Cancelling Acoustic Feedback with the A.L.M.I. Algorithm

5127 (J - 6)

Manuel Almiñana Iturbe,
Telecommunications Tech. Engineer; Prof. A&V Consultant, Albacete, Spain

Presented at
the 108th Convention
2000 February 19-22
Paris, France

AN AUDIO ENGINEERING SOCIETY PREPRINT

Cancelling Acoustic Feedback with the A.L.M.I. Algorithm.

Manuel Almiñana Iturbe
Telecommunications Tech. Engineer,AES Associate member; IEEE Member.
Electronic Equipments Professor - Third Parties Developer/ Consultant.

ABSTRACT

This paper describes a new algorithm called A.L.M.I., stands for "*LMS Modified Algorithm Including delay, filtering and frequency modulation*", which decorrelates any signal within some margins, so we have an effective algorithm in the struggle against the acoustic feedback. The theoretical formulation, a practical implementation example using the TMS320C5x DSP, and the results obtained are here explained.

0.- INTRODUCTION.

Let's suppose an application in a sound reinforcement system with a microphone, an amplifier and a loudspeaker; then acoustics feedback or coupling is produced when increased amplification gain over a given stable margin exists, something common in these systems.

As we know, the microphone captures the sound from the audio source, then it's amplified and if we have the loudspeaker in the same room it also captures its own amplified sound, starting the loop over and over again, therefore the acoustic feedback or acoustic coupling begins (see Figure 1). This phenomenon presents serious inconvenient and limits quantitatively the maximum system gain. Furthermore, it will present a characteristic frequency response, due to the fact that the microphone-loudspeaker distance or any reflection has a finite value, and it will often favour the feedback to the frequencies whose wavelength are related to these distances, therefore the signal is "coloured" and both feedback and reverberation are forced.

Although some equipment use active equalization (in a manual way, so that the sound technician reduces the gain in the conflicting bands; or automatic as well, through the use of auto-equalizers based on filter banks that detect the peaks on the frequency response, which was caused by the feedback, by band estimation), the total gain obtained does not tend to surpass 6 dB as compared to a system without protection. Furthermore, the circumstance that provokes the feedback can vary in time (for example, microphone-loudspeaker distance; phase response ...), and the detection speed of these equipments cannot eliminate the problem succesfully (what is relaxing for the reinforcement system operator).

On the other hand, researches on echo canceller devices for long-distance telephon calls have favoured the appearance of effective algorithms in this field and nowadays we can use digital signal processors which are capable of executing these algorithms in real time.

Starting from some theories on adaptive filters [1];[2];[3], telephonic echo cancellation applications [4];[5] and active noise control [6];(among others), the present development work has been finished obtaining the most efficient algorithm in the struggle against the phenomenon of the acoustic feedback.

The implementation of this algorithm ("A.L.M.I.", stands for *Algoritmo LMS Modificado con retardo, filtrado y modulación en frecuencia Incorporados* ,or in English, LMS Modified Algorithm Including delay, filtering and frequency modulation) on an integrated digital signal processor permits its utilization in real time, and it can be fulfilled on an electronic equipment intended for improving the electroacoustic systems as a rule, and the sound reinforcement through P.A. equipment in particular.

Returning to the previous figure, and since the microphone collects the signal *s(t)* to amplify and at the same time part of the signal that was already amplified enters, a feedback loop is established, characterized by its temporary impulse and frequency response. For example, as you can see in figure 2.

When the loop gain is over a given threshold, the system becomes unstable, being presented the "acoustic coupling" and/or the positive feedback. In this case, the system gain must be sufficiently reduced (in the appropriate band or globally), or we should cut the input signal, actions that will produce undesirable effects in the sound reinforcement system.

The feedback loop reflections and the frequency response composition will depend on multiple factors such as mike directionality, system frequency response, reflecting walls presence..., and furthermore these can vary in time.

As a rule we can characterize the feedback signal, in a given moment *i*, as:

$$r(i) = \sum_{k=0}^{\infty} h_k x(i-k) \tag{1}$$

where h_k are the coefficients that cause the feedback signal frequency response, and the temporary distribution of the reflections; and $x(t)$ the original input sound signal [7];[8].

According to the foregoing, interposing between microphone and amplifier a device capable of "subtracting" from the input signal the feedback corresponding part, we will obtain as a result to avoid that this phenomenon appears.

Reality provokes that this fact will be impossible, since first, we should know the input signal during an infinite time interval in order to extract the necessary information, and this should be available before it is produced. However, through some help and simplifications we can obtain useful practical results.

If in the moment *i* the input signal to the device is $s(i)+r(i)$, with $s(i)$ as the signal that must be amplified and $r(i)$ as the acoustic feedback signal, and if somehow we have the signal $r(i)$ separately, then we could subtract the $r(i)$ to the input obtaining the output $s(i)$ as a signal without feedback. This is showed in figure 3.

The difficulty of the problem lies in obtaining a signal r(i) before it has been produced. Though *a priori* it seems impossible, we can take into account certain factors and introduce some modifications in order to obtain an available r(i) signal.

From the characterization of the feedback signal according to (1), we wish to obtain the coefficients that permit to add an approximation to the signal *r(i)* which we will call *c(i)*, identified by its coefficients a_k :

$$c(i) = \sum_{k=0}^{N-1} a_k x(i-k) \tag{2}$$

available from moment i, and until a finite value *(N-1)*; so that:

$$r(i) = \sum_{k=0}^{\infty} h_k x(i-k) \approx c(i) = \sum_{k=0}^{N-1} a_k x(i-k) \tag{3}$$

The closer is the approximation between *r(i)* and *c(i)*, the better the system will operate, and in fact, the output signal will be free (of course, in certain approximation degree) of the feedback that took place in the acoustic path.

This affirmation implies:

$$a_k = h_k$$

that will be the objective of the algorithm. At this stage, we should take into account some characteristic factors of the process:

- Feedback is always delayed with respect to the output signal desired.
- For large values of k, the value of the corresponding coefficient h_k is negligible.
- The directional characteristics of the equipment hint to a mid-high frequency range as more susceptible of suffering feedback.

As we do not have the separate signals *s(i)* and *r(i)* (of the system input), we should use as a reference the only one available input signal, that is to say, the sum of both. Endowing the system with some "memory", and knowing that feedback will present a pronounced spectral component, we should decorrelate the input signal. This will present certain inconvenients, so the system will be more efficient under the following circumstances:

- When the *s(i)* signal doesn't present very pronounced frequency components, or, if any, these are not maintained for a long time.
- When the signal is free of special effects as delays, reverberations, etc.

For the obtaining of the a_k coefficients we will use an algorithm that minimize the difference between the input and the estimate of feedback, but introducing delay and filtering (since, otherwise, the logic conclusion is that the system simply makes its outputs equal to zero).

On the other hand, and to avoid an excessive decorrelation that would cause the output disappearance (or the maximum flatness as white noise), we will use a pseudo-frequency modulator from the two-way reading of an intermediate buffer. This will produce tone variations over the input signal, but they can be maintained within an imperceptible level.

Though in this introduction the development of the algorithm has been focused towards the elimination of the acoustic feedback, many other applications exist. Below, it is exposed a greater concretion of our algorithm in typical application environments:

- As an acoustic feedback supressor. So that the algorithm could accomplish this function, its parameters will be matched in a way that we can introduce the minimal distortion with the greater convergence speed. Elements such as the output delayer contribute to increase the amplification gain without feedback. In this case, the device will consist of an electronic appliance with input, output and power connections, in addition to the auxiliary elements for visualization and control. A device of these characteristics can be used in scenaries, press rooms, auditoriums, and as a rule in any sound reinforcement environment or electroacoustic transmission system.
- As a resident program for the processing in distributed time (off-line) in a computation platform with access to data files. In this case the algorithm can be designed for the cleansing of recorded signals with deficiencies such as excessive reverberation, echoes presence, etc. It will present a series of controls, mainly the type of Pre-LMS filter, which well fitted can be useful for this purpose.
- As a debugger of signals in other transmision types. Due to the unmatched impedance adaptation in transmission lines, the distortions introduced by reflected waves can produce deficiencies in the reception and even the destruction of the issuing equipment. In this case the processing is

accomplished in real time, and it will attempt to remove the reflected (echoed) signal.

- In all these applications in where we want to decorrelate a signal, or to suppress its "memory" feedback.

In any case, a device capable of executing an approximation to the exposed algorithm will be susceptible of manufacturing as an electronic appliance destinated to the professional and semi-professional sound market, in order to improve the available gain limited by acoustic feedback, improving in this sense the characteristics of a P.A. system, sound reinforcement or electroacoustic transmission.

1.- DESCRIPTION OF THE A.L.M.I. ALGORITHM.

After the initial stands have been explained in the previous paragraph, we will try to fulfill the mathematic signal processing operations needed in order to obtain the solution.

For the implementation of the dynamic search of minimization between the wished signal and the input one we will use an adaptive filter, concretely a L.M.S. (Least Mean Squares, also known as stochastic gradient algorithm [9]).

This algorithm consists of a filter (in our case, F.I.R. type) that can be adapted following the signal information that originates the difference between input and output from filter, as shows figure 4.

In our case the filter input is also the output, but manipulated congenially so that the system will be stable, as we will see afterwards.

This filter will reach the optimum value of its coefficients when the error signal, expressed as the difference between the filter input and the filter output will be null. In that moment, the filter input signal will be equal to the input signal of the system.

So that this can be certain, we need the coefficients update equation of the FIR that, in a short way, will be exposed below [1].

Being J the mean square error function of the difference signal (from now, $(i)=x(i)-f(i)=y(i)$). In this system, the output signal is also the error signal:

$$J(i)=\frac{1}{2}\int_{-\infty}^{\infty} e^{2}(i)\,p\left(e(i)\right)de(i)=\frac{1}{2}E\left[e^{2}(i)\right] \qquad (4)$$

where p(x) symbolizes the probability density function, and the hope statistics function (necessary because we do not know the signal *a priori*).

The minima equation is outlined according to:

$$\frac{\partial J(i)}{\partial a_k(i)}=0 \qquad (5)$$

and, writing the signals in function of its components, it permits to write:

$$\frac{\partial J(i)}{\partial a_k(i)}=-\left(E(y(i)x(i-n))-\sum_{j=0}^{N-1}E\left[x(i-n)x(i-j)\right]\cdot a_k(i)\right) \qquad (6)$$

Returning to the concrete signals that we are using, we outline the minimum condition using the average hope so that the minimal squared difference between the input and the feedback (treated as correlation) estimation can occur:

$$a_k(i+1)=a_k(i)-K\frac{\partial E(x(i)-c(i))^{2}}{\partial a_k(i)} \qquad (7)$$

with K fixing the gain or "convergence speed" for the algorithm.

As we don't know the average hope of the distribution for each time, let's do the convolution between the output signal of the adder (conveniently filtered, delayed and frequency modulated) with the obtained coefficients, what permits to obtain the same in each time *(i+1)* according to:

$$a_k(i+1) = a_k(i) + 2K\frac{1}{N}\sum_{n=0}^{N-1}\left[x(i-d-n) - er(i-d-n)\right] \bullet x(i-n) \tag{8}$$

with d = delay to fix, and N = length of the LMS FIR.

As we see, the system will subtract to each input signal an obtained one from its filtered, delayed and frequency modulated output. Basically, this is summarized in the figure 5.

So that this system can operate, we need also some auxiliary calculation elements, that will finally certify the A.L.M.I. (LMS Modified Algorithm Including delay, filtering and frequency modulation, hence forth).

One of these elements is a weighter in a way that when the signal levels be in low average, the convergence step (K) will be increased, and *vice versa.* It will be implemented using an integrator and an inversor (1/x), as a multiplicative covergence factor. Another element is the Previous-LMS filter. The election is critical, but as a rule it will present the form showed in figure 6.

That is to say, it must filter the continuous component (that would lead the LMS to error), and will present a growing step band in the frequencies of a greater probability of acoustic coupling (F2-F3). Their adjustment will depend on factors as sampling frequency, utilization environment, etc, and a practical implementation is showed in Figure 7. Furthermore, if it is configured as FIR, it will present a certain group delay, that conveniently fixed will permit to do without the "*d*" delayer.

We will also incorporate into the LMS coefficients updating system a block distribution, whose size will depend on the efficiency in the algorithm compilation platform.

The incorporation of a supplementary delay in the output can be considered imperceptible but adds strength to the final system; its value must be adjusted to an undertaking between what is noticeable and what is practical (about 4 ms works fine).

Finally, and to prevent the system of excessive input signal values during long time intervals, an input level compressor will be implemented; and normalized values for the convergence steps will be used.

All the necessary elements for this algorithm to result efficient and practical can be checked in the previous figure.

The values to adjust are the following:

- Gain control: Permitted maximum values; attack and release time; integration time.
- FIR LMS: Length, coefficients resolution.
- Coefficients update: Topology (in 16 units blocks); normalization; maximum step.
- Output buffer: maximum time for imperceptible, minimal so that it can result effective.
- Auxiliary calculations: power estimate variables, absolute values, etc.
- Pre-LMS FIR Filter: Frequency response; length.
- Pseudo - Frequency modulator: Buffer size; modulation speed.

And of course all those timing operations, input/output control, excessive levels displaying, reset control and effect bypass, etc.

We can also characterize the algorithm by its corresponding flow graph, so finally the algorithm remains perfectly fixed.

The particular manner of accomplishing the calculations will depend on the implementation platform. As an example an algorithm of this type on a DSP in real time execution can be studied in more detail in next paragraph. The A.L.M.I. algorithm flow chart can be followed in Figure 8.

2 - IMPLEMENTATION EXAMPLE.

Below it is exposed a real time A.L.M.I. practical implementation onto the DSP TMS 320C50 hardware [10]. The lists of the attached programs (annexes I & II) are prepared for its execution on the development platform called DSK (DSP Starter Kit), and with slight modifications the same can be used for a professional commercial device with a 48 kHz sampling frequency on DSP TMS320C51 or similar, with program in ROM.

For this implementation we use the physical support provided by the company Texas Instruments mod. DSK C50, that incorporates:

- Analog/digital and digital/analog converters (AIC); both with 14 bits resolution.
- TMS 320C50 (20 MHz) fixed-point Digital Signal Processor.
- Personal computer RS - 232 compatible programming port.

And the programs coded in assembler in its original format *.asm and/or data dump toward the DSP *.dsk

The execution takes place according to a classic program sequence in real time, through the use of interruptions, as figure 9 shows.

The code has been optimized for its application as an acoustic feedback canceller. It is tested that the improvements in the gain levels permitted before feedback occurs increase up to 26dB (with the input signal totally decorrelated, that's to say, white noise), values given experimentally.

The operations that take place are:

1.- Starting some values (filter coefficients, pointers to variables, etc.)
2.- Starting A/D and D/A converters control values. This example operates on a sampling frequency of 15.625 kHz. and 14 bits resolution (16/32 on processing).
3.- Starting jumps and service interruptions.
4.- The main program is inserted in the receiving interruption routine, that contains the A.L.M.I.

The hardware blocks graph will include a DSP (or microprocessor), A/D and D/A converters, the auxiliary circuits and some user controls. You can see all blocks in figure 10.

Achieving the follow-up of the algorithm, the DSP must carry out the following calculations:

- All the operations with the signal averages and estimates:
- *IN* as input signal.
- *LIM0* as $|IN|^2$. (Absolute of squared input).
- *LIM* as averaged temporary of *LIM0*, using a IIR structure:

$$LIM(i+1) = LIM(i) + LIM0(i)\cdot 10^{-6} - LIM(i)\cdot 10^{-6}$$

- *ABSX0* as absolute value of the E.R. signal.
- *ABSX* the average, as with *LIM* but with a time constant slightly smaller. Furthermore, with a resolution of 32 bits.
- *ABSX2* as the square of the previous value.
- *IABSX* as the inverse of ABSX (add to *OFFSET*).
- Increase the "*H*" block counter (values between 0 and 15).
- Increase the F.M. pointer counter (according to modulation index)
- Normalize the error signal to 16bits. (*Us*)

- Concerning the filter calculations and convolutions:
- Realize the Pre-LMS filtering through the 61 coefficients FIR.
- Calculate the feedback estimate as convolution between the LMS input with the coefficients.
- Calculate the coefficients update (*INCs*)
- Coeficients update.

- Other tasks are:
- Testing that there`s no buffer pointer overflow, and if it occurs proceed to its variation (whether it is to circulate, as the output delayer or two-way as that in F.M.)
- Attending the interruptions of the P.C. through RS - 232 port.
- Masking the output to 14 higher bits.

The coefficients calculation includes a consistent stabilization factor by subtracting to each coefficient its own weighted value.

As a rule, the calculations of the algorithm kernel are:

- Feedback Estimate:

$$er(i) = \sum_{k=0}^{N-1} a_k x(i-k)$$

through the convolution between the LMS coefficients with the error signal filtered in the Pre-LMS, delayed and pseudo-frequency modulated (*x(i)*).

- Temporary average estimate:

$$LIM(i) = (1-10^{-6})LIM(i) + LIM0(i)$$

for the gain control .(*LIM* is the absolute value to the squared averaged input signal *IN* during a time slot.

$$ABS(i) = (1-10^{-8})ABS(i) + ABS0(i) + 10^{-8} \cdot OFFSET$$

in order to obtain the *Us* normalization, and the squared estimation from the error signal towards gain control.

- Coefficients normalization:

$$(OUT \cdot IABSX) = U_K = \begin{cases} 0 < U_K > 7FFFh \Rightarrow 7FFFh \\ 0 > U_K < 8001h \Rightarrow 8001h \\ 8001h < U_K < 7FFFh \Rightarrow U_K \end{cases}$$

- Coefficients update:

Configured in blocks of 16 coefficients, 8 complete cycles are necessary

$$Uk(i) = \sum_{m=k}^{k+15} \frac{U(i-m)}{ABSX(i)} \cdot x(i-k-m)$$

to update 128 coefficients.

$$a_k(i+1) = a_k(i) \cdot (1-2^{-14}) + (2^4 \cdot IABSX(i) \cdot Uk(i)) \cdot 2^{-16}$$

the "leaky" factor $(1 - 2^{-14})$ makes the convergence slower, but stabilizes the coefficient value in case of high dispersions.

- A number of output data (*OUT*) are stored for the pseudo-frequency modulator and they are refreshed in each cycle, then reading them alternatively in a way or another using a speed that depends on the variable *IND*.

- The Pre-LMS filter is a FIR that also provides a delay of 31 sampling units to the final system output (*OUTTAP*) and to the input of the LMS FIR. The coefficients produce the following normalized response, showed in figure 11.

Finally, it is attached a map of follow-up variables for the set of the algorithm with its auxiliary elements, that corresponds to the "data received" interruption service routine (Rint), where the variables lists can be followed (by their names in the original), so you can follow them in figure 12. (The rest of the program is used for timming, initialization tasks, etc.)

With respect to hardware, and always seeking for some appropriate characteristics, below it is described the implementation of the A.L.M.I. in a digital processor of concrete signal together with the necessary elements for the operation on an electronic equipment that includes an input, an output and some practical additions. This appliance is susceptible of manufacturing and utilization for the intented aim; and the form and final characteristics will depend on secondary design criteria (form and size, type of connectors, etc.). On the other hand, due to the constant and rapid evolution of the semiconductors market, it is impossible to specify an unique set of integrated circuits as suitable, being many the alternatives.

In the example which is described as the prototype, the basic characteristics of the final device will be the following:

- Electronic device with analogic input and output.

- With A.L.M.I. real time execution applied to the input data.
- Uses a commercial low-cost fixed-point DSP of the brand Texas Instruments belonging to the family TMS 320C5x.
- The data are converted to digital from analogic and *vice versa* using a sampling frequency of 15.625 Hz, and a 14 bits resolution.
- The data processing is accomplished in fixed-point to 16/32 bits.
- It includes restart/reset control, a bypass control, and a discrete input gain selector, in addition to continuous level adjustment of the same by pot.
- Through luminous indicators (preferably LEDs) the operation state is visualized.
- It contains analogic amplifying stages, apart from the analogic filters that prevent the appearance of aliasing in the converters.
- It includes input and output balanced 6.3 mm stereo jack connectors.
- It includes power supply special connector used for unit feeding, +5, +12 and -12 volts referred to the ground (0 volts) or corresponding reference.

A decription of the device blocks is given in the following figure, and a possible physical designal where the indicated elements agree in both figures.

To manufacture this device, making allusion to the separate parts (algorithm and electronic circuit), the following elements will be needed:

For the physical support (hardware): the device will include the following elements:

- Input connectors (6) and (7), according to the specifications.
- Signal conditioner: compose of operational amplifiers as active elements; it includes the controls of discrete gain (8), it's labeled according to earnings for microphone or line) and continuous (3). adjustable pot, and the low-pass filters necessary. A practical implementation is accomplished with the integrated circuit TL 32088, with the necessary passive elements (resistors, capacitors, etc.).
- A/D and D/A converters, from integrated circuit TLC 32040 or TMS series (It has an advantage: it can be connected directly to the DSP, without additional logic (glueless); also Crystal, etc., or any which has a 2's complement serial bus.

- The digital signal processor model TMS 320C5x, that executes the A.L.M.I.
- A 27PC256FM ROM containing the program that implements the algorithm, that is read in the moment of the starting up of the circuit. The values dumped for the program are the lists in Annex II, using the hexadecimal data format in its own extension *.dsk.
- The regulators for the external/internal power supply.
- The power input (9) and output (10) connectors for chainning, permitting to huddle several of these devices being fed by a common source, adequately designed.
- The power "on" indicator (1) and " input level and/or excessive error" (clipping), through LEDs fed by a buffer.
- The restart/reset (5) and the effect bypass (4) buttons, the later with a built-on LED indicator.
- The physical support for all the electronic components (a double-sided customized designed printed circuit board).
- The physical support that provides mechanical hardiness and shellers to the system, using an adequately designed and mechanized box made of ABS plastic, aluminium or any other rigid material, and the rest of the auxiliary elements (wiring, etc.), all this according to the safety and electromagnetic interference applicable regulations.

The practical implementation has been built as a prototype, proving its behaviour with all the elements cited in this description, something which does not imply that in a subsequent designal phase and in a process of a massive manufacturing, some of the elements may be substituted by more adequate ones. As a preferred alternative, the construction of a model that uses a TMS Texas Instruments DSP is cited whose commercial reference is TMS 320C51, and since it incorporates a factory recorded memory (mask ROM), it permits to dispense of an external ROM, cheapening the manufacturing costs, pieces and ROM recording. Also, a commercial device will possess 16 bits resolution and converters will work at 44.1/48 kHz sampling frequency, since TMS320C51 can running at 40 MHz. You can see a proposed final termination and a block diagram in figures 13,14, with the references matched.

3.- SUGGESTED CHECKING.

Below it is exposed the suggested procedure so that you can prove the algorithm characteristics, possibilities and features. Firstly we will need a development starter kit (DSK) from manufacturer Texas Instruments including a fixed-point 20 MHz DSP TMS320C50, an A/D - D/A converter and RS 232 programming port. This DSK has a very low price (more information on web: http://www.ti.com/sc/docs/tools/dsp/c5xdevelopmentboards.html).

For the accomplishment of the tests we will need:

- A microphone.
- An audio amplifier.
- A loudspeaker.

And, optionally:

- A preamplifier or best a mixer.
- A noise/tones audio signal generator.
- Phonographic materials for the test with voice signals, music, etc.

As regards the procedure, the equipment will be wired as follows:

- The microphone or mixer output will be connected to the DSK audio input.
- The amplifier and loudspeaker will be connected to the DSK audio output.
- A serial RS 232 PC port will be connected to the DSK programming serial port.
- The DSK monitor will be loaded on PC (program supplied with DSK, dsk5d.exe).

Once the system is running, the algorithm implementation program supplied by the author (file: `almi.dsk`, Annex II) will be loaded onto DSK (please remember to set CNF bit properly, because of the use of memory bank 1, as follows: `MODIFY/REGISTERS:CNF=0`), and then it will execute the A.L.M.I., permitting to check the improvement on the system without the algorithm (to implement the bypass button the hardware must be modified, but you can load the pass program), executing the program that it does not make nothing with the signal, but provides the output from the input in equal conditions of sampling frequency, levels, etc., as the algorithm.

Also modifications to the algorithm parameters can be accomplished, so the file `almi.asm` will be edited on an ASCII editor and assembled and linked using dsk5a.exe (supplied with DSK), using the Annex I (program list).

4.- EVALUATION AND CONCLUSIONS.

Below the results obtained are exposed with the prototype that executes the algorithm "A.L.M.I.". Since it is very difficult to quantify levels and other magnitude implicate in the measurement of such dynamical device, some graphics are attached that intend to illustrate the operation. In any case, more information can be requested (the author can provide audio recordings with the algorithm in execution), though it is recommended that you submit it to objective tests yourselves. A comparative summary of these five graphics as well as a short commentary, is attached to each one:

Figure 15.- White noise floor at - 36 dB with an acoustic coupling.

This graph is obtained by introducing white noise as the source of signal whose level is fixed to - 36 dB from the maximum (0 dB=clipping or saturation in the converters), where, in 2 seconds counting from the start, a feedback is forced to occur with a high gain preamplified microphone. It is observed how the exponential growth of this couple quickly provokes the dinamic range saturation. Of course, the device is found in bypass, that't to say, the algorithm is **not** executed.

Figure 16.- Same conditions as in Fig.15, with A.L.M.I. set ON and +26 dB gain.

In this occasion we proceed as in the previous case (feeding with white noise), but rising the input level (both noise and microphone) in 26 dB with respect to previous level - 36 dB, therefore a narrow margin of 10 dB remains to locate howling/coupling growths. After 2 seconds from the start, the microphone is coupled, with a soft gain transition through those two seconds.

The algorithm decorrelates the couplings and adapts the equalization feedback path so that coupling DOES NOT APPEAR, and in 26 dB more gain than in the previous case, and all this in real time and without sharp or irregular variations.

You can also see that the algorithm does not return a pronnunciated or irregular frequency response, it rather observes the input signal flatness.

Figure 17.- 1 kHz pure tone with an acoustic coupling.

Now we change the test signal, using a 1 kHz pure tone and forcing an acoustic coupling near 500 ms from the start. Again the acoustic coupling appears, you can observe furthermore its harmonic and the trend to the saturation which is produced in an immediate way. The pure tone level was fixed to - 28 dB, and the algorithm is **not** executed.

Figure 18.- Same conditions as in Fig. 17, with A.L.M.I. set ON and +20 dB gain.

Now we proceed as in the previous case, with an input (1 kHz tone and microphone) adjustment set to +20 dB referred to the previous test. Near to 500 ms we force an acoustic coupling, then the algorithm tries to eliminate it.

You can see how the coupling frequencies (with waviness) try to be minimized, but the presence of the pure tone as useful input signal does not allow its total elimination. This effect, that sounds as a slight "vibrato" on the coupling, tends to be stabilized through time.

You must not miss the extreme adjustments which this test has been accomplished with, and that the saturation is never reached thanks to the algorithm.

Figure 19.- The A.L.M.I. fading away acoustic howlings.

This last graph is obtained submitting the device (of course, running the A.L.M.I.) to a pink noise input and a microphone input mixed with high gain (until 30 dB over the safety margin), and furthermore, varying the position through time (therefore frequency, phase and gain response) of the microphone with respect to the loudspeaker.

The algorithm avoids saturations, so that never reaches to the clip point, being presented the howlings caused by the acoustic feedback, but more attenuated and with a tendency to disappear, warning a hypothetical sound technician that the stable margin has been surpassed.

According to the previous graphics, and many other considerations that can not be presented in a graphic form, we can summarize the following conclusions, merely from a technical point of view:

ADVANTAGES:

- The algorithm results useful to avoid acoustic feedback and howling.
- The obtained margin widely go beyond the levels that auto/manual-equalizing systems can reach (Up to +26 dB guaranteed).
- The electronic equipment clipping or saturation by acoustic coupling is not produced.

DRAWBACKS:

- Frequency modulation on the signal is produced (though imperceptible, less than 0.0096 %).
- When the signal source is a very interrelated sound (synthetized pure tones, wood-wind sounds with great loads of harmonics, etc) the algorithm suspects of them, and attempts to cancel them (though for this to occur they should be maintained at high levels during a relatively long time interval and on the same frequency), and it seems that the input sounds were pitching.

As a conclusion we should be awake of the improvement that would suppose a device as the proposed, not as an expensive, professional range system with displays, complex adjustments, etc.; but as a small correcting system with an input, an output and a bypass button, or even, why not, built-in in wireless microphones, or in conference systems, S/P DIF outputs... directed to the consumer / semi-professional market.

The use of the program files (congenially registered and patented) is subordinated to the only execution for evaluation, test and educational purposes; as far as it is concerned, therefore if you are decided to execute them, you will have to request the corresponding permission to the author, that gladly will provide free of any charge (E-MAIL contact: m.almi@coitt.es)

5.- REFERENCES.

[1] Madisetty, V.K., and Williams, D.B., (editors) "The digital signal processing handbook," *CRC Press/ IEEE Press, , 1998;chap.18-20.*

[2] Proakis, J.G., and Manolakis, D.G., "Tratamiento digital de señales". *Prentice Hall Inc, 1997;chap. 8.*

[3] Gitlin, R. D., J. E. Mazo, and M. G. Taylor, "On the Design of Gradient Algorithms for Digitally Implemented Adaptive Filters," *IEEE Trans. Circuit Theory, Vol. CT-20, March 1983. pp. 125–136.*

[4] Messerschmitt, D. et.al.,"Digital Voice Echo Canceller with a TMS32020", from "Digital Signal Processing Applications with the TMS320 Family, Vol 1"; *Kun-Shan Lin (editor); Prentice Hall Inc, 1987.p. 415-454.*

[5] Kevin McCoy and Mansoor A. Chishtie, "Digital Voice Echo Canceler Implementation on the TMS320C5x (Application Report),*DNA Enterprises/Digital Signal Processing Applications — Semiconductor Group, T.I.pub. SPRA142 October 1994.*

[6] Sen M. Kuo, Ph.D. Issa Panahi, Ph.D. Kai M. Chung Tom Horner Mark Nadeski Jason Chyan, "Design of Active Noise Control Systems With the TMS320 Family", *Digital Signal Processing Products—Semiconductor Group,T.I.pub. SPRA042 June 1996.*

[7] Oppenheim, A. V., and R. W. Schafer, "Discrete-Time Signal Processing", *Prentice-Hall, Englewood Cliffs, NJ. 1989.*

[8] David Smalley "Equalization Concepts: A Tutorial.(Application Report)"; *Atlanta Regional Technology Center, T.I. pub. SPRA140 October 1994.*

[9] Tretter, Steven A., "Communication System Design Using DSP Algorithms"; *Plenum Press, 1995.; chap.3, 8, 15.*

[10] Texas Instruments "TMS320C5x User's Guide",*1997.*

ANNEX I: A.L.M.I. REAL-TIME IMPLEMENTATION PROGRAM LIST (TMS320C5x).

```
; Programa para amortiguación de realimentación acústica.
; Utilizando algoritmo A.L.M.I. para DSP TMS32050.
; última revisión: 16/2/99. (SET CNF bit=0) ©-®
;------------------------------------------------------------
        .mmregs    ; utiliza Registros Mapeados en Memoria
;------------------------------------------------------------
;       VARIABLES: etiquetas y direcciones.
        .DS 0120h; Buffer para LMS.APUNTA A X,
X       .word 0      ; Siguiente (por el DMOV)
X+1     .word 0
        .DS 01AEh
X14     .word 0      ; Añadidos 128+14
;-----------------------------------------------
        .DS 0300h  ; Buffer para FIR previo a LMS
XPRE        .word 0      ; Dirección 0=Xpre
;------------------------------------------------------------
        .DS 03A0h  ; entrada FMIN;
FMIN .word 0      ; Hasta 04FFh disponible.
;------------------------------------------------------------
;       COEFICIENTES PARA FILTRO FIR-LMS
        .ds 01000h
        .ps 01000h  ; total: 128
COEF        .word 0      ; Hata 256 con esta Fs.
        .ds 0107Fh
        .ps 0107Fh
ULCO        .word 0      ; Último coeficiente
;-----------------------------------------------
        .ds 0D00h
        .PS 0D00h
U0      .word 0      ; Salidas normalizadas.
U0+1 .word 0,0,0,0,0,0,0,0,0,0,0,0,0
U0+14 .word 0,0  ; Total 16, (8 bloques)
;-----------------------------------------------
        .ps 0E00h
        .ds 0E00h ; Incrementos para coeficientes
INCO .word 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
;------------------------------------------------------------
        .PS 0F00h; Coeficientes de FIR preLMS
PRELMS
  .word -1,1,-3,3,-6,2,-5,-3,-1,-14,3,-19,-6,-1,-49
  .word 66,-156,203,-352,419,-640,708,-986,1038,-1328
  .word 1309,-1602,1135,-1772,-5402,16002,-5402,-1772
  .word 1135,-1602,1309,-1328,1038,-986,708,-640,419
  .word -352,203,-156,66,-49,-1,-6,-19,3,-14,-1,-3,-5
  .word 2,-6,3,-3,1,-1
;------------------------------------------------------------
; Variables y constantes EN PÁGINA 02 DE DATOS
;------------------------------------------------------------
        .ds 0100h  ; Variables para LMS
IND     .word 1      ; índ. mod pseudo-FM
IN      .word 0      ; entrada de DRR
ABSX0 .word 0     ; Absoluto de OUT/absout.
ABSX.word 20      ; Promediado de ABSX0.
ABSXL .word 0     ; Bits menos significativos de ABSX
IABSX       .word 800  ; inverso de ABSOUT.
OUT  .word 0      ; entrada menos ER.
CU0  .word 0      ; Copia de U0 en página 02 datos.
OUTTAP .word 0 ; Salida con delay.
ER   .word 0      ; Estimación de realimentación.
OUTFM .word 0     ; salida del FM.
LIM0 .word 20     ; absoluto cuadrado de IN
LIM  .word 20     ; LIM0 promediado.
ABSX2 .word 0     ; Abs(OUT)**2
;------------------------------------------------------------
REPEFIR  .word 127  ; Para FIR LMS.
CERO            .word 0      ; Pues eso
UNO         .word 1      ; Un uno.
UNOSAT  .word 32767       ; Saturado positivo.
CONT            .word 0      ; Contador para buffer FM.
AR7MAX+1        .word 988    ;
AR7MAX      .word 987    ; Máximo para AR7.
AR7MIN      .word 930    ; Mínimo para AR7
UP .word 35759     ; CONSTANTE cód. arriba(MAR*+,AR7)
DOWN .word 35743 ; CONSTANTE cód. abajo (MAR *-,AR7)
DIR   .word 35759        ; VARIABLE de MAR
H       .word 0       ; Contador de bloque
GAIN .word 10     ; Ganancia de entrada INICIAL.
TEMP        .word 0       ; Temporal de direcciones
OFFSET .word 31           ; Para ABSX MIN
MASC        .word 65532 ; Máscara de salida (D/A de 14 bits)
;-----------------------
; PUNTERO A DELAY
        .ps 01100h
BC      .word 0       ; puntero a buffer circular.
;*************************************************************
;  Configura vectores de interupción                       *
;*************************************************************
        .ps     080ah    ;
rint:  B      VOY       ;0A; Interrupcción RINT.
xint:  B      VENGO  ;0C; Interrupción XINT.
;*************************************************************
; Inicialización del TMS32C050                              *
; Programa principal en 0a00h                               *
;*************************************************************
        .ps     0A00h
        .entry
        SETC   INTM     ; Desconecta interrupciones.
        LDP     #0              ; Apunta a la página...0.
        OPL    #08BEh,PMST  ; OR con el registro de estado.
        LACC   #0
        SAMM   CWSR         ; Estados de espera soft: 0
        SAMM   PDWSR        ;
        CLRC        CNF  ; Banco B0 en Data 0100-02FF.
        SETC   SXM              ; Modo de signo bit 15.
        SPLK   #022h,IMR    ; Usa XINT para sinc TX y RX.
        CALL   AICINI         ; Inicializa AIC.
        SPLK   #12h,IMR  ; Permite interrupciones de servicio.
        SETC   OVM          ; Modo acarreo para operaciones.
        CLRC  XF            ; Borra aviso clipping
        SPM 0               ; Preg Modo 0
;------------------------------------------------------------
;       Inicializo el buffer circular BIDI:
;------------------------------------------------------------
        LAR  AR7,#932   ; Inicio a buffer bidireccional FM.
        LAR  AR4,#0h    ; Inicio de contador para auto limiter.
        LACC        #128 ; Veloc. de RELEASE (Pot. de 2).
        SAMM        INDX ; + más rápido, - más lento.
;------------------------------------------------------------
        LAR  AR0,#COEF       ; =1000h
; BORRA EN 1000h-1200h:
        MAR  *,AR0; Usa AR0 para borrados.
        ZAP
        RPT  #01FFh      ; Borrado de 128 posiciones.
         SACL       *+
; BORRA COEFS, ETC
        LAR  AR0,#X      ; =120h EN data
; BORRA EN 120h-4FFh:
        RPT  #03DFh      ; Borrado de 128 posiciones.
         SACL       *+   ; Aún hay 0 en ACC
;------------------------------------------------------------
; BUFFER CIRCULAR
        LACC        #01100h   ; Para pre-delay.
        SAMM CBSR1    ; Inicio
        LACC        #0117Fh    ; 127--63
        SAMM  CBER1   ; Fin
        LACL #8; bit 3: b.c.1 ON, bits 2-0: AR0 = 00001000.
        SAMM        CBCR       ; Activación
        LAR  AR0,#BC    ; =1100h (restituye).
        LACC        #0B0Dh     ; Fijo, dirección para MAR+-
        SAMM        BMAR
```

```
;--------------------------------------------------------------------
      LAR   AR1,#FMIN ; =0390h Entrada buffer bidi.
      LAR   AR5,#XPRE ; =0300h Entrada para pre-FIR.
;--------------------------------------------------------------------
      LDP   #IN   ; Todo ocurrirá en página 0100H.
      CLRC    INTM          ; Espera interrupciones.............
;--------------------------------------------------------------------
espera:
      IDLE
      NOP
      B     espera ; Y atiende ISRs del AIC (o servicio).
;------------------------- ISR de recepción-----------------------
VOY:
      SETC      INTM; DESACTIVO INTERRUPCIONES.
;--------------------------------------------------------------------
      LDP   #IN   ; Por si página
;--------------------------------------------------------------
      LAMM      DRR  ; la entrada del A/D en IN
      SACLIN
;-----------------------------actualiza H (para bloques de coef.)
      LACC      H
      ADD  UNO
      AND  #000Fh
      SACLH     ; H=0...15
;-----------------------------cuadrado de abs(in)
      LACC      IN
      ABS
      SACLLIM0
      ZAP
      SQRA      LIM0
      PAC
      SACH      LIM0,1      ; En LIM0 (Abs(IN))**2
;--------------------------------------------------------------------
; AGC
      LACL GAIN ; GAIN en TREG1
      SAMM      TREG1
      MAR  *,AR0; No es muy intuituivo, pero pipeline..
      LACTIN    ; Según Treg1, o sea GAIN
      LMMR      TREG1,#CERO ; ponemos a cero.
      LAR  AR3,#X14; AR3 apunta a última muestra +14.
      SACH   IN  ; Ahora IN está ponderado según GAIN.
;--------------------------------------------------------------------
;     predelay:
      LACC      OUTFM     ; de este modo ahorramos.
      SACL*+    ; escribo, aumento.
      LACC      *,0,AR3; y leo antes de borrar en la
siguiente.
      SACLOUTTAP ;  es IN (-t); t=según buffer circular.
;----------------------
; ESTIMACIÓN DE REALIMENTACIÓN
;-------------------------------------ACTUALIZA 16 ULTIMOS
      RPT  #14
      DMOV      *-      ; X(K) -> X(K+1)
;----------------------- Movemos las 15 últimas muestras
;FIR (convolución Coeffs con muestras).
      ZPR           ; Aún apuntamos a X127
      LACC      UNO,14    ; Redondeo
      RPT  REPEFIR   ; K = n,n-1....0
       MACD COEF,*-     ; X(K)*A(I-K)+ACC ->ACC
      APAC                ;
      SACH      ER,1  ; ER*2 (por el signo)
      MAR  *+,AR4      ; A3 Apuntando a X0; ACT AR4
;---------DIFERENCIA ERROR
      LACC      IN    ; (IN)
      SUB  ER   ; Acc=in(t)-ER(t)
      SACLOUT  ; Guarda en OUT
      ABS
      SACLABSX0      ; y a ABSX0 para ABSX e IABSX.
;--------------------------------------------------------------------
;cálculo potencias:
      LACC      ABSX,16    ;
      ADDS      ABSXL
      SUB  ABSX,8      ; cte tiempo =9-7 ok
      ADD  ABSX0,8     ; El de antes, claro.
      ADD  OFFSET,8  ; Un mínimo.
      SACH  ABSX       ; Guarda ACC alto
      SACLABSXL        ; y bajo.
;-----------------------------------------------
      LACC LIM,16       ; LIM sólo a 16 bits.
      SUB  LIM,10       ; LIM más rápido , =10
      ADD  LIM0,10      ;
      SACH LIM
;-----------------------------------------
;     Calcula 1/ABSX
      LACC    UNO,16
      RPT  #14   ; 14+1, para máx 7fff con absx=0
       SUBC      ABSX; 1/ABSX con SUBCondicional
      SACL        IABSX,0   ; para aprovechar el rango.
;------------------------------------------------------------
      ZAP                     ; ABSX al cuadrado
      SQRA        ABSX          ; en ABSX2
      PAC                     ; ABSX2=(ABSX)^2
      SACH        ABSX2,1
;-------------------------------------
; RUTINA DE CLIPPING POR DIST.
;
      LACC        #128; Velocidad de RELEASE (Pot. de 2)
      SAMM        INDX; + más rápido, - más lento. 128 OK.
      LACC        ABSX2,1    ; Mejor con promedio,ABSX2.
      ADD  LIM  ; Comprueba los dos: LIM y ER.
      SUB  UNO,15      ; Resta umbral
      LACL GAIN ; Si es mayor: Nivel excesivo
      XC    2,GT
       SFR          ; Atenúa 6 dB
       SETC  XF ; Y avisa al mundo exterior
;-----------------------
      BANZD  SALTA,*0-,AR1; VUELVE (del) si AR4 no 0
      SACLGAIN
      LAR  AR2,AR7MAX+1; Apunta a fin de buffer FM.
;-----------------------------------------------------
      SUB  #0Eh  ; el ACCL aún contiene GAIN
      LACL GAIN ; Y no es máximo, entonces
      XC    2,LEQ       ; prueba a subirla...
       ADD       #1
       CLRC  XF ; Apaga pilotillo
      SACLGAIN ; y sube GAIN +6dB (en su caso)
;------------------------------------------------------------
SALTA:
;-----------------------------
      LACC        OUT         ; ACC<- OUT
;------------------------------------------------------------
; Mando OUT al modulador FM (apuntado por AR1=03A0h)
;------------------------------------------------------------
      SACL*,0,AR2      ; La salida del FIR LMS - Xn a FMIN
;------------------------------------------------------------
      RPT  #60   ; Repite REPFM + 1 veces
      DMOV      *-    ; Actualiza las muestras.
;--------------------------------
      LACC        #0B0Dh     ; Fijo, dirección para MAR+-
      SAMM        BMAR              ;
;------------------------------
      MAR  *,AR7; Utiliza AR7
      LACL CONT        ; carga contador en ACC bajo.
      ADD  IND,0; incrementa en IND
      SACLCONT         ; y vuelve a guardar (actualizado)
      AND  #01,15      ; comprobación de contador
      BCND        0B00h,NEQ   ; Si no es cero, salta
                  ; a rutina de actualización de AR7
;------------------------------------------------------------
vuelve: LACC *,0,AR5  ; carga ACC con apuntado por AR7.
      SACL        OUTFM; Ésta es la salida f-modulada
;------------------------------------------------------------
      SACL*       ; Guarda en AR5, para pre-FIR
;------------------------------------------------------------
      ADRK        #60   ; AR5 a última muestra (AR5)+60
      ZPR
      LACC        UNO,14
      RPT  #60   ; Repite 60+1 veces
```

```
     MACD  PRELMS,*-    ; Filtro FIR Pre LMS.
     APAC               ; Último resultado
     MAR *+,AR3         ; AR5 a 0300h (preparado)
     SACH      *,1,AR3; HACIA BUFF PARA LMS (0120h)
;----------------------------------------------------------------
;    Actualiza Taps
     LAR  AR3,#U0+14  ; Carga AR3 a U viejo.
     RPT  #13   ; K= 14,13....1
      DMOV   *-     ; U(K) -> U(K+1)
     DMOV     *   ; U(0) -> U(1) mueve sin decremento.
;----------------------------------------------------------------
     LT   OUT         ; Normalización de U's
     MPY  IABSX             ; Primero, OUT/ABSX
     PAC
     BCND      POS,GEQ  ; Si es positivo, salta
;NEGATIVO:
      ADD  UNOSAT  ; Si no, comprueba que es < 8000h
     BCND      NOSAT,GEQ
;SATURADO:
     BD   FIN   ; y si no, 8001h
      LACL     #0
      SUBUNOSAT
POS:
     SUB  UNOSAT   ; si es positivo
     BCND      NOSAT,LEQ; que no sea mayor que 7FFFh
     BD   FIN
      LACC     UNOSAT   ; Y si no, 7FFFh
      NOP
NOSAT:
     PAC
FIN:
     SACL*,0,AR3     ; Guarda 8001h < U0 < 7FFFh
     SACLCU0; Y haz una copia en CU0. (Para DP)
;----------------------------------------------------------------
;incs:
     LACC      #X+1 ; Apunta a la primera muestra+1
     ADD H      ; Le toca al bloque H...
     SACLTEMP
     LAR  AR3,TEMP ; AR3-> (X+1+H)
     LACL #7         ; N=0...8
     SAMM      BRCR
     LT   CU0  ; U0 en TREG0
     LAR  AR2,#INC0 ; AR2 apunta a los INC's.
     RPTB      BUCLE
      LACC UNO,10  ; ponderación
      MPY      *+    ; AR3->(muestras)
      RPT#14          ;
       MAC     U0+1,*+    ; AR3-> (muestras)
      MAR      *,AR2; AR2-> (INC0+N-1)
      LTA CU0
:BUCLE SACH *+,1,AR3      ; AR2-> (INC+N) (1)
;----------------------------------------------------------------
;ACTUALIZA taps:
     LACL #16  ; Bloques de 16 coeficientes
     SAMM      INDX
     LAR  AR3,#INC0 ; AR3 a INC's
     LACC      #ULC0          ;
     SUB  H       ;
     SACLTEMP     ;TEMP3 = dirección de A(H)
     LACL #6   ; (último coef-H)
     SAMM      BRCR      ; Repite 6+1 veces
     LACC      IABSX     ; Aquí se completa (1/ABSX)
     SAMM      TREG0
     LAR  AR2,TEMP
     MPY  *+,AR2
     SPM  2
     RPTB      BUCLE2
      LACC     *,16,AR3; LEE EL COEF PREVIO
      MPYA     *+,AR2
      SUB*,2,AR2    ; LEAKY (previene saturaciones)
:BUCLE2 SACH *0-,0,AR2
     LACC      *,16,AR2
     APAC
     SUB  *,2,AR2    ; ÚLTIMO LEAKY
```

```
     SACH  *0-,0,AR3 ; direccionamiento practiquísimo!
     SPM  0     ; NO shift de PREG.
;************************SALIDA
     LACC      OUTTAP   ; CARGA SALIDA
     XC   1,BIO   ; Pulsado bypass?( bp=XC 0,BIO)
      LACC     IN     ; Entonces, salida=entrada
     AND  MASC  ; DAC es de 14 bits, máscara FFFCh
     SAMM      DXR  ; Y YA LO TENGO EN LA SALIDA.
;***************************************
     RETE
;-----------------Rutina de recepción, no hace nada.----------------
VENGO:
     RETE              ; Vuelve,Y TXR.
;----------------------------------------------------------------
;    RUTINA PARA AR7
     .ps 0B00h
;    Comprobación AR7
;----------------------------------------------------------------
     BLDPDIR  ; Actualiza programa MAR
     LAMM      AR7  ; Carga ARMax en reg. Comparación
     SUB  AR7MAX   ; Resta valor máximo
     LACL DOWN     ; Con pipeline
     XC   1,GEQ    ; Ejecuta si AR7=> AR7MAX
      SACL  DIR
;---------------------------
     LAMM      AR7  ; Carga ARMin en reg. Comparación
     SUB  AR7MIN   ; Resta mínimo
     LACL UP   ; Tiempo de pipeline
     XC   1,LEQ    ; Ejecuta si AR7<= AR7MAX
      SACL     DIR
     BD   vuelve
;---------------------------
     MAR  *+,AR7; Instrucción modificada (+ -)CON AR4
;------------------------- ; Como en ACC H cargué ceros,
     SACH      CONT,0    ; borro contador, para próximo.
;----------------------------------------------------------------
     .ps 0C00h  ; Nueva ubicación para AIC
*****************************************************************
*     Inicialización del TLC32040 *
AICINI:
     SPLK      #01h,PRD
     SPLK      #20h,TCR ; pulsos a 10 Mhz
     MAR  *,AR0; Con AR0
     LACC      #0008h     ;
     SACL      SPC  ;
     LACC      #00c8h     ;
     SACL      SPC
     LACC      #080h      ;
     SACH      DXR
     SACL      GREG
     LAR  AR0,#0FFFFh
     RPT  #10000     ; Aguanta un rato...
     LACC      *,0,AR0     ; (.5ms a 50ns)
     SACH      GREG
     SETC      SXM  ;
     LACC      #8,9  ; TA, RA
     ADD  #8,2  ;
     CALL      AIC_2ND   ;
;----------------------
     LACC      #40,9       ; Inicializa registros TB y RB
     ADD  #40,2               ;
     ADD  #02h  ; Fs=(10e6/(2*40*8))=15.625 kHz
     CALL      AIC_2ND ;
;----------------------
     LACC      #40,2       ; Inicializa registro de control
     ADD  #03h  ; Max. gain
     CALL      AIC_2ND   ; No filtro
     RET              ;
AIC_2ND:
     LDP  #0
     SACH      DXR  ; Rutina de envío al AIC
     CLRC      INTM
     IDLE      ; Espera...
     ADD  #6h,15      ;
```

```
SACH    DXR ;
IDLE
SACL    DXR ;
IDLE
LACL    #0  ;
SACLDXR ; asegurar transmisión es importante....

IDLE
SETC    INTM
RET     ;
;-----------------------------------------------------
.end    ; y fin.
```

ANNEX II: HEXADECIMAL DATA MEMORY DUMP (TMS320C5x).


```
K_DSK5A_1.03_ALMI.DSK
10A0070A00F
90120M0000M000070000F
901AEM000070000F
90300M000070000F
903A0M000070000F
91000B000070000F
9107FB000070000F
90D00B0000B0000B0000B0000B0000B0000B0000B000070000F
90D08B0000B0000B0000B0000B0000B0000B0000B000070000F
90E0070000F
90E00M0000M0000M0000M0000M0000M0000M0000M000070000F
90E08M0000M0000M0000M0000M0000M0000M0000M000070000F
90F0070000F
90F00BFFFFB0001BFFFDB0003BFFFAB0002BFFFBBFFFD7FFF4F
90F08BFFFFBFFF2B0003BFFEDBFFFABFFFFBFFCFB00427FFEBF
90F10BFF64B00CBBFEA0B01A3BFD80B02C4BFC26B040E700EAF
90F18BFAD0B051DBF9BEB046FBF914BEAE6B3E82BEAE670B7CF
90F20BF914B046FBF9BEB051DBFAD0B040EBFC26B02C47FA26F
90F28BFD80B01A3BFEA0B00CBBFF64B0042BFFCFBFFFF7FE02F
90F30BFFFABFFEDB0003BFFF2BFFFFBFFFDBFFFBB00027FFD5F
90F38BFFFAB0003BFFFDB0001BFFFF7FFFAF
90100M0001M0000M0000M0014M0000M0320M0000M000070335F
90108M0000M0000M0000M0014M0014M0000M007FM0000700A7F
90110M0001M7FFFM0000M03DCM03DBM03A2M8BAFM8B9F7A2A7F
90118M8BAFM0000M000AM0000M001FMFFFC78BD4F
91100B000070000F
9080AB7980B0A3CB7980B0AF970835F
90A00BBE41BBC00B5D07B08BEBBF80B0000B882AB88287AFD8F
90A08BBE44BBE47BAE04B0022B7A80B0C00BAE04B001275F47F
90A10BBE43BBE4CBBF00BBF0FB03A4BB400BBF80B008077242F
90A18B8818BBF08B1000B8B88BBE59BBEC4B01FFB90A07F264F
90A20BBF08B0120BBEC4B03DFB90A0BBF80B1100B881A76C05F
90A28BBF80B117FB881BBB908B881EBBF08B1100BBF80729C8F
90A30B0B0DB881FBBF09B03A0BBF0DB0300BBC02BBE4079224F
90A38BBE22B8B00B7980B0A38BBE41BBC02B0820B90017DF3EF
90A40B1019B2010BBFB0B000FB9019B1001BBE00B900B7DE0DF
90A48BBE59B520BBBE03B990BB691AB880DB8B88B6B0174F22F
90A50B890DB010FBBF0BB01AEB9801B100AB90A0B108B7940BF
90A58B9008BBB0EB7790BBE58B1E10B0B0EBA390B100075DACF
90A60BBE04B9909B8BACB1001B3009B9006BBE00B9002700CBF
90A68B6A03B6204B3803B2802B281CB9803B9004B6A0C7E63BF
90A70B3A0CB2A0BB980CB6A10BBB0EB0A03B9005BBE59779A2F
90A78B5203BBE03B990DBBF80B0080B8818B110DB200C72244F
90A80B3F10B691ABF704BBE0ABBE4DB7FD9B0A8FB901A73607F
90A88B0213BBA0EB691ABF7CCBB801BBE4CB901AB100673374F
90A90B908ABBB3CB7790BBF80B0B0DB881FB8B8FB691270AA3F
90A98B2000B9012BBFBFB0001BE308B0B00B108DB900A7FE71F
90AA0B9080B783CBBE58B1E10BBB3CBA390B0F00BBE04710F4F
90AA8B8BABB998BBBF0BB0D0EBBB0DB7790B7780B730670E72F
90AB0B5405BBE03BE38CB0ABBB2011BE38CB0AC2B7D8078C2EF
90AB8B0AC3BB900B3011B3011BE3CCB0AC2B7D80B0AC379AB6F
90AC0B1011B8B00BBE03B908BB9007BBF80B0121B201975A60F
90AC8B901BB031BBB907B8809B7307BBF0AB0E00BBEC67D31DF        <EOF>
```

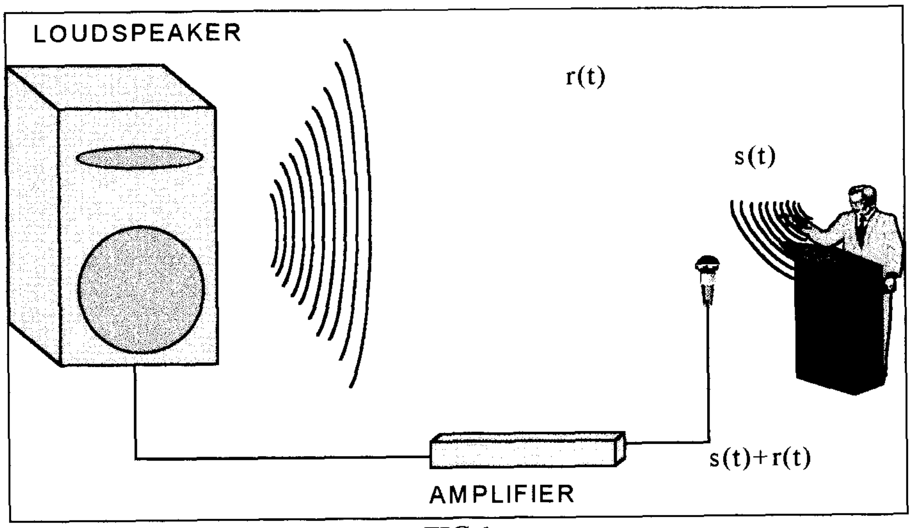


FIG.1

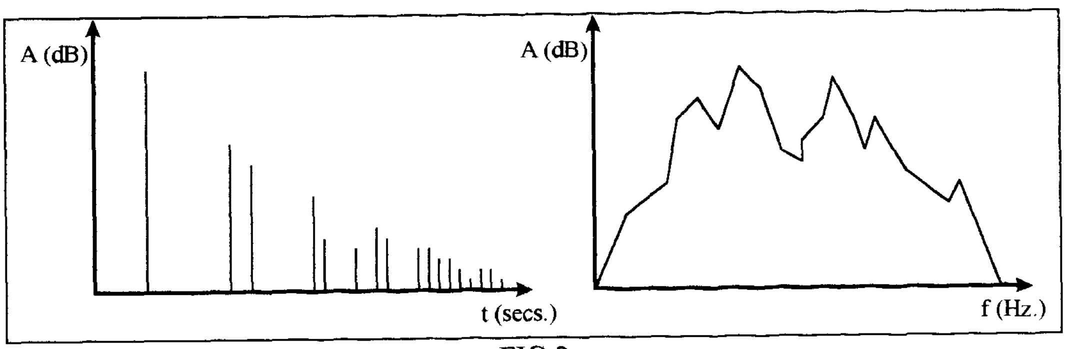


FIG.2

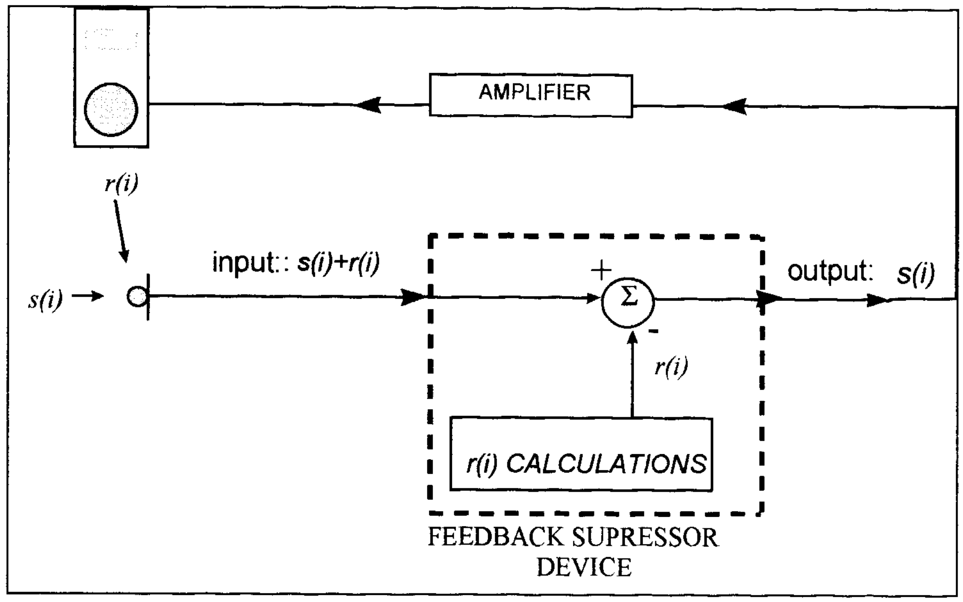

AMPLIFIER
r(i)
s(i)
input:: s(i)+r(i)
+
Σ
-
r(i)
output: s(i)
r(i) CALCULATIONS
FEEDBACK SUPRESSOR
DEVICE

FIG.3

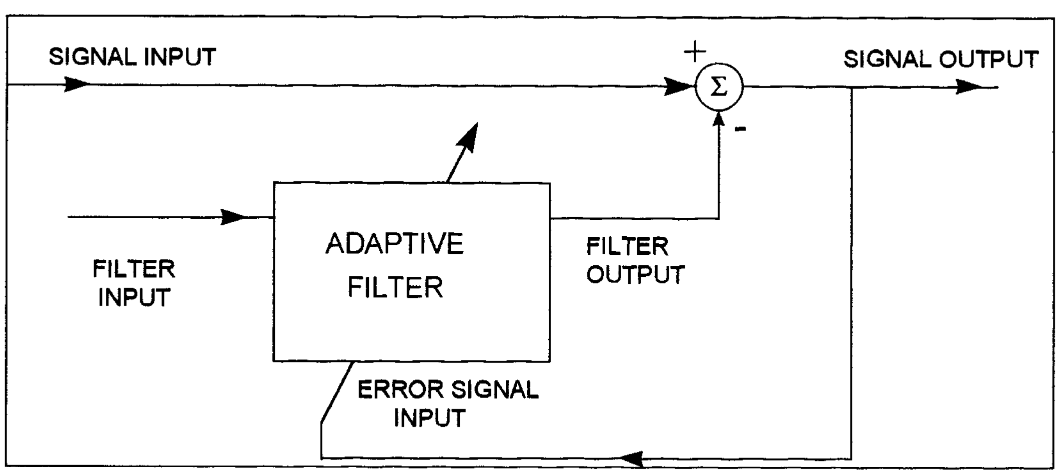

SIGNAL INPUT
+
Σ
-
SIGNAL OUTPUT
ADAPTIVE
FILTER
FILTER
INPUT
FILTER
OUTPUT
ERROR SIGNAL
INPUT

FIG.4

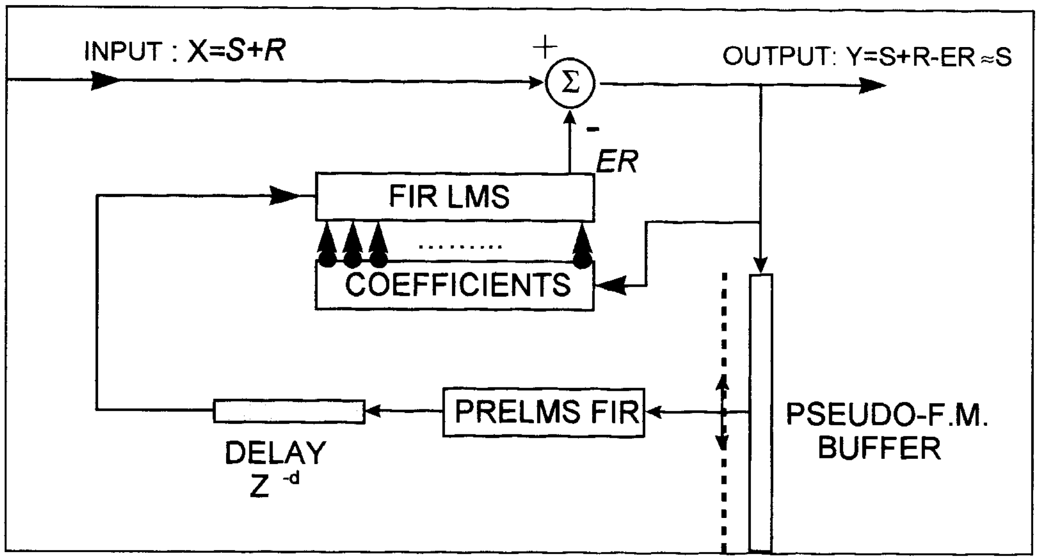

INPUT : X=S+R
+
Σ
-
OUTPUT: Y=S+R-ER ≈S
ER
FIR LMS
....
COEFFICIENTS
PRELMS FIR
PSEUDO-F.M.
BUFFER
DELAY
Z -d

FIG.5

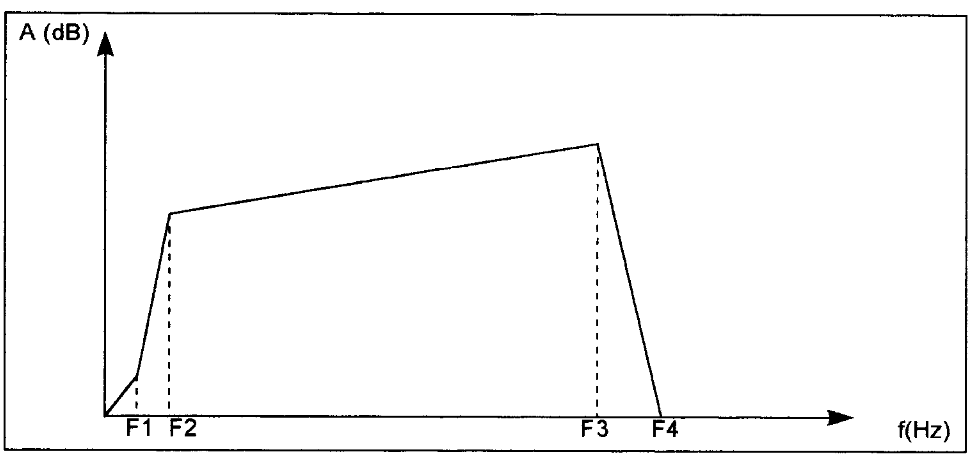

A (dB)
F1 F2
F3 F4
f(Hz)

FIG.6

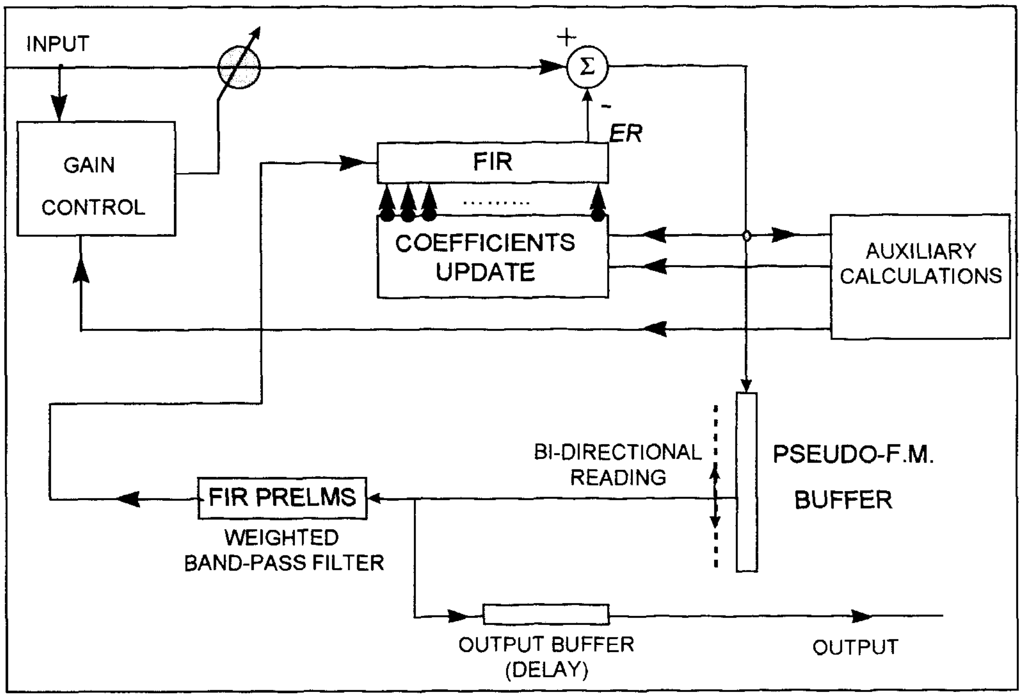

INPUT
+
Σ
-
ER
GAIN
CONTROL
FIR
COEFFICIENTS
UPDATE
AUXILIARY
CALCULATIONS
BI-DIRECTIONAL
READING
PSEUDO-F.M.
BUFFER
FIR PRELMS
WEIGHTED
BAND-PASS FILTER
OUTPUT BUFFER
(DELAY)
OUTPUT

FIG.7

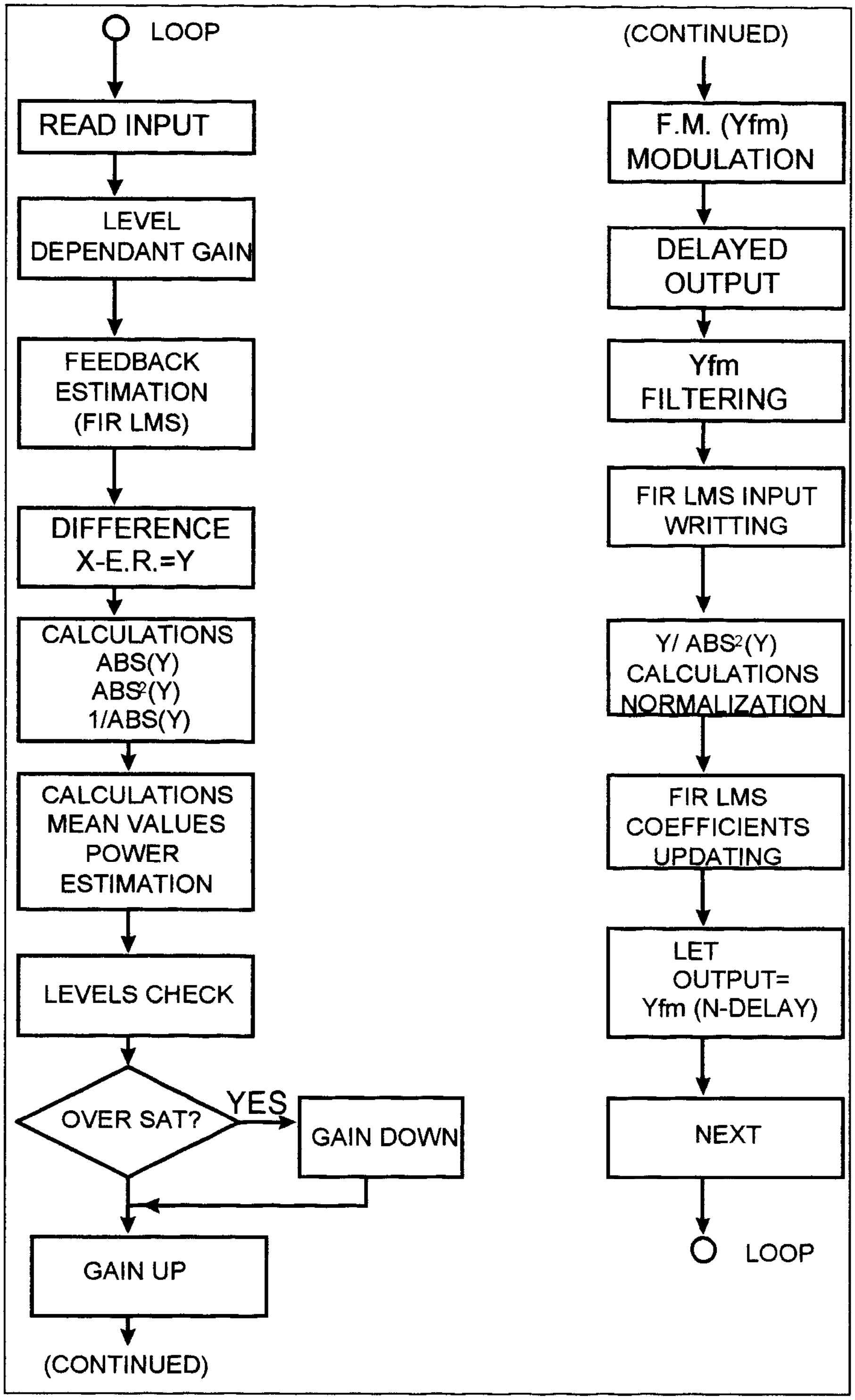

LOOP
READ INPUT
LEVEL DEPENDANT GAIN
FEEDBACK ESTIMATION (FIR LMS)
DIFFERENCE X-E.R.=Y
CALCULATIONS ABS(Y) ABS²(Y) 1/ABS(Y)
CALCULATIONS MEAN VALUES POWER ESTIMATION
LEVELS CHECK
OVER SAT?
YES
GAIN DOWN
GAIN UP
(CONTINUED)
(CONTINUED)
F.M. (Yfm) MODULATION
DELAYED OUTPUT
Yfm FILTERING
FIR LMS INPUT WRITTING
Y/ ABS²(Y) CALCULATIONS NORMALIZATION
FIR LMS COEFFICIENTS UPDATING
LET OUTPUT= Yfm (N-DELAY)
NEXT
LOOP

FIG.8

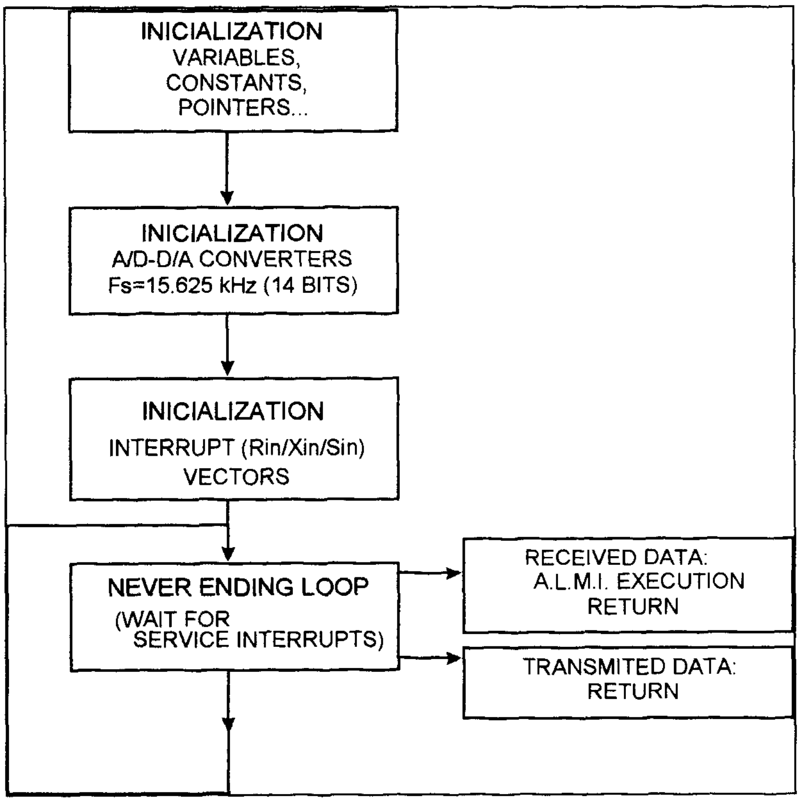

INICIALIZATION
VARIABLES,
CONSTANTS,
POINTERS...
INICIALIZATION
A/D-D/A CONVERTERS
Fs=15.625 kHz (14 BITS)
INICIALIZATION
INTERRUPT (Rin/Xin/Sin)
VECTORS
NEVER ENDING LOOP
(WAIT FOR
SERVICE INTERRUPTS)
RECEIVED DATA:
A.L.M.I. EXECUTION
RETURN
TRANSMITED DATA:
RETURN

FIG.9

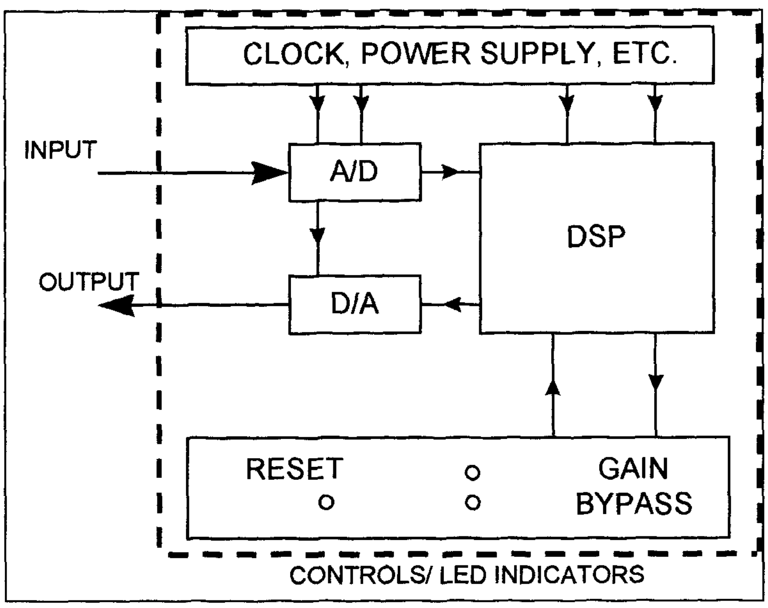
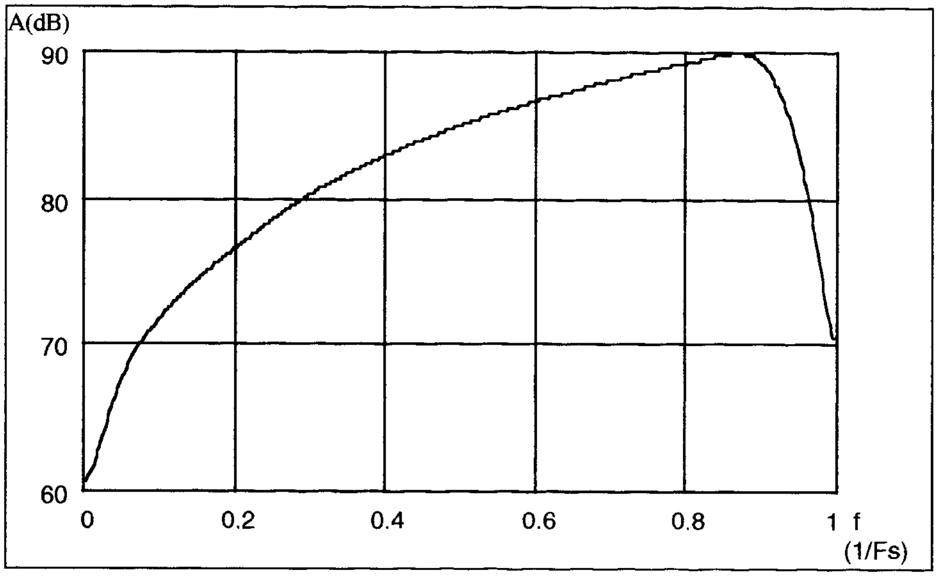

CLOCK, POWER SUPPLY, ETC.
INPUT
A/D
DSP
OUTPUT
D/A
RESET
GAIN
BYPASS
CONTROLS/ LED INDICATORS

FIG.10



FG.11

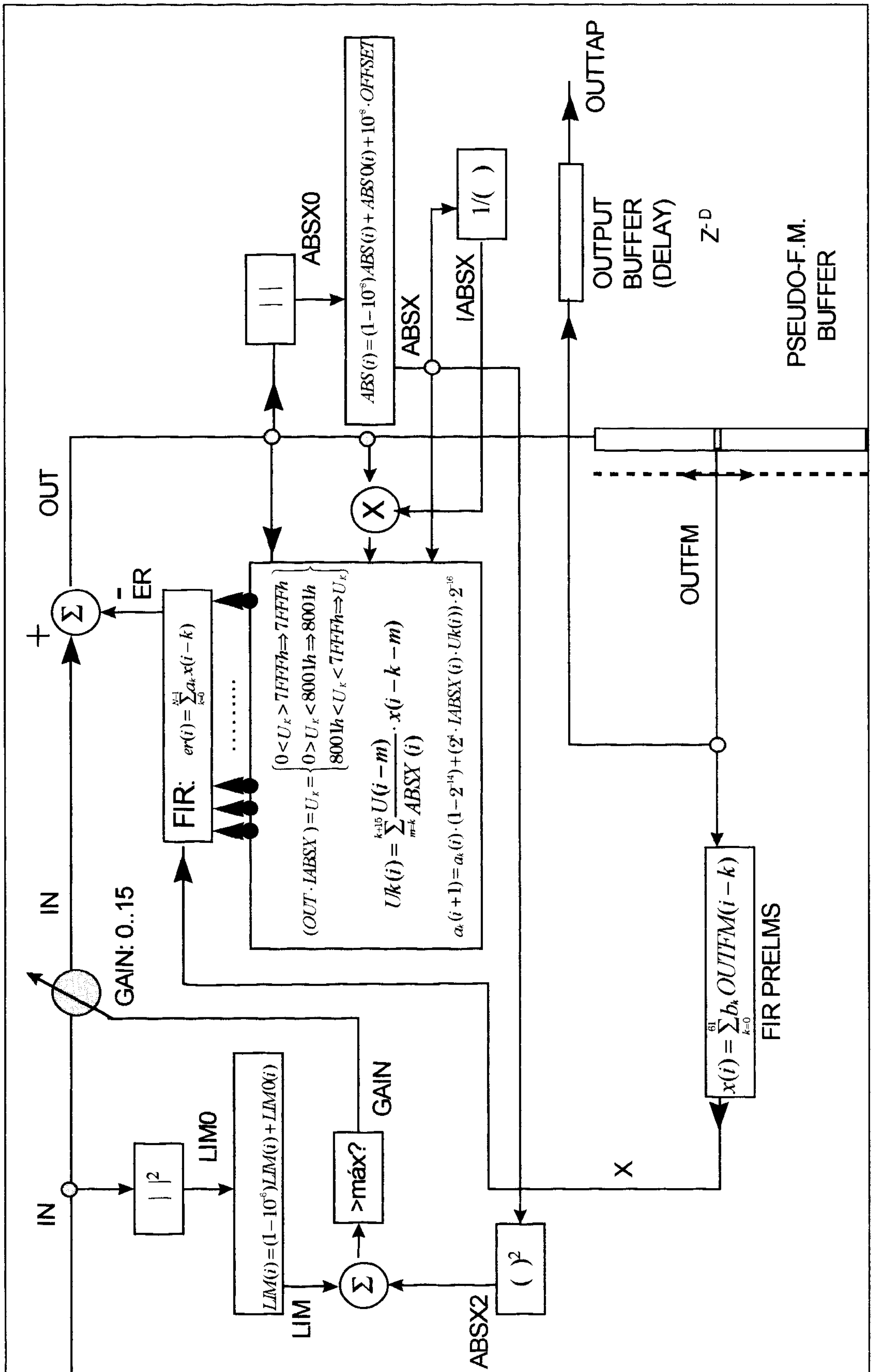

IN
OUT
ER
GAIN: 0..15
FIR
ABSX0
ABSX
IABSX
OUTTAP
OUTPUT
BUFFER
(DELAY)
PSEUDO-F.M.
BUFFER
OUTFM
FIR PRELMS
X
GAIN
>máx?
LIMO
LIM
ABSX2

FIG.12

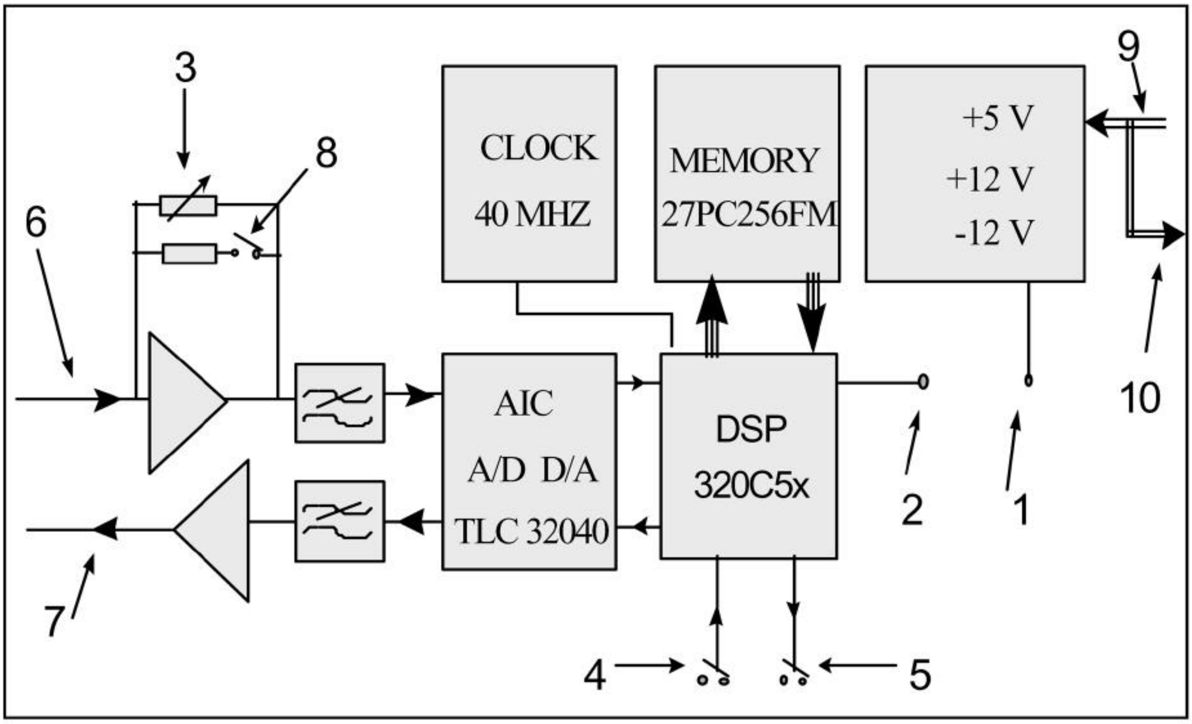

3
8
6
CLOCK
40 MHZ
MEMORY
27PC256FM
+5 V
+12 V
-12 V
9
10
AIC
A/D D/A
TLC 32040
DSP
320C5x
2
1
7
4
5

FIG.13

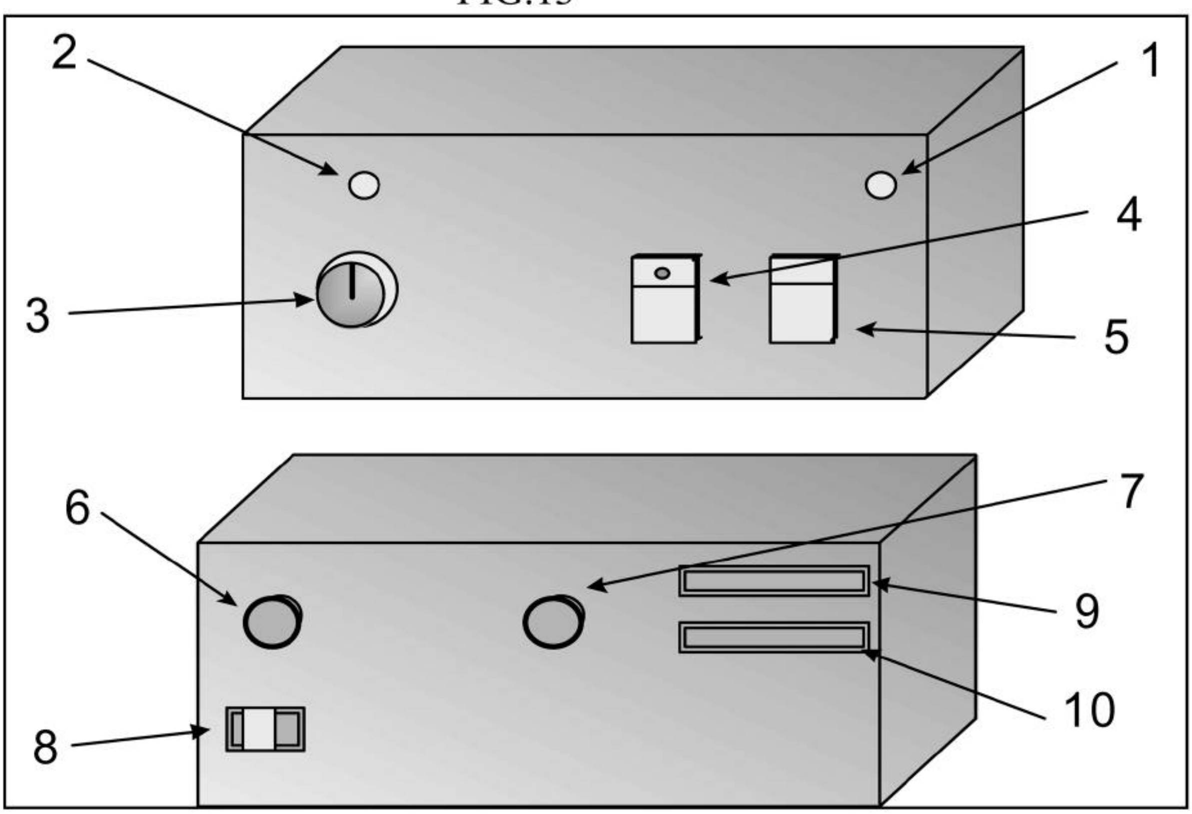

2
1
3
4
5
6
7
9
8
10

FIG.14

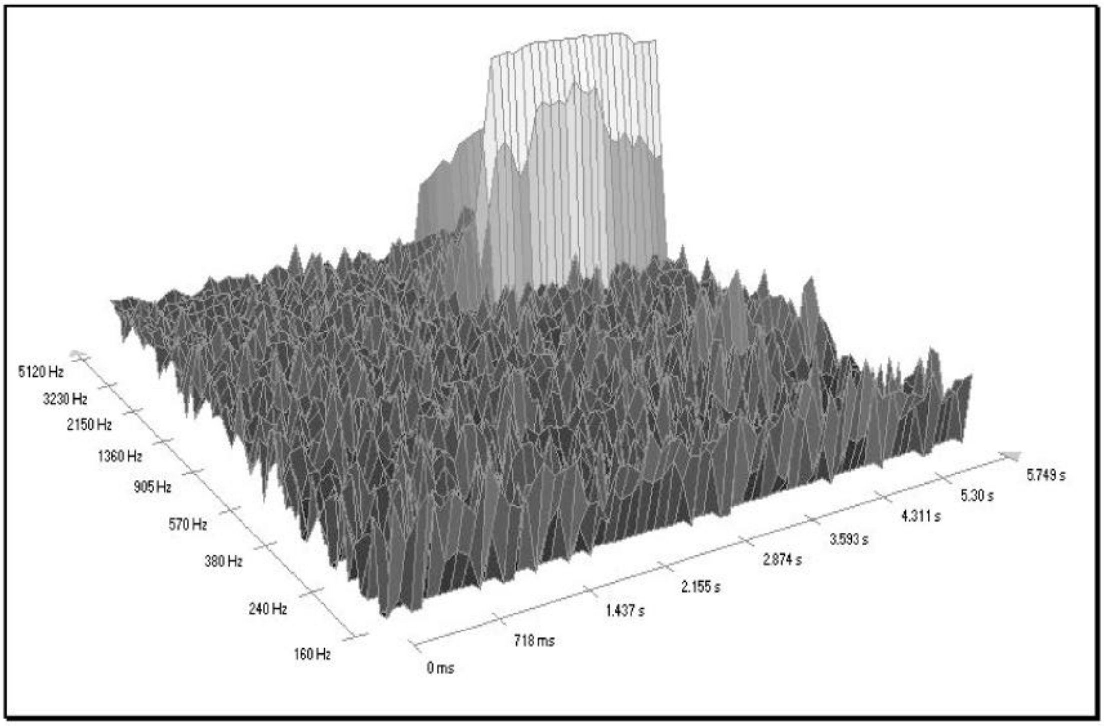

5120 Hz
3230 Hz
2150 Hz
1360 Hz
905 Hz
570 Hz
380 Hz
240 Hz
160 Hz
0 ms
718 ms
1.437 s
2.155 s
2.874 s
3.593 s
4.311 s
5.30 s
5.749 s

FIG. 15

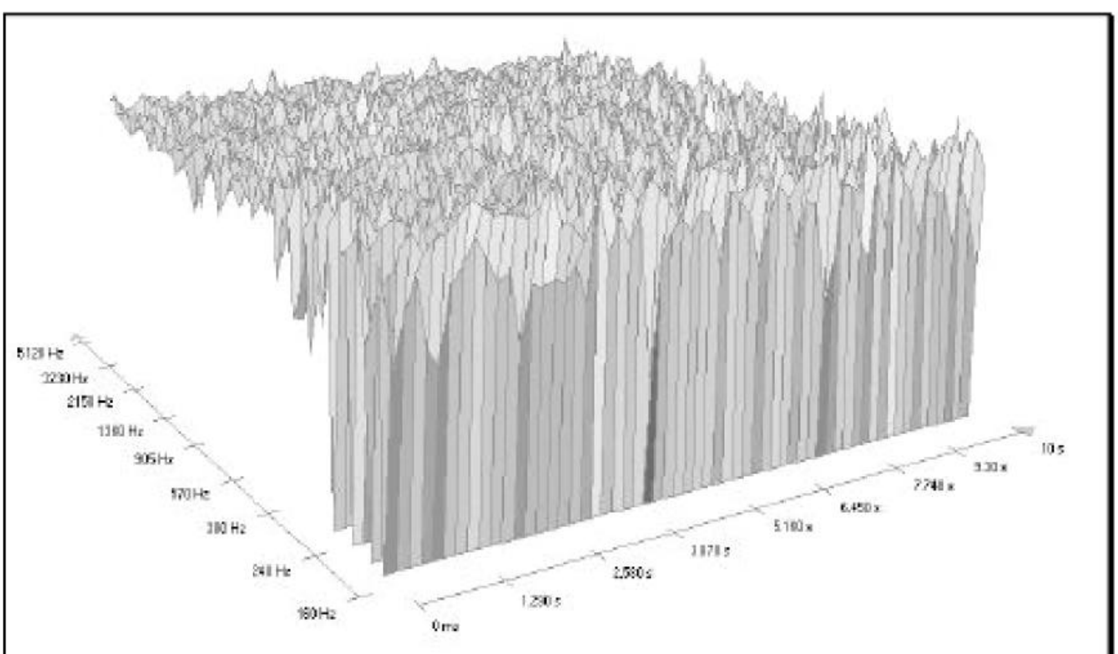

905 Hz
160 Hz
0 ms
1.290 s
2.580 s
5.160 s
6.450 s
7.740 s
10 s

FIG.16

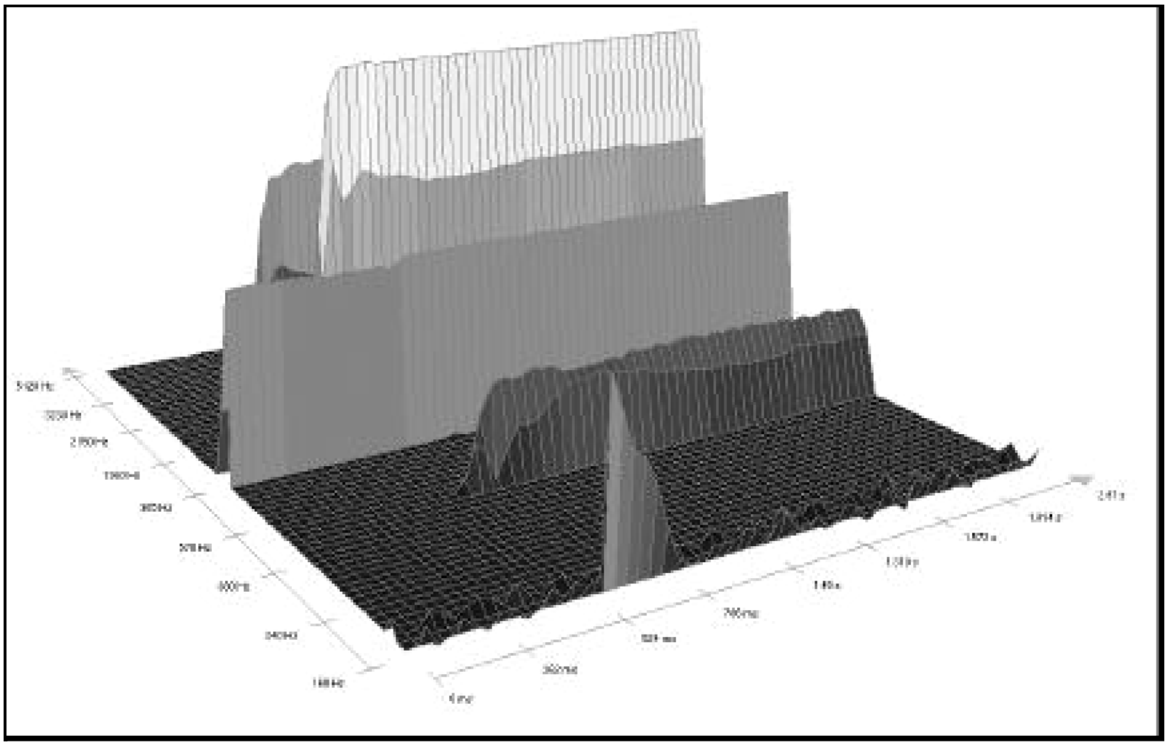


FIG.17

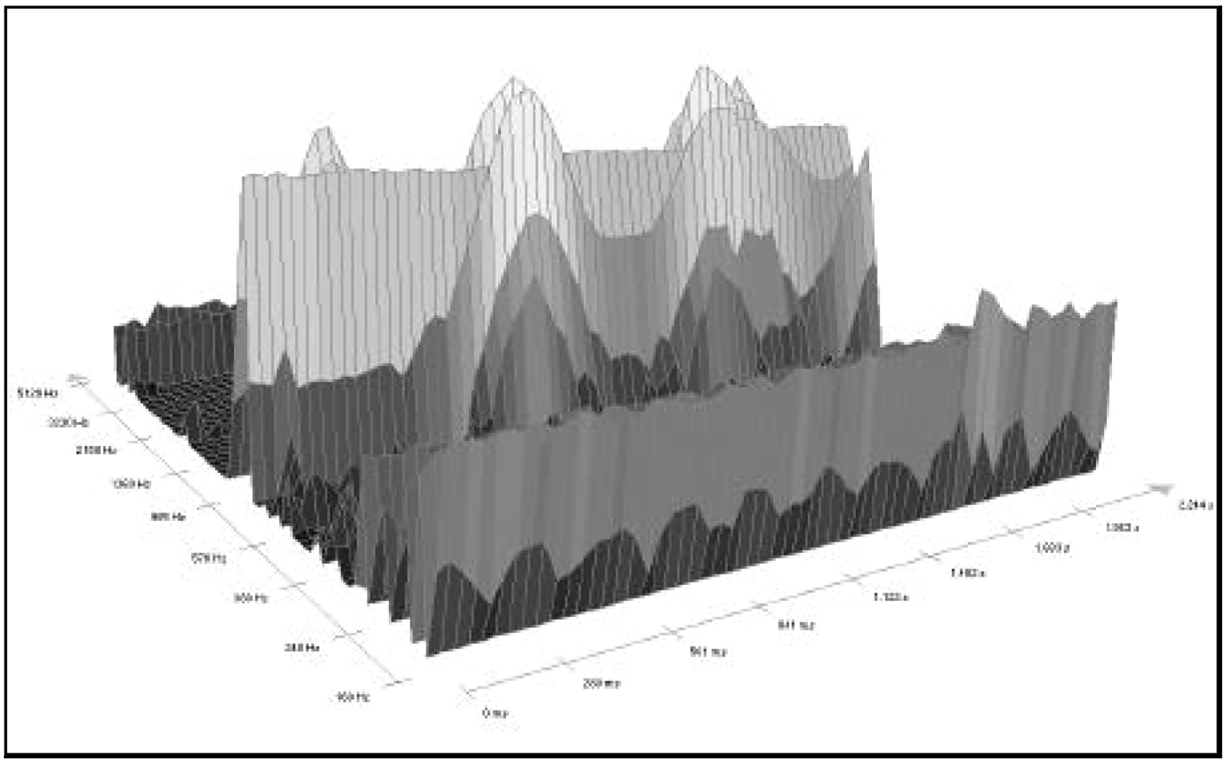


FIG.18

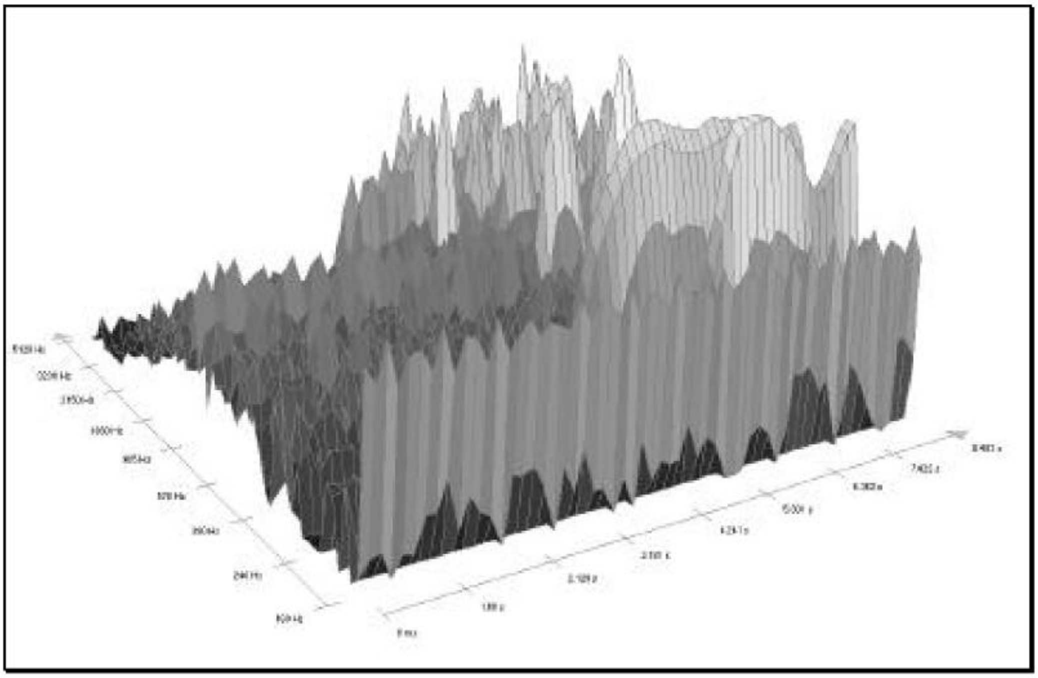


FIG.19