DATA PREDISTORTION FOR NONLINEAR SATELLITE CHANNELS

Jakob Harmon, Stephen G. Wilson
Charles L. Brown Dept. of Electrical and Computer Engineering
University of Virginia
Charlottesville, VA 22904
{jbh5x, sgw}@virginia.edu

Abstract—Power amplifiers, an integral component to most wireless communication systems, are inherently nonlinear devices that introduce out-of-band spectral regrowth and constellation degradation to the amplifier output. One solution is to operate in a highly backed-off state to achieve quasi-linear performance. Operating in this region requires a higher-saturated power rating for a given output power and reduces DC-to-RF efficiency. A more effective solution is to apply digital predistortion which pre-compensates for the harmful nonlinear effects. Compensation can be done at the waveform level or operate at the symbol rate, known as data predistortion. In this work we study data predistortion for commonly adopted higher-order modulations, and study performance as a function of amplifier drive level relative to saturation, and as a function of the predistorter’s memory span. A math model for a nonlinear satellite with transponder filtering is employed.

I. INTRODUCTION

POWER amplifiers (PAs) operated in their most efficient regime, near saturation, are nonlinear, producing amplitude-to-amplitude modulation (AM-AM) conversion as well as amplitude-to-phase modulation (AM-PM) conversion of a narrowband signal. These effects have been known for many years in the context of traveling-wave tube (TWT) amplifiers, a common type of amplifier used in satellite communication. The nonlinearity induces several harmful effects, including intermodulation in multicarrier environments, and spectral regrowth in single-carrier settings. Out-of-band spectral power can then increase to the point of not meeting regulatory specifications. Moreover, the in-band distortion produces degradation in bit error rates (BER) relative to that of linear amplifiers.

Newer satellite transmission standards that use higher-order modulation schemes, e.g. DVB-S2, are increasingly vulnerable to these nonlinearities because of their high peak-to-average ratios. Complicating matters, PAs exhibit memory effects where the output not only depends on the current input, but also on the magnitude of previous values of the input, producing additional distortion. To meet frequency mask requirements of regulatory agencies and bit-error rate (BER) performance requirements of customers, many satellite system operators are being driven to confront the effects of nonlinear power amplifiers.

One solution is to backoff the amplifier, i.e. reduce the drive level so that operation is more confined to the quasi-linear range of amplification. The disadvantages of this approach are two-fold: 1) a higher saturated power rating is needed to achieve a given desired linear power output, and 2) the DC-to-RF power efficiency suffers greatly.

The technique of waveform predistortion employs an approximate inverse nonlinearity ahead of the power amplifier such that the cascade operation is close to ideal, i.e. the output amplitude is just some constant $G$ times the input amplitude, and phase at the output is the same as that of the input. The earliest predistorters were analog devices with RF feedback, which were difficult to align, prone to instability, and seldom performed well. More recently, digital signal processing (DSP) technology has been applied, along with adaptive training of the predistorter, to achieve substantial improvement in the linearization task. These systems act on a baseband I/Q discrete-time version of the input signal (hence the common name digital predistorter), producing a precompensated I/Q waveform that is converted to continuous-time, upconverted, and amplified by the PA.

An alternative is to apply predistortion at the symbol rate, prior to pulse shaping, to pre-warp the complex symbol sequence so that the resulting receiver constellation suffers less distortion due to nonlinearity. The simplest data predistorters simply warp the constellation in amplitude and phase to make gross compensation for AM-AM and AM-PM effects. However, the satellite channel

This research was supported by the Virginia Space Grant Consortium (VSGC) and iDirect.
has memory due to both pulse shaping and matched filtering, as well as transponder filtering surrounding the PA itself. Thus, the data predistorter benefits from memory, and can be viewed as a nonlinear pre-equalizer for the discrete-time channel. Our focus in this work is to study data predistortion, and its performance for typical constellations, as a function of amplifier drive level and predistorter memory.

A. Applicability to NASA mission

The Tracking and Data Relay Satellite System (TDRSS), developed by NASA, relays mission and tracking data through nine operational satellites (with two currently under development) to/from orbiting user satellites including the Space Shuttle, the Hubble Telescope, and the International Space Station. Figure 1 shows the Tracking and Data Relay Satellite (TDRS) spacecrafts relaying user satellite data to a ground-station. Each relay satellite incorporates multiple inherently nonlinear traveling wave tube (TWT) power amplifiers that require severe input power backoff, which limits output power and amplifier efficiency. A high-order modulation scheme, such as 16-point quadrature amplitude modulation (16-QAM) recently proposed for TDRSS in [1], is especially vulnerable to the PA nonlinearities.

![Fig. 1. TDRSS system with TDRS spacecrafts relaying user satellite data to a ground-station [1]](image)

A robust digital predistortion technique along with a field-programmable gate array (FPGA) implementation can linearize the power amplifiers and significantly enhance TDRSS communication. Aside from application to TDRSS, other NASA communication systems such as the Deep Space Network, which connects with current science missions on Mars and those spacecraft passing other planets, would significantly benefit from using linearized power amplifiers.

B. System modeling

The simplest model for a nonlinear PA is a memoryless model, typified by models of Saleh [2] for example. It is now known, however, that memory effects need to be incorporated into PA models (and predistorters), at least for wideband operation, in order to achieve high performance. Memory can be incorporated in simple manner by Wiener models (linear filter followed by memoryless nonlinearity), by Hammerstein models (memoryless nonlinearity followed by linear filter), or both. More general Volterra models, as seen in equation (1), have been well-studied for this application, which relate the complex baseband output of the amplifier (in discrete-time) to the complex baseband input according to Schetzen [3].

\[
y_n = H[x_n] = \sum_i H_i[x_n] = \sum_{q_1=0}^{Q-1} a_1(q_1)x_{n-q_1} + \sum_{q_1=0}^{Q-1} \sum_{q_2=0}^{Q-1} \sum_{q_3=0}^{Q-1} a_3(q_1, q_2, q_3)x_{n-q_1}x_{n-q_2}x_{n-q_3}^* + \ldots
\]

(1)

The order (number of terms retained) and memory \( Q \) are key measures of model complexity. Some predistortion work has attempted to define a precise Volterra representation for an amplifier, then analytically solve for what is known as the \( p \)th-order Volterra inverse. A \( p \)th-order inverse for a nonlinear system \( H \) is another nonlinear system that, when cascaded with \( H \), leaves an end-to-end Volterra model whose kernels vanish through order \( p \), except for the identity kernel. Schetzen shows that the pre- and post-inverses of order \( p \) are identical. Such a predistortion technique, however, requires knowledge of the Volterra PA model, and moreover does not directly minimize end-to-end distortion.

The alternative is to formulate a data predistorter, either in the form of a lookup table or an algorithm (perhaps a memory polynomial [4]), and train it to have optimal characteristics, given some constraints. This is akin to training an adaptive receiver equalizer for ISI channels.

II. PREDISTORTER STRUCTURE AND TRAINING

The adopted structure for the data predistorter, shown in Figure 2, utilizes a lookup table (LUT) that is addressed by a span of \( 2L + 1 \) contiguous modulator symbols. The lookup table outputs a complex value \( v_n \), which can be viewed as the output of a nonlinear discrete-time filter, or equalizer, operating on a sequence of modulator symbols. Its function is to try to invert the nonlinear discrete-time channel with memory, operating
Harmon

The training method is a member of the so-called direct learning family of adaptive algorithms, wherein we sequentially update the table entries to minimize the error between the ideal constellation and the output of the eventual receiver. With reference to Figure 2, the target value for a given transmitted data symbol is denoted \( u_n \), whereas the received complex value corresponding to this transmitted data symbol is denoted \( z_n \), the result of the nonlinear system with memory. The ‘satellite channel’ is composed of an input demultiplexing filter \( h_u(t) \), nonlinear PA model \( g(\cdot) \), followed by an output multiplexing filter \( h_o(t) \) as seen in Figure 3. To maintain a desired input backoff (IBO) drive level, e.g. 3 dB, to the satellite channel during training, the output of the predistorter LUT is scaled by \( \alpha \). This scale factor is adjusted following each iteration based on the average power of the LUT and the desired drive level.

\[
\begin{align*}
  e_n &= u_n - \beta z_n .
\end{align*}
\]  

This error value will depend not only on the current symbol \( d_n \), but also on past and future data symbols due to the memory of the system. Such memory is manifest in the plot of receiver output when no predistortion is applied—a large cluster of points is seen corresponding to a given constellation point, due to this nonlinear ISI effect. (The cluster size is not due to additive noise, which is assumed to be negligible here.) We note that in each training batch, we desire several occurrences of a given data pattern, of span \( 2L+1 \) so that 1) additive noise can be averaged out, when present; and 2) the effect of system memory that exceeds our adopted memory can also be averaged out. This suggests a typical batch size \( N \) on the order of \( K \times M^{2L+1} \), where \( K \) is an integer on the range of say 1 to 10.

At each batch update, we adjust each LUT entry, according to the observed errors, in such a direction that should make the squared-error smaller for the next batch, conditioned on this specific data pattern. To do so, we first gather all measurements \( z_n \), that correspond to a certain data vector \( \vec{d} = (d_{n-L}, d_{n-L+1}, \ldots d_n, \ldots d_{n-(L+1)}) \). We find the centroid of this cluster of points, denoted \( \bar{z}_j \), scale by \( \beta \) as above, and form the error signal for the table entry \( j \) that corresponds to this data pattern. The updating is an LMS-style update, currently separately updating the magnitude and phase of the table entry at location \( j \) according to

\[
\begin{align*}
  A_j^{(i)} &= A_j^{(i-1)} + \mu_A |\bar{e}_j| , \quad (3) \\
  \theta_j^{(i)} &= \theta_j^{(i-1)} + \mu_\theta \angle \{\bar{e}_j\} , \quad (4)
\end{align*}
\]

where \( i \) is an iteration index. This training algorithm has previously been studied by the European Space Agency [5], although restricted to the case of memory span 3, apparently for complexity reasons. Presently, we use value 1 for both updating scale factors, choices that seem...
a good balance between convergence speed and table adjustment 'noise.'

It is beneficial to exploit constellation symmetry in training the table. All constellations studied here and that are typically utilized in satellite communication have four-quadrant symmetry. This means that the number of unique patterns that must be studied is $M^{2L+1}/4$. In training we group data clusters that correspond to a rotation of a first-quadrant constellation point for purposes of defining $\bar{e}_j$. This allows a smaller batch size (by 4) and thus faster convergence time. It also provides a better training quality when additive noise is present, since cases corresponding to a quadrant rotation should really all have the same table entry. When we actually apply the lookup table in transmission however, we have to 'un-rotate' from the first quadrant to the proper quadrant for symbol $u_n$. If we wish to implement this rotation on the fly, the LUT size can also be reduced by a factor of 4.

### III. Simulations and Results

In this section, we present simulation results of the techniques outlined. As previously mentioned, a Wiener-Hammerstein structure is adopted for the satellite channel. Input backoff (IBO) and output backoff (OBO) are defined relative to the amplifier saturation power levels. The PA model is based on the nonlinear AM-AM and AM-PM characteristics of the DVB-S2 TWT amplifier in [5]. Since the polynomial fit was completed for these characteristics in dB scale, a conversion to and back from dB is incorporated. The PA polynomial model is

$$|x(n)|_{dB} = 20 \log_{10} (|x(n)|)$$  \hspace{1cm} (5)

$$a(n) = \sum_{k=0}^{K} b_k |x(n)|_{dB}^k \quad \text{(AM-AM)}$$  \hspace{1cm} (6)

$$\theta(n) = \sum_{k=0}^{K} c_k |x(n)|_{dB}^k \quad \text{(AM-PM)}$$  \hspace{1cm} (7)

$$p(n) = 10^{a(n)/20} e^{j(\theta(n)+\angle x(n))}$$  \hspace{1cm} (8)

with complex input $x(n)$ and output $p(n)$, and uses the coefficients

$$b_0 = -1.150 \times 10^{-2} \quad c_3 = -1.285 \times 10^{-4}$$

$$b_1 = 5.454 \times 10^{-2} \quad c_4 = -4.295 \times 10^{-6}$$

$$b_2 = -3.974 \times 10^{-2} \quad c_5 = -4.118 \times 10^{-8}$$

$$b_3 = -5.128 \times 10^{-4}$$

This PA model alone, like those of [2], does not incorporate so-called memory effects, which are an increasingly recognized distortion effect associated with PAs, particularly for wideband signals. If the real PA had memory effects not modeled here, it is believed that the direct learning LUT approach is well-suited to precompensating for these effects as well, perhaps with minor losses in performance. The primary source of memory effects however in our model is transponder filtering at the PA input demultiplexer and output multiplexer. These filters have been chosen to be 4-pole Chebyshev characteristics with bandwidth-symbol duration product $B_s T = 1.02$.

Predistorter LUT batch updates continue until either of two stopping criteria are met. The first limits the number of training batches to 20 and the second relates to the constellation error. If the constellation error of the current batch is less than 1% better than the previous batch, then the LUT has essentially converged and additional iterations are not completed.

Figure 4 shows the set of points produced by the LUT as a result of training with 16-APSK modulation and memory span 3, where the full LUT address space is a modest 1 K. The amplifier drive level has input backoff 3 dB. The training batch size was 2 K symbols meaning the mean number of occurrences of each data pattern is 2 per batch. The figure shows the four clusters (out of 16) that are stored in the predistorter LUT, highlighted in black. The other three quadrants, a complex rotation of the first quadrant points, are plotted solely for reference but are not stored in the LUT. Figure 5 shows the ‘learning curve’ for the algorithm, which is a plot of the absolute error values versus time. There is a batch nature to this curve since expected updates are performed every $N$ samples. Clearly only a few batches are needed to gain convergence to an equilibrium condition. We note however that when memory span is increased, the convergence time is longer; both batch size and then number of batches need to increase to extract the benefits of more equalizer memory.

Next, Figure 6 presents the received constellation, following training, with 16-APSK and memory span 3, labeled 'precompensated output.' Again, the IBO is 3 dB. For comparison, we show the output data when no compensation is applied; the uncompensated points are clearly more scattered, due to both linear and nonlinear ISI, and exhibit gain compression of outer points as well as phase rotation effects due to AM-PM in the nonlinear amplifier. Also, Figure 7 shows the received constellation when a predistorter with memory span 5 is applied. A larger batch size of 256 K was used in training.
and the number of samples required for convergence is approximately 1000 times more than for the memory 3 predistorter due to the 256x increase in the LUT size.

As these figures show, predistortion is effective at removing these gross shifts of clusters, as well as tightening the receiver output clusters around a given constellation point. We measure the quality of the received constellation by the measured called \textit{error vector magnitude} (EVM).

Table I lists the EVM measure in decibels as a function of IBO, for cases of no compensation, a memoryless (span 1) predistorter, as well as performance with memory span 3 and 5. The larger memory span offers significant gain in EVM, or reduction in cluster scatter, due to its ability to better equalize both linear and nonlinear ISI, but primarily the latter. Again, Figure 7 presents the received constellation at IBO=3 dB, visually demonstrating the performance gain.

The discussion here has assumed noiseless operation.
Table I

Comparison of error vector magnitude reduction between no compensation and span 1 (memoryless), 3, and 5 memory predistorters for 16-APSK data. *This test reached the stopping criterion on constellation error whereas the others with span 5 reached the max number of allowed batches, causing this value to dip slightly.

<table>
<thead>
<tr>
<th>IBO (dB)</th>
<th>PA only</th>
<th>Predistorter with memory span 1</th>
<th>Predistorter with memory span 3</th>
<th>Predistorter with memory span 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>-15.6</td>
<td>-15.4</td>
<td>-18.9</td>
<td>-24.2</td>
</tr>
<tr>
<td>1.0</td>
<td>-15.6</td>
<td>-15.7</td>
<td>-19.6</td>
<td>-25.3</td>
</tr>
<tr>
<td>2.0</td>
<td>-15.8</td>
<td>-16.1</td>
<td>-20.6</td>
<td>-27.6</td>
</tr>
<tr>
<td>3.0</td>
<td>-16.0</td>
<td>-16.6</td>
<td>-21.4</td>
<td>-29.8</td>
</tr>
<tr>
<td>4.0</td>
<td>-16.3</td>
<td>-17.0</td>
<td>-22.0</td>
<td>-31.8</td>
</tr>
<tr>
<td>5.0</td>
<td>-16.7</td>
<td>-17.4</td>
<td>-22.6</td>
<td>-30.0*</td>
</tr>
</tbody>
</table>

Fig. 8. Comparison of total constellation degradation due to noise corresponding to $P_s = 10^{-1}$ and satellite channel distortion vs. IBO (dB) between no compensation and span 1 (memoryless), 3, and 5 memory predistorters

IV. Conclusions

Data predistortion, with memory span of 3 or 5 symbols, represents an effective means of compensating for satellite nonlinear distortion, and can improve link performance by 2-3 dB over no compensation. Training of the predistorter’s LUT has been described which employs loop-back reception to progressively improve the constellation quality.

Predistortion with memory span 3 can be easily implemented using LUT architectures, even for constellations such as 32-APSK, and has been developed in a Stratix II FPGA implementation. Implementation of memory 5 predistortion however, will likely require adoption of a functional form expressing the desired I/O behavior of the predistorter, e.g. memory polynomial or Volterra forms, with predistorter output computed in real-time in response to the data sequence. This remains an item for further study.

REFERENCES