Proc. of the 5th Int. Conference on Digital Audio Effects (DAFX-02), Hamburg, Germany, September 26-28, 2002

OPTIMIZING DIGITAL MUSICAL EFFECT IMPLEMENTATION FOR MULTIPLE PROCESSOR DSP SYSTEMS

Jiri Schimmel, Zdenek Smekal, and Petr Krkavec

Department of Telecommunications
Faculty of Electrical Engineering and Communication
Brno University of Technology
schimmel@feec.vutbr.cz, smekal@feec.vutbr.cz

ABSTRACT

In the area of digital musical effect implementation, attention has lately been focused on computer workstations designed for digital processing of sound, which perform all operations with audio signals in real time. They are in fact a combination of powerful computer program and hardware cards with digital signal processors. Thanks to the power enhancement of personal computer core, performing these operations in the CPU is currently possible. However, in most cases, digital signal processors are still used for these purposes because digital musical effect modelling is more effective and more precise with the digital signal processor. In addition to this, processing in digital signal processor saves the CPU computing power for other functions.

1. INTRODUCTION

The first step in optimizing an algorithm is optimizing the algorithm for the architecture of the digital signal processor (DSP) used [1], most frequently of the Motorola DSP563xx family. However, when we want to implement the algorithm on a multi-processor DSP system, we have further possibilities that depend on the given system architecture.

The optimization possibilities are presented on the algorithm of the DSound VL2 non-linear musical effect that uses the non-linear transfer characteristic of a tube amplifier with an ECC-83 tube in the starved-plate mode to enhance the musical signal spectrum by adding the higher harmonics. The algorithm is optimized for the TC Works Powercore system [2] with dynamic allocation of resources.

2. NON-LINEAR SYSTEM WITH VARIABLE RATIO OF HIGHER HARMONICS

2.1. Spectrum Components in Power Polynomial Approximation

When approximating the transfer characteristic with an n^{th}-order polynomial, the output voltage u_2 depends on the input voltage $u_1 = U_1 \cos(\omega_0 t)$ according to the equation

$$u_2 = a_0 + a_1 U_1 \cos(\omega_0 t) + \ldots + a_n U_1^n \cos^n(\omega_0 t), \qquad (1)$$

where a_0 to a_n are the coefficients of approximation polynomial.

We can express output signal of the system using Fourier series from which the amplitudes of individual harmonics can be determined. If we establish Euler form of cosine function to binomial series we get the general equation for establishing k^{th} harmonic of output signal

$$U_{2k} = \sum_{n=0}^{\infty} \frac{(2n+k)!}{2^{2n+k-1} n!(n+k)!} a_{2n+k} U_1^{2n+k} , \qquad (2)$$

where U_{2k} is the amplitude of the k^{th} harmonic of output signal, U_1 is the amplitude of the input signal, and a_n is the n^{th} coefficient of the polynomial that approximates the system transfer characteristic.

2.2. Possibilities of Change of Higher Harmonics Ratio

Possibility to change higher harmonics ratio of signal is important property of digital musical effects that use spectral features of system with non-linear transfer characteristic. According to analogue musical effects we can give a name "Saturation" to parameter allowing this change.

2.2.1. Change of working point and range

Higher harmonics ratio of output signal depends on polynomial section in which input signal is situated. Output signal spectrum of such system with two saturation settings is in Figure 1 and 2. However, change of input signal level causes change of saturation and input signal range for that is polynomial defined can be exceeded - it causes output signal limitation.

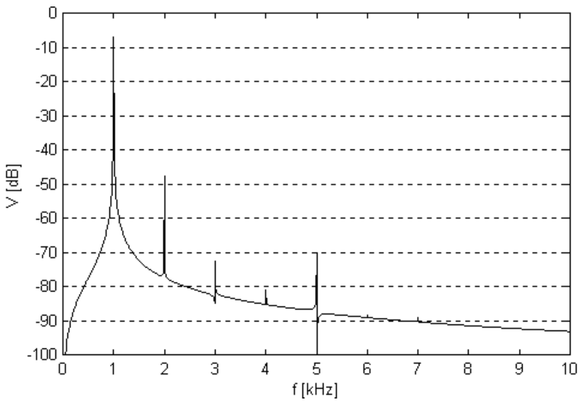


Figure 1: *Output signal spectrum with lower saturation.*

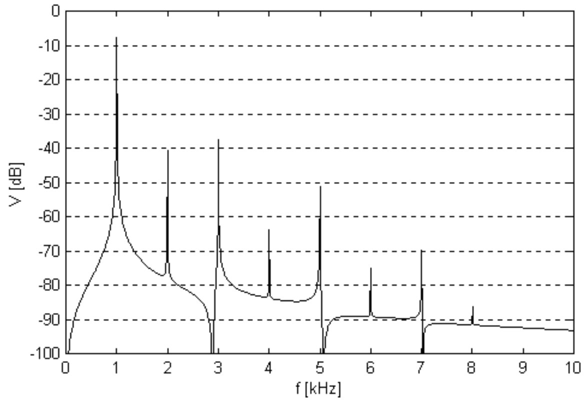


Figure 2: *Output signal spectrum with higher saturation.*

2.2.2. *Evaluation of polynomial coefficients*

Other possibility is evaluation of polynomial coefficients according to required harmonics ratio of output signal. If we modify equation (2) and introduce the substitution $B_k = U_{2k}/U_1$ we get the relation

$$\frac{2^{k-1}}{U_1^{2k-1}} B_k = \sum_{n=0}^{m} \frac{(2n+k)!}{2^{2n} n!(n+k)!} a_{2n+k} U_1^{2n-k} \qquad (3)$$

where B_k is the ratio of k^{th} harmonic amplitude with respect to the first harmonic amplitude, U_1 is the amplitude of the input signal, and a_n is the n^{th} coefficient of the polynomial that approximates the system transfer characteristic.

If we itemize equation (3) for $k = 0 \ldots m$, we get a system of m+1 equations with m+1 unknowns. The solution of this equation system are coefficient values of the polynomial that approximates the transfer characteristic of a non-linear system and satisfies the requirements for higher harmonic amplitude ratios with respect to the amplitude of the first harmonic. See [4] for details. However, the question is how to compute required higher harmonics ratios along one parameter (Saturation). Also computing power of algorithm is relevant.

2.2.3. *Change of first to higher harmonics ratio*

System in Figure 3 holds higher harmonics mutual ratios with change of saturation. System changes only higher harmonic amplitude ratios with respect to the amplitude of the first harmonic.

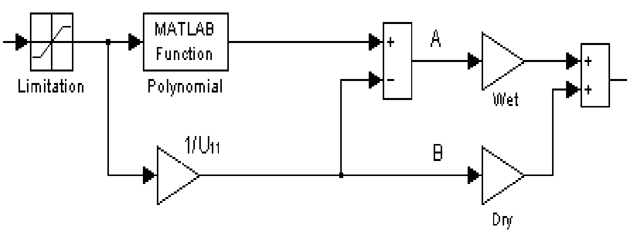


Figure 3: *Non-linear system with change of saturation.*

Input signal is subtracted from output signal after input signal passes through non-linear system, so only higher harmonics are obtained. According to (2) amplitude of the first harmonic U_{11} will be

$$U_{11} = \sum_{n=0}^{\infty} \frac{(2n+1)!}{2^{2n} n!(n+1)!} a_{2n+1} U_1^{2n+1} \qquad (4)$$

where U_1 is maximum value of input signal. Amplitude of all higher harmonics can be computed also according to (2). Signal of the higher harmonics u_A will be in point A in Fig. 5. In the worst case its maximum value will be

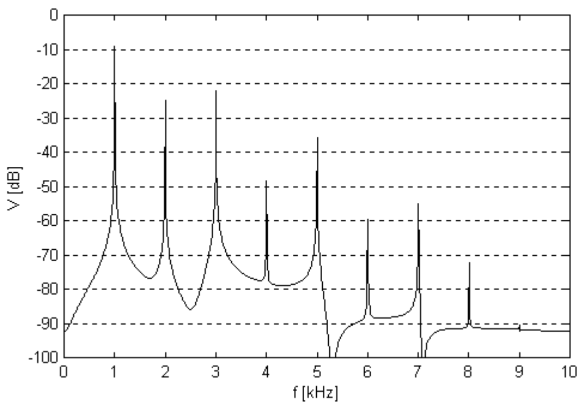
$$U_A = a_2 U_1^2 \cos^2(0) + \ldots + a_n U_1^n \cos^n(0) = \sum_{i=2}^{n} a_i U_1^2 \qquad (5)$$

The equation holds for output signal u_2 of the system in Figure 3

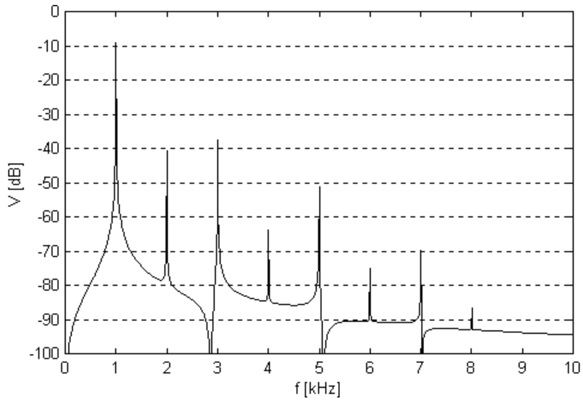
$$u_2 = Wet \cdot u_A + Dry \cdot u_B \qquad (6)$$

where $u_B = u_1/U_{11}$. If we introduce variable *Saturation*=*Wet/Dry* and assume $U_2 = U_1$, we can determine the *Wet* and *Dry* values from (5), (6), and (7).

As can be seen from Figure 4 and 5 the higher harmonics ratio changes independently with saturation settings and with input signal amplitude.



a)



b)

Figure 4: *Higher harmonics ratio of signal with amplitude of 1 and saturation of 1 (a) and 5 (b)*

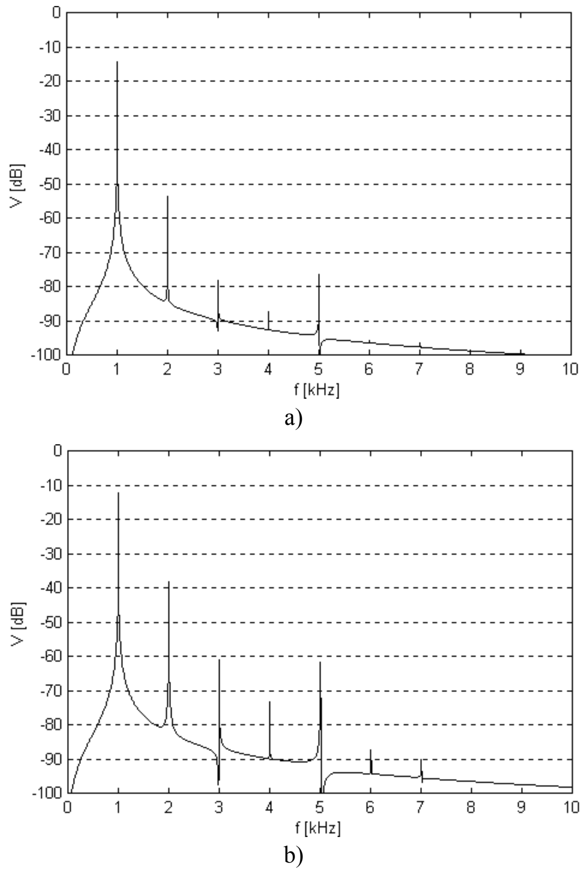


Figure 5: *Higher harmonics ratio of signal with amplitude of 0.5 and saturation of 1 (a) and 5 (b)*

3. ALGORITHM IMPLEMENTATION ON FIXED-POINT DIGITAL SIGNAL PROCESSOR

Fixed-point digital signal processors can work with numbers in interval $<-1,1-2^{n-1}>$, where n is data bus width. Input and output signal, and constants must be in this range. Transfer characteristic is used in whole range that is defined in input signal interval $<u_{1A},u_{1B}>$ for which the output signal is in range $<f(u_{1A}),f(u_{1B})>$. It is useful if $u_{1A}=-u_{1B}$. Then we produce new signal $u_1' = C.u_1$ by multiplying input signal by constant $C_1=(1-2^n-1)/u_{1B}$. For polynomial approximation we obtain equation

$$u_2 = a_0' + a_1'u_1' + a_2'u_1'^2 + a_3'u_1'^3 + ... + a_n'u_1'^n \quad (7)$$

$$a_k' = a_k/C_{1k} . \quad (8)$$

Processor ensures limitation outside range in which the approximation polynomial is defined. It holds that $f(u_{1A})=\min(u_2)$ and $f(u_{1B})=\max(u_2)$ if the approximation polynomial is creasing function in whole range. According to (9) we obtain equation by multiplying input signal by constant $C_2 = (1-2^{n-1}) / \max(f(u_{1A}), f(u_{1B}))$

$$u_2' = C_2u_2 = a_0'' + a_1''u_1' + a_2''u_1'^2 + ... + a_n''u_1'^n \quad (9)$$

$$a_k'' = C_2 . a_k' . \quad (10)$$

If we change approximation polynomial coefficients using equation (9) and (11) we ensure that polynomial is defined in whole range of input signal and output signal will be in arithmetic range of fixed-point DSP.

However, this does not ensure that coefficients are in $<-1,1-2^{n-1}>$ range and they are not too small (it increases computing inaccuracy). Equation (10) can be written in form

$$u_2' = K\left(a_0''' + a_1'''u_1' + a_2'''u_1'^2 + ... + a_n'''u_1'^n\right) \quad (11)$$

where $K = \max({}_a k'')/(1-2^{n-1})$ and $a_k''' = a_k''/K$. Block diagram of modified algorithm is in Fig. 6.

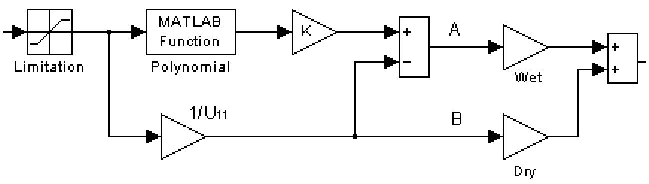


Figure 6: *Modified non-linear system with saturation*

3.1. Parameter Choice and Data Transfer

The digital musical effect dedicated to a DAW system can be divided into two parts - user interface and processing algorithm. The algorithm processes the digital musical signal according to variables that are obtained from the user interface. These variables needn't have a direct link to the algorithm variables. It is necessary to compute algorithm variables according to the settings of GUI parameters. However, this entails further (and in many cases useless) load of the processing algorithm.

Computing the algorithm variables according to the GUI parameters set can be performed directly in the user interface or in the background DSP code. In the former case the load of the host environment increases but only minimally because the computation has to be performed only when a parameter changes. A disadvantage is that a great number of variables has to be transferred several times in succession in the case of spontaneous parameter changes and the communication interface may get saturated.

3.1.1. Algorithm Optimizing

Equation holds for output signal value u_2 of the system in Fig. 6

$$u_2 = C_{DRY} \cdot U_{11} \cdot u_1 + C_{SAT} \cdot \left(K.f(u_1) - U_{11}u_1\right) \quad (12)$$

where u_1 is input signal value, U_{11} is first harmonic amplitude after input signal passes through non-linear system, $f(u_1)$ is function that approximates transfer characteristic of non-linear system, and C_{DRY}, C_{SAT}, and K are constants. This equation is too complex. We have to optimize it so number of performed mathematical operations is minimal. If we introduce substitutions

$$c_1 = C_{SAT} \cdot K$$
$$c_2 = U_{11} \cdot (C_{DRY} - C_{SAT}) \quad (13)$$

we can write equation (13) in form

$$u_2 = c_1 \cdot f(u_1) + c_2 \cdot u_1 \quad (14)$$

If function $f(u_1)$ is polynomial of n^{th} order, equation (15) is in form

$$u_2 = c_1\left(a_0 + a_1 u_1 + a_2 u_1^2 + \ldots a_n u_1^n\right) + c_2 \cdot u_1 \quad (15)$$

If we introduce substitutions $c_3 = c_1/2^m$ and $c_4 = c_2/(c_1.2^m)$ we obtain system equation suitable for implementation on fixed-point digital signal processor

$$u_2 = 2^m \cdot c_3\left(a_0 + c_4 u_1 + \sum_{k=1}^{n} a_k u_1^k\right) \quad (16)$$

where $2^m > \max(c_1)$ and a_k are coefficients of polynomial that approximates transfer characteristic of non-linear system. Only c_3 and c_4 coefficients change in (17) with change of *Saturation* parameter. This computation can be performed in plug-in interface and only c_3 a c_4 values can be transferred into digital signal processor memory. Next step is optimizing of n^{th}-order polynomial computation on digital signal processor. Polynomial equation can be written in the form

$$u_2 = a_0 + a_1\prod_{i=1}^{1} u_1 + a_2\prod_{i=1}^{2} u_1 + \ldots + a_n\prod_{i=1}^{n} u_1 \quad (17)$$

If we introduce state variables

$$v_0 = 1, v_k = u_1.v_{k-1} \text{ for } k = 1..\, n \quad (18)$$

we can write equation (18) in the form

$$u_2 = a_0 + a_1 v_1 + \ldots + a_n v_n = a_0 + \sum_{i=1}^{n} a_i v_i \quad (19)$$

Using equations (17), (19), and (20) we obtain equation of digital musical effect algorithm based on signal passing through non-linear system with transfer characteristic approximated by power polynomial with possibility of change of higher harmonics ratio:

$$u_2 = 2^m \cdot c_3\left(a_0 + c_4 u_1 + \sum_{k=1}^{n} a_k v_k\right) \quad (20)$$

where u_1 is input signal value, c_3 and c_4 are constant dependent on *Saturation* value, a_k are approximation polynomial coefficients, and v_k are state variables.

3.2. Serial and Parallel Processing

The number of instruction cycles that one DSP can perform during processing the elementary data blocks is most constraining factor. Thus the computing power linearly depends on the ratio of DSP clock rate and sampling frequency. Dividing the algorithm into more processes can be used especially in multi-processor systems with dynamical allocation of computing power. We can spread algorithm processing onto more processes that are in series or in parallel. The former case is suitable for algorithms, which are composed of several subsequent blocks (process 2 can process the data block that has been previously processed by process 1, while process 1 processes a next data block). The processing of N data blocks takes the time t_b

$$t_b = n(N + K - 1)\sum n_{cycle} t_{cycle} \quad (21)$$

where n is the data block length, K is the number of serial processes, n_{cycle} is the number of instruction cycles of the given process, and t_{cycle} is one instruction cycle period.

Parallel processing expects that the algorithm can be divided into parallel branches. The time of processing N data blocks t_b does not depend on the number of processes

$$t_b = n \cdot N \sum n_{cycle} t_{cycle}\,. \quad (22)$$

3.2.1. Algorithm Adaptation

The parallel division according to the audio channels is one of the ways of parallel processing. Hardware DO cycle can be used for processing more channels by one process if every channel is processed by same algorithm. However, it linearly increases DSP load and number of required instruction cycles can exceed computing power of one DSP in the case of multichannel digital audio effect. To avoid this problem, digital audio effect can be divided into more DSP processes that process smaller number of audio channels. These processes can share memory with algorithm variables and polynomial coefficients (see Figure 7). Control part of effect determines number of processes according to load of individual DSPs and the number of channels processed by individual processes.

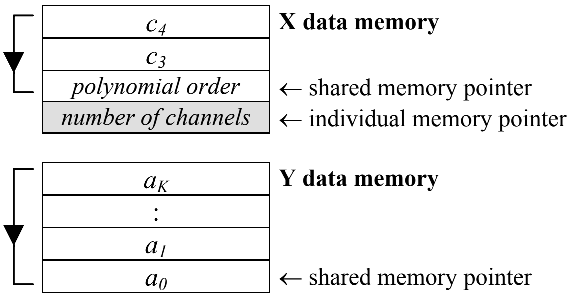


Figure 7: *DSP memory usage for more parallel prosesses*

4. CONCLUSIONS

Algorithm described above is used in DSound VL2 multi-channel valve interface. It processes as many channels as actually needed (this save computing power) and it can establish more processes if no DSP on Powercore board has enough computing capacity for processing all channels. Polynomial coefficients and order can be also changed during processing.

5. REFERENCES

[1] Smekal, Z., Schimmel, J., Krkavec, P. Optimizing Digital Musical Effects Implementation for Harvard DSP Architecture. *Proceedings of COST-G6 conference on Digital Audio Effects DAFx-01*, Limerick, December 6-8, 2001, pp.33-38, ISBN 187465363-1

[2] TC Powercore documentation. TC Works, 2001. (http://www.tcworks.de/)

[3] Čajka, J., Valsa, J. TERO 9 (Non-linear Circuits and Circuits with time variable parameters). Lecture notes VA AZ, Brno, 1966. (in Czech)

[4] Smekal, Z. et al.: Final research report on the solution of international project No OC G6 10 for the year 2001.