Efficient Time-Domain Simulation of Telecom Frontends Using a Complex Damped Exponential Signal Model

Piet Vanassche, Georges Gielen, Willy Sansen
Katholieke Universiteit Leuven
ESAT Laboratories
piet.vanassche@esat.kuleuven.ac.be

Abstract

This paper presents an efficient time-domain simulation approach for telecommunication frontends at architectural level. It is based upon the use of complex damped exponential modeling functions. These allow to construct accurate signal models for digitally modulated telecom signals, requiring only few modeling functions. Since these models are valid over a long range of time, they allow for a large timestep, which greatly speeds up time-domain simulation of the telecom frontends. Details of a simulation approach based upon this signal model are discussed. The approach is verified by experimental results.

1. Introduction

During the last decade, the telecommunication market is experiencing a tremendous growth. This includes both a growth in the number of consumers as in the number of applications. DECT, GSM, Bluetooth, ADSL and VDSL are just a few examples of evolving technologies that try to cope with the increasing demand for higher data rates. This makes it a very attractive and very competitive market, which requires short time to market in introducing new products. In order to deal with these requirements, designers are in need of efficient tools for evaluation (simulation) of their telecommunication systems at all levels of abstraction. This paper describes a technique that is particularly well suited for simulation at architectural level of the frontend part of these telecom systems.

Looking at the simulation algorithms that are currently available [2], it is seen that their performance strongly depends upon the problem at hand. For example, harmonic balance outperformes a SPICE-like approach when dealing with RF circuits and a periodic input, while SPICE-like techniques are better when dealing with strongly nonlinear baseband applications, even when the input signals are periodic. The reasons for this are inherent to the choices made when constructing the simulation algorithm. In order to identify the underlying mechanisms, we need to look at the overall structure of these algorithms. This structure is outlined in Fig. 1, which basically shows a template for the solution of a set of differential algebraic equations (DAE's)

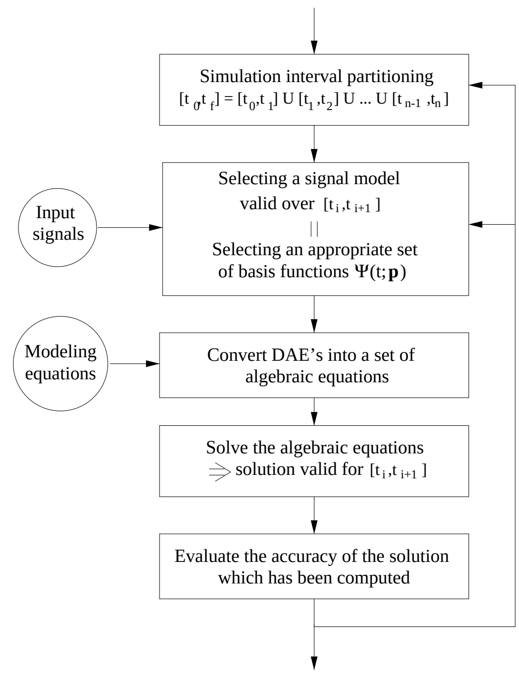

Figure 1. Global structure of a simulation algorithm.

$$\mathbf{F}\left(\frac{d\mathbf{x}}{dt}, \mathbf{x}, \mathbf{x}_{in}, t\right) = 0 \tag{1}$$

valid over the range $\left[t_0, t_f\right]$. Here $\mathbf{x}(t)$ and $\mathbf{x}_{in}(t)$ are respectively the unknown signals and the input signals. The figure clearly illustrates how the global algorithm can be decomposed into several subalgorithms. Different simulation algorithms provide different implementations for one or more of these subalgorithms. These choices determine the overall algorithmic performance.

An important step in the design of a simulation algorithm lies in the selection of the interval partitioning and signal modeling strategies and the interaction between them. Here, the first subalgorithm partitions the overall simulation interval $[t_0, t_f]$ into a number of subintervals $[t_k, t_{k+1}]$ within which a solution is easier to compute. The basic idea is that the signal variations become easier to model, and hence easier to compute, when the time interval is small and vice versa. The second subalgorithm selects a set of differentiable basis functions $\Psi_{ij}(t)$ for $j = 1, \ldots, J_i$ such that the signals $x_i(t)$ can be modeled as

$$x_i\,(t) = \sum_{j=1}^{J_i} X_{ij}\,\Psi_{ij}\,(t) \tag{2}$$

within the subinterval $\left[t_k, t_{k+1}\right]$ and this with sufficient accuracy. This signal modeling step turns equation (1), which is essentially a problem in an infinite number of variables, into the computation of the finite number of unknowns X_{ij}. From a computational point of view, this results in a complexity given by

$$C_{[t_0,t_f]} = N_T \times \overline{C}_{[t_k,t_{k+1}]} \tag{3}$$

where N_T is the number of subintervals and $\overline{C}_{[t_k,t_{k+1}]}$ is the average computational complexity involved in computing a solution over one of the subintervals. The latter complexity strongly depends upon the choice of the signal models. This choice determines the total number $J = \sum_{i=1}^{N} J_i$ of unknown variables X_{ij} that need to be computed for each subinterval $\left[t_k, t_{k+1}\right]$, and hence the size of the resulting systems of nonlinear and linear equations. Here, N is the total number of unknown signals $x_i(t)$. In order to keep J as low as possible, *it is important to select basis functions which correspond as closely as possible to the characteristics of the signals. These models may differ from signal to signal.*

As the interaction between the first two steps in Fig.1 is concerned, there are two limiting strategies that can be pursued in order to meet the required accuracy: fix the complexity (the J_i) per subinterval and adapt the size of the subintervals $\left[t_k, t_{k+1}\right]$ (the timestep), or fix the size of the subintervals and increase the number of modeling functions. SPICE for example applies the first strategy, while harmonic balance applies the second one. It is however also possible to choose a strategy in between these two extremes. This implies a trade-off between the number of subintervals and the computational complexity per subinterval. Besides the selection of the basis functions $\Psi_{ij}(t)$, this trade-off is an important parameter that can be used in the performance optimization of simulation algorithms. The optimum strategy will however often depend upon the system characteristics (like the degree of nonlinearity, etc.).

Traditional SPICE-like solvers [11] typically select a 2nd-order polynomial model and decrease the size of the subintervals (timestep) in order to ensure the accuracy of the signal model. This is often combined with adaptive stepsize control. In [13], a polynomial-based approach is presented, that allows you to make a trade-off between timestep and polynomial order. When dealing with telecom systems and their associated IF and RF modulated signals, these approaches however lead to a huge number N_T of subintervals (small timestep), resulting in long simulation times. The basic reason for this is that the polynomial basis is not suited for modeling high-frequency modulated signals. Harmonic balance [3] and its variants [5] remedy this by using the harmonic basis $\left\{e^{ji2\pi f_0 t}\right\}$ instead. This allows an efficient modeling of high-frequency, periodic signals. The number of harmonics is chosen in a way to make the signal model valid over the entire simulation interval $[0, T]$, with T the period of the input signals. This makes $N_T = 1$. These methods however only allow to find the steady-state solution of nonlinear circuits with periodic inputs. No transients can be taken into account and it is not possible to model the digitally modulated signals that serve as the actual inputs to telecommunication systems. The circuit envelope simulation technique [6] solves this problem in a way that can be seen as combining the polynomial and the harmonic basis. It is often combined with some kind of adaptive stepsize control. The algorithm however uses a global signal model, meaning that the basis functions are the same for all signals in the frontend. This implies that the simulator cannot take advantage of the sometimes widely different signal characteristics at different nodes in the frontend, something which is typical in telecom applications (high-frequency input, low-frequency output, or vice versa). This global model leads to an unnecessary increase in the number of basis functions per subinterval $\left[t_k, t_{k+1}\right]$ and hence in the number of unknowns, complicating the solution

This paper presents a simulation approach for weakly nonlinear telecom frontends at architectural level which tries to incorporate the advantages of the previous methods while avoiding the drawbacks. It makes use of a complex damped exponential signal model, which allows for efficient modeling of high-frequency modulated signals. Compared to harmonic balance, the introduction of the damping factor makes it possible to perform simulations in the time-domain and this for non-periodic input signals. The simulation algorithm is based on a runtime Volterra series expansion. It selects the necessary basis functions at runtime, avoiding the global signal model used in the circuit envelope method. The simulation stepsize can be used as a parameter to optimize the simulation performance. The algorithm also allows to compute wanted and unwanted signals seperately. This, together with the natural relationship between the exponential signal model and frequency content, greatly facilitates analysis of the results afterwards. It should provide designers with the information they need to make decisions at the architectural level.

The paper is organized as follows. Section 2 introduces the complex damped signal model. It is shown to be very efficient in modeling the high frequency digitally modulated signals involved in telecom frontends. In section 3, a time-domain simulation algorithm based upon this signal model is discussed. The algorithm is absolutely stable by construction and avoids the necessity of a global signal model. Experimental results are presented in section 4 and conclusions are drawn in section 5.

2. The complex damped exponential basis and its signal modeling capabilities

One of the important properties of telecommunication signals is the fact that they often contain greatly different time constants, especially when dealing with RF applications. A set of basis functions that has the natural ability to deal with this property is the complex damped exponential basis. In its most general form, a signal $x(t)$ is modeled as

$$x(t) = \sum_k A_k t^{n_k} e^{z_k t} \qquad (4)$$

with $n_k \in N$ and $z_k = p_k + j\omega_k \in C$. This signal model shows great resemblances with the harmonic basis $\left\{e^{jn2\pi f_0 t}\right\}$. The latter is actually a subset of the former. There are however some important differences. By adding the damping factor p_k, it becomes *possible to model real-life telecom signals* like OFDM, GMSK, etc. The harmonic basis on the other hand only allows periodic input signals. It also becomes *possible to perform simulations in the time-domain and to take transients into account.* In what follows, we'll mostly limit ourselves to the subset of (4) for which $n_k = 0$.

In order to get an idea of the modeling efficiency of the exponential basis with respect to telecommunication signals, we first compare it with the results obtained using a polynomial basis. For testing purposes, we use a GMSK modulated bitstream. Similar results can be obtained for other kinds of modulation strategies. Such a signal can be written as

$$\begin{aligned} s(t) &= Re\left[e^{j2\pi f_0 t} \times (\cos(\Phi(t)) + j\sin(\Phi(t)))\right] \\ &= Re\left[e^{j(2\pi f_0 t + \Phi(t))}\right] \end{aligned}$$

where $\cos(\Phi(t))$ and $\sin(\Phi(t))$ are the in-phase and quadrature signal components respectively. The GMSK

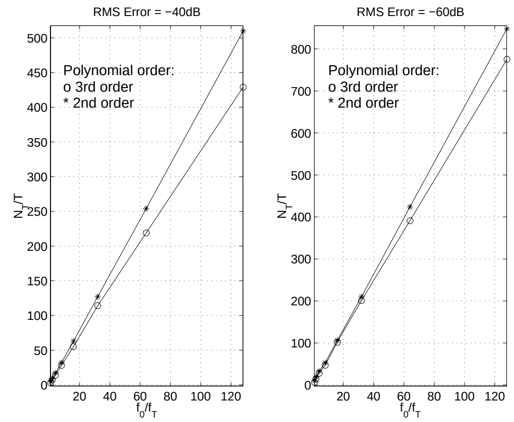

Figure 2. Number of subintervals necessary for a polynomial fit of a modulated GMSK signal.

symbol rate equals $f_T = 1/T$. For both signal modeling approaches (exponential and polynomial), J_{GMSK} and N_T were computed for varying values of the carrier frequency f_0. All polynomial fits are performed using a least-squares approach, while the exponential fitting was done using the HTLS algorithm described in [9]. The exponential fit required $J_{GMSK} = 4$, $N_T/T = 1$ and this for an RMS error equalling -40dB and $J_{GMSK} = 6$, $N_T/T = 1$ for an error of -60dB. The error was computed as the difference between the input samples and the resulting fit. *All of these numbers are independent of the modulation frequency* f_0. Fig.2 shows the resulting (normalized) number of subintervals N_T/T for the corresponding polynomial fits using 2nd and 3rd-order polynomial models. As is to be expected, N_T increases (decreased timestep) about linearly with the frequency f_0. The complexity of the corresponding simulation algorithm will hence also grow linearly with f_0. Since the exponential basis does not suffer from this drawback, it clearly provides a much more efficient signal model than polynomials, especially when f_0/f_T gets large (which is typically the case).

A second experiment demonstrates that also when using exponentials, there is a trade off possible between the length of the timestep (length of the modeling subinterval $[t_k, t_{k+1}]$) and the complexity of the signal model within $[t_k, t_{k+1}]$, as was discussed in section 1. Using the same GMSK signal (with $f_0 = 10 \cdot f_T$), Fig.3 plots the number of needed modeling exponentials versus the normalized timestep $\Delta t/T$. This is done for two different values of the accuracy of the resulting fit. Again, these numbers are independent of the carrier frequency f_0. This figure clearly illustrates how it is possible to increase the timestep (decrease the number N_T of subintervals) by increasing the number of basis functions (modeling complexity). Which

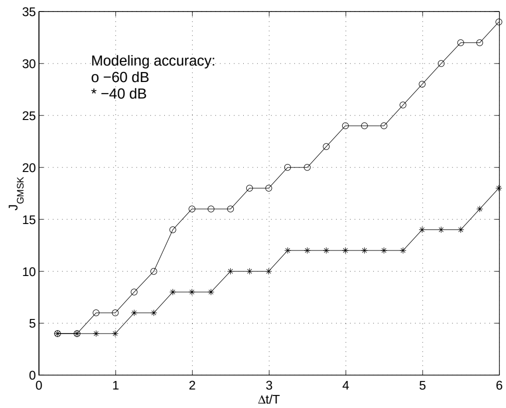

Figure 3. Number of modeling exponentials versus the length of the modeling interval.

choice of Δt is optimal depends upon the system degree of nonlinearity as will be described in section 4.

As a final note, it is worth mentioning that one could argue that the exponential method uses both A_k and z_k in (4) to obtain a good fit, effectively doubling the number of unknowns that have to be computed. The exponents z_k can however be determined on beforehand, based upon the knowledge of the input signals and their harmonics. Another approach would be to determine them using the results of some short trial simulations. Doing so allows us to *incorporate the input signal characteristics into the simulation algorithm*. Stated in another way, the basis functions $e^{z_k t}$ are chosen, before starting the actual simulation, because they are suited for modeling the signals that arise when a given set of input signals is applied. This is similar to harmonic balance where the functions $e^{j2\pi t/T}$ are chosen because they are very well suited to model T-periodic functions.

3. A simulation algorithm based upon the complex damped exponential basis

Having demonstrated the efficiency of the complex damped exponential basis in modeling telecom signals, we now briefly outline a simulation approach for weakly nonlinear telecom systems using this basis. The algorithm is absolutely stable by construction and avoids the necessity of a global signal model. The restriction of weak nonlinearity can be justified by the fact that telecommunication frontends are essentially designed to behave linearly (in its most general, time-varying, sense). The nonlinear behavior is actually parasitic and hence suppressed as much as possible. This makes telecom frontends weakly nonlinear in nature. In what follows, we also assume that the telecom system can be modeled as an interconnection of linear blocks and

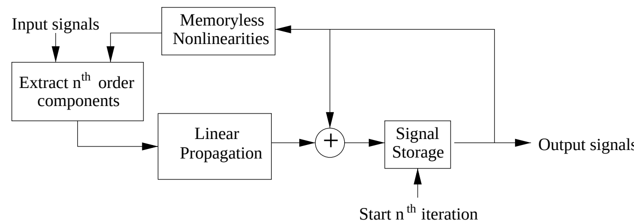

Figure 4. Runtime Volterra series expansion procedure for calculating the distortion components

static (memoryless) nonlinearities. This also poses no great restrictions, especially when dealing with high level models.

Linear signal propagation is straightforward. For a given state space representation $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ of a linear time-invariant system, an initial condition $\mathbf{x}_0$ and an input signal e^{zt}, the output can be written as (assuming the signal pole z to be different from the system poles, being the eigenvalues of the matrix $\mathbf{A}$)

$$\begin{aligned} \mathbf{x}(t) &= e^{\mathbf{A}t}\left(\mathbf{x}_0 - (z\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\right) \\ &\quad + (z\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}e^{zt}d\tau \qquad (5) \\ y_{out}(t) &= \mathbf{C}^T\mathbf{x}(t) + \mathbf{D}e^{zt} \qquad (6) \end{aligned}$$

which can easily be rewritten as a sum of complex damped exponentials by making use of the eigenvalue decomposition of the system matrix $\mathbf{A}$. The resulting propagation through the linear blocks is hence both fast and accurate. At this point, it can be noted that, because of the choice of the basis functions, the algorithm will be absolutely stable.

In order to deal with the nonlinear behavior, we use a runtime Volterra based expansion procedure as outlined in Fig.4. This procedure is analogous to the ones used in symbolic analysis techniques [12]. The signal components are computed in increasing order of nonlinearity. In a first step, the wanted signal components are propagated through the linear blocks. Next, these signals are propagated through the static nonlinearities of the different blocks and the 2nd-order components (products of 2 1st-order components) are computed at the outputs of the nonlinearities. These 2nd-order components are then propagated through the linear blocks resulting in the 2nd-order nonlinear components of the output signals. This procedure continues with the 3rd-order nonlinearities, etc. until the desired degree of nonlinearity has been reached. Here it should be reminded that we assumed the frontend model to be an interconnection of linear blocks and static nonlinearities. It is also clear that this procedure will only work for weak degrees of nonlinearity.

This procedure has the advantage of computing the basis functions for the signal models at runtime, avoiding the necessity of a global signal model. Also, because of the natural relationship between the exponential basis functions and frequency contents, it is easy to neglect (at runtime) frequency components which are not of interest, saving CPU time. This relationship also greatly facilitates the analysis afterwards. Finally, the Volterra series expansion has the advantage that it allows to compute wanted and unwanted components seperately, something that is impossible when using for example SPICE-like methods. This property also proves useful for designers in analyzing the simulation results.

One possible pitfall when using this approach, is an explosion of the necessary number of basis functions J. Given the number of exponentials J_{in} necessary to model the input signals and the system degree of nonlinearity nl, an upper bound for the total number of exponentials is given by

$$J \leq \frac{(J_{in} + nl)!}{J_{in}!nl!} \approx \frac{J_{in}^{nl}}{nl!} \qquad (7)$$

This shows that as nl increases, the number of basis functions tends to explode exponentially. This can however be controlled in several ways. First, it is possible to decrease J_{in} by increasing the number of subintervals N_T as illustrated in Fig.3. Other strategies involve a simplification of the input and output of the nonlinear blocks by pruning the exponentials with neglegible power or by using a sample and refit procedure. In the latter case, large sampling rates are avoided by applying the procedure to each frequency band separately. Separating the signals according to their frequency content is a trivial operation, as was already mentioned before. These simplifications introduce some errors, which however, if small enough, are neglegible effects since the nonlinearities themselves are small compared to the wanted signal components.

Finally, we mention the techniques that are used to switch between a sampled data and exponential representation. This conversion is useful when modeling the input signals, when reducing the number of exponentials through sample and refit or when dealing with hard nonlinearities (which are often easier to deal with in the sampled-data representation). The current implementation of this conversion is based upon the Hankel Total Least Squares (HTLS) exponential fitting algorithms presented in [9]. This algorithm allows to compute the minimum number of exponentials necessary to model the input data for a given accuracy. No discretization of the frequency axis is necessary as is the case with the Gabor transform. This makes it very well suited to deal with signals for which the ratio of the time constants is non-rational. These Hankel data matrix based algorithms also slightly outperform linear prediction (LP) fitting procedures as far as accuracy is concerned [1].

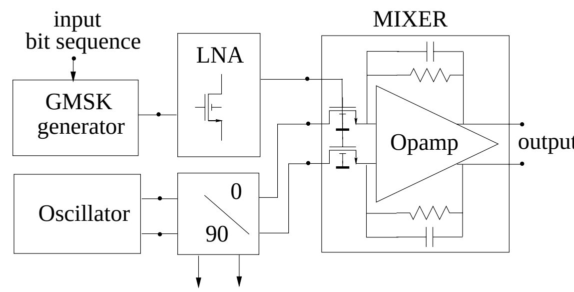

Figure 5. Schematic of the DCS1800 receiver

4. Experimental results

In order to demonstrate the efficiency of our approach, we present the results from the analysis of a low-IF DCS1800 receiver [7] using the simulation approach discussed above. All of the algorithms are implemented in Matlab$^{\text{TM}}$. The results were obtained on a Sun ULTRA30. Both CPU times and flop counts are presented.

The DCS1800 receiver is schematically depicted in Fig.5. The RF input is a GMSK signal according to the GSM specifications, with a symbol length $T = 3.7\mu s$ at a carrier frequency somewhere around $f_c = 1.8$ GHz. The low-noise amplifier (LNA) is a 12th-order small-signal model extracted from a SPICE netlist. For the mixer opamps, a two-stage macromodel was constructed from the circuit schematic where each stage is modeled using a transconductance plus a resistive and capacitive load. The transconductances are nonlinear such that the mixer has an IIP3 of 14dBm. The oscillator is taken to be an ideal cosine. Note that Fig.5 only shows the in-phase mixer.

In a first step, an exponential model for the input signal is computed. This is done by fitting the signal's complex baseband equivalent $e^{j\Phi(t)} = \cos(\Phi(t)) + j\sin(\Phi(t))$, the result of which is shown in Fig.6. Next, this is upconverted to $f_c = 1.8$ GHz. It takes about 0.1 seconds of CPU time and 1 Mflops (per GMSK symbol) to compute them. Two different models were constructed: one that is accurate up to -40dB and one up to -60 dB. As was already illustrated in figure 3, the higher accuracy is at the cost of a more complex model (more exponentials). Since these models need to be computed only once (they can be reused for different simulations), the corresponding CPU times and flop counts are not taken into account in subsequent simulation times.

Next, the input signal is propagated through the receiver, while varying the stepsize Δt of the simulation subintervals. This means that the exponential signal models at each node must be valid over a range Δt. This is done for both the mixer opamp nonlinearities turned on and off and for both levels of accuracy of the input signal models. When computing the nonlinear system behavior, the high-frequency components (at 3.6 GHz) are neglected, since their power

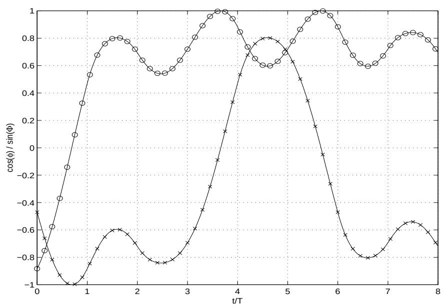

Figure 6. $\cos(\phi)$ **and** $\sin(\phi)$ **as a function of time for the GMSK signal. The solid lines represent the sampled-data values. The x-marks and circles represent samples of the exponential fit of respectively** $\cos(\phi)$ **and** $\sin(\phi)$**.**

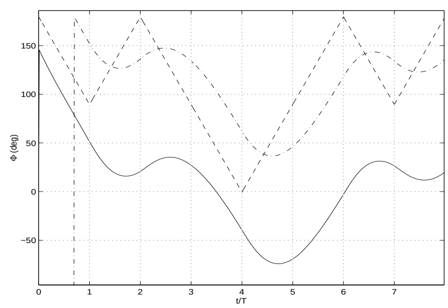

Figure 7. Input phase before (dashed line) and after (dashdot line) the Gaussian GMSK filter and the output phase extracted out of the mixer output signals (solid line).

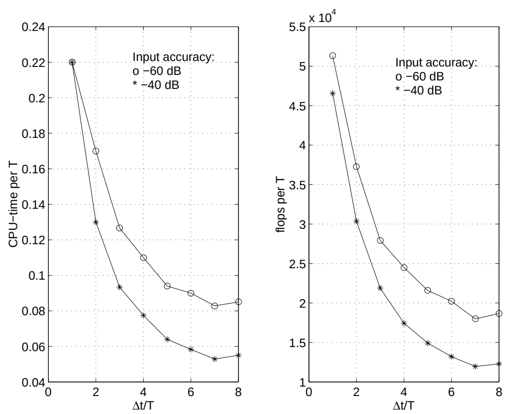

Figure 8. CPU times and flop counts per symbol period T **versus the stepsize** Δt **when the mixer opamp behaves strictly linear.**

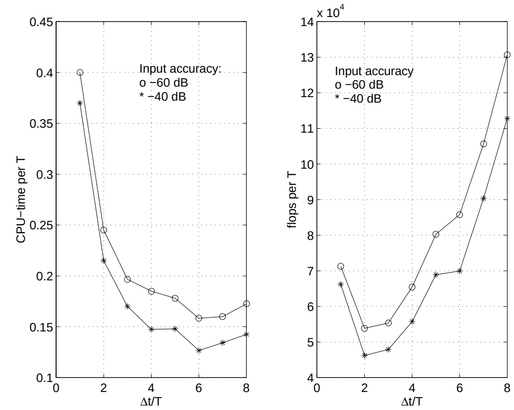

Figure 9. CPU-times and flopcounts per symbol period T **versus the stepsize** Δt **when the mixer opamp contains nonlinear behavior.**

is neglegible. In order to avoid explosion of the number of exponentials, signals are simplified using a sample and refit procedure. Fig.7 shows the input phase and the output phase, as extracted from the outputs of the in-phase and quadrature mixers. The resulting CPU-times and flopcounts per symbol period T are shown in the figures 8 and 9. These figures illustrate how the overall simulation time can be optimized by selecting an optimum value for the timestep Δt. They also show this optimum to depend upon the properties of the system being simulated. For simulation of the ideal system behavior only, the timestep Δt can be taken very large. The resulting decrease in the number of simulation subintervals is more important than the increased modeling complexity (number of exponentials) per subinterval. However, when the mixer opamps behave nonlinearly, the simulation algorithm generates extra exponentials to model the nonlinear signal components. For large Δt, this increase in signal modeling complexity becomes dominant and the overall simulation time starts to increase again. In this case it is suggested to decrease the timestep compared to the case where the opamp behaves linearly. The difference in optimal timestep in figure 9 between CPU times and flop counts is due to the extra overhead in Matlab function calls and memory management. Note that in both the linear an nonlinear case, the timestep Δt is still a multiple of the period T of the GMSK symbols.

In a second experiment, the exponential approach is compared to the SPICE-like Matlab integration method *ode15s*, which is a variable-order, adaptive stepsize method suited for solving stiff problems. The comparison was performed by applying a modulated GMSK signal to a opamp-RC lowpass filter (similar to the mixer in Fig.5, but with the MOSFETS replaced by their (unmodulated) resistance.

f_0/f_T	CPU gain	Flop gain	Acc./Dist. (dB)
4	8.0	8.2	-17
8	6.0	6.4	-20
16	11.3	12.4	-22
32	20.7	22.9	-23
64	42.4	44.9	-23
128	73.4	70.0	-25

Table 1. Comparison between classical polynomial-based integration methods and an exponential approach.

Opamp nonlinearities are included). The timestep for the algorithm based upon complex damped exponentials was chosen to be T, the length of one GMSK symbol. This step is independent of the carrier frequency f_0. The experiment is repeated for increasing values of the modulation frequency f_0. Table 1 presents the gain in both CPU time and flop count, obtained by using the exponential approach. The last column compares the energy contained in the difference between both methods and the energy contained in the signal distortion components (smallest relevant signal components). It is clearly seen that the performance gain increases about linearly with the carrier frequency. This is because of the increasing timestep needed by the *ode15s* integration method. The behavior of the CPU and flop gains for low modulation frequencies are explained by the fact that in these cases, the time step of the *ode15s* routine is determined by the impulse response of the amplifier, and not by the frequency content of the input signal.

5. Conclusions

The importance of telecommunication systems justifies the design of algorithms that increase simulation speeds by incorporating the system and signal properties into the algorithm. This paper has presented an approach based upon a complex damped exponential signal model. This set of basis functions incorporates the typical properties of digitally modulated telecom signals, with their many different time constants, in a natural way. This allows to construct simple models (few basis functions) that are valid over a long time-interval. This allows for an increased timestep, speeding up the simulations. The length of the timestep can be varied to optimize for speed. This exponential model was combined with a runtime Volterra series expansion based algorithm for simulation of weakly nonlinear telecom systems. The algorithm is absolutely stable and avoids the necessity of a global signal model. The algorithm also allows to compute wanted and unwanted signals separately. This, together with the natural relationship between the exponential signal model and frequency content, greatly facilitates analysis of the results afterwards by designers. The approach was verified experimentally with simulations on a DCS1800 receiver frontend.

6. Acknowledgements

This work has been supported in part by the Esprit project Salomon and the Flemish IWT.

References

[1] De Beer and Van Ormondt, “Analysis of NMR data using time-domain fitting procedures”, *In-vivo Magnetic Resonance Sectrocospy I: Probeheads, Radiofrequency Pulses, Spectrum Analysis* (M. Ruding and J. Seelig, Eds), pp. 201-248, Springer-Verlag, Berlin/Heidelberg, 1992

[2] K.Kundert, “Simulation Methods for RF Integrated Circuits”, *Proc. ICCAD*, pp.752-765, 1997

[3] K.Kundert and A.Sangiovanni-Vincentelli, “Simulation of Nonlinear Circuits in the Frequency Domain”, *IEEE Trans. on CAD*, Vol. CAD-5, No. 4, October 1986

[4] K.Kundert, J.K.White and A.Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*, Kluwer Academic Publishers, Dordrecht, 1990.

[5] O.J.Nastov and J.K.White, “Time-Mapped Harmonic Balance”, *Proc. DAC*, Session 36, 1999

[6] D.Sharrit, “New Method of Analysis of Communication Systems”, *MTTS’96 WMFA: Nonlinear CAD Workshop*, June 1996

[7] M.Steyaert et al., “A 2V CMOS Cellular Transceiver Front-End”, *Proc. of the ISSCC*, Session 16, 2000

[8] R.Telichevesky et al., “Fast Simulation Algorithms for RF Circuits”, *Proc. CICC*, pp. 21.1.1-21.1.8, 1996

[9] Sabine Van Huffel et al., “Algorithm for Time-Domain NMR Data Fitting Based on Total Least Squares”, *Journal of Magnetic Resonance*, A 100, pp. 228-237, 1994

[10] I.Vassiliou and A.Sangiovanni-Vincentelli, “A Frequency-Domain, Volterra Series-Based Behavioral Simulation Tool for RF Systems”, *Proc. CICC*, pp. 2.5.1-2.5.4, 1999

[11] J.Vlach and K.Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, 1983

[12] P. Wambacq, G.Gielen, P.Kinget and W.Sansen, “High-frequency distortion analysis of analog integrated circuits”, *IEEE Trans. on Circ. and Syst.*, vol. 46, no. 3, pp. 335-345, March 1999.

[13] B.Yang and J.Philips, “A multi-interval Chebyshev collocation method for efficient high-accuracy RF circuit simulation”, *Proc. DAC*, Session 11, 2000