The derived equations and empirical relations are primarily intended for use with a broad variety of test distributions and DJ types. For further improvements with respect to certain distribution shapes, it would thus be interesting to focus on specific test cases, where one of the tail parameters is for example known or can easily be approximated.

The residual analysis of fitted quantiles in section 4.1.3 highlighted a non-constant and correlated error structure for the outermost tail region in Q-domain. This makes the linear regression of quantiles still sub-optimal with respect to a maximum likelihood tail search [20]. However, an additional weighting and de-correlation of errors requires an excessive computational demand, and is thus not feasible. Nevertheless it would be interesting to compare the sQN error with the theoretical performance maximum for tail fitting methods. Such comparisons might for example become possible with fitting methods based on maximum likelihood approaches.

The generalized fitting principle in chapter 7 suffers from accuracy with a reduced number of bins R or a large shape parameter α. A detailed analysis of different fitness measures and their combinations, as well as the focus on a few specific tail shapes could possibly improve the performance.

Finally, with the given empirical error analysis of the presented sQN and QN methods, the extrapolation accuracy in all kind of future BIJM designs, diagnostic tools or measurement devices can be predicted as long as they provide jitter distributions. Thus, the derived empirical relations may also find a direct application in preliminary concept studies of novel systems.

A. Figure Data Files

The following lists denotes the MATLAB scripts which were used to post-process the C simulations, in order to generate the figures and tables of coefficients throughout this work. The scripts can be found in the sub-folders specified in the last column. All simulation data and results are documented on an appended DVD, available upon request at:
stefan.erb@gmx.at

Figure	Section	Page	MATLAB File	Folder
4.4 4.5	4.1.1	28 29	TestDist_analyze_bathtub	4/4/
4.7	4.1.3	32	qqtest_v3	3/5/
4.9	4.1.4	36	calc_time	4/8/
4.10	4.2	37	stim_ber_estimate_ML_RJDJ	4/3/
4.11 4.12	4.2.1	38 38	stim_ber_estimate_ML	4/3/
4.14 4.15	4.2.2	39 40	stim_ber_estimate_ML	4/3/
4.17 4.18 4.19 4.20	4.3.1	43 43 44 44	stim_ber_estimate_ML	4/3/
4.22 4.23 4.24	4.3.2	47 48 49	stim_ber_dPmin_Pmin_UImin	4/3/
4.25 4.26	4.3.3	50 51	stim_ber_estimate_ML	4/3/
4.29	4.4.2	54	stim_ber_estimate_ML	4/3/
4.30 4.31	4.4.2	56 56	stim_ber_estimate_ML	4/3/
4.32 4.33	4.4.3	57 57	stim_ber_estimate_ML	4/3/
5.2	5.1.1	61	app_tj_analysis	3/6/
5.3	5.1.1	62	stim_ber_estimate_ML_RJDJ	4/3/
5.4	5.1.2	63	app_tj_analysis	3/6/
5.8	5.3	69	stim_ber_estimate_ML_LTR	4/3/
5.10	5.3	71	stim_ber_estimate_ML_LTR	4/3/
5.12	5.4.1	73	stim_ber_estimate_ML_RJDJ	4/3/
5.13	5.4.1	74	stim_ber_estimate_ML_LTR	4/3/
5.14 5.15	5.4.1 5.4.2	75 78	stim_ber_N_RJmin_looped	4/3/

TABLE A.1.: List of MATLAB files to generate simulation figures.

Figure	Section	Page	MATLAB File	Folder
6.3	6.2	89	`stim_ber_dPmin_Pmin_UImin`	4/3/
6.4	6.2	90		
6.5	6.2	91		
6.6	6.2	91		
6.7	6.3	94	`stim_ber_estimate_poly`	4/3/
6.8	6.3	95		
6.9	6.3	96		
6.10	6.3	97		
6.11	6.4	99	`stim_ber_estimate_poly` `stim_ber_estimate_ML`	4/3/
6.12	6.4	100		
6.13	6.4	101		
6.14	6.4	102		
6.15	6.4	103		
6.16	6.4	104	`stim_ber_dPmin_Pmin_UImin`	4/3
7.6	7.3.1	116	`stim_ber_ggd_ML`	4/3/
7.7	7.3.1	117		
7.8	7.3.2	117		
7.9	7.3.3	118		
8.4	8.2.1	127	`cdr_stim_jit_FFT`	4/5/
8.5	8.2.1	128	`cdr_stim_jit_FFT`	4/5/
8.6	8.2.2	129	`cdr_stim_jit_FFT`	4/5/
8.7	8.2.2	130	`cdr_stim_transfer`	4/5
8.8		130		
8.9		131		
9.3	9.1.2	137	`stim_ber_estimate_ML_sin`	4/3/
9.4	9.2.1	139	`cdr_stim_JTOL`	4/5/
9.5	9.2.1	140	`cdr_stim_JTOL_loop`	4/5/
9.6	9.2.2	142	`cdr_stim_JTOL_Rloop`	4/5/
9.7	9.2.2	142	`cdr_stim_JTOL`	4/5/
10.5	10.2	149	`Gaussfit_v2`	4/6/
10.6	10.2	149	`Gaussfit_v2` `Gaussfit_erran_v1`	4/6/
10.7	10.2	150	`Gaussfit_v2`	4/6/
10.8	10.2	150	`GTX_Gaussfit_v1`	4/6/
10.9		151		
10.10	10.2	151	`RXTX_Gaussfit_v1`	4/6/
10.11	10.3	152	`Gaussfit_v2`	4/6/
10.12		152		
10.13		153		
10.14		154		
10.15	10.3	154	`Gaussfit_erran_v1`	4/6/

TABLE A.1.: List of MATLAB files to generate simulation figures. (continued)

Table	Section	Page	MATLAB File	Folder
5.1	5.1.1	62	app_tj_analysis	3/6/
5.2	5.1.1	64	app_tj_analysis	3/6/
5.4	5.4.1	76	stim_ber_N_RJmin_looped	4/3/
5.5		77		
5.6	5.4.3	80	stim_ber_N_RJmin_DNL_looped	4/3/
6.2	6.5	105	stim_ber_N_RJmin_looped	4/3/
6.3	6.5	106	stim_ber_N_RJmin_DNL_looped	4/3/

TABLE A.2.: List of MATLAB files to generate tables of coefficients.

<table>
<tr><th>System-C Testbench</th><th>Options File</th></tr>
<tr><th colspan="2">MATLAB Files for post-processing</th></tr>
<tr><td>BER_Test_stim_template</td><td>ber_test.opt</td></tr>
<tr><td colspan="2">stim_ber_estimate_ML
stim_ber_estimate_ML_LTR
stim_ber_estimate_poly
stim_ber_estimate_ML_RJDJ</td></tr>
<tr><td>BER_Test_stim_dPmin_Pmin_looped</td><td>ber_test_dPminPminloop.opt</td></tr>
<tr><td colspan="2">stim_ber_dPmin_Pmin_UImin</td></tr>
<tr><td>BER_Test_stim_UImin_Pmin_looped</td><td>ber_test_UIminPminloop.opt</td></tr>
<tr><td colspan="2">stim_ber_dPmin_Pmin_UImin</td></tr>
<tr><td>BER_Test_stim_N_RJmin_looped</td><td>ber_test_NRJminloop.opt</td></tr>
<tr><td colspan="2">stim_ber_N_RJmin_looped</td></tr>
<tr><td>BER_Test_RJ_tail</td><td>ber_test_RJ_tail.opt</td></tr>
<tr><td colspan="2">stim_ber_ggd_ML</td></tr>
<tr><td>cdr_stim_jit_fft</td><td>jit_fft_an.opt</td></tr>
<tr><td colspan="2">cdr_stim_jit_FFT</td></tr>
<tr><td>cdr_stim_jit_transfer</td><td>jit_trans_an.opt</td></tr>
<tr><td colspan="2">cdr_stim_transfer</td></tr>
<tr><td>cdr_stim_JTOL</td><td>jtol_an.opt</td></tr>
<tr><td colspan="2">cdr_stim_JTOL
cdr_stim_JTOL_loop
cdr_stim_JTOL_Rloop</td></tr>
</table>

TABLE A.3.: List of System-C testbenches for simulations and MATLAB post-processing files.

Own Publications

[C1] S. Erb and W. Pribyl, “An Accurate and Efficient Method for BER Analysis in High-Speed Communication Systems", *IEEE European Conf. on Circuit Theory and Design (ECCTD'09)*, Aug. 2009.

[C2] S. Erb and W. Pribyl, “A Behavioral Modeling Approach for Jitter Analysis in Charge-Pump PLLs", *Austrian Workshop on Microelectronics (AUSTROCHIP'09)*, Oct. 2009.

[C3] S. Erb and W. Pribyl, “Comparison of Jitter Decomposition Methods for BER Analysis of High-Speed Serial Links", *IEEE Symp. on the Design and Diagnostics of Electronic Circuits and Systems (DDECS'10)*, Apr. 2010.

[C4] S. Erb and W. Pribyl, “Design and Performance Considerations for an On-Chip Jitter Analysis System", *IEEE Int. Symp. on Circuits and Systems (ISCAS'10)*, May 2010.

[C5] S. Erb and W. Pribyl, “An Approach to Generalized Jitter and BER Analysis", *Austrian Workshop on Microelectronics (AUSTROCHIP'10)*, Oct. 2010.

[C6] S. Erb and W. Pribyl, “A Method for Fast Jitter Tolerance Analysis of High-Speed PLLs", *IEEE Conf. Design Automation and Test in Europe (DATE'11)*, Mar. 2011.

[C7] S. Erb, M. Stadler and W. Pribyl, “An FPGA based Diagnostic Tool for Jitter Optimization in Serial High-Speed Transceivers", *IEEE Ph.D. Research in Microelectronics & Electronics (PRIME'11)*, Jul. 2011, submitted for publication in Feb. 2011.

[C8] S. Erb and W. Pribyl, “Design Specification for BER Analysis Methods using Built-in Jitter Measurements", *IEEE Trans. VLSI Systems*, submitted for publication in Oct. 2010.

[C9] S. Erb, “Method and Device for Predicting a Figure of Merit from a Distribution", U.S. Patent Application, US2010/0 246 650 A1, Sep. 30, 2010.

Bibliography

[1] P. Acco, M. P. Kennedy, C. Mira, B. Morley, and B. Frigyik, “Behavioral Modeling of Charge Pump Phase Locked Loops,” in *IEEE Int. Symp. Circuits and Systems (ISCAS’99)*, 1999, pp. 375–378.

[2] O. Agazzi, M. Hueda, H. Carrer, and D. Crivelli, “Maximum-likelihood sequence estimation in dispersive optical channels,” *J. Lightwave Technology*, vol. 23, no. 2, pp. 749–763, Feb. 2005.

[3] Agilent Tech. (2003, Feb.) Jitter Analysis Techniques for High Data Rates. White Paper. Cited 2010-09-17. [Online]. Available: www.agilent.com

[4] S. Ahmed and T. Kwasniewski, “Efficient Simulation of Jitter Tolerance for All-Digital Data Recovery Circuits,” *IEEE Midwest Symp. Circuits and Systems (MWSCAS’07)*, pp. 1070–1073, Aug. 2007.

[5] S. Ali, “Basics of Chip-to-Chip and Backplane Signaling,” *IEEE Solid-State Circuits Conf. Tutorial (ISSCC’08)*, Feb. 2008.

[6] G. Balamurugan, B. Casper, J. Jaussi, M. Mansuri, F. O’Mahony *et al.*, “Modeling and Analysis of High-Speed I/O Links,” *IEEE Trans. Advanced Packaging*, vol. 32, no. 2, pp. 237–247, May 2009.

[7] R. E. Best, *Phase-Locked Loops - Design, Simulation and Applications*, 5th ed. New York (NY): McGraw-Hill, 2003.

[8] W. Beyene, C. Madden, J.-H. Chun, H. Lee, Y. Frans *et al.*, “Advanced Modeling and Accurate Characterization of a 16 Gb/s Memory Interface,” *IEEE Trans. Advanced Packaging*, vol. 32, no. 2, pp. 306–327, May 2009.

[9] L. Bizjak, “Development of a PLL Blocks Library for accurate Time-Domain simulations and Clock Analysis Software,” Master’s thesis, Università degli studi di Udine, I, 2005.

[10] G. E. P. Box and M. E. Muller, “A Note on the Generation of Random Normal Deviates,” *Ann. Math. Statistics*, no. 29, pp. 610–611, 1958.

[11] M. Brownlee, “Low Noise Clocking for High Speed Serial Links,” Ph.D. dissertation, Oregon State University, US, 2006.

[12] Y. Cai, A. Bhattacharyya, J. Martone, A. Verma, and W. Burchanowski, “A Comprehensive Production Test Solution for 1.5Gb/s and 3Gb/s Serial-ATA - based on AWG and Undersampling Techniques,” *IEEE Int. Test Conf. (ITC’05)*, Nov. 2005.

[13] Y. Cai, S. Werner, G. Zhang, M. Olsen, and R. Brink, “Jitter Testing for Multi-Gigabit Backplane SerDes - Techniques to Decompose and Combine Various Types of Jitter,” *IEEE Int. Test Conf. (ITC’02)*, pp. 700–709, Oct. 2002.

[14] Y. Cai, B. Laquai, and K. Luehman, “Jitter Testing for Gigabit Serial Communication Transceivers,” *IEEE Design & Test of Computers*, vol. 19, no. 1, pp. 66–74, Jan.-Feb. 2002.

[15] B. Casper and F. O'Mahony, "Clocking Analysis, Implementation and Measurement Techniques for High-Speed Data Links - A Tutorial," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 1, pp. 17–39, Jan. 2009.

[16] A. Chan and G. Roberts, "A Jitter Characterization System Using a Component-Invariant Vernier Delay Line," *IEEE Trans. VLSI Systems*, vol. 12, no. 1, pp. 79–95, Jan. 2004.

[17] A.-S. Chao and S.-J. Chang, "A Jitter Characterizing BIST with Pulse-Amplifying Technique," *IEEE Asian Test Symp. (ATS'09)*, pp. 379–384, Nov. 2009.

[18] K.-H. Cheng, J.-C. Liu, C.-Y. Chang, S.-Y. Jiang, and K.-W. Hong, "Built-in Jitter Measurement Circuit With Calibration Techniques for a 3-GHz Clock Generator," *IEEE Trans. VLSI Systems*, vol. PP, no. 99, pp. 1–11, Jun. 2010.

[19] S. Coles, *An Introduction to Statistical Modeling of Extreme Values*. London (GB): Springer, 2001.

[20] M. J. Crawley, *The R Book*. Chichester (GB): John Wiley & Sons, 2007.

[21] B. Daniels, R. Farrell, and G. Baldwin, "Arbitrary Order Charge Approximation Event Driven Phase Lock Loop Model," *IET Irish Signals and Systems Conf. (ISSC)*, Jul. 2004.

[22] N. Da Dalt, M. Harteneck, C. Sandner, and A. Wiesbauer, "Numerical Modeling of PLL Jitter and the Impact of its Non-White Spectrum on the SNR of Sampled Signals," *Southwest Symp. Mixed-Signal Design (SSMSD'01)*, pp. 38–44, Feb. 2001.

[23] N. Da Dalt and C. Sandner, "Introduction to PLL Jitter Definitions," presentation slides, Feb. 2001.

[24] N. Da Dalt, "Cheetah CDR L90 v2 Measurements," Infineon Technologies, Jan. 2007, confidential.

[25] N. Da Dalt, "Theory and Implementation of Digital Bang-Bang Frequency Synthesizers for High Speed Serial Data Communications," Dissertation, Technische Hochschule Aachen, D, 2007.

[26] A. R. DiDonato and A. H. Morris, Jr., "Computation of the Incomplete Gamma Function Ratios and their Inverse," *ACM Trans. Math. Softw.*, vol. 12, no. 4, pp. 377–393, 1986.

[27] Q. Dou and J. Abraham, "Jitter Decomposition by Time Lag Correlation," *IEEE Int. Symp. Quality Electronic Design (ISQED'06)*, Mar. 2006.

[28] Q. Dou and J. Abraham, "Jitter Decomposition in High-Speed Communication Systems," *IEEE European Test Symp. (ETS'08)*, pp. 157–162, May 2008.

[29] J. Eckle-Kohler and M. Kohler, *Eine Einführung in die Statistik und ihre Anwendungen*. Berlin (D): Springer, 2008.

[30] W. F. Egan, *Phase-Lock Basics*, 2nd ed. Hoboken (NJ): John Wiley & Sons, 2007.

[31] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, 2nd ed. Hoboken (NJ): John Wiley & Sons, 1993.

[32] Y. Fan, Y. Cai, L. Fang, A. Verma, W. Burchanowski *et al.*, "An Accelerated Jitter Tolerance Test Technique on ATE for 1.5GB/s and 3GB/s Serial-ATA," *IEEE Int. Test Conf. (ITC'06)*, pp. 1–10, Oct. 2006.

[33] Y. Fan and Z. Zilic, “Accelerating Jitter Tolerance Qualification for High Speed Serial Interfaces,” *IEEE Int. Symp. Quality of Electronic Design (ISQED'09)*, pp. 360–365, Mar. 2009.

[34] A. Fog. (2008, Feb.) “C++ Library of Pseudo Random Number Generators”. Cited 2008-10-09. [Online]. Available: www.agner.org/random/

[35] A. Frisch, “Jitter Measurement System and Method,” U.S. Patent 2002/0 106 014 A1, Aug. 8, 2002.

[36] F. M. Gardner, *Phase Lock Techniques*, 3rd ed. Hoboken (NJ): John Wiley & Sons, 2004.

[37] T. Georges, “Non-Gaussian timing jitter statistics of controlled solitons,” in *Optical Fiber Communications (OFC'96)*, Feb. 1996, pp. 232–233.

[38] F. Ghenassia, Ed., *Transaction Level Modeling with SystemC*. Dordrecht (NL): Springer, 2005.

[39] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore (MD): Johns Hopkins University Press, 1996.

[40] V. Grigoryan, C. Menyuk, and R.-M. Mu, “Calculation of Timing and Amplitude Jitter in Dispersion-Managed Optical Fiber Communications using Linearization,” *J. Lightwave Technology*, vol. 17, no. 8, pp. 1347–1356, Aug. 1999.

[41] T. Grötker, S. Liao, G. Martin, and S. Swan, *System Design with SystemC*. Boston (MA): Kluwer Academic Publishers, 2002.

[42] A. Hajimiri and T. Lee, *The Design of Low Noise Oscillators*. Dordrecht (NL): Kluwer Academic Publishers, 1999.

[43] B. Ham, “Methodologies for Jitter and Signal Quality Specification,” INCITS, Tech. Rep., Jun. 2005.

[44] Y. Hamaizi and A. El-Akrmi, “Soliton Propagation in Fiber Systems,” in *ICTON Mediterranean Winter Conf. (ICTON-MW'09)*, Dec. 2009, pp. 1–4.

[45] P. K. Hanumolu, “Design Techniques for Clocking High Performance Signaling Systems,” Ph.D. dissertation, Oregon State University, US, 2006.

[46] P. Hanumolu, M. Brownlee, K. Mayaram, and U.-K. Moon, “Analysis of Charge-Pump Phase-Locked Loops,” *IEEE Trans. Circuits Syst. I*, vol. 51, no. 9, pp. 1665–1674, Sept. 2004.

[47] P. Hanumolu, B. Casper, R. Mooney, G.-Y. Wei, and U.-K. Moon, “Jitter in High-Speed Serial and Parallel Links,” *IEEE Int. Symp. Circuits and Systems (ISCAS'04)*, vol. 4, pp. IV–425–428, May 2004.

[48] T. Hashimoto, H. Yamazaki, A. Muramatsu, T. Sato, and A. Inoue, “Time-to-Digital Converter with Vernier Delay Mismatch Compensation for High Resolution On-Die Clock Jitter Measurement,” in *IEEE Symp. VLSI Circuits*, Jun. 2008, pp. 166–167.

[49] S. Haykin, *Communication Systems*, 4th ed. Hoboken (NJ): John Wiley & Sons, 2001.

[50] C. D. Hedayat, A. Hachem, Y. Leduc, and G. Benbassat, “Modeling and Characterization of the 3rd Order Charge-Pump PLL: a Fully Event-driven Approach,” *Analog Integrated Circuits and Signal Processing*, vol. 19, no. 1, pp. 25–45, 1999.

[51] G. Hänsel, K. Stieglbauer, G. Schulze, and J. Moreira, “Implementation of an Economic Jitter Compliance Test for a Multi-Gigabit Device on ATE,” *IEEE Int. Test Conf. (ITC’04)*, pp. 1303–1312, Oct. 2004.

[52] D. Hong, C.-K. Ong, and K.-T. Cheng, “Bit-Error-Rate Estimation for High-Speed Serial Links,” *IEEE Trans. Circuits Syst. I*, vol. 53, no. 12, pp. 2616–2627, Dec. 2006.

[53] D. Hong, “Efficient Test Methodologies for High-Speed Serial Links,” Ph.D. dissertation, University of California, Santa Barbara, US, 2008.

[54] D. Hong and K.-T. Cheng, “An Accurate Jitter Estimation Technique for Efficient High Speed I/O Testing,” *IEEE Asian Test Symp. (ATS’07)*, pp. 224–229, Oct. 2007.

[55] D. Hong and K.-T. Cheng, “Bit-Error Rate Estimation for Bang-Bang Clock and Data Recovery Circuit in High-Speed Serial Links,” *IEEE VLSI Test Symp. (VTS’08)*, pp. 17–22, May 2008.

[56] D. Hong, C.-K. Ong, and K.-T. Cheng, “BER Estimation for Serial Links Based on Jitter Spectrum and Clock Recovery Characteristics,” *IEEE Int. Test Conf. (ITC’04)*, pp. 1138–1147, Oct. 2004.

[57] J.-C. Hsu and C. Su, “BIST for Measuring Clock Jitter of Charge-Pump Phase-Locked Loops,” *IEEE Trans. Instrumentation and Measurement*, vol. 57, no. 2, pp. 276–285, Feb. 2008.

[58] J.-L. Huang, “A Random Jitter Extraction Technique in the Presence of Sinusoidal Jitter,” *IEEE Asian Test Symp. (ATS’06)*, pp. 318–326, Nov. 2006.

[59] M. Hueda, D. Crivelli, H. Carrer, and O. Agazzi, “Parametric Estimation of IM/DD Optical Channels Using New Closed-Form Approximations of the Signal PDF,” *J. Lightwave Technology*, vol. 25, no. 3, pp. 957–975, Mar. 2007.

[60] K. Ichiyama, M. Ishida, T. J. Yamaguchi, and M. Soma, “Novel CMOS Circuits to Measure Data-Dependent Jitter, Random Jitter, and Sinusoidal Jitter in Real Time,” *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 5, pp. 1278–1285, May 2008.

[61] *SystemC Specification 1666*, IEEE Std., Rev. 2.1, 2005.

[62] *Nitrophy Analog Core Reference Manual*, Infineon Technologies, 2007, confidential.

[63] A. Jantsch, *Modeling Embedded Systems and SoC’s*. US: Elsevier Science, 2003.

[64] J. Jaussi, G. Balamurugan, J. Kennedy, F. O’Mahony, M. Mansuri *et al.*, “In-Situ Jitter Tolerance Measurement Technique for Serial I/O,” in *IEEE Symp. VLSI Circuits*, Jun. 2008, pp. 168–169.

[65] K. Jenkins, A. Jose, Z. Xu, and K. Shepard, “On-chip Circuit for Measuring Jitter and Skew with Picosecond Resolution,” in *Proc. of IEEE Int. Conf. Integrated Circuit Design and Technology (ICICDT’08)*, Jun. 2008, pp. 257–260.

[66] S.-Y. Jiang, K.-H. Cheng, and P.-Y. Jian, “A 2.5-GHz Built-in Jitter Measurement System in a Serial-Link Transceiver,” *IEEE Trans. VLSI Systems*, vol. 17, no. 12, pp. 1698 –1708, Dec. 2009.

[67] S. G. Johnson. (2009, Nov.) “The NLopt Nonlinear-Optimization Package”. MIT. Cited 2010-01-31. [Online]. Available: ab-initio.mit.edu/nlopt

[68] J. Kim, “On-Chip Measurement of Jitter Transfer and Supply Sensitivity of PLL/DLLs,” *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 56, no. 6, pp. 449–453, Jun. 2009.

[69] K. K. Kim, J. Huang, Y.-B. Kim, and F. Lombardi, “Analysis and Simulation of Jitter Sequences for Testing Serial Data Channels,” *IEEE Trans. Industrial Informatics*, vol. 4, no. 2, pp. 134–143, May 2008.

[70] J. M. Kizer and C. J. Madden, “Method for Estimating RJ and DJ,” U.S. Patent US 2006/0 059 392 A1, Mar. 16, 2006.

[71] R. Koenker, *Quantile Regression.* New York (NY): Cambridge University Press, 2005.

[72] M. Kossel and M. Schmatz, “Jitter Measurements of High-Speed Serial Links,” *IEEE Design & Test of Computers*, vol. 21, no. 6, pp. 536–543, Nov.-Dec. 2004.

[73] M. Kubicek, “In-System Jitter Measurement Using FPGA,” in *20th Int. Conf. Radioelektronika*, Apr. 2010, pp. 1–4.

[74] K. Kundert. (2006, Aug.) Modeling Jitter in PLL-based Frequency Synthesizers. Cited 2008-05-27. [Online]. Available: www.designers-guide.org

[75] A. Kuo, T. Farahmand, N. Ou, S. Tabatabaei, and A. Ivanov, “Jitter Models and Measurement Methods for High-Speed Serial Interconnects,” *IEEE Int. Test Conf. (ITC’04)*, pp. 1295–1302, Oct. 2004.

[76] A. Kuo, R. Rosales, T. Farahmand, S. Tabatabaei, and A. Ivanov, “Crosstalk Bounded Uncorrelated Jitter (BUJ) for High-Speed Interconnects,” *IEEE Trans. Instrumentation and Measurement*, vol. 54, no. 5, pp. 1800–1810, Oct. 2005.

[77] P. Larsson, “A Simulator Core for Charge-Pump PLLs,” *IEEE Trans. Circuits Syst. II*, vol. 45, no. 9, pp. 1323–1326, Sep. 1998.

[78] T. Lee and A. Hajimiri, “Oscillator Phase Noise: A Tutorial,” *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 326–336, Mar. 2000.

[79] Y. Lee, C.-Y. Yang, N.-C. Cheng, and J.-J. Chen, “An Embedded Wide-Range and High-Resolution Clock Jitter Measurement circuit,” in *IEEE Conf. Design, Automation and Test in Europe (DATE’10)*, 2010, pp. 1637–1640.

[80] H. Le Gall, “Estimating of the Jitter of a Clock Signal,” U.S. Patent 7 487 055, Feb. 3, 2009.

[81] H. Le Gall, “Jitter Estimation Circuit,” ST Microelectronics, Mar. 2011, private communication, confidential.

[82] M. P. Li, *Jitter, Noise, and Signal Integrity at High-Speed.* Boston (MA): Prentice Hall, 2007.

[83] M. Li, “Jitter Challenges and Reduction Techniques at 10 Gb/s and Beyond,” *IEEE Trans. Advanced Packaging*, vol. 32, no. 2, pp. 290–297, May 2009.

[84] M. Li, J. Wilstrup, R. Jessen, and D. Petrich, “A New Method for Jitter Decomposition Through its Distribution Tail Fitting,” *IEEE Int. Test Conf. (ITC’99)*, pp. 788–794, Sep. 1999.

[85] M. Li, J. Wilstrup, R. Jessen, and D. Petrich, “Method and Apparatus for Analyzing Measurement,” U.S. Patent 6 298 315, Oct. 2, 2001.

[86] T. Mak, M. Tripp, and A. Meixner, “Testing Gbps Interfaces without a Gigahertz Tester,” *IEEE Design & Test of Computers*, vol. 21, no. 4, pp. 278–286, July-Aug. 2004.

[87] M. Mansuri, A. Hadiashar, and C.-K. K. Yang, “Methodology for On-Chip Adaptive Jitter Minimization in Phase-Locked Loops,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 11, pp. 870–878, Nov. 2003.

[88] MathWorks Inc. (2009, Mar.) “Optimization Toolbox User’s Guide V4.2 (R2009a)”. MATLAB Documentation. [Online]. Available: www.mathworks.com

[89] “Physical Layer Performance: Testing the Bit Error Ratio (BER),” Technical Article, Maxim Inc., Sep. 2004.

[90] S. McClure, “Digital Jitter Measurement and Separation,” Master’s thesis, Texas Tech University, US, 2006.

[91] J. A. McNeill and D. Ricketts, *The Designer’s Guide to Jitter in Ring Oscillators*. New York (NY): Springer, 2010.

[92] S. E. Meninger, “Low Phase Noise, High Bandwidth Frequency Synthesis Techniques,” Ph.D. dissertation, Massachusetts Institute of Technology, US, 2007.

[93] M. Miller, “Estimating Total Jitter Concerning Precision, Accuracy and Robustness,” *DesignCon*, Feb. 2007.

[94] M. Miller, “Measuring Components of Jitter,” U.S. Patent 7 516 030, Apr. 7, 2009.

[95] M. Miller. (2007) Normalized Q-scale analysis: Theory and background. EDN. Cited 2011-04-12. [Online]. Available: www.edn.com

[96] M. Müller, R. W. Stephens, and R. McHugh, “Total Jitter Measurement at Low Probability Levels, Using Optimized BERT Scan Methods,” White Paper, Agilent Technologies, 2005.

[97] P. Muller and Y. Leblebici, “Jitter Tolerance Analysis of Clock and Data Recovery Circuits using Matlab and VHDL-AMS,” in *Proc. of Forum on Design Languages (FDL’05)*, 2005.

[98] F. Nan, Y. Wang, F. Li, W. Yang, and X. Ma, “A Better Method than Tail-fitting Algorithm for Jitter Separation Based on Gaussian Mixture Model,” *J. of Electronic Testing: Theory and Applications*, vol. 25, no. 6, pp. 337–342, Dec. 2009.

[99] R. Nonis, “Phase Noise Modeling in PLL Frequency Synthesizers,” Master’s thesis, Università degli studi di Udine, I, 2002.

[100] K. Nose, M. Kajita, and M. Mizuno, “A 1-ps Resolution Jitter-Measurement Macro Using Interpolated Jitter Oversampling,” *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2911–2920, Dec. 2006.

[101] C.-K. Ong, D. Hong, K.-T. Cheng, and L.-C. Wang, "Jitter Spectral Extraction for Multi-Gigahertz Signal," *IEEE Asia - South Pacific Design Automation Conf. (ASP-DAC'04)*, pp. 298–303, Jan. 2004.

[102] C.-K. Ong, D. Hong, K.-T. Cheng, and L.-C. Wang, "A Clock-Less Jitter Spectral Analysis Technique," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 8, pp. 2263–2272, Sept. 2008.

[103] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. New York (NY): Pearson, 2010.

[104] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, and A. Ivanov, "Jitter Models for the Design and Test of Gbps-Speed Serial Interconnects," *IEEE Design & Test of Computers*, vol. 21, no. 4, pp. 302–313, July-Aug. 2004.

[105] H. Pang, J. Zhu, and W. Huang, "Jitter Decomposition by Fast Fourier Transform and Time Lag Correlation," in *IEEE Int. Conf. Communications, Circuits and Systems ICCCAS*, Jul. 2009, pp. 365–368.

[106] E. Parzen, "Nonparametric Statistical Data Modeling," *J. American Statistical Association*, vol. 74, no. 365, pp. 105–121, Mar. 1979.

[107] M. Perrott, "Fast and Accurate Behavioral Simulation of Fractional-N Frequency Synthesizers and Other PLL/DLL Circuits," in *IEEE Design Automation Conf. (DAC'02)*, Jun. 2002, pp. 498–503.

[108] M. Perrott, M. Trott, and C. Sodini, "A Modeling Approach for Σ-Δ Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug. 2002.

[109] M. H. Perrott. (2008, Apr.) "CppSim System Simulator Package V4". MIT. Cited 2009-06-09. [Online]. Available: www.cppsim.com

[110] M. H. Perrott, "Digital Phase-Locked Loops," *IEEE Solid-State Circuits Conf. Tutorial (ISSCC'08)*, Feb. 2008.

[111] A. Popovici, "Fast Measurement of Bit Error Rate in Digital Links," *IEE Proc. Communications, Radar and Signal Processing*, vol. 134, no. 5, pp. 439–447, Aug. 1987.

[112] M. J. Porsani and T. J. Ulrych, "Levinson-Type Algorithms for Polynomial Fitting and for Cholesky and Q-Factors of Hankel and Vandermonde Matrices," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 63–70, Jan. 1995.

[113] M. J. D. Powell, "The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives," Department of Applied Mathematics and Theoretical Physics, Cambridge England, Tech. Rep., 2009.

[114] J. Proakis, *Digital Communications*, 4th ed. New York (NY): McGraw-Hill, 2001.

[115] R.-D. Reiss and M. Thomas, *Statistical Analysis of Extreme Values*, 2nd ed. Basel (CH): Birkhäuser, 2001.

[116] F. Scholz. (2008, May) Applications of the Noncentral t-Distribution. Stat 498B Industrial Statistics. Cited 2011-04-12. [Online]. Available: www.stat.washington.edu/fritz/

[117] F. Scholz, "Nonparametric Tail Extrapolation," *Boeing Information & Support Services, ISSTECH-95-014*, 1995. [Online]. Available: www.stat.washington.edu/fritz/

[118] *Serial ATA Specification*, Serial ATA International Organization Std., Rev. 2.6, 2006.

[119] M. Shimanouchi, "An Approach to Consistent Jitter Modeling for Various Jitter Aspects and Measurement Methods," *IEEE Int. Test Conf. (ITC'01)*, pp. 848–857, Oct.-Nov. 2001.

[120] M. Shimanouchi, M. Li, and D. Chow, "New Modeling Methods for Bounded Gaussian Jitter (BGJ)/Noise (BGN) and their Applications in Jitter/Noise Estimation/Testing," *IEEE Int. Test Conf. (ITC'09)*, pp. 1–8, 2009.

[121] K. Shu and E. Sánchez-Sinencio, *CMOS PLL Synthesizers.* Boston (MA): Springer Science, 2005.

[122] R. Staszewski, C. Fernando, and P. Balsara, "Event-driven Simulation and Modeling of Phase Noise of an RF Oscillator," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 4, pp. 723–733, Apr. 2005.

[123] R. W. Stephens, "Jitter Analysis: The dual-Dirac model, RJ/DJ, and Q-Scale," White Paper, Agilent Technologies, Dec. 2004.

[124] R. Stephens, "Separation of Random and Deterministic Components of Jitter," U.S. Patent 7 149 638, Dec. 12, 2006.

[125] R. Stephens, "Separation of a Random Component of Jitter and a Deterministic Component of Jitter," U.S. Patent 7 191 080, Mar. 13, 2007.

[126] N. Stojanovic, "Tail Extrapolation in MLSE Receivers Using Nonparametric Channel Model Estimation," *IEEE Trans. Signal Processing*, vol. 57, no. 1, pp. 270–278, Jan. 2009.

[127] V. Stojanovic, "Channel Limited High-Speed Serial links - Modeling, Analysis and Design," Ph.D. dissertation, Stanford University, US, 2004.

[128] J. Sun, M. Li, and J. Wilstrup, "A Demonstration of Deterministic jitter (DJ) Deconvolution," *IEEE Instrumentation and Measurement Technology Conf. (IMTC'02)*, vol. 1, pp. 293–298, May 2002.

[129] S. Sunter and A. Roy, "On-chip Digital Jitter Measurement, from Megahertz to Gigahertz," *IEEE Design & Test of Computers*, vol. 21, no. 4, pp. 314–321, July-Aug. 2004.

[130] S. Tabatabaei and A. Ivanov, "Embedded Timing Analysis: A SoC Infrastructure," *IEEE Design & Test of Computers*, vol. 19, no. 3, pp. 22–34, May-June 2002.

[131] C.-C. Tsai and C.-L. Lee, "An On-Chip Jitter Measurement Circuit for the PLL," *IEEE Asian Test Symp. (ATS'03)*, pp. 332–335, Nov. 2003.

[132] S. Vamvakos, C. Werner, and B. Nikolic, "Phase-Locked Loop Architecture for Adaptive Jitter Optimization," in *Int. Symp. Circuits and Systems (ISCAS '04)*, vol. 4, May 2004, pp. IV–161–164.

[133] R. C. Walker, "Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems," in *Phase-Locking in High-Performance Systems: From Devices to Architectures*, B. Razavi, Ed. IEEE Press, 2003, pp. 34–45.

[134] Z. Wang, "An Analysis of Charge-Pump Phase-Locked Loops," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 10, pp. 2128–2138, Oct. 2005.

[135] S. Weinstein, “Estimation of Small Probabilities by Linearization of the Tail of a Probability Distribution Function,” *IEEE Trans. Communication Technology*, vol. 19, no. 6, pp. 1149–1155, Dec. 1971.

[136] S. Wisetphanichkij and K. Dejhan, “Jitter Decomposition by Derivatived Gaussian Wavelet Transform,” *IEEE Int. Symp. Communication and Information Technology (ISCIT’04)*, vol. 2, pp. 1160–1165, Oct. 2004.

[137] Xilinx Inc. (2008, Nov.) Virtex-5 FPGA RocketIO GTX Transceiver User’s Guide. ug198.pdf. Cited 2011-02-15. [Online]. Available: www.xilinx.com

[138] Xilinx Inc. (2009, Oct.) ML50x Evaluation Platform User’s Guide. ug347.pdf. Cited 2011-02-15. [Online]. Available: www.xilinx.com

[139] T. Yamaguchi, H. Hou, K. Takayama, D. Armstrong, M. Ishida *et al.*, “An FFT-based Jitter Separation Method for High-Frequency Jitter Testing with a 10x Reduction in Test Time,” *IEEE Int. Test Conf. (ITC’07)*, pp. 1–8, Oct. 2007.

[140] T. Yamaguchi, M. Soma, M. Ishida, H. Musha, and L. Malarsie, “A New Method for Testing Jitter Tolerance of SerDes Devices Using Sinusoidal Jitter,” *IEEE Int. Test Conf. (ITC’02)*, pp. 717–725, Oct. 2002.

[141] J. Yin and L. guang Zeng, “A Statistical Jitter Tolerance Estimation Applied for Clock and Data Recovery Using Oversampling,” *IEEE Region 10 Conf. (TENCON’06)*, pp. 1–4, Nov. 2006.

[142] I. Zamek and S. Zamek, “Definitions of Jitter Measurement Terms and Relationships,” *IEEE Int. Test Conf. (ITC’05)*, Nov. 2005.

[143] J. Zhu and W. Huang, “Jitter Analysis and Decomposition Based on EMD/HHT in High-Speed Serial Communications,” in *Proc. of IEEE Int. Conf. Testing and Diagnosis (ICTD’09)*, Apr. 2009, pp. 1–4.