

ACTIVE NOISE CONTROL OF SOUND PROPAGATION IN THE DUCT

A DISSERTATION

*Submitted in the partial fulfillment of the
requirement for the award of the degree
of*

MASTER OF TECHNOLOGY

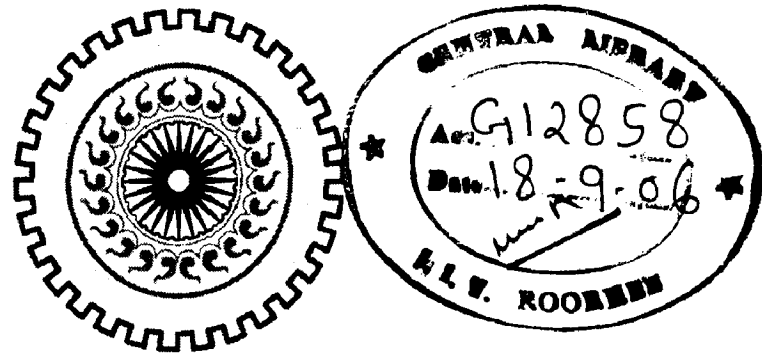
in

MECHANICAL ENGINEERING

(With specialization in Machine Design)

By

PAGADALA RAVI PRAKASH



DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247667(INDIA)

JUNE, 2006

CANDIDATE'S DECLARATION

I here by declare that the work, which is being presented in the dissertation work, entitled "**ACTIVE NOICE CONTROL OF SOUND PROPAGATION IN THE DUCT**", submitted in partial fulfillment of the requirements for the award of **Master of Technology in Mechanical & Industrial Engineering** with the specialization in **Machine Design, Indian Institute of Technology Roorkee, Roorkee** is an authentic record of my own work carried out during the period from July 2005 to June 2006 under the guidance of **Dr. M. BHATTACHARYA**, Professor and **Dr. S.C JAIN**, Professor Department of Mechanical & Industrial Engineering, Indian Institute of Technology Roorkee, Roorkee.

The matter embodied in this project work has not been submitted for the award of any other degree.

Date: June 30, 2006.

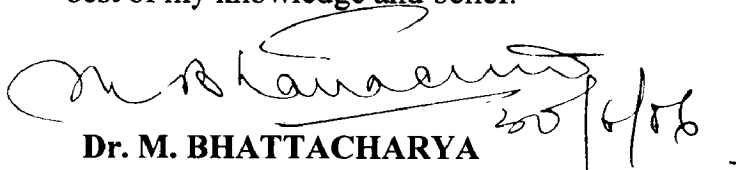
Place: Roorkee



(PAGADALA RAVI RAKASH)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.



Dr. M. BHATTACHARYA

Professor,

MIED,

I. I. T. Roorkee,

Roorkee-247667, India.



Dr. S.C JAIN

Professor,

MIED,

I.I.T. Roorkee,

Roorkee-247667, India

ACKNOWLEDGEMENT

I take the opportunity to pay my regards and deep sense of gratitude to my guides **Dr. M. Bhattacharya**, Professor and **Dr. S.C. Jain**, Professor, Department of Mechanical & Industrial Engineering, Indian Institute of Technology Roorkee, Roorkee, for their valuable guidance, keen co-operation, constant inspiration and cheerful encouragement through out the work.

I am thankful to **Dr. Mehra**, Professor and HOD, Department of Electronic & Computer Science Engineering, Indian Institute of Technology Roorkee, Roorkee, for providing me all the possible support.

I am also thankful to **Mr. Dinesh**, **Mr. Yadav**, **Mr. Mohamood** of the Department of Mechanical & Industrial Engineering, for their kind and timely co-operation rendered while working in their laboratories.

Special thanks are also due to my friends Sandeep, Ullas, Venu, Rama Rao, Sudhakar and Ravi Chand for their constant support and encouragement to finish this work.

I am happy to express my gratitude to my family whose love and blessings made this possible.

Finally, I would like to thank all those who have helped me during this work.

(Pagadala Ravi Prakash)

ABSTRACT

This report presents general background information about ANC methods. Contrasts between passive and active noise control are described and circumstances under which ANC is preferable are given. An ideal filtering problem is discussed to introduce notation, properties, and analysis techniques that can be applied to adaptive transversal filters and adaptive algorithms: Steepest descent and LMS are discussed in detail. The report details the design of a closed loop feedforward adaptive noise control with LMS algorithm and the implementation of that design in real time with a DSP kit (TMS320C6711). An experiment on a circular flow duct for active control of the acoustic transmissions produced $1/3^{\text{rd}}$ octave band acoustic source at central frequencies of 50Hz, 150Hz, 400Hz and 600Hz is performed. The attenuation of sound pressure level 16dB to 20dB for open loop adaptive control system and 32dB for closed loop adaptive control system is obtained at measuring section M4.

CONTENTS

Acknowledgement.....	i
Abstract.....	ii
Contents.....	iv
List of figures.....	vii
List of tables.....	ix
Chapter 1: INTRODUCTION	1
1.1 The General Concept of Acoustic Noise Control.....	1
1.2 The Development of Active Techniques for Acoustic Noise Control.....	4
1.3 Motivation.....	5
1.4 Organization of Thesis	5
Chapter 2: LITERATURE REVIEW.....	6
Chapter 3: EXPERIMENTAL SETUP AND PROCEDURE.....	10
3.1 Sound Fields in Ducts.....	10
3.2 Modes in Ducts	11
3.3 Impedance in Ducts.....	12
3.4 Passive Noise Control Approaches	13
3.4.1 Side-branch Resonator	14
3.4.2 Expansion Chamber.....	16
3.4.3 Helmholtz Filter.....	16
3.4.4 Dissipative Passive Control Techniques.....	17
3.5 Active Noise Control Approaches.....	18
3.5.1 Reference and Error Signal Quality.....	19
3.5.2 Types of ANC Systems.....	20
3.5.2.1 The Broadband Feed forward System.....	21
3.5.2.2 The Narrowband Feed forward System.....	22
3.5.2.3 The Feedback ANC System.....	24
3.5.2.4 The Multiple-Channel ANC System.....	25

Chapter 4:	ADAPTIVE SIGNAL PROCESSING TECHNIQUES.....	27
4.1	Open and Closed Loop Adaptation.....	28
4.2	Application of Closed Loop Adaptation	30
4.3	Optimal Transversal Filter.....	34
4.4	Steepest Descent Algorithm.....	37
4.5	LMS algorithm.....	42
Chapter 5:	INTRODUCTION TO SIGNAL PROCESSOR.....	46
5.1	Signal Processor Fundamentals.....	46
5.2	On Board Peripherals Used.....	46
5.2.1	EDMA Controller.....	47
5.2.2	Multi channel Buffered Serial Port (McBSP).....	47
5.2.3	TLC320AD535 On-board A/D converter8.....	48
5.3	DSP/BIOS Programming Techniques.....	49
5.4	CCS Programming Model.....	51
Chapter 6:	SIMULATION AND IMPLEMENTATION ON DSP KIT C6711....	52
6.1	Introduction to Audio Example.....	52
6.2	Software Interrupts.....	52
6.3	Pipe or PIP Module.....	53
6.4	Execution Semantics.....	54
6.5	DSS_txPipe for reuse by the audio function.....	55
6.6	Details of Algorithms and Flowcharts.....	56
Chapter 7:	DESIGN AND FABRICATION OF EXPERIMENTAL SET-UP.....	65
7.1	General lay-out of rig.....	65
7.2	Mountings for speakers.....	65
7.3	Rectangular Duct And Ateunuator.....	67
7.4	Microphone Holders.....	68
7.5	Experimental Procedure.....	71

Chapter 8: RESULTS AND DISCUSSION.....	73
8.1 Discussion on Open Loop Adaptation Results.....	73
8.2 Discussion on Closed Loop Adaptation Results.....	75
Chapter 9: CONCLUSION AND FUTURE WORK.....	80
REFERENCES.....	81
APPENDIX.....	84

LIST OF FIGURES

Fig.1.1	Physical Concept of Active Noise Cancellation General Applications of Active Noise Control.....	2
Fig.3.1	Plane wave sound propagation in a duct.....	12
Fig. 3.2	A side-branch resonator.....	15
Fig.3.3	An expansion chamber in an exhaust system.....	16
Fig.3.4	Helmholtz. or low pass, filter.....	17
Fig.3.5	Feedforward active noise control system for attenuating sound propagation in a duct.....	19
Fig.3.6	Microphone Mounting Method to Reduce Flow Turbulence.....	20
Fig. 3.7	Single-Channel Broadband Feedforward ANC System in a Duct	22
Fig .3.8	Feedback ANC System	24
Fig.3.9	Narrowband Feed forward ANC System.....	25
Fig.3.10	Multiple-Channel ANC System for a 3-D Enclosure	26
Fig .4.1	Open loop adaptation	29
Fig .4.2	Closed loop adaptation.....	29
Fig.4.3	Signals in closed loop adaptation	31
Fig. 4.4	Examples showing various configurations:(a) prediction;(b) system identification;(c)equalization;(d)interference canceling.....	33
Fig. 4.5	General transversal filtering problem	34
Fig 4.6	Contour plot of error surface	36
Fig.4.7	Convergence behavior of weight vector.....	40
Fig.5.1	Code Composer Studio Programming Model	51
Fig.6.1	Operation semantics of Pipe module.....	53
Fig.6.2	Execution semantics of an audio I/O example	54
Fig.6.3	More details of execution semantics	55
Fig.6.4	Simulation model of a FIR filter with LMS algorithm.....	56
Fig.6.5	Flow chart of FIR filter with LMS algorithm.....	60

Fig.6.6	Sine wave input of 50Hz frequency to the simulation model.....	61
Fig.6.7	Residual noise levels from the simulation model for 50Hz sine wave input.....	61
Fig.6.8	Sine wave input of 150Hz frequency to the simulation model.....	62
Fig.6.9	Residual noise levels from the simulation model for 150Hz sine wave input.....	62
Fig.6.10	sine wave input of 400Hz frequency to the simulation model.....	63
Fig.6.11	Residual noise levels from the simulation model for 400Hz sine wave input.....	63
Fig.6.12	sine wave input of 600Hz frequency to the simulation model.....	64
Fig 6.13	Residual noise levels from the simulation model for 600Hz sine wave input.....	64
Fig.7.1	Graphical view of socket and mounting for speakers.....	66
Fig.7.2	Socket and mounting for speakers.....	67
Fig.7.3	Rectangular duct having core 75 mm dia. of expanded metal mesh.....	68
Fig 7.4	Microphone mounting for duct wall pressure measurement.....	69
Fig.7.5	microphone mounting and 1 inch microphone to sense the primary noise level.....	70
Fig.7.6	The schematic diagram for the adaptively active control experiment of the acoustic transmission of a sound source in a duct.....	71
Fig.7.7	The experimental setup for the adaptively active control experiment of the acoustic transmission of a sound source in an aperture.....	72
Fig.8.1	Representation of different sections.....	74
Fig.8.2	The acoustic pressures generated by a primary pure-tone source at 50 Hz, and is measured at a point behind the secondary source.....	76
Fig.8.3	The acoustic pressure generated by a primary pure-tone source at 150 Hz, and is measured at a point behind the secondary source.....	77
Fig.8.4	The acoustic pressure generated by a primary pure-tone source at 400 Hz, and is measured at a point behind the secondary source.....	78
Fig.8.5	The acoustic pressures generated by a primary pure-tone source at 600 Hz, and is measured at a point behind the secondary source.....	79

NOMENCLEATURE

SYMBOL	DESCRIPTION
ANC	Active Noise Control
DSP	Digital Signal Processor
NMP	Non Minimum Phase
$x(n), s$	input signal to filter
$y(n)$	output signal from filter
$e(n), \varepsilon$	error signal
RPM	Revolutions Per Minute
AGC	Automatic Gain Control
n	noise signal
n^1	correlated version of the noise
FIR	Finite Impulse Response
L	Filter length
w	weighing coefficients
T	transpose
$E []$	statistical expectation
$d(n)$	desired filter output
p	cross correlation vector
J	cost function
R	correlation matrix
μ	step size parameter

SYMBOL	DESCRIPTION
Λ	diagonal matrix
λ_i	i^{th} eigen value is denoted
ξ	transformation of the weight error vector
SISO	Single Input Single Output
tr(R)	trace of R
EDMA	Enhanced Direct Memory Access
MCBSP	Multi Channel Buffered Serial Port
AICs	Analog Interface Chips
DX	Data Transmit
DRR	Data Receive Register
DXR	Data Transmit Register
XSR	Transmit Shift Register
RSR	Receive Shift Register
RBR	Receive Buffer Register
TSR	Interrupt Service Routine
MEM	Memory Section Manager
CCS	Code Composer Studio
SWI	Software interrupts
HWI	Hardware Interrupt
IDL	Idle Loop
S1, S2	primary source speaker
S2	control speaker
M1	reference microphone
M2	error microphone
M3	measuring microphone

CHAPTER 1

INTRODUCTION

1.1 The General Concept of Acoustic Noise Control

Acoustic noise problems in the environment have become more noticeable for several reasons.

- Because of a large number of high power industrial equipment used.eg.Engines, motors, transformers, blowers, compressors, fans, hammers, ball mills, presses etc.
- Noise exposure to people living in high density housings in proximity of highways.
- The use of lighter materials for building and transportation equipment, for higher strengths and cost considerations.

Two types of acoustic noise exist in the environment. One is caused by turbulence and is totally random. Turbulent noise distributes its energy evenly across the frequency bands. It is referred to as broadband noise, and examples are the low frequency sounds of jet planes and the impulse noise of an explosion. Another type of noise, called narrowband noise, concentrates most of its energy at specific frequencies. This type of noise is related to rotating or repetitive machines, so it is periodic or nearly periodic. Examples of narrowband noise include the noise of internal combustion engines in transportation, and in refrigerators, and vacuum pumps used to transfer bulk materials in many industries.

There are two approaches to controlling acoustic noise: passive and active. The traditional approach to acoustic noise control uses passive techniques such as enclosures, barriers, and silencers to attenuate the undesired noise. Passive silencers use either the concept of impedance change caused by a combination of baffles and tubes to silence the undesired sound (reactive silencers) or the concept of energy loss caused by sound propagation in a duct lined with sound-absorbing material to provide the silencing (resistive silencers). Reactive silencers are commonly used as silencers on internal combustion engines which have dominant tonal characteristics, while resistive silencers are used mostly for duct-borne fan noise which is a broadband in nature. These passive

resistive silencers are valued for their high attenuation over a broad frequency range. However, they are relatively large, costly, and ineffective at low frequencies, making the passive approach to noise reduction often impractical. Furthermore, these silencers often create an undesired back pressure if there is airflow in the duct.

In an effort to overcome these problems, active noise control is receiving considerable attention. The active noise control system contains an electro acoustic device that cancels the unwanted sound by generating an antinnoise (antinoise) of equal amplitude and opposite phase resulting in destructive interference. The original, unwanted sound and the antinnoise acoustically combine, resulting in the cancellation of both sounds. Fig.1 shows the waveforms of the unwanted noise (the primary noise), the canceling noise (the antinnoise), and the residual noise that results when they superimpose. The effectiveness of cancellation of the primary noise depends on the accuracy of the amplitude and phase of the generated antinnoise.

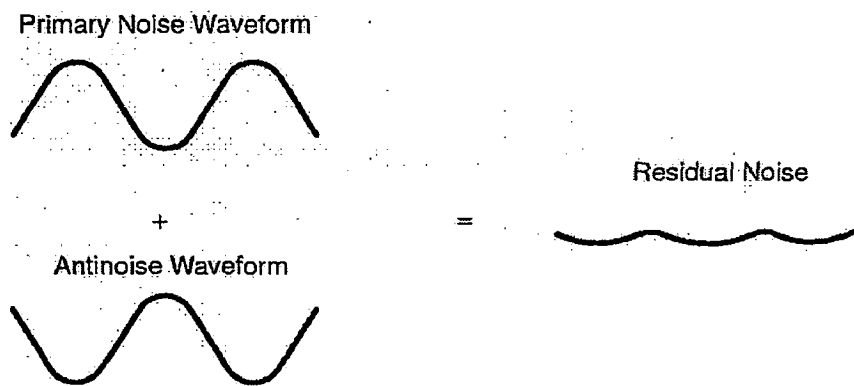


Fig. 1.1: Physical Concept of Active Noise Cancellation General Applications of Active

Noise Control

The successful application of active control is determined on the basis of its effectiveness compared with passive attenuation techniques. Active attenuation is an attractive means to achieve large amounts of noise reduction in a small package, particularly at low frequencies (below 600 Hz). At low frequencies, where lower sampling rates are adequate and only plane wave propagation is allowed, active control offers real advantages.

From a geometric point of view, active noise control applications can be classified in the following four categories:

- Duct noise: one-dimensional ducts such as ventilation ducts, exhaust ducts, air-conditioning ducts, pipe work, etc.
- Interior noise: noise within an enclosed space
- Personal hearing protection: a highly compacted case of interior noise
- Free space noise: noise radiated into open space

Specific applications for active noise control now under development include attenuation of unavoidable noise sources in the following end equipment.

- Automotive (car, van, truck, earth-moving machine, military vehicle)
 - Single channel (one dimensional) systems: Electronic muffler for exhaust system, induction system, etc.
 - Multiple channel (three dimensional) systems: Noise attenuation inside passenger compartment and heavy-equipment operator cabin, active engine mount, hands-free cellular phone, etc.
- Appliance

Single-channel systems: Air conditioning duct, air conditioner, refrigerator, washing machine, furnace, dehumidifier, etc.

Multiple-channel systems: Lawn mower, vacuum cleaner, room isolation (local quiet zone), etc.

Industrial: fan, air duct, chimney, transformer, blower, compressor, pump, chain saw, wind tunnel, noisy plant (at noise sources or many local quiet zones), public phone booth, office, cubicle partition, ear protector, headphones, etc.

Transportation: airplane, ship, boat, helicopter, snowmobile, motorcycle, diesel locomotive, etc.

The algorithms developed for active noise control can also be applied to active vibration control. Active vibration control can be used for isolating the vibrations from a variety of machines and to stabilizing various platforms in the presence of vibration disturbances. As the performance and reliability continue to improve and the initial cost

continues to decline, active systems may become the preferred solution to a variety of vibration-control problems.

1.2 The Development of Active Techniques for Acoustic Noise Control

Active noise control is developing rapidly because it permits significant improvements in noise control, often with potential benefits in size, weight, volume, and cost of the system. The book[1] provides detailed information on active noise control with an emphasis on the acoustic point of view.

The design of an active noise canceler using a microphone and an electronically driven loudspeaker to generate a canceling sound was first proposed and patented by Lueg [2]. While the patent outlined the basic idea of ANC, the concept did not have real world applications at that time. Because the characteristics of an acoustic noise source and the environment are not constant, the frequency content, amplitude, phase, and velocity of the undesired noise are nonstationary (time varying). An active noise control system must be adaptive in order to cope with these changing characteristics.

In the field of digital signal processing, there is a class of adaptive systems in which the coefficients of a digital filter are adjusted to minimize an error signal (the desired signal minus the actual signal; the desired signal is typically defined to be zero). A duct-noise cancellation system based on adaptive filter theory was developed by Burgess[3]. Later in the 1980s, research on active noise control was dramatically affected by the development of powerful DSPs and the development of adaptive signal processing algorithms [4]. The specialized DSPs were designed for real-time numerical processing of digitized signals. These devices enabled the low-cost implementation of powerful adaptive algorithms [5] and encouraged the widespread development and application of active noise control systems based on digital adaptive signal processing technology.

Many modern active noise cancellers rely heavily on adaptive signal processing without adequate consideration of the acoustical elements. If the acoustical design of the system is not optimized, the digital controller may not be able to attenuate the undesired noise adequately. Therefore, it is necessary to understand the acoustics of the installation and to design the system to assist the adaptive active noise controller to carry out its

work. For electrical engineers involved in the development of active control systems, Nelson's book [1] provides an excellent introduction to acoustics from the active noise control point of view.

1.3 Motivation

Internal combustion engines typically generate sound at the firing frequency and several of its are higher harmonics. The higher harmonics are amenable to suppression by reactive silencers (passive noise control technique) which fall in the range of a few hundred and few kilo Hz with compact silencers. However the fundamental firing frequency range is 30 Hz to 250 Hz. For this case large silencers in dimensions are required. So for this range of frequency active noise control appears to offer the most effective solution.

1.4 Organization of Thesis

Chapter I includes the introduction of thesis, brief description about the general concept of acoustic noise control and the development of active techniques for acoustic noise control. Chapter II gives a brief literature review. Chapter III includes the different passive and active noise controlling methods in ducts. Chapter IV deals with the variety of adaptive signal processing techniques. Chapter V gives the overview of DSP kit model TMS320C6711. Chapter VI discusses on simulation of FIR closed loop adaptive filter with LMS algorithm and interfacing the DSP kit with PC. Chapter VII is about the details of experimental set up and experimental procedure. Chapter VIII consists of results and discussion. Chapter IX gives concluding remarks.

Kristiansen [6] investigated of quarter-wave resonators where the volume is divided by a slanted wall, creating volumes of decreasing and increasing cross-sectional areas. It was shown that such a resonator has two fundamental frequencies, on either side of the original. Simple prediction formulae were given and coincide well with experimental results.

Lamancusa [7] presented expression for the transmission loss for the case of chambers of unequal length and area. It was also shown that significant tuning benefits result with chambers of unequal size. For evenness of response, with a minimum passband, chamber length ratios of 2:1 and connecting tube lengths equal to the shortest chamber are superior.

Selamet *et al* [8] developed two-dimensional analytical solution to determine the acoustic performance of a perforated single-pass, concentric cylindrical silencer filled with fibrous material. To account for the wave propagation through absorbing fiber and perforations, the complex characteristic impedance, wave number, and perforation impedance are employed. With expressions for the eigenvalues and eigenfunctions of sound propagation in the perforated dissipative chamber, the transmission loss is obtained by applying a pressure and velocity matching technique.

Chen *et al* [9] discussed the improvement on the acoustic transmission loss of a duct by adding some Helmholtz resonators. Therefore, the calculation on the transmission loss of a duct in a rigid wall by modifying the formula derived by **Wilson and Soroka [10]**, its relating measurement and both the calculation of improved transmission loss of a duct by adding band-pass filters, and measurement of its involved transmission loss were included in his study. The results of this study shows that agreement of the calculated transmission loss of a duct in a rigid wall with its related measurement is quite good at frequencies above 200 Hz, and the improvement of almost 28 dB on the transmission loss of the adopted duct at the frequency where the related transmission loss is to be improved can also be obtained when some appropriate Helmholtz resonators are added.

Galland *et al* [11] gave the complete design procedure of a new type of acoustic liner, combining passive absorbent properties of a porous layer and active control. The final objective was to achieve, for a wide frequency range, a targeted impedance, which was theoretically predetermined to produce the best noise reduction when applied to a flow duct. The optimization phase is carried out on the passive and active part of the system.

poole *et al*[12] constructed a unidirectional array of secondary sources successfully around a rectangular duct, using loudspeaker drive units and electronic delays. Sound propagating in the direction of these sources was sampled and a control signal applied to the sources which in turn acted to significantly reduce the amplitude of the sound. Pure tones at frequencies around 150 Hz have been attenuated by more than 50 dB but results with band-limited noise have been less successful.

Valimaki *et al* [13] described a DSP implementation of an active noise control system which attenuates disturbing noise in a ventilation duct. The system includes a flat panel loudspeaker, two microphones, and a DSP hardware unit which consists of a Texas Instruments TMS320C32 floating-point signal processor and fast 16-bit analog-to-digital and digital-to-analog converters. The principle of adaptive feedforward control using filtered-x LMS algorithm was employed to cancel broadband noise at frequencies between 200 and 800 Hz.

Randolph *et al* [14] described a principal component least mean square (PC-LMS) adaptive algorithm that has considerable benefits for large control systems used to implement feedforward control of single frequency disturbances. The algorithm is a transform domain version of the multichannel filtered-x LMS algorithm. The transformation corresponds to the principal components (PCs) of the transfer function matrix between the sensors and actuators in a control system at a single frequency. The method is similar to other transform domain LMS algorithms because the transformation can be used to accelerate convergence when the control system is ill-conditioned.

Bai *et al* [15] proposed an adaptive spatially feedforward algorithm for broadband attenuation of noise in ducts. Acoustic feedback generally exists in this active noise control structure. **Munjal and Eriksson [16]** derived an ideal controller for the spatially

feedforward structure. The ideal controller can be partitioned into two parts. The first part represents a repetitive controller that can be implemented by an infinite impulse response (IIR) filter, whereas the second part represents the dynamics of transducer that can be implemented by a finite impulse response (FIR) filter. In his paper, the IIR filter was merged with the original plant. The FIR filter was adaptively updated by the least-mean-square (LMS) algorithm to accommodate perturbations and uncertainties in the system. The proposed algorithm was implemented via a floating point digital signal processor and compared with other commonly used algorithms such as the Filtered-X LMS algorithm, the feedback neutralization algorithm.

kang *et al* [17] studied mean intensity based active control for the cancellation of radiated noise out of the duct exit . The active intensity control strategy was derived based on the relation of the exterior sound field radiated out of the duct termination and the interior sound field of the duct. One of the characteristics of this control strategy is that the maximum possible control performance can be maintained regardless of the sensor location, compared with the conventional local pressure control methods at either interior downstream or exterior field positions. This is a simple consequence of the active intensity at the interior downstream being not space-dependent as long as it is plane wave. A time-domain adaptive filtering method for the active intensity control was also suggested and experimental results for an open ended duct based on the adaptive filtering method were presented. For the purpose of practical comparison, experimental results for conventional sound pressure control based on the well known filtered LMS algorithm were also presented. The experimental results showed the potential of the active intensity control strategy for reducing the emitted noise out of the duct exit.

Jing Yuan [18] proposed a hybrid active noise controller (ANC) to solve some existing problems, which are related to the non-minimum phase (NMP) path models between unallocated sensors and actuators in many ANC systems. For hybrid ANC schemes, the NMP path causes design difficulties to both feedforward and feedback control. These problems can be solved effectively by adding an extra actuator in the ANC system. A new design procedure is presented to take the greatest advantage of the extra actuator.

Jerome *et al* [19] implemented hybrid absorption systems which achieve high sound absorption over a broad frequency range. This work was an experimental study of a broadband hybrid absorption system which was comprised of a layer of sound-absorbing material (the passive component) positioned at a distance from a movable wall (the active component) inside an impedance tube. The movable wall was used to impose desired boundary conditions in the cavity behind the passive layer, thereby increasing the absorption of the system at frequencies where the passive material was not independently effective. Both pressure-release (i.e., minimizing the pressure at the back surface of the layer) and impedance-matching (i.e., minimizing the reflected wave from the layer) boundary conditions were studied.

CHAPTER 3

CONTROL OF SOUND PROPAGATION IN DUCTS

The final group of noise control problems which we will briefly discuss here concerns the control of sound propagation in ducts. Ducts can be viewed as enclosures where one dimension is very long, often terminating into open space. Common examples of a duct include the airways used in central heating and cooling systems, and any piping system (including vehicle exhaust systems). Another example of a duct which may not be so obvious is a long hallway connecting two adjacent rooms or halls. The essential acoustic ingredient for a duct is that sound waves be constrained in two dimensions while being allowed to travel more-or-less freely in the third. For this reason, ducts are often referred to in acoustics literature as waveguides, where the constraining walls guide the travel of sound waves in the third dimension.

Before examine the passive and active noise control approaches to attenuating sound propagation in ducts, examine the important characteristics of the sound field in ducts. Once a set of characteristics is established it is straightforward to understand and optimize the physical mechanisms behind the noise control approaches.

3.1 Sound Fields in Ducts

The idea that a duct is simply an enclosed space with the boundary removed on one side provides us with clues about the structure of a sound field in a duct. It is intuitive that in describing the sound field we should essentially considering the bounded sides and the open sides separately. The bounded sides can be expected to have a modal response, similar to that of a fully enclosed space. When a sound wave fits in the cross section the response will peak. The frequency where this occurs is a resonance frequency. If, for example, the duct cross section is rectangular, then we will have resonances for what are essentially axial and tangential enclosure modes.

The open side can be expected to have a response similar to that associated with free space sound radiation in the absence of any boundaries. The sound waves will simply travel away and there will be no resonance response associated with this side of

the duct. This intuitive model of sound fields in ducts is essentially correct. There are, however, some details which must be added to provide a complete description.

3.2 Modes in Ducts

Ducts do have a modal form of response. However, unlike modes in an enclosed space, duct modes are restricted to being one- or two-dimensional (associated with the cross section). Duct modes also propagate, or move down the duct.

The fundamental, or lowest frequency, mode in a duct is the plane wave mode. Referring to Fig. 3.1, a plane wave has a uniform sound pressure distribution in the duct cross section, and has the same waveform of pressure distribution down the duct as an acoustic wave in free space. A plane wave does not have a “true” resonance response associated with cross section. In theory, the plane wave mode has a resonance frequency of 0 Hz, being an axial mode with an infinite length in one direction.

All modes in the duct other than the plane wave mode are referred to as higher-order modes. These modes have an enclosure-like modal pressure distribution in the cross section. It was mentioned that duct modes will travel, or propagate, down a duct. We can view the acoustic energy that is flowing down a duct to be divided up amongst the traveling modes; each mode carries a bit of the total energy. One of the most important results in duct acoustics is that a duct mode can only travel if the frequency of sound is greater than or equal to what we have been thinking of as the resonance frequency of the mode. Because of this result, ducts modes are usually referred to as having a cut-on frequency, rather than a resonance frequency. At all frequencies above this one, the duct will be “cut-on” and physically allowed to carry acoustic energy down the duct. If the frequency is below a cut-on of a given duct mode the mode is said to be “cut-off,” or evanescent.

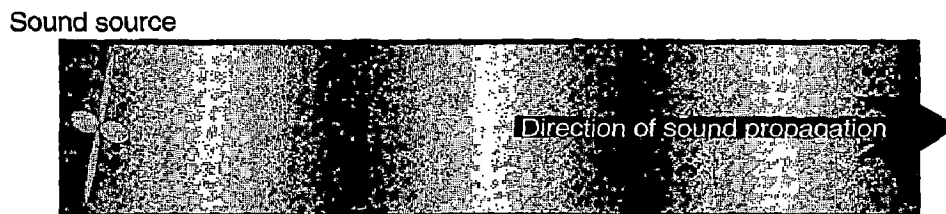


Fig.3.1: Plane wave sound propagation in a duct.

Note that the pressure distribution is uniform in all duct cross sections, and that the pressure peaks and troughs travel down the duct at the speed of sound.

The fact that duct modes can only travel and carry energy at frequencies above their cut-on frequency has significant implications for noise control approaches. For example, if the duct cross section is small compared to the frequency of interest, and so none of the higher-order modes are cut-on, then essentially all of the acoustic energy flowing down the duct will be carried by the plane wave mode. Therefore, any noise control approach we may wish to adopt must specifically target the plane wave mode. Higher order modes do not explicitly have to be targeted for attenuation.

The process of cutting-on a mode is not binary. It is not the case at 1 Hz below the cut-on that the mode is doing nothing, and at 1 Hz above the cut-on it is carrying a large amount of acoustic energy. As mentioned, when the frequency is below the cut-on of a mode the response of the mode will decay with increasing distance from the source. As the frequency approaches cut-on, the rate of decay decreases. To illustrate the mode is like a bit of chewing gum stuck to the bottom of your shoe. While it will stay attached to the location of the noise source, it can be “stretched” in the same way chewing gum will stretch when you try to pull it off your shoe. The closer the noise source frequency is to cut on, the longer it will stretch. At cut-on, a bit of the mode is finally allowed to break away, just like a bit of chewing gum coming off your shoe. The entire mode does not suddenly propagate down the duct, just a bit of it. As the frequency increases beyond the cut-on, more and more bits of the mode can propagate, and so the acoustic energy carried by the mode increases as the frequency increases beyond the cut-on frequency. Therefore, when assessing how important a duct mode is in carrying the portion of the total acoustic energy, and so assessing how much attention should be paid to it in the noise control process, it is not enough to know simply whether the mode is cut-on. When it cut on is also important.

3.3 Impedance in Ducts

When a sound waves travels in free space, it sees only one impedance. Technically, this is the specific acoustic impedance, defined by the ratio of (pressure)/(particle velocity), where particle velocity is the velocity of a small bit of the continuous medium in which the acoustic wave flows. Specific acoustic impedance for a

given medium is defined by the product: (speed of sound in the medium) x (density of the medium). If a duct is infinitely long, and so the acoustic wave never reached the end, then the impedance the wave would see in the duct would be the same as the free space impedance.

However, most real ducts are very much finite in length. When the wave hits the end of the duct, it will usually hit an impedance change in moving from the duct to the open environment into which the duct exhausts. This impedance change will cause some of the traveling wave to be reflected and move back upstream. This scenario has obvious implications for acoustic energy flow and noise control approaches.

Calculation of the actual impedance seen by an acoustic wave as it looks down a duct with some specific end conditions is complicated. However, a few general comments can be made.

- Impedance is a complex number quantity. The real part of the impedance, which is the resistance, is associated with acoustic energy flow. The imaginary part of the impedance, which is the reactance, is not associated with acoustic energy flow. Therefore, to reduce acoustic energy flow, it is necessary to reduce the real part of the impedance only. If the end conditions of the duct were such that there was no real part to the impedance, then the acoustic power output of the duct would be zero. Achieving this is often the aim of vehicle muffler design.
- For a duct exhausting into free space, the resistance is dependent upon a number of quantities. These include: duct cross-sectional area, duct perimeter, duct length, characteristics of the medium (such as air), frequency of sound, and the velocity of air flowing down the tube.
- It is possible to have a given end arrangement such that the impedance of the duct is the same as that of the free space environment. In this case, there will be no reflection of the sound waves back upstream. Wave reflection requires an impedance change.

3.4 Passive Noise Control Approaches

The noise control practitioner has a wide variety of techniques available for tackling the problem of sound propagation inside of, and eventually out of, ducts this. While these methods can again be divided into two groups, acoustic energy flow

redirection and acoustic energy flow reduction, the redirection options are rather trivial from a technical standpoint. They basically consist of pointing the duct exhaust in another direction, such as straight up, or erecting a wall in front of the exhaust to direct the energy flow elsewhere.

The acoustic energy flow reduction techniques for the duct noise problem can be further divided into two categories: those which specifically aim to alter the duct impedance, referred to as reactive techniques; and those which specifically aim to absorb the acoustic energy as the wave propagates down the duct, referred to as dissipative techniques. In general, reactive techniques are best applied to low-frequency noise problems as they tend to provide more compact solutions than dissipative techniques. Dissipative techniques are best applied to mid- and high-frequency noise problems as they tend to work over a wider frequency range and are cheaper and simpler to build. We will first consider reactive techniques for passive noise control, and restrict ourselves to three common devices: the side-branch resonator (which includes the Helmholtz resonator), the expansion chamber, and the Helmholtz filter.

3.4.1 Side-branch Resonator

A side-branch resonator is a useful device for attenuating pure tone sound propagation in a duct, such as might arise from a fan. A side-branch resonator is basically a tuned piping arrangement placed off the main piping run, as shown in Fig 3.2. The tuned arrangement is commonly a sealed section of pipe with a length equal to one-quarter the wavelength of the target frequency (a “quarter-wave stub”), or a “volume” section connected to the pipe via a smaller orifice section (a “Helmholtz resonator,” which is shown in the diagrams of Fig 3.2). The aim of the side-branch resonator is to essentially offer the sound wave a parallel alternative to the main piping run, a parallel that is designed to have negligible impedance at the target frequency. This is analogous to running through the jungle and being offered an alternative, parallel path of an open field. The state of negligible impedance occurs at the resonance of the side-piping section, which is why it is called a resonator. If the resonator is perfect and so the impedance is zero, then all of the acoustic energy will flow into the resonator and none will continue down the duct.

For the Helmholtz resonator sketched in the figure, the resonance frequency is defined by

$$f_{res} = \frac{c}{2\pi} \sqrt{\frac{A}{LV}}$$

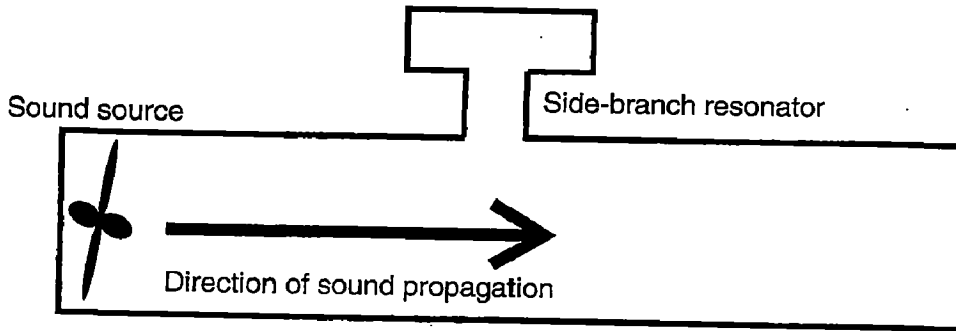


Fig. 3.2. A side-branch resonator.

Where c is the speed of sound (343 meters per second, nominally), A is the cross-sectional area of the pipe connecting the volume to the pipe, L is the length of the connecting pipe, and V is the volume of the enclosed space behind. It should be noted that it is difficult to manufacture a “perfect” side-branch resonator, one which will resonate at some specified frequency when it is placed in the duct. It is common to include some way of adjusting the system for in situ tuning.

In order for the resonator to be most effective in providing sound attenuation, it should be placed at a point in the duct where the acoustic pressure is maximum. Intuitively, the pressure will then be most sensitive to an impedance change. This is commonly an odd multiple of one-quarter wavelengths downstream from the noise source (fan, etc.).

3.4.2 Expansion Chamber

An expansion chamber is basically a relatively large opening in the piping system, as shown in the diagram of Fig 3.3.

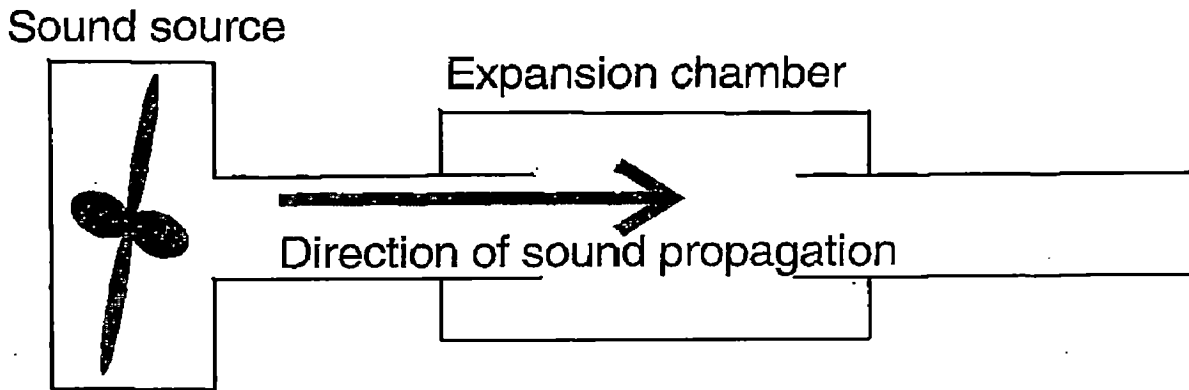


Fig. 3.3: An expansion chamber in an exhaust system.

They are perhaps the most common form of muffler fitted on two-stroke engines. Expansion chambers are able to provide sound attenuation over a wider range of frequencies than the previously discussed side-branch resonator. Referring to the figure, the ability of the expansion chamber to provide sound attenuation at low frequencies is limited by the resonance of the volume/exhaust tube combination. This resonance frequency can be calculated using the same equation as given for calculating the resonance frequency of the Helmholtz resonator, where the speed of sound c is often higher as the medium is hot, high-speed exhaust gas. At the resonance frequency, the expansion chamber actually amplifies the noise, rather than attenuating it. Above the chamber/tailpipe resonance frequency, attenuation will be provided up to the point where resonances of the actual piping system come into play.

3.4.3 Helmholtz Filter

An extension of the expansion chamber idea is the low-pass, or Helmholtz, filter shown in the diagram of Fig. 3.4. This device is commonly used to suppress pressure fluctuations in flowing gas. Qualitatively, the performance of the system can be viewed as a magnification of the expansion chamber performance. Low-frequency attenuation is limited by the resonance frequencies of the chambers, as at resonance the noise will be amplified. Attenuation is provided at frequencies above this. The precise high-frequency performance is dependent upon the tailpipe conditions (long, short, etc.).

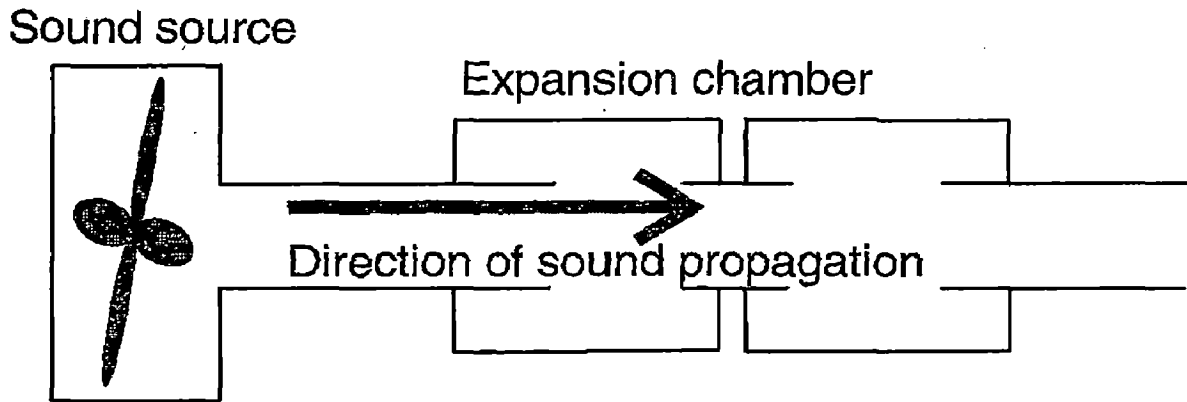


Fig. 3.4. Helmholtz. or low pass, filter.

In general, the Helmholtz filter attenuates higher frequencies while having little effect upon low frequencies, where low is defined by the chamber resonance frequencies (these can be very low). For this reason, it has the form of a “low-pass filter.” Low-frequency components of the sound field (only) are allowed to pass.

3.4.4 Dissipative Passive Control Techniques

Dissipative approaches to passive noise control in ducts aim to absorb the sound field as it travels along, converting the energy in the acoustic field to a minute amount of heat. Dissipative passive noise control most commonly takes the form of a porous lining material (sometimes referred to as “fluff”) placed on the walls of the duct. In industrial settings, the fluff is often protected by a thin, limp piece of plastic, similar in many ways to kitchen plastic wrap, and/or a perforated metal panel. In the perforated panel case, provided that the open area provided by the holes is approximately 25% or more of the total panel area, the effect on the acoustic performance of the liner is negligible (provided that the holes themselves do not “whistle”).

Proper design of a dissipative duct liner requires balancing many parameters. Some general trends are as follows:

- The three most important parameters for determining the level of sound attenuation for a given duct are: (i) the flow resistivity of the lining material; (ii) the thickness of the lining material relative to the diameter of the open (unlined) portion of the duct; and (iii) the area of duct over which the liner is installed. The

area is defined by both the length of lined section and how much of the duct perimeter is lined (all sides, two sides, etc.).

- The level of sound attenuation provided varies with frequency. When plotted as a function of frequency, the attenuation properties of a given lining arrangement generally peak at some middle frequency and fall off at higher and lower frequencies. The division of high, low, and middle frequencies is based upon the ratio of the radius of the open area in the center of the duct to the wavelength of sound. Performance is often best when these are approximately equal. In general, the greater the peak attenuation (in the middle), the worse the performance at frequencies away from the peak.
- Generally, a thin layer of lining (say, one-fifth of the radius of the open area of the duct) will have a greater peak performance than a thick layer of lining. However, the performance away from the peak will be significantly worse.
- Dissipative techniques work best when several higher-order modes in the duct are cut-on. Conversely, performance is generally the poorest for the plane wave mode. This makes sense intuitively, as the higher-order modes impinge upon the sides of the duct whereas the plane wave mode does not.
- Generally, to achieve any sort of meaningful attenuation, the duct lining should be placed over a length which is ten times or more the diameter of the open section of duct.

3.5 Active Noise Control Approaches

The application of active noise control to problems of sound propagation in ducts was one of the originally envisaged uses when the technology was patented over 60 years ago. Even today, it is the most popular application of the technology. While there is a variety of possible ways to implement an active noise control system in a duct, we will concentrate our discussion here on the feedforward systems. These are by far the most popular, and arguably most useful, arrangements for the duct noise problem.

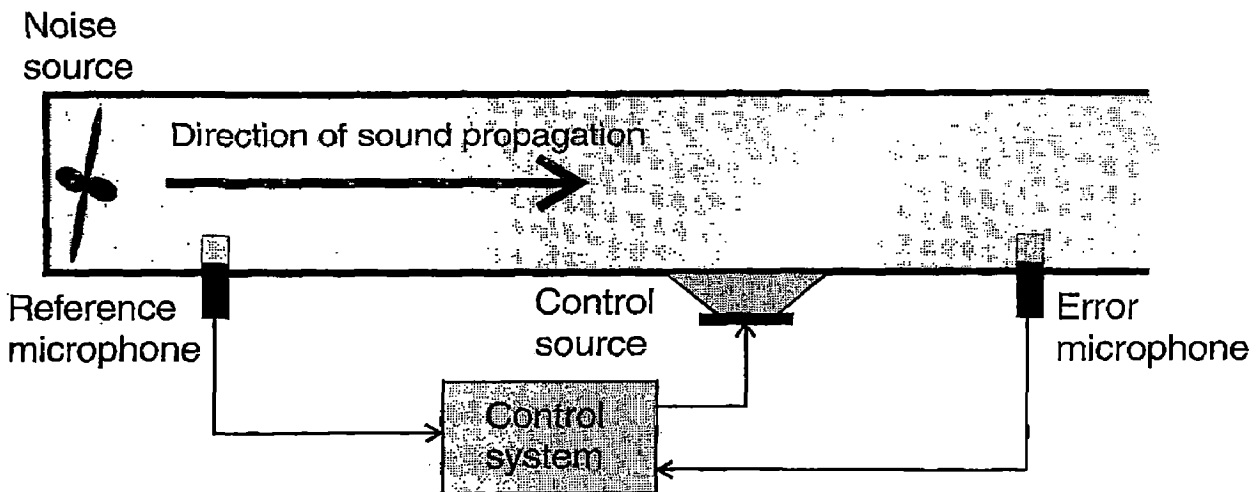


Fig 3.5. Feedforward active noise control system for attenuating sound propagation in a duct.

Shown in Fig 3.5 is a diagram of the main components of a feedforward active noise control system in a duct. Ignoring the controller part of the system for the moment (for a later chapter), the most important items, insofar as determinants of performance of the duct active noise control system, are in no particular order:

- The quality of the reference signal, in particular, the ability of the reference sensor to provide a good measurement of the propagating sound field without a large amount of corruption from air flow past the microphone;
- The quality of the error signal measurement (similar to (1));
- The separation distance between the reference sensor and control source;
- The location of the control source in the duct and
- The characteristics of the duct response, in particular, what modes are cut-on or close to cut-on, for the target frequency range.

3.5.1 Reference and Error Signal Quality

The signal from the reference sensor (the reference signal) provides the controller with an indication of the impending disturbance. Given this signal, it is the job of the controller to calculate a suitable canceling signal. It is obvious, then, that the reference sensor must actually be measuring the sound field propagating down the duct. For active noise control implementations targeting sound fields in ducts, is not straightforward. For

making measurement of sound field in air-conditioning duct a microphone inserted into the duct, chances are that turbulent pressure fluctuations over the microphone diaphragm resulting from the flow. The actual sound field measurement will can be the case even for buried somewhere in the air flow noise.

To remedy the situation use is made of cellular wind screens over microphone for low air velocities. It is common to see wind screens, which resemble foam balls, placed on microphones in public address systems. But these are usually ineffective in a duct active noise control implementation. The air flow noise is simply too much of a problem for common wind screens. Most commercial-grade active noise control systems in ducts use high quality “antiturbulence” microphone systems for both reference and error signal acquisition. The remedy is that placing the microphone in a small, outer turbulence tube connected with the duct through a small slit, as shown in Fig. 3.6, can significantly increase the coherence. The placement of the microphone in the outer turbulence tube also has advantages in component protection and maintenance.

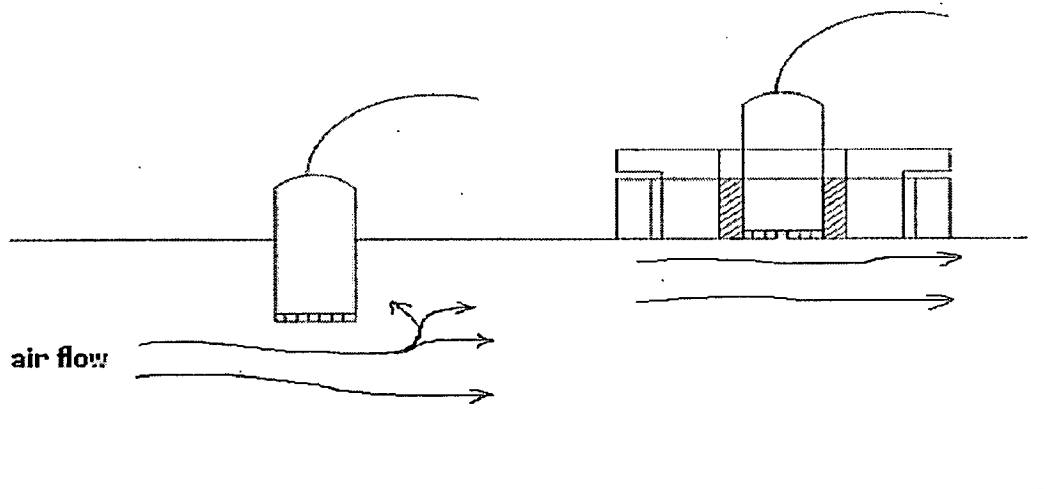


Fig. 3.6: Microphone Mounting Method to Reduce Flow Turbulence

3.5.2 TYPES OF ANC SYSTEMS

Broadband noise cancellation requires knowledge of the noise source (the primary noise) in order to generate the antinoise signal. The measurement of the primary noise is used as a reference input to the noise canceler. Primary noise that correlates with the reference input signal is canceled downstream of the noise generator (a loudspeaker)

when phase and magnitude are correctly modeled in the digital controller.

For narrowband noise cancellation (reduction of periodic noise caused by rotational machinery), active techniques have been developed that are very effective and that do not rely on causality (having prior knowledge of the noise signal). Instead of using an input microphone, a tachometer signal provides information about the primary frequency of the noise generator. Because all of the repetitive noise occurs at harmonics of the machine's basic rotational frequency, the control system can model these known noise frequencies and generate the antinoise signal. This type of control system is desirable in a vehicle cabin, because it will not affect vehicle warning signals, radio performance, or speech, which is not normally synchronized with the engine rotation.

Active noise control systems are based on one of two methods. Feed forward control is where a coherent reference noise input is sensed before it propagates past the canceling speaker. Feedback control is where the active noise controller attempts to cancel the noise without the benefit of an upstream reference input.

Feed forward ANC systems are currently preferred. Systems for feed forward ANC are further classified into two categories:

- Adaptive broadband feed forward control with an acoustic input sensor
- Adaptive narrowband feed forward control with a nonacoustic input sensor

3.5.2.1 The Broadband Feed forward System

A considerable amount of broadband noise is produced in ducts such as exhaust pipes and ventilation systems. A relatively simple feed forward control system for a long, narrow duct is illustrated in Fig 3.7. A reference signal $x(n)$ is sensed by an input microphone close to the noise source before it passes a loudspeaker. The noise canceller uses the reference input signal to generate a signal $y(n)$ of equal amplitude but 180° out of phase. This antinoise signal is used to drive the loudspeaker to produce a canceling sound that attenuates the primary acoustic noise in the duct through the destructive interference.

The basic principle of the broadband feed forward approach is that the propagation time delay between the upstream noise sensor (input microphone) and the active control source (speaker) offers the opportunity to electrically reintroduce the noise

at a position in the field where it will cause cancellation. The spacing between the microphone and the loudspeaker must satisfy the principles of causality and high coherence, meaning that the reference must be measured early enough so that the antinoise signal can be generated by the time the noise signal reaches the speaker. Also, the noise signal at the speaker must be very similar to the measured noise at the input microphone, meaning the acoustic channel cannot significantly change the noise. The noise canceller uses the input signal to generate a signal $y(n)$ that is of equal amplitude and is 180° out of phase with $x(n)$. This noise is output to a loudspeaker and used to cancel the unwanted noise.

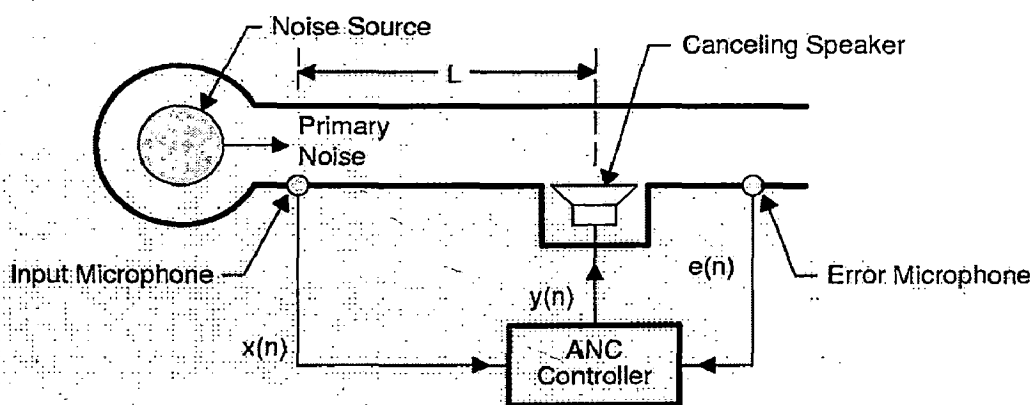


Fig. 3.7 Single-Channel Broadband Feedforward ANC System in a Duct

The error microphone measures the error (or residual) signal $e(n)$, which is used to adapt the filter coefficients to minimize this error. The use of a downstream error signal to adjust the adaptive filter coefficients does not constitute feedback, because the error signal is not compared to the reference input.

3.5.2.2 The Narrowband Feedforward System

In applications where the primary noise is periodic (or nearly periodic) and is produced by rotating or reciprocating machines, the input microphone can be replaced by a nonacoustic sensor such as a tachometer, an accelerometer, or an optical sensor. This replacement eliminates the problem of acoustic feedback

The block diagram of a narrowband feedforward active noise control system is

shown in Fig. 3.8 The nonacoustic sensor signal is synchronous with the noise source and is used to simulate an input signal that contains the fundamental frequency and all the harmonics of the primary noise. This type of system controls harmonic noises by adaptively filtering the synthesized reference signal to produce a canceling signal. In many cars, trucks, earth moving vehicles, etc., the revolutions per minute (RPM) signal is available and can be used as the reference signal. An error microphone is still required to measure the residual acoustic noise. This error signal is then used to adjust the coefficients of the adaptive filter.

Generally, the advantage of narrowband ANC systems is that the nonacoustic sensors are insensitive to the canceling sound, leading to very robust control systems. Specifically, this technique has the following advantages:

- Environmental and aging problems of the input microphone are automatically eliminated. This is especially important from the engineering viewpoint, because it is difficult to sense the reference noise in high temperatures and in turbulent gas ducts like an engine exhaust system.
- The periodicity of the noise enables the causality constraint to be removed. The noise waveform frequency content is constant. Only adjustments for phase and magnitude are required. This results in more flexible positioning of the canceling speaker and allows longer delays to be introduced by the controller.
- The use of a controller-generated reference signal has the advantage of selective cancellation; that is, it has the ability to control each harmonic independently.
- It is necessary to model only the part of the acoustic plant transfer function relating to the harmonic tones. A lower-order FIR filter can be used, making the active periodic noise control system more computationally efficient.
- The undesired acoustic feedback from the canceling speaker to the input microphone is avoided.

3.5.2.3 The Feedback ANC System

In this scheme, a microphone is used as an error sensor to detect the undesired noise. The error sensor signal is returned through an amplifier (electronic filter) with magnitude and phase response designed to produce cancellation at the sensor via a loudspeaker located near the microphone. This configuration provides only limited attenuation over a restricted frequency range for periodic or band-limited noise. It also suffers from instability, because of the possibility of positive feedback at high frequencies. However, due to the predictable nature of the narrowband signals, a more robust system that uses the error sensor's output to predict the reference input has been developed. The regenerated reference input is combined with the narrowband feed forward active noise control system.

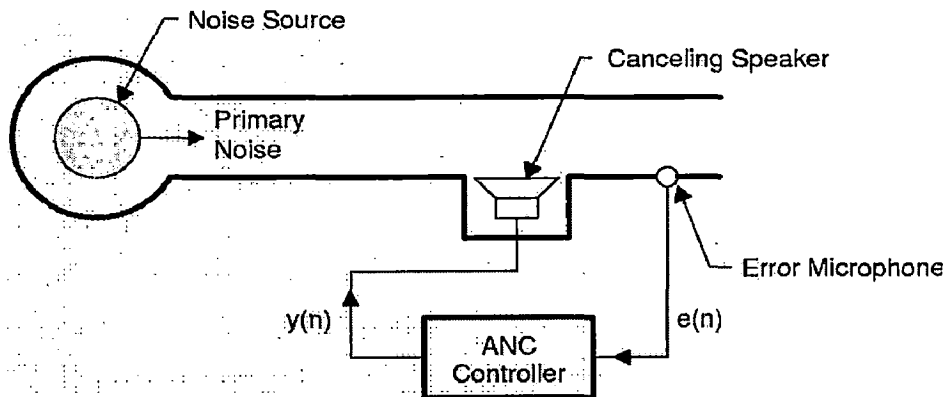


Fig. 3.8 Feedback ANC System

One of the applications of feedback ANC recognized by Olson is controlling the sound field in headphones and hearing protectors. In this application, a system reduces the pressure fluctuations in the cavity close to a listener's ear. This application has been developed and made commercially available.

The block diagram of a narrowband and feed forward active noise control system is shown in Fig. 3.9. The nonacoustic sensor signal is synchronous with the noise source and is used to simulate an input signal that contains the fundamental frequency and all the harmonics of the primary noise. This type of system controls harmonic noises by adaptively filtering the synthesized reference signal to produce a canceling signal. In many cars, trucks, earth moving vehicles, etc., the revolutions per minute (RPM) signal is available and can be used as the reference signal. An error microphone is still required to

measure the residual acoustic noise. This error signal is then used to adjust the coefficients of the adaptive filter.

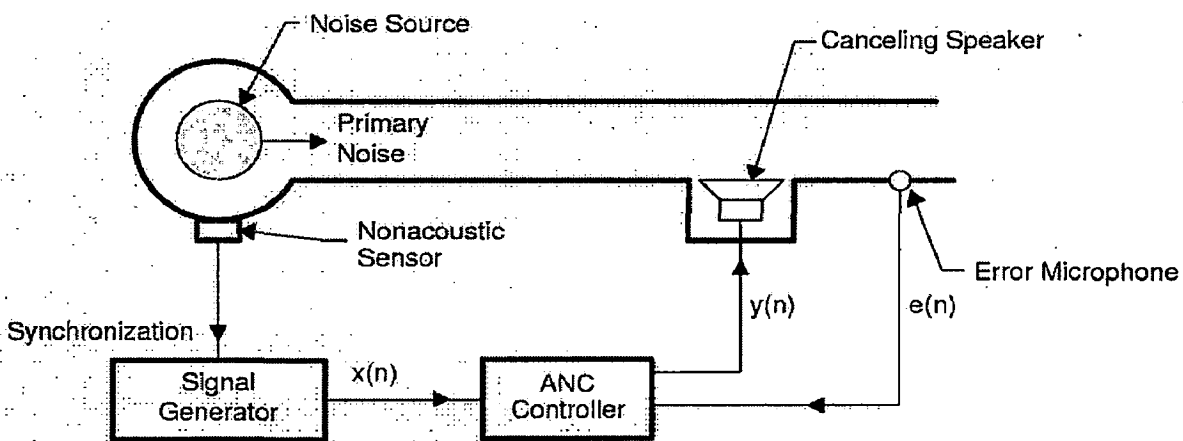


Fig 3.9: Narrowband Feed forward ANC System

3.5.2.4 The Multiple-Channel ANC System

Many applications can display complex modal behavior. These applications include:

- Active noise control in large ducts or enclosures
- Active vibration control on rigid bodies or structures with multiple degrees of freedom
- Active noise control in passenger compartments of aircraft or automobiles

When the geometry of the sound field is complicated, it is no longer sufficient to adjust a single secondary source to cancel the primary noise using a single error microphone. The control of complicated acoustic fields requires both the exploration and development of optimum strategies and the construction of an adequate multiple-channel controller. These tasks require the use of a multiple-input multiple-output adaptive algorithm. The general multiple-channel ANC system involves an array of sensors and actuators. A block diagram of a multiple-channel ANC system for a three-dimensional application is shown in Fig. 3.10.

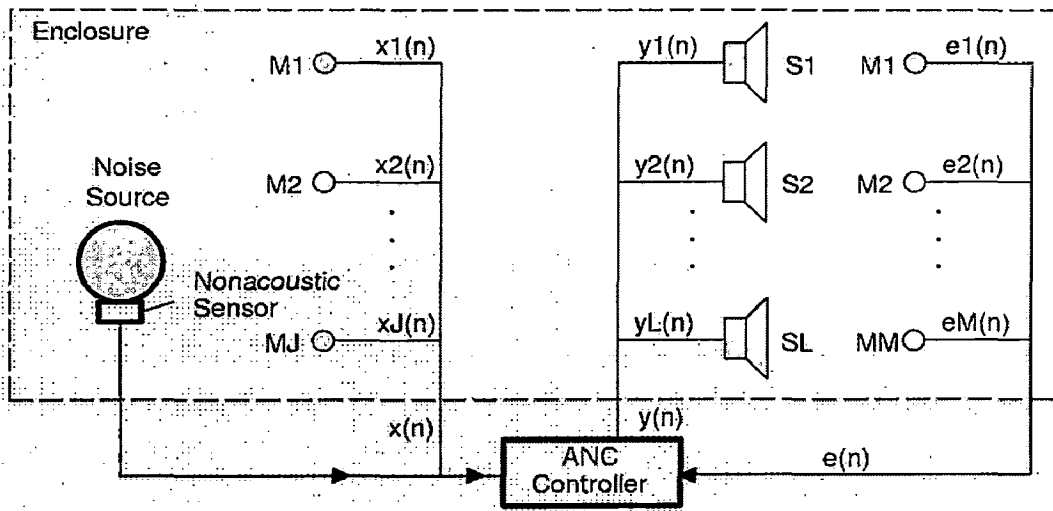


Fig. 3.10. Multiple-Channel ANC System for a 3-D Enclosure

CHAPTER 4

ADAPTIVE SIGNAL PROCESSING TECHNIQUES

An adaptive automation is a system whose structure is alterable or adjustable in such a way that its behavior or performance (according to some desired criterion) improves through contact with its environment. A simple example of an automation or automatic adaptive system is the automatic gain control (AGC) used in radio and television receivers. The function of this circuit is to do adjust the sensitivity of the receiver inversely as the average incoming signal strength.

Adaptive systems usually have some or all of the following characteristics:

- They can automatically adapt (self optimize) in the face of changing (non stationary) environments and changing system requirements.
- They can be trained to perform specific filtering and decision making tasks. Synthesis of systems having these capabilities can be accomplished automatically through training. In a sense, adaptive systems can be “programmed” by a training process.
- Because of the above, adaptive systems do not require the elaborate synthesis procedures usually needed for non adaptive systems. Instead, they tend to be “self designing”.
- They can extrapolate a model of behavior to deal with new situations after having been trained on a finite and often small number of training signals or patterns.
- To a limited extent, they can repair themselves; that is, they can adapt around certain kinds of internal defects.
- They can usually be described as nonlinear systems with time varying parameters.
- Usually, they are more complex and difficult to analyze than nonadaptive systems, but they offer the possibility of substantially increased system performance when input signal characteristics are unknown or time varying.

Current applications for adaptive systems are in such fields as communications, radar, sonar, seismology, mechanical design, navigation systems, and biomedical electronics.

4.1 Open and Closed Loop Adaptation

Several ways to classify adaptive schemes have been proposed in the literature. One can classify adaptive systems in terms of open-loop and closed-loop systems. The open-loop adaptive process involves making measurements of input or environmental characteristics, applying this information to a formula or to a computational algorithm, and using the results to set the adjustments of the adaptive systems. Closed-loop adaptation, on the other hand, involves automatic experimentation with these adjustments and knowledge of their outcome in order to optimize a measured system performance. The latter process is also called adaptation by “performance feedback”.

The principles of open and closed loop adaptation are illustrated in Fig. 4.1 and 4.2. In open-loop systems, characteristics of input signal and other data is used to adjust the processor and in closed-loop system output signal is also used. The processor is adjusted to keep its performance optimized according to preselected criterion. The other data in these figures may be data about the environment of the adaptive system, or in the closed loop case, it may be desired version of the output signal.

When designing adaptive process, many factors determine the choice of closed loop versus open loop adaptation .the availability of input signals and performance indicating signals is a major consideration. Also the amount of computing capacity and type of computer required to implement the open loop and closed loop adaptation algorithms will generally differ. Certain algorithms require the use of a general purpose digital computer, where as other algorithms could be implemented far more economically with special purpose chips or other apparatus.

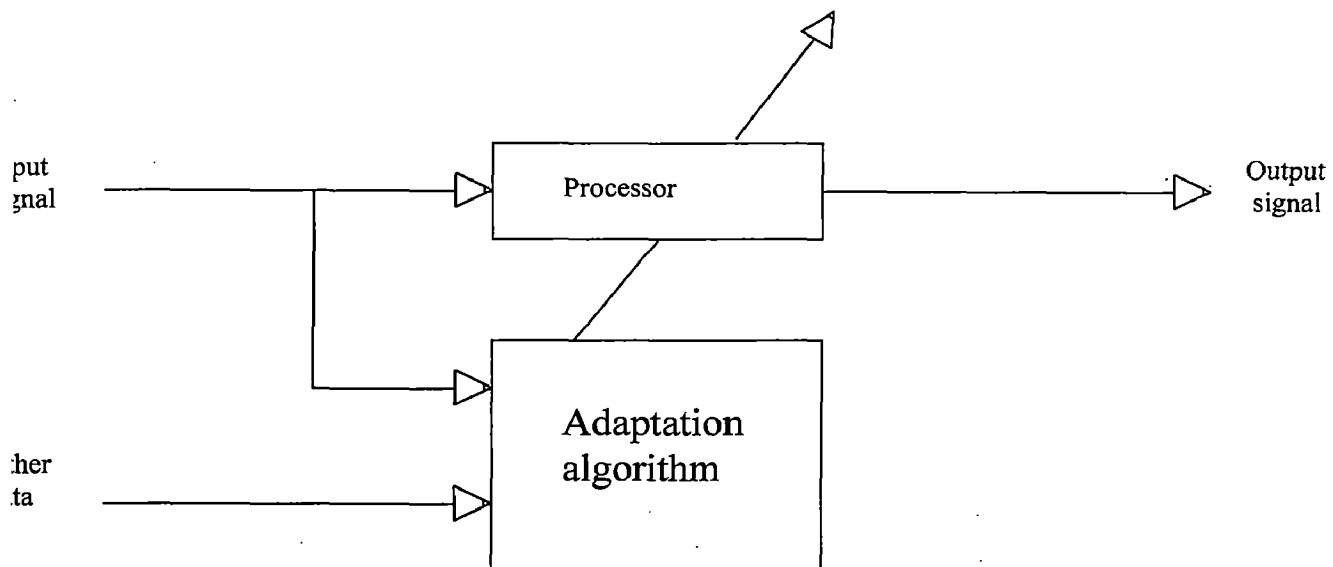


Fig. 4.1: Open loop adaptation

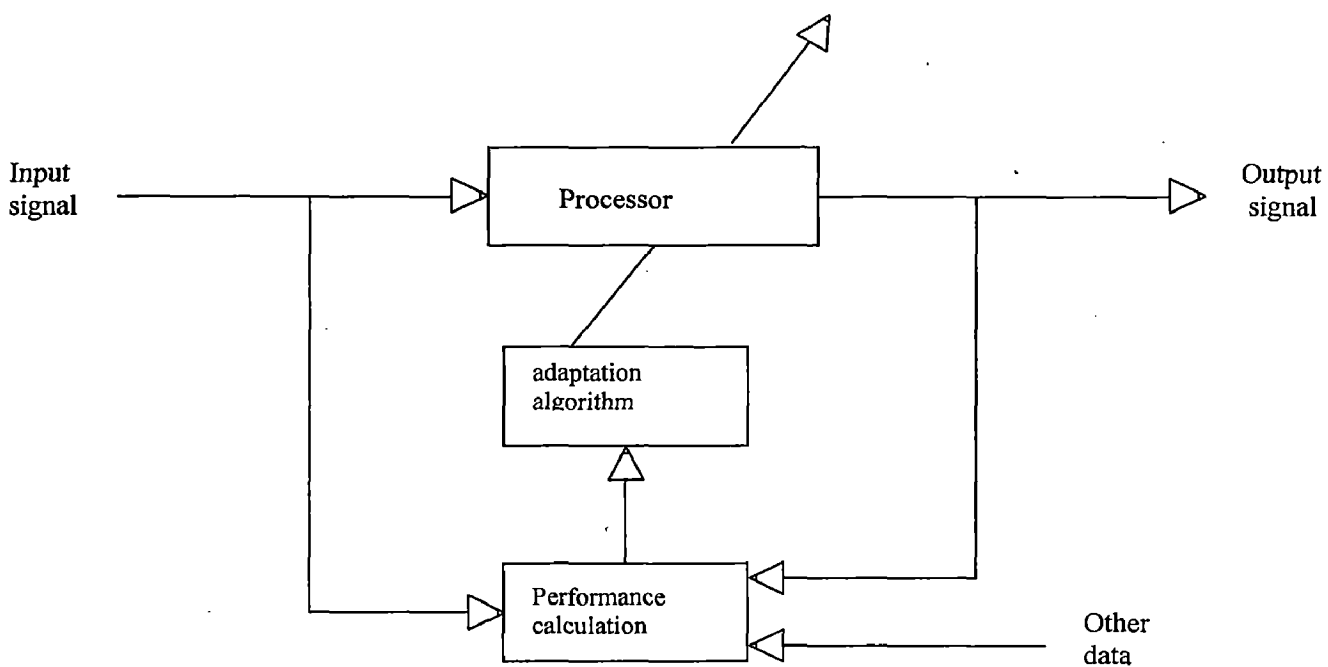


Fig 4.2: Closed loop adaptation

Closed loop adaptation has the advantage of being workable in many applications where no analytical synthesis procedure either exists or is known; for example, where error criteria other than mean square are used where systems are nonlinear or time variable, where signals are nonstationary, and so on. Closed loop adaptation can also be used effectively in situations where physical systems component values are variable or inaccurately known. Closed loop adaptation will find the best choice of component values. In the event of partial system failure, an adaptation mechanism that continually monitors the performance will optimize this performance by adjusting and reoptimizing the intact parts. As a result, system reliability can often be improved by the use of performance feedback.

The closed loop adaptation process is not always free of difficulties. In certain situations, performance functions do not have unique optima. Automatic optimization is an uncertain process in such situations. In other situations, the closed loop adaptation process, like a closed loop control system, could be unstable. The adaptation process could diverge rather than converge. In spite of these possibilities, performance feedback is a powerful, widely applicable technique for implementing adaptation

4.2 Application of Closed Loop Adaptation

The performance feedback process is represented more specifically in Fig. 4.3 we call the input signal x and define a desired response signal d , which is assumed to represent the desired output of the adaptive system. The desired d is, for our purpose here, the other data in Fig. 4.2

The error signal, ε , is the difference between the desired output signal and the actual output signal, y , of the adaptive system, thus altering its response characteristics by minimizing some measure of the error, thereby closing the performance loop. Some examples of the applications are given in Fig. 4.4.

The prediction application in Fig. 4.4(a) is perhaps the simplest of four. The desired signal is the input signal, s , and a delayed version of the latter is sent to the adaptive processor, which must therefore try to predict the current input signal in order to have y cancel d and drive signal, ε , toward zero.

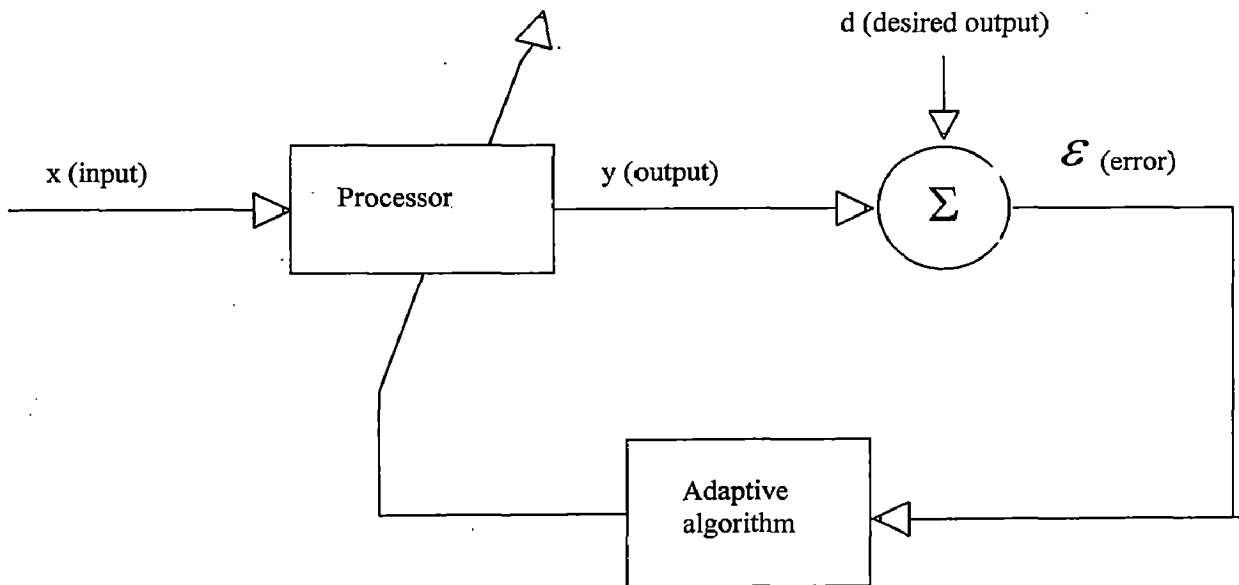
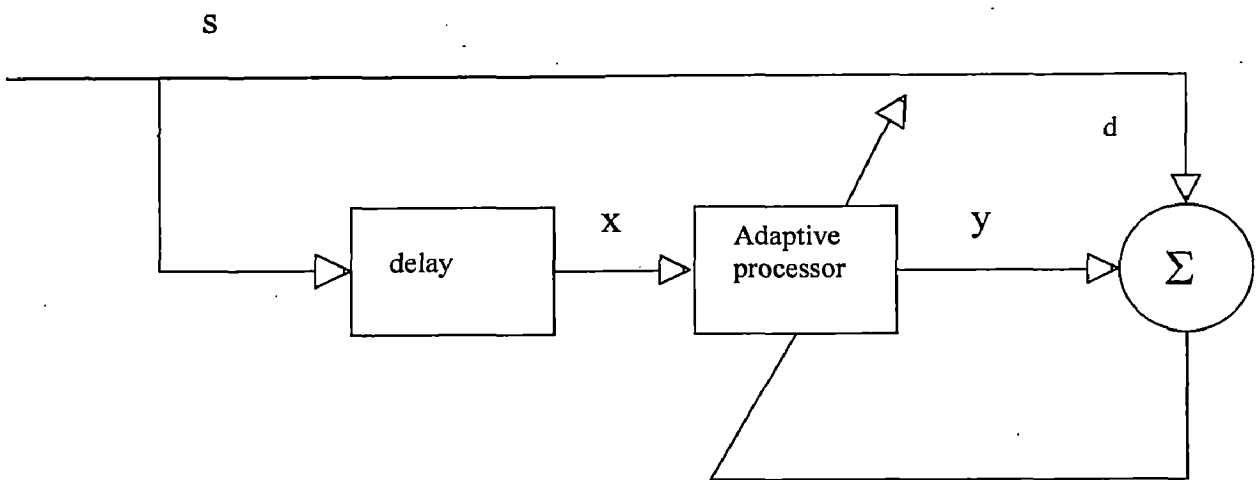
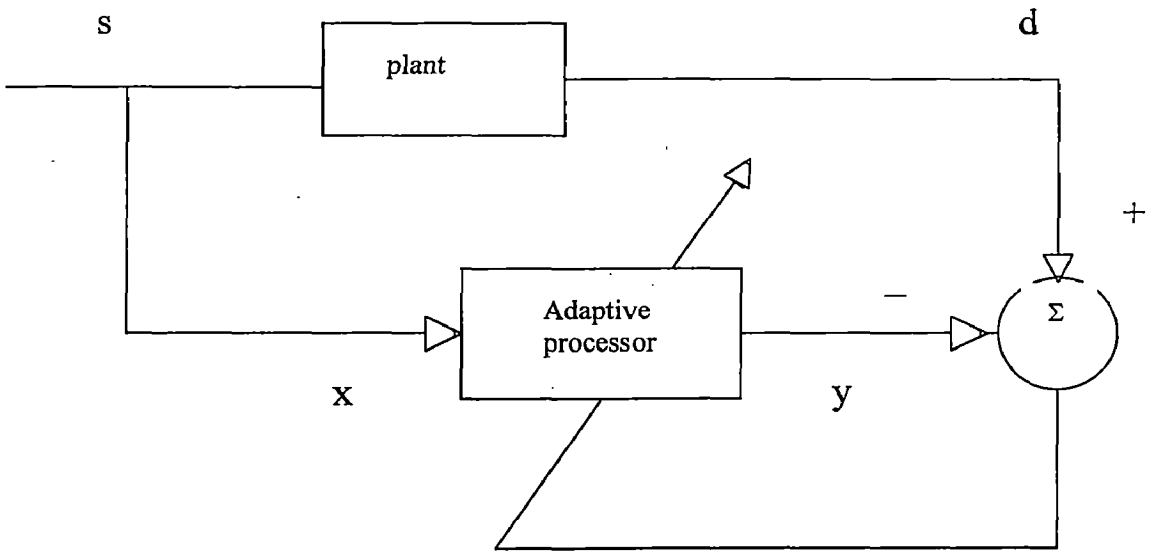


Fig 4.3: Signals in closed loop adaptation

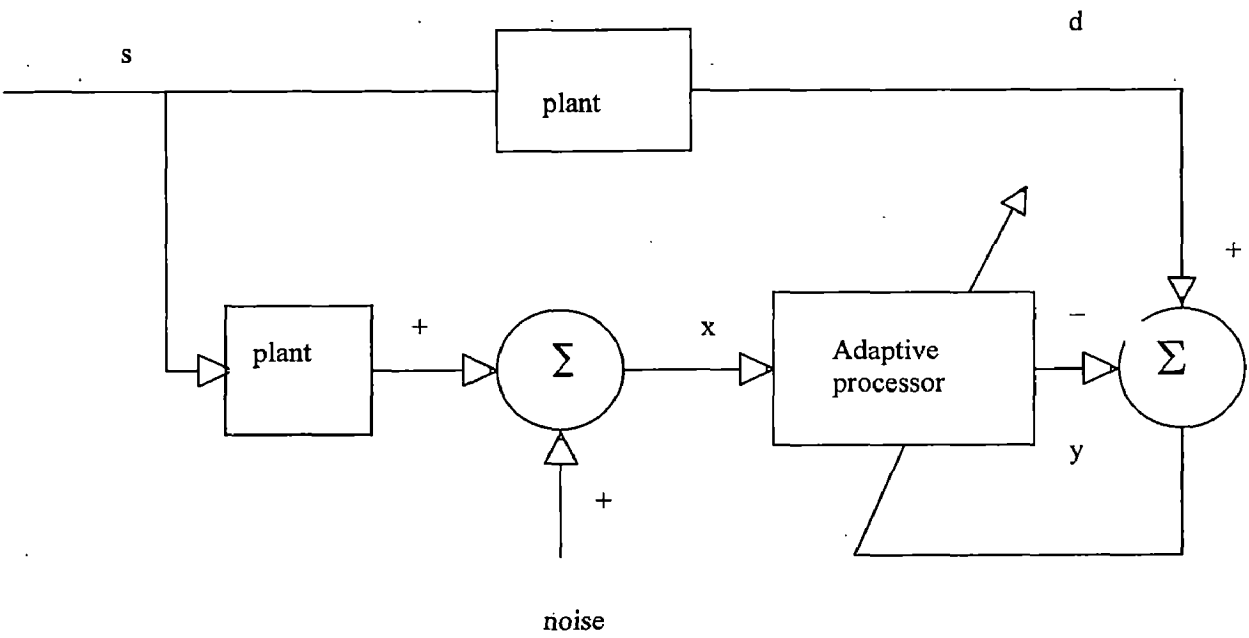
The system identification application in Fig. 4.4(b) is also easy to understand. Here a broadband signal, s , is the input to the adaptive processor as well as to an unknown plant. To reduce signal \mathcal{E} , the adaptive processor tries to emulate the plant's transfer characteristic. After adaptation the plant is identified in the sense that its transfer function can be specified as essentially the same as that of the adaptive processor.



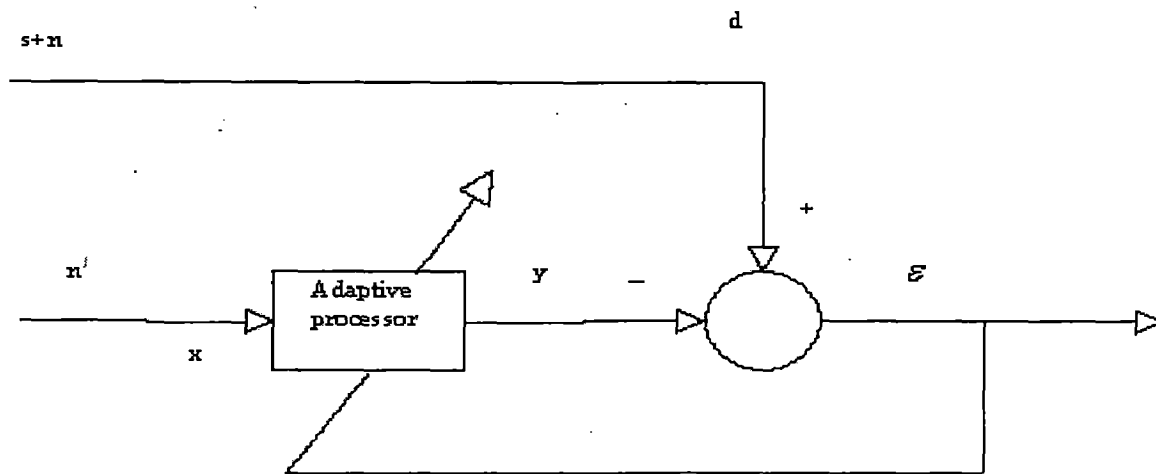
(a)



(b)



(c)



(d)

Fig 4.4 examples showing various configurations:

(a) prediction;(b) system identification;(c)equalization;(d)interference canceling;

Adaptive system identification or modeling can be used as such, to model a slowly varying plant whose input and output signals are available, for example, in vibration studies of mechanical systems.

In inverse modeling application the adaptive processor attempts to recover a delayed version of the signal, s , which is assumed to have been altered by the slowly varying plant and to contain additive noise. The delay in figure is to allow for the delay, or propagation time, through the plant and adaptive processor. Adaptive equalization could be used to undo the effects of a transducer, a communication channel, or some other system, or to produce an inverse model of an unknown plant. It is also applicable in the digital filters, as well as in adaptive control problems and so on.

Finally, Fig 4.4(d) shows the adaptive processor in an interference canceling configuration, here the signal, s , is corrupted by additive noise, and a distorted but correlated version of the noise, n_1 , is also available. The goal of the adaptive processor in this case is to produce an output, y , that closely resembles n_1 , so that the overall output ϵ will closely resemble s .

4.3 Optimal Transversal Filter

An ideal filtering problem is discussed here to introduce notation, properties, and analysis techniques that can be applied to adaptive transversal filters. The filtering problem of interest is that of a transversal filter whose output is a weighted summation of delayed values of an input time series, shown schematically in Fig. 4.5.

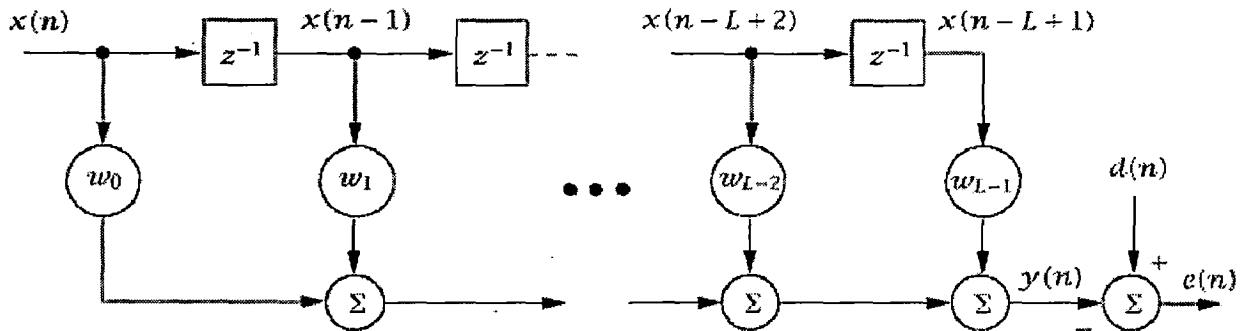


Fig. 4.5 General transversal filtering problem

This schematic representation, where Z^{-1} is a unit delay operator, is a well-accepted form for representing a finite impulse response (FIR) filter. The input signal to the filter consists of L tap-delay, or consecutively sampled values of a time series x , denoted as the $(L \times 1)$ vector $x(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ where n denotes the time index and T denotes transpose (for clarity, matrices, vectors, and scalars will be represented by upper-case bold, lower-case bold, and regular face letters, respectively). The tap inputs $x(n)$ are weighted by a time-invariant, $(L \times 1)$ vector of tap weights, $w = [w_0, w_1, \dots, w_{L-1}]^T$. The time invariance of w will be relaxed when adaptive algorithms are discussed. The filter output at time n is denoted as $y(n)$ and is the sum of the weighted input values. A desired filter output, $d(n)$, is also known at each time step, and the error between the desired and actual filter outputs at time n is defined as

$$e(n) = d(n) - y(n) \quad \dots \quad 4.1$$

$$= d(n) - w^T x(n) \quad \dots \quad 4.2$$

where the second equation is written using vector notation.

An infinite impulse response (IIR) filter could also be used to filter $x(n)$ and produce an output $y(n)$ for comparison with the desired, $d(n)$. However, the FIR filter in the figure is preferred due to its inherent stability which is especially important when an adaptive algorithm is used to modify the coefficients in the weight vector w .

The filtering problem is to determine the tap weight vector w that minimizes the difference in a mean square sense between the actual and desired filter outputs. The mean square criterion is used here because the resulting filtering problem requires solution of a quadratic function of the weight vector w , and a quadratic performance criterion is smoothly varying and has a unique global minimum. Using $E[\]$ to denote statistical expectation, the mean square error criterion can be written as a function of the tap weight vector as

$$J = E[e(n)^2] \tag{4.3}$$

$$= E[(d(n) - w^1 x(n))^2] \tag{4.4}$$

$$= E[d^2(n)] - 2w^1 E[x(n)d(n)] + w^1 E[x^1(n)x(n)]w \tag{4.5}$$

$$= E[d^2(n)] - 2w^1 p + w^1 R w \tag{4.6}$$

where $p = E[x(n)d(n)]$ is the $(L \times 1)$ cross correlation vector between the input and desired time series, and $R = E[x'(n)x(n)]$ is the $(L \times L)$ correlation matrix of the input time series. The stationarity assumption means R and p are constant with time. The symbol J will be used here to represent the function to be optimized, which is usually called a cost function. The last term in Eq. 4.6 depends on the square of w , hence the cost function varies quadratically with w . Assuming the correlation matrix R is positive definite, there is a weight vector that corresponds to a minimum cost; this weight vector will be denoted w_{opt} . If the input time series contains at least $(L/2)$ frequency components, the correlations between the tap-delayed values in x will be significant, and the resulting correlation matrix R will be full rank.

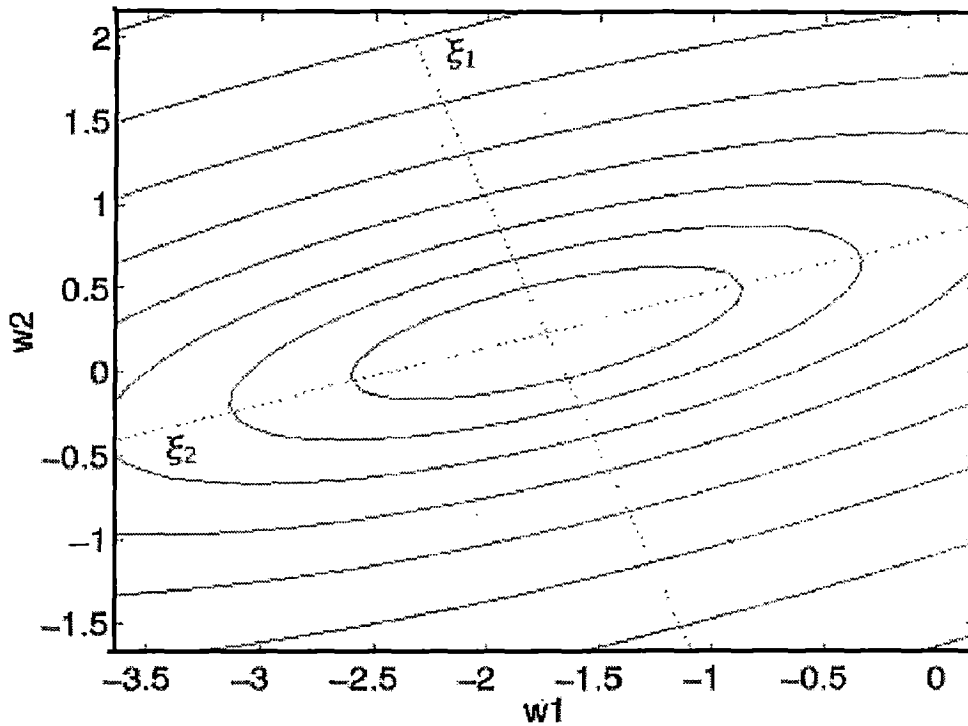


Fig 4.6: Contour plot of error surface

The expectations used to determine R and p are taken over many ensembles of data from the random processes x and d , hence R and p are called ensemble average properties. The optimal weight vector defined in Eq. 4.6 is therefore an ensemble average optimum. In almost any practical application of the techniques described here the ensemble average values of R and p will be unknown. For this reason the solution in Eq. 4.6 represents an ideal solution that will be used to measure the effectiveness of more practical solution algorithms.

It is helpful to envision the cost function as a multidimensional bowl-shaped surface in a space defined by the elements of the tap weight vector w . A plot of a hypothetical cost function, J , defined by two tap weights, w_1 and w_2 , is shown in Fig. 4.6. The elliptical curves define contours of equal cost that increase away from the minimum cost at the center of the plot, and the principal axes of the ellipses are shown as dotted lines labeled ξ_1 and ξ_2 . Assuming the correlation matrix R is positive definite, the opening of the bowl-shaped function points upward. The coordinates defining the center of the plot in the figure, which corresponds to the minimum of the cost surface, define the elements of the optimal tap weight vector, w_{opt} . The principal axes and eccentricity of the

elliptical contours are functions of the eigenvectors and eigenvalues, respectively, of R . The steepness of the error surface in the direction of the i^{th} principal axis is proportional to the i^{th} eigenvalue of R , and therefore the elongation of the bowl-shape can be quantified by the condition number, or ratio of the largest to the smallest eigenvalue, of R . The condition number is greater than or equal to one, and if it equals one, the contours of equal cost are circles. For the surface shown in the plot, the condition number of R was 10.0, hence the equal cost contours have an elliptical shape.

The tap weight vector corresponding to the minimum cost can be found by differentiating J with respect to w and setting the derivative to zero since this stationary point is known to be a minimum. Differentiating J yields

$$\nabla J = -2(p - R w) \quad \dots\dots\dots 4.7$$

If R is positive definite and thus invertible, the tap weight vector that minimizes J is given by

$$w_{opt} = R^{-1} p \quad \dots\dots\dots 4.8$$

Therefore knowing the correlation matrix R of the input signal and the cross correlation p between the input and desired signals, a unique optimal weight vector can be computed from Eq. 4.8. Although the times series are assumed to be stationary, in many practical applications the statistics could change slowly with time, hence the optimal tap weight vector would also change with time. For these reasons, adaptive algorithms are often used to determine the optimal tap weight vector.

4.4 Steepest Descent Algorithm

The adaptive algorithms of interest here are based on gradient descent techniques, of which the steepest descent algorithm is one example. The steepest descent algorithm relies on a recursive equation that moves an arbitrary tap weight vector towards the optimum tap weight vector using the negative of the gradient of the cost function. The steepest descent algorithm is ideal in the sense that the exact gradient vector is assumed to be known at each iteration. The behavior of more practical algorithms can be easily related to the behavior of the steepest descent algorithm, hence the properties of this

algorithm will be discussed in some detail. In this discussion, time invariance of the tap weights is no longer assumed, and the computations described here represent iterations computed at each time step.

The weight update recursion for the steepest descent algorithm specifies that the tap weight vector at iteration $(n + 1)$ is computed from the weight vector at the previous iteration plus a correction based on the negative of the gradient of the cost function. This recursion is written

$$w(n+1) = w(n) + \frac{\mu}{2}[-\nabla J(n)] \quad \dots\dots\dots 4.9$$

where $\nabla J(n)$ denotes the vector gradient of the cost function with respect to each element of the tap weight vector. The symbol μ denotes a step size parameter that controls how much the weight vector is changed at each iteration. The algorithm essentially relies on the idea that the tap weight vector is slowly adjusted toward its optimal value that minimizes the cost function. This approach can be thought of as starting from an arbitrary point on the cost surface and sliding down the surface in its steepest direction to the minimum. The convergence of w to the optimum w_{opt} is assured as long as the step size parameter μ satisfies a few simple constraints discussed later.

$$\nabla J(n) = -2(p - R w(n)) \quad \dots\dots\dots 4.10$$

Substituting this result into Eq.4.9 yields

$$w(n+1) = w(n) + \mu(p - R w(n)) \quad \dots\dots\dots 4.11$$

The path in L dimensional space taken by the tap weight vector during convergence is of particular interest. Because the steepest descent algorithm changes the weight vector according to the gradient of the cost surface, it is reasonable to assume the shape of that surface will determine the convergence behavior of w . This hypothesis is confirmed with a few manipulations of the weight update relation in Eq. 4.11, and using the fact that $p = R w_{opt}$ (Eq. 4.8) to produce

$$w(n+1) = w(n) + \mu(R w_{opt} - R w(n)) \quad \dots\dots\dots 4.12$$

$$= w(n) + \mu R(w_{opt} - w(n)) \quad \dots\dots\dots 4.13$$

A weight error vector is defined as $\varepsilon(n) = w(n) - w_{opt}$, which describes the deviation of $w(n)$ from the optimal vector at iteration n . Subtracting the optimal weight vector from both sides of Eq. 4.13 and substituting ε yields

$$w(n+1) - w_{opt} = w(n) - w_{opt} - \mu R(w_{opt} - w(n)) \quad \dots\dots\dots 4.14$$

$$\varepsilon(n+1) = (I - \mu R)\varepsilon(n) \quad \dots\dots\dots 4.15$$

where I is the identity matrix. This is a difference equation describing the behavior of the weight error vector as a function of the iterative step number, n . Further simplification is possible by expressing the correlation matrix R in terms of its eigenvectors and eigenvalues. Specifically, a non-defective (i.e. distinct eigenvalues) square matrix can be diagonalized with a similarity transformation $R = V \Lambda V^{-1}$ where the columns of V contain the eigenvectors of R , and Λ is a diagonal matrix containing the eigenvalues of R . The i th eigenvalue is denoted λ_i . Substituting for R in Eq. 4.15 produces

$$\varepsilon(n+1) = (I - \mu V \Lambda V^{-1})\varepsilon(n) \quad \dots\dots\dots 4.16$$

$$= V (I - \mu \Lambda) V^{-1} \varepsilon(n) \quad \dots\dots\dots 4.17$$

At this point it is helpful to introduce a new symbol $\xi(n) = V^{-1} \varepsilon(n)$, so the expression can be written

$$\xi(n+1) = (I - \mu \Lambda)\xi(n) \quad \dots\dots\dots 4.18$$

The variable ξ represents a transformation of the weight error vector, ε , to the principal coordinates, or principal components, of the adaptive algorithm. The transformation of the control problem to its principal coordinates is often used to analyze the convergence behavior of an adaptive algorithm. This is a useful form for studying the convergence because the matrix Λ is diagonal, hence the principal components, or modes, of the control system are decoupled and converge independently of one another. A recursion relation can therefore be written to describe the behavior of the i^{th} mode as

$$\xi_i(n+1) = (I - \mu \lambda_i)\xi_i(n) \quad \dots\dots\dots 4.19$$

Thus the convergence behavior of the weight vector in the steepest descent algorithm is determined by the eigenproperties of the matrix R .

Eq. 4.19 can be used to bound the step size parameter, μ . Consider the convergence behavior of the i th principal coordinate ξ_i , whose initial value at time $n = 0$ is denoted by $\xi_i(0)$. Solving Eq. 4.19 for $\xi_i(n)$ gives the solution.

$$\xi_i(n) = (1 - \mu\lambda_i)^n \xi_i(0) \quad \dots\dots\dots 4.20$$

The algorithm will converge if each transformed error term ξ_i goes to zero as n goes to infinity, which will occur as long as $|1 - \mu\lambda_i| < 1$ for all eigenvalues. Because the matrix R is positive definite, its eigenvalues are all greater than zero, hence the constraint on μ is satisfied only if

$$0 < \mu < \frac{2}{\lambda_i} \text{ for all } i \quad \dots\dots\dots 4.21$$

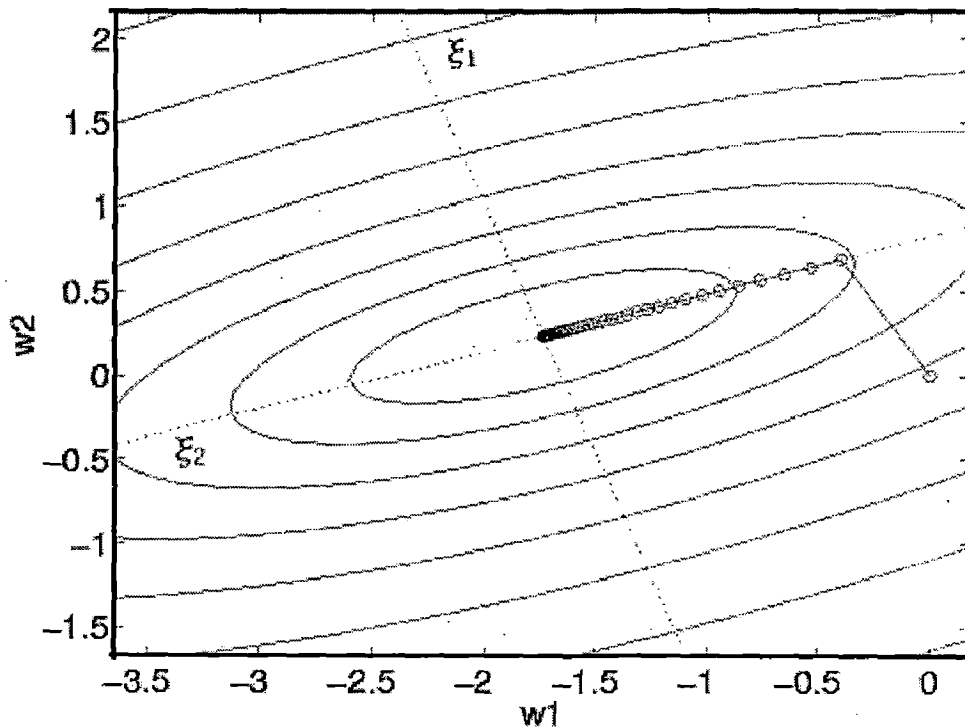


Fig 4.7 Convergence behavior of weight vector

The upper bound on μ is given by $\mu < (2/\lambda_{\max})$, where λ_{\max} is the largest eigenvalue of R . Assuming μ is bounded by the largest eigenvalue, ξ_i will decrease with time and

will be contained by an exponential envelope. The time constant for the decay of the i^{th} mode is

$$\tau_i = \frac{-1}{\ln(1 - \mu\lambda_i)} \dots\dots\dots 4.22$$

The time constant decreases if either μ or λ_i , increases. Therefore the principal coordinates with large eigenvalues will converge more quickly than principal coordinates with small eigenvalues. This gives rise to the terminology "fast" and "slow" modes of convergence to describe the convergence behavior of the tap weight vector w in the steepest descent algorithm. The convergence speed of the slow modes can be sped up by increasing the step size μ ; however, the maximum step size is constrained by the maximum eigenvalue as shown above. If the ratio of the largest to the smallest eigenvalue of R is very large, the convergence speed of the algorithm could be severely limited by the convergence of the slow modes. On the other hand, if the condition number of R is unity, ideal convergence behavior can be obtained because there are no slow modes of convergence.

The convergence of a tap weight vector that is adjusted using the steepest descent algorithm is shown on a hypothetical error surface in Fig. 4.6. This is the same error surface that was shown earlier in Fig. 4.5, and the correlation matrix used to generate the error surface had a condition number of 10.0. The value of the weight vector at each iteration during convergence is indicated by a circle, and a line connecting the circles defines the convergence path of w . The path starts at the right-hand side of the plot, where $w_1 = w_2 = 0$, and the final value of the weight vector is at the center of the plot where the cost function has its minimum. For this example, the step size μ was set to one half of its maximum theoretical value, or $\mu = 1/\lambda_{\max}$. The path of the weight vector initially moves in the steepest direction, nearly parallel to the first principal axis, ξ_1 . Once the cost function is minimized in this direction, the weight vector begins to move in the direction of the second principal axis, ξ_2 , eventually reaching the cost function minimum. If μ had been set to a larger value, the convergence path would have resembled a damped sine wave with oscillations about the path shown in the plot. If μ had been set to a smaller value, convergence would have taken longer and the path in the

figure would be smoother. The convergence behavior of w can thus be described as being underdamped if $1/\lambda_{\max} < \mu < 2/\lambda_{\max}$, critically damped if $\mu = 1/\lambda_{\max}$ as shown in the figure, or overdamped if

$$0 < \mu < 1/\lambda_{\max} \quad \dots\dots\dots 4.23$$

This discussion illustrates a fundamental weakness of the steepest descent algorithm that carries over to related algorithms: the convergence speed is dependent on the eigenvalue spread of the input correlation matrix R . The maximum value of the step size parameter μ is determined by λ_{\max} , but the convergence time is almost always limited by λ_{\max} , so when R has a large condition number the convergence time can be excessively long. The only time the slowest mode will not limit the convergence speed is if the tap weight vector happens to be initialized on a principal axis of one of the fast modes of R , in which case the convergence path would always be orthogonal to the principal axis of a slow mode. Because R describes the statistics of the input signal, the convergence is said to be dependent on the statistics of the input signal.

For the SISO system pictured in Fig. 4.1, R describes the correlation between the tap inputs. If the input data is white, the tap inputs are uncorrelated, and R is diagonal with a condition number of 1. The convergence of the steepest descent algorithm will be optimal in this case, since all modes converge at the same rate. As the dynamic range of the input signal increases and certain spectral components have more power than others, the condition number of R will increase, causing a corresponding decrease in the convergence speed of a steepest descent algorithm. In fact, the minimum and maximum eigenvalues of the correlation matrix of a discrete time stochastic process are bounded by the minimum and maximum values of the power spectral density of the process. An input signal with a large dynamic range could present problems for a steepest descent-type of algorithm. Later in the chapter a few methods will be described that seek to improve the convergence speed of gradient descent algorithms by pre-whitening a non-white input signal.

4.5 LMS algorithm

The steepest descent algorithm is a convenient way to solve for the optimal tap weight vector, but it requires knowledge of the exact gradient at each time step. A more

practical algorithm is the least mean square (LMS) algorithm, which uses the instantaneous estimate of the gradient at each time. Many practical implementations of adaptive filters use the LMS algorithm, in part because it is easy to program, debug, and understand. Principal component control is derived from the LMS algorithm, hence it is worth describing some of the details of the algorithm here.

The steepest descent algorithm is called a deterministic gradient descent algorithm because the true gradient is known at each time step, whereas the LMS algorithm is called a stochastic gradient descent algorithm because it relies on an estimate of the gradient. The LMS algorithm can be easily derived from the weight update equation for steepest descent, given earlier as

$$w(n+1) = w(n) + \mu(p - Rw(n))$$

where $(p - Rw(n))$ is the deterministic gradient of the cost function. An equivalent form of this equation is obtained by rearranging the expected value of Eq. 4.1 as follows:

$$E[e(n)x^1(n)] = E[d(n)x^1(n)] - E[w^1 x(n)x^1(n)] \quad \dots\dots\dots 4.24$$

$$= p^1 - w^1 R \quad \dots\dots\dots 4.25$$

$$E[x(n)e(n)] = p - Rw \quad \dots\dots\dots 4.26$$

The instantaneous approximation to the left hand side is $(x(n)e(n))$. Substituting this estimate for the gradient into the weight update equation yields

$$w(n+1) = w(n) + \mu x(n)e(n) \quad \dots\dots\dots 4.27$$

which is the weight update equation for the LMS algorithm. The simplicity of this algorithm is apparent, since at each time step the weight update is dependent only on the current input vector $x(n)$ multiplied by the current error, $e(n)$.

The stochastic gradient makes the LMS algorithm very easy to implement but unfortunately it complicates an analysis of the convergence behavior of the algorithm. Various discussions of the convergence behavior of the LMS algorithm, given with varying degrees of mathematical rigor, can be found in the references. Fortunately, because the LMS algorithm is closely related to the steepest descent algorithm, an understanding of the general behavior of the LMS algorithm can be obtained by comparison with steepest descent. In particular, the average behavior of the weight error vector in the LMS algorithm resembles the behavior of the weight error vector in the

steepest descent algorithm, which was shown earlier in Eq. 4.15. The two algorithms are similar only on an average basis, because the convergence path taken by a single realization of the LMS algorithm will look like a noisy version of the convergence path for steepest descent. If one were to average many such convergence paths for the LMS algorithm the result would resemble the path taken by the steepest descent algorithm. Just as the convergence of the weight vector for the steepest descent algorithm was shown to be given by a combination of decaying exponential terms each with its own time constant of decay, the convergence of the LMS algorithm is given by a summation of noisy decaying exponential terms. This means the convergence behavior of the LMS algorithm depends on the statistics of the input signal, just as was true for the steepest descent algorithm, and thus one can speak of "fast" and "slow" modes of convergence. Each mode corresponds to a different frequency component of the tap-delay input vector, x .

To maintain stable convergence of the LMS algorithm, the step size parameter must be bounded by constraints similar to those derived for the steepest descent algorithm. The references given different values for the precise bounds on μ , which is perhaps an indication of the difficulty in rigorously analyzing the convergence behavior of a stochastic gradient algorithm. The step size parameter must be bounded just as for steepest descent,

$$0 < \mu < \frac{2}{\lambda_{\max}} \dots\dots\dots 4.28$$

where λ_{\max} is the maximum eigenvalue of the correlation matrix R . A more practical approach, based on the reasoning that λ_{\max} is not generally known, is to bound μ by

$$0 < \mu < \frac{2}{\text{tr}(R)} \dots\dots\dots 4.29$$

where $\text{tr}(R)$ is the trace of R , i.e. the sum of the eigenvalues of R . The trace of R is approximately equal to the power of the input signal x times the number of filter coefficients.

The noisy gradient in the LMS algorithm prevents the final value of the tap weight vector from converging to the optimal weight vector, and instead w oscillates about the optimum value. The magnitude of the oscillations is called the convergence

misadjustment, and is proportional to the step size parameter μ . A high value of μ produces fast convergence but higher misadjustment once converged, while a small μ produces less final misadjustment at the expense of increased convergence time.

CHAPTER 5

INTRODUCTION TO SIGNAL PROCESSOR

5.1 Signal Processor Fundamentals

The TMS320C6711 is a Texas Instruments DSP belonging to the TMS320 family of DSPs. The C6711 DSP is a floating point signal processor with 167 MHz clock speed and supports up to eight 32 bit instructions per cycle. It supports the VelociTI architecture which provides very long instruction words. The core CPU consists of 32 general purpose registers of 32 bit word length and eight functional units: two multipliers and six ALUs. The C62xIC67x family has the following features:

- Peak 1336MIPS at 167MHz.
- Peak 1G FLOPS at 167 MHz for single-precision operations.
- Peak 250M FLOPS at 167 MHz for double-precision operations.
- Peak 688M FLOPS at 167 MHz for multiply and accumulate operations.
- Hardware support for single-precision (32-bit) and double-precision (64-bit) IEEE floating-point operations.
- 32 - 32-bit integer multiplication with 32- or 64-bit result.
- A variety of memory and peripheral options are available for the C62xIC67x.
- Large on-chip RAM for fast algorithm execution.
- 32-bit external memory interface supports SDRAM, SBSRAM, SRAM, and other asynchronous memories, for a broad range of external memory requirements and maximum system performance.
- Host port access to 'C62xIC67x memory and peripherals.
- Multi channel DMA controller.
- Multi channel serial port(s).

5.2 On Board Peripherals Used

The C6000 families of processors support a number of on board peripherals that are supplied with C6000 device. These include EDMA Controller, AD535 A/D and D/A converter and MCBSP which are used in our application.

5.2.1 EDMA Controller

The enhanced direct memory access (EDMA) controller, like the DMA controller, transfers data between regions in the memory map without intervention by the CPU. The EDMA allows movement of data to and from internal memory, internal peripherals, or external devices to occur in the background of CPU operation. The EDMA has sixteen independently programmable channels allowing sixteen different contexts for operation. In addition to the features of the DMA controller, the EDMA also has the following features:

- Sixteen channels: The EDMA can keep track of the contexts of sixteen independent transfers
- Linking: Each EDMA channel can be linked to a subsequent transfer to perform after completion.
- Event synchronization: Each channel is initiated by a specific event. Transfers may be either synchronized by element or by frame.

5.2.2 Multi channel Buffered Serial Port (McBSP)

The 'C62x/C67x multi channel buffered serial port (McBSP) is based on the standard serial port interface found on the TMS320C2000 and 'C5000 platforms. The standard multi channel buffered serial port interface provides:

- Full-duplex communication
- Double-buffered data registers, which allow a continuous data stream
- Independent framing and clocking for reception and transmission
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices
- External shift clock generation or an internal programmable frequency shift clock
- Multichannel transmission and reception of up to 128 channels.
- A wider selection of element sizes including 8-, 12-, 16-, 20-, 24-, or 32-bit
- μ -Law and A-Law companding.
- 8-bit data transfers with LSB or MSB first

- Programmable polarity for both frame synchronization and data clocks
- Highly programmable internal clock and frame generation.
- The McBSP consists of a data path and control path. Seven pins connect the control and data paths to external devices.

Data is communicated to devices interfacing to the McBSP via the data transmit (DX) pin for transmission and the data receive (DR) pin for reception. Control information in the form of clocking and frame synchronization is communicated via CLKX, CLKR, FSX, and FSR. The peripheral device communicates to the McBSP via 32-bit-wide control registers accessible via the internal peripheral bus. The CPU or DMA controller reads the received data from the data receive register (DRR) and writes the data to be transmitted to the data transmit register (DXR). Data written to the DXR is shifted out to DX via the transmit shift register (XSR). Similarly, receive data on the DR pin is shifted into the receive shift register (RSR) and copied into the receive buffer register (RBR). RBR is then copied to DRR, which can be read by the CPU or the DMA controller. This allows internal data movement and external data communications simultaneously. The remaining registers accessible to the CPU configure the control mechanism of the McBSP.

5.2.3 TLC320AD535 On-board A/D converter

The TLC320AD535 dual channel voice/data codec is a mixed-signal broadband connectivity device. The TLC320AD535 is comprised of a two-channel codec and analog hybrid circuitry with two independent serial ports for communication with the host processor and external resistors and capacitors for setting gain and filter poles. The device also contains microphone bias and amplification, audio mixing capabilities in the voice channel, programmable gain control, and three (SPKR_LEFT, SPKR_RIGHT, and MONOUT) speaker drivers. The device operates with either 5-V analog, 5-V digital, and 5-V monitor power supply or 3.3-V analog, 3.3-V digital and 5-V monitor power supply or 5-V analog, 3.3-V digital, and 5-V monitor power supply. It is available in a single 64-pin PM (QFP) package. It features:

- Analog, Digital, and Monitor Amp Power Supplies: 5 V or 3.3 V
- Separate Software Power-Down Modes for Data and Voice Channels

- Independent Voice and Data Channel Sample Rates up to 11.025 kHz
- 16-Bit Signal Processing
- Dynamic Range of 80 dB in the Data and Voice Channels
- Total Signal-to-Noise + Distortion of 77 dB for the ADCs
- Total Signal-to-Noise + Distortion of 74 dB for the DACs
- Programmable Gain Amplifiers
- 600-W TAPI Audio and Data Channel Drivers
- 60-W Headphone Driver With Programmable Gain Amplifier.
- 8-W AT41 Differential Speaker Driver With Programmable Gain Amplifier
- Maximum Microphone Bias of 5 mA at 2.5 V/1.5 V
- Maximum Handset Reference of 2.5 mA at 2.5 V/1.5 V
- Maximum Data Channel Reference of 10 mA at 2.5 V/1.5 V
- 5-V MVDD Power Reset Circuit
- Flash Write Enable Circuit, for Writing the Flash Memory Device
- Available in a 64-Pin PM (QFP) Package Operating From -40°C to 85°C

The codec portion of the TLC320AD535 device performs the functions required for two channels of analog-to-digital conversion, digital-to-analog conversion, low pass filtering, control of analog input and output gains, internal over sampling coupled with internal decimation and interpolation, and two 16-bit serial port interfaces to the host processor. The two serial ports operate independently and are capable of operating at sample rates. The maximum sample rate of either codec channel is 11.025 kHz. For signal data transmitted from the ADC or to the DAC, a primary serial communication is used. A secondary communication reads or writes words to the control registers, which control both the options and the circuit configurations of the device.

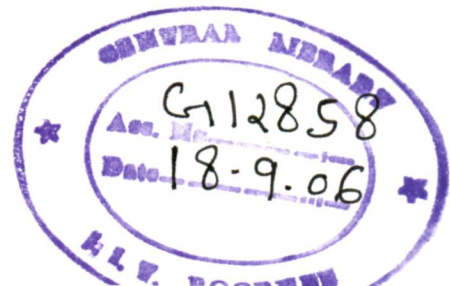
5.3 DSP/BIOS Programming Techniques

DSP/BIOS is a scalable real-time kernel. It is designed for applications that require realtime scheduling and synchronization, host-to-target communication, or real-time instrumentation. DSP/BIOS provide preemptive multi-threading, hardware abstraction, real-time analysis, and configuration tools. The DSP/BIOS provides with the following facilities to the programmer:

- A program can dynamically create and delete objects that are used in special situations. The same program can use both objects created dynamically and objects created with the Configuration Tool.
- The threading model provides thread types for a variety of situations. Hardware interrupts, software interrupts, tasks, idle functions, and periodic functions are all supported. You can control the priorities and blocking characteristics of threads through your choice of thread types.
- Structures to support communication and synchronization between threads are provided. These include semaphores, mailboxes, and resource locks.
- Two I/O models are supported for maximum flexibility and power. Pipes are used for target/host communication and to support simple cases in which one thread writes to the pipe and another reads from the pipe. Streams are used for more complex I/O and to support device drivers.
- Low-level system primitives are provided to make it easier to handle errors, create common data structures, and manage memory usage.

In our application we have used a number of features of DSP/BIOS to perform scheduling and synchronization. These include:

- **Threading:** A thread is basically any independent stream of instructions executed by the DSP. A thread is a single point of control that can contain a subroutine, an interrupt service routine (ISR), or a function call. We have used hardware interrupts, software interrupts, mailboxes and other features for synchronizing data.
- **Memory Management:** The Memory Section Manager (MEM module) manages named memory segments that correspond to physical ranges of memory. Various sections of the hardware can be divided into segments for code and data. In each of them dynamic memory allocation and dynamic object creation can be enabled and disabled. A set of functions are provided to be used to dynamically allocate and free variable-sized blocks of memory.
- **Input/Output:** Input and output for DSP/BIOS applications are handled by stream, pipe, and host channel objects. Each type of object has its own module for managing data input and output. Data pipes are used to buffer streams of



input and output data. These data pipes provide a consistent software data structure you can use to drive I/O between the DSP device and all kinds of real-time peripheral devices.

The above features are explained in detail in the details of the audio example that we have used in implementing the codec.

5.4 CCS Programming Model

The Code Composer Studio Provides with an integrated development environment for developing application for the signal processor in C, C++ and assembly language. The summary of this programming model is shown in Fig. 5.1. CCS provides with a number of debugging tools like Watch Window, Graph Viewer, Statistics Viewer, Stack View and a number of other tools to make development easier on the signal processor.

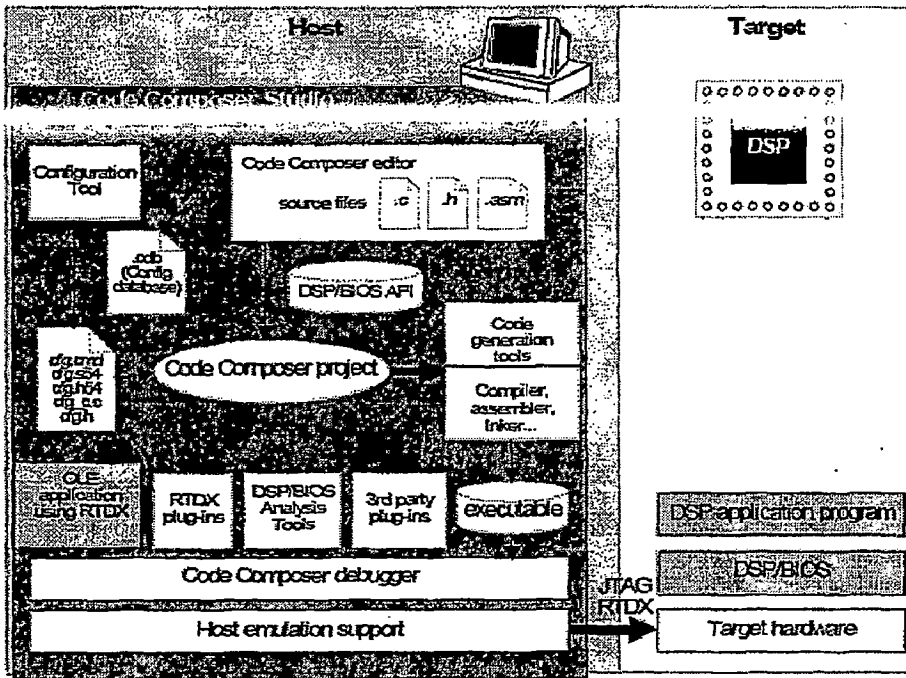


Fig. 5.1 Code Composer Studio Programming Model

CHAPTER 6

SIMULATION AND IMPLEMENTATION ON DSP KIT C6711

The implementation on the signal processor was a fairly difficult task as learning about the signal processor is equivalent to learning a whole new platform, with its own Kernel (DSP/BIOS) hardware, peripheral and development tools. A major part of our effort was gone into familiarizing with the various concepts used in this application.

6.1 Introduction to Audio Example

Audio example is a sample program provided by TI which uses all the features discussed in Chapter 5 to perform an audio amplification task. Data transfer is essential for any digital signal processing application. Texas Instruments DSP/BIOS kernel provides basic runtime services used for managing data transfer. The SP/BIOS pipes are used to buffer streams of program input and output data. Data transfer is scheduled through the use of DSP/BIOS software interrupts. These software interrupts, patterned after hardware interrupt routines, are the foundation for structuring DSP/BIOS applications in a prioritized hierarchy of real-time threads. This audio example demonstrates how to use DSP/BIOS APIs for scheduling data transfer between the hardware I/O peripherals and the target DSP.

6.2 Software Interrupts

The SWI (Software interrupts) module manages software interrupt service routines, which are patterned after HWI (hardware interrupt) routines, are triggered programmatically through DSP/BIOS API calls, such as SWI_post, from client threads. Once triggered, execution of a SWI routine will strictly preempt any current background activity within the program as well as any SWIs of lower priority; HWI hardware interrupt routines on the other hand take precedence over SWIs and remain enabled during execution of all handlers, allowing timely response to hardware peripherals with the target system. Software interrupts or SWIs provide a range of threads that have intermediate priority between HWI functions and the background idle loop (IDL).

6.3 Pipe or PIP Module

The DSP/BIOS Buffered Pipe Manager or PIP Module manages block I/O (also called stream-based or asynchronous I/O) used to buffer streams of program input and output typically processed by embedded DSP applications. Each pipe object maintains a buffer divided into a fixed number of fixed length frames. All I/O operations on a pipe deal with one frame at a time. Although each frame has a fixed length, the application may put a variable amount of data in each frame (up to the length of the frame). Note that a pipe has two ends. The writer end is where the program writes frames of data. The reader end is where the program reads frames of data.

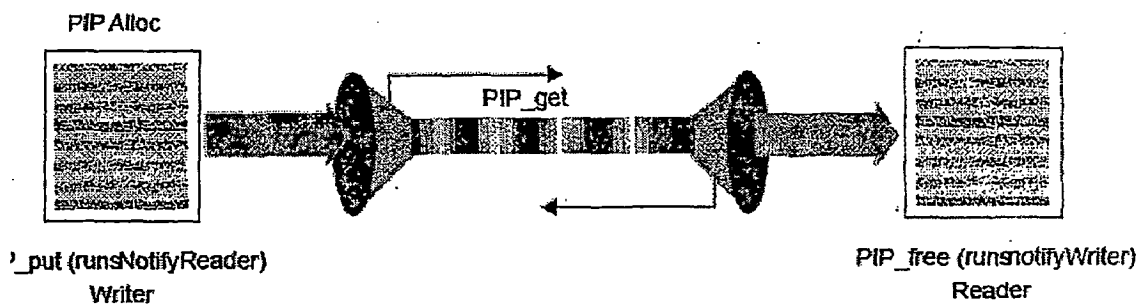


Fig. 6.1: Operation semantics of Pipe module

Data notification functions (`notifyReader` and `notifyWriter`) are performed to synchronize data transfer. These functions are triggered when a frame of data is read or written to notify the program that a frame is free or data is available. These functions are performed in the context of the function that calls `PIP_free` or `PIP_put`. They may also be called from the thread that calls `PIP_get` or `PIP_alloc`. Once `PIP_alloc` is called, DSP/BIOS checks whether there are more full frames in the pipe. If so, the `notifyReader` function is executed. After `PIP_alloc` is called, DSP/BIOS whether there are more empty frames in the pipe. If so, the `notifyWriter` function is executed. A pipe should have a single reader and a single writer. Often, one end of a pipe is controlled by a hardware TSR (ex: Serial Port Receiver TSR) and on the other end is controlled by a software interrupt function. Pipes can also be used to transfer data within the program between two application threads.

6.4 Execution Semantics

The TSR for the serial port receive interrupt copies each new 32 bit data sample in the Data Receive Register (DRR) to a frame from the DSS_rxPipe pipe object. When the frame is full, the TSR puts the frame back into DSS_rxPipe to be read by the audio function.

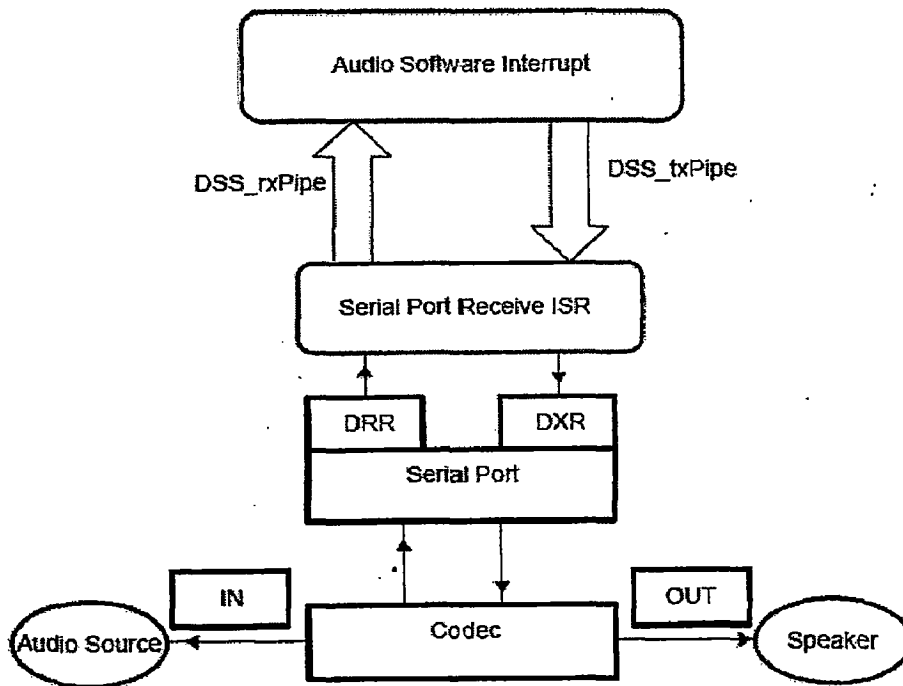


Fig. 6.2: Execution semantics of an audio I/O example

The TSR for the serial port receive interrupt copies each new 32 bit data sample in the Data Receive Register (DRR) to a frame from the DSS_rxPipe pipe object. When the frame is full, the TSR puts the frame back into DSS_rxPipe to be read by the audio function. As the audio function will just read a frame from DSS_rxPipe and copy it to a frame in DSS_txPipe. The transmit rate will be the same as the receive rate: 48 kHz. This allows us to further simplify the example by enabling only the receive interrupt for the serial port. The transmit interrupt for the serial port is not enabled.

The TSR for the receive interrupt will also take care of the transmit process in the serial port. It will take a full frame from DSSjxPipe and write a 32-bit word from the frame to the 32-bit serial port Data Transmit Register (DXR) each time the interrupt is handled. When Lile WijOic

6.5 DSS_txPipe for reuse by the audio function.

The function DSS_init (found in dss.c) takes care of the initialization of the serial port and the codec. DSS_init programs the sampling rate of the codec and sets the bits in IMR and IFR to enable the serial port receive interrupt, etc.

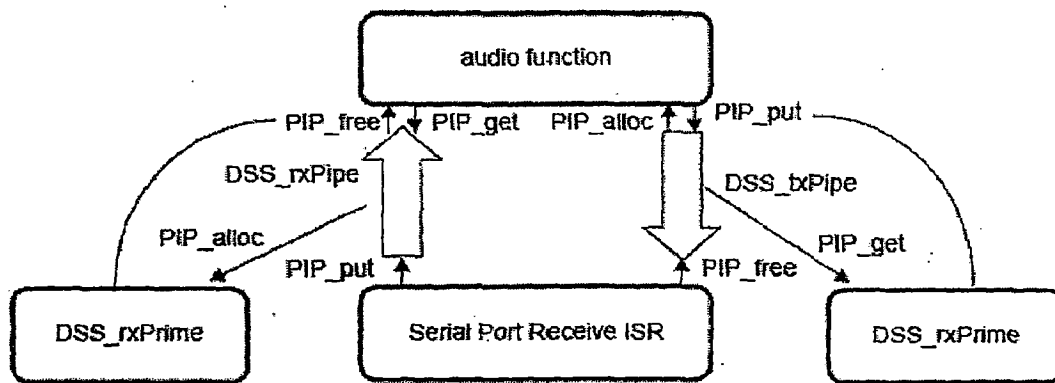


Fig. 6.3: More details of execution semantics

When a full frame is put in DSSrxPipe the notifyReader function will clear the second bit in the mailbox for audioSW1. When an empty frame is available in DSSjxPipe, the first bit in the mailbox for audioSW1 is cleared. In this way, audioSW1 is posted only when there is a full frame available in DSSrxPipe and an empty frame available in DSS_txPipe. The notifyWriter for DSS_rxPipe, DSS_rxPrime, is a C function that can be found in dss.c. DSS_rxPrime calls PIP_alloc to allocate an empty frame from DSS_rxPipe that will be used by the TSR to write the data received from the codec. DSS_rxPrime is called whenever an empty frame is available in DSS_rxPipe (and the TSR is done with the previous frame). The TSR calls DSS_rxPrime after it is done filling up a frame (see Fig. 6.3). The notifyReader for DSSjxPipe, DSS_txPrime, is a C function that can be found in dss.c. DSS_txPrime calls PIP_get to get a full frame from DSS_txPipe. The data in this frame will be transmitted by the ISR to the codec. DSS_txPrime is called whenever a full frame is available in DSS_txPipe (and the TSR is

done transmitting the previous frame). The TSR calls DSS_txPrime after it is done transmitting a frame to get the next full frame.

6.6 Details of Algorithms and Flowcharts

The details of algorithms used and the program flow are shown in the following pages with the help of flowcharts. It is to be noted that program flow and algorithms are the same for the real-time as well as non-real-time implementations.

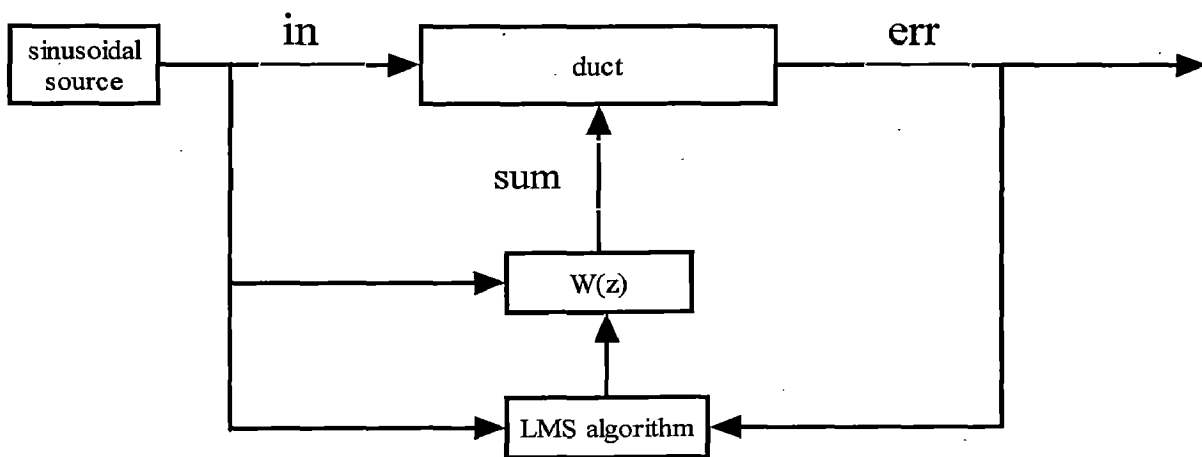
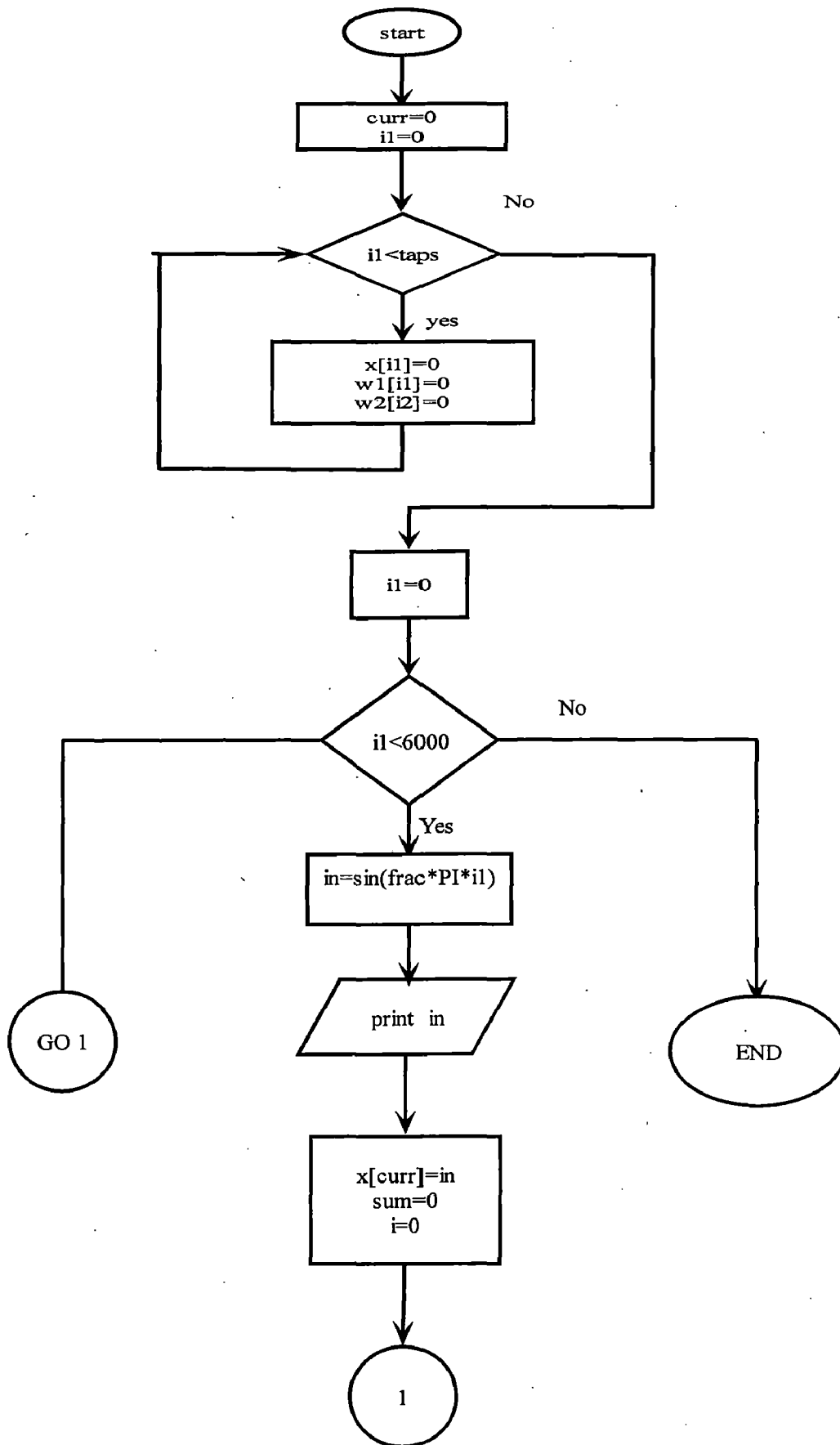
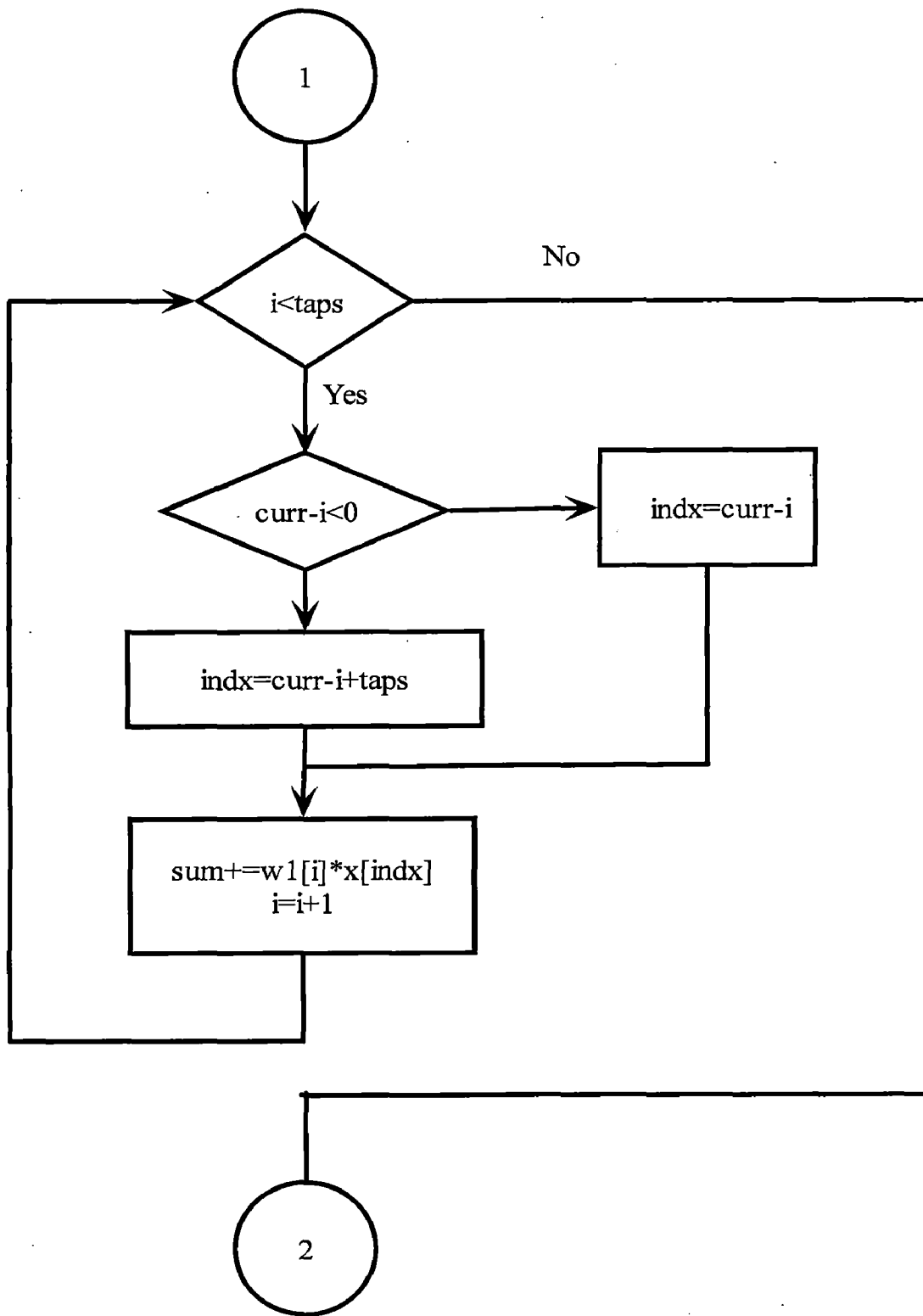
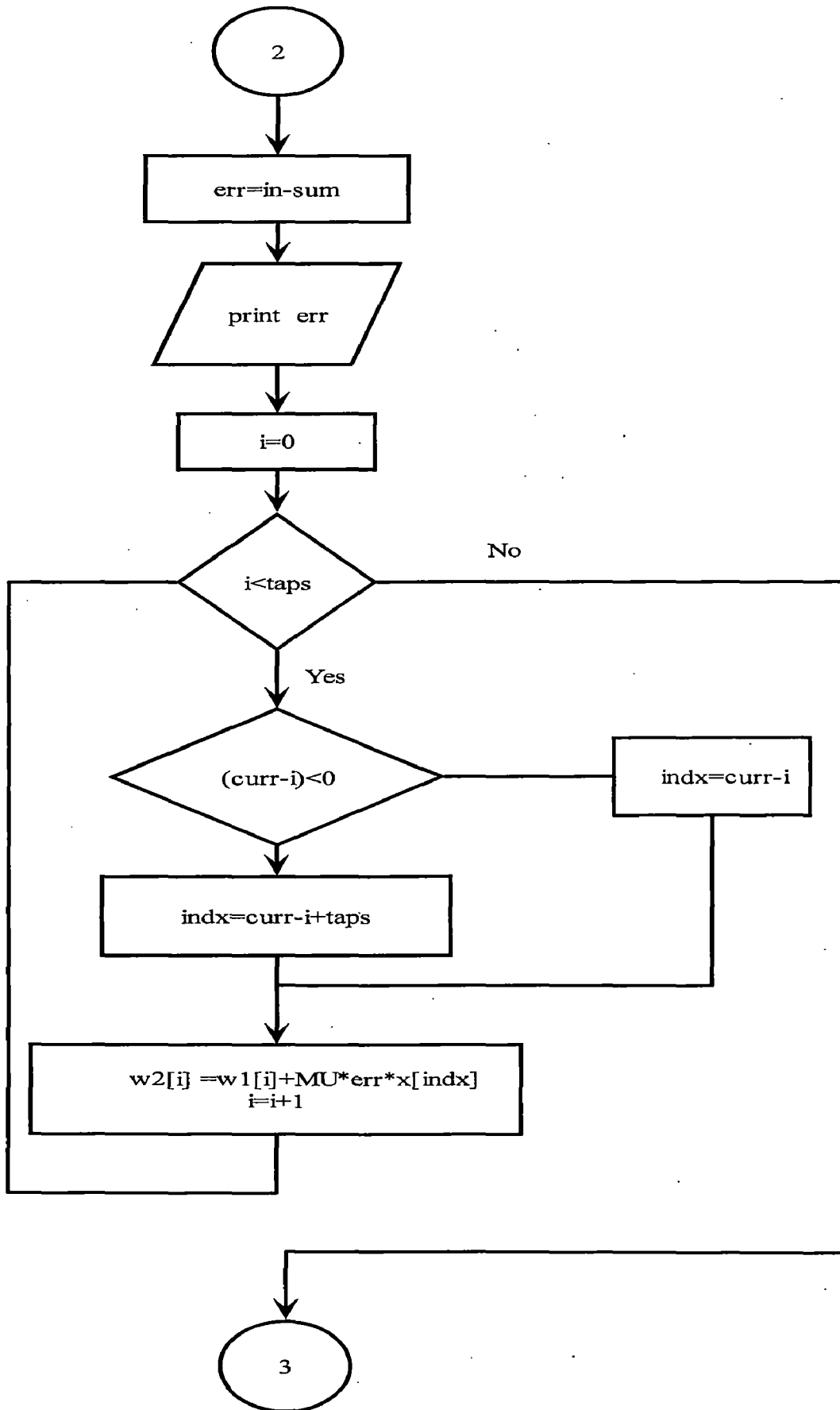


Fig. 6.4: Simulation model of a FIR filter with LMS algorithm

The sine wave is procured by using the sine function and the value is stored in a variable 'in'. This value is processed by the FIR filter through the weighing coefficients $W(z)$ which are initialized firstly to zero. These coefficients are optimized by the LMS algorithm such that the error to be minimized. This LMS algorithm requires two input values: 'in','err' as shown in above Fig.6.4. This algorithm is already explained in chapter 4 and the complete computation procedure is showed through the flowcharts as given below.







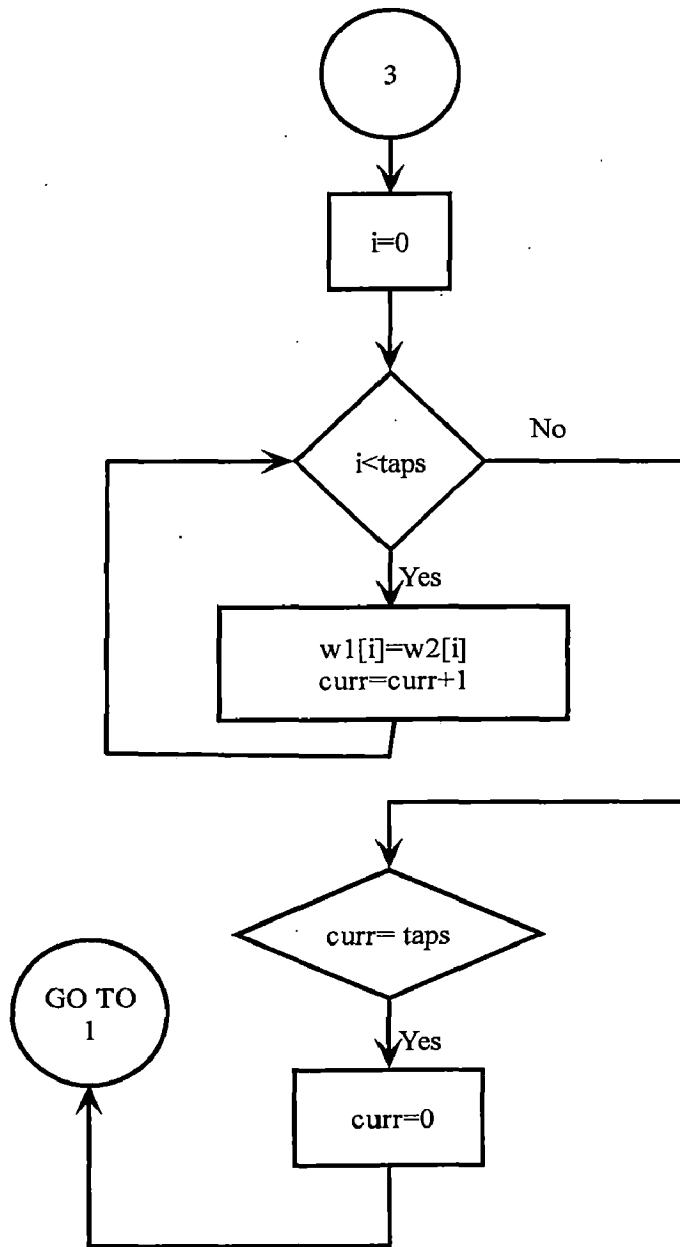


Fig 6.5 flow chart of FIR filter with LMS algorithm

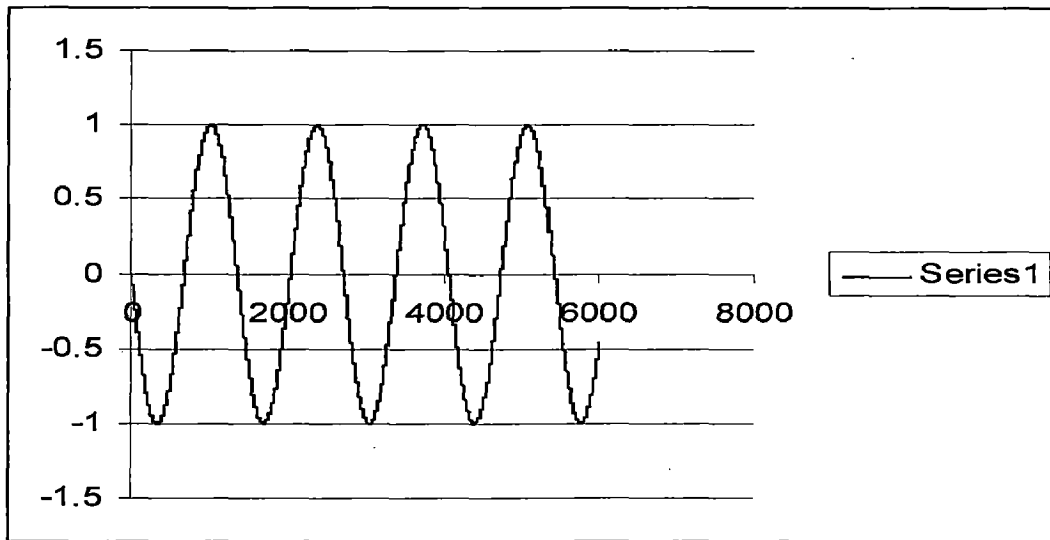


Fig. 6.6: sine wave input of 50Hz frequency to the simulation model

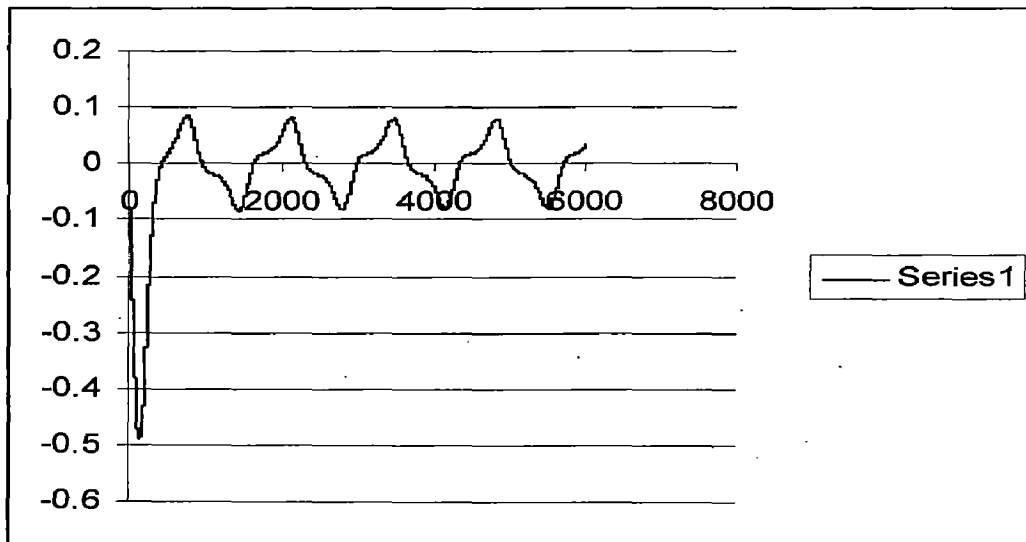


Fig. 6.7 Residual noise levels from the simulation model for 50Hz sine wave input

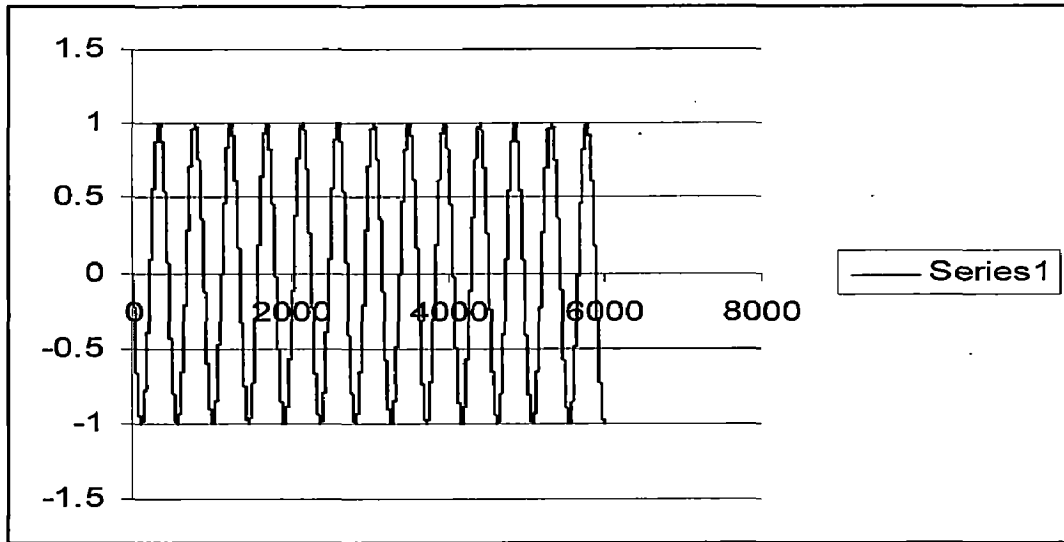


Fig. 6.8: Sine wave input of 150Hz frequency to the simulation model

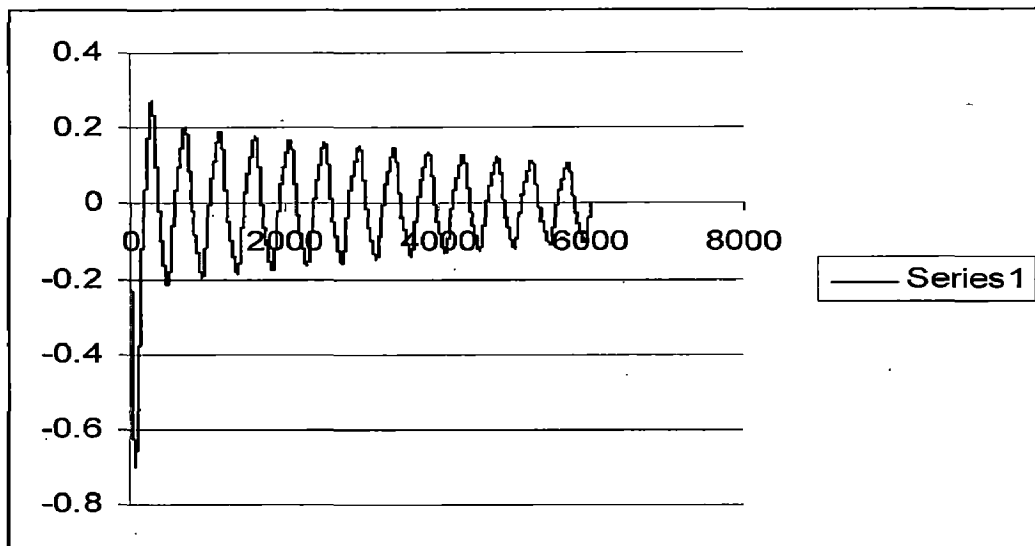


Fig. 6.9: Residual noise levels from the simulation model for 150Hz sine wave input

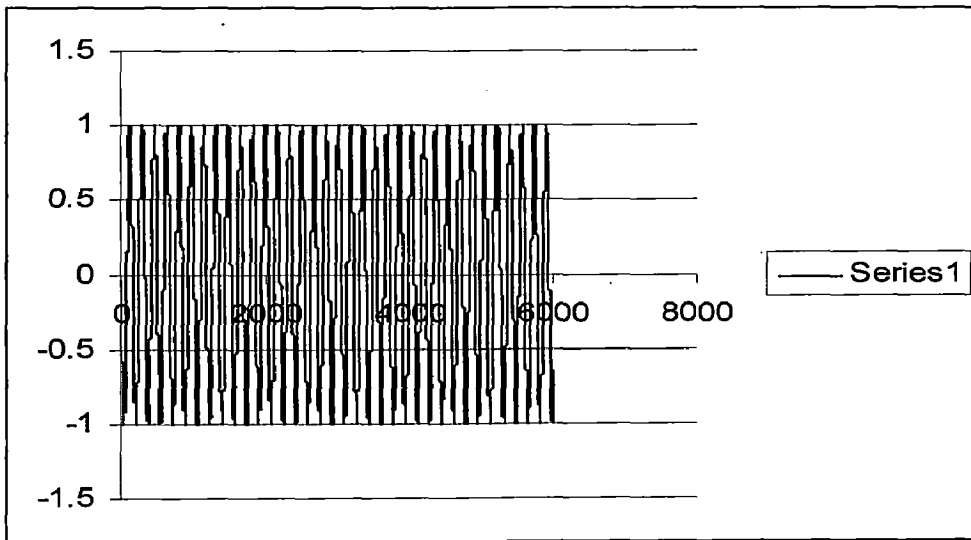


Fig. 6.10: sine wave input of 400Hz frequency to the simulation model

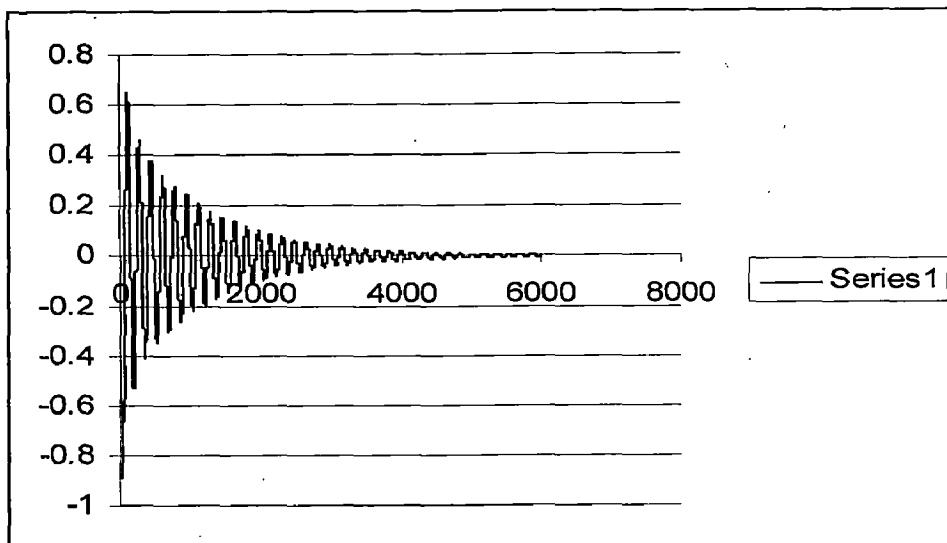


Fig. 6.11: Residual noise levels from the simulation model for 400Hz sine wave input

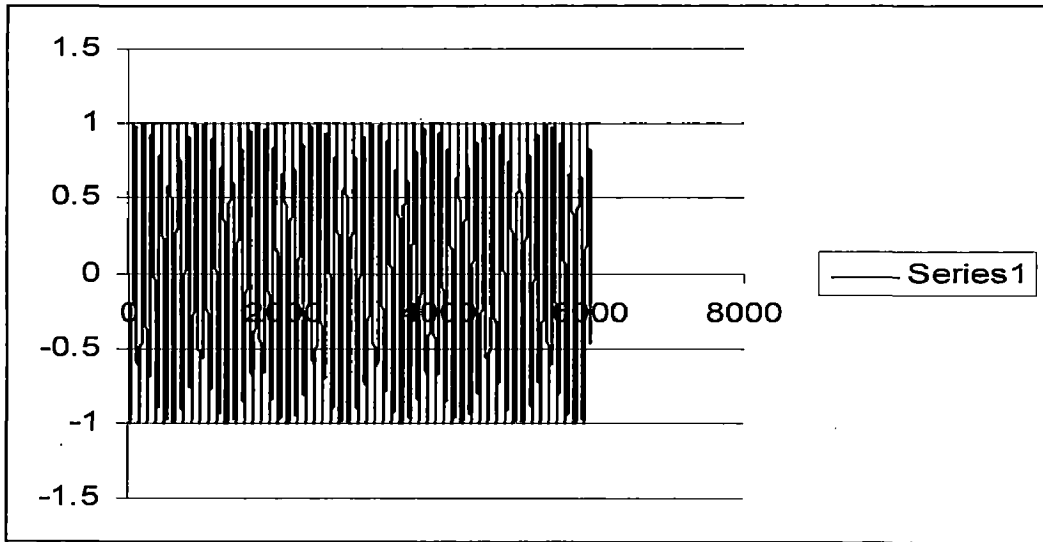


Fig. 6.12: sine wave input of 600Hz frequency to the simulation model

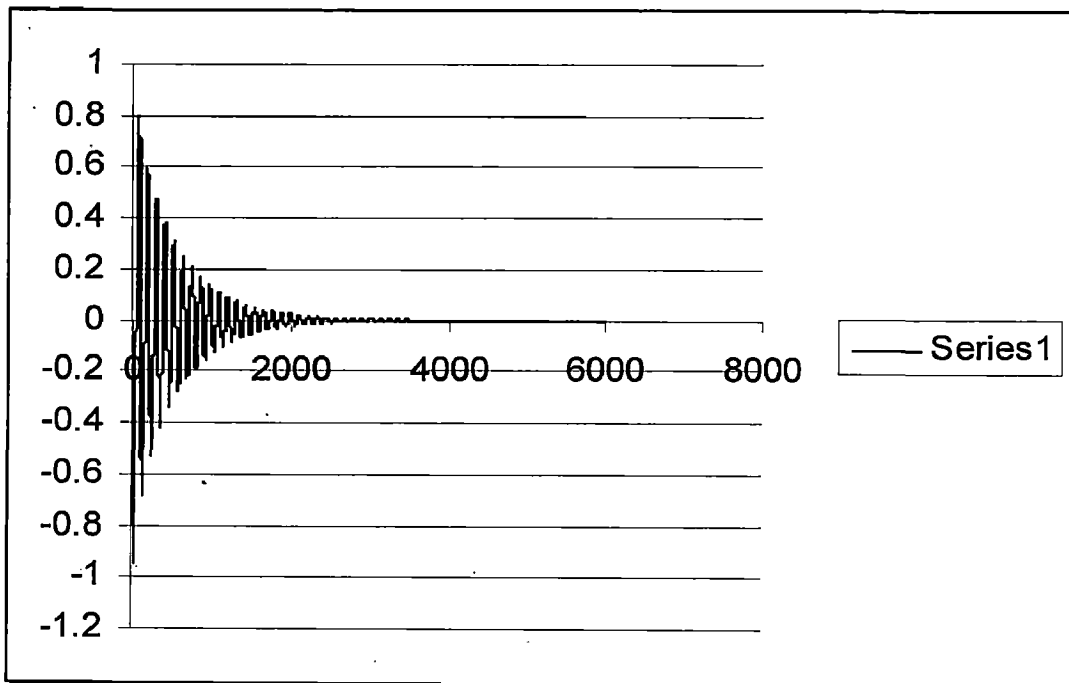


Fig 6.13: Residual noise levels from the simulation model for 600Hz sine wave input

CHAPTER 7

DESIGN AND FABRICATION OF EXPERIMENTAL SET-UP

7.1 General lay-out of rig

The rig consists of following main sections:

- A source of compressed air provided by a reciprocating compressor,
- A compressed air line and a flow regulating valve,
- A lined duct to attenuate noise,
- Main test section.

The compressed air supply was provided by a reciprocating compressor. The radiated noise level from the compressor did not cause significant interference with the measurements as the compressor were switched of at testing period. A long air pressure pipe of 16mm diameter connected the compressor to agate valve inside the test room. A relief valve is also provided near a 'T' junction so that when gate valve was closed, the air could be dumped into the atmosphere .A small expansion chamber was provided after the gate valve to attenuate the noise generated by the turbulence at the valve.

A pressure regulating valve with a filter was installed further downstream in the compressed air line .This valve helped in maintaining a constant flow rate for a particular valve setting. A 3.6 m long rectangular duct lined with rock wool was provided before the test section to attenuate the high frequency noise generated at the valve and also due to turbulence in the line.

The air coming out of this lined duct, then passed through the test section. The test section consisted of an acoustic driver unit providing the source signal and control driver unit providing control signal. Fixed point microphone fixtures were mounted at different positions on the test pipe.

7.2 Mountings for speakers

A fixture for mounting two speakers which together providing driving signal was made as shown in figure. Two short pipe stubs were inserted into holes drilled radially in

an M.S.socket opposite to each other and welded. The two holes were drilled on the same transverse plane. Same procedure was repeated to fabricate speaker mounting to generate the control signal.

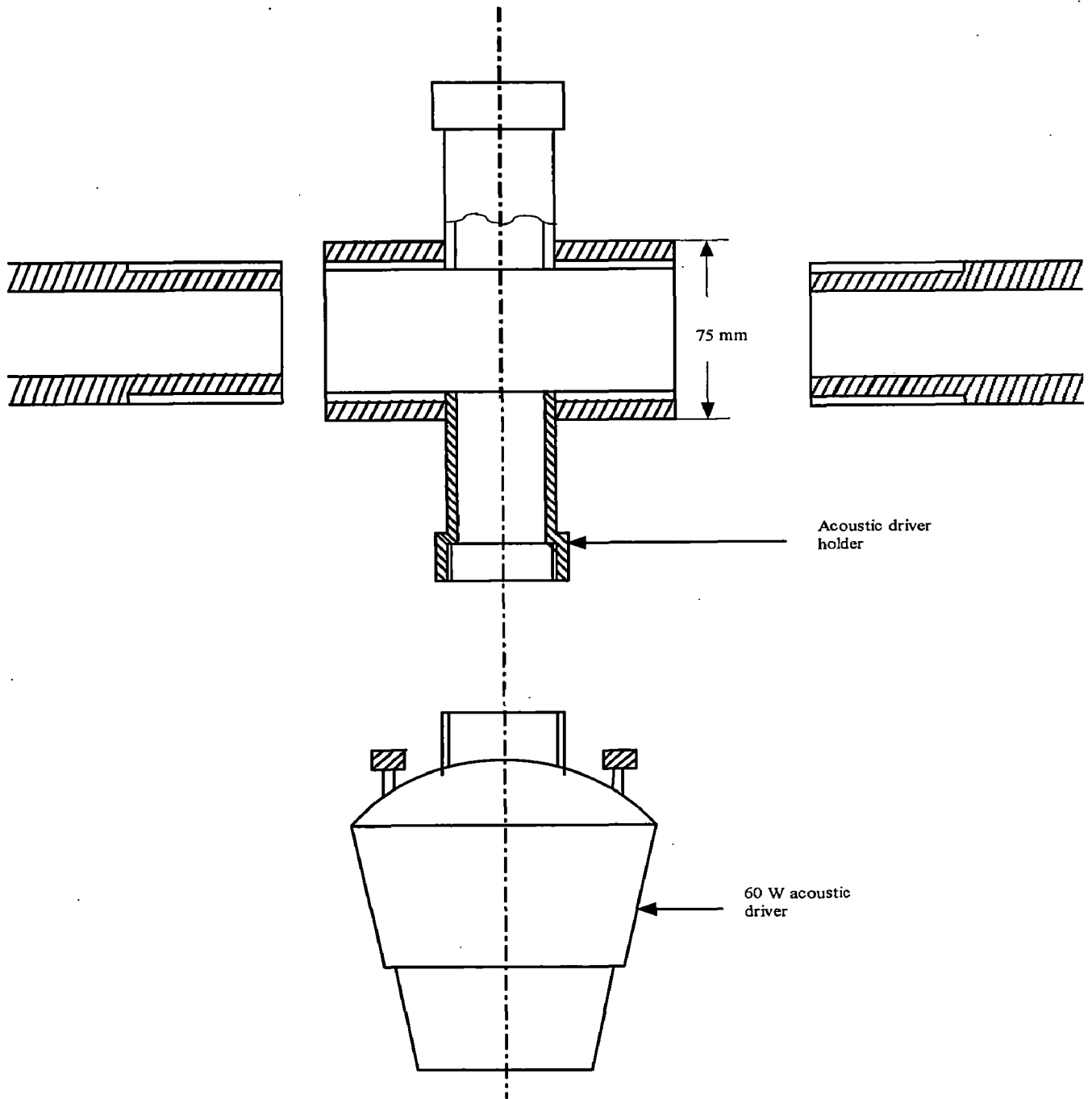


Fig. 7.1 Graphical view of socket and mounting for speakers

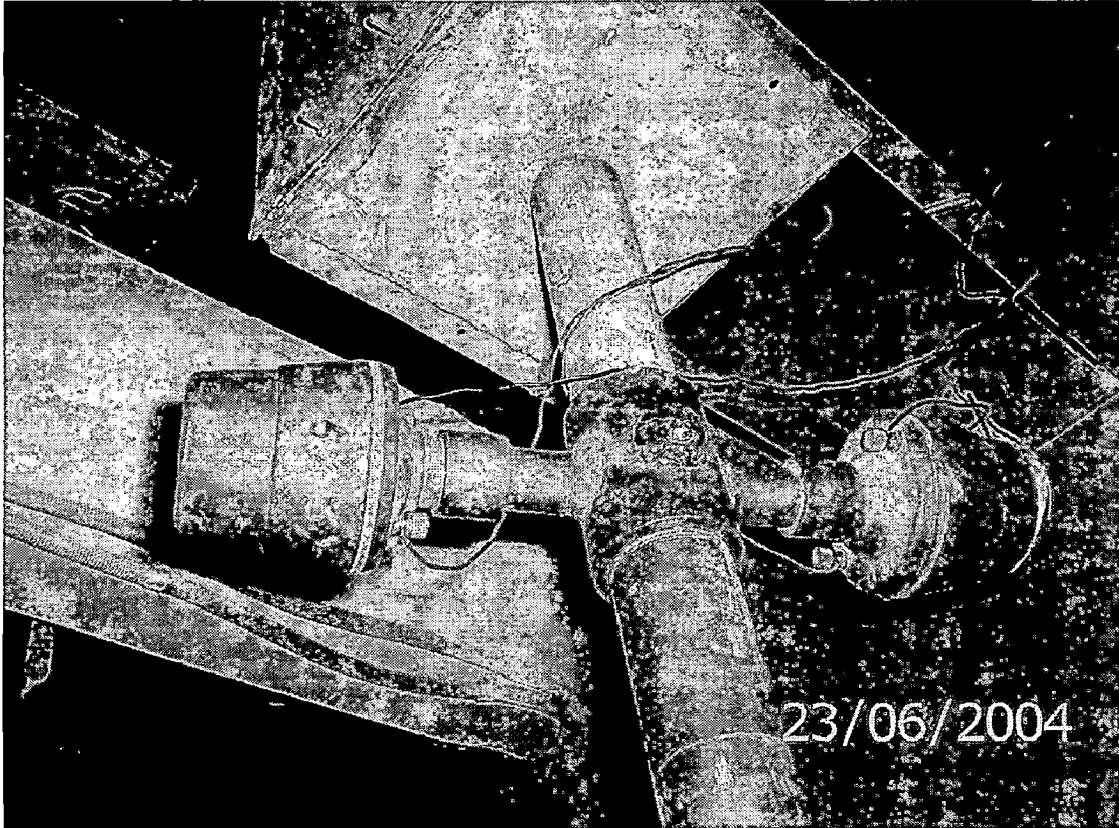


Fig. 7.2: Socket and mounting for speakers

6.3 RECTANGULAR DUCT AND ATTENUATOR

To attenuate the noise due to flow turbulence, compressor and valve noise generated at the control valve, a line duct of about 3.65 m length was made. The outer section of duct was 230 mm × 230 mm square made from sheet metal. The duct had a removable top cover which was bolted to the main duct. The ends of the duct were closed by welding M.S. plate 2 mm thick and having a 60 mm diameter, hole in the center. Two 210 mm length of 60 mm dia pipe were welded to the end plates of the duct and the other ends were threaded.

The duct core of 75 mm dia was made of expanded metal mesh. This was centrally located in the duct. A thin cotton gauge was wrapped all round the mesh. The annular space was fitted with rock wool. The cover was clamped with a number of bolts and 5mm thick rubber gasket was used between the cover and the flange. Fig.7.3 shows assembled duct.

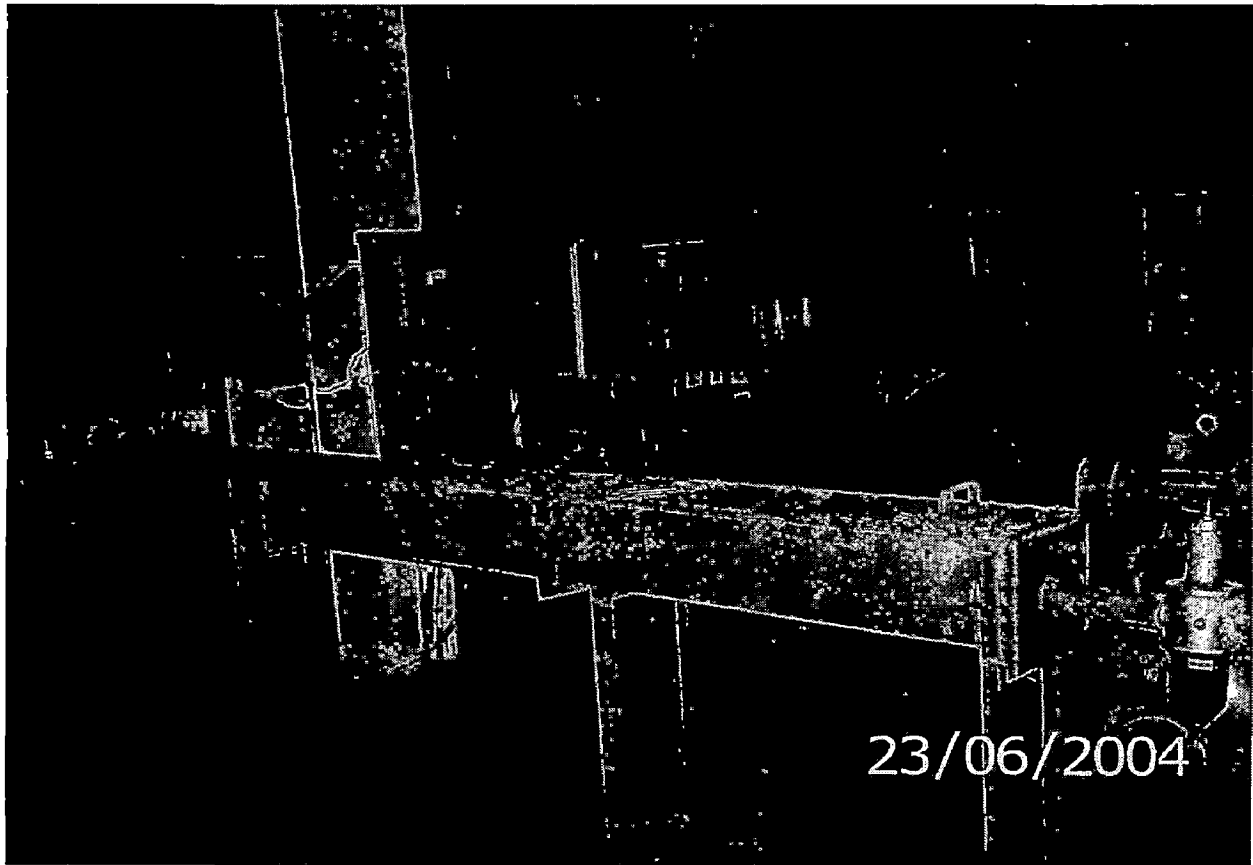


Fig. 7.3: Rectangular duct having core 75 mm dia. of expanded metal mesh

7.4 MICROPHONE HOLDERS

Microphone holders were provided on the wall of the duct for the mounting of microphones to measure the acoustic pressure at fixed wall positions in the duct. These microphone holders were made in two sizes, one for holding B & K 4145 (1") microphone and other B & K (1/2") 4133 microphone.

A rubber ring provides the necessary grip and sealing around the microphone. Outer surface of the pipe was machined flat before brazing the holders, see fig.7.5. This was done to reduce the length of the neck of the Helmholtz resonator formed. 2 mm dia hole was drilled on the flattened surface of the pipe at the centre of holder and a microphone is placed above this cavity. This is done to get better measurement of sound pressure level without forming any eddies around the microphone and without changing the direction of air flow.

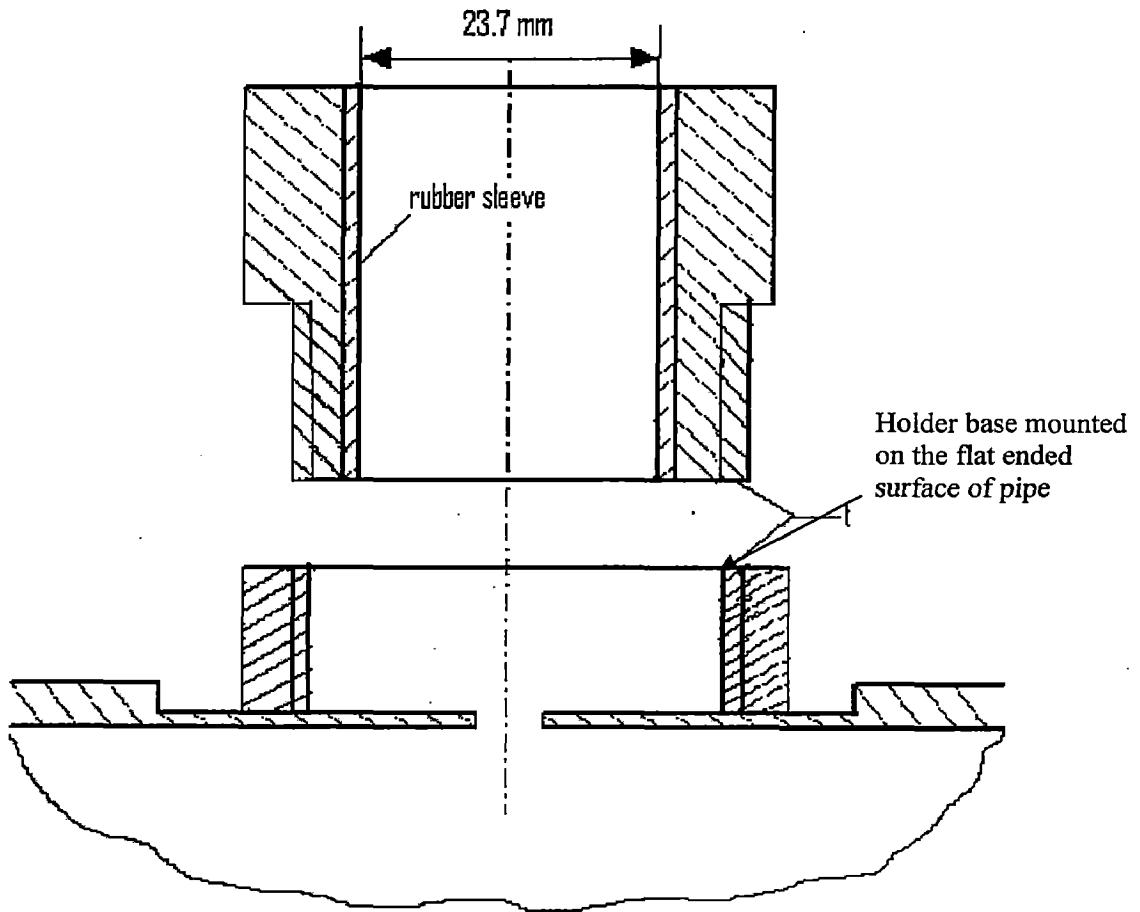


Fig 7.4: Microphone mounting for duct wall pressure measurement

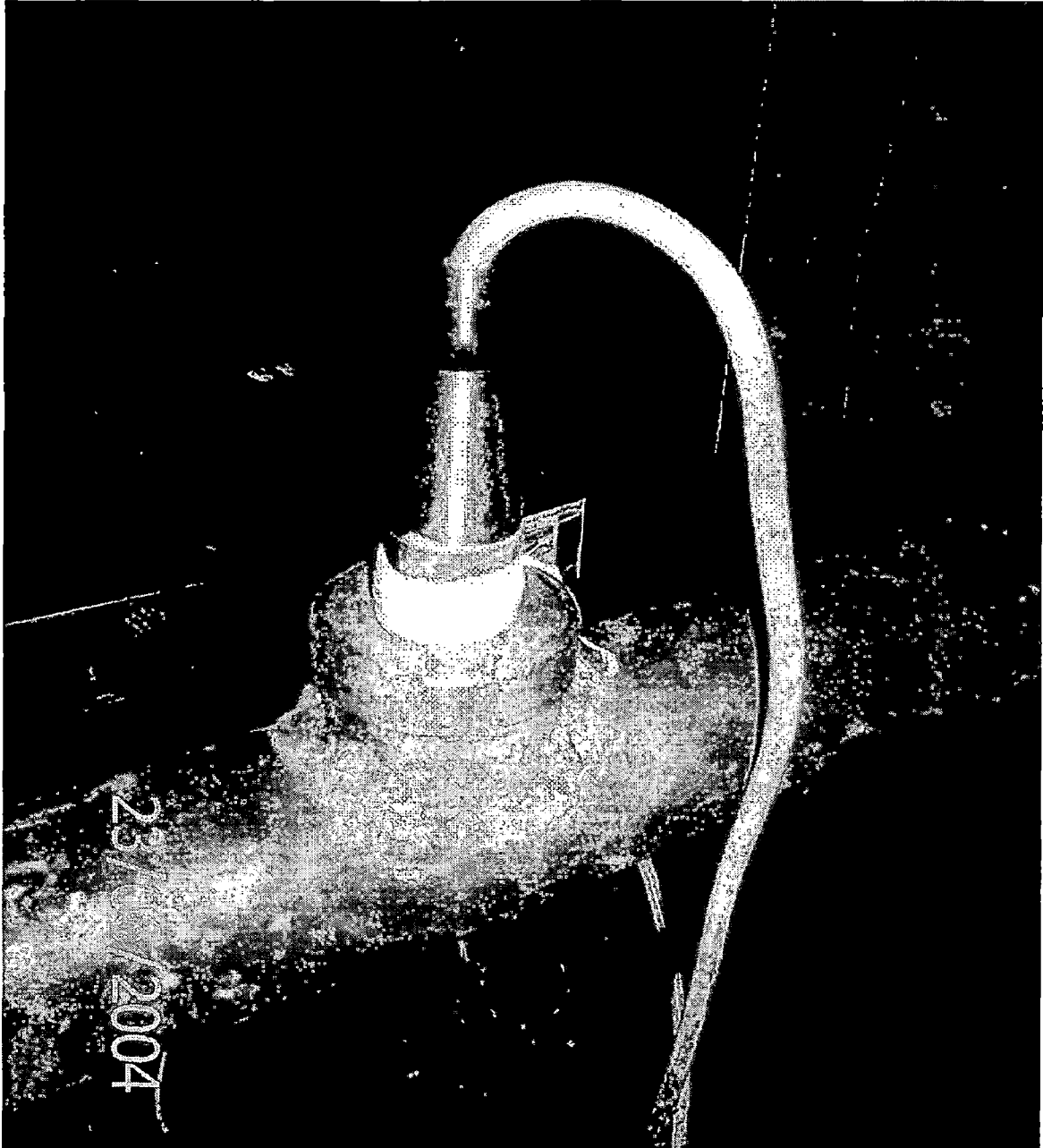


Fig. 7.5: microphone mounting and 1 inch microphone to sense the primary noise level

7.5 Experimental Procedure

A complete configuration of an adaptive control system for the active control of sound transmission produced by an acoustic source in a duct is shown in Fig. 7.6. In this figure, the primary and secondary sources are assumed, respectively, to be uncontrollable and controllable. Regarding the noise signal generated by the primary source is picked up by reference microphone. Then it is amplified by a charge amplifier and transferred, to become a discrete input of the adopted adaptive filter, by using an analog-to-digital converter. Meanwhile, the signal received by the error microphone just behind the secondary source must also be amplified to become an adaptive input of the adopted filter for the required adaptive processing. By combining two input signals of the adaptive filter described above, the resulting output is transferred, to become a continuous signal, by using a digital-to-analog converter. Regarding the corresponding high frequency component of noise signal produced by the used D/A converter, this must be filtered by using a low-pass filter, before it is amplified, to be the driving signal of the secondary source. This procedure is repeated for 50Hz, 150Hz, 400Hz, 600Hz sine inputs and residual noises are shown in Figs. 8.1 to 8.4.

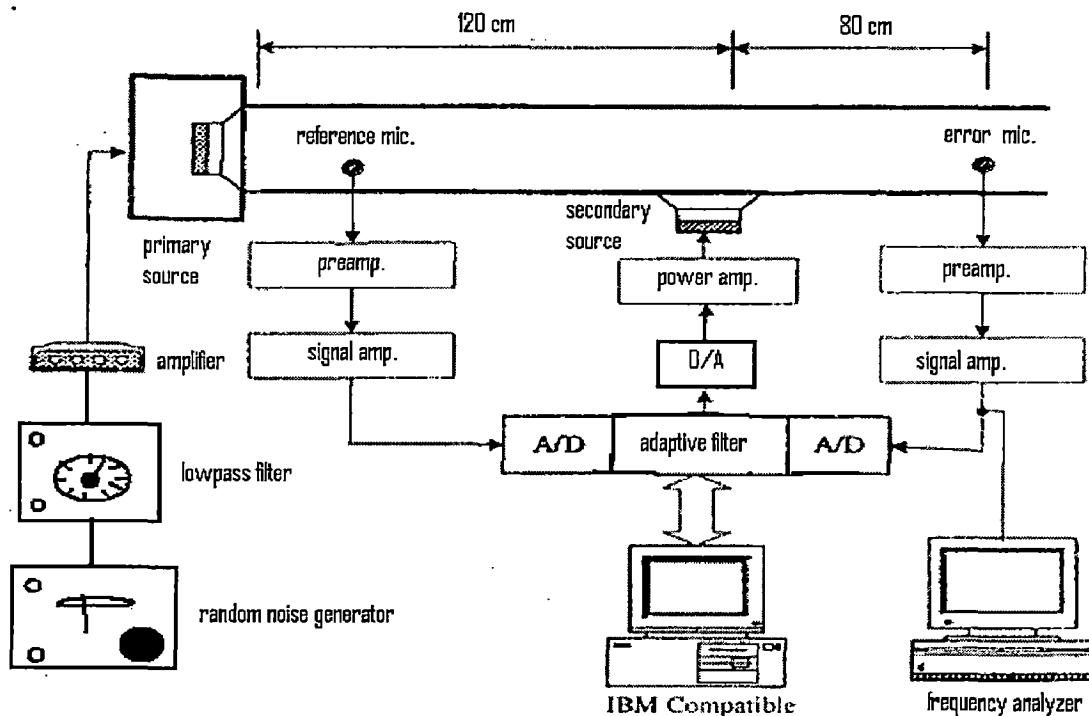


Fig.7.6: The schematic diagram for the adaptively active control experiment of the acoustic transmission of a sound source in a duct.



Fig. 7.7 The experimental setup for the adaptively active control experiment of the acoustic transmission of a sound source in an aperture.

CHAPTER 8

RESULTS AND DISCUSSION

By using the arrangement of the equipment shown in Fig.7.6 and by following the adaptive control procedure as described before, the experiment for active control of the acoustic transmissions produced by 1/3rd octave band acoustic source of central frequencies at 50Hz, 150Hz, 400Hz and 600Hz is performed. These frequencies were chosen to correspond to the entire fall in the fundamental firing frequency of I.C engines. The sound pressure spectrum above about 2.5 kHz is largely dominated by broadband flow noise and is also less intensive. The corresponding sound pressure levels at the measuring section were measured without adaptive control, open loop adaptation and closed loop adaptation and results are shown in Figs. 8.2 to 8.5.

8.1 Discussion on Open Loop Adaptation Results

The primary excitation was supplied by the speaker S_1 mounted at the section 1. The sound field setup in the duct is sampled at section 1 which is fed to the adaptive filter. The output of the adaptive filter powers the secondary source speaker S_2 at section 2. A destructive interference results in reduction in sound pressure level at the measuring section 4 almost of 16dB to 20dB with the use of open loop adaptation. The level of reduction achieved is somewhat limited because the sound field experienced by the reference microphone M_1 at section 1 is not only constituted of the incident waves from the primary source moving downstream. A complex system of interference pattern generated by the waves reflected from the duct termination, traveling upstream which returned with near 180^0 phase change at the open termination similar to a silencer tail

pipe. If had the microphone at section 1 sensed only the incident waves traveling down stream and adaptive control just about reversed the phase and amplified it to the secondary speaker at section 3, it would have resulted in near total cancellation of the sound field. To achieve this, sound field needed to be decomposed into incident and reflected sound waves using a two microphone measuring system and suitable wave decomposition software in real time.

Facilities available in the lab did not permit such an effort in this pilot study. Another, reason for a lower reduction in noise level is due to the excitation not being pure tonal but band limited white noise, making it difficult for the control device to carry out phase alterations in the signal fed to the secondary source S_2 for all the frequencies contained within the $1/3^{\text{rd}}$ octave band resulting in total cancellation.

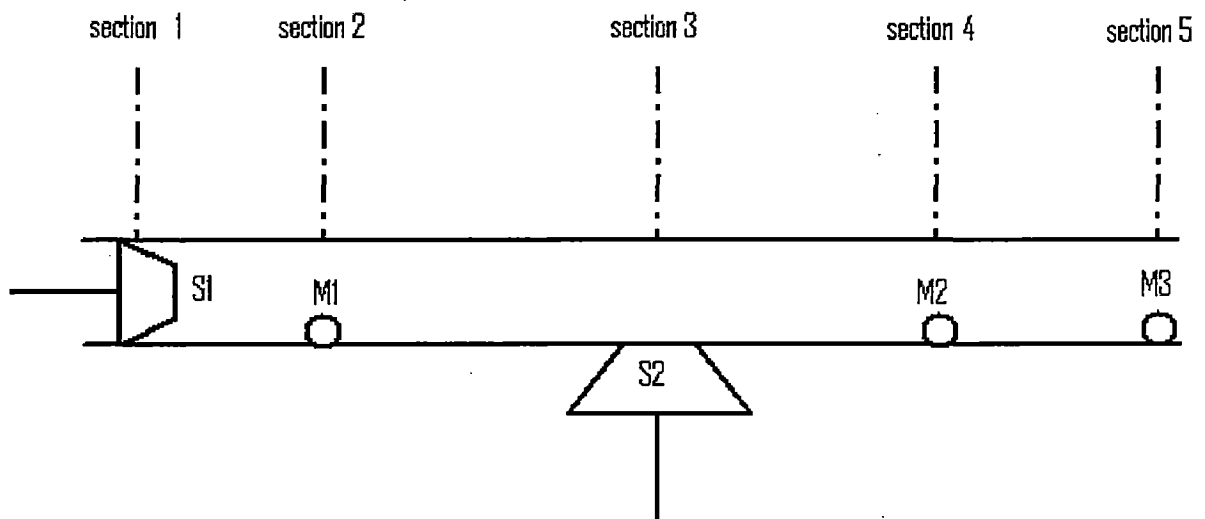


Fig.8.1: Representation of different sections.

8.2 Discussion on Closed Loop Adaptation Results

The attenuation achieved through active noise control was further enhanced by using a closed loop adaptation by using the same DSP kit. Which required the installation of an error microphone M_2 , which measured the residual field downstream of secondary source S_2 at section '3'. The residual field signal sensed fed into the DSP kit to complete the feed back path. The error microphone is used to further modify the signal driving the secondary source speakers S_2 . With this the attenuation achieved by active noise control system could be enhanced as seen by the results shown in fig 8.2 – 8.4. Attenuation of the order of 32db has been obtained.

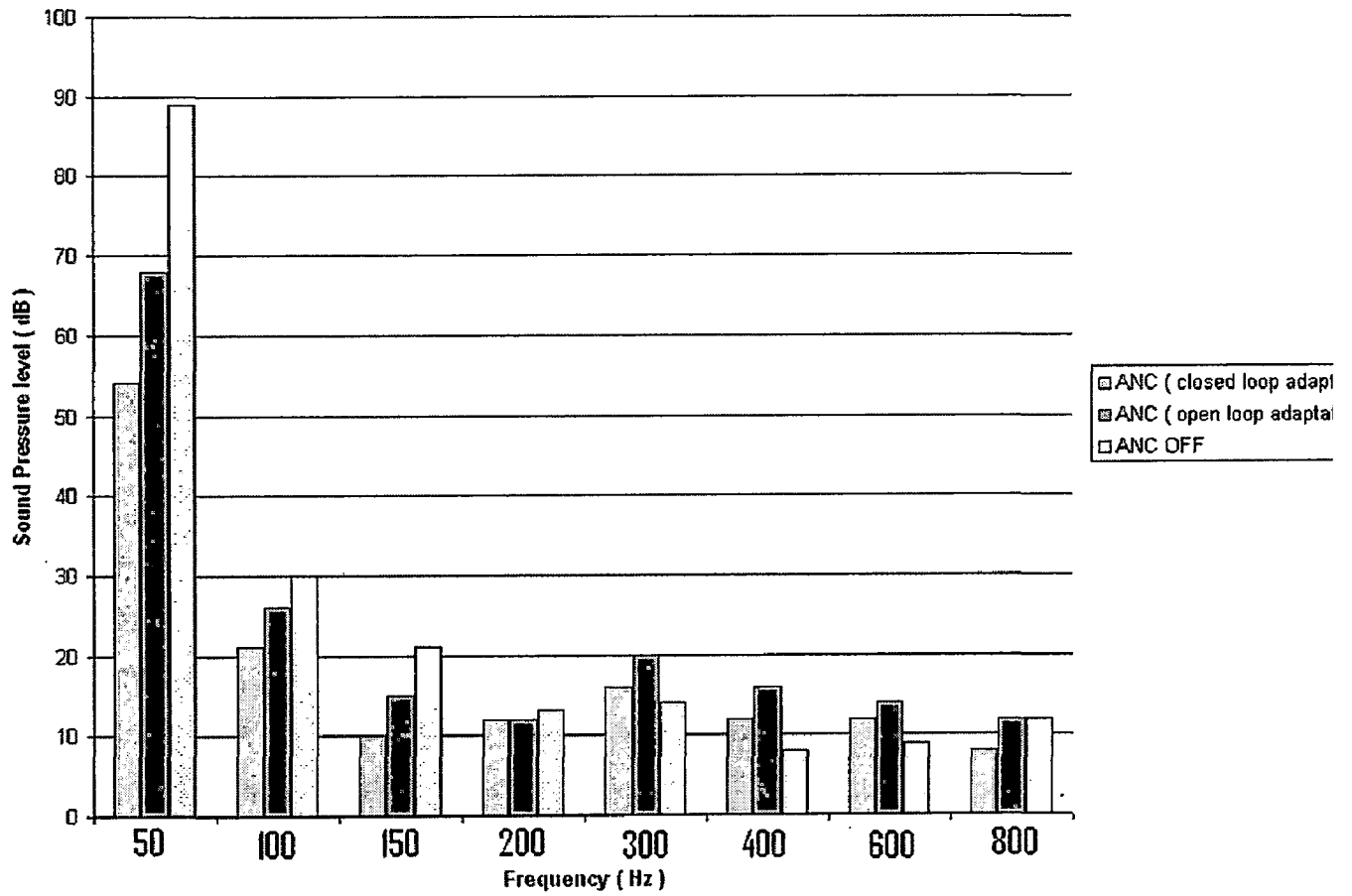


Fig. 8.2 The acoustic pressures generated by a primary pure-tone source at 50 Hz, and is measured at a point behind the secondary source.

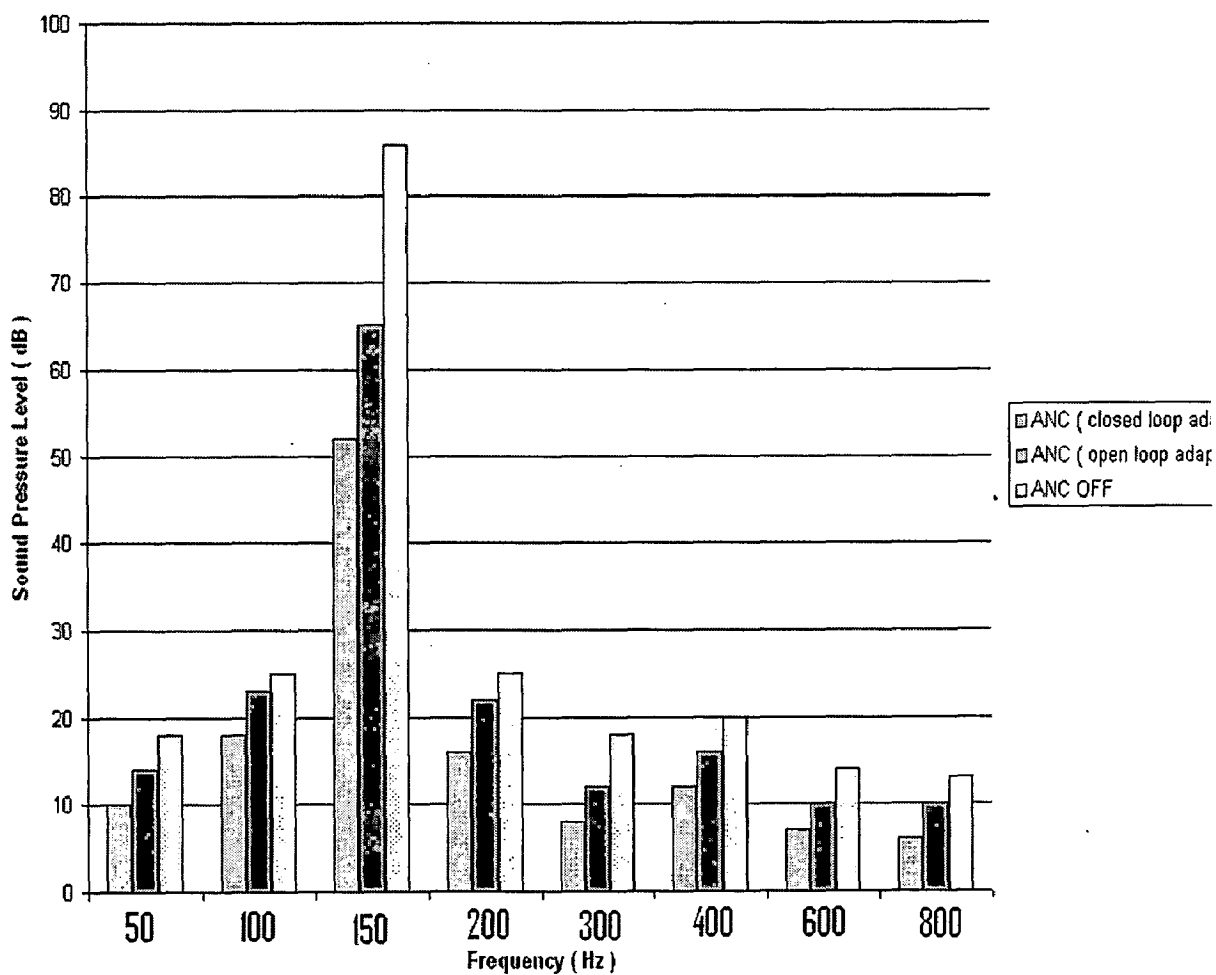


Fig. 8.3 The acoustic pressure generated by a primary pure-tone source at 150 Hz, and is measured at a point behind the secondary source.

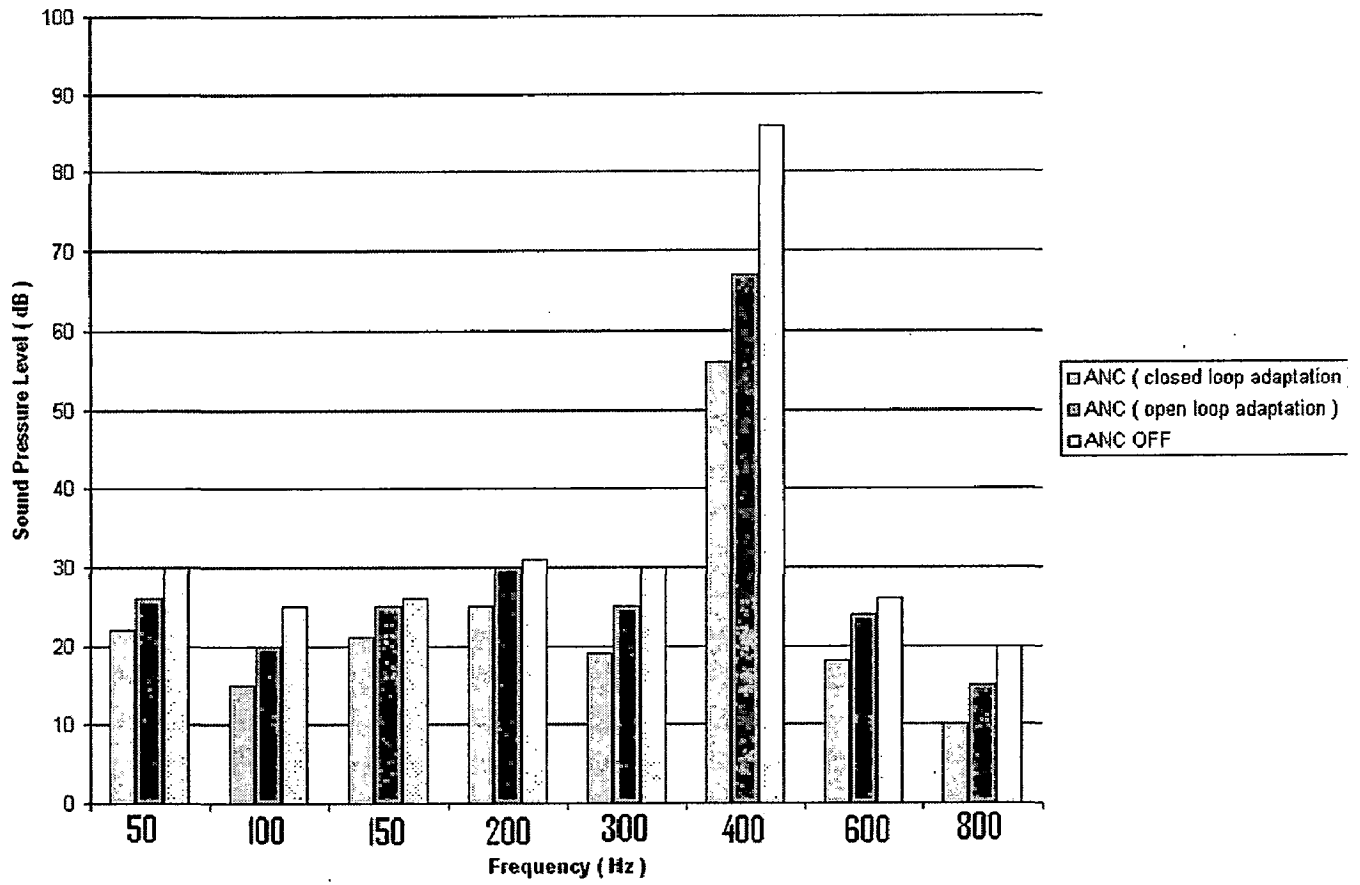


Fig. 8.4 The acoustic pressure generated by a primary pure-tone source at 400 Hz, and is measured at a point behind the secondary source.

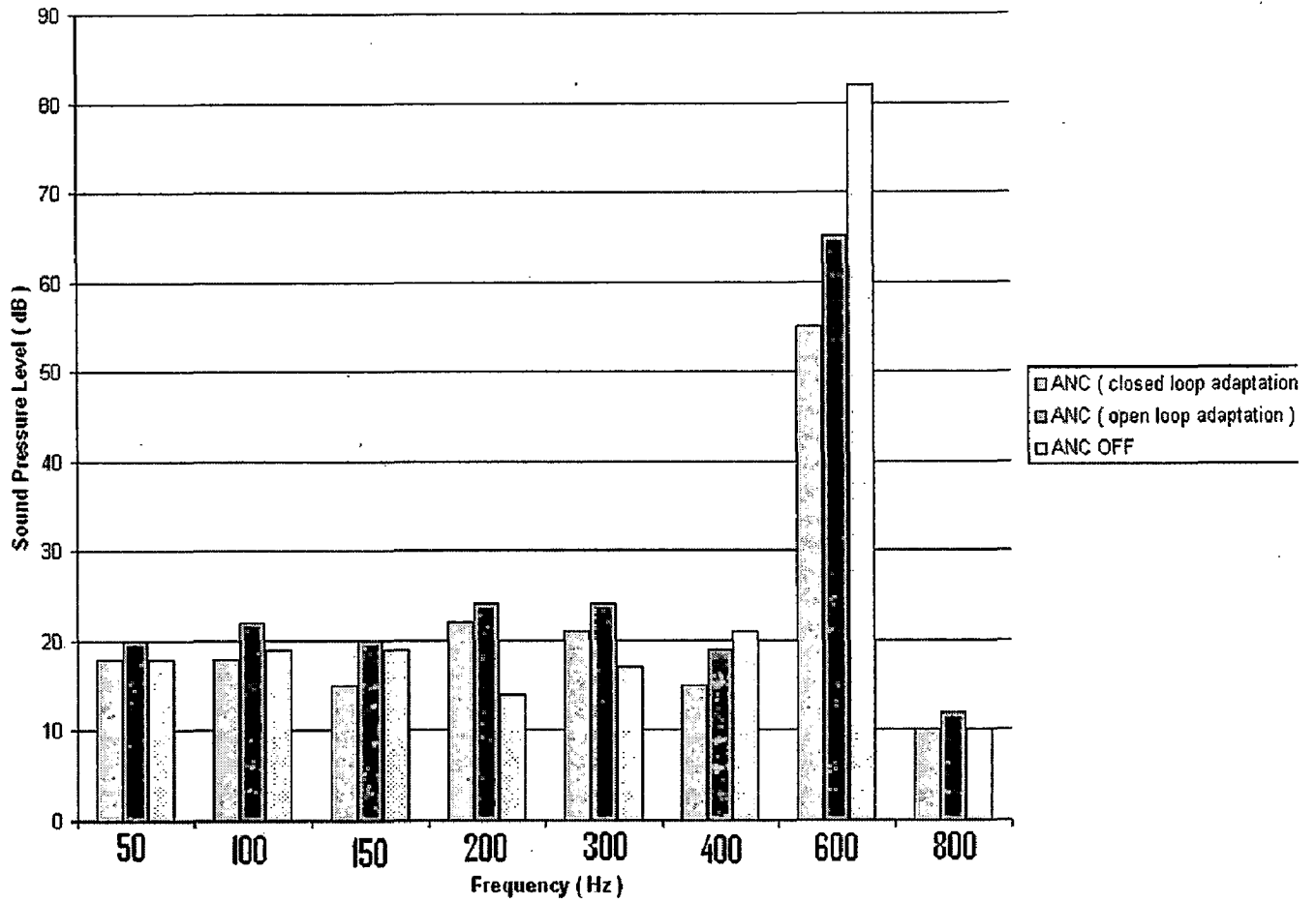


Fig. 8.5 The acoustic pressures generated by a primary pure-tone source at 600 Hz, and is measured at a point behind the secondary source.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

Though this pilot study it has been demonstrated that active noise control through wave cancellation has been realized in the laboratory. Attenuations to the tune of 30 dB were realized in the low to medium frequency range, with minimal equipment, without the use of wave decomposition.

This demonstration is likely to be useful in reducing the ^{volume} ~~value~~ of reactive silencer. (Passive noise control device) which tend to become bulky when call upon to attenuate the lower frequencies. When the burden of low frequency noise reduction taken off from these passive devices and passed on to the active noise control procedure the bulk of the passive device is bound to reduce.

Scope for Future Work

The basic demonstration of active noise control has been accomplished in the lab with the severely limited instrumentation.

The work can be extended with the help of wave decomposition technique to isolate the incident wave train from the standing wave field inside the duct, with atleast a pair of micro phones and better still with an array of 3 to 4 microphones. subsequently processing this incident wave signal in real time and feeding it though open/closed loop adaptive control device to the secondary source speakers 'S₂' so as to achieve an affective active noise control.

REFERENCES

1. **Nelson, P.A., Elliott, S.J.**, "Active Control of Sound", Academic Press, San Diego, C.A., 1992.
2. **Lueg, P.**, "Process of Silencing Sound Oscillation", U.S. Patent No.2, 043, 416, June, 1936.
3. **Burgess, J.C.**, "Active Adaptive Sound Control in a Duct: A Computer Simulation", *J.Acoust.Soc.Am* 70, 1981, 715-726.
4. **Widrow, B., Stearns, S.D.**, "Adaptive Signal Processing", Prentice Hall, Englewood Cliffs, N.J., 1985.
5. **Kuo, S.M., Chen, C.**, "Implementation of Filters with the TMS320C25 or TMS320C30 Family", Volume 3, edited by P.Papamichalis, Prentice-Hall, Englewood Cliffs, N.J., 191-271, 1990.
6. **Kristiansen, Ulf R.**, "A Different Type of Resonator for Acoustics", *Applied Acoustics* 26, 1989, 175-179.
7. **Lamancusa, J.S.**, "The Transmission Loss of Double Expansion Chamber Mufflers with Unequal Size Chambers", *Applied Acoustics* 24, 1988, 15-32.
8. **Selamet, A., Xu, M.B., and Lee, I.J.**, "Analytical Approach for Sound attenuation in Perforated Dissipative Silencers", *J. Acoust. Soc. Am.* 115(5), 2004.
9. **Chen, K.T., Chen, Y.H., Lin, K.Y., and Weng, C.C.**, "The Improvement on the Transmission Loss of a Duct by Adding Helmholtz Resonators", *Applied Acoustics* 54, 1998, 71-82.

10. **Wilson, G. P. and Soroka, W.W.**, “Approximation to the Diffraction of Sound by a Circular Aperture in a Rigid Wall of Finite Thickness”, *Journal of the Acoustical Society of America* 37, 286-297.
11. **Galland, M.A., Sellen, N., and Hilbrunner, O.**, “Noise reduction in a flow duct by active control of wall impedance”, American Institute of Aeronautics and Astronautics.
12. **Poole, J.H.B., and Ieventhall, H.G.**, “An Experimental Study of Swinbanks' Method of Active Attenuation of Sound in Ducts”, *Journal of Sound and Vibration* 49, 1976, 257-266.
13. **Valimaki, V., Marko, A., Seppo, R., and Jukka, L.**, “Adaptive Noise Cancellation in a Ventilation Duct Using a Digital Signal Processor”, *DSP Scandinavia '97 Digital Signal Processing Conference*, 14.3.1997.
14. **Randolph, H.**, “A Principal Component Algorithm for Feedforward Active Noise and Vibration Control”, Blacksburg, Virginia, April, 1998.
15. **Mingsian R. Bai., Pingshun, Z.**, “Design of a Broadband Active Silencer using μ synthesis”, *Journal of Sound and Vibration* 269, 2004, 113-133.
16. **Munjal and Eriksson**, “ An Analytical , One Dimensional, Standing Wave Model of a Linear Active Noise Control System in a Duct”, *Journal of Acoustical Society of America* 84, 1988, 1086-1093.
17. **kang, S.W., and kim,Y.H.**, “Active Intensity Control for the Reduction of Radiated Duct Noise”, *Journal of Sound and Vibration* 201, 1997, 595-611.

18. **Jing Yuan**, “A hybrid active noise controller for finite ducts”, *Applied Acoustics* 65, 2004, 45-57.

19. **Jerome P. Smith, Brody D. Johnson, and Ricardo A. Burdisso**, “A broadband passive-active sound absorption system”, Vibration and Acoustics Laboratories, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Virginia.

APPENDIX A:

C program for FIR filter to remove sine signal using LMS algorithm

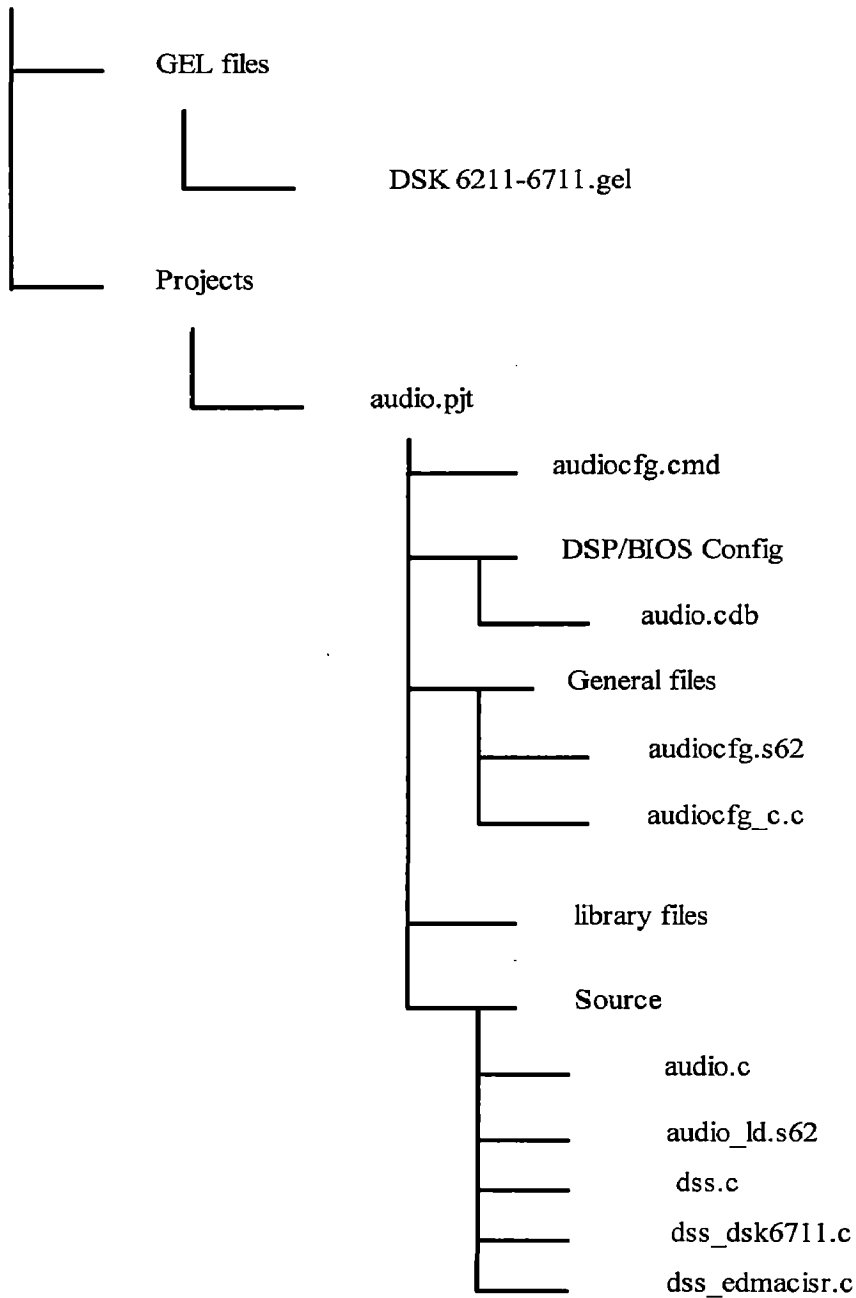
```
#include<stdio.h>
#include<math.h>
#define taps 32
#define PI 3.1415
#define MU 0.00049  main()
{
    float x[taps],w1[taps],w2[taps];
    float in,frac;
    float err,sum;
    int i1,curr=0,indx,i;
    FILE *fp1,*fp2;
    fp1=fopen("input.m","w");
    fp2=fopen("output.m","w");
    printf("\n Enter frequency of sine :");
    scanf("%g",&frac);
    for(i1=0;i1<taps;i1++)
    x[i1]=w1[i1]=w2[i1]=0;
    for(i1=0;i1<6000;i1++)
    {
        in=sin(frac*PI*i1);
        x[curr]=in;
        fprintf(fp1,"%g\n",in);
        sum=0;
        for(i=0;i<taps;i++)
        {
            if((curr-i)<0)
                indx=curr-i+taps;
            else
                indx=curr-i;
            sum+=w1[i]*x[indx];
        }
        err=in-sum;
        fprintf(fp2,"%g \n",err);
        for(i=0;i<taps;i++)
        {
            if((curr-i)<0)
                indx=curr-i+taps;
            else
                indx=curr-i;
            w2[i] =w1[i]+MU*err*x[indx];
        }
        for(i=0;i<taps;i++)
            w1[i]=w2[i];
        curr++;
        if(curr==taps)
            curr=0;
    }
}
```

```
fclose(fp1);  
fclose(fp2);
```

```
}
```

APPENDIX B: Arrangement and sequence of files of CCS working window

FILES




```

/*
 * Copyright 2001 by Texas Instruments Incorporated.
 * All rights reserved. Property of Texas Instruments Incorporated.
 * Restricted rights to use, duplicate or disclose this code are
 * granted through contract.
 *
 */
/* "@(#) DSP/BIOS 4.60.22 12-07-01 (barracuda-j15)" */
/*
 * ===== audio.c =====
 */

#include <std.h>

#include <pip.h>
#include <log.h>
#include <trc.h>
#include <hst.h>

#include <string.h> /* memset() */
#include <stdarg.h> /* var arg stuff */

#include "dss.h"
#include "audio.h"

#define PI 3.1415
#define MU 0.00049
#define taps 32

Int PHASE = 0; /* number of samples of phase difference */

static Uns allocBuf(PIP_Obj *out, Uns **buf);
static Void freeBuf(PIP_Obj *out);
static Uns getBuf(PIP_Obj *in, Uns **buf);
static Void process(Uns *src, Int size, Uns *dst);
static Void putBuf(PIP_Obj *out, Uns size);
static Void writeBuf(PIP_Obj *out, Uns *buf, Uns size);

static Int loadVal = 0;
static PIP_Obj *hostPipe = NULL;

/*
 * ===== main =====
 */
Void main()
{
    Int size;
    Uns *buf;

#if AUDIO_USEHST

```

```

hostPipe = HST_getpipe(&hostOutputProbe);
#endif

DSS_init(); /* Initialize codec and serial port */

LOG_printf(&trace, "Audio example started!!\n");

/* Prime output buffers with silence */
size = allocBuf(&DSS_txPipe, &buf);
memset(buf, 0, size * sizeof(Uns));
putBuf(&DSS_txPipe, size);

size = allocBuf(&DSS_txPipe, &buf);
memset(buf, 0, size * sizeof(Uns));
putBuf(&DSS_txPipe, size - PHASE);

/* Fall into BIOS idle loop */
return;
}

/*
 * ===== audio =====
 */
Void audio(PIP_Obj *in, PIP_Obj *out)
{
    Uns *src, *dst;
    Uns size;

    if (PIP_getReaderNumFrames(in) == 0 || PIP_getWriterNumFrames(out) == 0) {
        error("Error: audio signal falsely triggered!");
    }

    /* get input data and allocate output buffer */
    size = getBuf(in, &src);
    allocBuf(out, &dst);

    /* process input data in src to output buffer dst */
    process(src, size, dst);

    /* check for real-time error */
    if (DSS_error != 0) {
        LOG_printf(&trace,
            "Error: DSS missed real-time! (DSS_error = 0x%x)", DSS_error);
        loadVal -= 1;
        DSS_error = 0;
    }

    /* output data and free input buffer */
    putBuf(out, size);
    freeBuf(in);
}

/*
 * ===== error =====
 */
Void error(String msg, ...)

```

```

{
va_list va;
va_start(va, msg);

LOG_error(msg, va_arg(va, Arg)); /* write error message to sys log */
LOG_disable(LOG_D_system);      /* stop system log */

for (;;) {
; /* loop for ever */
}
}

/*
* ===== load =====
*/
Void load(Arg Arg_prd_ms)
{
static int oldLoad = 0;
Int prd_ms = ArgToInt(Arg_prd_ms);

/* display confirmation of load changes */
if (oldLoad != loadVal) {
oldLoad = loadVal;
LOG_printf(&trace,
"load: new load = %d000 instructions every %d ms", loadVal, prd_ms);
}

if (loadVal) {
AUDIO_load(loadVal);
}
}

/*
* ===== step =====
*/
Void step(void)
{
static Int direction = 1;

if (loadVal > MAXLOAD) {
direction = -1;
}
if (loadVal <= 0) {
direction = 1;
}

loadVal = loadVal + (100 * direction);
}

/*
* ===== allocBuf =====
*/
static Uns allocBuf(PIP_Obj *out, Uns **buf)
{
PIP_alloc(out);
*buf = PIP_getWriterAddr(out);
}

```

```

    return (PIP_getWriterSize(out));
}

/*
 * ===== freeBuf =====
 */
static Void freeBuf(PIP_Obj *out)
{
    PIP_free(out);
}

/*
 * ===== getBuf =====
 */
static Uns getBuf(PIP_Obj *in, Uns **buf)
{
    PIP_get(in);
    *buf = PIP_getReaderAddr(in);
    return (PIP_getReaderSize(in));
}

/*
 * ===== process =====
 */
static Void process(Uns *src, Int size, Uns *dst)
{
    float tmp_value[32],x[32];
    static float x[32],w1[32],w2[32];
    float inp,err,sum;
    int indx,i;
    static int curr;

    for (i = size - 1; i >= 0; i--)
    {
        x [i] = (float)src[i];
    }

    while(size--)
    {
        inp=src[i];
        x[curr]=inp;
        sum=0;
        for(i=0;i<taps;i++)
        {
            if((curr-i)<0)
                indx=curr-i+taps;
            else
                indx=curr-i;
            sum+=w1[i]*x[indx];
        }

        err=inp-sum;

        for(i=0;i<taps;i++)
        {
            if((curr-i)<0)

```

```

        indx=curr-i+taps;
    else
        indx=curr-i;
        w2[i] =w1[i]+MU*err*x[indx];
    }
    for(i=0;i<taps;i++)
        w1[i]=w2[i];
    curr++;
    if(curr==taps)
        curr=0;

    dst=(Uns)sum;

} /* if hostPipe is non-NULL write data to this pipe */
if (hostPipe != NULL) {
    writeBuf(hostPipe, dst, size);
}
}

/*
 * ===== putBuf =====
 */
static Void putBuf(PIP_Obj *out, Uns size)
{
    PIP_setWriterSize(out, size);
    PIP_put(out);
}

/*
 * ===== writeBuf =====
 */
static Void writeBuf(PIP_Obj *out, Uns *buf, Uns size)
{
    Uns dstSize;
    Uns *dst;
    Int i;

    /* allocate outBuffer for output */
    dstSize = allocBuf(out, &dst);

    /* copy data to output buffer (but not more than output buffer size) */
    for (i = (dstSize >= size) ? size : dstSize; i > 0; i--) {
        *dst++ = *buf++;
    }

    /* put buffer to output pipe */
    putBuf(out, size);
}

```