A Method for Fast Jitter Tolerance Analysis of High-Speed PLLs

Stefan Erb, Wolfgang Pribyl
Institute of Electronics – Graz University of Technology
Inffeldgasse 12/I, 8010 Graz, Austria
{stefan.erb, wolfgang.pribyl}@tugraz.at

***Abstract*—We propose a fast method for identifying the jitter tolerance curves of high-speed phase locked loops. The method is based on an adaptive recursion and uses known tail fitting methods to realize a fast optimization combined with a small number of jitter samples. It allows for efficient behavioral simulations, and can also be applied to hardware measurements. A typical modeling example demonstrates applicability to both software and hardware scenarios and achieves simulated measurement times in the range of few hundred milliseconds.**

I. Introduction

Clock and data recovery (CDR) circuits in serial high-speed interfaces such as S-ATA [1] or Fiber Channel [2] use phase locked loops (PLLs) for synchronization. The influence of timing jitter is crucial, since noise and distortions along the transmission channel can highly affect signal integrity. Thus, especially high-speed PLLs have to provide a certain robustness against timing variations, commonly referred to as jitter tolerance (JTOL). Interface standards thus often specify tolerance masks [2] which must be guaranteed by the CDR.

A typical JTOL measurement scheme as depicted in fig. 1 uses a modulated clock source with corresponding pattern generator, and is characterized by the injected sinusoidal jitter of 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) can compare the recovered data with the expected original one, and determine the resulting BER. Equivalently, a time interval analyzer (TIA) can measure the time difference between the zero crossing of the analog input and the recovered clock edge. The obtained values are referred to as IO jitter and with a collected histogram, the BER can also be determined.

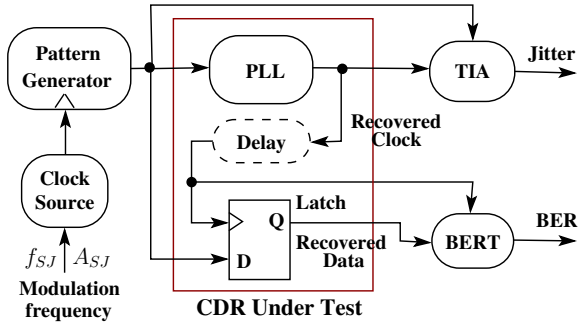


Fig. 1. JTOL measurement scheme using TIA or BERT.

The relation between TIA and BERT measurements is shown in fig. 2. A TIA measures jitter values at every bit transition of the received data stream and constructs the probability function or CDF out of a collected distribution. A typical representation form is the bathtub plot as shown at the bottom, with the tails given as a function of sampling time. The BERT measurement is easy to implement since bit errors are simply counted at the receiver side. The result is a single probability value of the jitter distribution, so that a CDR requires an additional delay element or phase shifter in order to identify the overall bathtub curve.

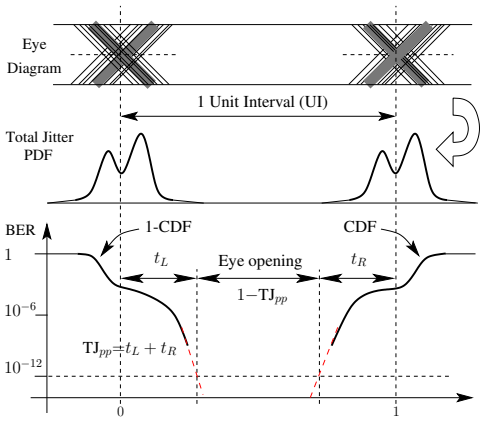


Fig. 2. Bathtub function of a measured jitter distribution.

A key problem is that specification standards require very low target error rates of typically 10^{-12} which can not be measured. In a 3Gbit SATA interface the transmission of 10^{12} bits consumes $5.5\,\mathrm{min}$, while production tests have to be carried out in less than a second. Jitter tolerance measurements are even more challenging, because they have to identify the sinus amplitude where the CDR exactly produces the target BER. This means, we must search for the A_{SJ} value where both bathtub tails in fig. 2 cross each other at the 10^{-12} level.

So far, the JTOL test problem has been addressed either from a measurement or a simulation perspective. Methods for hardware measurements have been proposed in [3]–[5], where especially the principle in [4] is very efficient as it is based on a linearizing transform with subsequent tail

978-3-9810801-7-9/DATE11/©2011 EDAA

extrapolation. This BERT based approach is simple to realize, but it requires a considerable amount of detected bit errors and is thus too time consuming if the whole jitter tolerance curve has to be identified over varying frequency f_{SJ}. Simulation methods work equivalent to TIA based measurements. They additionally use statistical models [6] or special waveforms [7] to minimize the required amount 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 paper we propose an analysis method where the jitter tolerance of a PLL can be determined very quickly using an adaptive algorithm. The method is sufficiently fast to be used for behavioral simulations, but also for TIA based jitter measurements. A minimum measurement time is achieved by automatically adapting the sample size of collected jitter distributions according to the dynamics of the PLL. Built-in jitter measurement (BIJM) systems with a real time TIA feature are area consuming, but have already been realized successfully using high resolution time-to-digital converters [8], [9]. If such circuits are combined with our proposed approach, very fast JTOL measurements can be carried out on hardware structures.

In the following section we give a detailed description of the proposed algorithm. As an example, we apply our method to a 3Gbit high-speed PLL model and demonstrate applicability to both software and hardware scenarios. A brief summary is given in the conclusions.

II. JTOL Algorithm Principle

A JTOL method is a search algorithm which identifies the jitter amplitude A_{SJ} where the tested PLL produces the target BER. This is an inverse problem, which must be solved for every desired jitter frequency f_{SJ}. According to fig. 2 the goal is to cross the two bathtub tails at BER=10^{-12}. As a major limitation we are not able to measure or simulate the 10^{-12} level in a feasible amount of time, and thus use tail fitting algorithms from [10] to extrapolate the distributions down to the target BER. For performance comparison we apply two different algorithms, as briefly described in the following subsection. 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 measurements require N to be as small as possible.

As a solution to this problem we propose a twofold search method. 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

$$A_{SJ}(n+1) = A_{SJ}(n) + \mu \cdot e(n) \qquad (1)$$

This equation is widely used in adaptive filter theory [11] where least-mean-square algorithms or Kalman filters are implemented, and offers a high robustness against statistical variations of the error term $e(n)$. With a block size of N jitter values, the tail fitting algorithms in sec. II-A provide a cost function and specify the error $e(n)$. The jitter amplitude for the next iteration $A_{SJ}(n+1)$ is then determined according to eq.(1), using the old value $A_{SJ}(n)$ and the error $e(n)$, which is scaled by the learning rate parameter μ.

The second algorithm part adaptively adjusts the sample size N for each collected jitter distribution. The method starts with a minimum size N_{min} and decides after each iteration, whether N must be increased or not. As soon as a maximum N_{max} is successfully reached, the search algorithm converges. This second algorithm part 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 close to the final result. It is described in sec. II-B.

A. Tail Fitting Method and Cost Function

Tail fitting methods [10], [12] commonly assume Gaussian tail behavior at both distribution endings and try to identify the Gaussian model parameters for correct extrapolation. In this context, the Gaussian quantile normalization has become very popular, as it allows for simple and accurate extrapolations. We thus use two methods based on this principle, which have also been described and compared in [10].

Measured bathtub functions are basically transformed into quantile domain, where a Gaussian tail appears as a straight line and can thus be extrapolated easily via linear regression. This transform is given by the Q-function:

$$Q(p) = \sqrt{2} \cdot \mathrm{erfc}^{-1}\big(2 \cdot p\big)\,, \quad p = \mathrm{CDF}(x) \qquad (2)$$

An example for such transformed tails is given in fig. 3. It shows fitted tails of a probability distribution with N=10^6 jitter samples, when extrapolated down to the target probability 10^{-12} in the unit bit interval (UI). In our example, extrapolations range over six orders of magnitude. Note, that the amplitudes of identified tails (dashed lines) differ slightly because of statistical tail variations.

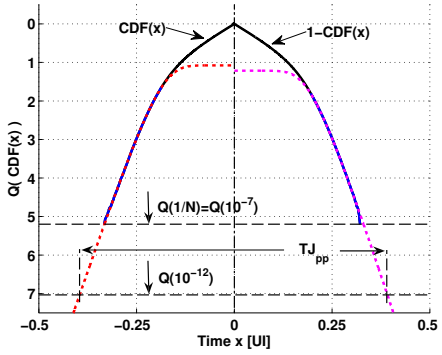


Fig. 3. Tail fitting principle according to [10].

The first fitting method, scaled Q-normalization (sQN) fully identifies the three Gaussian model parameters amplitude A, standard deviation σ and mean μ, and thus requires more

computational demand. The second method, conventional Q-normalization (QN) is a simplified version of the first one. It assumes A=1, and is thus significantly faster but also less accurate. The two fitting methods are applied to both left and right distribution tails, which finally allows for estimating the timing budget or total jitter value TJ_{pp} at 10^{-12}:

$$TJ_{pp} = t_L + t_R \tag{3}$$

$$t_{L(R)} = \mu_{L(R)} + \sigma_{L(R)} \cdot Q(10^{-12}/A_{L(R)}) \tag{4}$$

For the adaptive JTOL algorithm in eq.(1) a suited error term $e(n)$ should behave proportional to the injected jitter amplitude A_{SJ}. In [4] it is demonstrated that, besides the injected sinusoidal, all jitter sources in a JTOL measurement can be considered as uncorrelated and approximately constant. In Gaussian quantile domain (eq.(2)) this allows for the construction of a linear relationship, which is valid over a certain amplitude range.

$$Q = a \cdot A_{SJ} + b \tag{5}$$

Theoretically we would thus only need two measurement points to determine the unknown parameters, but unfortunately the tail extrapolations suffer from statistical variations. This impedes a direct calculation of A_{SJ}, but we can 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 (TJ_{pp}=1) can be determined by rewriting eq.(3):

$$\frac{(1-\mu_R-\mu_L)}{\sigma_L+\sigma_R} = \begin{cases} \frac{\sigma_L \cdot Q(p/A_L)+\sigma_R \cdot Q(p/A_R)}{\sigma_L+\sigma_R} & \text{(sQN)} \\ Q(p), \text{ with } A_L=A_R=1 & \text{(QN)} \end{cases} \tag{6}$$

For the QN method $Q(p)$=Q_{est} is directly calculated from the left hand side, while for sQN it can be determined recursively with a simple Newton iteration. Q_{est} must approach the desired target BER=10^{-12}, which gives the normalized error term $e(n)$:

$$e(n) = \frac{Q_{est} - Q(10^{-12})}{Q(10^{-12})} = \frac{Q_{est}}{7.03} - 1 \tag{7}$$

which is used together with the adaptive algorithm eq.(1).

B. 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 whether 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 fig. 4, 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 block of jitter samples has been collected. After applying the tail fitting method, the error $e(n)$ from eq.(7) is determined and used for updating the recursion in eq.(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.

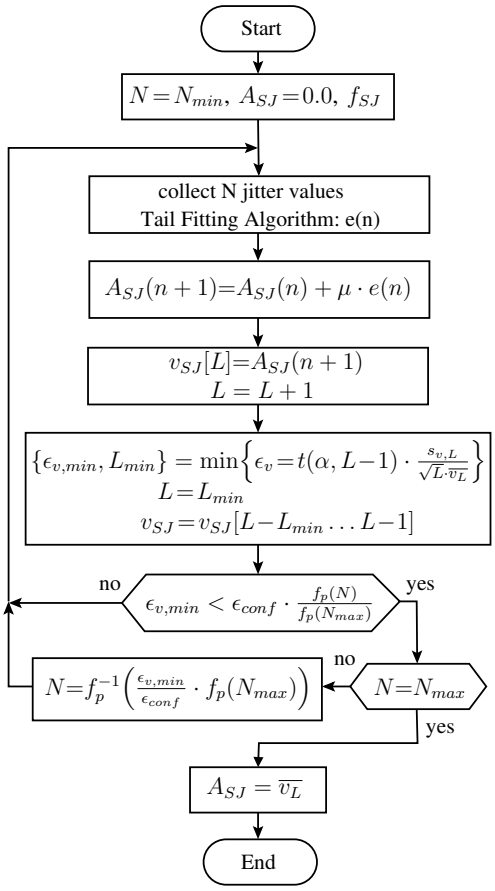


Fig. 4. Flow graph of JTOL analysis algorithm.

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. We can derive a measure for the statistical variation of $A_{SJ}(n)$ if we only consider L last recursions. Assuming a normal distribution, we can specify the confidence interval of A_{SJ} using a t-statistic:

$$\epsilon_v = t(\alpha, L-1) \cdot \frac{s_{v,L}}{\sqrt{L} \cdot \overline{v_L}} \tag{8}$$

with

$$\overline{v_L} = \frac{1}{L} \sum A_{SJ,i} \tag{9}$$

$$s_{v,L} = \sqrt{\frac{\sum A_{SJ,i}^2 - \overline{v_L} \cdot \sum A_{SJ,i}}{L-1}} \tag{10}$$

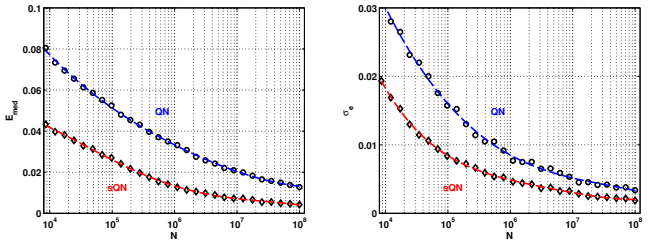
where ϵ_v is the estimated confidence bound of a t-distribution with confidence level α=0.95 and L−1 degrees of freedom. It is proportional to the empirical standard deviation $s_{v,L}$ and normalized by the empirical mean $\overline{v_L}$. If ϵ_v falls below the target deviation ϵ_{conf}, the JTOL algorithm converges.

The length L of the array v_{SJ} is variable. It is continuously incremented at every recursion and truncated if a subset of its newest elements exhibits less statistical variation. This condition is verified by searching the minimum epsilon value over all lengths:

$$\epsilon_{v,min} = \min\{\epsilon_v,\ v_{SJ}[L-k \dots L-1] \mid k{=}2 \dots L\} \quad (11)$$

The minimum epsilon search yields an optimistic estimate of the actual statistical confidence of A_{SJ} values. It allows us to quickly change to a higher sample size N as soon as the observed optimistic tolerance $\epsilon_{v,min}$ falls below a known comparison threshold. Hence, we optimize the algorithm with respect to a minimum number of recursions.

The ideal comparison threshold corresponds to the expected error of the tail fitting method. We can only approximate this error behavior, because the 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, we expect collected jitter distributions to consist of a bounded sinusoidal component combined with Gaussian random jitter. For this case in [10] worst case distribution shapes with combined sinusoidal and Gaussian jitter components were derived. The worst case error of both fitting methods can thus be represented as a simplified function of sample size N. In fig. 5 the median error E_{med} (left) and corresponding standard deviation σ_e (right) curves are plotted. The true TJ_{pp} value was approximated



(a) Median error E_{med} (b) Standard deviation σ_e

Fig. 5. Worst case error behavior of tail fitting methods over varying N. $K{=}250$ evaluations, 4^{th} order regression polynomials.

numerically and $K{=}250$ tail fits were carried out for each of the different sample sizes. 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} can not be compensated by the JTOL algorithm since the underlying distribution shape is basically unknown, but the standard deviation σ_e can be used as a pessimistic indicator for choosing the right sample size N. The logarithmic scaling in fig. 5(b) leads to a polynomial

$$f_p(N) = p_0 + p_1 \cdot \log(N) + \dots + p_4 \cdot \log(N)^4 \quad (12)$$

with the coefficients from table I. In our case the covered range is $N{=}[10^4 \dots 10^8]$. In order to compare the actual confidence interval $\epsilon_{v,min}$ with the expected error of the fitting method,

TABLE I
POLYNOMIAL REGRESSION COEFFICIENTS FOR σ_e.

Alg.	p_0	p_1	p_2	p_3	p_4
QN	0.2036	-0.03269	0.001823	$-3.466{\cdot}10^{-5}$	0.0
sQN	0.3493	-0.08615	0.008218	$-3.530{\cdot}10^{-4}$	$5.71{\cdot}10^{-5}$

we can thus use the error polynomials and formulate the condition:

$$\epsilon_{v,min} < \epsilon_{conf} \cdot \frac{f_p(N)}{f_p(N_{max})} \quad (13)$$

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 (fig. 4) this condition decides whether the obtained A_{SJ} estimates are sufficiently accurate, so that jitter distributions of higher sample size N have to be collected. 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 algorithm structure guarantees for a strictly monotonic increase of N until N_{max} is reached. At $\epsilon_{v,min}{<}\epsilon_{conf}$ and $N{=}N_{max}$ the convergence criterion is met. Note, that in order to identify a complete jitter tolerance curve, the JTOL algorithm from fig. 4 must be repeated for every desired frequency value f_{SJ}. Some additional speed up can thus be achieved, when the amplitude result of the last frequency is used as initial value for the next one.

III. Application Example

In this section we apply the developed JTOL algorithm to a behavioral model for charge-pump PLLs, which has been developed in [13]. First we introduce the key parameters of the model and provide an example for the JTOL algorithm at a single jitter frequency f_{SJ}. Then we proceed to complete jitter tolerance curves and performance comparisons for different model parameter configurations.

The modeled charge-pump PLL has been optimized with respect to fast and accurate time domain simulations on a system level, in order to allow for the statistical jitter analysis. It consists of an Alexander type phase detector (BB-PD), a charge-pump, a loop filter (LF) with subsequent gain regulator and the voltage controlled oscillator (VCO) (fig 6). The key pa-

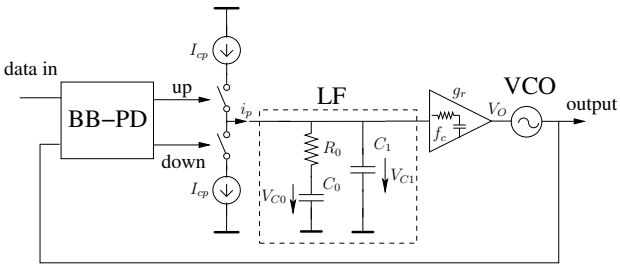


Fig. 6. Block scheme of the charge-pump PLL model.

rameters are listed in table II. They include analog component values, VCO properties with a noise model (Leeson process),

a gain regulator pole, phase detector delay and metastability range, as well as the applied data pattern for compliance testing. The presented PLL has also been realized as test structure for a 3Gbps high-speed interface.

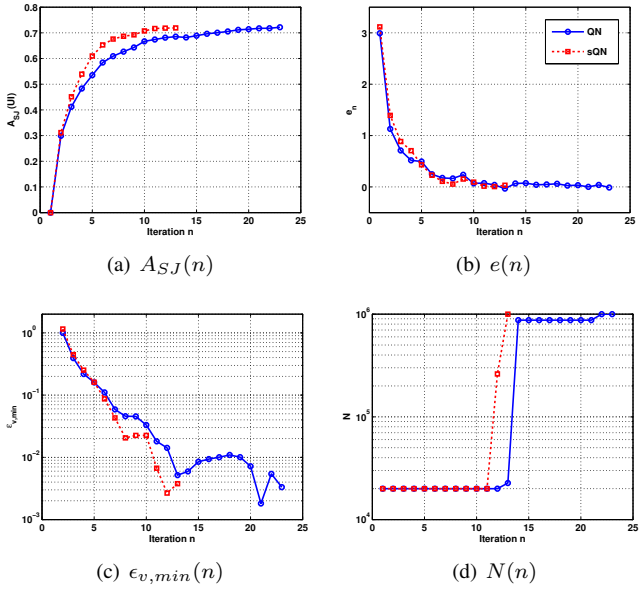
TABLE II
DEFAULT MODEL PARAMETERS FOR FIG 6.

Component	Default parameter values
Loop filter	R_0=700Ω, C_0=70pF, C_1=2pF
Charge-pump	I_{cp}=5μA
VCO	K_v=2.7GHz/V, f_{fl}=10MHz A_1=−120dBC @ f_1=10MHz A_{PhN}=−138dBC
Gain Regulator	g_r=1.0, f_c=250MHz
Phase detector	t_{del}=150psec, V_{meta}=±1mV
Data pattern	Pat=LBP [2] at 3Gbps

TABLE III
JTOL ALGORITHM SETTINGS.

N_{min}=$2{\cdot}10^4$	N_{max}=10^6	ϵ_{conf}=0.005	μ=0.11

In fig. 7 the JTOL algorithm behavior is demonstrated with the described PLL model at f_{SJ}=10MHz, using the settings from table III. For the QN method (solid curves) the first twelve iterations are carried out with N_{min} 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 eq.(13). Over successive iterations, the N parameter increases monotonically toward N_{max} until finally $\epsilon_{v,min}{<}\epsilon_{conf}$ is reached. The sQN method (dashed curves) in fig. 7 needs fewer iterations and less jitter samples, and thus converges significantly faster. This is due to the more accurate tail fitting principle, which also means less undesired error bias (fig. 5(a)).



(a) $A_{SJ}(n)$ (b) $e(n)$

(c) $\epsilon_{v,min}(n)$ (d) $N(n)$

Fig. 7. Example for the JTOL search algorithm, fig. 4.

The learning rate parameter μ must be selected in a way to support quick convergence. If it is too large, A_{SJ} values will exhibit large statistical variations and the JTOL algorithm will require many iterations before the obtained confidence interval is sufficiently low. Otherwise if μ is too small, the JTOL search will converge before A_{SJ} reaches the true value, simply because amplitude variations also become very small. The ideal μ depends on the dynamics of the investigated system and thus changes with varying jitter frequency f_{SJ} and parameter settings. In fig. 8 we estimated the convergence performance of the JTOL algorithm over varying μ. Therefor,

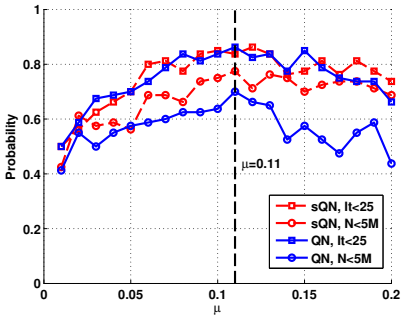


Fig. 8. Probability for successful JTOL algorithm runs.

we used four different model parameter settings, each at twenty frequency points (the same as in fig. 9) and counted successful JTOL runs. The two criteria for successful JTOL analysis include an iteration number of $I{<}25$ and overall sample size $\overline{N}{<}5$M. For the present high-speed PLL we identified a best suited learning rate of μ=0.11.

A minimum sample size of $N_{min}{=}2\cdot10^4$ proved sufficient for initial tail estimates. $N_{max}{=}10^6$ forms a trade-off between fitting accuracy and simulation time, since the given PLL model simulates 10^6 bits within one minute on a 2.4GHz AMD Opteron workstation. Here, we accept a worst case error bias of 1.3% for the sQN method (see fig. 5(a)). The confidence interval ϵ_{conf} is chosen small enough so that the final statistical variance of A_{SJ} can be neglected.

For the given PLL model we assume two different analysis scenarios. First, a software based scenario uses a simulator time resolution of 1fsec, which yields a resolution of $R_{sim}{=}3.3{\cdot}10^5$ for the 3Gbit interface. That is, the unit interval and corresponding distributions are divided into a discrete amount of R_{sim} bins. Second, a hardware simulation scenario reduces this value to R=128, which allows us to consider the limited precision of TIA based measurements.

In tables IV and V we compare the performance of the JTOL algorithm over different model parameter settings. We investigate three fitting principles: QN, sQN, and non-adaptive sQN with a constant sample size $N{=}N_{max}$ (c-sQN). The overall sample size $\overline{N}$ is measured over twenty JTOL analysis runs of varying jitter frequency, in a range of $f_{SJ}{=}[10^6 \ldots 10^8]$ MHz. This value directly reflects the measurement or simulation time needed to identify a complete JTOL curve. The sQN method yields best results with minimum $\overline{N}$ values, besides the R_0=400Ω case where the algorithm suffers from a convergence problem, caused by a few frequencies with

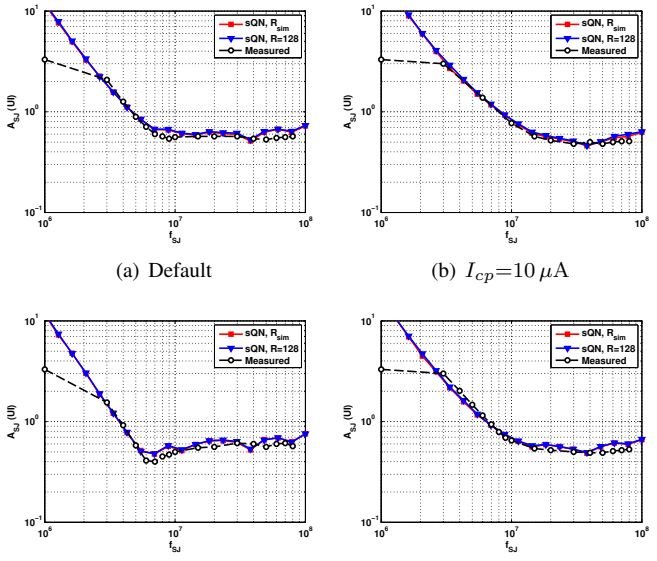
TABLE IV
JTOL ANALYSIS RESULTS FOR $R{=}R_{sim}{=}3.3 \cdot 10^5$.

Param.	Alg.	Default	$\mathbf{I_{cp}}{=}10\mu\mathrm{A}$	$\mathbf{R_0}{=}400\Omega$	$\mathbf{K_v}{=}4\mathrm{GHz/V}$
$\overline{\mathbf{N}}$ (M)	QN	67.6	98.7	**72.2**	85.4
	sQN	**44.0**	**74.8**	79.0	**68.3**
	c-sQN	174	293	225	236
$\overline{\mathbf{t_c}}$ (s)	**QN**	**22.6**	**34.0**	**25.4**	**30.4**
	sQN	476	862	630	688
	c-sQN	898	1404	1284	1328
$\overline{\mathbf{I_f}}$	QN	12.8	20.0	15.7	17.3
	sQN	12.6	19.7	13.4	16.4
	c-sQN	**8.7**	**14.7**	**11.3**	**11.8**

TABLE V
JTOL ANALYSIS RESULTS FOR $R{=}128$.

Param.	Alg.	Default	$\mathbf{I_{cp}}{=}10\mu\mathrm{A}$	$\mathbf{R_0}{=}400\Omega$	$\mathbf{K_v}{=}4\mathrm{GHz/V}$
$\overline{\mathbf{N}}$ (M)	QN	122	**95.6**	**68.5**	116
	sQN	**76.7**	107	84.3	**70.5**
	c-sQN	204	296	219	258
$\overline{\mathbf{t_c}}$ (s)	**QN**	$\mathbf{< 0.08}$	$\mathbf{< 0.04}$	$\mathbf{< 0.04}$	$\mathbf{< 0.08}$
	sQN	1.10	1.67	1.30	1.31
	c-sQN	0.91	1.16	1.06	1.09
$\overline{\mathbf{I_f}}$	QN	16.1	19.1	14.0	17.6
	sQN	15.5	19.5	16.7	15.9
	c-sQN	**10.2**	**14.8**	**11.0**	**12.9**

highly unstable distributions. $\overline{t_c}$ is the overall computation time consumed by tail fitting, and is thus a measure for the complexity of the fitting algorithm. In our model, QN is more than one order of magnitude faster than sQN. $\overline{I_f}$ describes the average number of iterations per jitter frequency. In tab. IV the $\overline{I_f}$ values for sQN are only slightly better than for QN (unlike the clear difference in fig. 7) because the initial amplitude is taken from the result of the previous frequency and thus, already close to the final result. The constant sample size method c-sQN yields the best $\overline{I_f}$ due to the constant error bias (see fig. 5(a)) of the fitting algorithm.



(a) Default (b) $I_{cp}{=}10\,\mu$A

(c) $R_0{=}400\Omega$ (d) $K_v{=}4$GHz/V

Fig. 9. JTOL curves at different parameter settings.

In fig 9 the obtained jitter tolerance curves for the sQN method with both R_{sim} and $R{=}128$ are plotted together with manually measured JTOL curves of the same PLL hardware structure. Note that the measurements have an amplitude limit of $A_{SJ,max} = 3.3$UI (reached at $f_{SJ}{=}1$MHz), and that their differences with simulated curves are given by model inaccuracies which do not reflect algorithm performance. The hardware oriented model simulation with only $R{=}128$ bins matches excellent with the R_{sim} high resolution scenario and can handle the different loop dynamics over varying jitter frequency also very well.

A complete JTOL measurement over twenty frequency points with the QN method typically requires $N{\approx}100$M samples ($\approx$1.5h of simulation time), which results in $\overline{t_N}{\approx}33$ms of hardware measurement time for the 3Gbit interface. Together with $\overline{t_c}{\approx}80$ms for calculations and 1ms additional time buffer per iteration we yield $\overline{t_I}{\approx}\overline{I_f} \cdot 20 \cdot 1\mathrm{ms}{=}320$ms which gives an overall time consumption of $\overline{t_N}{+}\overline{t_c}{+}\overline{t_I} = 433$ms for the complete JTOL curve.

IV. Conclusion

We presented a fast and accurate method for the identification of jitter tolerance curves in high-speed PLLs. An adaptive algorithm determines jitter amplitudes recursively, while optimized with respect to a small number of iterations and sample size. We demonstrated, that the proposed method can be applied to both software simulations and hardware measurements.

Acknowledgment

This work was funded by the Austrian research initiative FIT-IT under the project acronym JUDY.

References

[1] *Serial ATA Specification*, S-ATA Std., Rev. 2.6, 2006.

[2] B. Ham, "Fiber Channel – Methodologies for Jitter and Signal Quality Specification," INCITS, Tech. Rep., Jun. 2005.

[3] Y. Fan, Y. Cai, L. Fang, A. Verma, W. Burchanowski, Z. Zilic, and S. Kumar, "An Accelerated Jitter Tolerance Test Technique on ATE for 1.5GB/S and 3GB/S Serial-ATA," *IEEE ITC*, pp. 1–10, Oct. 2006.

[4] Y. Fan and Z. Zilic, "Accelerating Jitter Tolerance Qualification for High Speed Serial Interfaces," in *IEEE ISQED*, 2009.

[5] 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 ITC*, pp. 717–725, 2002.

[6] 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*, 2005.

[7] S. Ahmed and T. Kwasniewski, "Efficient Simulation of Jitter Tolerance for All-Digital Data Recovery Circuits," *IEEE MWSCAS*, pp. 1070–1073, Aug. 2007.

[8] A. Chan and G. Roberts, "A jitter characterization system using a component-invariant Vernier delay line," *IEEE Transactions on VLSI Systems*, vol. 12, no. 1, pp. 79 – 95, jan. 2004.

[9] 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. on VLSI Systems*, vol. 17, no. 12, pp. 1698 –1708, Dec. 2009.

[10] S. Erb and W. Pribyl, "Comparison of Jitter Decomposition Methods for BER Analysis of High-Speed Serial Links," in *DDECS*, Aug. 2010.

[11] S. Haykin, *Adaptive Filter Theory*. New York: Wiley, 2002.

[12] M. P. Li and J. Wilstrup et al., "A New Method for Jitter Decomposition Through Its Distribution Tail Fitting," *IEEE ITC*, 1999.

[13] S. Erb and W. Pribyl, "A Behavioral Modeling Approach for Jitter Analysis in Charge-Pump PLLs," *Austrochip Workshop*, 2009.