JAVA FUNCTIONS FOR POWER AMPLIFIER LINEARIZATION TECHNIQUES

Robert Santucci, Andreas Spanias
Sensor and Signal Information Processing Center
School of Electrical, Computer, and Energy Engineering
Arizona State University

Abstract

This paper presents Java DSP functions and modules that we have developed to familiarize students with the mixed-signal application of digital signal processing (DSP) via interactive simulation software targeting the highlighting of fundamental design tradeoffs. The new functions implement several advanced methods and demonstrate tradeoffs that relate to the areas of digital communications, digital signal processing, and radio-frequency circuit design.

The Java functions developed have been tested and validated within our research group. Tutorials for each of the three modules have been developed and used with mixed upper-level undergraduate and entry-level graduate student classes. These functions have been assessed and they were shown to help students considerably in acquiring familiarity with basic concepts as well as various design tradeoffs using the presented training modules.

Introduction

The paper describes software that illustrates several aspects of amplifier linearization. In particular, it presents new modules, functions, tutorials, and online computer exercises that we have added to Java-DSP that familiarize senior-level undergraduate DSP students with the mixed-signal processing techniques used in wireless transmitters. J-DSP [1,2,3,4,5] allows students to make system-level design tradeoffs in interactive software simulations while masking the complexity of underlying circuits and algorithms. Smartphone and tablet versions of J-DSP are available, including the 2012 Premier Award winning iJDSP [6,7]. J-DSP can interface directly with DSP boards[8], or analyze specific systems such as MP3 [9].

The specific cases and modules developed deal with the inclusion of highly-efficient non-linear amplifiers into transmitters modulating both amplitude and phase. Such modulation is used in orthogonal frequency-division multiplexing (OFDM). The first module examines the mechanism by which the amplifier becomes non-linear. The two remaining modules address techniques and design parameters available for linearizing the amplifier. Altering these parameters teaches undergraduates system-level design, and shows them how design decisions made in DSP can affect circuit performance requirements.

We illustrate several design tradeoffs using simple block diagram simulations in J-DSP that expose undergraduate students to mixed-signal processing concepts. In examples, we show the interdependence between three separate senior-level courses: radio-frequency (RF) circuit design, DSP, and communications. The new functions cover various power amplifier design tradeoffs that span DSP, communications, and mixed-signal systems. After completing tutorials [10] covering basic concepts surrounding the new functions, students were given an assessment before and after completing the tutorials.

The primary benefit of performing these exercises using J-DSP over conventional tools, such as MATLAB coding, is that it allows emphasizing design tradeoffs and not implementation details, which would take students a considerable amount of time to debug and identify if failures are due to bugs in their implementation or system-level design choices.

Introduction to RF Amplifiers

In this section, we give an introduction to the utility and applications of RF power amplifiers (PAs), and then we continue with a description of impairments present in practical amplifiers and ways to correct them.

With the rapid deployment currently underway for smartphone technology, and new data intensive usage models, it has become critical for cellular phone carriers to get the highest possible data-rate through the relatively fixed available bandwidths [11]. This has forced cellular phone designers to move away from traditional constant amplitude modulation to modulation schemes that make more efficient use of available bandwidth by modulating both amplitude and phase to transmit more bits per symbol. As a result of modulating amplitude, current cellular phones must utilize transceivers with strict linearity requirements throughout the dynamic range of the modulated signal [12].

In addition to transmitting more bits per symbol, wireless providers want to transmit these symbols faster. In an uncontrolled wireless environment, the wireless signal can take multiple paths to the receiving antenna. The varying paths to the receiving antenna can be either direct line-of-sight or involve multiple reflections from buildings, mountains, or even walls within a building. These unique paths can each have different propagation delays, and this can cause problems when the symbol rate is very rapid. One way around this is to transmit many symbols in parallel very slowly. Thus, if the duration of each symbol is much longer than the difference in time between the longest and shortest path, the path delay effect is largely mitigated.

The natural next question is how to transmit multiple symbols in parallel wirelessly without using many independent transceivers. This can be accomplished by constructing the signal to be transmitted in a specific new manner. The traditional approach has been to transmit symbols as a series in time that moves as fast as the available bandwidth and spectrum allow. In the newer approach the symbols are not allocated as a time-series, but rather to different frequencies in the discrete-time frequency-domain. Then, the time-domain signal to be transmited is generated by taking the Inverse Fast Fourier Transform (IFFT) of the discrete frequency spectrum. Because the different frequency tones are orthogonal, this technique is known as Orthogonal Frequency Domain Multiplexing (OFDM)[13,14]. OFDM allows very tight allocation of the transmitted frequency spectrum to get maximum bit-rate with the smallest spillover into adjacent frequencies and can avoid multipath effects, but it does have some downsides. One of these downsides is that after the IFFT is taken the phase of each independent modulated sinusoid combines in a nearly random manner, which causes the time-domain signal to have a wide amplitude variation, or dynamic range [15].

In a cellular transmitter, the power amplifier is often the largest consumer of power as it drives the high power signal out of the antenna. PA power efficiency has a significant effect on handset battery life. The transmitted signal can be made very linear by keeping the amplifier biased in such a manner that the radio frequency (RF) signal never takes the amplifier into a compression region. However, designs where the bias point remains stationary over a wide-range of output signal levels require a large amount of bias power. This bias power is not transmitted out of the antenna, but rather is dissipated as heat.

By increasing the amplifier input amplitude until a portion of each RF cycle is clipped, the percent of power expended for biasing can be made lower and thus the overall power efficiency of the amplifier can be increased. As a downside, this clipping reduces the fundamental gain and introduces harmonics that must be filtered away. The level at which compression occurs within a given amplifier is most commonly a fixed amplitude. As a result of the fixed compression level, the fraction of an RF cycle spent in compression and consequently fundamental gain becomes a function of the amplitude of the signal being amplified [12], as shown in Figure 1.

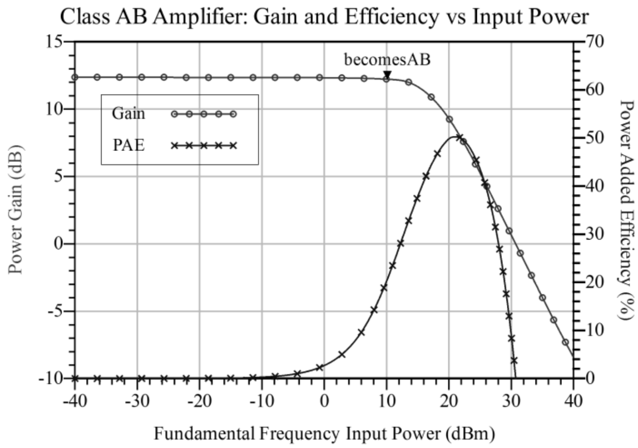

Figure 1: Class AB Amplifier. Efficiency peaks when gain compression is present, but the amplifier is not yet saturated.

The first J-DSP module demonstrates the non-linear gain and harmonic energy produced by various degrees of clipping that result from operating an amplifier in its energy-efficient class AB region. Students are able to alter both the incoming signal amplitude and the clipping level, and monitor changes in gain and harmonic content. The key point students learn from the first module is how clipping of a periodic signal changes the gain of the fundamental tone and creates new harmonic tones whose magnitude increase as more clipping is performed.

In early wireless digital communications systems, the amplitude-dependent gain changes were avoided by modulating data only via phase changes. This was done in systems such as in Global System for Mobile Communications (GSM) phones, which largely avoid distortion in their constellation due to amplifier non-linearity. Transmitting amplitude modulated data without distortion requires the transmitter to have a constant gain over some window of transmitted amplitudes. An example of a transmitted signal constellation with amplitude distortion is shown in Figure 2.

For the best possible overall performance it would be beneficial to develop a system in which the amplifier could be operated in its highly-efficient compressed region where gain is nonlinear while having a system to mitigate the nonlinear effects. Digital predistortion is one method of altering the transmitted signal in such a way that when the altered signal is passed through the non-linear distortion of the amplifier, the desired signal is produced. This is done by inserting a new gain-expansive block in advance of the gain-compressive amplifier, as shown in Figure 3.

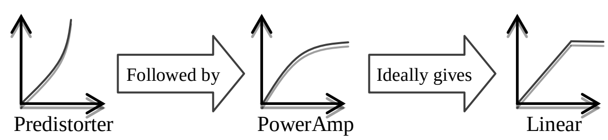

Figure 3: Predistortion System Gains.

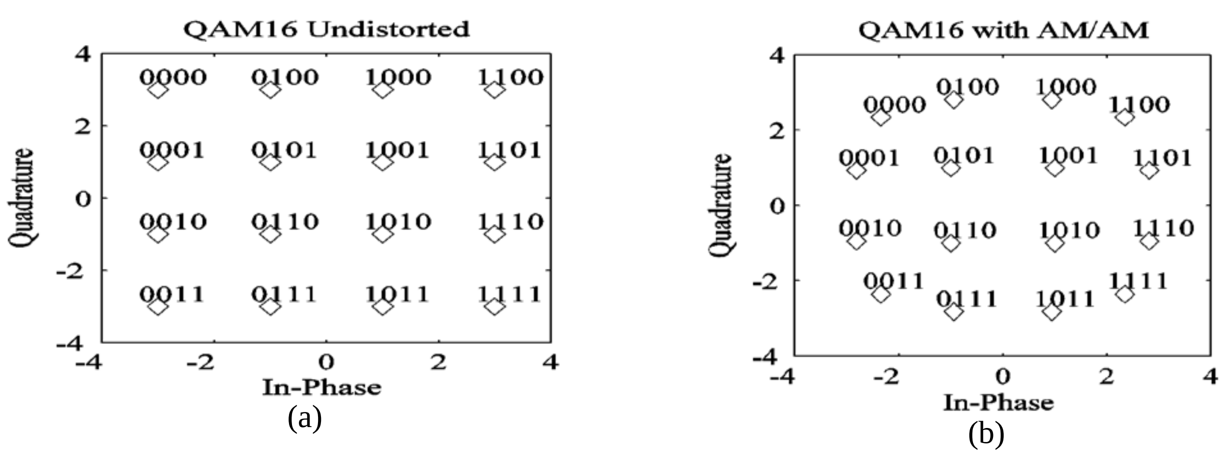

Figure 2: QAM16 (a) with an ideal amplifier and (b) if amplifier gain is compressed.

Because amplifier gain and output capability changes continually with operating voltage, temperature, and aging, a co-located receiver periodically measures how the signal is being distorted [16]. A typical predistortion system topology is shown in Figure 4.

In J-DSP, we have written a new module for doing linearization using a technique based upon selecting a correction factor from gain-based look-up-tables (LUT). There is also a module for doing linearization based on training a neural network to invert non-linear amplifier behavior. For both linearization modules, the frequency spectrum of amplifier output both with and without predistortion is shown. This allows students to see spectral leakage resulting from the non-linear amplifier gain. Two metrics used to quantify amplifier non-linearity are shown within the modules. Those metrics are adjacent channel power ratio (ACPR) and error vector magnitude (EVM). Additionally, plots of raw amplifier, predistorter, and net system gains as a function of input magnitude are shown.

For the gain-based look-up-table predistorter, plots are provided to show the correction factors associated with each LUT bin and the frequency with which each bin is utilized. This allows students to experiment with different numbers of bins in addition to different learning factors. For the neural network predistorter, selections exist for the students to experiment with network size and layer count.

Non-linear Gain and Clipping

Class AB amplifiers are commonly used in transmitters to achieve higher efficiency. In a class AB amplifier, the conduction angle is between 180° and 360°, meaning the amplifier is at maximum output voltage between 0 and 50% of the time. This clipping usually occurs at a fixed threshold. For small amplitudes, the sinusoid may not reach this clipping threshold and have a conduction angle of 360°, implying more linear class A operation. As a result of the fixed clipping threshold, the conduction angle is a function of the input signal magnitude. This dependency is demonstrated in Figure 5. A decrease in conduction angle causes two effects to occur: a decrease of fundamental-frequency gain and an increase of harmonic energy.

A power series is frequently fit using least-squares techniques to model the non-linear transfer function through the amplifier. This fitting yields amplifier gain G as a power series with weighting parameters a_k, where K is the maximum order of nonlinearity[17].

$$G(v(n)) = \sum_{k=1}^{K} a_k |v(n)|^{k-1} \quad (1)$$

When a multi-tone signal is fed into the amplifier, mixing products are produced as the harmonics are multiplied. Two sets of these mixing products produce problems. First, mixing products that produce tones around the signal frequency are problematic. These mixing

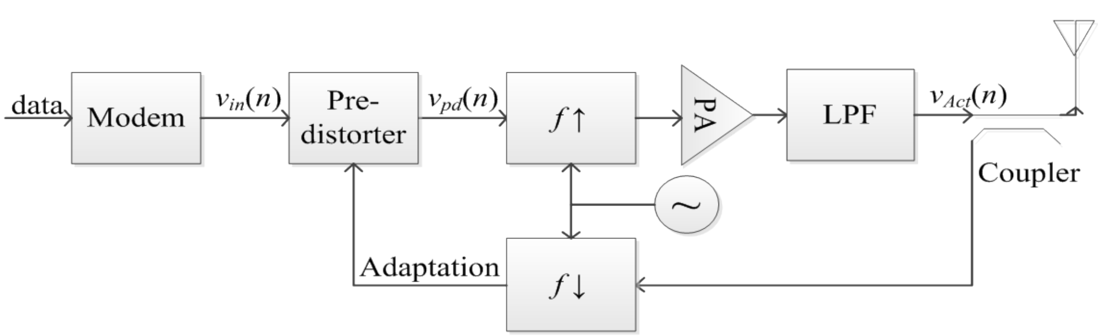

Figure 4: Typical predistortion system, from [9].

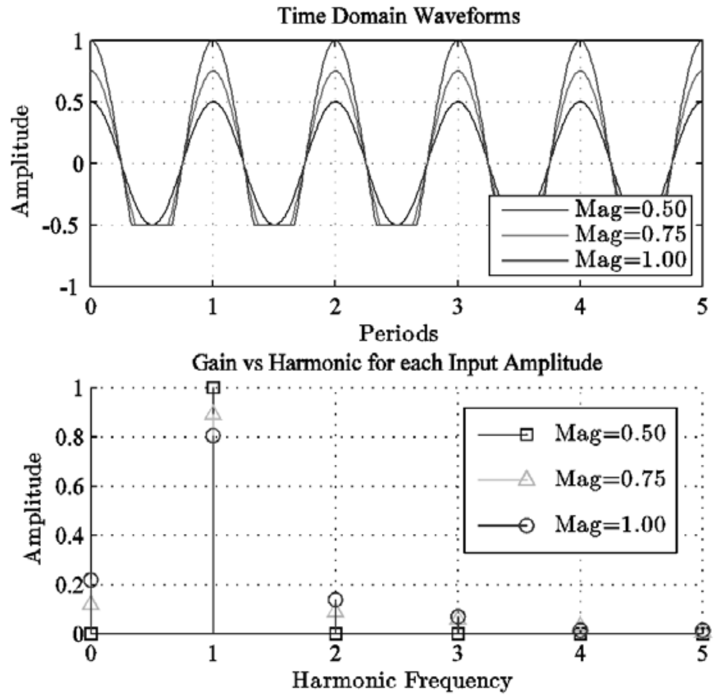

Figure 5: Variable Gain and Harmonics, modified from [18].

products result near the signal frequencies and are thus difficult to filter out and result in spectral spreading. Secondly, mixing products are produced near DC. These are demonstrated with predistortion as a spectral broadening.

This class AB clipping mechanism, the generation of harmonics, and the variation of the fundamental gain can be demonstrated in J-DSP utilizing the following test bench. In the example exercise, the input frequency is considered to be at the RF data rate. It can be seen that reducing the amount of power into the power amplifier (PA), also known as backing off the power, reduces the level of the harmonic content. The fundamental gain is computed as a function of the input amplitude in the tutorial, as shown in Figure 6.

Gain-Based Look-up-Table Predistortion

The gain-based look-up-table exploits the fact that the amplifier distortion occurs primarily as a slowly varying function of the input power of the signal $v_{pd}(n)$ fed into the amplifier [19]. If the amplifier is monotonic, then increasing the input power will also increase the output power of the amplifier and similarly for decreases in

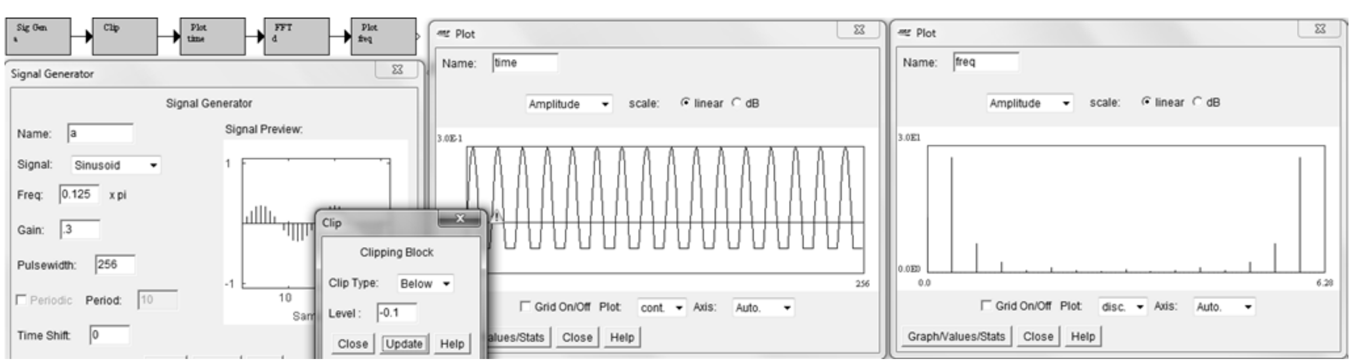

Figure 6: Simple Schematic for Demonstrating the Class AB Clipping Concept in J-DSP.

power. If the gain of the amplifier at any given input power is not as desired, then the input power can be artificially increased or decreased to hit the desired gain. This artificial decrease or increase to achieve the desired gain is done by the predistorter. In the gain-based look-up-table predistorter, input magnitudes are split into regions based on contiguous input powers [19]. The regions are henceforth referred to as bins, and for each bin a correction factor is developed. The look-up-table portion of the name refers to the selection of gain-correction factor, b, by selecting a bin based on the input signal power. The correction factor is learned using the LMS algorithm [20,21]. A DSP flowchart showing implementation is shown in Figure 7. This learning process is an example of an adaptive algorithm to which students are exposed.

The error $e(n)$ between the actual transmitter output $v_{act}(n)$ measured by the receiver at time step n and the desired transmitter output $v_{des}(n)$ for a given modem value $v_{in}(n)$ is used to adapt the gain correction factor b for the selected bin by the formula below, where μ controls the rate of adaptation:

$$b_{selected}(n+1) = b_{selected}(n) + 2\mu e(n) v_{in}^{*}(n). \qquad (2)$$

In a practical system, there is the potential lag of several symbol times while measuring the actual output. In that case, care must be taken to properly synchronize the desired signal gain path with the actual output path by the insertion of delay. Accordingly, this delays the update of $b_{selected}$ by additional cycles. Care must be taken to ensure the updates are done properly and to avoid overshooting. This can be done either by decreasing μ and applying incremental updates, or by blocking further updates while a calculation is in progress.

When designing the look-up-table, one of the tradeoffs in the design that must be made is the number of bins for the LUT. Choosing a smaller number of bins allows decreasing the memory that must be allocated to the LUT and also decreases the number of weights that must be trained via adaptation. This enables faster adaptation if the amplifier is being operated in a highly dynamic environment, perhaps where an object interferes with the antenna’s impedance seen by the power amplifier. Creation of more LUT bins increases memory requirements and the number of samples required for training, but allows for correcting a smaller and more consistent region of the amplifier’s gain versus the input power curve.

Non-linearity in the power amplifier manifests itself as a gain that changes as a function of the magnitude being transmitted at a given time instant. In the frequency domain, this can be seen as a convolution of the original data signal with a weak signal with bandwidth similar to the original signal. The result of this is a spectral broadening, or leakage, of the transmitted signal into adjacent bands. This leakage can entirely block receivers on adjacent bands. Given a receiver intending to receive a signal from distant transmitter A while a nearby transmitter B transmits in the adjacent frequency channel,

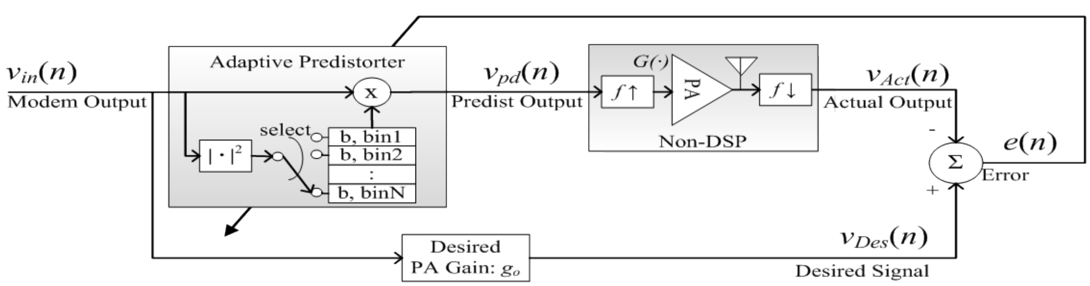

Figure 7: Gain-Based Look-up-Table Predistortion, modified from [19].

it is possible for spectral leakage from the nearby transmitter B to mask the desired signal from distant transmitter A. Smart allocation of frequencies within a cellular system can alleviate this problem to some extent, but it can still influence channel spacing, or the frequency gap between adjacent channels. To use frequencies as efficiently as possible it is desirable to keep this channel spacing as small as possible. The ratio of signal power to the power leaked into the adjacent channel is known as the adjacent channel power ratio. In equation form, where V_{act} is the discrete time frequency spectrum of the transmitted signal, ACPR is:

$$ACPR_{dB} = 10\log_{10}\left(\frac{\sum_{k=N+1}^{2N}|V_{act}(k)|^2}{\sum_{k=1}^{N}|V_{act}(k)|^2}\right). \qquad (3)$$

The distortion introduced by the changing amplifier gain also causes errors within the signal bandwidth. In an OFDM scheme, a certain number of OFDM subcarriers are usually reserved for pilot tones. These pilot tones make identifying any phase shift or gain change between signal generation and measurement possible [22]. In a simple scheme, these tones are equally spaced. The ratio between the original pilot tone on subcarrier k and the complex value received on that subcarrier is calculated and found as $H(k)$. For all of the data carrying subcarriers between pilot tones, the scaling $H(k)$ can be found via interpolation of the two nearest pilot tones. After applying the scaling factor $H(k)$ to all subcarriers, a hard decision is made selecting the constellation point nearest the measured complex value on the subcarrier. Error Vector Magnitude (EVM) is a metric quantifying the distance between the measured complex value and the ideal constellation point placement. Where $V_{hard}(k)$ is the hard decision constellation point for subcarrier k, and $V_{act}(k)$ is the value at the point of measurement, and $H(k)$ is the scaling, or channel estimate, for subcarrier k, EVM is defined according to (4).

$$EVM_{dB} = 10\log_{10}\left(\frac{\sum_{k=1}^{N}\left|\frac{V_{act}(k)}{H(k)} - V_{hard}(k)\right|^2}{\sum_{k=1}^{N}|V_{hard}(k)|^2}\right). \qquad (4)$$

For an ideal signal, no power would leak into adjacent bands and the constellation points would be exactly where OFDM generated them. Consequently, an ideal signal would have EVM and ACPR of both -∞ dB. Requirements for average EVM and ACPR depend on the communications standard used. By improving the EVM and ACPR performance, predistortion allows either increasing the transmited power available from an amplifier or substitution of an amplifier with a cheaper, lower power device. Both ACPR and EVM are calculated within the J-DSP modules to provide a metric quantifying performance of the power amplifier with and without predistortion, as shown in Figure 8.

Neural Network based Predistortion

An alternate approach to doing the predistortion proposed in [23] utilizes neural networks. The goal of this approach is to train the neural network to the inverse function of the power amplifier, excepting the desired gain, and place that network in the system prior to the amplifier. Neural network predistortion starts to outperform gain-based LUT predistortion as the distortion becomes more complex. An example of when distortions get more complex is for wideband signals when the amplifier's bias network is not behaving ideally. In these cases, the state of the bias and thus the amplifier's instantaneous gain are dependent on the current and recent previous symbols transmitted [16]. While the gain-based LUT has been extended in [24,25] by making the assumption that the non-linear portion of the distortion is generated based on the power of the current symbol and that the memory effects due to previous symbol powers are linear, this assumption is a simplification of the circuit behavior. This assumption is not needed for neural networks. To allow easier presentation of tradeoffs in our tutorial, the neural network is being applied to a simple memoryless power amplifier.

The neural network used in digital predistortion consists of a set of neurons arranged in layers. The first layer consists of the

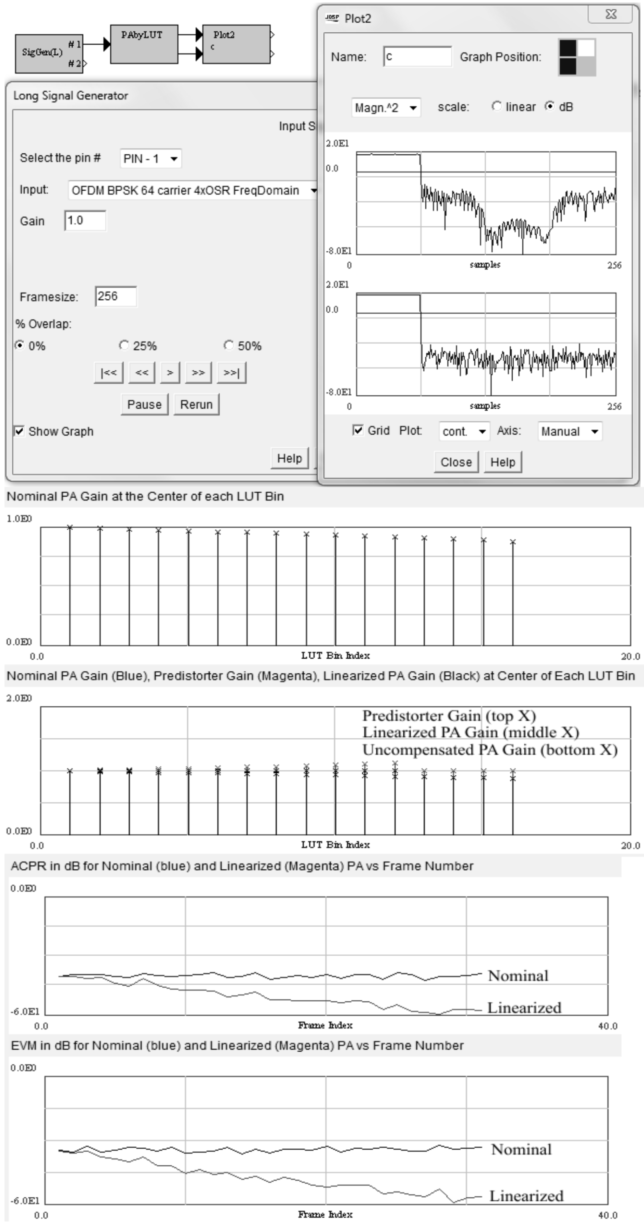

Figure 8: Gain-based LUT Predistortion Simulation.

inputs from the baseband modem. The final layer consists of one node with output amplitude in the case of the memoryless amplifier. Inbetween these two layers are additional hidden layers, whose inputs and outputs are not visible from outside of the network. The hidden layer outputs are a mathematical construct and have no correlation to the underlying device physics. The neurons in each hidden layer take a weighted sum of each of the previous layer's outputs and a bias. This weighted sum is then passed through the hyperbolic tangent function to produce the neuron output, which is a function with continuous derivatives and gives the network the ability to model smooth nonlinearities. The output layer takes as input a weighted sum of the final hidden layer's neurons and a bias, and then passes them through a linear output function which allows scaling of the values to the appropriate range. To make the network work appropriately, the weights applied to each interconnect, shown dashed, between neurons on Figure 9 must be trained to an appropriate value.

The selection of network topology and computation of proper weights is what makes applying neural networks difficult. The objective is to identify the inverse function of the power amplifier gain, except for the desired gain, and place it prior to the amplifier so that when the amplification is performed only the desired gain remains. This is implemented by

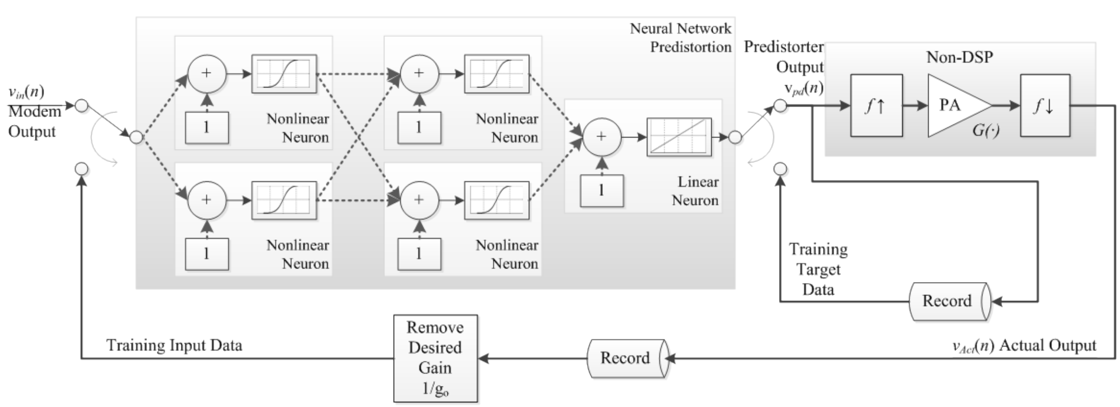

Figure 9: Neural Network Based Predistorter Using Two Hidden Layers, modified from [10].

recording the input and output from the RF section of the power amplifier. The weights are initially set to random values normally distributed about zero. After each frame, the weights are updated using back-propagation, a process that works backwards through the network attributing each weight to an effect on the output error. To do this, a training set of data representative of the desired network behavior must be attained. To model the inverse function of the amplifier, the training inputs are the power amplifier output values divided by the desired gain. The training target data is then the input to the power amplifier. Due to the non-linear characteristics of the data, the Levenberg-Marquardt algorithm [26] has been found to be the most effective method for weight determination. To avoid passing actual data through the untrained network, for the first packet, the predistorter is bypassed and raw modem output is sent directly to the power amplifier. The performance of the network could be further optimized if regularization, or scaling of the inputs to consume the whole range of the hyperbolic tangent, was performed.

One of the issues in selection of network topology is to make sure the network is being trained to the behavior of the circuit, rather than fitting the network to every point in the training data set, including noise. This over-fitting can usually be seen as an approximation function with too much oscillating curvature, which should not exist for any expected power amplifier. Various strategies exist for fighting this problem including network pruning and cross-validation [27]. In network pruning, the network attempts to remove as many neurons as possible while maintaining a good fit against the training data. In training and validation, a portion of the training data is used for calculating network weights. The error in estimation of the validation data is then computed to ensure that it is becoming lower as the network is trained.

For the internal versions of J-DSP tested within our lab, the Encog Java neural network framework [28] is used to implement the network computations. To keep simulations quick, only 125 training iterations, or epochs, are made through the network per pass. Better results could be attained with more epochs. An example of a neural network predistortion simulation in J-DSP is shown in Figure 10.

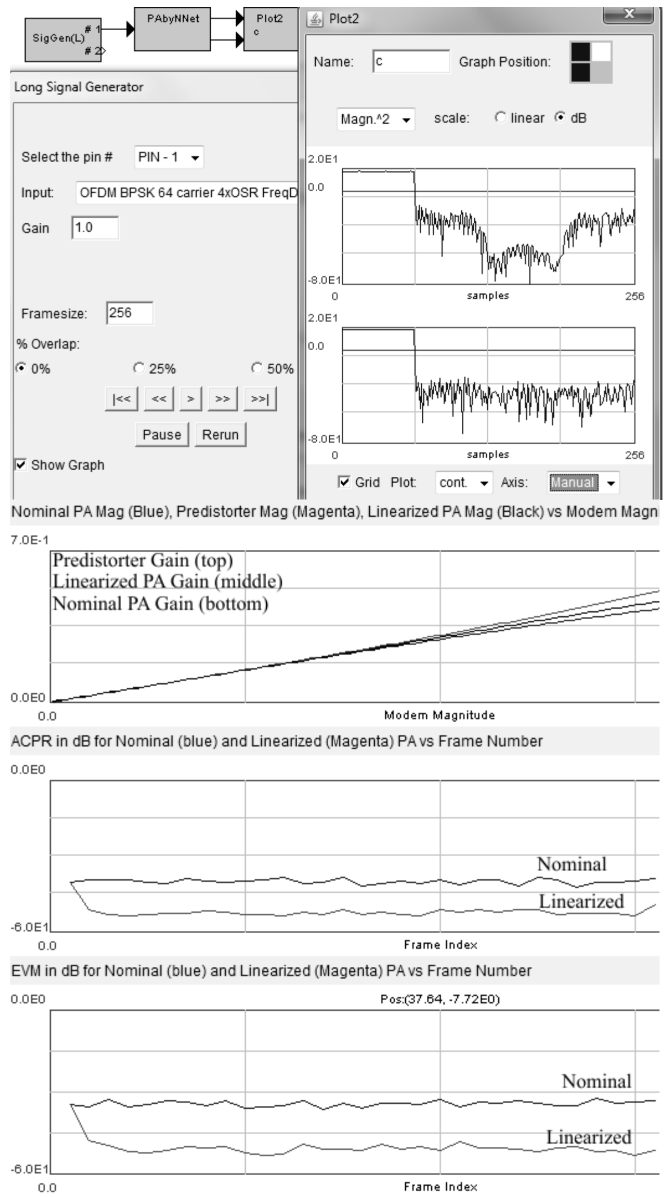

Figure 10: Neural Network Predistortion Sim.

Labs, Assessments, and Overview

An early version of the lab exercises used for the J-DSP power amplifier linearization modules was presented in [10], including a lengthy set of exercises. For this paper, abridged versions of our tutorials were used allowing students to complete the J-DSP exercises in a single class session. The abridged exercises and questionnaires used are available for download [29]. The classes were mixed upper-division undergraduate and entry-level graduate DSP and communications courses. At the start of the class, prior to doing the tutorials, students completed a pre-quiz of basic questions on amplifier clipping and linearization. The tutorials walked students through the new J-DSP functions and asked them to complete some basic exercises to demonstrate design tradeoffs. The same quiz was reissued after the exercises. The difference in the scores between the pre- and post-tutorial quizzes were used to demonstrate improvement in the students' understanding of the presented material. The results of the quizzes are shown in Table 1.

Table 1: Student Quiz Results.

Question Topic	% Right Pre-Quiz	% Right Post-Quiz	% Improved
where amplifier efficiency is highest	43	70	27
where amplifier linearity is highest	61	86	25
how amplifier gain changes with increasing signal amplitude	54	70	16
what type of filter a real amplifier can be viewed as	7	53	46
what happens to frequency spectrum of signal through non-linear amplifier	57	72	15
spectral leakage causes and effects	86	84	-2
what type of filter a predistorter can be viewed as	21	72	51
what the combined gain shape of a combined linearizer and predistorter should be	68	67	0

Significant improvements in student results were observed in the quiz completed after completing the tutorials relative to those completed prior to the exercises. For the six questions with the lowest pre-test performance, the median number of students answering correctly increased by about half. For the two questions answered correctly most often pre-test, student performance did not improve significantly. This probably resulted from the pre-test questions being too easy to identify from the multiple choices made available, especially in the case of the spectral leakage cause and effect question in which 86% of the students answered correctly despite not having signficant background on power amplifier gain compression.

Students were asked to complete a self-assessment survey about what they felt they had learned and what could be improved in the tutorials and J-DSP. Results of the self-assessments are shown in Table 2.

Table 2: Student Learning Self Assesment.

Question asked of students as a results of doing exercises:	← Strongly Agree … ..Strongly Disagree→ (1)	(2)	(3)	(4)	(5)	Avg.
Learned difference between ideal and real amplifier	11	25	7	0	0	1.91
Understand clipping, conduction angle, and gain	37	5	0	0	0	1.13
Understand need for PA linearization	15	24	4	0	0	1.73
Were provided a comprehensive summary of design tradeoffs between linearity and efficiency	14	20	9	0	0	1.88
Knew the tradeoff between clipping and production of harmonics	14	22	6	1	0	1.86
Saw sufficient clarity in the graphical outputs generated for all exercises	15	20	7	0	0	1.81
Learned the importance of choosing optimum power back off in OFDM	11	24	7	0	0	1.90

In general, students found the exercises easy to understand. A couple of students commented that they understood the tradeoff between linearity and performance, but not enough regarding efficiency and power-backoff due to the lack of quantization of efficiency. Future versions of this tool could be improved by having some power efficiency tracking built into the amplifier modeling. The shortened version of tutorial also reduced the amount of experiments sweeping power back-off, thus explaining the lesser reported understanding for that metric. These exercises should probably be extended through two class periods for more exhaustive coverage of linearization.

Conclusions

This paper presented a summary of J-DSP simulation modules used as a tutorial to teach amplifier linearization and predistortion techniques to senior undergraduate and graduate students. Tutorials based on the J-DSP modules may be used by instructors in a classroom setting to demonstrate the tradeoffs between the signal processing, RF circuit design, and communications requirements.

The tutorials were demonstrated in a classroom environment with multiple groups of senior-level undergraduate and entry-level graduate students. The tutorials guided students through example amplifier linearization scenarios demonstrating linearizer configurations that yield both good and bad performance. Students showed an improvement in understanding after completing the tutorials based on 22% improvement on average on questions answered correctly.

Acknowledgements

This work has been funded in part by NSF CCLI (TUES) phase 3 award number 0817596.

Bibliography

1. Arizona State University. (2013, April) Java-DSP. [Online]. http://jdsp.engineering.asu.edu/

2. A. Spanias and V. Atti, "Interactive online undegraduate laboratories using J-DSP," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 735-749, November 2005.

3. A. Spanias et al., "On-line signal processing using J-DSP," *IEEE Signal Processing Letters*, vol. 11, no. 10, pp. 821-825, October 2004.

4. H. Kwon, V. Berisha, V. Atti, and A. Spanias, "Experiments with Sensor Motes and Java-DSP," *IEEE Trans. on Edu.*, vol. 52, no. 2, pp. 257-262, May 2009.

5. V. Atti, "Development of MATLAB and JAVA software models for Algebraic CELP algorithms," Arizona State University, Tempe, AZ, Master's Thesis 2002.

6. J. Liu et al., "Interactive DSP laboratories on mobile phones and tablets," in *Proc. of Int'l Conf. on Acoustics, Speech, and Sig. Proc.*, Kyoto, 2012, pp. 2761-2764.

7. S. Ranganath et al., "WIP: Performing signal analysis laboraties using Android devices," in *Proc. of IEEE Frontiers in Education Conf.*, Seattle, 2012, pp. 1-2.

8. A. Spanias et al., "Interfacing Java-DSP with a TI DSK for use in a Signal Processing Class," *ASEE Computers in Education Journal*, vol. XVII, no. 3, pp. 27-35, July-Sept 2007.

9. C.W. Huang, J. Thiagaragan, A. Spanias, and C. Pattichis, "A Java-DSP interface for analysis of the MP3 algorithm," in *DSP and IEEE Sig. Proc. Edu. Workshop*, Sedona, 2011, pp. 168-173.

10. R. Santucci and A. Spanias, "Use of Java-DSP to Demonstrate Power Amplifier Linearization Techniques," in *American Soc. for Eng. Edu.*, Vancouver, 2011.

11. John Proakis, *Digital Communications*, 4th ed. New York, USA: McGraw Hill, 2001.

12. S. Cripps, *RF Power Amplifiers for Wireless Communications*. Norwood, MA: Artech House, 1999.

13. Andrea Goldsmith, *Wireless Communications*. New York, NY: Cambridge University Press, 2005.

14. Mahesh Banavar, A. Narasimhamurthy, and Tependelenlioglu C., *OFDM Systems for*

Wireless Communications, 1st ed., A. Spanias, Ed. USA: Morgan & Claypool, 2010.

15. Lawrence Larson and et al, "Digital Control of RF Power Amplifiers for Next-Generation Wireless Communi-cations," in *Proc. of ESSIRC*, Grenoble, France, 2005, pp. 39-44.

16. J.H.K. Vuolevi, T. Rahkonen, and J.P.A. Manninen, "Measurement Techniques for characterizing memory effects," *IEEE Trans. on Microwave Theory & Techniques*, vol. 49, no. 8, PP. 11383-1389, Aug 2001.

17. J Cavers, "The Effect of Data Modulation Format on Intermodulation Power in Non-Linear Amplifiers," in *IEEE Vehicular Tech. Conf.*, 1994, pp. 489-493.

18. R. Santucci, A. Spanias, T. Gupta, and M. Shah, "Advanced functions of Java-DSP for use in electrical and computer engineering courses," in *American Society for Engineering Education*, Louisville., 2010.

19. J Cavers, "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Trans on Vehicular Tech*, vol. 39, no. 4, pp. 374-385, November 1990.

20. B. Widrow, J. McCool, and M. Ball, "The Complex LMS Algorithm," *Proc. of the IEEE*, vol. 63, no. 4, pp. 719-720, April 1975.

21. A. Spanias, *Digital Signal Processing: An Interactive Approach*, 1st ed. Tempe, Arizona: LL Press, 2007.

22. S Coleri, M. Ergen, A Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. in Broadcasting*, vol. 48, no. 3, pp. 223-229, September 2002.

23. S Boumaiza and F. Mkadem, "Wideband RF power amplifier predistortion using real-valued time-delay neural networks," in *Proc. of the European Microwave Conf*, Rome, Italy, 2009, pp. 1449-1452.

24. P. Jardin and G. Baudoin, "Filter Lookup Table Method for Power Amplifier Linearization," *IEEE Trans. on Vehicular Tech.*, vol. 56, no. 3, pp. 1076-1087, 2007.

25. R. Santucci and A. Spanias, "A block adaptive predistortion algorithm for transceivers with long transmit-receive latency," in *2010 4th Intl Symp on Comm., Control and Sig. Proc.*, Limassol, 2011.

26. M.T. Hagan and M.B Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989-993, 1994.

27. Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

28. Heaton Reserach. (2012, November) Encog Machine Learning Framework. [Online]. http://www.heatonresearch. com/encog

29. R. Santucci. (2013, June) JDSP RFPA web. [Online]. http://jdsp.engineering .asu.edu/ RFPAweb/

Biographical Information

Robert Santucci is a Ph.D. candidate at Arizona State University in the School of Electrical, Computer, and Energy Engineering. His research interests include amplifier linearization and distributed estimation. He has worked in wireless transceivers and PMIC design. He has obtained an M.S. degree in Electrical Engineering from Arizona State University, and Bachelor of Science degrees in Electrical Engineering and in Computer Science from Embry-Riddle Aeronautical University.

Andreas Spanias is a Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU). He is also the founder and director of the Sensor, Signal, and Information Processing (SenSIP) industry consortium. His research interests are in the areas of adaptive signal processing, speech processing, and audio sensing. He and his student team have developed Java-DSP. He is a fellow of IEEE and a member of ASEE.