COMPANDING DIGITAL SIGNAL PROCESSORS

Ari Klein and Yannis Tsividis

Department of Electrical Engineering, Columbia University, New York, NY, USA

ABSTRACT

We discuss a technique whereby internal signals in a DSP can be controlled externally without causing output disturbances. This is used to extend the technique of companding (compressing and expanding), widely used in transmission and sound recording, to digital signal processors. It is shown that this technique allows all signals involved, including those at the input of an analog-to-digital converter and those internal to a fixed-point DSP, to be close to full scale, thus spanning most of the available bits and making possible a large signal-to-noise-plus-distortion ratio over a large input range.

1. INTRODUCTION

Companding (compressing/expanding) is widely used in transmission and sound recording [1, 2] to compress the dynamic range of input signals, so that the latter remain well above the noise in the channel or the storage medium even at low input levels; at the output, the dynamic range is restored (expanded). Since the transmission or storage medium does not (ideally) modify the signal, the output envelope (and the actual output) can generally be recovered from the compressed signal, as a one-to-one relation exists between the envelope of the latter and the original input envelope. One may consider applying this technique to systems of the type shown in Fig. 1(a), composed of analog-to-digital converters (ADCs), digital signal processors (DSPs), and digital-to-analog converters (DACs); the input to the ADC is, for simplicity, assumed to be an already sampled analog value. In applications where large amounts of computation are needed, such as multimedia, it is desirable to implement the ADC and fixed-point arithmetic with as few bits as possible; companding should help combat the effect of the resulting significant quantization error. However, a straightforward application of companding to such systems runs into problems. If a compressor is used in front of the ADC, envelope information cannot be independently recovered at the output, as the signal envelope is in general modified by the DSP. It may appear at first sight that this problem can be solved by dividing the input signal, $u(n)$, by its envelope, $e_u(n)$, to produce a signal $\hat{u}(n)$ with compressed envelope, and transmitting $e_u(n)$ directly to the output in order to restore the signal envelope, as shown in Fig. 1(b); this, however, assumes that the output envelope in the original system of Fig. 1(a) is the same as the input envelope, which in general is not the case. For example, if the DSP were simply a k-delay block, $y(n)$ should equal $u(n-k)$. However, the compressed input $\hat{u}(n) = \frac{u(n)}{e_u(n)}$, passing through the k-delay block, gives $\hat{y}(n) = \frac{u(n-k)}{e_u(n-k)}$ at the output of the DSP, which, when multiplied by the input envelope, gives $\hat{y}(n) = [\frac{e_u(n)}{e_u(n-k)}]u(n-k)$ as the final output, which is not equal to the desired $u(n-k)$. DSPs with more complicated dynamics will have more complicated distortion.

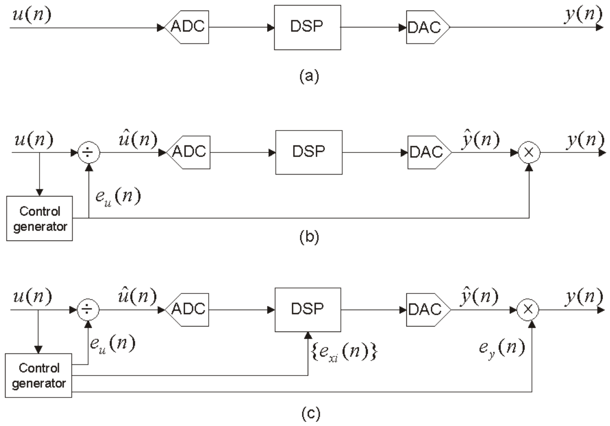

Fig. 1. (a) A prototype ADC-DSP-DAC system. (b) An attempt to introduce companding to the system in (a). (c) Properly companding system.

Thus, if the input-output behavior of the modified system is to remain identical to that of the original system, changes must be made to the state variables inside the processor, as is the case with analog systems [3, 4]. This is indicated in Fig. 1(c).

In special cases where the DSP dynamics are extremely fast relative to variations in $e_u(n)$, the distortion introduced by the simple input-output companding shown in Fig. 1(b) will be relatively low. For example, in the case of the k-delay block, if $e_u(n)$ is approximately equal to $e_u(n-k)$ for all n, then $y(n)$ is approximately equal to $u(n-k)$, as desired. In general, however, the distortion introduced by applying the technique of Fig. 1(b) is intolerable, in which case companding of the form shown in Fig. 1(c) becomes necessary. In this paper, we discuss our initial attempt to accomplish such proper companding in DSPs, and present simulation results that confirm the ideas presented.

2. EXTERNALLY LTI DSPS

By performing a linear time-varying transformation on the internal states of a general linear, time-invariant (LTI) system, one can produce a system in which internal state variable control may be achieved without disturbing the final output [4]. Here, we will extend this technique by also allowing independent control of the input and output, and we will apply it to DSPs. Consider a LTI discrete-time m^{th} order system as shown in Fig. 2(a), with single input $u(n)$, state vector $x(n) = (x_i(n))$, and single output $y(n)$. The quantity $q(n)$ represents the quantization error of the ADC, and will be assumed to be zero for now. The state equations of this system are of

This work was supported in part by National Science Foundation Grant CCR-02-09109.

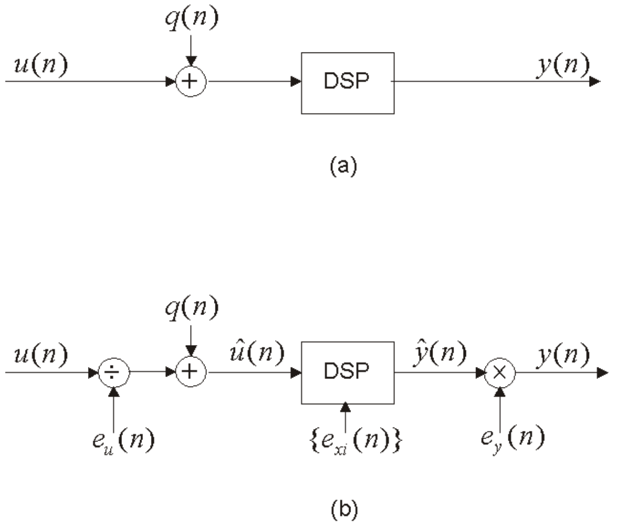

Fig. 2. (a) A discrete-time system. (b) The system in (a) with controls for adjusting its internal signals.

the form:

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + du(n) \end{aligned} \tag{1}$$

where $A = (a_{ij})$ is a $m \times m$ matrix, $B = (b_i)$ is a $m \times 1$ column vector, $C = (c_j)$ is a $1 \times m$ row vector, and d is a scalar. This system will be referred to as the "prototype system", and corresponds to the system in Fig. 1(a), assuming that the ADC and DAC are of infinite resolution.

We want to create the system of Fig. 2(b), which, if fed by the same input $u(n)$, produces the same zero-state output $y(n)$, but whose input $\hat{u}(n)$, output $\hat{y}(n)$, and state variables $\hat{x_i}(n)$ are modified with respect to the corresponding variables in the prototype as follows:

$$\begin{aligned} \hat{u}(n) &= \frac{u(n)}{e_u(n)} \\ \hat{y}(n) &= \frac{y(n)}{e_y(n)} \\ \hat{x_i}(n) &= \frac{x_i(n)}{e_{x_i}(n)}, \quad i = 1, ..., n \end{aligned} \tag{2}$$

where $e_u(n)$, $e_y(n)$ and $e_{x_i}(n)$ are appropriate positive control signals. For example, these signals can be approximately equal to the envelope of the corresponding signals in the prototype, but this is not a necessary assumption for the derivation to follow. Using (2) in (1) produces the state equations of a new system:

$$\begin{aligned} \hat{x}(n+1) &= \hat{A}(n)\hat{x}(n) + \hat{B}(n)\hat{u}(n) \\ \hat{y}(n) &= \hat{C}(n)\hat{x}(n) + \hat{d}(n)\hat{u}(n) \end{aligned} \tag{3}$$

where $\hat{A} = (\hat{a}_{ij})$, $\hat{B} = (\hat{b}_i)$, $\hat{C} = (\hat{c}_j)$, and $\hat{d}(n)$ have the same dimensions as A, B, C and d, and their elements, after some algebra, are found to be given by:

$$\begin{aligned} \hat{a}_{ij}(n) &= a_{ij}\frac{e_{x_j}(n)}{e_{x_i}(n+1)} \\ \hat{b}_i(n) &= b_i\frac{e_u(n)}{e_{x_i}(n+1)} \\ \hat{c}_j(n) &= c_j\frac{e_j(n)}{e_y(n)} \\ \hat{d}(n) &= d\frac{e_u(n)}{e_y(n)} \end{aligned} \tag{4}$$

Thus, in the modified system in Fig. 2(b) and in equations (3)-(4), the internal signals can be controlled by the e-controls. These internal signals can thereby be scaled to any desired value with respect to the corresponding internal signals of the prototype, and this can be done dynamically without causing disturbances at the output. Thus, if the e-signals are assumed to be an independent control input, the system is internally time-varying, but its external behavior is time-invariant and identical to that of the prototype LTI system. If the e-controls are derived from the input, as suggested in Fig. 1(c), the system becomes internally nonlinear, while its external behavior is still the same as that of the prototype. As shown in Fig. 1(c), in a complete system including an ADC and a DAC, the input and output scaling is in the analog domain, and corresponds to variable-gain amplification.

The input-output behavior of the two systems in Fig. 2 will be identical only if there is no quantization; in the presence of the latter, the output quantization errors of the two systems (for a given input) will be different. The above equations can be re-written including the quantization error $q(t)$. Due to lack of space, suffice it to observe that, whereas in Fig. 2(a) the input to the DSP is $u(n) + q(n)$, in Fig. 2(b) that input becomes $e_u^{-1}(n)u(n) + q(n)$. When the envelope of $u(n)$ is small, the factor $e_u^{-1}(n)$ can be made large; this strengthens the signal whereas the error remains the same, and results in an improvement of the signal-to-quantization-error ratio by a factor of $e_u^{-1}(n)$. Similar results can be obtained for the errors due to limited-precision fixed-point arithmetic in the DSP.

3. COMPANDING DSPS

In the development that led from equations (1) and (2) to equations (3)-(4), no specific form was assumed for the e-controls; the above derivation shows that, in principle, the systems in Fig. 2 would be input-output identical if there were no quantization error, independent of the exact nature of the e-controls. However, in the presence of ADC quantization effects and fixed-point arithmetic, these controls should be chosen appropriately to help minimize the quantization error at the output. All signals in the prototype will be assumed normalized so that the maximum value of their envelope is 1. The internal signal envelopes in the companding system should be kept close to this maximum value, so that most bits are exercised for adequate signal-to-noise ratio. In simple cases with low-order DSPs, all signals can have a similar envelope, in which case the e-controls can all be appropriately scaled and delayed versions of the input envelope (plus a small positive safety margin, to avoid division by 0); this results in a very simple implementation. However, the best possible choice for the e-controls is to make each equal to the envelope of the corresponding signal in the prototype, so that the companded signals $\hat{u}$, $\hat{y}$ and $\hat{x_i}$ in (2) have approximately constant envelopes; the approximation results from quantization errors in the signals of the prototype and from imperfections in the envelope extraction process. We will adopt this choice in order to test the principles presented. Thus, in the rest of this paper, $e_v(n)$ will represent the envelope of a signal $v(n)$. For now, the envelope can be considered a slowly-varying signal that connects the peaks of $|v(n)|$, plus a small positive safety margin as above. The e-controls can be derived by using the prototype as a companion system, obtaining its u, y, and x_i signals, and computing their envelopes. The limited precision of the prototype is not a serious problem, as it is not important to develop the envelope very accurately; all that is required is an approximation, to help make the envelopes of the companded signals large and roughly constant. In some cases, it should be possible to share the same hardware between the prototype and the companding system. It should be noted that only ratios of envelopes are involved in controlling the companding system, as seen in (4). In most cases, this results in great simplification. Specifically, it is almost always only necessary to compute envelopes for a very small subset of the x_i signals from the prototype. Such simplifications will be seen in the example presented in the next section.

4. CASE STUDY: A COMPANDING REVERBERATOR

4.1. Prototype

As an example, we consider the implementation of companding on the reverberator shown in Fig. 3 [5]. Taking the states at the outputs of each of the k delay elements, the state equations are:

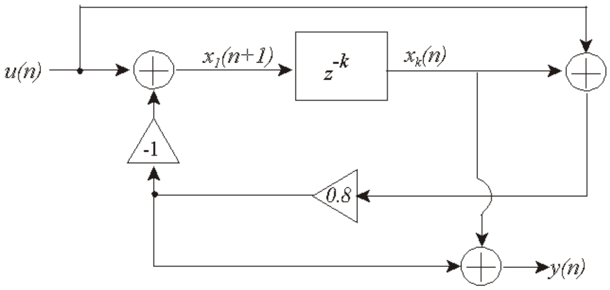

Fig. 3. A simple all-pass reverberator stage.

$$\begin{array}{rcl} x_1(n+1) & = & -0.8x_k(n)+0.2u(n) \\ x_i(n+1) & = & x_{i-1}(n), \;\; 2{\leq}i{\leq}k \\ y(n) & = & 1.8x_k(n)+0.8u(n) \end{array} \qquad (5)$$

When these equations are put in the form of (1), we get: $a_{i,i-1} = 1, \;\; 2{\leq}i{\leq}k$; $a_{1k} = -0.8$; $b_1 = 0.2$; $c_k = 1.8$; $d = 0.8$. All other entries in A, B and C are zero.

4.2. Companding System

To develop the corresponding coefficients for the companding system using (4), we first observe that, for the state variables in the z^{-k} block, we have $x_i(n+1) = x_{i-1}(n), \;\; 2{\leq}i{\leq}k$, and $x_k(n) = x_1(n+1-k)$. Thus, corresponding relations also hold for the envelopes of these state variables, implying that
$e_{x_i}(n+1) = e_{x_{i-1}}(n), \;\; 2{\leq}i{\leq}k$, and $e_{x_k}(n) = e_{x_1}(n+1-k)$. Using these relations in (4), we observe that in all elements of the form $\hat{a}_{i,i-1}$, the e-controls cancel out. Thus, applying (4), we obtain: $\hat{a}_{i,i-1} = 1, \;\; 2{\leq}i{\leq}k$; $\hat{a}_{1k} = -0.8\frac{e_{x_k}(n)}{e_{x_1}(n+1)}$; $\hat{b}_1 = 0.2\frac{e_u(n)}{e_{x_1}(n+1)}$; $\hat{c}_k = 1.8\frac{e_{x_k}(n)}{e_y(n)}$; $\hat{d} = 0.8\frac{e_u(n)}{e_y(n)}$. All other entries in $\hat{A}$, $\hat{B}$ and $\hat{C}$ are zero. The state equations (3) for the companding processor become:

$$\begin{array}{c} \hat{x}_1(n+1) = [-.8\frac{e_{x_1}(n+1-k)}{e_{x_1}(n+1)}]\hat{x}_k(n) + [.2\frac{e_u(n)}{e_{x_1}(n+1)}]\hat{u}(n) \\ \hat{x}_i(n+1) = \hat{x}_{i-1}(n), \;\; 2{\leq}i{\leq}k \\ \hat{y}(n) = [1.8\frac{e_{x_1}(n+1-k)}{e_y(n)}]\hat{x}_k(n) + [.8\frac{e_u(n)}{e_y(n)}]\hat{u}(n) \end{array} \qquad (6)$$

As seen above, $k-1$ of the k e-controls corresponding to the k states of the k-delay block cancelled out in the ratios in (4). This will occur for every k-delay block in every system. This result makes sense; if the input of a k-delay block is properly companded, then all the internal states of this block will also be properly companded, so no e-controls for the intermediate state variables are needed. This greatly simplifies the implementation.

4.3. Quantitative Results

We have used Matlab/Simulink to implement, at a sampling rate of 44.1 kHz, a reverberator consisting of a cascade of two stages, each of them being of the form shown in Fig. 3, with $k = 2000$ for the first stage and $k = 4410$ for the second. We note that in the first and third equations in (6), the right-hand sides consist of sums of two terms, which add up to the companded variables on the left. Although the variables on the left are properly companded and have roughly constant envelopes, the envelopes of the individual terms in the above sums cannot, in general, be expected to be constant. For this reason, assuming that the companded variables are represented by N bits, we used N bits for the factors in brackets, and $2N$ bits for the products of these factors with the companded variables; after $2N$ bit summation, the results were converted to N bits.

The prototype system and the companding system were simulated using a sinusoidal input. The simulations were run in fixed-point with 8, 9, 10, and 11 bit systems, and used the same precision for the envelope computation. For the purposes of this experiment, the envelope $e_v(n)$ of a signal $v(n)$ was a small non-negative amount (to avoid division by 0) plus a stair-case signal with a step starting at each local peak of $|v(n)|$, except at points where this signal was lower than $|v(n)|$, in which case it was replaced by the latter. This approach does not catch all nuances of the envelope variation of $v(n)$, so the resulting companding signals still have some envelope variation; nevertheless, good results are obtained, as will be seen. Since detecting a local peak at time-step n_0 requires knowledge of $|v(n_0+1)|$, a single-sample delay was inserted in front of the divider in Fig. 1(c) to make envelope computation a causal process.

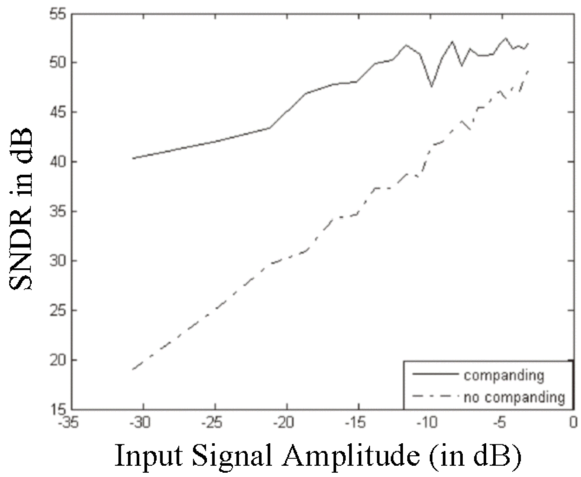

Fig. 4. SNDR for 11-bit implementations of the processor of Fig. 1(a) and the companding processor of Fig. 1(c).

The steady-state output was observed for various input amplitudes. Fig. 4 shows the signal-to-noise-plus-distortion ratio (SNDR) for the case of 11 bits. As seen, whereas the SNDR for the non-companding system varies in proportion to the input signal, the SNDR of the companding system stays relatively constant over a large range of input values. The same qualitative behavior was also verified for the cases of 8, 9, and 10 bits.

We used speech waveforms to assess the performance of the prototype and of the companded system under realistic conditions. We first simulated both processors for the case of no quantization, i.e. we assumed that the ADC and DAC had infinite precision, and we used double-precision floating-point arithmetic. The purpose of this simulation was to verify the theoretical results in Sec. 2, which assumed no quantization in the processors. The state variable waveforms of the two systems were very different; the ones in the prototype had widely varying envelopes, whereas the ones in the companded system had envelopes that were roughly constant, as expected. Yet the output of the companding system in Fig. 1(c) was virtually identical to the output of the prototype in Fig. 1(a), with the two outputs differing by at most 10^{-15}. These results verified the fact that, although the companding system is internally nonlinear, it is input-output equivalent to the prototype system.

Next, we ran the three systems of Fig. 1 in 8-bit fixed-point, with the ADCs and DACs also being 8-bit. The results are shown in Fig. 5. The speech input in Fig. 5(a) was common to all three systems. The state variable shown is the input to the k-delay block in the first stage of the reverberator. The envelope of this state variable varies considerably in the prototype system, as shown in Fig. 5(b), whereas it is roughly constant at a large value in the companded system, as shown in Fig. 5(c). (Note that the vertical scales of Fig. 5(b) and Fig. 5(c) are different.) At the same time, the output of the latter, in Fig. 5(f), is essentially identical to that of the prototype, in Fig. 5(d). In contrast, the output of the system of Fig. 1(b), shown in Fig. 5(e), is grossly distorted, illustrating the inadequacy of that approach and the need to also compand the state variables, as proposed in this paper.

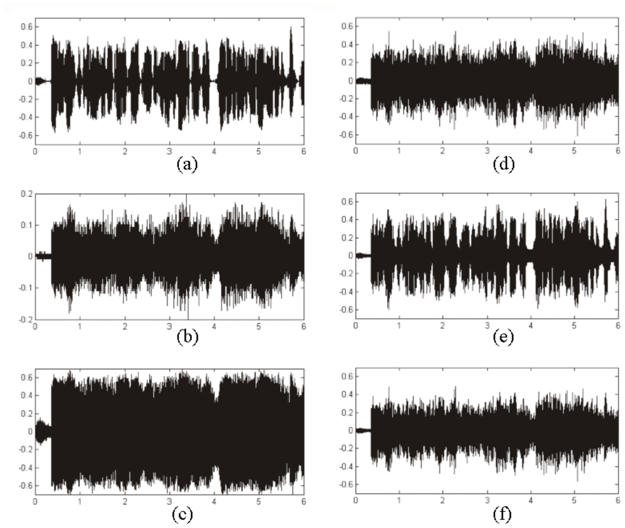

Fig. 5. Waveforms of input, output, and internal state in the presence of a speech signal, all versus time in seconds: (a) Input. (b) State in prototype system of Fig. 1(a). (c) State in companded system of Fig. 1(c). (d) Output of prototype system in Fig. 1(a). (e) Output of system in Fig. 1(b). (f) Output of companded system in Fig. 1(c).

4.4. Qualitative Results

We also performed listening tests with speech and music inputs. When run in full precision, the output of the system in Fig. 1(c) sounded identical to that of the prototype in Fig. 1(a), as expected, whereas the output of the system in Fig. 1(b) sounded grossly distorted. With fixed-point arithmetic, a background "hiss," caused by quantization noise, was audible at the output of both the prototype of Fig. 1(a) and the properly companded processor in Fig. 1(c). However, whereas the prototype system's hiss was relatively constant, in the companding system it became less audible during softer music passages, and was inaudible when the signal was relatively quiet. Representative audio files have been posted on a Web site [6].

5. CONCLUSIONS

This paper has developed a method that allows dynamic scaling of the internal variables in a DSP, without producing transients at the output. This method has been applied in the reduction of the effect of quantization error in ADCs and fixed-point DSPs, by using companding of the signals involved, so that most of the bits available are exercised most of the time. Thus companding, a technique widely used in transmission and recording, was shown to be transferable to DSPs by properly taking the dynamical nature of the latter into account. The resulting companding processor is internally nonlinear; despite this, assuming no quantization, its input-output behavior remains identical to that of the original LTI system used as a prototype. In the presence of quantization, it was shown that the companding system has significantly smaller quantization error than the original system.

Simulations were run for several cases and with sinusoidal, speech, and music signals, and quantitative and listening tests were performed. The results obtained have confirmed the theoretical principles, and the quantitative performance obtained suggests that a hardware implementation may well be worth undertaking. An economical hardware implementation of the scheme proposed may find use in such applications as multimedia, where large amounts of processing are needed, and thus fixed-point arithmetic using a small number of bits is highly desirable.

6. REFERENCES

[1] Member of the Technical Staff of Bell Telephone Laboratories, *Transmission Systems for Communications*, Western Electric Company, Winston-Salem, NC, 1970.

[2] R. M. Dolby, "Signal compressors and expanders," US Patent 3,345,416, Oct. 1974.

[3] E. Blumenkrantz, "The analog floating point technique," in *Proc. 1995 IEEE Symp. Low-Power Electronics*, Oct. 1995, pp. 72–73.

[4] Y. Tsividis, "Externally linear time-invariant systems and their application to companding signal processors," *IEEE Trans. Circuits and Systems II*, vol. 44, pp. 65–85, Feb. 1997.

[5] S. Mitra, *Digital Signal Processing: A Computer-Based Approach*, p. 782, McGraw-Hill, New York, 2001.

[6] "http://www.cisl.columbia.edu/∼aek84/spconf2006.html," .