A NOVEL APPROACH TO MULTIPLE REFERENCE FREQUENCY DOMAIN ADAPTIVE CONTROL

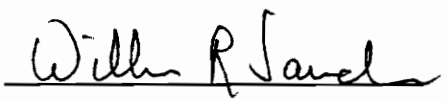
by

Michael A. Vaudrey

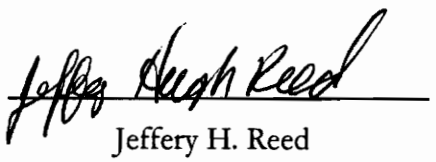
Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

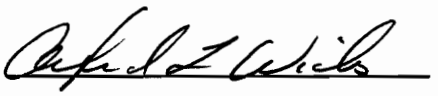
APPROVED:



William R. Saunders, Chair



Jeffery H. Reed



Alfred L. Wicks

December 19, 1996

Blacksburg, Virginia

Keywords: Coherence, Frequency Domain Adaptive, Multi-Input, Active Noise Control

A NOVEL APPROACH TO MULTIPLE REFERENCE FREQUENCY DOMAIN ADAPTIVE CONTROL

by

Michael A. Vaudrey

William R. Saunders, Chairman

Mechanical Engineering

(ABSTRACT)

Adaptive feedforward control of any physical system, acoustical, vibrational or other, requires what is termed as an uncontrollable coherent reference signal. That is, a signal which is highly representative (coherent) of the disturbance to be controlled which is not affected by the control actuator itself. Creating the *coherent* portion of this requirement for a certain class of problems is the motivation of this work.

Most physical disturbances do not originate from a single source, but rather maintain contributions from a number of (possibly) correlated paths. For engineers who have access to only a single-input single-output (SISO) adaptive controller, the multi-source disturbance presents a difficult design issue. Simply adding the references in a linear combination can result in a signal which is not coherent at any frequency. Appropriately amplifying and suppressing coherent and incoherent signals prior to their linear combination can result in a signal which accurately represents the disturbance at all frequencies. This is precisely the task that the newly developed coherent output power (COP) filters perform.

By calculating the coherent (or partial coherent) output power of each of the candidate references before control occurs, frequency domain filters are designed to remove incoherent portions of each signal. The advantages of performing the COP filtering procedure are very apparent when compared to the simple linear combination of signals. Coherence, and thus control performance, can be drastically improved. The COP filtering technique offers a means for system identification and computational savings not apparent in the conventional adaptive array, which solves the same multi-source problem.

ACKNOWLEDGEMENTS

Without the continuing patience and encouragement from my fiancée Amanda Snow, this thesis never would have been possible. I am eternally grateful for everything she has given me.

I would like to thank my parents (Walt and Sandra) and my sister (Carol) for all of their support during the last twenty months. They helped me maintain a positive attitude in the face of many hardships and never stopped believing that I would always reach my goals.

In the trenches, Will Saunders deserves all the credit for daily affirmation and technical support. His insight, positive attitude, and immutable interest have made this work both possible and enjoyable.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

With the recent developments of high speed sophisticated Digital Signal Processing (DSP) hardware systems, adaptive filtering and control technologies have become increasingly more realizable. Applications of Bernard Widrow's Least Mean Square algorithm [1] are widespread. System identification, linear predictive coding, adaptive line enhancement, active vibration control, and adaptive beamforming are only a few of the practical implementations of adaptive filtering and control techniques.

Continuing advances in adaptive filtering techniques have made active noise control (ANC) a more practical technology. While the developments contained herein consistently refer to specific implementation in ANC, the results can be applied to various other applications exhibiting similar design requirements.

1.1 MOTIVATION - THE MULTI-SOURCE PROBLEM

The physical behavior of sound can be understood by examining the mechanisms of wave propagation. For a single tonal noise one can relate its three dimensional propagation in air to a two dimensional wave generated in a pond by throwing a rock into its center. Even more rudimentary, a one dimensional sine wave can be used to represent a tonal noise. Summing with it a sine wave that is equal in magnitude and 180° out of phase, results in pure "cancellation" of the original sine wave. This is the basis of ANC. By introducing into the physical environment, a secondary noise source with the ability to generate an equal in magnitude opposite in phase noise field, the existing noise field can be canceled.

Active noise control techniques are most effective in environments which contain primarily low frequency disturbances. Due to hardware limitations and acoustical physics, ANC's

high frequency effectiveness tends to deteriorate in the frequency range exceeding 1 kHz. Typical passive measures have a high pass characteristic which depends on cost and spatial orientation of the sound absorbing material. There are numerous references (including [4, 5]) which examine noise control methodologies and their respective effectiveness.

There are two distinct methods of *active* noise control, feedback and feedforward. The application of this work is directed toward *adaptive feedforward* control only. While feedback noise control is effective for some applications, the structure and methodology differs notably from that of feedforward and thus is not applicable to this study. Adaptive feedforward control in general, and specific application to noise control, are discussed in detail in Chapter 3.

The physics of ANC and acoustic wave propagation make the seemingly simple principle of active noise control much more difficult to implement in practice. (Further details of these phenomena are provided in Chapter 3 and [2, 4]). One of the primary physical limitations in performing adaptive feedforward ANC is the necessity of having an uncontrollable reference signal which resembles the undesirable noise field. "Resembles" is a qualitative word which has been quantified by a signal processing measure termed *coherence.* The thrust of this work presents a tool which can be used to create this coherent reference.

To develop this work in the light of practical applications, two specific physical examples have been chosen. The first is a common piece of office equipment which generates annoying levels of noise in an enclosed environment, a dot-matrix printer. Although laser and ink-jet printers have nearly eliminated the market for these types of printers, their high volume capabilities continue to find a market niche in low-quality high speed applications. The second practical application (which is introduced next) is that of a common automobile. Noise control, both internal and external, is a continuing research initiative with many car manufacturers.

For a given piece of machinery, or noisy environment in general, the cause of the noise can not typically be attributed to a single noise source. An automobile in motion can generate

tire noise, fan noise, and engine noise from the combustion within the cylinders. Each of these sources contributes to the overall sound pressure level arriving at the operators' ears. If a feedforward noise control approach only used information from the engine (reference) to cancel the entire noise field, the tire noise and fan noise would remain. To further complicate this problem, it should be clear that the fan noise as well as the tire noise will be somewhat related to the engine noise, since it is the rotation of the drive shaft that is coupled to the rotation of the fan and movement of the automobile.

If it were possible to extract only the useful information from each suspected noise source signal, these pieces could be linearly combined to create a single coherent reference which resembles all of the undesirable noise. By examining each of the signals independently as well as with respect to one another, it is possible to represent each signal's independent contribution to the overall noise. This is typically termed the "energy source identification problem" in the signal processing community [3].

1.2 GENERAL PROCEDURE

A large number of active noise control problems can be cast in the form of the energy source identification problem. Figure 1.1 illustrates the physical layout of this type of problem for a general system. The measured output represents the disturbance noise which the engineer desires to cancel. For the automobile noise control problem this would be the interior noise measured at the user's ear. Continuing from right to left in Figure 1.1, the measured output originates from many independent (or possibly correlated) sources after they have been altered by some physical plant, denoted with transfer functions subscripted by "p". These sources can be thought of as the fan noise, the engine noise, and the tire/road noise. The physical plants associated with each of these represent any alterations of these signals via acoustic or structural paths, before they arrive at the user. The summation of all of these sources occurs at the output (microphone) by the physical process of sound pressure level summation. If it were possible to independently measure each of the outputs, the problem would be cast in a different format than the MISO system here. Each of the sources in Figure 1.1 is then measured by some electronic transducer

(subscripted with "e"), each of which have dynamics associated with them represented by the associated transfer functions.

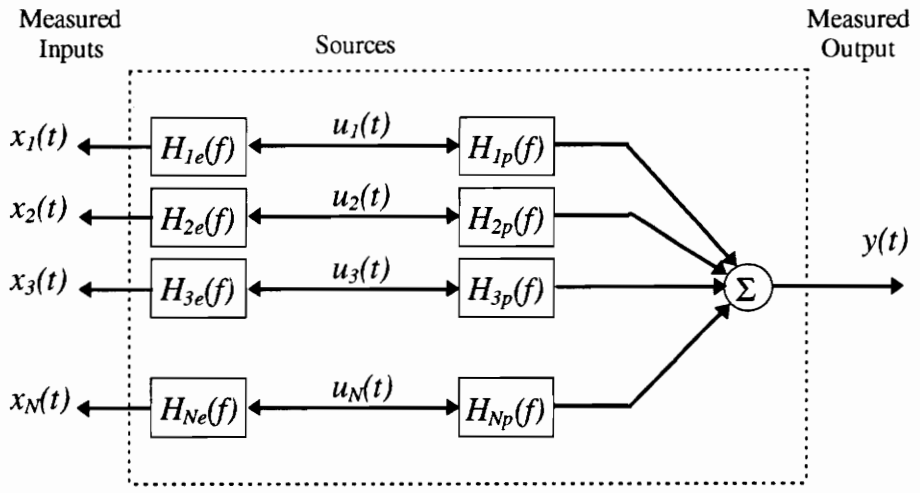


FIGURE 1.1 ENERGY SOURCE IDENTIFICATION PROBLEM

Any physical system or control problem that can be reformulated to fit this structure, can benefit from the technology described herein. There are several variations of adaptive feedforward control which can now be used in an attempt to control the measured output as a function of the measured inputs.

Before discussing these methods, a brief introduction regarding adaptive control and the *coherence* function is necessary. Figure 1.2 shows a simplistic interpretation of active noise control achieved using adaptive feedforward methods.

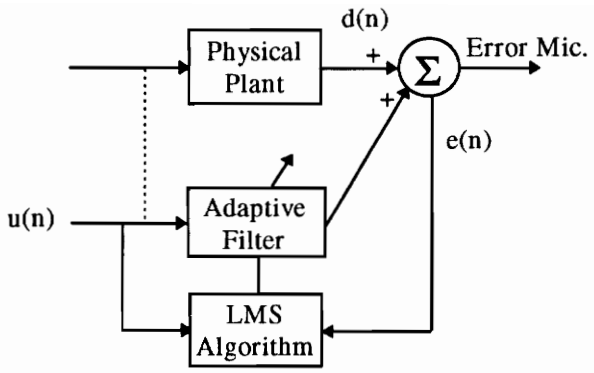


FIGURE 1.2 ANC VIA ADAPTIVE FEEDFORWARD CONTROL

The input to the adaptive filter and LMS algorithm is (hopefully) similar to the input to the physical plant, as indicated by the dotted line. This path corresponds to a horizontal path in Figure 1.1 without the electric transducer transfer function included. To determine if this path can accurately represent the input to the physical plant, a measure of system linearity exists called the *coherence function.* Measuring the coherence function between the output of the plant and the candidate reference signal *u(n)*, a determination can be made concerning the effectiveness of the adaptive controller which conforms to Equation 1.1 [2]. The uncontrolled plant output is also known as the disturbance noise or the error microphone signal e(n).

$$\frac{S_{ee}}{S_{dd}} = 1 - \gamma_{ud}^2 \qquad \textbf{(1.1)}$$

The coherence function, represented by γ^2, is a frequency dependent measure of system linearity which varies between zero and one, one indicating perfect linearity. Should the coherence between the reference signal and the disturbance noise be unity, the autospectrum of the error signal, S_{ee}, will approach zero as desired. If the coherence between the two signals is zero, the error will remain unchanged from the original disturbance and no effective control will be generated. With this qualitative interpretation of coherence and its relation to adaptive noise control, several methods of control can be described.

The adaptive array structure uses a bank of parallel adaptive filters to receive each of the measured inputs as a separate reference signal. The outputs of the adaptive filters are summed to form a single control signal. With the overall goal being to minimize this global error, each adaptive filter receives the single error signal and updates the weights based on the LMS algorithm. This methodology is thoroughly investigated in regard to adaptive antenna arrays [60]. The adaptive array has also been applied to noise control and has been discussed by Elliot and Nelson in some detail [2]. While this is a valid technology, it is not the goal of this work to analyze adaptive arrays in great detail. It is necessary however, to compare and contrast performance of the proposed method relative to alternative methods as well as with respect to the solutions obtained with the developments in this work, since they each effectively solve the same problem.

Shown in Figure 1.3 is an alternative to the adaptive array which greatly reduces the number of computations and complexity of the adaptive array control scheme.

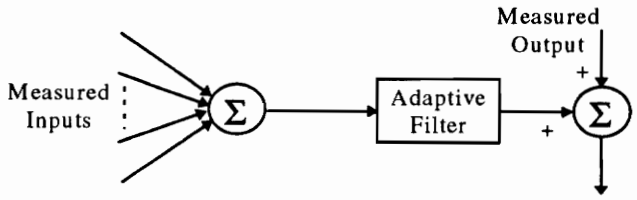


FIGURE 1.3 SIMPLE REFERENCE SUMMATION

All of the measured inputs, assumed to be responsible for portions of the measured output, are summed to be used as the input to a single-input single-output (SISO) adaptive controller. While this reduces the complexity from the adaptive array concept, the simple summation can be very detrimental to the creation of a coherent reference. Reasons for this will be explained in Chapters 4 and 5.

The innovative methodology which is the basis of this work represents a more mathematically correct alternative to Figure 1.3 and the adaptive array solution. Coherent output power (COP) filtering: 1) is introduced to improve the performance of simplified MISO systems and 2) uses information from the measured output of the energy source

identification problem and from all of the measured inputs to correctly combine them to form an optimally coherent reference.

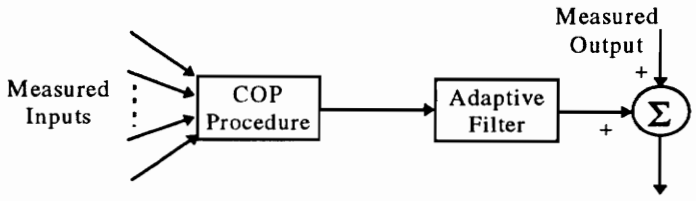


FIGURE 1.4 COP FILTER PROCEDURE

This method holds many practical advantages from a system identification standpoint as well as for achieving less computationally complex control. By removing only the portions of each of the inputs which are directly related to the output, the signals can be linearly combined to form a reference which is completely correlated with the measured output. This procedure can also be interpreted as an orthogonalization process where the measured inputs are "rotated" to form a linearly independent basis set to span the space of the measured output. Singular value decomposition and the Graham Schmidt procedure are well known methods of performing these operations on matrices and multidimensional spaces.

Given the basic motivation and ideas behind the development of the COP filtering procedure, each of the individual chapters will be outlined.

1.3 SCOPE

The development and implementation of these ideas require a multidisciplinary approach. The main structure includes analog signal processing, correlation and spectral analysis, digital signal processing (DSP), adaptive signal processing and control; and in an application environment, acoustics and active noise control. The experimental implementation requires an understanding of programming DSP hardware. Thoroughly investigating each of these fields is clearly beyond the scope of this thesis. However, specific tools from each of these areas of study will be applied to the development of the methodology contained herein.

With the overview provided in Section 1.2, Chapter 2 explores literature from each of these fields which pertains directly to the problem under investigation or the tools used in this work. There are seemingly endless journal articles, conference papers, and correspondences detailing many aspects of each area of study under investigation. The references addressed in Chapter 2 are believed to give an adequate knowledge base sufficient enough to develop these ideas. The references also represent current advances and studies pertaining to methodologies which may meet the same ends through different means or suggest alternative tools which may be applied to these developments. Most importantly, the literature reviews bears out the uniqueness of the proposed COP/MISO control design.

As will be examined throughout this study, adaptive arrays represent a mature field of study which meets a similar goal through very different means. This work will investigate adaptive arrays to the extent of making an educated comparison with the developed methods. It is not intended to be an authoritative examination of adaptive arrays, merely an introduction for the purposes of comparing and contrasting the similarities and differences between adaptive arrays and the COP filtering procedure. The literature review of this subject area will mirror these goals.

As was briefly mentioned, active noise control can be affected via two fairly different schemes, feedback and feedforward. Since the methodologies contained herein should not be applied to feedback control techniques, no further mention of them will be included. Chapter 3 provides a discussion of the tools necessary for achieving active control of sound through adaptive *feedforward* technologies only. This chapter includes some basic mathematical development of the LMS algorithm, qualitative discussions, and MATLAB simulations of several aspects of noise control. The filtered reference is introduced along with adaptive arrays. The bulk of this chapter is devoted to developing the necessary ideas in order to present the concept of Frequency Domain Adaptive filtering. This method provides valuable computational advantages when used in conjunction with the COP filters.

Throughout Chapter 3 and the remainder of this work, assumptions regarding basic understanding of block diagram algebra and Fourier analysis are made. Any feedback

controls and signal processing textbooks will have each of these areas described in sufficient detail.

In order to design the filters mentioned in Section 1.2, an in-depth understanding of analog signal processing and coherence analysis is required. Chapter 4 defines necessary terminology and reviews the detailed mathematics behind coherence, partial coherence, and multiple coherence. These principles are then applied to several special case studies which will be used for the remainder of this work. Each noise control problem is a specific case and must be treated as such. It is difficult in practice to have a system which works in all environments under all conditions. To predict every possible scenario which could be encountered is beyond this study. Therefore a few common and specific cases will be examined, specifically the multi-input single-output (MISO) energy source identification problem and variations thereof. Careful interpretation of the results will allow the reader to create his or her own solution, tailored specifically to the task at hand.

Having developed the necessary tools for solving the coherent reference problem, Chapter 5 presents the bulk of innovative work. The coherent output power filtering methodology is introduced and explained in detail. Applying the tools described in Chapters 3 and 4, a design algorithm is then developed. Applications to system identification and physical interpretations are explained as comparisons are made to simple source summations and the conventional adaptive arrays. MATLAB simulations as well as mathematical proof verify the results explained in Chapter 5.

To further demonstrate the effectiveness of the COP filter method for coherent source development, an experimental application has been developed. Chapter 6 describes the hardware and software used in demonstrating the active noise control of a two tonal complex in a laboratory setting.. Finally, Chapter 7 summarizes the work and presents ideas for future advances in this area. Many improvements and further in-depth analyses can be derived from the results generated herein.

Chapter 2

Review of Germane Literature

This chapter is intended to make the reader aware that a wealth of literature exists in each of the relevant subject areas pertaining to this work. Only a small portion of the most applicable and meaningful references are included in this study. It should provide an adequate summary of the prior work done in the areas of adaptive signal processing, analog signal processing, and multi-channel active noise control. It will also become evident that the COP filter design methodology and implementations have not previously been developed, described, or implied in the context of any known publications.

Active noise control (ANC) in and of itself is a mature and well defined field. Although physical behavior of acoustical waves prevents many desirable applications of ANC from being entirely realized, implementation of ANC has found several successful industrial applications [2]. Multiple-input single-output (MISO) systems have been investigated to some extent in the ANC community and to a greater extent in the form of adaptive antenna arrays. Since the COP filter technology focuses on solving problems encountered in MISO systems, the literature pertaining to solving these types of problems will be of prime concern.

As was briefly mentioned in Chapter 1, the COP filter design is based on the properties of each individual input relative to each other and the output. Thus a great deal of detailed signal processing must take place before and during implementation. The highly relevant areas within signal processing involving the MISO system structure include data orthogonalization and the energy source identification problem. Each of these are tools which can be used to design the COP filters. However, it is not evident in any of the literature, that a connection has been made between the signal processing and signal modification prior to this work.

2.1 MISO ADAPTIVE FILTERING

Some terminology surrounding adaptive feedforward control as applied to ANC will be summarized with respect to Figure 2.1 before proceeding. A strict differentiation must be made between *feedback* ANC and *feedforward* ANC. Figure 2.1 illustrates the general structure of feedforward ANC using the LMS algorithm developed by Bernard Widrow [1]. This is considered to be a feedforward controller because the error signal e(n) does not directly modify the input u(n).

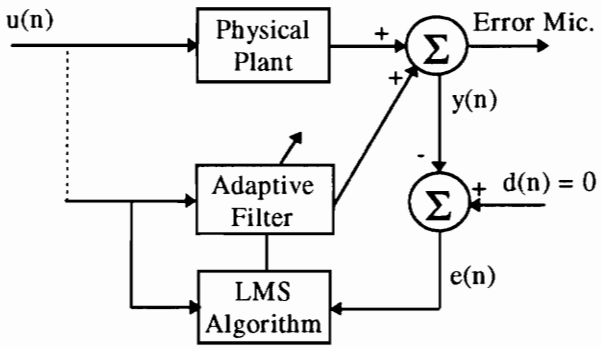


FIGURE 2.1 PRACTICAL IMPLEMENTATION OF ADAPTIVE FEEDFORWARD ANC

A feedback arrangement involves a conventional PID or optimal feedback controller which uses the error signal to modify the control signal in the context of the disturbance rejection problem [2]. This type of structure is not considered in the implementation and design of the COP filters.

The necessary terminology inherent in Figure 2.1 involves the nomenclature of the signals and their behavior with respect to one another. The *reference signal* u(n), is considered to be the input to the adaptive filter as well as the physical plant which is generating the undesired sound pressure or *disturbance* (shown as the unlabeled output of the physical plant). Ideally this reference will be highly correlated (or coherent) with the disturbance thus allowing the adaptive filter to simply modify the amplitude and phase via the LMS algorithm. The unlabeled output of the adaptive filter is termed the *control signal.* The control signal and disturbance noise meet at the error microphone which produces the signal y(n) when the SPL's are summed. In the algorithm, the error signal is generated by subtracting y(n) from

the desired signal d(n). In active noise control where the goal is to reduce the overall sound arriving at the error microphone, the desired signal is always zero. Thus by reducing the *mean square* error, the LMS algorithm is reducing the SPL at the microphone (y(n)) by updating the adaptive filter.

2.1.1 ACTIVE NOISE CONTROL

The terminology of MISO systems when related to ANC can refer to several different configurations of Figure 2.1. Often, MISO or MIMO adaptive control configurations refer to increasing the number of error microphones and number of control speakers. This is typically done in attempts to yield more global control of sound as in [2, 22, 23]. This configuration requires that cross paths between each of the microphones and control speakers be taken into account. This represents a viable solution to a different problem than that which is solved by the COP filters.

Multiple *reference* control is often termed *multi-channel* adaptive control to differentiate it from the MIMO control achieved by multiple error microphones and speakers. For the implementation of the COP filters herein, the end result is a multi-channel system which requires control by a SISO adaptive controller as shown in Figure 2.2.

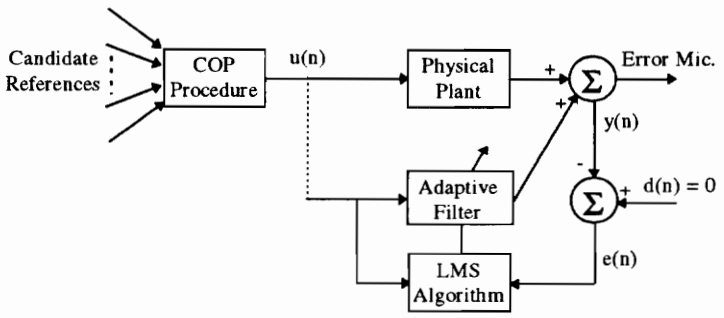


FIGURE 2.2 PRACTICAL IMPLEMENTATION OF COP ANC

There is no reason why additional microphones and control speakers cannot be added to the structure in Figure 2.2, each using u(n) as a coherent reference. The differences in the MISO system handled by the COP design and the MISO or MIMO systems of global noise control should be apparent and differentiated. From this point on, the MISO system will

refer to the multiple-reference, single-input single-output controller as shown in Figure 2.2. Computational and practical reasons for approaching the problem in this manner will become apparent in Chapter 5.

In addition to approaching global control, the microphone/control speaker array system has been thoroughly investigated by a number of researchers. Zelinski, Allen *et al.*, Martin, Vary, Laugesen, and Elliott have all independently discussed this methodology for reducing acoustic wave reverberations in "live" rooms [24 - 27]. The same approach has been investigated by Boll *et al.* and Simmer *et al.* in order to enhance speech in the presence of unwanted noise [28 - 30]. None of these studies directly address the multiple reference problem although some imply multi-channel control is achievable via adaptive arrays. This topic is discussed more thoroughly in Section 2.1.3.

The most relevant points regarding the implementation of the COP filters are found in Guicking and Bronzel's "Multi-Channel Broadband Active Noise Control in Small Enclosures"[31]. Their work also provides a justification and "launching point" for the work which was done in developing the COP filters. Namely, they mention that prior to any implementation "...one should analyze the noise generation mechanism, and particularly identify the relevant primary sources." As will be shown, this is precisely the automated pre-processing which is performed by the COP filters.

2.1.2 Frequency Domain Adaptive Filtering

Filtering in the frequency domain offers a computational savings over filtering in the time domain for large order filters. Since the COP filters themselves are designed in the frequency domain, it is intuitive and practical that rather than converting back to the time domain, the adaptive filtering portion should also be done in the frequency domain.

This particular methodology is nearly as old as the LMS algorithm itself, dating back to 1978 [32]. Other analyses of this technology are also provided in [33 - 37]. John Shynk summarized much of the previous work done in frequency domain adaptive filtering in [38]. Given the obvious computational advantages, it is surprising that more ANC applications

do not employ this approach. In order to effectively control broadband noise, large filter orders are required. Typically the limitation in the performance of the traditional LMS algorithm implemented using large filter orders is seen in the lack of computational horsepower of DSP's to handle the large number of multiplication's. The frequency domain LMS offers a significant computational savings as will be seen in Chapter 3.

The uniqueness inherent in this work does not originate from the possibility of frequency domain adaptive control as applied to ANC. Kijimoto *et al.*, Shen, Deisher, and Koers have all presented work in this area as seen in [39 - 42]. However, these are among the few applications existing which examine the frequency domain approach to active noise control. Reasons for this may include the complexity required in implementing frequency domain convolution and correlation operations as well as the overwhelming popularity and ease of use of the conventional time domain LMS algorithm.

2.1.3 ADAPTIVE ARRAYS

The adaptive array is the LMS implementation of the multi-reference problem which the COP filter also addresses. The general structure of an adaptive array using multiple references, a single control speaker, and a single error microphone is shown in Figure 2.3.

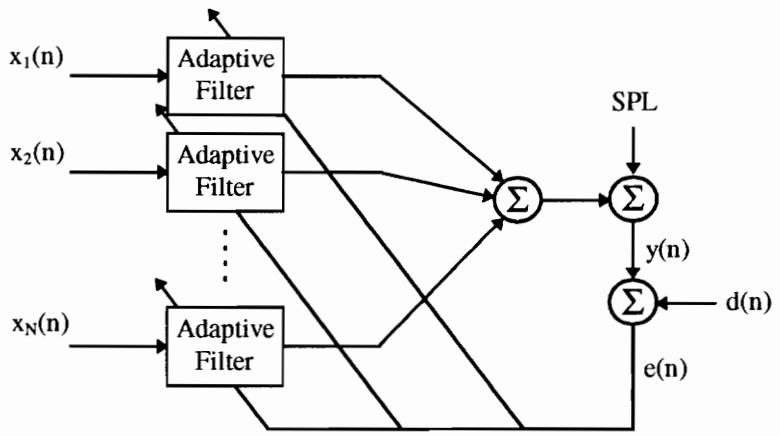


FIGURE 2.3 ADAPTIVE ARRAY

Among the first researchers to develop and investigate this type of structure was Sydney Applebaum [43]. The motivation for the development of this structure was in order to

optimize the signal to noise ratio (and thus the coherence) in an antennae array. The structure of the adaptive antennae array (AAA) is *not* shown in Figure 2.3. The implementation of an AAA is slightly different since the reference signal(s) are not directly obtainable, they are often created from the error signals incident on each antenna. While much investigation of the AAA [44 - 48] has indirectly contributed to the development of a similar structure applicable to ANC, the antennae array itself is not considered relevant to the COP filter implementation.

The structure in Figure 2.3 is the application of the AAA to the noise control problem. The separate inputs (references) are indicated by $x_N(n)$. Each of these references may be coherent to some extent with the disturbance (SPL). Elliot and Nelson are among some of the only researchers to discuss the application of this structure to active noise control in detail [2]. Since the multi-channel algorithm provides a method to achieve the same goal as the COP filter implementation, it is necessary to investigate the similarities and differences as well as the advantages and disadvantages of the methods when compared with each other. These comparisons are made in Chapter 5.

2.2 SIGNAL PROCESSING

The development of the COP filters relies heavily on the ability to perform accurate signal processing on stationary ergodic data. Much of the actual signal processing performed in order to design the COP filter itself has been fully developed by Bendat and Piersol and adopted here in the context of the active noise control problem. The actual implementation of the COP filter idea has not been discussed.

2.2.1 GENERAL

Bendat and Piersol have produced many pieces of literature both jointly and independently, that have become the basis for spectral analysis of random data [3, 6, 8, 49]. While they are considered the experts in the field of correlation analysis, their literature typically only mentions applications to system identification, rarely control. Using their developments in

the context of adaptive feedforward control permits accurate assessment of the ability to perform effective control in any given system.

Talbot effectively analyzed the reliability of phase determination between two signals based on the coherence at a given bin [50]. Suppose the coherence between two signals was known to be less than unity (i.e. non-ideal). The system identification of the frequency response function (FRF) is based on the cross-spectrum and the autospectrum of the input to output and input, respectively. If the coherence is non-unity, there is some definable error that will be present in the FRF phase measurement. Talbot statistically analyzes the effects of changing coherence on the FRF phase. His results are able to quantify the phase difference (or error in actual vs. measured phase) with respect to the measured coherence function. This essentially establishes "a criterion for selecting coherence peaks which were significant but which were not so small as to cause a large variation in the phase difference between the associated signals". (The magnitude criterion is established in Chapter 4).

This study is applicable to examining the reliability of the coherence measurement in the context of system identification but is only indirectly related to the COP filtering procedures. Ultimately the magnitude of the coherence function determines the amount of control achieved in a given situation. Thus phase determination is not of strict concern since the goal is simply to maximize the coherence function.

2.2.2 Energy Source Identification

The signal processing problem which is solved in order to determine the COP filter design is commonly termed the energy source identification problem [3]. Figure 2.4 illustrates the general structure of this type of problem.

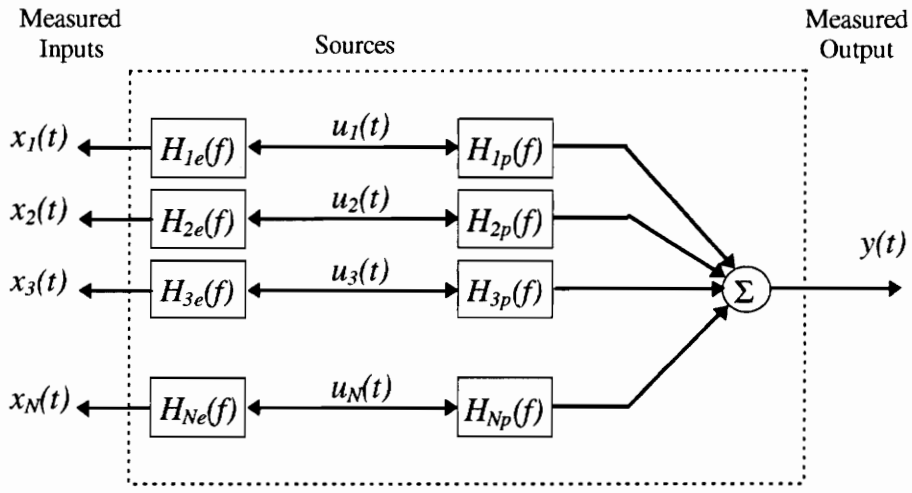


FIGURE 2.4 ENERGY SOURCE IDENTIFICATION PROBLEM

Essentially, the original sources, *u(t)*, are being measured through some electrical transducer transfer functions subscripted by *e*. The sources also experience some physical process denoted by the subscripts *p*. The result of this process is the measured output which is the sum of all the physical process which may be thought of as the disturbing acoustic noise. There is only one output because it is impossible to differentiate the physical processes from each other once they have become acoustic (or other combinational) energy. However, it is possible to differentiate these signals from each other via the measured *inputs*.

The energy source identification problem is to determine the energy contribution each input has to the overall measured output. These individual powers are then defined as coherent output powers (COP). Bendat and Piersol effectively describe the generalized solution to these types of problems in [3].

In 1977, Alfredson investigated the use of these techniques as applied to a diesel engine [51]. The goal was to identify the significant radiating surfaces contributing to the overall acoustic noise. His results were inconclusive. It is not evident that he took into account the directionality existing between the cross paths of the multiple inputs that he was examining. **This is a key element in the development of an accurate solution to this problem.** Wang *et al.* also approached the same problem in a similar manner in 1983 [52]. Although

they were familiar with (and referenced) Alfredson's prior work, they also approached the problem without taking the direction of the cross paths into account. Their estimates for the coherent output power of each input are clearly incorrect when examined with respect to [3].

The cross path direction estimate itself is a difficult task and can result in incorrect results if the wrong direction is assumed. Several methodologies have been proposed to solve this problem. Bendat and Piersol propose that ordering the inputs based on the magnitude of the ordinary coherence function of each input with respect to the output can provide an estimate of cross-correlation direction. The concept is that if the contribution (at a frequency) of a source one has a more linear system to the output than that of a source two, any contamination of the input will likely be *from* one *to* two. (i.e. a system is more likely to exist *from* one *through* two to the output than *from* two *through* one to the output). This can clearly result in inaccurate estimates.

Park and Kim developed a more mathematically sound and intuitive method for estimating the frequency dependent cross path direction based on causality checking [53]. The idea they develop is that the direction of the cross path can be determined by examining the impulse response of the system. If it demonstrates an acausal response, the wrong direction has been assumed. This is a much more accurate method of determining input priority but can become computationally intensive when examined for multiple inputs over a wide range of frequency bins. (The cross path direction can be frequency dependent).

These developments all represent the types of tools required to perform the COP filter design and analysis. However, no mention is ever made regarding using the coherent output power as a method of improving the overall coherence of a signal to be used in control. The concept of using this information in order to design a filter seems unique and innovative. Although the applications of this technology are focused around improving the coherence of a MISO system signal, the idea of altering a signal based on its spectral content characteristics may be applicable to many other platforms.

2.3 OTHER RELEVANT WORK

Since it has been established that the measure of coherence of a reference signal to the disturbing noise is directly related to the effectiveness of control, it is not surprising that the relationship has received considerable attention. Three categories of research related to the COP filter innovations can be classified as not exclusively being focused on either adaptive control or signal processing. Being a multi-disciplinary development, these references hold special relevance to the COP filtering ideas. These categories include tools which may permit efficient calculation of the COP filters, methods which achieve the same ends through similar means, and work which provides justification for the techniques pursued.

In 1978 Owsley described a method of adaptive orthogonalization of data records, essentially an approach used to determine the power spectrum adaptively [54]. This is in no direct way related to the COP filtering technology but may provide a method to determine the necessary spectral measures adaptively, thus avoiding expensive averaging operations. Lindquist and Haas developed a similar scheme which determines the autocorrelation and cross-correlation functions of data sets adaptively [55]. Each of these developments imply the joining of spectral analysis and adaptive filtering, but only in the context of analysis and not direct modification of the signal, or even control. They can be viewed as tools which may be used in design of the COP filters as opposed to the traditional averaging of FFT operations discussed in Chapters 4 and 5.

In a practical sense, the COP filtering procedure creates orthogonal (or perpendicular) components of the input signals so that each has its own unique contribution. In 1993, Subbaram and Abend presented a method for orthogonalizing input signals via singular value decomposition (SVD) [56]. As applied to adaptive antenna arrays, this methodology involved a completely different structure, as the error signals were used to generate the reference signal. However, as mentioned regarding Figure 2.3, the adaptive antenna array can be adapted to suit the ANC problem. This implies that Subbaram's methods can also be used to generate orthogonal signals in ANC. This was not specifically considered as it is beyond the scope of this work. The goal is to introduce a new technology which provides a

practical interpretation of the coherent reference(s), not implement prior work in a new field.

Gelle presents yet another approach which can be used to generate a coherent reference [57]. He proposes that a phase locked loop can be applied to the traditional error signal for each harmonic component in order to create a coherent reference signal. This is strictly different from the energy source identification problem for which the COP filters provide a solution. Multiple partially coherent references are assumed to exist whereas generating the reference from an error signal is neither necessary nor desired. Practically speaking, the error microphone signal will be reduced by the adaptive filter thus reducing the signal power used to generate the reference. At some point the reference will become unacceptably low and the performance at the error microphone will be less than optimal.

If there were any technical doubts or misgivings about the ability to determine system linearity via the coherence function, work done by Patton *et al.* and Lou *et al.* will suppress them [58, 59]. Patton investigated the coherence function and how it pertains to accurate determination of the frequency response function. While this has very little relationship to the implementation of the COP filters themselves, it provides justification that coherence is the correct measure to be using to identify system linearity. If the system existing between the reference signal and the disturbance noise is linear, the coherence will accurately indicate that fact. Lou also arrives at similar results when comparing the system identification and coherence function using point spectra versus using other orthogonalization techniques. It was determined that point spectra (or DFT) present an accurate measure of system linearity as applied to the coherence function.

It is evident that the combination of signal processing and adaptive control is required in order to achieve acceptable performance. Analysis studies and various implementations of orthogonalizing data have shown this to be the case. Since it has been well established that system linearity can be accurately predicted by appropriate coherence measurements, and adaptive feedforward control is a function of coherence, it is only intuitive that the two areas can be combined to create the COP filter solution. Prior work in these areas

undoubtedly indicates that the COP filters represent a valid solution to the energy source identification problem. However, it is also apparent that no one has yet developed this particular technology and implemented it in an adaptive controller of any kind, ANC or otherwise.

Having analyzed the representative literature in several pertinent fields, the COP filter development will be described. As mentioned earlier, the design requires a detailed understanding of adaptive filtering as well as analog and digital signal processing. These two areas are discussed in necessary detail before describing the COP filters themselves in Chapter 5.

Chapter 3

Adaptive Filtering

Signal processing plays an integral role in today's technical community as well as society at large. Applications range from consumer electronics, manufacturing robots, high (or low) end audio systems, to biomedical engineering. Digital signal processing (DSP) as a field has enjoyed a tremendous acceleration in technology attributable to the development of high speed hardware such as analog to digital converters.

Many applications in DSP can be approximated as linear time invariant (LTI). These systems carry with them the luxury of deterministic methods of finding an optimal or nearly optimal solution (usually in the form of a filter design). Adaptive signal processing is a field within the area of digital signal processing which relies on the basics of DSP while adding the statistics of stochastic signal processing. The statistically optimal solution to *stationary* systems can be found in Wiener filter theory. Non-stationary inputs require self-designing solutions which can change dynamically in time without requiring prior knowledge of the input. Adaptive filter theory provides just this solution.

Adaptive filter design and control techniques are fairly well developed despite the fact that it is a relatively new field. Bernard Widrow is credited with the development of the highly popular LMS (least mean squares) algorithm, created in the late 1960's and early '70's [1]. Adaptive filtering techniques allow real-time digital filter variations ideal for applications in time-variant (and in some respects non-linear) systems. Neural networks are also enjoying a new resurgence due to the continued increase in computational horsepower in modern DSP and computer systems. They are implemented as a method for dealing with highly nonlinear processes but still derive their basis in DSP theory.

It is clearly beyond the scope of this chapter to examine adaptive signal processing and adaptive filtering in a textbook fashion by exploring multiple solutions to multiple problems. The approach taken in the investigation and implementation of adaptive filtering will begin with a brief introduction to the theory behind Wiener filters and the most common method of adaptive filtering: the LMS algorithm. Several assumptions are made concerning background regarding digital signal processing (DSP). The basics of convolution, filtering, and Fourier analysis are assumed. Terminology such as finite-impulse response (FIR) filtering and block diagram algebra are also taken as known. With DSP emerging as one of the most important technologies for the future, a wealth of literature is available which effectively discusses these areas [11, 12, 13, 14].

Words such as *stationary* and *non-stationary* are briefly mentioned in this chapter in order to develop the theory for adaptive filters. They are characteristics which describe the behavior of signals with respect to time. It is not strictly necessary that these characteristics be completely understood in order to present the material on adaptive filters. Chapter 4 addresses this terminology in a more strict sense as it applies to coherent output power and the COP filters.

With a structure and theory background having been developed, the focus will turn to the common practical application of adaptive signal processing being investigated throughout this work: active (acoustic) noise cancellation. It should be emphasized that throughout Chapter 3, the term "noise" will refer to the undesirable acoustic disturbance which will be actively controlled. Chapter 4 redefines "noise" as uncorrelated content in measured signals. The physical problem as well as a small amount of acoustical theory will be described including the appropriate structure and implementation for the adaptive filter. Most adaptive filtering problems require a somewhat tailored solution, the acoustic noise cancellation structure taking no exception.

Having developed a physical problem (acoustic noise cancellation) and a possible solution (the LMS adaptive algorithm), three revised methods of the LMS will be examined: the filtered reference LMS, adaptive arrays, and block adaptive filtering as the LMS algorithm is

realized in the frequency domain. Each of these implementations as well as the conventional LMS are all demonstrated in terms of MATLAB generated simulations of the acoustic noise cancellation problem, carrying a common scenario of noise generated from a motor from either a printer or an automobile.

The filtered reference approach is necessary in *any* practical implementation of adaptive filtering as applied to active noise control. This topic will be explained in some detail in Section 3.3 and forgone thereafter. It is a methodology which is straightforward to implement but adds a level of complexity that may detract from the topics being presented. It will be assumed that any practical implementation of ANC will also include the filtered reference, but as will be explained, is not necessary in simulation.

The adaptive array represents an alternative method to achieve similar results as the COP filters demonstrate in Chapter 5. The brief introduction included in this chapter can be supported by a wealth of literature as mentioned in Chapter 2. The relationships between the adaptive array implementation and the COP filter design approach are carefully examined in Chapter 5.

Frequency domain adaptive filtering is also a fairly mature field in that its feasibility and implementation have been thoroughly studied. Since the COP filters are digital filters designed in the frequency domain, it is intuitive that the adaptive implementation of this procedure should also be realized in the frequency domain. High computational savings and intuitive physical interpretation of results make the frequency domain approach the most attractive of implementations discussed herein. Therefore, special attention will be paid to several possible structures, and the analytical developments throughout Chapter 3 should be considered to focus and culminate to Section 3.5.

3.1 ADAPTIVE SIGNAL PROCESSING

3.1.1 OPTIMAL LINEAR FILTERING

The optimal filtering problem is often presented as an introduction to the development of adaptive filtering [1,10]. Adaptive filtering and the LMS algorithm in particular can be viewed as a real time update of continually solving the optimal linear filtering problem. The method most commonly used is presented here in the form of Wiener filters.

The solution to the Wiener filtering problem takes a statistical approach to determining the optimal filter for a given situation. The signal flow diagram depicted in Figure 3.1 shows the beginning concepts behind developing a statistically optimal linear filter. The signals are organized as follows: u(n) is the sampled input, y(n) is the digitally filtered output, d(n) is the desired sample value at time n, and e(n) is the error or the difference in the desired value and the actual value of the filter output.

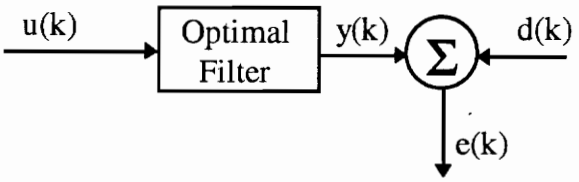


FIGURE 3.1 OPTIMAL LINEAR FILTER

At this point it is necessary to depart from generalities and develop the Wiener solution based on real valued signals and an FIR filter for the optimal filter realization. (Hereafter the tap delay line filter structure will be assumed to ensure stability of the filter). This development is indeed a subset of the Wiener filters but is sufficient to demonstrate the theory [1,10].

The criteria set forth to determine the optimal filter solution is typically based on a minimization of the mean square error (cost function), where the error is defined by Equation 3.1. The rigorous details of the derivation of the Mean Square Error will be summarized so only a brief explanation of the procedure is presented here. Using Figure

3.1, the basic definition of the error signal can be stated as a difference of the controller output and the desired signal.

$$e_k = y_k - d_k \tag{3.1}$$

In general, y_k is a vector convolution of inputs and weights. Depending on the desired order (size) of the optimal filter, $\mathbf{X_k}$ and $\mathbf{W_k}$ can be arbitrarily large. (Bold capital letters will henceforth refer to vectors). For the sake of brevity, these definitions will hold for a first order filter with a bias term w_{0k}.

$$\mathbf{X_k} = \begin{bmatrix} X_k \\ X_{k-1} \end{bmatrix} \tag{3.2}$$

$$\mathbf{W_k} = \begin{bmatrix} W_{0k} \\ W_{1k} \end{bmatrix} \tag{3.3}$$

Squaring Equation 3.1 produces what is defined as the Mean Square Error (MSE), also represented by J. Assuming the signal is stationary with time, the expected value (statistical mean) of the equation for the MSE is performed which creates two new matrices that will be defined as the input correlation matrix (**R**) and the cross-correlation matrix (**P**).

$$\mathbf{R} = E\left[\mathbf{X_k X_k^T}\right] \tag{3.4}$$

$$\mathbf{P} = E\left[\mathbf{d_k X_k}\right] \tag{3.5}$$

The expected value can be thought of as an average signal strength over time. (This concept will be more thoroughly discussed in Chapter 4). Assuming the signals are

stationary ergodic simply implies that the average for a single time record does not change significantly over time. Therefore the MSE equation can then be written:

$$MSE \equiv J(n) = E\left[d_k^2\right] + \mathbf{W^T R W} - \mathbf{2P^T W} \qquad (3.6)$$

With the MSE defined the following statements can be made. For a non-zero input it is obvious that the input correlation matrix **R**, is positive definite and symmetric. The squaring operation ensures this property. For a system with two weights (making **W** two by one) a three dimensional graph of the MSE can be constructed as shown in Figure 3.2.

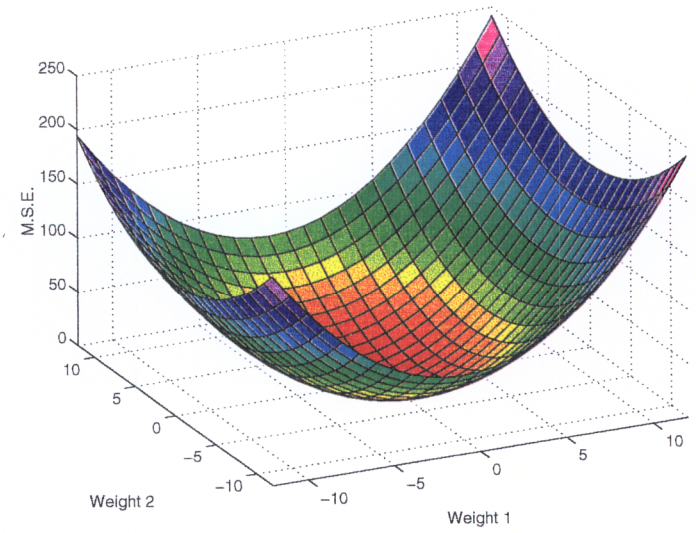


FIGURE 3.2 EXAMPLE ERROR SURFACE FOR 2 WEIGHT FILTER

This surface will be a paraboloid which is always in the positive "z" direction (since it is impossible to have a negative MSE). The performance surface is shown here for a first order filter, as it is impossible to depict a four dimensional performance surface for three (or more) weights.

Visually, the solution to the optimal filtering problem should be obvious. The lowest point on the performance surface depicted in Figure 3.2 indicates the minimum MSE and thus the optimal weights w_o and w_i for the first order case. Mathematically, setting the derivative

(with respect to the weight vector) of Equation 3.6 equal to zero will provide the optimal solution of the form:

$$W_{opt} = \mathbf{R}^{-1}\mathbf{P} \qquad (3.7)$$

There are several difficulties inherent in this development if applied to systems with *non-stationary* inputs. They are summarized as follows:

- *The optimal weight vector will change with time*
- *Prior knowledge of the input correlation matrix* R *is required*
- *Repeated computation of the inverse of* R *is time consuming*

The Method of Steepest Descent is a mathematical precursor to the development of the LMS algorithm and is based on prior knowledge of the input **R**. The LMS algorithm is an implementation of optimal filtering which allows the mean square error to be minimized, avoids computing the inverse of the autocorrelation matrix, and does not require prior knowledge of the input.

3.1.2 THE METHOD OF STEEPEST DESCENT

A brief and somewhat informal summary of the steepest descent algorithm will aid in the derivation of the LMS algorithm itself. The steepest descent algorithm is a *recursive* gradient search method for determining the Wiener solution to a given filtering problem.

The gradient of the performance surface shown in Figure 3.2 is defined as the partial derivative of the mean square error *J(n)* with respect to the filter weights. Therefore taking the derivative of Equation 3.6 in this manner produces the gradient equation shown in Equation 3.8.

$$\nabla J(n) = 2\mathbf{RW} - 2\mathbf{P} \qquad (3.8)$$

By the method of steepest descent, the updated values of the adapted weights are computed by the recursive relation defined in Equation 3.9.

$$\mathbf{W}(n+1) = \mathbf{W}(n) + (\tfrac{1}{2})\,\mu\,[-\nabla J(n)] \qquad (3.9)$$

This indicates that starting from some initial guess (at time n=0), the tap weights are updated according to the negative of the gradient multiplied by some scalar μ. (The one-half is used to cancel the 2's in Equation 3.8; it is somewhat arbitrary since this factor can be absorbed by the convergence parameter μ). It can then be seen that the algorithm merely travels down the error surface until it reaches the optimal value at which point the gradient is zero and the weights become the Wiener solution.

Several important features (limitations) set this apart from the LMS algorithm. The ensemble averaged quantities for the correlation matrix (**R**) and the cross-correlation matrix (**P**) are required to perform the computations in Equations 3.8 and 3.9. This limitation does not lend itself to a real-time adaptive implementation where prior knowledge of these quantities is often unavailable or impractical to calculate. Intuitively, it is desired to have an immediate estimate of the gradient to use as the approximation for the local slope. Therefore the primary motivation behind the LMS is to derive the stochastic gradient to act as an estimate of the deterministic gradient used in the steepest descent method.

3.1.3 THE LMS ALGORITHM

Stochastic gradient searching relies on the assumption that an *estimate* of the current gradient is an acceptable approximation to the deterministic value of the gradient as calculated by Equation 3.8. Therefore if the gradient vector could be known exactly at each time step, the weight update would converge to the Wiener solution each time as in the steepest descent method. Unfortunately, practical applications do not present this luxury.

Following the same development as the steepest descent, the LMS algorithm uses the approach that instantaneous measurements of the gradient can be used to approximate the gradient for each time step. (This is the simplest and possibly noisiest method of determining the gradient, much like taking a derivative using a simple difference over time). Using italics to represent the estimated values, the LMS algorithm (developed using the steepest descent method) is as follows.

$$R = \vec{u}(n)\vec{u}^{H}(n) \tag{3.10}$$

$$P = \vec{u}(n)\vec{d}^{*}(n) \tag{3.11}$$

Before proceeding, a block diagram is used to illustrate the physical meaning of the signals described in Equations 3.10 and 3.11. The general concept is as discussed in Section 3.1.2. The LMS algorithm uses the system input u(n) and the error between the adaptive filter output y(n) and some desired signal d(n), to design a set of weights **W** to continually update the adaptive filter.

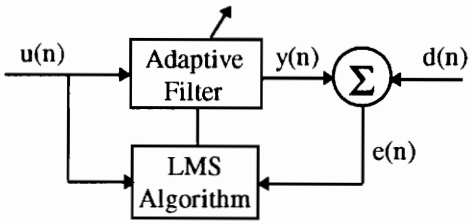


FIGURE 3.3 SIMPLE ADAPTIVE FILTER DIAGRAM

Substituting Equations 3.10 and 3.11 into Equations 3.8 and 3.9, the LMS weight update recursion is formed.

$$W(n+1) = W(n) + \mu\, \vec{u}(n)[\vec{d}^{*}(n) - \vec{u}^{H}(n)W(n)] \tag{3.12}$$

The output of the adaptive filter defined as y(n) then becomes the convolution of the weights with the input data. As in the steepest descent method, the value for μ governs

convergence rate and stability. Although the author is familiar with independence theory and the principle of orthogonality, the convergence criteria derivation is beyond the scope of this discussion. In general, a smaller value of μ increases the time constant of the weight update and a larger value increases the excess error thus amplifying gradient noise (excess mean squared error). An upper bound is created on μ by setting it less than the inverse of the largest eigenvalue of **R**. Since this is typically unknown, the tap input power is used in its place as a rough estimate [10].

Therefore in general, the conventional LMS algorithm can be summarized as follows:

- *Given:* $\mathbf{u}(n) = M$ *(number of weights)-by-1 input vector at time* n $d(n) =$ *desired filter response at time* n
- *Find:* $\mathbf{W}(n+1) = M$*-by-1 estimate of tap weights at time* $(n+1)$
- *Compute:* $e(n) = d(n) - \mathbf{W}^H(n)\mathbf{u}(n)$ *and* $\mathbf{W}(n+1) = \mathbf{W}(n) + \mu\,\mathbf{u}(n)e^*(n)$

The tap weight vector is often initialized to the null vector. The LMS algorithm is clearly a very simple weight update which requires only the past M inputs to make an estimate of the local gradient. The instantaneous estimates of **R** and **P** typically have large variances which would imply that the success of the LMS is limited. Due to its recursive nature these effects are usually averaged during the adaptation process.

It is typically impossible to achieve the Wiener solution because the algorithm operates on estimates rather than actual deterministic inputs. The excess mean square error and misadjustment, quantitative values to describe the amount of deviation from the Wiener solution, will not be addressed in this discussion. Suffice it to say that their effects *can* be estimated from independence theory and the weight error vector correlation matrix.

3.2 ACTIVE NOISE CONTROL

The general development of the LMS algorithm discussed in Section 3.1.3, can be applied to a variety of control or filtering situations. Each application has a slightly different

modification to the basic structure shown in Figure 3.3, causing adaptive filtering to become a situation dependent problem. Applications such as adaptive beamforming, adaptive line enhancement and system identification each have alternate structures and goals. Each situation however, holds a common thread in that a cost function is developed and minimized via the LMS algorithm.

This section will now focus on the application of the LMS algorithm to acoustic noise control. After describing some elementary physics of noise control and presenting a physical structure, a MATLAB simulation of the LMS algorithm applied to a tonal noise is examined.

3.2.1 ACOUSTICAL PHYSICS OF ANC

Noises from disturbances such as a printer, a motor running, or car tires spinning on the road all originate from a cyclic change in pressure of the surrounding air, thus creating sound. These sound waves travel out from the source in three dimensions much like a pebble thrown into a pond generates circular waves in two dimensions. To "control" these waves, an anti-noise source is generated to counter the effects.

To simplify matters, the wave can be thought of in a single dimension with time, such as a sine wave. If a sine wave of equal magnitude and 180° out of phase were superimposed on the original noise disturbance, the combined sound pressure level (SPL) effect would equal zero, thus no noise would be heard.

Extending this concept to three dimensional space introduces several complications which have limited the popularity (and success) of many active noise control ventures. The "anti-wave" is typically generated from a control speaker being driven by the LMS algorithm. Figure 3.4 illustrates some of the physical apparatus required, assuming a motor is creating the unwanted noise. The basic concept is that the error microphone signal is minimized by the LMS algorithm operating on the reference signal. The desired signal is (of course) zero thereby causing the error signal ($e = d - y$) to be the negative of the error microphone signal

which is y. Figure 3.4 introduces two new (physical) issues which are beyond the control of the LMS algorithm. These are the concepts of collocation and the uncontrollable reference.

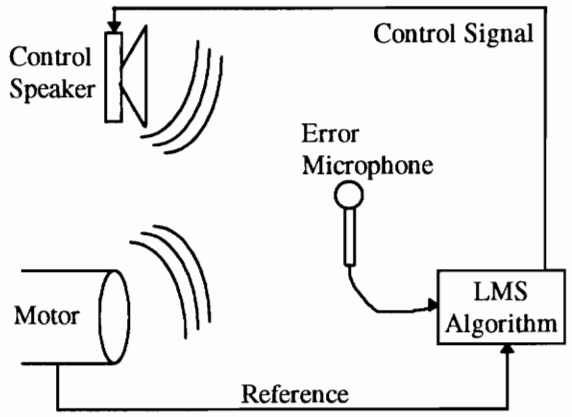


FIGURE 3.4 PHYSICAL IMPLEMENTATION OF ANC

Collocation refers to the location of the control source (speaker) with respect to the disturbance (motor) and is an issue in other types of physical plant control such as vibrations. Returning to the pebble making waves in a pond, the collocation concept can be easily illustrated. If somehow one pebble could be thrown directly on top of another with a time delay equal to 180° of phase, the second wave would exactly cancel the first at all points in the pond, thus creating "global control" of the wave. If however, a pebble were thrown at the same time adjacent to the original one, the destructive interference would only occur at a few points in the pond as seen in Figure 3.5.

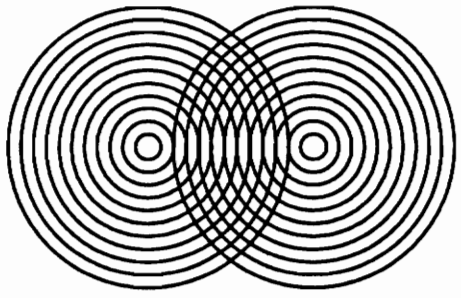


FIGURE 3.5 EFFECTS OF NON-COLLOCATION IN ANC

This is clearly an effect that varies with frequency and distance between the disturbance and the control. However if the two sources were collocated, the cancellation would exist at all frequencies and locations.

The concept of a coherent, uncontrollable reference is necessary to ensure the feedforward characteristics of the entire system. Figure 3.4 shows a simple line leaving the motor to represent the reference signal. Choice of this signal and the sensor used to develop it is *very* important in achieving the desired control and is the entire basis and motivation of the COP filters. Since this reference signal is driving, and being altered only in phase and magnitude by the adaptive filter, it needs to be highly correlated with the actual disturbance noise. Although the strict definitions of coherence (a measure of correlation) are presented in Chapter 4, a simple development here will assist in illustrating the importance of a coherent reference and provide a motivation for the in depth study in Chapter 4.

The coherence is a frequency dependent measure of system linearity (or correlation) which can assume values between zero and unity, unity indicating a perfectly linear system. It is shown in Chapter 5 and in [2], that the ratio of the power spectral density of the error signal after control to that before control is given by Equation 3.13.

$$\frac{S_{ee}}{S_{dd}} = 1 - \gamma_{rd}^2 \qquad \textbf{(3.13)}$$

Therefore if the coherence between the reference and the disturbance (d) is unity, the converged power spectrum of the error signal is zero. As mentioned in Chapter 1, noise typically originates from multiple processes thereby requiring multiple coherent references. It is the basis of this work to build a single coherent reference (from multiple references) for use in adaptive filtering.

Returning to the scenario in Figure 3.4, the best choice to maximize coherence would clearly be a microphone reference since the generated noise *is* the disturbance. (In this case the adaptive filter would need only to shift each frequency in phase to create the anti-noise.

This ideal scenario would also depend on the actual location of the reference microphone relative to the error microphone). The primary difficulty with this approach is that the control speaker will directly affect the reference microphone thereby creating a *feedback* path which can (and usually does) go unstable. A good example of an *uncontrollable* reference is an encoder measuring the RPM of the motor. Although the signal is (hopefully) highly correlated with the disturbance noise, no matter how loud the control source becomes, the encoder (and thus the "uncontrollable reference") will not be affected.

These effects are physical phenomenon encountered in active noise control and are used to evaluate its success or failure in certain applications. They are mentioned briefly here to provide a background in understanding the complexities of ANC. Since the focus is on signal processing with ANC as an *application*, attention will now return to the LMS algorithm, its simulation, and useful modifications.

3.2.2 MATLAB SIMULATION

Armed with a sufficient understanding of the LMS algorithm as well as how it can be applied to active noise control, a MATLAB simulation was developed to demonstrate the successful operation of the entire system. While there are many limitations and idealities in simulations, an effort was made to create a realistic scenario. Further refinement of the design is discussed in Section 3.3 where another element of realism encountered in the practical application of active noise control is added to the simulation. (Appendix A contains the MATLAB code used to perform this simulation).

Continuing with the idea of a motor generating an unwanted disturbance in the form of a tonal noise, a simulation has been designed. Assume an uncontrollable reference which is highly correlated to the disturbance noise is available in the form of an electronic signal. The disturbance noise then derives from the motor spinning and a single tone is generated. Figure 3.6 shows the reference and the disturbance signals each at 2 Hz sampled at 16 Hz.

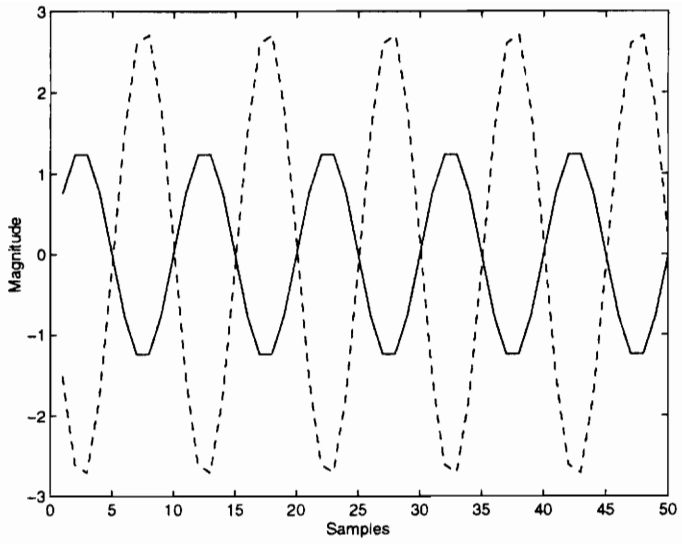


FIGURE 3.6 REFERENCE AND PLANT (DOTTED)

From a signal processing perspective, the reference is passed through a linear time invariant plant (or filter) to produce the plant output. (Simulated by a simple phase and magnitude change). The physical process represented by the filter is the transfer function created by the motor. This is illustrated in the block diagram of Figure 3.7.

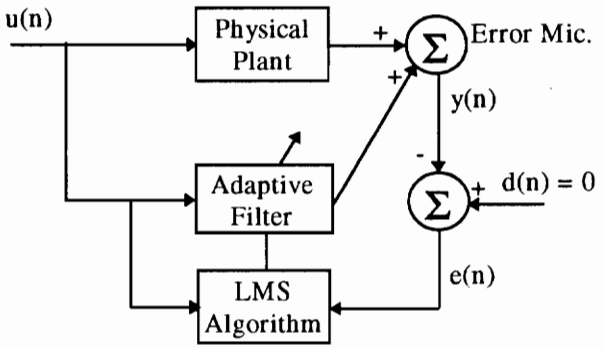


FIGURE 3.7 ANC PHYSICAL IMPLEMENTATION DIAGRAM

An interesting difference is apparent between the general block diagram in Figure 3.3 and the applied system in Figure 3.7. Here the y(n) signal is the SPL *sum* of the disturbance noise and the controller output. Since the desired response at the error microphone is zero, the error signal is merely the negative of the error microphone signal (as mentioned earlier).

Using the generated signals shown in Figure 3.6, an initial tap weight vector equal to the null vector, and a μ chosen such that the system converged, the LMS code was run for 400 iterations sampling at 16 Hz. Figure 3.8 shows the time response of the error microphone signal (also the negative of the error signal as defined by the development of the LMS algorithm).

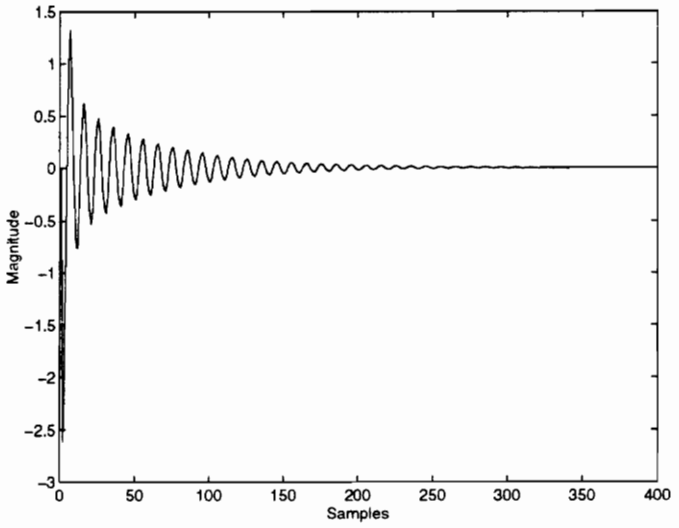


FIGURE 3.8 ERROR MICROPHONE SIGNAL

It is clear that the adaptive filter is able to design itself within 150 iterations which corresponds to about 9 seconds at 16 Hz. Examining Figure 3.9, a new perspective on the operation of this algorithm can be obtained. The control speaker output signal and the disturbance are plotted together to demonstrate the wave cancellation which takes place. The adaptive filter is updated (based on the reference signal and the error signal) in order to filter the reference signal such that its output matches amplitude and is 180° out of phase with the disturbance. Therefore, in essence the filter is designed to model the plant (plus 180° of phase to achieve cancellation) much like a system ID.

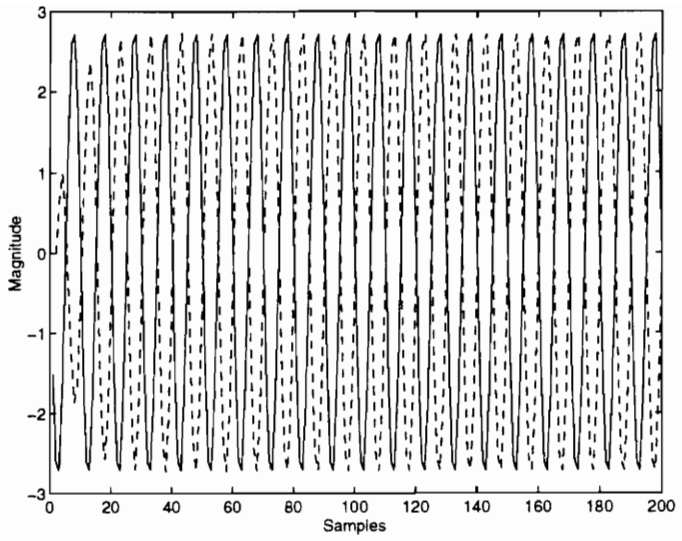


FIGURE 3.9 PLANT AND CONTROL (DOTTED)

Choice of filter order and convergence parameter are somewhat deterministic and somewhat based on trial and error. For this case, a μ of **0.1** and a filter order of **2** achieved nearly the same response as the one shown in Figures 3.8 and 3.9, where a μ of **0.01** and a filter order of **9** were used.

It is intuitive that only a single delay (two weights) is necessary to match magnitude and phase of a sine wave disturbance (with a proper reference). When systems become more complicated and multiple modes of control are desired, larger filter orders and smaller μ's are necessary to achieve effective control and maintain stability. These are control issues specific to certain applications. For this case, a filter order of 1 is sufficient to cancel the sine wave, while the μ that creates a critically damped system is one half the inverse of the largest eigenvalue of the input correlation matrix, approximately 0.1.

3.3 THE FILTERED REFERENCE

3.3.1 MOTIVATION

Re-examining Figures 3.4 and 3.7 (the physical implementation and block diagram of active noise control) Figure 3.10 adds another element of complexity. The filtered reference (also referred to as the filtered-X) refers to the filtering of the reference input, u(n), with an estimate of the control to error path. Before going further, a review of the physics which require modeling will intuit the reasons behind doing this.

Figure 3.4 shows the physical system involved in achieving active noise control. As mentioned earlier, the adaptive filter essentially models the motor to error microphone path (which includes the dynamics of air, the motor itself, and the microphone) in order to filter the reference signal and cancel the disturbance. Continuing along this same path, the control output must pass through a smoothing filter (as it is a zero order held output), an amplifier, a speaker, and the air before it reaches the error microphone. This is all termed the control to error path and is illustrated in block diagram form in Figure 3.10.

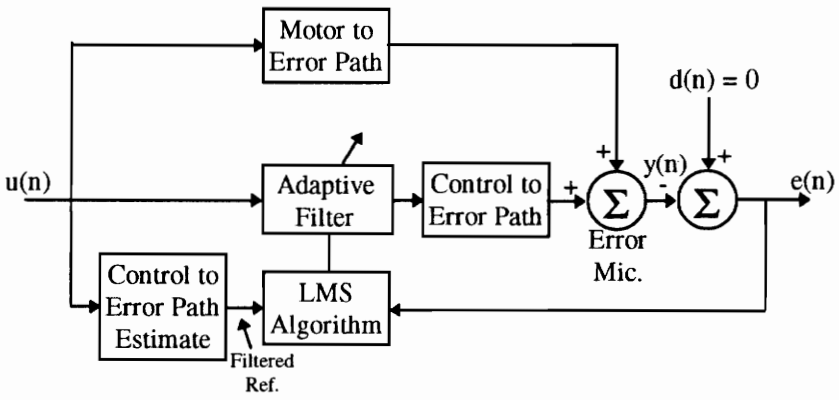


FIGURE 3.10 FILTERED REFERENCE BLOCK DIAGRAM

The configuration described in Section 3.2 fails to take these physics into account and the system works well because these dynamics were not included in the simulated plant. (This is one of the many trappings of a simulation which may not become apparent until actual implementation is attempted). The practical application of this system however, requires

the reference to be filtered by an estimate of the control to error path. The filtered reference then takes into account the dynamics which occur *after* the control signal leaves the adaptive filter; dynamics which are not "visible" by the LMS update described in Section 3.2.

An ideal alternative to Figure 3.10 would be to have the control to error path occur *prior* to the adaptive filter, in which case the control signal would act directly on the disturbance and the reference would already have been filtered. This is demonstrated in Figure 3.11. Unfortunately, it is physically impossible to achieve this effect so it is *estimated* by filtering the reference signal with an approximation of the control to error path. Since it is also impossible to know the control to error path exactly, a model is created (through system ID methods which may also involve the LMS algorithm) and used in its place.

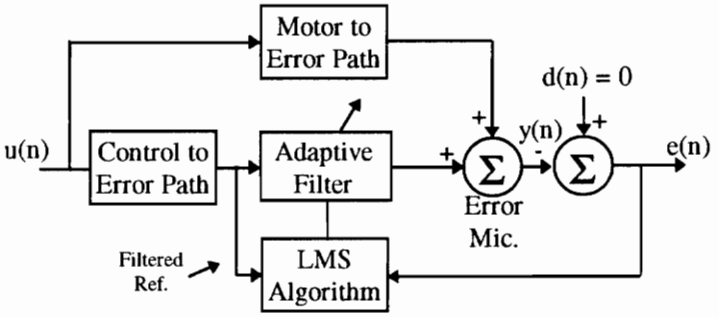


FIGURE 3.11 REDEFINED FILTERED REFERENCE BLOCK DIAGRAM

As a matter of system stability, the FIR adaptive filter should not become unstable if a suitable convergence parameter is chosen based on the reference signal power. However, if the control to error path is incorrectly modeled by more than 90° of phase, the system will become unstable due to inaccurate system identification.

3.3.2 MATLAB SIMULATION

For the sake of simplicity, the system identification portion of the filtered-X simulation was assumed to be exact (i.e. the exact filter used to simulate the control to error path was also used to filter the reference signal for the LMS algorithm). The simulation itself, performed in MATLAB, holds very little difference between the simulation examined in Section 3.2.

(Appendix A shows only the differences). The LMS algorithm operates exactly the same, it merely uses a filtered version of the reference while the output of the adaptive filter experiences the control to error path.

The control to error path was randomly chosen to be the all-pole (stable) filter shown in Equation 3.14.

$$\frac{1}{z^6 - 0.8\,z^5 - 0.94\,z^4 + 1.01\,z^3 + 0.0969\,z^2 - 0.2482\,z + 0.0473} \quad \textbf{(3.14)}$$

With the control to error path included in the simulation, the new dynamics create a slightly different behavior of the LMS algorithm. Since the value for μ depends on the magnitude of the reference (now filtered) signal, it should be clear that a convergence parameter which was once stable, may not be now.

To maintain continuity and continue with the motor noise scenario, the same reference signal and plant output were generated, also at a sampling frequency of 16 Hz. Figure 3.12 shows each of these signals again. With the same filter order (of 9) and a new convergence parameter of 0.0005, Figure 3.13 shows the error microphone signal which is generated during operation of the filtered-X simulation. The convergence parameter of 0.01 used in the previous LMS simulation caused this system to diverge. This is due to the altered reference signal (and thus maximum eigenvalue). The convergence of this particular case is significantly better when compared with the original LMS simulation due to the optimization of μ over a number of trials. (This was intentionally not done in Section 3.2 in order to demonstrate the affects the convergence parameter has on system response).

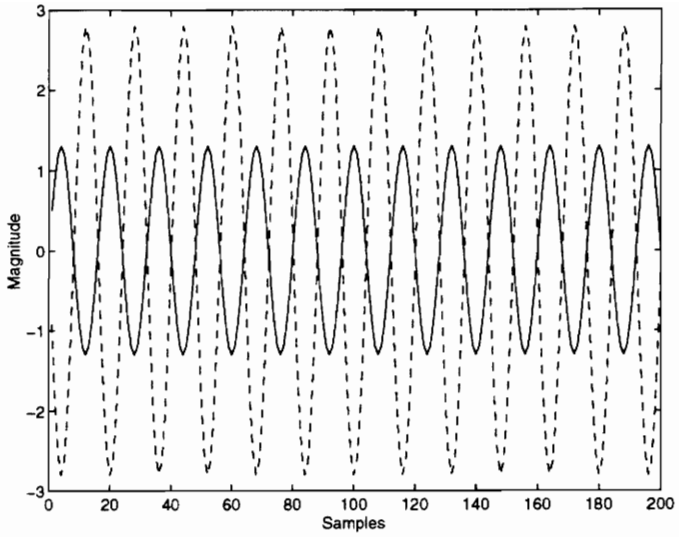


FIGURE 3.12 REFERENCE AND PLANT (DOTTED)

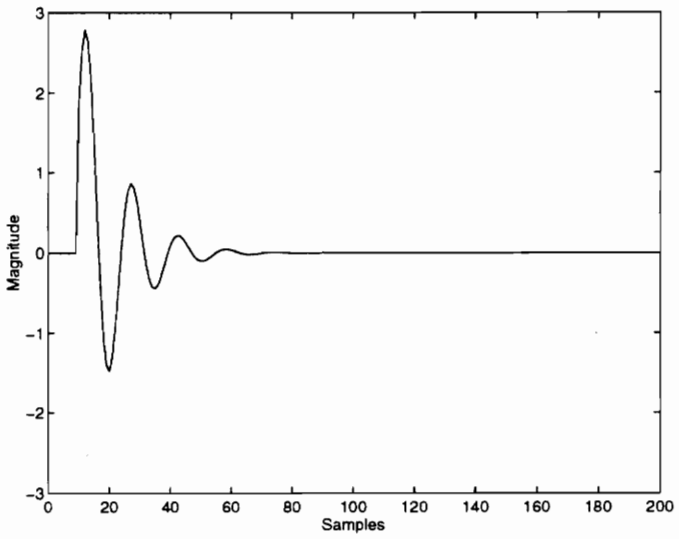


FIGURE 3.13 FILTERED REFERENCE ERROR MICROPHONE

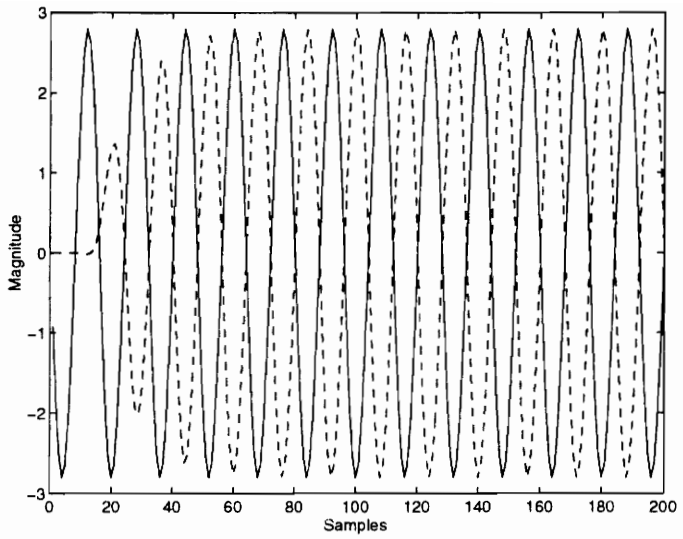


FIGURE 3.14 PLANT AND CONTROL (DOTTED)

Figure 3.14 shows the familiar control signal achieving the magnitude and 180° of phase with respect to the plant (disturbance) output for the filtered-X simulation. While this still may be somewhat abstract, a second simulation showing the error microphone signal should put the filtered-X in perspective.

Suppose the filtering of the reference signal were ignored and the same LMS algorithm were applied to the system described by Figure 3.10. Nowhere in the update is the control to error path taken into account. Using the same values for filter order, convergence parameter, and control to error path transfer function, the LMS algorithm was examined *without* using the filtered reference. Figure 3.15 shows the error microphone signal for this case.

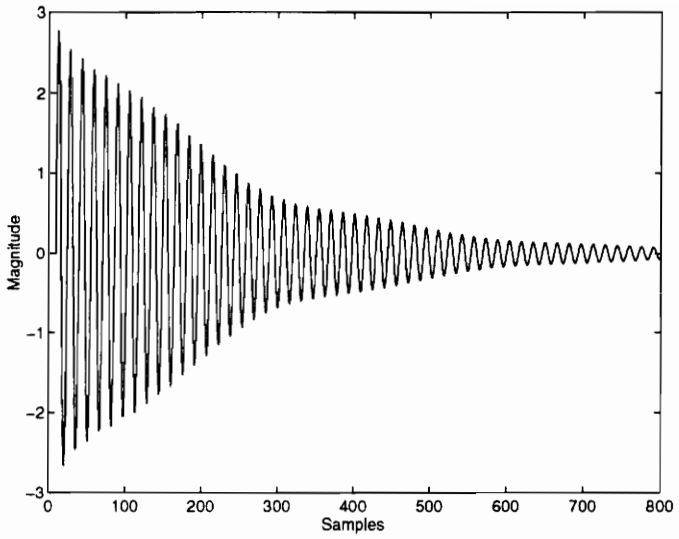


FIGURE 3.15 NON-FILTERED REFERENCE ERROR - STABLE

The convergence, when compared with the filtered-X version of Figure 3.13, takes more than eight times as long and has an underdamped behavior. It is clear that this particular case clearly finds an acceptable solution, given time, but many practical applications are not as robust. In fact, examining this situation in a bit more detail, it becomes apparent why the non-filtered reference still allows the filter to remain stable, when normally it shouldn't.

The nature of the phase delay in the control to error path is quite significant in the evaluation of the LMS algorithm. If the phase delay inherent in this path is *less* than 90°, the adaptive filter will be able to account for the delay and successfully filter the reference to cancel the disturbance. Two practical methods to envision this are 1) the error performance surface and 2) the weight update equation as applied to a sinusoidal.

The error surface in Figure 2 can be generalized for a multi-dimensional space by examining it's behavior in 3 dimensions. The weight update follows a path toward the negative gradient based on the reference and the error. If the reference is more than 90° out of phase with what the adaptive filter "thinks" it should be, the gradient vector directing the weight update will pass the horizontal and proceed in the direction opposite of the bottom of the surface, thus diverging. A second (and more practical) way to understand the 90°

leeway, is to envision a sinusoidal reference more than 90° out of phase with the actual control signal. This would cause the sign in the weight update equation (Equation 3.9) to change and the LMS would diverge. This can be demonstrated with the example used in the filtered-X simulation thus far, by first examining the frequency response of the control to error path.

Figure 3.16 shows the magnitude and phase of the filter described by Equation 3.13. The poor convergence (but convergence nonetheless) illustrated in Figure 3.15 was for a disturbance frequency of 1 Hz. It is clear from Figure 3.15 that the control to error path did *not* impart more than 90° of phase on the adaptive filter output and will therefore achieve control.

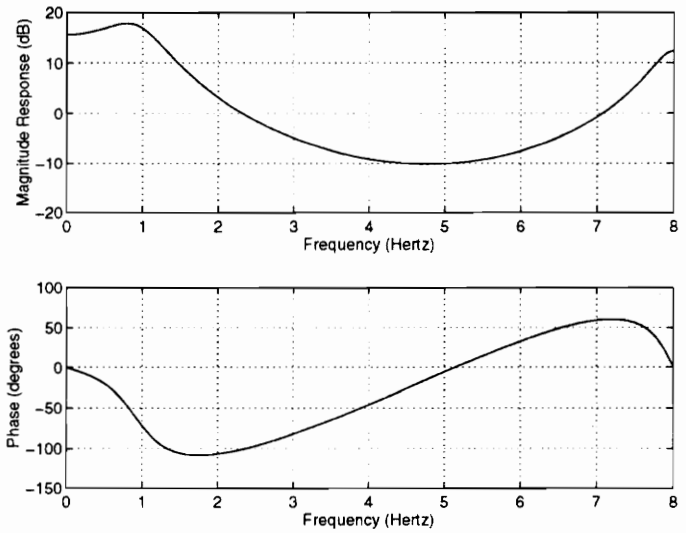


FIGURE 3.16 CONTROL TO ERROR PATH FREQUENCY RESPONSE FUNCTION

If however, the same exact parameters are used while increasing the disturbance frequency to 1.5 Hz (a frequency where the phase lag is clearly over 90°) the (non-filtered-X) LMS will always diverge. Figure 3.17 shows the error signal for this case.

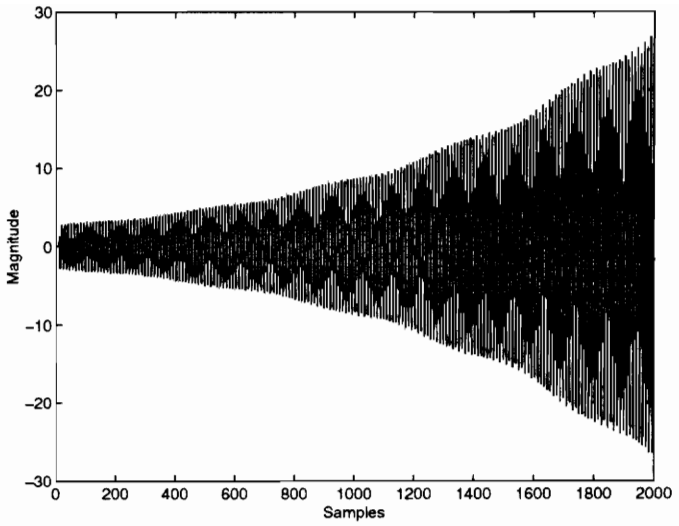


FIGURE 3.17 NON-FILTERED REFERENCE ERROR SIGNAL - UNSTABLE

Since most physical (acoustic) plants have much more phase lag than 90°, it is clear why the filtered-X is necessary to maintain stability in the LMS algorithm. Whenever the control signal cannot act directly on the error signal (or measured output), an estimate of that path should be included in the reference signal. This is also a common solution to the same problem in vibration control. (Even this case almost always has actuator dynamics).

Having discussed the filtered-X LMS implementation and its necessity in practical applications, it will be assumed that any control algorithm subsequently introduced will also include the filtered reference. This includes adaptive arrays, the frequency domain LMS, and the COP filtering methodology. (The filtered reference will not be explicitly described in the context of each of these methods in order to limit complexity and maintain proper focus in each subject area). When the COP filters are described in detail, it will become evident that they are also pre-filtering each of the reference signals much like the filtered-X algorithm. These two operations are separate and distinct from each other, and will be emphasized as such during the practical implementations explored in Chapter 6.

3.4 ADAPTIVE ARRAYS

Adaptive arrays are presented here as a method of handling the multi-reference problem which the COP filters are also intended to solve. The strict comparison of the two methods will be examined in Chapter 5, after explaining the COP filter design in detail. As a background for that comparison, the general structure and implementation are discussed here.

3.4.1 CONCEPTS AND IMPLEMENTATION

As was briefly mentioned in the introduction, the noise control problem as well as many other adaptive control implementations, require multiple reference signals to accurately represent the noise (or other) signal to be controlled. Reconsidering the automobile noise control problem, the undesirable noise can originate from three possible sources: the engine, the fan, and the tire/road noise. Each measured signal will mostly likely be obtained from separate sensors. A tachometer may measure the fan, an accelerometer might measure the engine (block), and a microphone could measure the tire/road noise.

Simple summation of these three signals to form a suitable reference is not an acceptable solution to form a coherent reference. Incoherent frequency components from one signal could easily overpower the coherent components in another signal, thereby reducing the overall coherence. (Chapters 4 and 5 address these issues in more rigorous detail). In order to combine these signals to form a signal that is coherent with the output, the COP filters are required. Upon the generation of the single coherent reference, a simple filtered-X LMS algorithm can be used to achieve control. As an alternative to implementing the COP filters, the adaptive array presents a method to separately use each of these references in an array of LMS controllers.

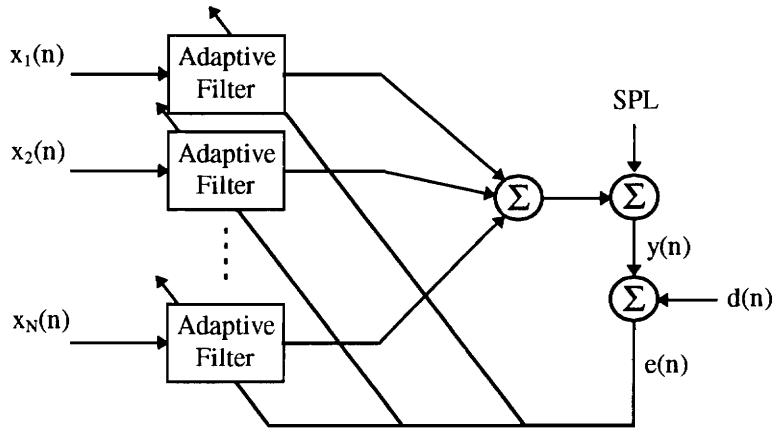


FIGURE 3.18 ADAPTIVE ARRAY

Figure 3.18 illustrates the structure of the adaptive array. Commonly used in adaptive antennae arrays, the same structure is also well suited for noise control using multiple reference signals. Each adaptive filter block represents an independent LMS algorithm as described in Section 3.2. They each accept the separate reference signals while giving as outputs independent control signals. The operation of determining the correctly correlated reference and correct control signal rests with each independent controller. The control signals from each of the outputs are then summed to produce a global control signal used to cancel the existing SPL (for noise control). Based on the desired response, a global error signal is created which is passed back to each of the adaptive filters for the weight update.

The adaptive filters are able to seek out the coherent references regardless of cross path interaction between the inputs and subsequently minimize the overall error signal. Recall that the metric of control for the LMS algorithm was defined in terms of the *ordinary coherence* of the input (reference) to the disturbance noise as defined in Equation 3.13. The metric of control defined for the adaptive array is also a coherence measure, but is defined as the *multiple coherence*. While this is formally defined in Chapter 4, it is essentially a measure of the coherence of all of the combined inputs to the overall output noise [2]. Equation 3.15 defines this metric in terms of the optimal mean square error and the output power spectra.

$$\frac{J_0(\omega)}{S_{dd}} = 1 - \eta_{y:x}^2 \tag{3.15}$$

Therefore, if the multiple coherence is equal to unity, the optimal mean square error (and hence the error autospectrum) will become zero. To gain a full understanding of the ramifications of this metric when applied using the LMS algorithm, a separate analysis is required. The COP filters support this same result while using a different implementation and taking advantage of the deterministic nature of stationary ergodic signals.

When compared with using a single reference, the adaptive array increases the number of operations required in a single sample period by the same number of filters required. If a SISO adaptive filter were used in conjunction with a maximized reference, the number of real time computations will decrease dramatically. This is examined more carefully in Chapter 5.

3.4.2 MATLAB SIMULATION

As a practical illustration of the implementation of the adaptive array, consider a process where one candidate reference signal has a tonal at 2 Hz and the second reference signal contains a tonal at 6 Hz. Each of these signals are then used as the two inputs for an adaptive array structure shown in Figure 3.18. The disturbance (noise) to be canceled by the algorithm is the combination of the two tonals existing in the two reference signals. Therefore the coherence will be high for the first tonal in the first input whereas the coherence will be high for the second tonal in the second input. Figures 3.19 and 3.20 show the Fourier transform of each of the two input signals respectively, where the SNR has been fixed at approximately 40 dB.

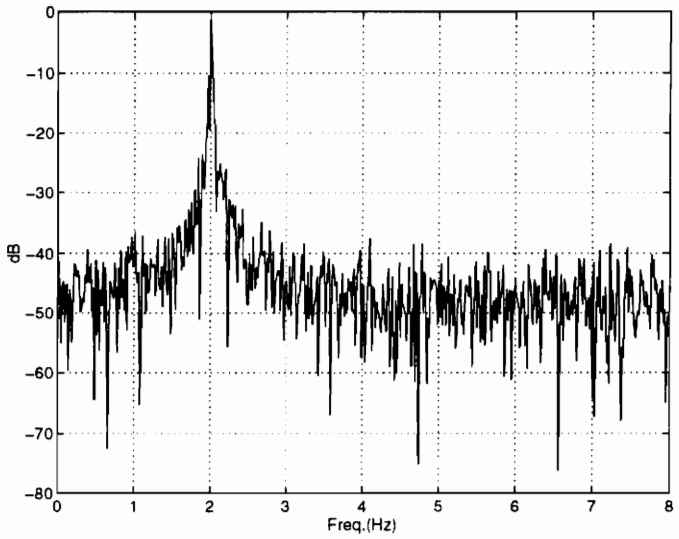


FIGURE 3.19 REFERENCE INPUT 1 - ADAPTIVE ARRAY EXAMPLE

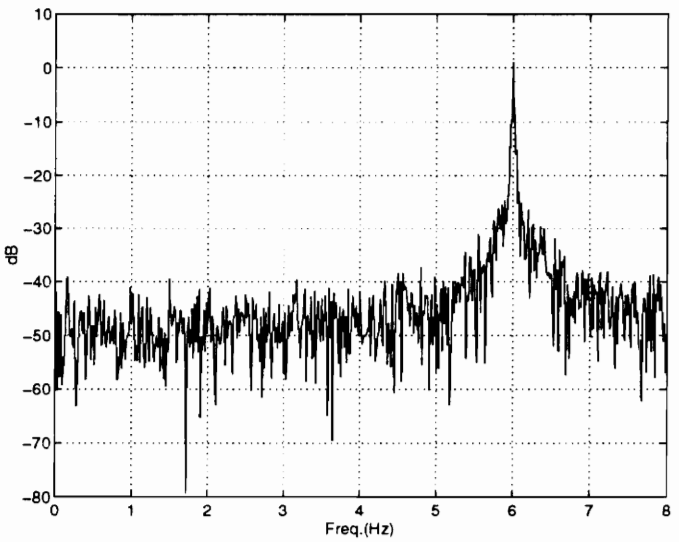


FIGURE 3.20 REFERENCE INPUT 2 - ADAPTIVE ARRAY EXAMPLE

After convergence is achieved, the performance can be measured in decibel reduction of the tonals in the frequency domain as easily as the prior metrics using the time domain error signal. Figure 3.21 shows the power spectrum of the disturbance signal as measured before and after convergence of the adaptive array.

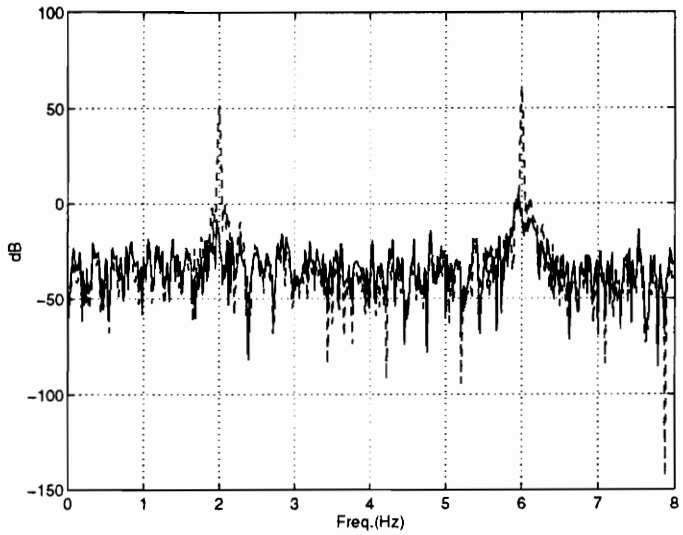


FIGURE 3.21 PERFORMANCE - ADAPTIVE ARRAY EXAMPLE

As can be seen in Figure 3.21, the adaptive array is able to suppress the tonals into the noise quite successfully. Although the adaptive array is a commonly implemented solution to the problem of multiple references, its shortcomings and disadvantages need to be carefully examined as well.

Chapter 5 examines adaptive arrays in a bit more detail as they relate to the solutions achieved by the COP filter implementation. It is necessary to compare the two methods as they are different means to achieve the same ends. It is not the goal to redefine adaptive arrays but rather to use the technology to compare with the COP filter innovations developed herein. It is suspected that the minimal introduction and illustration of adaptive arrays presented here is sufficient to develop the comparison examined in Chapter 5.

Attention is now turned to a final adaptive filtering technology which provides an efficient alternative to the conventional LMS and coordinates well with the COP filtering methodologies.

3.5 BLOCK ADAPTIVE FILTERING

Although the simulation used to illustrate block adaptive filtering is presented from an active noise control perspective, the methodology can be applied to any implementation using the conventional LMS algorithm described in Section 3.2. The filtered reference version of this method will not be discussed, however for reasons previously mentioned it is a necessity in even the simplest of active noise control problems.

Block adaptive filtering differs from the conventional LMS algorithm in that it operates on a block of input data rather than on a sample by sample basis. This will cause the weights and thus the control signal to remain constant over the given block size. Before casting the LMS algorithm in the frequency domain, filtering in the frequency domain as well as the overlap save method are discussed.

3.5.1 FREQUENCY DOMAIN FILTERING

It is commonly stated in the signal processing community that "convolution in the time domain is multiplication in the frequency domain". While to some extent this is true, key points are often omitted when attempting to implement this idea. For finite length data sequences, the result of multiplying two DFTs of two data sequences together actually results in the *circular convolution* of the two data sets.

To illustrate this point, consider the multiplication of the DFTs of the data sequences $x_1(t)$ and $x_2(t)$ to result in the DFT of a third data sequence, $x_3(t)$.

$$X_3(k) = X_1(k)X_2(k) \tag{3.16}$$

The capital letters in Equation 3.16 indicate the frequency domain version of the respective time signals while the index k indicates the finite frequency bin for the DFT. Taking the inverse DFT of Equation 3.16 and substituting the DFT expressions for each of the frequency domain versions of the signals, produces Equation 3.17

$$x_3(m) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)\sum_{l=0}^{N-1} x_2(l)\left[\sum_{k=0}^{N-1} e^{j2\pi k(m-n-l)/N}\right] \qquad \textbf{(3.17)}$$

where N is the DFT/IDFT size. Equation 3.17 can be rewritten in terms of the two original time sequences as

$$x_3(m) = \sum_{n=0}^{N-1} x_1(n)x_2((m-n))_N \qquad m = 0,1,\ldots,N-1 \qquad \textbf{(3.18)}$$

where the variable N indicates the modulo operation [12]. Although this relation is of the form of the linear convolution sum, the modulo operation forces the frequency data to be copied back on itself resulting in erroneous, aliased data. For reference purposes the linear convolution operation performed by an FIR filter is represented in Equation 3.19

$$y(n) = \sum_{n=-\infty}^{\infty} x(n)h(m-k) \qquad \textbf{(3.19)}$$

where the output (y(n)) of the FIR filter is represented by the convolution sum of the input x(n) and the impulse response of the filter h(n), (also equivalent to the weights in the adaptive FIR filter).

The effects of the modulo operation can be taken into account in order to achieve a linear convolution result from the multiplication of the DFTs. One might wonder "why perform all of these operations in the frequency domain when it is easily implemented in the time domain?" First, the FFT is computationally efficient method of implementing the DFT which when applied as a filtering operation, reduces the number of computations when compared to time domain convolution. Second, the frequency domain provides an intuitive interpretation of the COP filtering technology, especially applicable in active noise control.

To implement linear convolution in the frequency domain, consider two finite duration sequences, x(n) of length L (representing the input to the filter) and h(n) of length M

(representing the FIR filter's impulse response). It is desired to compute the output sequence y(n) as described by Equation 3.19 via linear convolution in the frequency domain. Equation 3.19 gives the output result for all time; however, the nonzero result only exists for finite time of length L+M-1. To accurately represent the output sequence in the frequency domain via multiplication, the FFT length must therefore be at least equal to L+M-1. (Radix-2 FFT operations are implemented efficiently for FFT sizes that are power's of two; therefore the size is often rounded accordingly).

Taking the FFT of sequences shorter than the specified length requires zero padding the data. For an FFT of length L+M-1, x(n) will have M-1 zeros appended and h(n) will have L-1 zeros appended. The IFFT of the product of X(k) and H(k) has a sufficient number of points to represent the output in the time domain as a linear convolution [12].

This process is defined for finite length sequences for which the output can be determined by large enough FFT sizes. In practice, continuous filtering of an input does not facilitate a single FFT operation on the entire set of data. The filtering of long data sequences requires finite length Fourier transforms calculating the linear convolution for blocks of data over time. This can be accomplished in one of two ways, the *overlap-add method* or the *overlap-save method.* The overlap save method is presented here and used in the demonstration of the frequency domain version of the LMS algorithm.

The general idea behind the overlap save method for filtering long data sequences is very similar to that which is used in finite length data sequences. FFT operations over acceptable lengths are used to determine the linearly convoluted output. Special precautions must be used in order to ensure a continuous flow of data through the filter. The general algorithm for the overlap save method is best illustrated in Figure 3.22 [19].

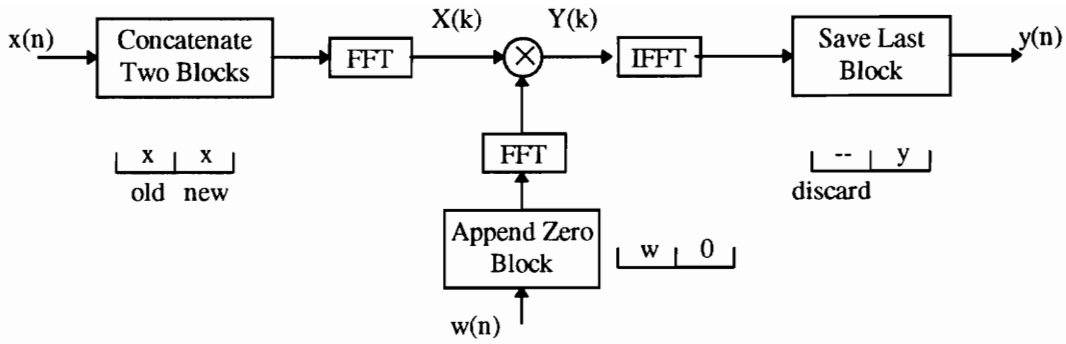


FIGURE 3.22 OVERLAP SAVE ALGORITHM

Depending on the desired FFT size, the length of each data sequence can be arbitrary conforming to the sizes mentioned for finite length sequences. The most common (and efficient) implementation provides 50% overlap of the input data, saving M points of old input data and concatenating it with M points of new data. Each FFT and IFFT is 2M points long and the last M points of the inverse FFT are saved as the non-aliased linear convolution.

The frequency domain LMS algorithm has both a convolution and a correlation operation which can be performed in the frequency domain. (This might already have been suspected since in the time domain the LMS requires filtering (convolution) and vector multiplication for the gradient calculation corresponding to correlation).

3.5.2 FREQUENCY DOMAIN LMS

The frequency domain version of the LMS algorithm described in Section 3.2 offers a significant reduction in the overall number of computations as well as an intuitive description of the meaning of the filter weights. There are three separate methods of performing the LMS algorithm in the frequency domain. They are termed the constrained, unconstrained, and circular methods. Each employs a different level of accuracy obtained by the convolution and correlation operations as they relate to the time domain.

The constrained LMS algorithm uses the overlap save method on both the convolution and correlation operations in the frequency domain. This level of computation maintains the linear convolution and correlation results as if they were performed in the time domain. The algorithm/block diagram for this structure is shown in Figure 3.23.

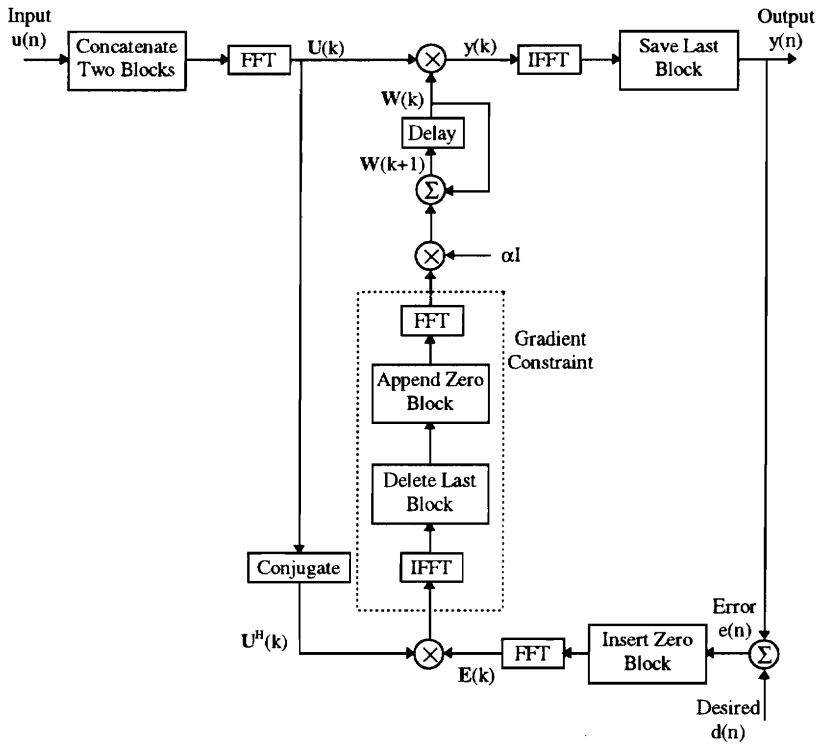


FIGURE 3.23 FULLY CONSTRAINED FREQUENCY DOMAIN LMS ALGORITHM

The term "constrained" refers to the limitations placed on the correlation calculation of the gradient as shown in Figure 3.23. The top portion of this figure is identical to the overlap save algorithm describe in Figure 3.22, only presented as a portion of the LMS algorithm. The benefits in performing linear convolution and correlation in the frequency domain are obvious in that the errors encountered in circular convolution are avoided. The costs however, can be evaluated in terms of the number of computations incurred by the additional FFT operations. After introducing two additional methods where computations can be further reduced, the computational complexity of all methods will be compared with that of the conventional LMS.

The unconstrained version of the frequency domain LMS algorithm modifies Figure 3.23 by removing the operations enclosed by the dotted line, indicating the linear correlation constraint on the gradient calculation in the frequency domain. By doing this two FFT operations are removed at the expense of performing a circular correlation to determine the instantaneous gradient. Depending on the performance desired, this method may be beneficial from a computational standpoint.

The third and least accurate method of frequency domain adaptive filtering is the circular LMS algorithm as shown in Figure 3.24.

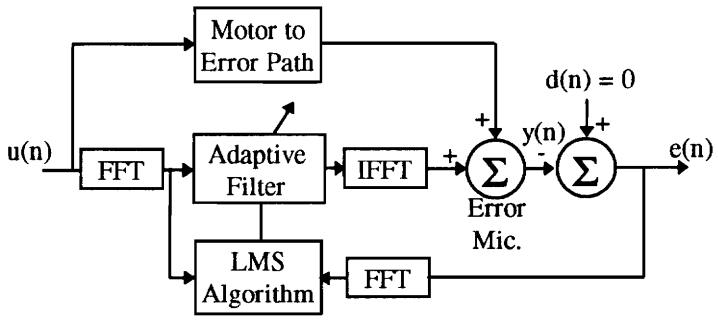


FIGURE 3.24 FREQUENCY DOMAIN LMS ALGORITHM

Both the convolution and correlation operations are circular, adding significantly to the expected error in the time domain. The computational savings realized from the circular method arise from the *size* of the FFT. Because the convolution operation is circular, the FFT operations are only required to be of length N, unlike the operations shown in Figure 3.22 which involve zero padding and long input data sequences. None of the output data is discarded. The detriments of this method as well as the unconstrained algorithm are realized in Section 3.5.3 where simulations demonstrate the added error found in circular convolution and/or correlation. Tradeoffs can be considered in terms of the computational complexity.

To summarize the computational complexity of the three algorithms, consider first an adaptive FIR filter of order M. The conventional LMS algorithm requires M multiplication's to compute the output of the adaptive filter and M multiplication's to update the weight vector. This results in a total of 2M multiplication's for *each* output

sample to be calculated. Therefore, $2M^2$ multiplication's are required to calculate M output samples using the conventional time domain LMS algorithm.

Assuming 50% overlap using the overlap save method, the fully constrained frequency domain LMS requires 5 FFT (or IFFT) operations each of length 2M. In addition, computation of the output vector and correlation process each require 4(2M) multiplication's (in the frequency domain) to compute a block of output data M points long. Since each FFT (or IFFT) operation requires N $\log_2(N)$ multiplication's (for an N point DFT), the total number of multiplication's required then becomes 5(2M) $\log_2(2M)$ + 8(2M). Removing the gradient constraint also removes the need to compute two FFT operations bringing the total number of multiplication's to 3(2M) $\log_2(2M)$ + 8(2M) for the unconstrained frequency domain algorithm. Finally, the circular convolution method reduces the required FFT size from 2M to M bringing the total number of computations to 3(M) $\log_2(M)$ + 8(M). Table 3.1 summarizes the computational comparisons of each of these methods for various order filters.

TABLE 3.1 NUMBER OF MULTIPLICATION'S FOR LMS ALGORITHMS

	M	**16**	**32**	**128**	**1024**
Time Domain LMS	$2M^2$	512	2,048	32,768	2,097,152
Fully Constrained	5(2M) $\log_2(2M)$ + 8(2M)	1,056	2,433	12,288	130,023
Unconstrained	3(2M) $\log_2(2M)$ + 8(2M)	736	1,665	8,192	83,886
Circular	3(M) $\log_2(M)$ + 8(M)	320	735	3,703	39,846

The computational advantages of using the frequency domain algorithms become apparent for all methods for filter orders above 32. If broadband control of sound is desired, many more weights are typically required, usually on the order of 1000. For any of the implementations of the frequency domain algorithm, computational savings are quite significant with filters of this size.

It is clear that the block multiplication of the vectors **U(k)** and **E(k)** in the weight update calculates a vector of weights in the frequency domain for each block of M input data

points. The FFT computed for (say) 32 points of input data provides non-duplicate magnitude and phase information for only 16 frequency bins. This will correspond to 32 weights calculated, 2 weights per frequency bin one each for magnitude and phase. The seemingly duplicate information is required to perform the IFFT operation without loss of information or aliasing.

It should be clear that there is a tradeoff between computational complexity and update speed. Since a data block of **2M** points of data is required, there is a control output delay of the same number of samples. Therefore, if the input or plant is changing faster than the algorithm is able to update the control output, the performance will suffer. The benefit in applying this approach is found when the computation of the convolution in the time domain begins to limit the sampling rate capabilities of the DSP. Implementing a frequency domain controller will reduce the number of computations per data point but will limit tracking capabilities for non-stationary inputs. Since a stationary ergodic assumption is necessary for implementation of the COP filters, non-stationarity will not be considered. As will be seen in Chapter 5, frequency domain calculation of the COP filters is required, thus a frequency domain controller will help reduce the total number of computations necessary for the entire process.

The approaches mentioned above have been compared analytically as well as computationally. It has not been mentioned quantitatively however, how each method performs versus one another. As will be seen in the final section, simulation of the frequency domain controllers adequately describes their performance in the light of circular versus linear operations.

As a conceptual introduction consider a sine wave or tonal disturbance. For a single tone residing on a bin center, circular convolution will produce no leakage and the algorithm will converge as illustrated in the next section. If however, the input (reference) frequency does not fit into the sampling window exactly, the effects of circular convolution will cripple the convergence of the frequency domain LMS to a point which provides unacceptable levels of error. Adding the constraint of linear convolution, the unconstrained LMS provides a

slightly reduced error. Adding the full gradient constraint will produce a minimum error identical to that of the conventional time domain LMS, even for tonals off bin center.

3.5.3 MATLAB SIMULATION

Although the overlap-save method is the preferred implementation of the convolution operation between the inputs and filter weights, the unconstrained method will work provided the input is on a bin center. Typically (in practice) this can never be assured due to non-stationarity of the input or desire for broadband control. Appendix A contains the MATLAB code for all of the following simulations.

The first case examined uses the unconstrained algorithm with a filter length (and thus FFT point size) of 64 at a sample rate of 16 Hz. Locating the input frequency at 1 Hz ensures the interference will reside on a bin center and the unconstrained problem should suffice without realizing the detrimental effects of the circular convolution and correlation operations. A convergence parameter was chosen in order to demonstrate the adaptation process visually, emphasizing the block adaptive output which remains constant over the block duration.

The reference frequency and plant model are shown in Figure 3.12 and are the same as those used in the prior two simulations, thus are not repeated here. Figures 3.25 and 3.26 illustrate the convergence of the control signal and the error signal respectively.

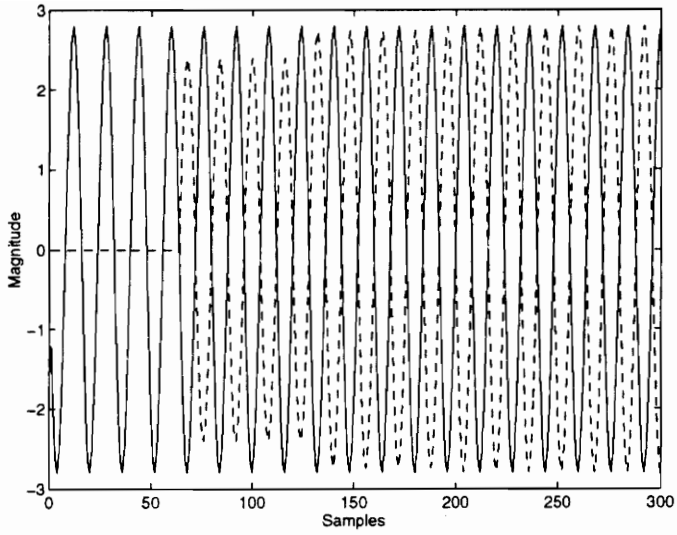


FIGURE 3.25 CIRCULAR FREQUENCY DOMAIN LMS PLANT AND CONTROL (DOTTED)

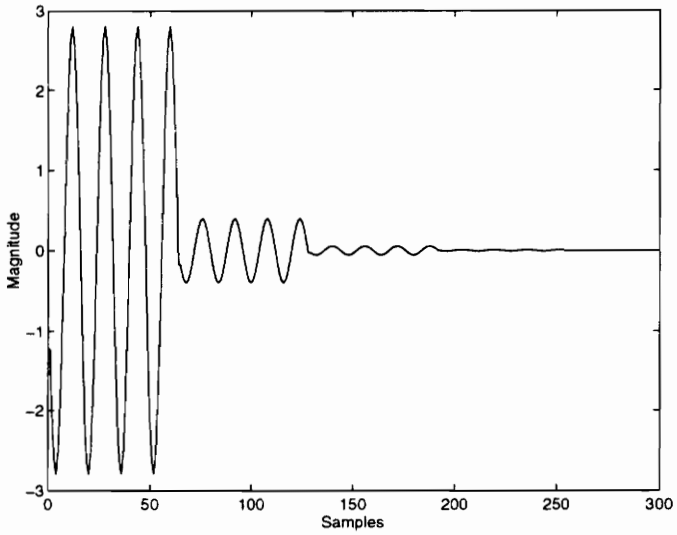


FIGURE 3.26 CIRCULAR FREQUENCY DOMAIN LMS ERROR

The control (and thus error) only change every block of 64 samples as was predicted; and the effects of circular convolution are not apparent as the error signal converges to zero. The value of the convergence parameter and number of FFT points (order) can be chosen so the optimal solution is reached in a fewer number of samples. However, using FFT sizes

below 16 points removes the computational advantages over the conventional LMS and would be somewhat counter-productive, given the poor tracking capabilities of block adaptive filtering.

To illustrate the poor convergence the unconstrained algorithm exhibits, a second trial is used. Moving the interference (and thus the reference) frequency to an off bin center value (from 1 Hz to 1.33 Hz) it is expected that the circular convolution effects will not be negligible. Figure 3.27 shows the error microphone signal for such a case. The leakage between the blocks of data creates error that the circular convolution assumption cannot prevent. (If a similar test were run using the conventional LMS algorithm, the 1.33 Hz interference presents no problem in acceptable convergence).

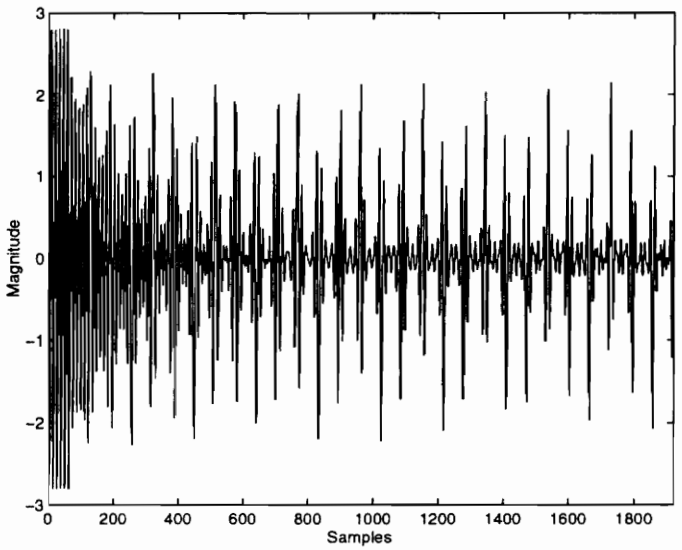


FIGURE 3.27 CIRCULAR FREQUENCY DOMAIN LMS ERROR

Continuing with this same example, the converged value of error for off bin center frequencies can be reduced by incorporating the linear convolution operations via the overlap save technique. Using the same convergence rate and the unconstrained version of the frequency domain LMS, Figure 3.28 illustrates the error signal after convergence.

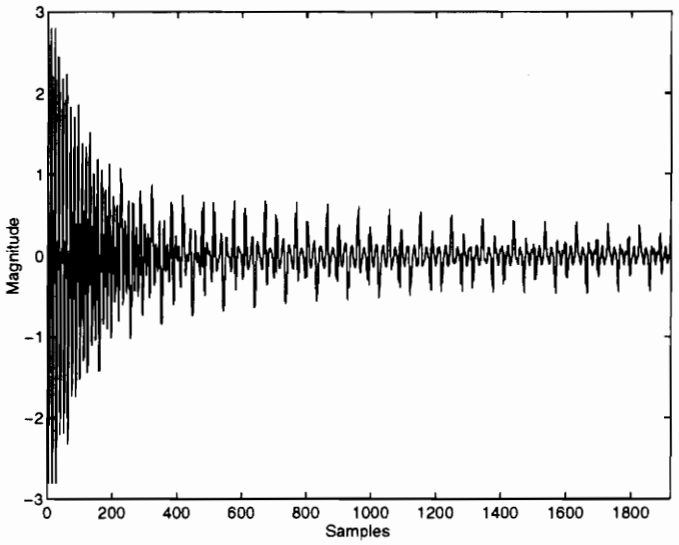


FIGURE 3.28 UNCONSTRAINED FREQUENCY DOMAIN LMS ERROR

While the error is significantly reduced from the circular version shown in Figure 3.27, it is still not sufficient to cancel the tonal entirely due to the linear correlation operation in the calculation of the gradient. Whether this level of error can be tolerated given the computational savings of Table 3.1 is application dependent.

Adding the final linear correlation constraint on the gradient to the linear convolution filtering operations, results in the fully constrained frequency domain LMS algorithm. For an off bin center disturbance (1.33 Hz), the error is shown to converge as in the time domain LMS using fewer computations in Figure 3.29.

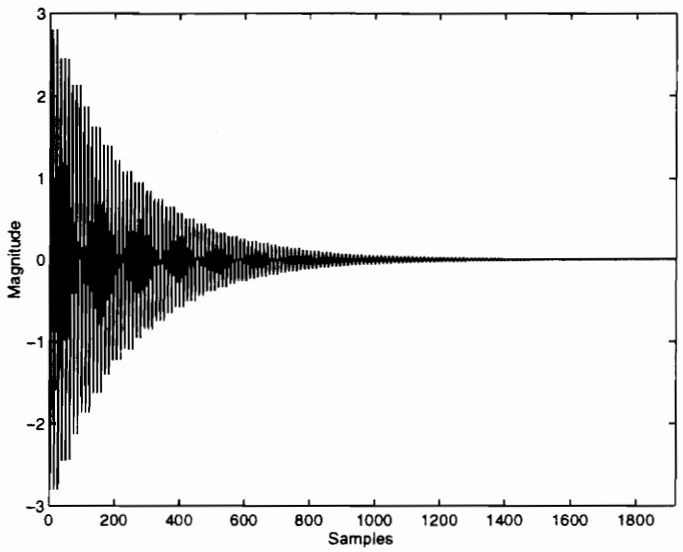


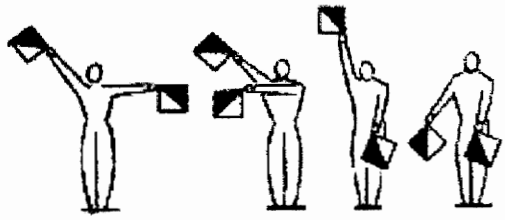
FIGURE 3.29 FULLY CONSTRAINED FREQUENCY DOMAIN LMS ERROR

The frequency domain (block adaptive) LMS algorithm presents a computationally advantageous alternative to the conventional LMS algorithm. The efficient convolution and correlation operations performed in the frequency domain require a slightly different structure and implementation using 5 spectral transformations (for the overlap save method). For broadband control (or any non-simulated control) the unconstrained alternative is not acceptable due to excessive leakage. Additionally, the buffering delay required to collect data may preclude its use in rapidly time varying or non-stationary systems.

The COP filter implementation requires a detailed spectral analysis of the candidate references with respect to each other and the disturbance noise. The frequency domain result fits well with the frequency domain LMS algorithm in that the FFT of the input data has already been performed. In addition, the computational savings achieved in using the frequency domain LMS will help offset the computations required to design the COP filters. Before proceeding with the COP filter design and implementation, Chapter 4 describes all the technical analog signal processing background.

Chapter 4

Signal Processing



Analog and digital controls, system identification (frequency response or time domain determinations), and energy source path classification all rely on the ability to perform accurate signal processing on a given system. The contributions made by an effective signal processing analysis are all too often de-emphasized in the engineering community. The benefits of understanding signals and their behavior with each other and the environment are numerous when applied to many engineering applications. Special application of correlation and spectral analyses to control systems (specifically acoustic noise control) will be presented as a precursor to the coherent output power filter development.

Spectral analysis as applied to system identification is an integral part of the development of the COP filters described in Chapter 5. Before proceeding it is necessary to present the pertinent background information used in developing the filter design. Following, this chapter introduces some common terminology and definitions used throughout the remainder of this work. Efforts have been made to apply standard notations and definitions as used in [3, 6] to the ideas presented herein. Bendat and Piersol are considered to be among the few experts in the field of (analog) signal processing and their notation has been adopted throughout the engineering community.

Given the background of random signal analysis and the necessary assumptions, correlation measures and ideal system relationships will be presented. These developments are all required in order to explain the coherence function and its related measures of partial and multiple coherence as discussed in Section 4.3. To fully describe the acoustic noise control problem most frequently encountered, the energy source identification problem is then explained. Various physical structures and configurations of this problem exist and will be

separated as case studies. Each of these instances will be defined and analyzed prior to introducing the COP filter design methodology in Chapter 5.

4.1 INTRODUCTION

4.1.1 DEFINITIONS AND TERMINOLOGY

Physical systems such as vibrating machinery, acoustic systems or even the stock market, produce data which can be classified as either *deterministic* or *non-deterministic*. Simply put, future events of deterministic systems can be predicted with reasonable accuracy within some standard deviation. Thus deterministic data can be described by some mathematical function of the variable in question, typically time.

To illustrate the meanings behind the various classification of deterministic data, consider an ideal mass, spring, damper system. This type of system can be completely defined by a simple second order differential equation as seen in Equation 4.1.

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t) \qquad \textbf{(4.1)}$$

Solving this non-homogeneous differential equation yields an expression which can describe the position, velocity, and acceleration of the mass as a function of time and the exciting force *F(t)*, also a function of time. Although physical systems are typically regarded as being time dependent, the time variable can be substituted by another variable as necessary.

If the forcing function in Equation 4.1 were eliminated and the system was initialized from some arbitrary position (i.e. free response), the non-homogeneous solution would oscillate (depending on the damping) to rest in agreement with Equation 4.2.

$$x(t) = C_1 e^{(-\xi+\sqrt{\xi^2-1})\omega_n t} + C_2 e^{(-\xi-\sqrt{\xi^2-1})\omega_n t} \quad \textbf{(4.2)}$$

The constants are dependent upon the initial conditions (velocity and position). As time approaches infinity it is easily seen that x(t) approaches zero. This type of data, while deterministic, is considered *non-periodic.* Within the umbrella of systems described by non-periodic data, are data considered to be "almost-periodic" and "transient" [6]. Transient data is similar to that which is described by Equation 4.2 which can be described by some time varying function. Almost-periodic data can typically be described by a time varying equation expressed as a function of frequency. (This can be thought of as two unrelated sinusoids combined in time where the phase is shifted with respect to each other).

Considering Equation 4.1 without the velocity term (i.e. undamped) the solution to the homogeneous second order differential equation will conform to Equation 4.3.

$$x(t) = Ce^{\pm j\omega_n t} \quad \textbf{(4.3)}$$

This type of deterministic, or mathematically defined, data is considered to be *periodic.* Within the classification of deterministic periodic data exists both *sinusoidal* and *complex sinusoidal.* Each of these types of time-varying data repeat themselves at some specified period in time and can thus be described as a single sinusoid (Equation 4.3) or as a sum of multiple sinusoids as in a Fourier series.

Acoustic systems are typically much more complicated than the simple second order system used to illustrate deterministic data concepts above. Often, sound fields are considered to be non-deterministic, as their magnitude and phase is dependent on variables such as room dynamics and measurement location. Because of this, the data analysis/estimation procedures in acoustics are based on statistics rather than mathematical equations.

Non-deterministic, or *random* data, cannot be explicitly described by a mathematical equation. Within the classification of random data, processes can be either *stationary* or *non-stationary.* Although mathematical equations may not be adequate to describe some random

processes, for many types of systems statistical measures *can* be used to predict future events based on prior data. These instances can be classified as stationary random data sequences and are the type with which acoustical (and vibrational) systems are most frequently concerned.

Analytical developments and mathematical relations for stationary random processes will be developed in Section 4.2. In somewhat practical terms, stationary processes can be further divided into two subgroups being either *ergodic* or *non-ergodic.* Illustrations of these terms and a definition of stationary data can be facilitated via an arbitrary plot of time history ensembles of a random data process.

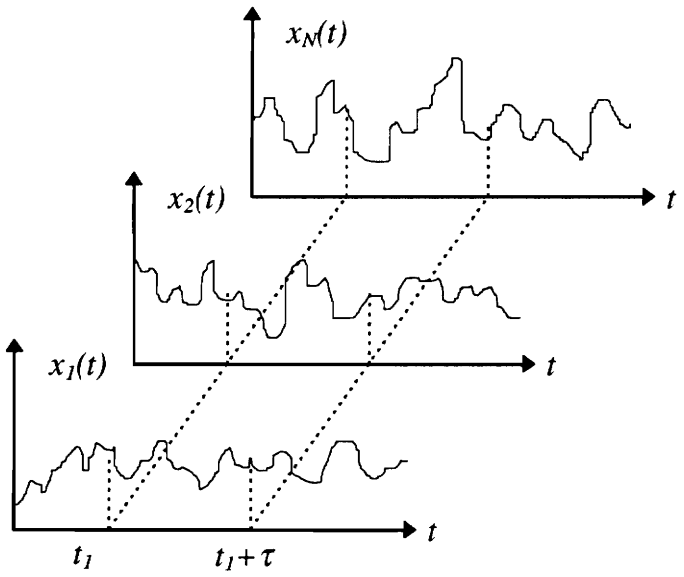


FIGURE 4.1 RANDOM PROCESS ENSEMBLE

Figure 4.1 represents all the possible time history records $x_1(t)$, $x_2(t)$, ... $x_N(t)$ which have been recorded for a given system. (These individual records are also termed "sample records" or "sample functions")[6]. The set of all these records is then termed an *ensemble.* Two statistical descriptions should be used at this point in order to define the process. For the ensemble in Figure 4.1, an average over the set of sample records at time t_1 is defined as the mean value μ, as shown in Equation 4.4. The *autocorrelation function* represents a correlation between two values of the ensemble as shown in Equation 4.5.

$$\mu_x(t_1) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} x_k(t_1) \tag{4.4}$$

$$R_x(t_1, t_1 + \tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} x_k(t_1) x_k(t_1 + \tau) \tag{4.5}$$

Equation 4.5 will be explained in more detail and related to physical values in the following section. Currently the autocorrelation can be considered to be a measure of the periodic nature of the signal, i.e. how related is a current (ensemble averaged) sample to a sample which will occur at τ from the present. If the values in Equations 4.4 and 4.5 vary as the time t_1 varies, the process is considered to be non-stationary. Should the mean value and autocorrelation remain constant, independent of changes in the observation time t_1, the process can be considered as *weakly stationary* (a.k.a. stationary in the wide sense). In order for the random process to be classified as stationary in the strict sense, or *strongly stationary*, all the possible sample records must be used to calculate all the possible statistical moments and each should be time invariant. Since it is both impractical and impossible to obtain an infinite amount of data, weakly stationary signals are typically assumed to be strongly stationary for most applications.

Ergodicity refers to the ability of *one* sample record to accurately represent the statistical moments of an ensemble averaged data set. Suppose Equations 4.4 and 4.5 are rewritten to represent the same values (mean and autocorrelation) for a single sample record. They can then be expressed as an integration over time as in Equations 4.6 and 4.7.

$$\mu_x(k) = \lim_{T\to\infty} \frac{1}{T} \int_0^T x_k(t) dt \tag{4.6}$$

$$R_x(\tau,k) = \lim_{T\to\infty} \frac{1}{T} \int_0^T x_k(t) x_k(t+\tau) dt \qquad \textbf{(4.7)}$$

If the random process is considered stationary by the criteria introduced above, it can be considered ergodic if Equation 4.6 equals the result of Equation 4.4 and likewise for the autocorrelation measures. This is to say that a process is ergodic if a single sample record has the same mean and autocorrelation (statistical moments) for adjacent sample records as well as ensemble averages. This property of random data is convenient in the sense that less data is required to completely describe the statistical nature of the system. If these properties are not commensurate with the observed behavior, and the system demonstrates stationary properties, the data is considered to be non-ergodic.

The other classification of random signal data falls under the category of non-stationary data. If the sampled data fails to qualify for any of the above metrics and is clearly random (without mathematical description) it is considered non-stationary. Current technology prevents accurate representation of this type of data, as one or more of the statistical moments changes with time. It would be necessary to take an infinite number of sample records to create an ensemble average representative of the system. In the near future artificial neural networks may provide a useful tool in assessing these types of data records.

4.1.2 Scope and Assumptions

As previously mentioned, the most commonly encountered acoustic (noise) environments can be classified as random stationary processes. Deterministic mathematical expressions are far too complex to accurately describe mode shapes of common room acoustics beyond the first few fundamental modes. Depending on certain room dimensions and the associated environment, the three dimensional wave equation for a reverberant enclosure can only be solved using finite element methods. These concepts should be understood by the noise control engineer but are unnecessary in performing the analysis required to develop the COP filters.

In addition to being random stationary processes, acoustic systems (exhibiting steady state behavior) can also be classified as ergodic. The assumption of a random stationary ergodic process is the basis of the ability to perform accurate system measurements. Developments in the remainder of this chapter will subscribe to this assumption without restricting it's application to acoustic systems. Vibration, analog-to-digital conversion, system identification, and adaptive signal processing all rely on the assumption that the data is random stationary ergodic.

The remaining classifications of systems including deterministic, stationary non-ergodic, and non-stationary random processes will not be considered. Deterministic processes represent a class of relatively simple systems which can be identified via Newton's laws or sophisticated system identification methodologies. Non-ergodic and non-stationary systems theoretically require an infinite amount of data to make assumptions about the system behavior. The spectral analysis and correlation methodologies to be developed *do not apply* to non-ergodic or non-stationary data.

As with any statistical treatment of data, errors resulting from these estimates contribute to the overall uncertainty of the results. Clearly it is impossible to collect an infinite amount of data as required by Equations 4.6 and 4.7; thus, the calculation of the mean value will be inaccurate to some extent. These types of errors resulting from having too little data to accurately represent the true statistical moments will be ignored for simplicity in the analysis. It will thus be assumed that enough data has been collected in each of the simulations and experiments to represent the system appropriately. Bendat and Piersol provide an excellent coverage of these types of errors in [3]. Further assumptions to be made regarding the statistical analysis will be stated in Section 4.2.

Many of the developments in the following section begin with the time domain statistical moments and are then extended to the frequency domain for a more practical illustration of systems. In depth descriptions of Fourier analyses and the sampling theorem will be forgone in these developments. A detailed treatment of these areas can be obtained in [7]. All of the material presented throughout Sections 4.2 and 4.3 is summarized from [3, 6].

This is necessary in order to accurately and effectively explain energy source identification problems and the development of the COP filters in Section 4.4 and Chapter 5, respectively.

A final indication regarding terminology should be clarified before proceeding. Throughout the remainder of this chapter, the term "noise" will refer to uncorrelated content unless otherwise used in a noise *control* context. Uncorrelated content refers to measured, unknown random data existing in the measurements of any signals. This will become clear within the context of the material.

4.2 SPECTRAL ANALYSIS METRICS

4.2.1 AUTOCORRELATION AND CROSS-CORRELATION FUNCTIONS

The autocorrelation function briefly introduced in the preceding section will be defined here for stationary ergodic random data via the covariance function. Since the autocorrelation function can be classified as a special case of the cross-correlation function, the cross-correlation will be examined first.

To establish a linear relationship between two sets of data (not necessarily time-based) the *covariance function* can be used. This is defined as seen in Equation 4.8.

$$\sigma_{xy} = E[(x-\mu_x)(y-\mu_y)] = \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y) \qquad \textbf{(4.8)}$$

The E denotes the expected value operator which is a statistical averaging operation over all possible samples. Each data sequence has an associated mean value (μ) which is used to determine the average product of the difference in data sequence values with their respective means over the entire data set. For two unrelated data sets the summation will result in a zero value for covariance as the positive and negative products oppose each other equally. In order to adapt the covariance function to time sequences it needs to be evaluated for a given delay term τ. This time delay represents the period at which

correlation may or may not exist between the two time history records. Applying Equation 4.8 to two time history data sequences for a given time delay τ, Equation 4.9 shows the covariance function between *x(t)* and *y(t)*.

$$\begin{aligned} C_{xy}(\tau) &= E[(x(t)-\mu_x)(y(t+\tau)-\mu_y)] \\ &= \lim_{T\to\infty}\frac{1}{T}\int_0^T (x(t)-\mu_x)(y(t+\tau)-\mu_y)dt = R_{xy}(\tau)-\mu_x\mu_y \end{aligned} \quad \textbf{(4.9)}$$

R_{xy} is then defined in Equation 4.10.

$$R_{xy}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)y(t+\tau)dt \quad \textbf{(4.10)}$$

Equation 4.10 is defined as the cross-correlation function between the two time history data sequences *x(t)* and *y(t)*. Equation 4.10 provides a quantitative measure of how closely two data sequences are related for a given time delay (or period) τ. Thus it should be intuitive that if a sinusoid exists in each data set which has a period τ, the cross-correlation function calculated at that delay will be greater than at other delay points.

Should the mean value of either data sequence equal zero, Equation 4.9 will equal Equation 4.10 (i.e. the covariance function will equal the cross-correlation function). There are special methodologies in dealing with non-zero mean data sequences which will not be addressed here. Further developments will assume time history data sequences to have zero-mean values, since this is entirely achievable as a pre-processing task in practical applications. The added level of complexity of non-zero mean values can be reviewed in [3].

As should be apparent given the necessary terminology, the autocorrelation is simply a special case of the cross-correlation of Equation 4.9. If x(t) is equal to y(t), the autocorrelation can be rewritten as in Equation 4.11 assuming a zero mean value.

$$C_{xx}(\tau) = E[(x(t))(x(t+\tau))] = \lim_{T\to\infty} \frac{1}{T}\int_0^T x(t)x(t+\tau)dt = R_{xx}(\tau) \qquad \textbf{(4.11)}$$

Thus for a zero mean data sequence the autocovariance equals the autocorrelation and the terminology is interchangeable. Future references to Equation 4.11 will be termed as autocorrelation.

To present a physical interpretation of the autocorrelation function, Figure 4.2 illustrates $R_{xx}(\tau)$ plotted versus τ for a simple sine wave and for wide-band random noise.

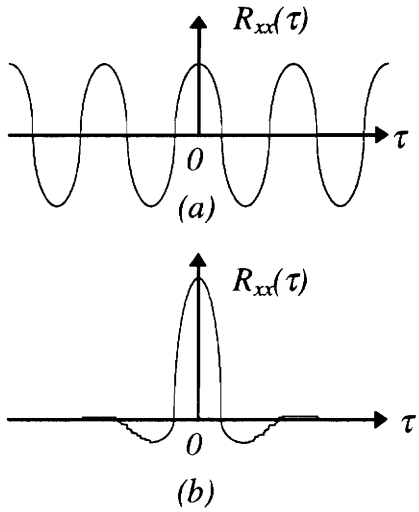


FIGURE 4.2 AUTOCORRELATION EXAMPLES

Figure 4.2a illustrates that for a delay of zero, the relationship of the data sequence x(t) with itself is a maximum, (i.e. the current sample can accurately predict the current sample). For a time delay equal to the period or integer multiple thereof, the current sample can also predict the future sample at that point. This is certainly intuitive when discussing a sine wave. Figure 4.2b is the autocorrelation for random noise which approaches the delta function for an infinite time sequence. The physical interpretation is that the current sample can only accurately predict the current sample, not a sample at any time advance into the future.

While these relationships are simple to understand in the time domain, the frequency domain versions of the autocorrelation and cross-correlation offer a more practical and useful interpretation in terms of the frequency variable.

4.2.2 AUTOSPECTRUM AND CROSS-SPECTRUM

The Fourier transform of a data sequence represents the frequency domain version of the signal as a complex number. The Fourier transform of a time series is shown in Equation 4.12.

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \qquad \textbf{(4.12)}$$

The result is a complex number which can represent the magnitude and phase of the signal in the frequency domain using appropriate mathematical relations. The Fourier transform of the cross-correlation function described in Section 4.2.1 is termed the *cross-spectral density function* or the *cross-spectrum*. The notation for this quantity (as a function of frequency) is $S_{xy}(f)$ as shown in Equation 4.13.

$$S_{xy}(f) = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-j2\pi f\tau}d\tau \qquad \textbf{(4.13)}$$

This represents the two-sided function which includes both positive and negative frequencies. Physically, the negative frequency data in a Fourier transform has no direct meaning. However, the inverse Fourier transform requires this data to maintain resolution when converting back to the time domain.

For the special case where $x(t)$ is equal to $y(t)$, the cross-spectrum becomes the *autospectrum* as shown in Equation 4.14 for the two sided result.

$$S_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau \qquad \textbf{(4.14)}$$

Since it is impossible to satisfy either Equation 4.14 or 4.13 by integrating over all possible time delays, S_{xx} and S_{xy} can be arrived at by more practical methods via the Fourier transform. Assuming stationary ergodic random data records of *x(t)* and *y(t)*, the auto and cross-spectral density functions can be expressed as

$$S_{xy} = \lim_{T\to\infty} \frac{1}{T} E[X_k^*(f,T) Y_k(f,T)] \qquad \textbf{(4.15)}$$

$$S_{xx} = \lim_{T\to\infty} \frac{1}{T} E[\left|X_k(f,T)\right|^2] \qquad \textbf{(4.16)}$$

where,

$$X_k(f,T) = \int_0^T x_k(t) e^{-j2\pi ft} dt \qquad \textbf{(4.17)}$$

$$Y_k(f,T) = \int_0^T y_k(t) e^{-j2\pi ft} dt \qquad \textbf{(4.18)}$$

and the asterisk is the conjugate operator. Practically speaking the ergodic assumption must still be valid for this to produce accurate results. Finite length Fourier transforms operate on a *single* data set, which for it to accurately represent the system, the data must be ergodic.

Relating this to the time domain calculations of spectra in Equations 4.13 and 4.14, one can recognize that convolution in the time domain is equivalent to multiplication in the frequency domain. This property was introduced in Chapter 3 via the frequency domain adaptive algorithm which reduces computational overhead by performing efficient Fourier

transforms and multiplying. Since finite length Fourier transforms are being used to represent infinite amounts of data in a single sequence, it is necessary to average this data over T records. It should be emphasized that this is a different format than ensemble averaging as required by non-ergodic data. The assumption of ergodicity is still in place for these operations; however, a single record is now divided into multiple "windows" (represented by subscript k) in order to obtain an average that represents what Equations 4.13 and 4.14 might result in should they have the ability to take an infinite amount of data.

Clearly the problem introduced with this analysis is that an infinite number of averages would be required by Equations 4.15 and 4.16 in order to represent an infinite sequence, unless T (the transform size) were infinite. Typically, for random data fields, sixty averages is considered adequate for stationary data and moderate sized Fourier transforms. For acoustical environments this number will vary depending on the behavior of the noise field.

This is the methodology most commonly used to implement calculation of the spectral density functions. The primary difference in implementation and what is described here is that the data is typically discretized and the operations are performed digitally. The Fast Fourier Transform (FFT) is an efficient implementation of the discrete Fourier Transform as previously introduced in Chapter 3.

4.2.3 DESCRIPTIONS OF THE IDEAL SYSTEM

Efforts have been made to reduce the seemingly abstract concepts with practical instances. This is a difficult task without using actual physical systems, but rather discussing applications to arbitrary data. Applying some of these quantitative measures to an actual physical system should effectively illustrate some of the benefits of performing the above analyses.

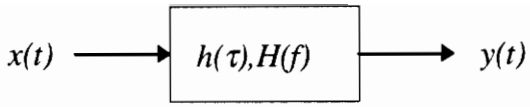


FIGURE 4.3 IDEAL SYSTEM

Figure 4.3 illustrates a very simplistic block diagram where h(τ) represents the impulse response of the system and H(f) represents the frequency response (i.e. the Fourier transform of the impulse response). The input time signal *x(t)* is thus modified in some way by the system to produce *y(t)*. For now it will be assumed that the system does not change with time, (i.e. it is linear time invariant or LTI). If this is the case, then for stationary ergodic inputs (such are the type in question), *y(t)* will also be a stationary ergodic signal and will be defined by the convolution integral as given in Equation 4.19.

$$y(t) = \int_0^\infty h(\tau)x(t-\tau)dt \qquad \textbf{(4.19)}$$

As was previously mentioned, convolution in the time domain is equivalent to multiplication in the frequency domain. The ideal system relationships for the single-input single-output (SISO) system can be developed in either the time domain (using the auto and cross-correlations) or equivalently in the frequency domain using the Fourier transform [3]. Only the frequency domain developments are introduced here.

Following the initial transients of the LTI SISO system in Figure 4.1, the input and output time series can be expressed in terms of their frequency domain counterparts as shown in Equation 4.20.

$$Y(f,T) = H(f)X(f,T) \qquad \textbf{(4.20)}$$

Using this frequency domain relationship the following subsequent Equations 4.21 can be developed.

$$Y^*(f,T)Y(f,T) = H^*(f)X^*(f,T)H(f)X(f,T)$$
$$X^*(f,T)Y(f,T) = X^*(f,T)H(f)X(f,T) \quad \textbf{(4.21)}$$

If each equation is averaged over a theoretically infinite number of sample windows as described by Equations 4.15 and 4.16, using the expectation operator, the results are seen as Equations 4.22 and 4.23

$$S_{yy}(f) = \left|H(f)\right|^2 S_{xx} \quad \textbf{(4.22)}$$

$$S_{xy}(f) = H(f)S_{xx} \quad \textbf{(4.23)}$$

These relations will be used extensively for the remainder of this work. These are important relations which describe linear systems in terms of the *frequency response function H(f)* and the auto and cross-spectrums, all of which are measurable by methods previously mentioned. It is instructive at this point to reiterate the conditions under which these particular relations are valid.

Clearly the autospectrum and cross-spectrum were developed under the assumption of stationary ergodic signals. The input can be assumed to be as such as can the output provided the system is operating in a steady state, noiseless, and LTI manner. The assumption of steady state is typically valid, as transients die out very quickly unless the system (filter) is unusually large. Operating in a noiseless environment is, however, typically an erroneous assumption and the impact of measurement noise must at least be considered. Measurement noise will be addressed in Section 4.3. The matter of system linearity can become quite complex and is discussed in detail in Section 4.3 as coherence. Since the measure of system linearity is directly proportional to the effectiveness of an adaptive controller (in any setting), the discussion in Section 4.3 strictly addresses ways to quantify this linearity. It should be noted that adaptive filters, by their very nature, are considered to be time variant. Therefore the linear ideal system relationships presented above will not

apply to outputs of adaptive filters. (Typically feasibility analysis occurs prior to control which is precisely the basis of the COP filters).

Before continuing with a more detailed discussion of system linearity a few words regarding notation should be considered. It was mentioned previously that the notation defining the auto and cross-spectral density functions has been standardized as using S subscripted by the variable(s) for which it identifies. This is considered to be the two-sided spectrum, containing data for the positive *and* negative frequencies in the Fourier transform. It can be shown that the one-sided spectra (including only positive frequency) data for both the auto and cross-spectral density functions conform to

$$\begin{aligned} G_{xy}(f) &= 2S_{xy}(f) \\ G_{xx}(f) &= 2S_{xx}(f) \end{aligned} \tag{4.24}$$

for all positive and non-zero frequencies where G denotes the one-sided frequency spectra[6]. Given these relationships the ideal system relationships can be rewritten as Equations 4.25 and 4.26.

$$G_{yy}(f) = \left|H(f)\right|^2 G_{xx} \tag{4.25}$$

$$G_{xy}(f) = H(f)G_{xx} \tag{4.26}$$

As expected the one-sided vs. two-sided spectra result in the same system relationships. Strictly as a matter of convenience and physical system understanding the one-sided spectra will be used for the remainder of the analytical developments regarding coherence and COP filter development. The negative frequency content of the two-sided spectra holds no physical meaning but can be substituted in any expressions which are defined by the one-sided spectra and will provide the same result.

4.3 A STUDY IN COHERENCE

Strictly speaking, the ordinary coherence function is a quantitative measure of the linearity existing between two signals as a function of frequency. This measure is quite useful in many applications which involve the need to recognize physical processes, system identification and specifically adaptive feedforward control. It is shown in Chapter 5 that certain coherence measures can directly predict the effectiveness of any given control configuration based on the reference signal and the error microphone. This was briefly mentioned in Chapter 3 and more closely examined as a measure of control in Chapter 5.

The immediate developments are primarily concerned with developing a working analytical knowledge of the various coherence measures as applied to all applicable generalized systems. With this, any specific system configuration which will be encountered within the confines of the defined noise control problem can be analyzed with the same methodology. The ordinary coherence is first introduced and developed as the most rudimentary and easily understood measure of system linearity. Effects of noise are examined for the single-input single-output (SISO) system. Multi-input single-output (MISO) system identification is then addressed in the context of partial coherence functions.

4.3.1 ORDINARY COHERENCE

The ordinary coherence function follows from the development of the *cross-spectrum inequality* which is developed now [6]. It should be re-emphasized that each of the random data measures including all spectral density functions are valid only for stationary ergodic random data. Therefore the following results including the ordinary coherence function and partial coherence functions rely strictly on the same assumption.

Equations 4.17 and 4.18 represent the Fourier transforms of the real valued time sequences $x(t)$ and $y(t)$. For any real valued constants a and b and the phase between $x(t)$ and $y(t)$ defined as $\theta_{xy}(f)$, it can be stated that Equation 4.27 is valid for all frequencies.

$$\frac{1}{T}E[\left|aX_k(f,T)+bY_k(f,T)e^{j\theta_{xy}(f)}\right|^2]\geq 0 \qquad \textbf{(4.27)}$$

Expanding the squared term, allowing T to approach infinity, and using the relations provided in Equations 4.15 through 4.23 gives the inequality shown in Equation 4.28.

$$a^2S_{xx}(f)+2ab\left|S_{xy}(f)\right|+b^2S_{yy}(f)\geq 0 \qquad \textbf{(4.28)}$$

Dividing both sides of Equation 4.28 by b^2 (assuming $b \neq 0$) Equation 4.29 provides a quadratic equation in (a/b).

$$\left(\frac{a}{b}\right)^2 S_{xx}(f)+2\left(\frac{a}{b}\right)\left|S_{xy}(f)\right|+S_{yy}(f)\geq 0 \qquad \textbf{(4.29)}$$

Reverting for a moment to the solution of a quadratic equation such as the one shown in Equation 4.29; it can be shown that since the equation is greater that zero, the roots which solve the equation cannot be real. Given this, the *discriminant* of a quadratic equation is defined as the quantity under the radical in the expression for the roots which solve Equation 4.29. Since the roots of equation 4.29 must be imaginary, the discriminant must be less than or equal to zero to produce a square root of negative one. Equation 4.30 expresses this inequality.

$$4\left|S_{xy}(f)\right|^2-4S_{xx}(f)S_{yy}(f)\leq 0 \qquad \textbf{(4.30)}$$

Equation 4.30 can be rearranged to prove the cross-spectrum inequalities shown in Equations 4.31 and 4.32. Equation 4.32 derives from substituting Equations 4.24 into 4.31.

$$\left|S_{xy}(f)\right|^2 \leq S_{xx}(f)S_{yy}(f) \tag{4.31}$$

$$\left|G_{xy}(f)\right|^2 \leq G_{xx}(f)G_{yy}(f) \tag{4.32}$$

The ordinary coherence function can now be defined as a real valued function of frequency as shown in Equation 4.33

$$\gamma_{xy}^2(f) = \frac{\left|G_{xy}(f)\right|^2}{G_{xx}(f)G_{yy}(f)} \tag{4.33}$$

Given the cross-spectral inequality in Equation 4.32, bounds can be placed on the coherence function as it can only take on values between zero and one. Should the limit of Equation 4.32 be reached as the cross-spectrum is equal to the two autospectra, the coherence function would clearly become one.

To illustrate this extreme for ordinary coherence, revisit the ideal system as shown in Figure 4.3. Under the assumptions made by stationary random ergodic data and having a LTI noiseless system, Equations 4.25 and 4.26 remain valid. Substituting these ideal system relations into the ordinary coherence function, the result is unity for all frequencies.

$$\gamma_{xy}^2(f) = \frac{\left|H(f)G_{xx}(f)\right|^2}{G_{xx}(f)\left|H(f)\right|^2 G_{xx}(f)} = 1 \tag{4.34}$$

Physically speaking, for all frequencies the output (G_{yy}) is directly correlated or coherent with the input (G_{xx}). That is, it can be said with confidence that a relatively noiseless linear system exists between the measured input and measured output time series.

Before analyzing the ordinary coherence measure any further, it should be explained why averages are strictly required to accurately determine the coherence. Both the one-sided and

two-sided auto and cross-spectral density functions are expressed in terms of the expected value operator, indicating averaging operations should be performed as prescribed in the previous section. Should the expected value not be used in determining the ordinary coherence (or other spectral density measures), obvious errors will result. Equation 4.35 illustrates the ordinary coherence for a single data record not averaged over time.

$$\gamma_{xy}^2(f) = \frac{X^*(f,T)Y(f,T)Y^*(f,T)X(f,T)}{X^*(f,T)X(f,T)Y^*(f,T)Y(f,T)} = 1 \qquad \textbf{(4.35)}$$

Therefore, for any initial measurement procedure done without averaging, the coherence at all frequencies will be one. The number of averages required for accurate measurements of ergodic data is typically sixty as mentioned earlier.

At this point one might be inclined to ask: "Why shouldn't the coherence always equal one for all systems if I'm performing the measurements correctly for a given input and output?" As seen in Equation 4.34, any input/output relationship for a linear noiseless system *will* in fact result in a coherence of one for all frequencies (provided all are being excited and measured). The only conditions under which the coherence value (at any frequency) will be less than one, are those which violate the aforementioned restrictions placed on Equation 4.34. Ignoring FFT resolution errors (as they will be considered minor), all possible errors can be classified into one of these two categories:

1. System in question is non-linear, or time variant.
2. Measurement (or other) noise/signal present at either the input or output.

Whether or not a system is linear time invariant is a determination which must be made through an understanding of the physical system under investigation. This is strictly dependent on the application to which these measurements are to be made. For acoustic (and vibration) systems, the LTI assumption is typically valid during the course of taking measurements. Should the room environment be rearranged between separate measurements, a time variant system might be created. Since all averages are taken under a

single data set following the ergodicity assumption, this phenomenon will not affect any simulated or experimental results regarding the acoustic environments discussed herein.

Therefore, a decreased coherence value for a LTI system can only be attributed to extraneous noise or other signals present in either the input or output. As will be described in Section 4.3.2, extraneous signals contributing to the output can behave as noise when using the ordinary coherence function for evaluation. However, should the engineer have the measuring capabilities to observe the suspected corrupting signal, the partial coherence measurement procedures can account for these errors. This will begin the evaluation necessary to completely describe the MISO problem which is required for designing the COP filters. For the case where measurement (or other such as environmental) noise affects the input and/or output, the following example is presented.

Figure 4.4 illustrates a modification of the ideal system previously described in Figure 4.3. The system $H(f)$ still represents a LTI system with input and output time signals $u(t)$ and $v(t)$, respectively. The main difference here is that each of the measured values of the input and output ($x(t)$ and $y(t)$) don't necessarily represent the actual input and output due to the measurement noises $m(t)$ and $n(t)$. Although these signals are traditionally termed "measurement noise", they can be considered as any type of extraneous noise which cannot be explicitly observed or controlled.

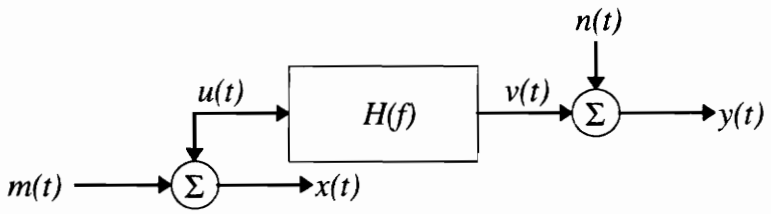


FIGURE 4.4 SISO NOISY SYSTEM

Beginning with the time signals, the *actual* measured quantities can be written in terms of the true input and output and the noise as

$$x(t) = u(t) + m(t)$$
$$y(t) = v(t) + n(t) \qquad \textbf{(4.36)}$$

If the noise terms are uncorrelated with each other as well as the input and output signals, the cross-correlation terms approach zero. Taking appropriate Fourier transforms and expected values the following relationships can be written in terms of each signal's spectral density functions.

$$G_{um} = G_{vn} = G_{mn} = 0$$
$$G_{xx} = G_{uu} + G_{mm}$$
$$G_{yy} = G_{vv} + G_{nn} \qquad \textbf{(4.37)}$$
$$G_{xy} = G_{uv}$$

The final relationship derives from the fact that each of the cross terms created from the $E[X^*Y]$ operation approach zero and can be eliminated. (All the above autospectral relationships remain a function of frequency; the notation has been dropped for convenience).

As explained for the ideal case, the true coherence (that between $u(t)$ and $v(t)$) will be unity as expected by Equation 4.34. However, including the noise terms in what are to be the *actual* measured quantities $x(t)$ and $y(t)$, the following expression can be developed for the coherence between $x(t)$ and $y(t)$.

$$\gamma_{xy}^2 = \frac{|G_{xy}|^2}{G_{xx}G_{yy}} = \frac{|G_{uv}|^2}{[G_{uu} + G_{mm}][G_{vv} + G_{nn}]} \qquad \textbf{(4.38)}$$

This expression describes the ordinary coherence between the measured signals in terms of the actual signals and the noise. Examining this expression, it should be expected that if neither noise term were present, the expression for γ_{xy}^2 would be the same as the expression

for γ_{uv}^2 and both should equal unity for the LTI plant as shown in Equation 4.39 derived from Equation 4.34..

$$\gamma_{xy}^2 = \frac{|G_{xy}|^2}{G_{xx}G_{yy}} = \frac{|G_{uv}|^2}{G_{uu}G_{vv}} = \gamma_{uv}^2 = 1 \qquad \textbf{(4.39)}$$

If however, either or both of the noise terms remain non-zero, they will clearly adversely affect the coherence between *x(t)* and *y(t)* by increasing the size of the denominator without changing the numerator. Therefore any uncorrelated noise measured at either the input or output not passing through the system itself, will decrease the measured ordinary coherence. Even though the actual coherence for the system itself may be acceptable from a control standpoint, the measured values will indicate differently. Since the control process must be done using the *measured* signals rather than the *true* signals, it is of no consequence that the ideal system is behaving acceptably.

An important question to answer before proceeding to the MISO problem is: "how much noise is tolerable before coherence suffers?". To answer this question it is necessary to set a lower bound on the "acceptable" level of coherence. Bendat [8] identifies a level of coherence of 0.8 as being "sufficiently high" for any assumptions to be made about the system integrity. For the purposes of future discussions, a coherence of 0.9 will be considered as acceptable. Justification for this decision can be found by evaluating the coherence function value in terms of a signal to noise ratio.

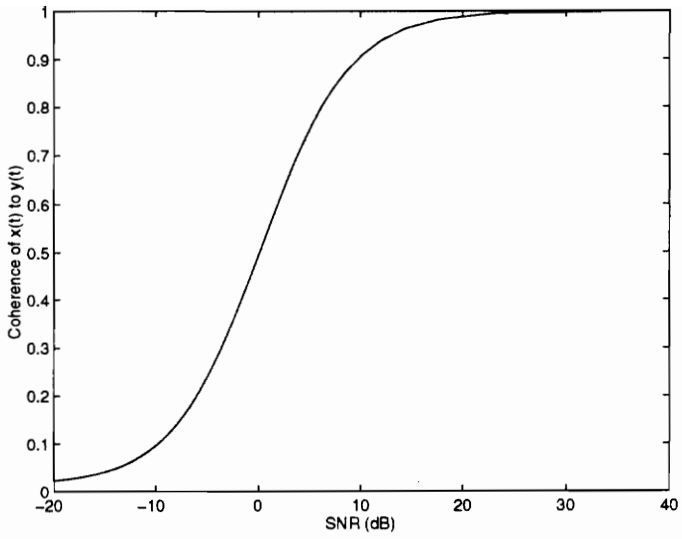


FIGURE 4.5 COHERENCE VS. SIGNAL TO NOISE RATIO

Figure 4.5 illustrates the signal to noise ratio in decibels as defined by Equation 4.40 in terms of the coherence function, where the signal amplitude is one and the standard deviation of the noise is represented by σ.

$$SNR = 10 \times \log_{10}\left(\frac{1}{\sigma^2}\right) \tag{4.40}$$

For Bendat's suggested value of 0.8 the signal to noise ratio is approximately 6 dB. This corresponds to four times in linear power and would typically be acceptable. In terms of control performance the measure is actually 80% of the available effectiveness is being achieved. A coherence of 0.9 corresponds to 90% control effectiveness and 9.5 dB of SNR and will be used for future performance evaluation.

In nearly all practical applications, the calculated coherence will be less than one. This is primarily as a result of measurement (or other) noise. Given this information and the effect noise has on the coherence function, further inclusion of the noise terms in analytical developments will be excluded. It will be assumed that noise will continually adversely

affect the results but aside from careful experimentation, there is little remedy for such problems. As for the problem of a measurable noise (or signal) affecting the output, the ordinary coherence will suffer similarly, but the partial coherence function has the capability of determining the correct linear system contributions.

4.3.2 PARTIAL COHERENCE

The methodology of calculating partial coherence functions is based on the concept of determining what are termed *conditioned spectra.* These are auto or cross-spectra, as previously defined, with the linear effects of other signals conditioned out. Before proceeding with the partial coherence definitions, conditioned spectra are introduced for the two-input single-output system. These spectra are then generalized for the three-input and multi-input single-output systems. Using these definitions, partial coherence functions are presented and defined for the two and multi-input cases.

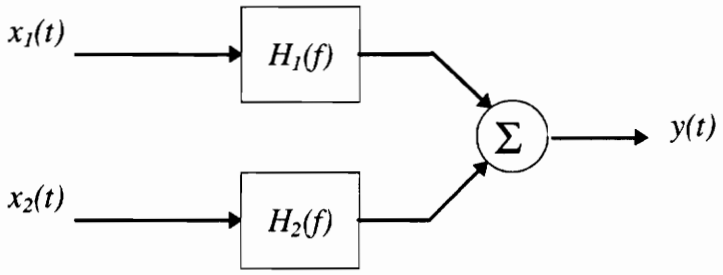


FIGURE 4.6 TWO-INPUT SINGLE-OUTPUT SYSTEM

Figure 4.6 shows a simple two-input single-output system where the two inputs have no correlation with each other. As mentioned in Section 4.3.1, if input $x_1(t)$ were not measured and included in the partial coherence calculations, evidence of is existence would manifest itself in a lower ordinary coherence calculated from $x_2(t)$ to $y(t)$. Performing this calculation, Equation 4.41 demonstrates that the effect of $x_1(t)$ on the ordinary coherence calculation between $x_2(t)$ and $y(t)$.

$$\gamma^2_{x_2 y} = \frac{\left|G_{x_2 v}\right|^2}{G_{x_2 x_2} G_{vv} + G_{x_2 x_2} G_{uu}} \tag{4.41}$$

The temporary signals $u(t)$ and $v(t)$ represent the outputs of H_1 and H_2, respectively. For the LTI system H_2, the ordinary coherence between input $x_2(t)$ and $v(t)$ is equal to unity. By injecting $x_1(t)$ into the total output, the ordinary coherence suffers. If it were possible to remove the effects of $x_1(t)$ (i.e. $u(t)$) from $y(t)$ just to perform the coherence calculation, it would be expected that the coherence would more accurately represent the contribution of $x_2(t)$ to the total output. This is precisely the motivation for the conditioned spectral density function.

As a more generalized case, consider the two input case described above in which the input $x_1(t)$ has a certain amount of correlation with $x_2(t)$ as pictured in Figure 4.7.

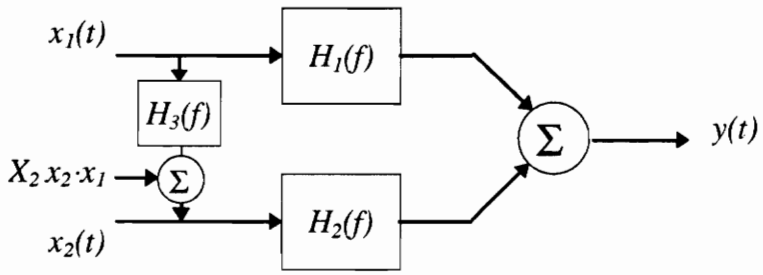


FIGURE 4.7 TWO-INPUT SINGLE-OUTPUT SYSTEM - CORRELATED

Some new notation has been introduced which will be explained within the context of the conditioned spectral density function. The *cross path* from $x_1(t)$ to $x_2(t)$ contains a transfer function H_3 and an additive term which represents the portion of the signal of $x_2(t)$ which is not influenced by $x_1(t)$. Typically the only signals which are measurable are $x_1(t)$ and $x_2(t)$, thus the cross path is inaccessible. Given these measures however, it is possible to extract the portion of $x_2(t)$ not pertaining to $x_1(t)$ via a least squares approximation.

Revisiting Section 4.3.1 developments it should be clear that the coherence between $x_1(t)$ and $x_2(t)$ will not be unity as long as $X_2x_2 \cdot x_1$ is non zero. That is, as long as $x_2(t)$ has an original contribution unaffected by $x_1(t)$, the path between the two appears noisy. The path direction for this implementation is shown from $x_1(t)$ to $x_2(t)$. This is completely arbitrary for these developments and is highly frequency dependent. Section 4.4 investigates path direction as a function of frequency in more detail. Should the path be reversed, the conditioned spectra can be calculated by simply renaming the signals and substituting appropriate subscripts as is now demonstrated.

Given signals $x_1(t)$ and $x_2(t)$ it is possible through a linear least squares prediction to determine the independent nature of each signal without the effects of the other signals[6]. For the system in Figure 4.7 it is desirable to know the frequency dependent contribution $x_2(t)$ has on the output $y(t)$ without $x_1(t)$ affecting the measurement of either signal. Consider examining each signal independently as shown in Figure 4.8.

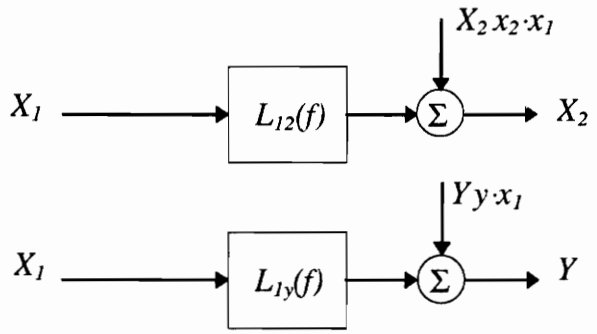


FIGURE 4.8 CONDITIONED SPECTRA

The capital letters indicate Fourier transforms of the data over long records while the dot operator implies that the suffixed signal is conditioned *from* the prefixed signal. The transfer functions $L_{12}(f)$ and $L_{1y}(f)$ are arrived at via optimal methods addressed in [6]. Essentially it is assumed that the "noise" term is zero and the transfer function which then exists for the

for the ideal system becomes the optimal solution. They can be represented in terms of auto and cross-spectral density functions as seen in Equations 4.42.

$$L_{12} = \frac{G_{x_1 x_2}}{G_{x_1 x_1}}$$
$$L_{1y} = \frac{G_{x_1 y}}{G_{x_1 x_1}} \qquad \textbf{(4.42)}$$

Simple block diagram mathematics allows a solution for the conditioned records in terms of the Fourier transforms and the optimal filters in Equations 4.42.

$$X_{2\cdot 1} = X_2 - L_{12} X_1 = X_2 - \left(\frac{G_{12}}{G_{11}}\right) X_1$$
$$Y_{y\cdot 1} = Y - L_{1y} X_1 = Y - \left(\frac{G_{1y}}{G_{11}}\right) X_1 \qquad \textbf{(4.43)}$$

Some simplifying notation has been introduced in Equations 4.43. Rather than denoting $x_1(t)$ as such, the 1 subscript has been used to replace the explicit signal notation. Since Equations 4.43 represent single records, it is necessary to average the information to achieve accurate accounts when using these conditioned spectra in partial coherence function calculations. By inspection from Section 4.2.3, the conditioned auto and cross-spectra can be written in standard notation in terms of the conditioned records (Equations 4.43) for finite length T Fourier transforms as shown in Equations 4.44.

$$G_{22\cdot1} = \frac{2E[X^*_{2\cdot1} X_{2\cdot1}]}{T}$$
$$G_{yy\cdot1} = \frac{2E[Y^*_{y\cdot1} Y_{y\cdot1}]}{T} \qquad \textbf{(4.44)}$$
$$G_{2y\cdot1} = \frac{2E[X^*_{2\cdot1} Y_{y\cdot1}]}{T}$$

As mentioned earlier, the signals can be renamed in the block diagram as well as in the notation to provide definitions for all conditioned spectra in terms of the Fourier transforms of their records as shown in Equations 4.44. Substituting Equations 4.43 into Equations 4.44, several expressions can be developed in terms of the normal auto and cross-spectral density functions and the ordinary coherence function

$$G_{yy\cdot1} = G_{yy} - \frac{|G_{1y}|^2}{G_{11}} = G_{yy}(1-\gamma^2_{1y})$$
$$G_{2y\cdot1} = G_{2y} - \left(\frac{G_{1y}}{G_{11}}\right)G_{21} \qquad \textbf{(4.45)}$$

The expression for the conditioned cross-spectrum of $x_1(t)$ to $y(t)$ without the linear effects of $x_2(t)$ can be written by rearranging the subscripts of the second Equation 4.45. Likewise, the conditioned autospectrum of $x_1(t)$ or $x_2(t)$ without the effects of the other or $y(t)$ can be written by substituting appropriate signals in the first Equation 4.45. They can also be directly determined via Equations 4.43 and 4.44.

Having an accurate measurement of each individual signal in Figure 4.7 it is now possible to introduce the definition of *partial coherence functions* for a two-input single-output system. The partial coherence function can be thought of in the traditional sense of the ordinary coherence function, only eliminating the unwanted linear effects of other measured signals. Revisiting the two-input problem of Figure 4.6, the $x_1(t)$ was adversely affecting the output signal $y(t)$ and thereby reducing the coherence. If $x_1(t)$ were measured, its linear effects (via $H_1(f)$) on $y(t)$ could be conditioned out. A more generalized problem is shown in Figure 4.7

where $x_1(t)$ is affecting both $x_2(t)$ and $y(t)$. Given the partial coherence function, each of the linear paths which may exist in either situation can be realized. The partial coherence is defined in Equation 4.46 as the ordinary coherence function is, between the conditioned spectra developed in Equations 4.45.

$$\gamma^2_{2y\cdot1} = \frac{\left|G_{2y\cdot1}\right|^2}{G_{22\cdot1}G_{yy\cdot1}} \qquad \textbf{(4.46)}$$

Thus the partial coherence between $x_2(t)$ and $y(t)$ represents the linear system which exists (i.e. $H_2(f)$) when $x_1(t)$ is removed from both signals. By definition this should be somewhat intuitive based on the conditioned spectra and the definition of the ordinary coherence function. It should be noted that the partial coherence function is also a function of frequency and varies between zero and unity for the same reasons as the ordinary. To generalize Equation 4.46, the partial coherence can be calculated between any two signals without the effects of another by rearranging the subscript notations and calculating the conditioned spectra accordingly.

The correlated input problem presented in Figure 4.8 presents a generalized two input problem where the path direction is defined. Should a cross path be suspected in practice and included in the partial fraction calculation, the direction *is* important. If attempts are made to remove the affects of $x_2(t)$ from $x_1(t)$ errors will result since $x_1(t)$ affects $x_2(t)$ not vice versa. This direction is also frequency dependent and will be addressed in somewhat more detail in Section 4.4 and Chapter 5.

As a special case of the two-input problem, Figure 4.7 presents an uncorrelated two-input single-output system. The partial coherence function is necessary in order to independently determine linearity between each input and the output due to the presence of the other input at the output. Equation 4.47 shows the result of solving for the partial coherence in strict terms of the ordinary spectra and coherence functions.

$$\gamma^2_{2y\cdot 1} = \frac{\left|G_{2y}G_{11} - G_{1y}G_{21}\right|^2}{G^2_{11}G_{22}G_{yy}(1-\gamma^2_{12})(1-\gamma^2_{1y})} \qquad \textbf{(4.47)}$$

For the case under investigation where the two inputs are mutually uncorrelated, the cross-spectral estimate G_{21} and the ordinary coherence between $x_1(t)$ and $x_2(t)$ will be equal to zero. In this case Equation 4.47 reduces to Equation 4.48.

$$\gamma^2_{2y\cdot 1} = \frac{\gamma^2_{2y}}{(1-\gamma^2_{1y})} \qquad \textbf{(4.48)}$$

Examining this relation it can be seen that if the ordinary coherence between $x_1(t)$ and $y(t)$ is equal to one, the relationship is meaningless as the partial coherence approaches infinity. This supports the assumption that erroneous results will arise if attempts are made to create conditioned spectra which are unnecessary. A class of rules will be presented to describe a well defined model where these relations can be used with confidence. Since several of these criteria are based on the *multiple coherence* function, that will be discussed first in Section 4.3.3. Equation 4.47 and the relationships between the inputs and the outputs can also be well defined via the multiple coherence function. Before moving to multiple coherences however, the partial coherence function should be generalized for the multiple-input single-output case.

The initial assumption to develop the MISO relationships will be that the cross paths between the inputs are all hierarchical as shown in Figure 4.9. That is, $x_1(t)$ is unaffected by any of the subsequent inputs whereas $x_N(t)$ is affected by all the prior inputs.

The paths between the inputs aren't direct correlations. For simplicity they were drawn as lines but actually represent only a partially correlated relationship. For the two-input case it is clear that $x_1(t)$ would be conditioned from $x_2(t)$ and the output to represent the linear contribution of $x_2(t)$ to $y(t)$.

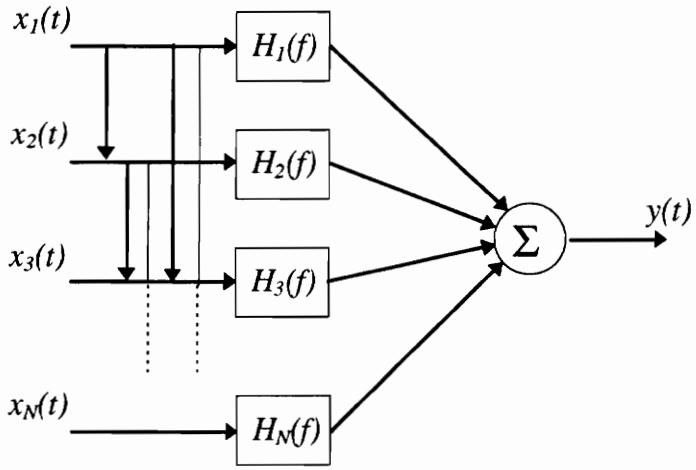


FIGURE 4.9 MULTIPLE-INPUT SINGLE-OUTPUT CORRELATED SYSTEM

For the three-input case, examining the linear path from $x_3(t)$ to $y(t)$ will require the removal of both linear effects of $x_1(t)$ and $x_2(t)$ from $x_3(t)$ and $y(t)$. Before this partial coherence calculation can be made, an extension of the conditioned spectra to the three input case must be defined. Equation 4.49 shows the conditioned cross and autospectra following the "trend" set by Equation 4.45 for the two input case.

$$G_{yy\cdot 12} = G_{yy\cdot 1} - L_{2y} G_{y2\cdot 1} = G_{yy\cdot 1} - \left(\frac{G_{2y\cdot 1}}{G_{22\cdot 1}} \right) G_{y2\cdot 1} \qquad \textbf{(4.49)}$$

The conditioned cross-spectra can also be calculated by the same methodology by renaming the signals subscripts. Given that this is the case, a general relation for the conditioned cross-spectral density function can be developed where the conditioned autospectral density function is a special case in that i = j.

$$G_{ij\cdot r!} = G_{ij\cdot(r-1)!} - L_{rj}G_{ir\cdot(r-1)!}$$
$$L_{rj} = \frac{G_{rj\cdot(r-1)!}}{G_{rr\cdot(r-1)!}} \qquad \textbf{(4.50)}$$

The factorial operation indicates that all previous inputs should be conditioned from the one being calculated. Since each calculation depends on the previous record being also conditioned, the number of computations to achieve these results increases factorially when adding inputs.

The partial coherence function for the case of three inputs is defined by the ordinary coherence function relationship of the conditioned records of one of the inputs to the output without the linear effects of the other inputs. Equation 4.51 demonstrates this relation.

$$\gamma^2_{3y\cdot 12} = \frac{\left|G_{3y\cdot 12}\right|^2}{G_{33\cdot 12}G_{yy\cdot 12}} \qquad \textbf{(4.51)}$$

As with each of the other partial and ordinary coherence functions, the inputs can be reordered to calculate the conditioned coherence between any of the signals. For the generalized system this relationship can be expressed as a factorial, removing all the effects of the other signals from the current as seen in Equation 4.52.

$$\gamma^2_{iy\cdot(i-1)!} = \frac{\left|G_{iy\cdot(i-1)!}\right|^2}{G_{ii\cdot(i-1)!}G_{yy\cdot(i-1)!}} \qquad \textbf{(4.52)}$$

The hierarchical arrangement used to developed the partial coherence function relationship should not be de-emphasized. The multiple coherence makes use of this feature to create a single metric to describe how all of the inputs being measured relate to the output.

4.3.3 MULTIPLE COHERENCE

The *multiple coherence function* is a technique which is used to determine the contribution of all of the combined measured inputs to the measured output while taking into account any correlation existing between the measured inputs. For example, suppose an (output) signal $y(t)$ were being measured with respect to two input signals $x_1(t)$ and $x_2(t)$ as in the previous examples. The difference in this case will then be that $y(t)$ contains signal content from three signals, (i.e. an $x_3(t)$ signal is also contributing to $y(t)$ without being independently measured). For a noiseless system, the multiple coherence will predict a value less than unity because all of the output is not represented in the measured signals. (This is a multiple-input single-output extension of the noise problem described in Section 4.3.1). If however, the third input is subsequently measured and used in the multiple coherence calculation, it will predict a value of unity for all excited frequencies.

The mathematical derivation of the multiple coherence is extensive and highly dependent on proper notation. Bendat in [8] details this derivation quite effectively. The results, notation, and practical implementations of the multiple coherence are presented here. For the generalized multiple-input problem of Figure 4.9, the ideal system relationships can be summed to give an expression for G_{yy} equivalent to Equation 4.53 for a noiseless system.

$$G_{yy} = \sum_{i=1}^{q} H_i^* G_{iy} \tag{4.53}$$

The multiple coherence can be shown for a multiple-input single-output system as shown in Figure 4.9 to be

$$\gamma_{y:x}^2 = \frac{G_{yy} - G_{nn}}{G_{yy}} \tag{4.54}$$

The colon notation is read as the coherent contribution of all the inputs $x(t)$ to the total output $y(t)$. For the case where all inputs are measured and G_{nn} is zero, the multiple

coherence will be unity. In general the multiple coherence for a set of all N inputs is written in terms of partial and multiple coherences as shown in Equation 4.55.

$$\gamma^2_{y:x} = 1 - \prod_{i=1}^{N} (1 - \gamma^2_{iy\cdot(i-1)!}) \qquad \textbf{(4.55)}$$

Applying Equation 4.55 to the special case of two inputs where neither input is correlated, the resulting relation is shown in Equation 4.56.

$$\gamma^2_{y:x} = \gamma^2_{1y} + \gamma^2_{2y} \qquad \textbf{(4.56)}$$

Realizing the validity of this relationship in the context of Equation 4.48 affords a new understanding of the relationship of all these measurements. If the two uncorrelated inputs contribute equally to the total power, the ordinary coherences will each be 0.5, as the opposing input acts like noise on *y(t)*. When viewed as the multiple coherence in Equation 4.56, it is evident that all of the contribution toward *y(t)* is being measured because the multiple coherence is unity. Removing the effects of either signal from the other by calculating the partial coherence as in Equation 4.48, it is seen that a linear system does in fact exist because the partial coherence is also unity. While the multiple coherence of Equation 4.55 for correlated inputs is slightly more complicated, it's relationship to the general expression for partial coherences in Equation 4.52 is equally apparent.

In an attempt to tie each of these measures together, four scenarios/conditions can be described which must be satisfied to achieve a well defined system to which these measures will be accurate.

1. If any of the ordinary coherence functions *between* individual inputs are equal to unity, one of the two inputs should be eliminated. Clearly if this were the case, a linear system exists between the two inputs (at that frequency) and redundant information is being evaluated with respect to the output.

2. If any of the ordinary coherence functions between any input and the output (at any frequency) are equal to one, the other inputs are not contributing to the output and should be eliminated (at that frequency). In practice this may be difficult to determine due to the existence of cross paths as defined in the first condition. Cross paths should then be identified before applying this condition.

3. If the multiple coherence between any input and the set of other inputs being measured, is unity, the original input can be obtained by a linear combination of the other inputs and should thus be eliminated. This indicates that multiple cross paths exist between inputs and another input; therefore redundant information concerning the output is present.

4. If the multiple coherence between all measured inputs and the output is too low, many of the subsequent assumptions about the physical environment will be invalid. In theory, if all inputs are being measured, the multiple coherence for a noiseless system will be unity regardless of input correlation. Should the multiple coherence be less than unity for a noiseless system, one or more of the contributing inputs is not being measured. The acceptable value of multiple coherence depends on statistical models and environmental conditions but will be estimated as 0.5 in accordance with [6, 8].

These scenarios should provide a more practical understanding of the application of the multiple coherence function. Essentially it determines whether or not all the measured inputs can effectively represent the output for a noiseless system. To examine the physical contribution each input has to the total output power, it is necessary to look at each individual signal in the context of the "coherent power" it is contributing to the output.

4.3.4 COHERENT OUTPUT POWER

The measure of coherent output power identifies the fractional contribution of each individual input to the total output power using ordinary and partial coherences much like the multiple coherence.

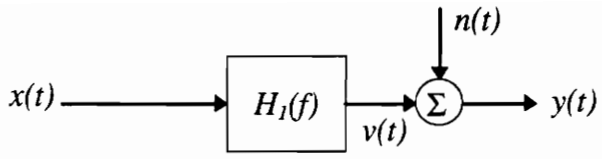


FIGURE 4.10 SISO SYSTEM

To introduce this idea consider the SISO noisy system in Figure 4.10. Using the ideal system relations previously developed as well as simple block diagram algebra, the following expressions can be written relating the various spectra.

$$\begin{aligned} G_{xy} &= H(f)G_{xx} \\ G_{vv} &= |H(f)|^2 G_{xx} \\ G_{vv} &= \left|\frac{G_{xy}}{G_{xx}}\right|^2 G_{xx} = \gamma_{xy}^2 G_{yy} \end{aligned} \qquad \textbf{(4.57)}$$

The product of the ordinary coherence with the output autospectrum is termed the *ordinary coherent output power spectrum.* This represents the portion of the output spectrum G_{yy} due solely to the input *x(t).* This particular derivation is somewhat trivial without the noise term as the entire output would be due solely to the input and the coherence would be unity. Taking the noise into account (that is $G_{yy} = G_{vv} + G_{nn}$), Equation 4.58 shows the ordinary coherence from *x(t)* to *y(t)* represents the fractional portion of the output power due to the signal *v(t).*

$$\gamma_{xy}^2 = \frac{G_{vv}}{G_{yy}} \qquad \textbf{(4.58)}$$

These power ideas can be extended to the case of multiple inputs using the multiple coherence function in place of the ordinary coherence function. For a noiseless system the total coherent output spectrum can be defined using colon notation as

$$G_{y:x} = G_{y:N!} = \gamma^2_{y:q!} G_{yy} \tag{4.59}$$

For the case where all the contributing inputs are being measured, the multiple coherence will be unity and Equation 4.59 is somewhat trivial, stating that the total power of all the measured inputs equals the total power of the output as expressed in Equation 4.53.

The usefulness of this relationship arises when one or more of the inputs contributing to the overall output is not taken into account. In addition, the necessity of ordering the inputs can be derived using this relationship as well as the independent power contribution of each signal. This is a key element in developing the intuitive structure of the COP filters and will be explained in detail.

Beginning with the two input case, Equation 4.59 can be rewritten for the system shown in Figure 4.7 for correlated input signals.

$$G_{y:12} = \gamma^2_{y:12} G_{yy} \tag{4.60}$$

The notation used in the following derivation redefines $x_1(t)$ as 1 and $x_2(t)$ as 2. Using definitions previously developed for ideal system relations and the multiple coherence function, Equations 4.61 derive the contributing power from each input.

$$
\begin{aligned}
G_{y:12} &= \gamma^2_{y:12} G_{yy} \\
&= G_{yy}[1-(1-\gamma^2_{1y})(1-\gamma^2_{2y\cdot1})] \\
&= G_{yy} - G_{yy} + G_{yy}\gamma^2_{2y\cdot1} + G_{yy}\gamma^2_{1y} - G_{yy}\gamma^2_{1y}\gamma^2_{2y\cdot1} \\
&= G_{yy}\gamma^2_{2y\cdot1}(1-\gamma^2_{1y}) + G_{yy}\gamma^2_{1y} \\
&= \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot1} G_{yy\cdot1}
\end{aligned}
\qquad \textbf{(4.61)}
$$

As in Equation 4.57 the first term of the last relationship in Equations 4.61 represents the ordinary coherent output power of the first input signal. The second term represents the partial coherent output power of the second input without the effects of the first input. As mentioned earlier, no physical meaning results from extracting the linear contribution of the second input from the first input if the cross path is as in Figure 4.7. If the cross path is determined to be *from* the second input *to* the first input, the Equations 4.61 would merely exchange subscripts.

The importance of determining cross path direction should now be very apparent. Suppose the assumption of direction were incorrect and Equation 4.61 were used regardless, each of the two terms would be incorrect. The term representing the ordinary coherent output power from the first input would be corrupted by the cross term originating from the second input. The partial coherent output power calculation would be attempting to subtract the linear effects of $x_1(t)$ from $x_2(t)$ when there are in fact, none.

Generalizing these developments to the multi-input case is possible via recognizing trends in the evaluation of the three-input case. Beginning with the multiple coherent output power for the three input case, a similar relationship for the individual signal contributions can be derived in the same manner as shown in Equation 4.62.

$$
\begin{aligned}
G_{y:123} &= \gamma^2_{y:123} G_{yy} \\
&= G_{yy}[1-(1-\gamma^2_{1y})(1-\gamma^2_{2y\cdot 1})(1-\gamma^2_{3y\cdot 12})] \\
&= \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot 1} G_{yy\cdot 1} + \gamma^2_{3y\cdot 12} G_{yy}(1-\gamma^2_{1y})(1-\gamma^2_{2y\cdot 1}) \\
&= \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot 1} G_{yy\cdot 1} + \gamma^2_{3y\cdot 12} G_{yy\cdot 12}
\end{aligned}
\tag{4.62}
$$

For ordered inputs as shown in Figure 4.9, Equation 4.62 represents the individual contribution each signal independently makes after taking into account the effects of the other inputs. As can be surmised, ordering of the inputs for the multi-input case is just as crucial for the determination of the appropriate power contribution of each input signal to the output.

Bendat and Piersol in [3] propose a methodology which initially ranks the inputs based on the level of the *ordinary* coherence of each individual input to the output. (This is of course a frequency dependent ordering where the cross paths could have different directions at different frequencies). This method of ordering inputs is more of an intuitive approach rather than a strict mathematical treatment. The supposition is that if a linear path (highly coherent) exists directly from one input to the output it shouldn't be originating from an input cross path. This approach can clearly produce incorrect results since no other information is used regarding the existence of correlation. A more reliable approach for ordering inputs is explained in [9]. Park and Kim propose that checking the causality between inputs over narrow frequency ranges provides accurate cross path direction determination. While this method is more reliable and intuitive, it is far to computationally complex to implement in this study. Suffice it to say that the cross path direction *can* be effectively determined and the ordinary coherence ordering methodology will be employed in subsequent developments.

To fully generalize the coherent output power in terms of the multiple input problem, the general expression for the multiple coherence given in Equation 4.55 is used.

$$
\begin{aligned}
G_{y:x} &= \gamma^2_{y:N!} G_{yy} \\
&= \left(1 - \prod_{i=1}^{N} (1 - \gamma^2_{iy \cdot (i-1)!}) \right) G_{yy} \\
&= \gamma^2_{1y} G_{yy} + \gamma^2_{2y \cdot 1} G_{yy \cdot 1} + \cdots + \gamma^2_{Ny \cdot (N-1)!} G_{yy \cdot (q-1)!}
\end{aligned}
\tag{4.63}
$$

Developments made prior to this point are all somewhat theoretical without direct application to a physically meaningful system. Section 4.4 uses results of the prior discussion to apply to what's termed the *energy source identification problem.* This class of problem is common in the ANC community as well as the engineering community in general. The COP filter implementation developed in Chapter 5 can be applied to any problem which can be structured, conformed, or redefined in terms of this class of problem.

4.4 THE ENERGY SOURCE IDENTIFICATION PROBLEM

Although the examples which will be described are related directly to active noise control, the nature of the energy source identification problem lends itself to a much wider range of applications. First, the structure of the problem and notation will be established without referring to specific practical applications. Then, by casting two theoretical noise control problems in the same form, a physical understanding of the signals can be established. Various levels of complexity of the two-input and multi-input configurations will be completely defined in terms of the linear system measures mentioned throughout this chapter.

4.4.1 GENERAL CONFIGURATION

There are typically two process which can be measured for a given system, the cause and the effect. In most applications however, there are many possible/probable causes resulting in a single "effect". This type of system is systematically defined by the energy source identification problem. The system level diagram of this type of problem is shown in Figure 4.11.

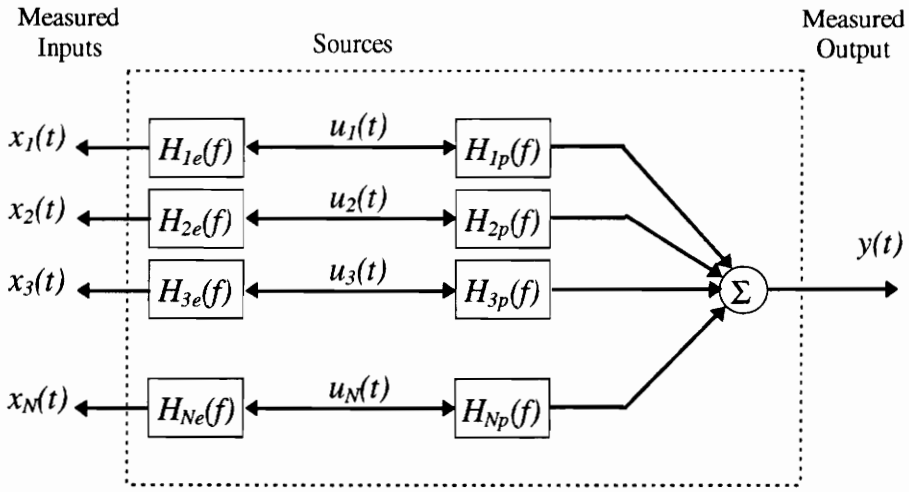


FIGURE 4.11 MISO SYSTEM AS THE ENERGY SOURCE ID PROBLEM

The dotted line represents the aspects of the system which are inaccessible to the observer. On the left side, the engineer has the capability to measure some version of the originating sources through some type of transducer transfer function denoted with the *e* subscripts. On the right side, each of the individual sources being measured are suspected to contribute to the overall measured output through some "plant" frequency response functions denoted with a *p* subscript. Although the originating sources are drawn as being uncorrelated for brevity, the paths between the sources *can* be (and usually are) related as discussed in the previous section.

Two candidate noise control problems were suggested during the introductory remarks in Chapter 1 which will now be cast in terms of Figure 4.11. The problem of automobile noise control is constantly under investigation by many car manufacturers in an effort to claim the "quietest ride". Suppose active noise control was under consideration as a methodology for reducing interior noise. Through extensive signal processing and coherence testing, the fan noise, the engine noise, and the tire noise were identified as the primary contributors to the overall interior noise. These sources would be somewhat intangible and would be identified as the signals $u_1(t)$, $u_2(t)$, and $u_3(t)$ in Figure 4.11. Measurements of these three sources can be realized either through tachometers, accelerometer, microphones or a host of other transducers, each with a transfer function subscripted *e*. The path taken by the identified source to the operator's ears is identified as the plant and includes any acoustic or mechanical frequency response occurring prior to the noise entering the cabin. As for the correlation between the inputs, it should be obvious that since the fan is powered by the engine, and the tire noise is a function of the car speed (also a function of the engine speed), there may be a high degree of correlation between the original sources, and thus the measured inputs.

As another example of a noise control situation that can be cast as an energy source identification problem, consider a piece of office equipment such as a printer. The measured noise $y(t)$ is generated from a number of possible sources within the mechanisms of the printer itself. The movement of the paper as it is fed through the printer and the stepper motor which turns the cylinder that advances the paper can each contribute to the overall noise. While it may be difficult to accurately measure the sound generated by the movement of the paper, it can be assured that it will be correlated to some extent with the stepper motor rotating the cylinder.

Most noise control problems can be cast as an energy source identification problem by definition. Conceivably many vibration control problems also conform to the general structure presented above. To fully describe mathematically, the general case for multiple correlated sources, an introduction to the special case of two inputs will first be explained.

4.4.2 SPECIAL CASE: TWO INPUTS

Figure 4.12 illustrates the energy source identification problem encountered when two correlated sources are contributing to the measured output.

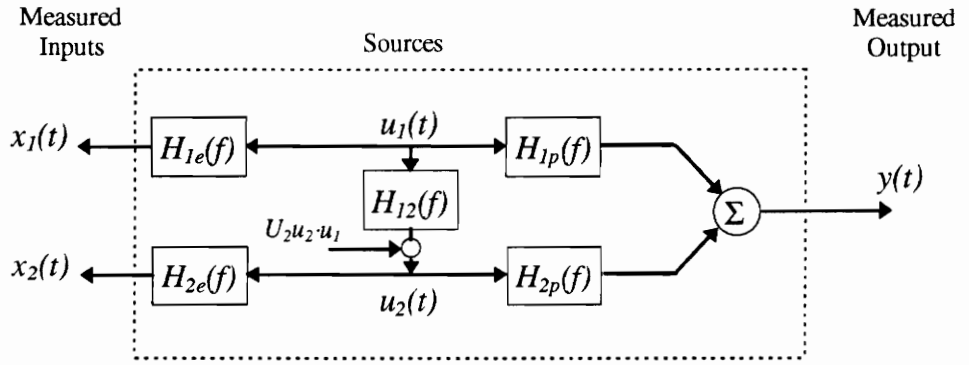


FIGURE 4.12 TWO-INPUT SINGLE-OUTPUT ID PROBLEM

The configuration shown in Figure 4.12 illustrates the fully generalized two input problem. The only remaining generality would be to switch the direction of the cross path from $u_2(t)$ to $u_1(t)$. Since this is a frequency dependent direction, it will be assumed that the subsequent analyses rely on a "downward" cross path. The conclusions drawn can easily be reformulated to demonstrate otherwise. The general relationship for the multiple coherent output power will be developed for the general case and several configurations will be examined. Before proceeding some relationships between the signals' spectra will be developed.

Examining the cross path, the term defined as the portion of $u_2(t)$ not pertaining to $u_1(t)$ will be renamed using an n for brevity and to emphasize that it behaves as a noise term. The following relationships between $u_1(t)$ and $u_2(t)$ apply.

$$G_{u_2u_2} = \left|H_{12}\right|^2 G_{u_1u_1} + G_{nn}$$
$$G_{u_1u_2} = H_{12} G_{u_1u_1} \tag{4.64}$$

The extra "noise" term prevents the cross path from conforming to the ideal system relationships and the ordinary coherence will thus be less than unity as witnessed in Section 4.3.1. Rewriting the ideal system relationships for the remainder of the measured paths it can be shown that the ordinary coherence between the two inputs $x_1(t)$ and $x_2(t)$ is equivalent to the ordinary coherence between the originating sources $u_1(t)$ and $u_2(t)$. Given this relationship, the following can be determined regarding the cross-spectra.

$$\gamma^2_{x_1x_2} = \gamma^2_{x_1u_2}$$
$$\frac{\left|G_{x_1x_2}\right|^2}{\left|H_{1e}\right|^2 G_{u_1u_1} \left|H_{2e}\right|^2 G_{u_2u_2}} = \frac{\left|G_{u_1u_2}\right|^2}{G_{u_1u_1} G_{u_2u_2}} \tag{4.65}$$
$$G_{x_1x_2} = H_{1e} H_{2e} G_{u_1u_2}$$

This relationship will be used in deriving the multiple coherent output power in terms of the originating sources. In order to verify the results of the derivation, the output power itself should be determined in terms of the originating sources. Examining the "plant" side of Figure 4.12, the following equation can be developed for the total output power in terms of the originating sources as well as the measured inputs.

$$
\begin{aligned}
Y(f) &= H_{1p}U_1(f) + H_{2p}U_2(f) \\
Y^*Y &= H_{1p}U_1H_{1p}^*U_1^* + H_{1p}U_1H_{2p}^*U_2^* + H_{2p}U_2H_{1p}^*U_1^* + H_{2p}U_2H_{2p}^*U_2^* \\
G_{yy} &= \left|H_{1p}\right|^2 G_{u_1u_1} + H_{2p}^*H_{1p}G_{u_2u_1} + H_{1p}^*H_{2p}G_{u_1u_2} + \left|H_{2p}\right|^2 G_{u_2u_2} \\
&= \frac{\left|H_{1p}\right|^2}{\left|H_{1e}\right|^2}G_{x_1x_1} + \frac{H_{2p}^*H_{1p}}{H_{2e}H_{1e}}G_{x_2x_1} + \frac{H_{1p}^*H_{2p}}{H_{1e}H_{2e}}G_{x_1x_2} + \frac{\left|H_{2p}\right|^2}{\left|H_{2e}\right|^2}G_{x_2x_2}
\end{aligned}
\tag{4.66}
$$

It is educational to demonstrate the effects of making an erroneous assumption concerning the existence of the cross path in Figure 4.12. If it is assumed that a cross path does not exist, the multiple coherent output power would be equivalent to the sum of the ordinary coherent output powers of the two inputs. This can be seen by examining the general form of Equation 4.63 and realizing that the partial coherence and conditioned spectra would not be necessary. Making the assumption of no cross path, the multiple coherent output power is developed in Equations 4.67.

$$
\begin{aligned}
G_{y:x} &= G_{y:12} = \gamma_{1y}^2 G_{yy} + \gamma_{2y}^2 G_{yy} \\
G_{y:12} &= \frac{\left|\frac{H_{1p}}{H_{1e}}G_{x_1x_1} + \frac{H_{1e}^*H_{2p}}{H_{1e}H_{2e}}G_{x_1x_2}\right|^2}{G_{x_1x_1}} + \frac{\left|\frac{H_{2p}}{H_{2e}}G_{x_2x_2} + \frac{H_{2e}^*H_{1p}}{H_{1e}H_{2e}}G_{x_2x_1}\right|^2}{G_{x_2x_2}} \\
G_{y:12} &= \frac{\left|H_{1p}\right|^2}{\left|H_{1e}\right|^2}G_{x_1x_1} + 2\frac{H_{1p}^*H_{2p}}{H_{1e}H_{2e}}G_{x_1x_2} + \frac{\left|H_{2p}\right|^2}{\left|H_{2e}\right|^2}\frac{\left|G_{x_1x_2}\right|^2}{G_{x_1x_1}} \\
&\quad + \frac{\left|H_{2p}\right|^2}{\left|H_{2e}\right|^2}G_{x_2x_2} + 2\frac{H_{2p}^*H_{1p}}{H_{2e}H_{1e}}G_{x_2x_1} + \frac{\left|H_{1p}\right|^2}{\left|H_{1e}\right|^2}\frac{\left|G_{x_2x_1}\right|^2}{G_{x_2x_2}} \\
&\neq G_{yy}
\end{aligned}
\tag{4.67}
$$

Comparing this result with Equation 4.66 (the actual value of the output spectral density) it is seen that the ordinary coherent output powers will always erroneously overestimate the true output power as long as a cross path exists. For the special case where no correlation exists between the two originating sources, all the cross terms in Equations 4.66 and 4.67

will become zero and thus the sum of the ordinary coherent output powers *will* equal the true output power.

As discussed for the multiple coherent output power, the conditioned spectra and partial coherences must be used to accurately predict the individual contributions of measured inputs as long as cross paths exist. Assuming the direction shown in Figure 4.12 is accurately known for a given frequency, the multiple coherence output power as developed in Equation 4.61 will predict the correct value of the total output power (at that frequency).

$$
\begin{aligned}
G_{y:x} &= G_{y:12} = \gamma_{1y}^2 G_{yy} + \gamma_{2y\cdot 1}^2 G_{yy\cdot 1} \\
G_{y:12} &= \frac{\left|G_{1y}\right|^2 G_{22} + \left|G_{2y}\right|^2 G_{11} - G_{2y}^* G_{21} G_{1y} - G_{12} G_{1y} G_{2y}}{G_{22} G_{11} - \left|G_{21}\right|^2} \\
G_{y:12} &= \frac{\left|H_{1p}\right|^2}{\left|H_{1e}\right|^2} G_{x_1 x_1} + \frac{H_{1p}^* H_{2p}}{H_{1e} H_{2e}} G_{x_1 x_2} + \frac{H_{2p}^* H_{1p}}{H_{2e} H_{1e}} G_{x_2 x_1} + \frac{\left|H_{2p}\right|^2}{\left|H_{2e}\right|^2} G_{x_2 x_2} \\
&= G_{yy}
\end{aligned}
\qquad \textbf{(4.68)}
$$

Many tedious algebraic manipulations were removed from this derivation and some notation was mixed to facilitate understanding. As expected, the multiple coherent output power accurately predicted the true output based on the conditioned spectra.

As a matter of terminology, consider the first relationship of Equation 4.68, the multiple coherent output power for a two-input system. The first term is usually denoted as the ordinary coherent output power and the second term is typically termed the partial coherent output power. Since it has been established that the correct order of inputs is necessary to accurately predict the total output power from the multiple coherent output power, the assignment of signal names is somewhat arbitrary. In addition, the (ordinary or partial) contribution of each signal to the multiple coherent output power is highly frequency dependent and will usually "flip-flop" throughout the spectrum for physical systems. Therefore, the terms in the multiple coherent output power equation will simply be termed

the *coherent output powers* or COP's since the delimitation of whether it is ordinary or partial is frequency and system dependent.

To investigate various realizations of the two-input system of Figure 4.12, the general expression for the multiple coherent output power can be used. The case for which no cross path exists has been examined and it has been determined that the sum of the ordinary coherent output powers will accurately represent the total output power. If this assumption is erroneously made, Equation 4.67 results.

For the next realization, suppose no cross path exists, both sources are contributing to the overall output, but only one of the inputs is being measured. The physical counterpart to this situation is easily realizable as a case where all of the contributing inputs are not known. Regarding the automobile example, the engineer hasn't yet determined that the tire noise contributes to the overall interior noise and suspects the engine is the only source. This situation is shown in Figure 4.13.

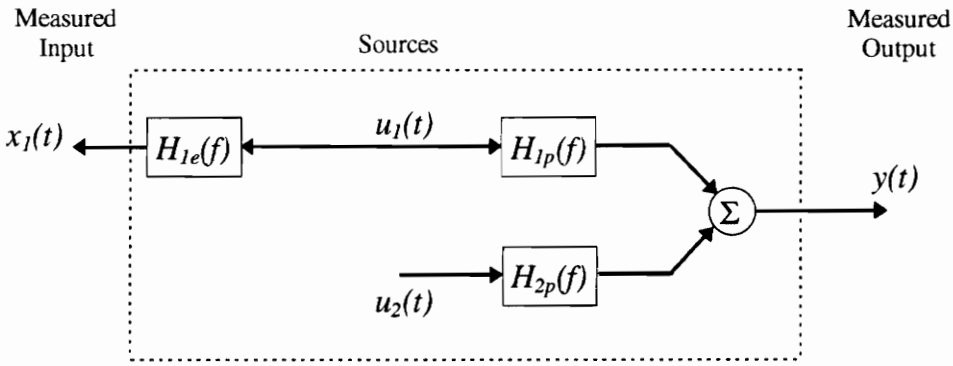


FIGURE 4.13 TWO-INPUT SINGLE-OUTPUT ID PROBLEM: SPECIAL CASE

This under specified case will simply result in a lower value of coherence being measured between the input and the output. With only one input being measured the engineer is forced to assume that the total output power (from Equation 4.68) is the ordinary coherent output power for the first measured input. As in 4.38 and 4.41, $u_2(t)$ will behave as a noise term on the output thereby lowering the coherence. Depending on the contribution of $u_2(t)$

to the measured output, the engineer should recognize that due to the low coherence, a second input should be investigated as directed by the third rule for a well defined system.

As an alternative situation, suppose each of two (candidate) inputs were being measured but only one of the originating sources were actually contributing to the overall measured output. This situation is shown in Figure 4.14.

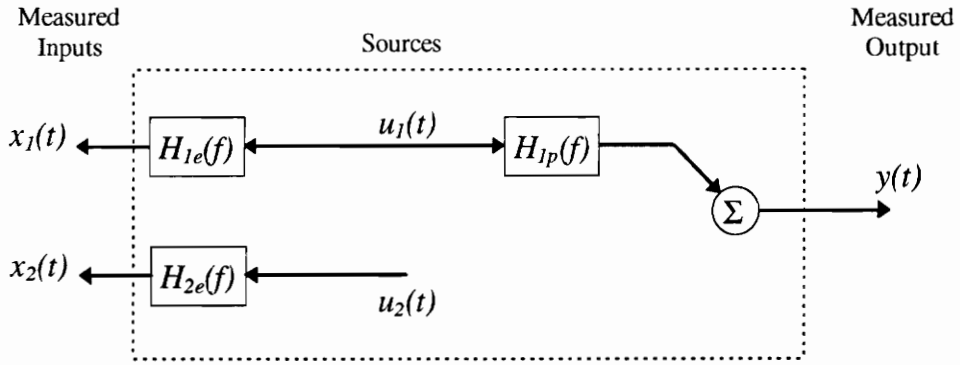


FIGURE 4.14 TWO-INPUT SINGLE-OUTPUT ID PROBLEM: SPECIAL CASE

This case indicates that the engineer has what will be termed a "bad reference". The signal $x_2(t)$ has no linear path to the output and thus no appreciable contribution to the overall noise. Returning to Equation 4.68 (the specialized two input case of Equation 4.63), the multiple coherent output power will attempt to represent the contribution of each input. The (ordinary) coherent output power of the first input will essentially represent all of the total output power and it will be immediately apparent that the (partial) coherent output power of the second input will be calculated to be zero since there is no linear path to the output. This becomes a valuable tool in the development of the COP filters as it can easily identify cases where a poor choice has been made regarding the input.

Many other special cases of the system shown in Figure 4.12 and described by Equation 4.68 can be imagined. It should be evident that for the two-input case, Equation 4.68 describing the multiple coherent output power can be solved for any possible situation with or without cross paths. The resulting solution when compared to the measured output power will allow the engineer to make a determination regarding the inputs based on the

rules set forth which delimit a well defined system. These ideal scenarios are all based on a noiseless system where the cross path directions are known or at least obtainable. To automate this entire process and use this information to benefit active noise control procedures as well as system identification is the goal of the COP filters. Before detailing the COP filtering methodology, the two-input case described above is generalized for the multiple-input case.

4.4.3 General Case: Multiple Inputs

Given Equation 4.63, the general relation for the multiple coherent output power, it is a simple matter to extend the two-input single-output identification problem to the multiple-input problem. Figure 4.11 graphically describes the MISO identification problem with the exception of the correlated inputs. As has been previously described, it is a necessary assumption that the inputs first be ordered based on the direction of the correlated paths between them. Since this is typically a function of frequency, a large number of directional paths can exist.

For the case of three inputs, assuming no measurement noise and ordered inputs, the following expression for the multiple coherent output power is valid.

$$\begin{aligned} G_{y:x} &= \gamma^2_{y:123} G_{yy} \\ &= \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot1} G_{yy\cdot1} + \gamma^2_{3y\cdot12} G_{yy\cdot21} \\ &= G_{yy} \end{aligned} \tag{4.69}$$

This is proven in the same manner as shown in the previous section for two inputs. Extending this to the multiple case via Equation 4.63, Equation 4.70 shows the multiple coherent output power for the energy source identification problem.

$$
\begin{aligned}
G_{y:x} &= \gamma^2_{y:N!} G_{yy} \\
&= \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot 1} G_{yy\cdot 1} + \cdots + \gamma^2_{Ny\cdot(N-1)!} G_{yy\cdot(q-1)!} \\
&= G_{yy}
\end{aligned}
\tag{4.70}
$$

Thus for a set of ordered inputs, the sum of the coherent output powers of each of the inputs will equal the total output power provided the conditions for a well defined system are met and no noise is present in the measurements. Equation 4.70 gives each signal's individual contribution as a function of frequency, to the overall measured output. It also has the capability of identifying inputs which are not useful or inputs which solely contribute 100% of the total output power in a single frequency bin. These are very useful pieces of information when applied to system identification and adaptive control (noise or otherwise).

In noise control engineering specifically, adaptive feedforward control requires a coherent reference to generate the anti-noise as described in Chapter 3. The first step in performing such control is finding the best coherent reference possible. As has been the theme throughout, this almost never occurs in a single signal. Therefore for multiple signal coherent spectrum analysis, the above procedures are the only viable solution for determining accurately, the contribution of various signals to the overall noise. The COP filters developed in the next chapter furnish a means for automating this procedure while providing a physical interpretation regarding the contribution of each signal to the overall output.

Chapter 5

Coherent Output Power Filtering

Each of the tools have now been presented which are required in order to design and implement the COP filtering procedure. The energy source identification problem coupled with adaptive control procedures can be used to generate a noise control methodology which can drastically improve performance in many cases.

This chapter is organized as follows. Since maximizing coherence is the basis of the motivation of the COP filter procedure, its relation to control performance will be revisited in detail. After redefining the problem in terms of noise control, the COP filter design methodology is presented. The physical interpretation will be intuitive while the mathematical description is somewhat more complicated. Following a section on the description of the physical implementation of the COP filters as well as several generalized simulations, the mathematical relationship to the simple summation is explored. While the full solution to the exact relationship is not provided, adequate numerical results present a means for understanding the similarities and differences. Since adaptive arrays are designed to solve the same problem as the COP filtering techniques, a comparison is made regarding the performance of each.

One final clarification regarding terminology is necessary before proceeding. Until now, use of the term "noise" has been interpreted in one of two possible ways: either as undesirable sound pressure level or uncorrelated content on signal measurements. Since the COP filtering procedure employs both fields of study, the differentiation between the two possible meanings should now be apparent based solely on context. If "noise" is used in conjunction with control, it will refer to acoustic SPL.

5.1 GENERAL

5.1.1 COHERENCE AND FEEDFORWARD CONTROL

It is commonly known within the adaptive signal processing community, that in order to achieve acceptable performance from an adaptive feedforward controller, the coherence between the input and the disturbance must be relatively high. This can be mathematically shown by considering a simplified control architecture as shown in Figure 5.1.

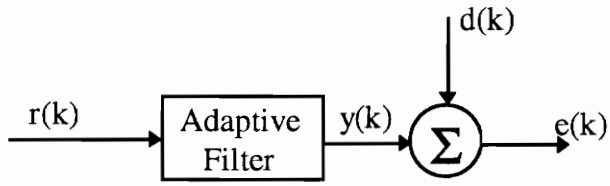


FIGURE 5.1 LEAST SQUARES ESTIMATION PROBLEM

This figure is identical to that which is shown in Figure 3.1 with some notation differences. The optimal filter is now approximated by an adaptive filter using the LMS algorithm. Each of the other signals are exactly the ones used to develop the performance surface and gradient search analysis. Using identical developments as in Section 3.1.1, an expression for the autospectrum of the error signal can be created from simple expected value approximations of the time signals.

$$S_{ee} = S_{dd} + S_{rd}^{*}H + H^{*}S_{rd} + H^{*}S_{rr}H \qquad \textbf{(5.1)}$$

(H represents the adaptive filter and each of the quantities are frequency dependent). This expression is of the same quadratic form as Equation 3.6 for the mean square error. Since it is also desired to minimize the autospectrum of the error, taking the derivative of Equation 5.1 and setting it equal to zero results in the optimal adaptive filter solution of Equation 5.2.

$$H_{opt} = -\frac{S_{rd}}{S_{rr}} \tag{5.2}$$

(The same analysis for the M.S.E. in Chapter 3 resulted in Equation 4.7). Continuing on, Equation 5.2 can be substituted into Equation 5.1 to develop an expression for the minimum value of the autospectrum of the error signal, given that the adaptive filter is able to design itself in the form of the optimal filter of Equation 5.2. Equation 5.3 shows the result of this substitution.

$$S_{ee\,\min} = S_{dd} - \frac{|S_{rd}|^2}{S_{rr}} \tag{5.3}$$

Now, if each side of Equation 5.3 is divided by the autospectrum of the disturbance signal (noise to be canceled) Equation 5.4 provides a control measure in terms of the familiar ordinary coherence function.

$$\frac{S_{ee\,\min}}{S_{dd}} = 1 - \frac{|S_{rd}|^2}{S_{rr}S_{dd}} = 1 - \gamma_{rd}^2 \tag{5.4}$$

Examining this in some practical detail, it should be evident that if the coherence is unity between the reference signal and the disturbance (for all frequencies), the minimum of the autospectrum of the error signal will be zero as desired. Different levels of control can be achieved for varying values of the ordinary coherence. But it should also be evident that if the coherence is zero, the minimum error will equal the disturbance and the adaptive controller will offer no substantial authority over the noise.

5.1.2 MISO Problem Redefined

Before proceeding with the design of the COP filters, the problem which they actually solve will be reintroduced. Thus far the MISO problem of multiple references and one controller has been discussed in terms of adaptive control and energy source identification. In order

to understand the "big picture" each of these structures needs to be shown with respect to each other.

Returning to the automobile example, suppose an engineer had identified two primary sources of disturbance contributing to the overall noise, those being the engine and the fan. One control strategy that he may initially decide to take could be to sum the signal from the engine and the signal from the fan to create a single reference signal to be used in a SISO controller. The system level diagram for this structure is shown in Figure 5.2.

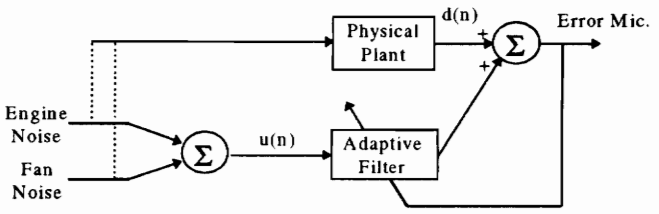


FIGURE 5.2 SIMPLE REFERENCE SUMMATION

The dotted lines between the two references and the physical plant indicate that at least *some* degree of coherence has been established between these signals and the disturbance. Since coherence is frequency dependent, it is likely that some frequency bins in each signal have considerable power but small coherence. This is precisely the problem encountered in performing this simple summation.

Suppose that the engine noise had a power spectrum which resembled the one shown in Figure 5.3 for a normalized frequency range. Also assume the power spectrum for the fan noise has a similar power spectrum as shown in Figure 5.4. For this simple illustration of ordinary coherence it will be assumed that each of the reference signals are mutually uncorrelated.

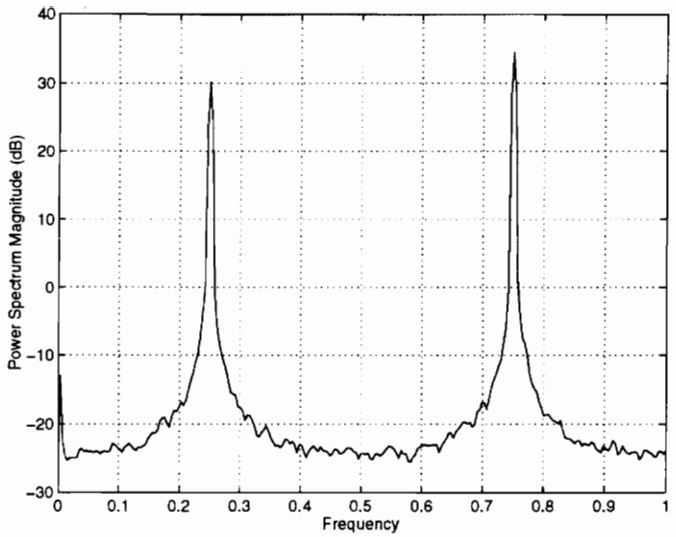


FIGURE 5.3 ENGINE NOISE POWER SPECTRUM

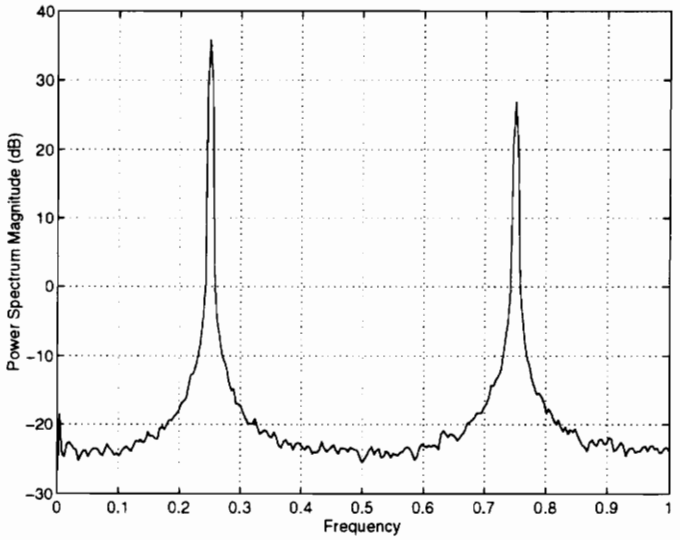


FIGURE 5.4 FAN NOISE POWER SPECTRUM

It should be evident that if these two signals were added together in a simple summation, the peaks would fall on top of each other. If all of these peaks contribute to the overall SPL, this problem is less serious than if one did and one did not. This is precisely the case which is illustrated now.

Suppose only the *first* peak in the engine noise is directly correlated to the overall noise inside the cabin, while the *second* peak in the fan noise is directly correlated to similar noise. Therefore all other frequency bins provide no coherence.

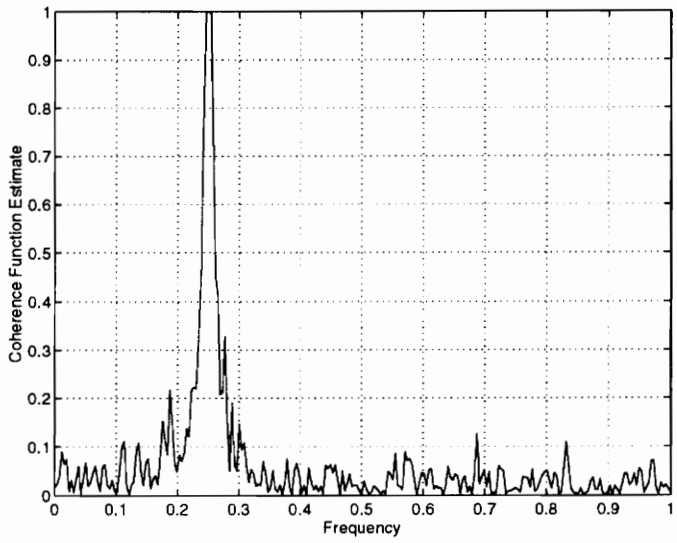


FIGURE 5.5 COHERENCE FROM ENGINE NOISE TO INTERIOR NOISE

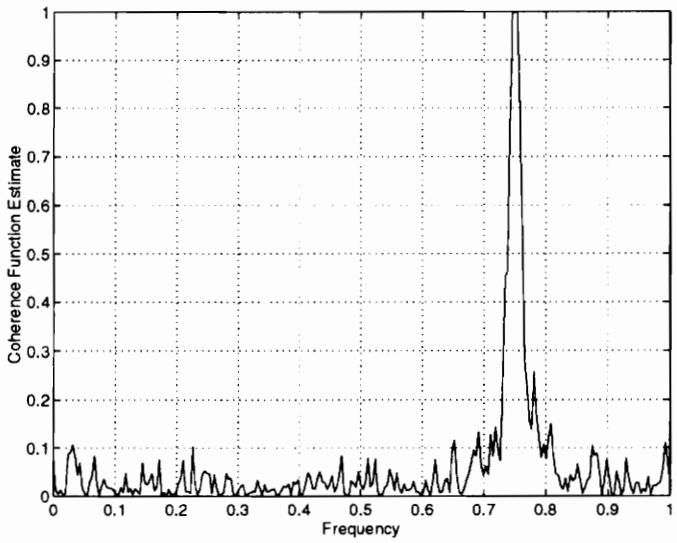


FIGURE 5.6 COHERENCE FROM FAN NOISE TO INTERIOR NOISE

If no opposing frequencies contribute to the overall noise, the ordinary coherence is sufficient to describe the linear path between each input and the output. Although each signal has excellent coherence for a single tone, the opposing tone in each signal will behave as uncorrelated signal content in a simple summation and corrupt the coherence of the sum as seen in Figure 5.7.

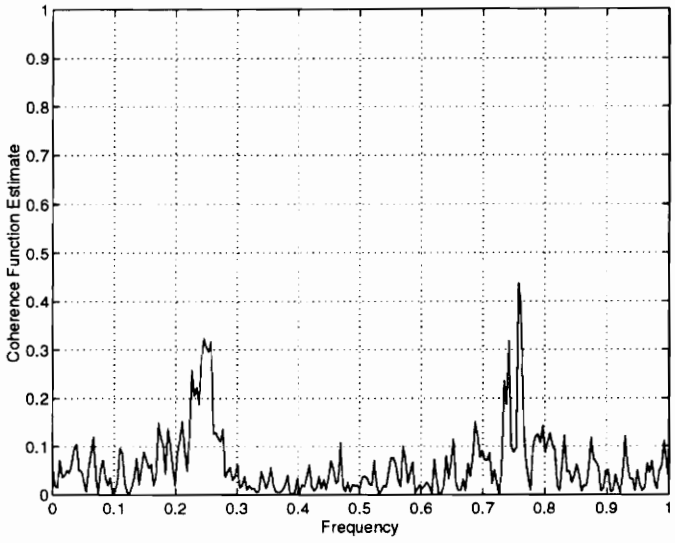


FIGURE 5.7 COHERENCE FROM SIMPLE SUM TO INTERIOR NOISE

So the low tone in the fan signal (which was incoherent) was added to the coherent tone in the engine signal at that frequency thereby causing a SNR issue. Since the incoherent tone is of a greater magnitude than the coherent one, the coherence of the sum is decreased according to Figure 4.5.

Two ways now exist which provide a solution to this problem: the adaptive array and the COP filtering technique. The adaptive array as discussed in Chapter 3, will be addressed in a comparison format in Section 5.5. The COP filtering technique will now be defined.

5.2 COHERENT OUTPUT POWER FILTER DESIGN

From an intuitive standpoint, the problem of the simple summation mentioned above can be easily solved. To maintain coherence of the sum, the coherent portion of each signal

must be amplified to a point where combining it with each of the other incoherent frequencies has no effect on the total coherence. Simply put, the noise should be suppressed while the coherent portions of the signal should be amplified. This is precisely the calculation which is performed by the COP filtering procedure.

5.2.1 TWO INPUT DESIGN

Continuing with the automobile example introduced above, the COP filtering procedure can be systematically defined by Figure 5.8

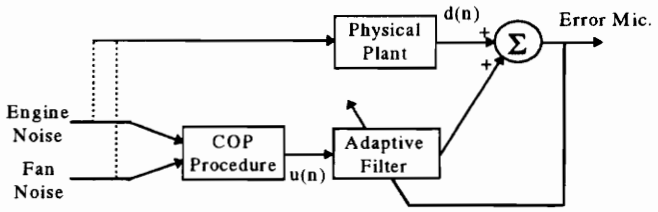


FIGURE 5.8 COP FILTER SYSTEM LEVEL DIAGRAM

The noise control system above, and any other similar systems having the same structure can be cast in terms of the energy source identification problem. Figure 5.9 uses the simplified diagram shown in Figure 5.8 and the physically defined signals to illustrate the energy source identification problem.

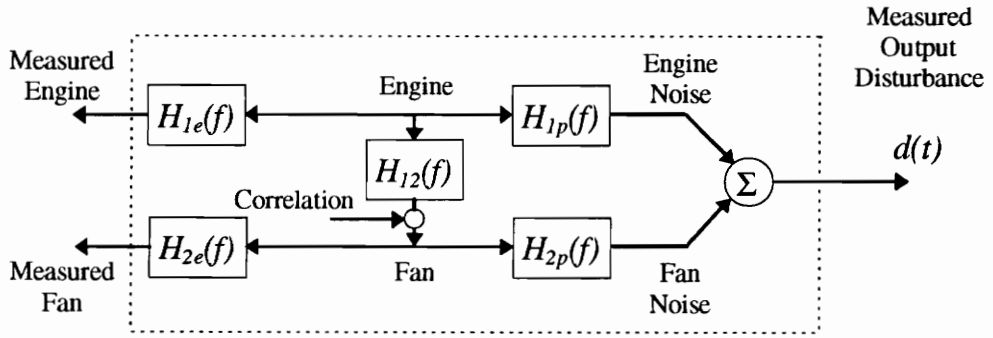


FIGURE 5.9 AUTOMOBILE ENERGY SOURCE IDENTIFICATION

Many of the transfer functions shown in Figure 5.9 are not explicitly illustrated in Figure 5.8, even though they will always exist. The fully generalized two input case shown, also includes a correlation path which *will* typically exist as well.

Returning to the energy source identification problem for this type of system, recall that on a frequency-by-frequency basis, the output power contribution of each of the measured inputs can be determined. For the case where the cross path exists from the engine to the fan, Equation 5.5 defines the total output from each input for the noiseless case.

$$G_{y:12} = \gamma^2_{y:12} G_{yy} = \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot1} G_{yy\cdot1} \qquad \textbf{(5.5)}$$

Recalling this relation from Equation 4.61, this indicates that the contribution to the overall output power from the first input (the measured engine) is given by the ordinary coherence times the overall output power. This particular term is denoted as the *coherent output power*. The contribution of the second input (the measured fan) is given by the second term in Equation 5.5, called the *partial coherent output power*. Therefore, for the noiseless case $G_{y:12} = G_{yy}$ by combining the coherent and partial coherent output powers based on the path direction.

To illustrate this in terms of actual signals, reconsider the automobile example; the following modification includes the possibility that a cross path does exist. Figure 5.10 shows the coherence *between* the two inputs is not negligible and must be taken into account as defined in Equation 5.5.

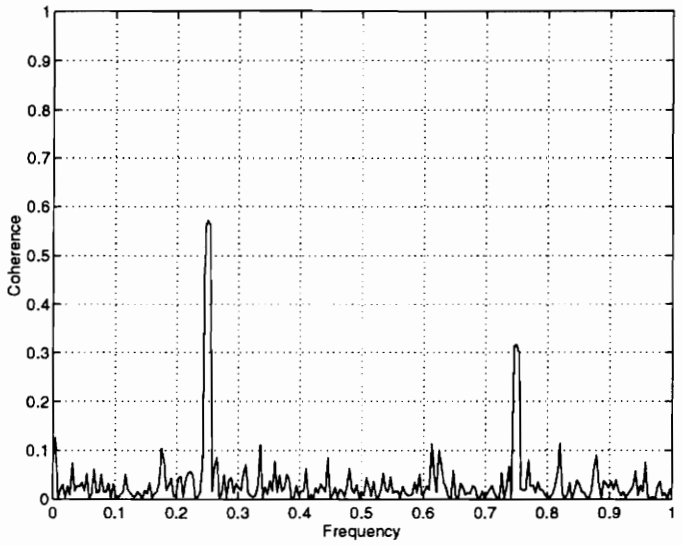


FIGURE 5.10 CORRELATED PATH COHERENCE (ENGINE TO FAN)

It has been proven mathematically that Equation 5.5 is valid for the noiseless case. Here it is shown in terms of actual signals. Since determination of the correct multiple coherent output power is highly dependent on cross path direction, a method (as opposed to a guess) was chosen to make this determination. As mentioned earlier, Park in [53] presented a reliable but complex method for just this task. Bendat and Piersol [3] provide a much simpler algorithm wherein the inputs are ordered based on their ordinary coherence to the measured output. This is the method which will be used for the remainder of the simulations as well as in the laboratory experiments.

As mentioned before, the plant (or disturbance noise) contains the low peak from the engine noise and the high peak from the fan noise. Neither of the other peaks included in either signal contribute to the overall noise reaching the interior of the automobile. A block diagram of this arrangement is shown in Figure 5.11 for further clarity.

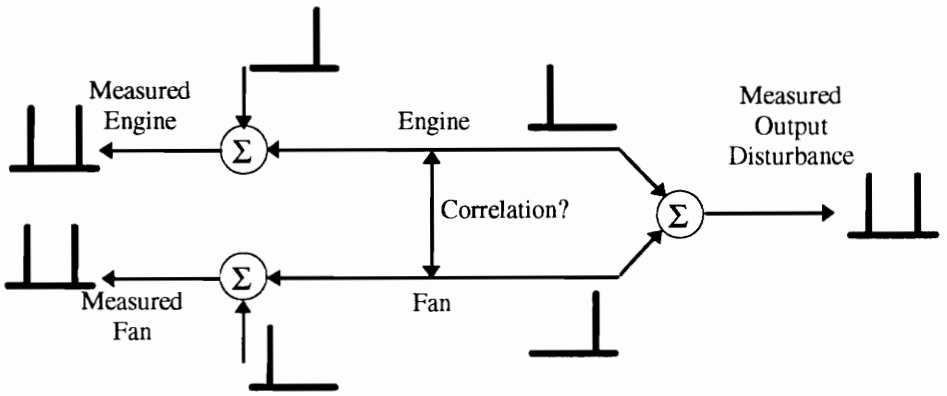


FIGURE 5.11 ILLUSTRATION OF AUTOMOBILE NOISE AND SIGNALS

Each of the dark lines indicates what the approximate power spectrum of each signal is at each location in the block diagram. The correlation between the engine and the fan is assumed to exist, however the direction is determined on a bin by bin basis. The power spectrum of the measured output is now shown in Figure 5.12.

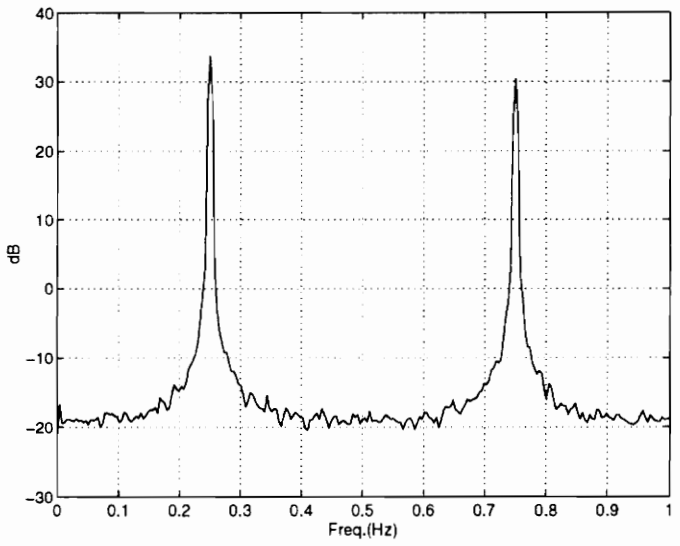


FIGURE 5.12 POWER SPECTRUM OF SIMULATED PLANT

The simulated plant represents the interior noise of the automobile which is the disturbance signal. Each of the two components in Equation 5.5 represent the individual contribution of the engine and the fan to this overall interior noise. Figures 5.13 and 5.14 show the

calculated components of Equation 5.5 for the engine signal (input one) and the fan signal (input two), respectively.

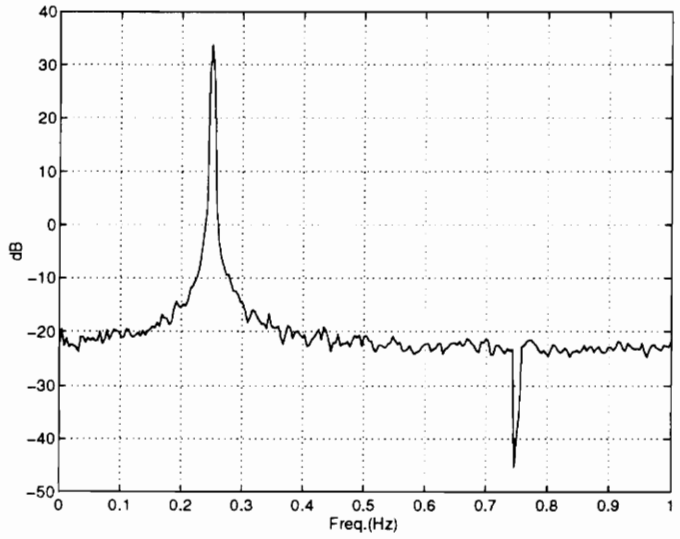


FIGURE 5.13 POWER CONTRIBUTION OF MEASURED ENGINE SIGNAL

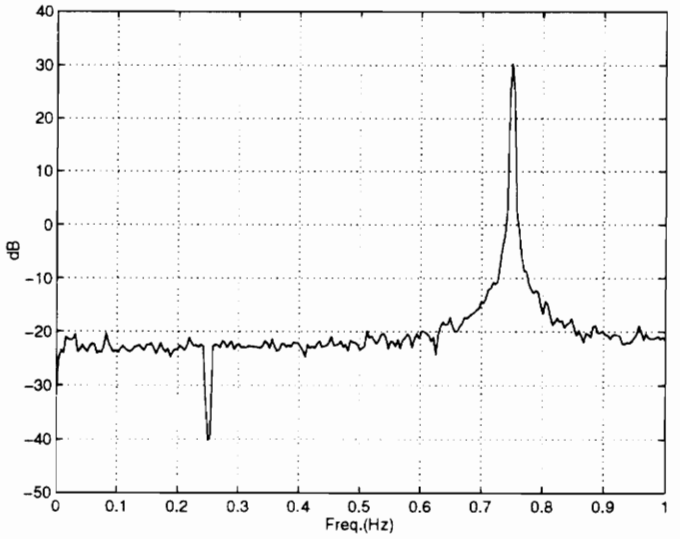


FIGURE 5.14 POWER CONTRIBUTION OF MEASURED FAN SIGNAL

According to the developments in Chapter 4 and Equation 5.5, the combination of these two signals should result in Figure 5.12 for the noiseless case. It should be emphasized that the cross path existence (as well as direction) must be taken into account to provide an accurate result. Figure 5.15 is the linear sum of Figure 5.13 and 5.14 which is identical to Figure 5.12.

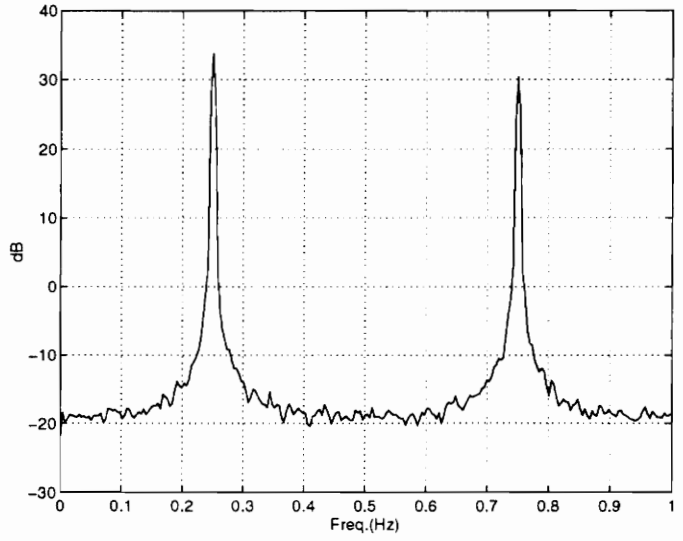


FIGURE 5.15 MULTIPLE COHERENT OUTPUT POWER OF ENGINE AND FAN

Since it has now been shown both analytically and in simulation that the multiple coherent output power is equal to the total output power for the noiseless case, the COP filters can be defined.

It is evident that the individual contributions of each input to the output can be represented by their respective coherent (or partial coherent) output powers. Thus in the frequency domain, the shape of the coherent output power represents the true frequency contribution of each individual signal. **The COP filters for each individual input are then designed as having the frequency response function that matches the coherent (or partial coherent) output power of each signal.** Therefore, upon filtering each input with the appropriately determined COP filter, the components of the signal not contributing to the overall power, will be suppressed while the coherent components will be amplified. This

process is nearly identical in concept to creating an orthogonal set of input signals which span the space of the output signal. Once this is done, the outputs of each of the COP filters can be linearly combined as in Equation 5.5 to represent a coherent reference which spans the space of the output.

Conceptually, this corresponds to applying a type of "graphic equalizer" to each input which has a level control for each frequency bin of the FFT. The components which are coherent are then boosted and the frequencies which represent uncorrelated content can be suppressed. The block diagram for the COP filtering procedure, including control, is shown in Figure 5.16.

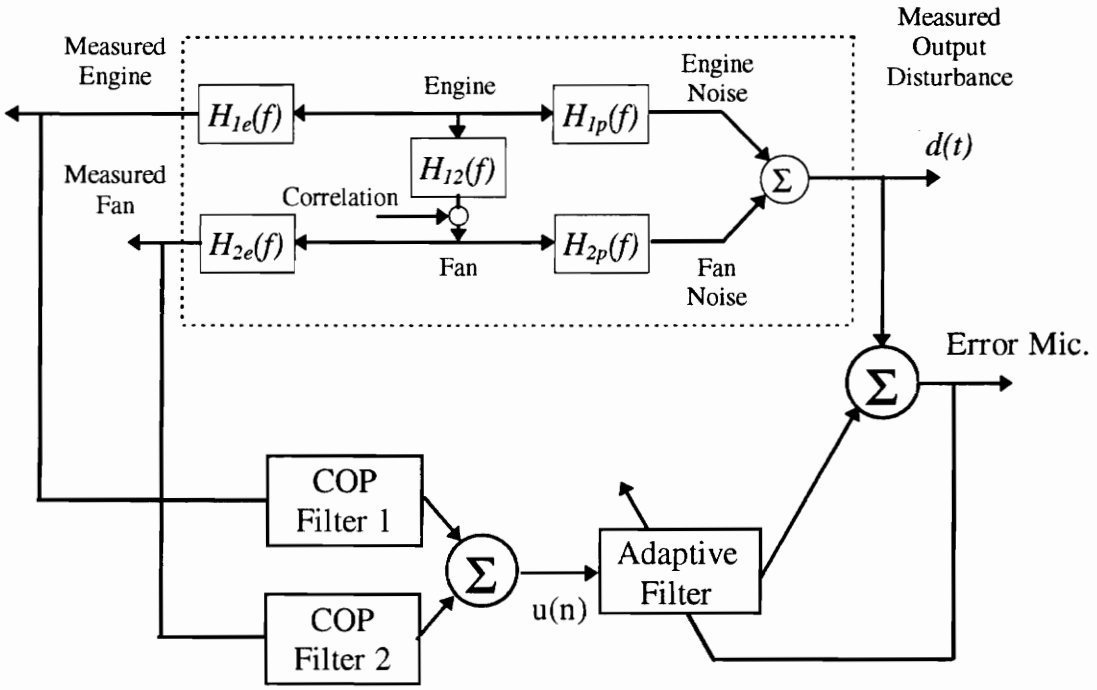


FIGURE 5.16 TWO-INPUT SINGLE-OUTPUT COP FILTERING SYSTEM

For the configuration shown above and the automobile example, the frequency response function magnitudes of the two COP filters can be defined as shown in Equations 5.6.

$$\left|COP_1\right| = \gamma_{1y}^2 G_{yy}$$
$$\left|COP_2\right| = \gamma_{2y\cdot1}^2 G_{yy\cdot1} \qquad \textbf{(5.6)}$$

where y is the output or disturbance of the system. The shape of each of these filters in the frequency domain is shown in Figures 5.13 and 5.14. Whether the value of the filter is the coherent output power or the partial coherent output power is dependent on the direction of the cross path and is a function of frequency. Therefore, for each frequency the actual value of the COP filter may be either the coherent or partial coherent output power. Since it is unknown until implementation which one will be used for each signal, the nomenclature of the filtering technology was generalized by classifying all such filters as coherent output power (COP) filters. (In fact, most should be referred to as partial coherent output power (PCOP) filters).

The magnitude is the critical component in this methodology since the phase of each bin is altered by the adaptive controller in order to cancel the noise. Also, the coherence calculation is not dependent on the phase information. However, the phase itself cannot simply be ignored in practical applications. Two possible interpretations of how to handle the phase design of the filter can be explored.

Recalling the overlap save technique of filtering in the frequency domain, it is possible to multiply the FRF of the filter by the Fourier transform of the input and modify only the magnitude in that particular operation. However, the operation of doing the FFT and IFFT impart linear phase lag throughout the filtering process, so the overall result of the overlap save technique using a zero phase FRF is that of an FIR filter having the magnitude as designated by Equations 5.6.

To redesign the COP filter as a tap delay line linear phase (FIR) filter, a linear phase vector can be created which has a magnitude of unity at each bin. Simply multiplying this by the magnitude of the COP filter previously designed, will provide a linear phase FRF of the COP filter. This can be transformed to the impulse response of the COP filter by the IFFT operation. For FFT sizes of N, the magnitude data will only be useful for N/2 bins.

However, the filter order will still be referred to as N since the IFFT is of order N and requires as many frequency domain data points to provide an N point FIR impulse response. The rigorous DSP terminology for this approach of FIR filter design is called the "frequency sampling technique" of digital filter design [13].

If it is desired to convert the COP filter to the equivalent FIR filter, a second less accurate (or appropriate) method can be applied. The magnitude being the most important component of the filter, it is possible to ignore the phase upon conversion back to the time domain, creating a zero vector for the imaginary components of the frequency response. The inverse FFT of this type of FRF provides the desired magnitude response of the COP filter while imparting a very small amount of non-linear phase to the FRF. In many cases this can be ignored because the adaptive filter performing the actual control operations provides the necessary phase for cancellation. The only criteria is that the input (reference) signal be coherent with the disturbance. Since the coherence is a function of magnitude only, phase in this portion of the design is not considered critical.

Actual implementation and simulation of the COP filter procedure will be provided in Section 5.3. Before proceeding however, generalizing the COP filters for multiple inputs will be addressed.

5.2.2 GENERALIZED DESIGN

Just as in the energy source identification problem, the COP filter design can be expanded to include the generalized case for any number of correlated inputs. Recall the multiple coherent output spectrum as defined in Equation 4.70.

$$
\begin{aligned}
G_{y:x} &= \gamma^2_{y:N!} G_{yy} \\
&= \gamma^2_{1y} G_{yy} + \gamma^2_{2y\cdot 1} G_{yy\cdot 1} + \cdots + \gamma^2_{Ny\cdot(N-1)!} G_{yy\cdot(N-1)!} \\
&= G_{yy}
\end{aligned}
\tag{5.7}
$$

For the noiseless case, the total output spectrum is equal to the sum of the coherent (and/or partial coherent) output powers of the correctly ordered inputs. Each of the terms in Equation 5.7 represent the individual contribution of each of the inputs to the overall output. In the same manner that the COP filters for two inputs were designed, COP filters for each of many inputs can be designed using Equation 5.7. For the case shown in Figure 4.9 where each of the inputs in the multi-input problem are ordered based on their mutual correlation, the magnitude of each individual COP filter can be expressed as shown in Equations 5.8.

$$
\begin{aligned}
\left|COP_1\right| &= \gamma^2_{1y} G_{yy} \\
\left|COP_2\right| &= \gamma^2_{2y\cdot 1} G_{yy\cdot 1} \\
\left|COP_3\right| &= \gamma^2_{3y\cdot 12} G_{yy\cdot 12} \\
\left|COP_n\right| &= \gamma^2_{ny\cdot(n-1)!} G_{yy\cdot(n-1)!}
\end{aligned}
\tag{5.8}
$$

Therefore for ordered input records the magnitude of each COP filter is the coherent (or partial coherent) output power of the respective input. One might wonder at this time as to why the partial coherence shape cannot be used as opposed to the coherent output powers. It is conceivable that two signals much like the engine and the fan could have two highly coherent components at the same frequency. Suppose the engine had a much larger peak and was contributing the most *power* to the output at that frequency. If the partial coherence rather than the partial coherent output power were used, the fan and the engine would receive equal weighting even though the engine was the primary contributor. Using the coherent *power* rather than the coherence *function* provides an indication of the coherent power contribution rather than an existence of a linear path. Thus the COP more accurately represents the actual contribution of each signal to the overall output.

Having presented the actual design and usage of the COP filters, an algorithm for their development will now be presented. Several scenarios and simulations will be addressed before examining a mathematical representation of the COP filters. As was previously mentioned, comparisons will be made to the simple summation both experimentally and analytically, as well as to the more common adaptive array.

5.3 COP FILTER IMPLEMENTATIONS

There are several possible arrangements for implementing this type of technology in practical situations. This section is concerned with describing some of the different realizations and developing an algorithm for implementing each of them. Several MATLAB simulations will then be introduced in order to show the end result of using the COP filter technology. The advantages and disadvantages with respect to the simple reference summations and the adaptive array are examined separately in following sections.

5.3.1 PROCEDURAL ALGORITHMS

The first and intended embodiment of the COP filtering algorithm assumes that the reference signals and disturbance signals are stationary and ergodic. This is clearly necessary because the calculation of the filter shape itself relies on determining the coherent (or partial coherent) output powers which rely on this assumption. Figure 5.17 illustrates the physical architecture of the COP filtering system for three inputs which can easily be generalized to the multiple input case.

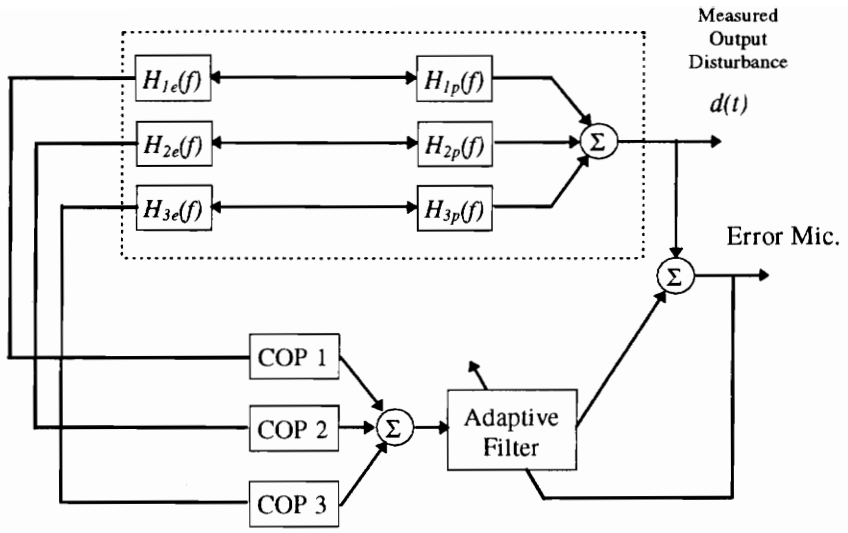


FIGURE 5.17 THREE INPUT SINGLE OUTPUT COP FILTERING SYSTEM

As should now be apparent, the COP filters are *designed* before any control begins and are subsequently used to individually filter each reference signal before being used for control. They are a function of the reference signals being measured and the measured output disturbance. Before any control begins, the error microphone is monitoring the output disturbance directly and is used as the "output" signal to create the COP filters. This type of preliminary design is also required to perform the filtered-X LMS algorithm in order to correctly identify the control to error path. Given this particular structure, it is now possible to organize into several steps, a procedural algorithm for implementing the COP filters.

- *Select candidate references and the desired error signal*
- *Before any control takes place, order each input signal for each bin*
- *Compute COP filters for each bin and save*
- *Filtering each input with its respective COP filter and adding them to form a single coherent reference, determine the control to error path for the filtered-X LMS*
- *Perform filtered-X control*

These are the general steps used in order to perform the COP filtering procedure for active noise control or active vibration control. The first decision which should be made

concerning the overall "big picture", is what type of computer architecture will be used. Specifically it is assumed that DSP chips will be used to perform all of the operations since they are much more efficient and specifically designed for these tasks. The question then remains as to whether one or two processors will be used. All of the above steps can be done using a single processor which runs in a variety of states. The first being the COP filter determination, the second being the control to error path identification, and finally the actual control. The difficulty with this arrangement is that filter sizes become limited when so many filtering operations are required within a single sample period. Using frequency domain (or block) filtering reduces these computations at the expense of update speed.

A second alternative is to use one processor to perform the COP filtering operations and a second to perform the system identification and control. Ultimately two parallel processors can perform at twice the sample rate for a given filter size. This is of course more expensive so design tradeoffs must be determined based on cost and speed. Concerning each design step in the list above, many possible alternatives exist. Since there are many variations within each step, they will now each be examined individually in further detail.

Select Candidate References

Typically it is required that the engineer seek out possible coherent reference signal(s) before any control can take place. This can be a time consuming operation which can quite possibly yield no acceptable results. If several signals are suspected to contribute to the overall noise, they are each individually tested for ordinary coherence. From the developments in Chapter 4, it should be clear that this is an unacceptable measure of linearity for a MISO system. It is likely that the engineer will find no coherent references when in fact they all have a degree of partial coherence. When combined using the COP filtering procedure, they will result in a single coherent reference.

The COP procedure then allows the engineer to simply "plug in" all candidate references (depending on how many inputs the algorithm has been created for) and visually see which references can be considered useful or not. Each of the individual COP filters can be displayed in real time indicating each of the signals individual contribution to the overall

power. Therefore as a preprocessing technique the COP filters remove a high level of uncertainty in selecting candidate references. They will automatically "throw away" non useful information. If one of the references produces a flat COP filter of very low magnitude, it is likely that the engineer should eliminate that signal as a candidate reference and investigate other alternatives.

Concerning the selection of an error signal, this depends on the location of the zone of silence desired. Since collocation of sources is often impossible, global control cannot be achieved. The noise control engineer relies on canceling the sound field in a single location to be adequate. The choice of the error signal for design of the COP filters should be the same as that chosen for control. Since control is occurring at the error microphone, it is desirable to have a reference which is coherent with the error microphone signal itself, else Equation 5.4 is violated.

Order Each Input Signal

The necessity of ordering input signals has been established in Chapter 4 in terms of the energy source identification problem. Since the solution to this problem is the basis of the COP filter design, it is imperative that this step be as accurate as possible. Two specific methods for performing this operation have been suggested.

Bendat and Piersol [3] suggest ordering inputs for multi-input systems by ranking them based on their ordinary coherence to the output. This is simply an approximation which can clearly be violated if significant cross paths exist between inputs. Even though this is only an estimation, it is the method which is used in all of the simulations and experiments demonstrating COP filters.

Park in [53] presents a more accurate method of determining the cross paths between inputs based on the checking of causality between each input. Just as in using the ordinary coherence, this must be determined on a bin by bin basis causing the computational complexity to be quite large. This method should be used if extremely accurate system identification is desired when using the COP filter design. (Since each filter represents the

individual contribution of each input, this method can also be considered as a type of system identification). It will be assumed that in most cases the ranking of inputs based on ordinary coherence will be sufficient.

Compute COP Filters for Each Bin

This section of the design algorithm has few alternatives since the basis of the design of the filters themselves are fixed. It should be emphasized that before proceeding, the COP filter for each input at each bin will likely be different based on the ordering of the inputs. This is illustrated below.

Consider three signals *a, b,* and *c* which are candidate references for a three input COP controlled ANC scheme as shown in Figure 5.17. Having completed the ordering procedure, each COP filter can be designed on a bin by bin basis by following Table 5.1.

TABLE 5.1 COP FILTERS FOR THREE INPUT CASE

Order of Inputs Previously Determined	COP Filter on Signal a	COP Filter on Signal b	COP Filter on Signal c
a ⇒ b ⇒ c	$S_{yy}\gamma^2_{ay}$	$S_{yy\cdot a}\gamma^2_{by\cdot a}$	$S_{yy\cdot ab}\gamma^2_{cy\cdot ab}$
a ⇒ c ⇒ b	$S_{yy}\gamma^2_{ay}$	$S_{yy\cdot ac}\gamma^2_{by\cdot ac}$	$S_{yy\cdot a}\gamma^2_{cy\cdot a}$
b ⇒ a ⇒ c	$S_{yy\cdot b}\gamma^2_{ay\cdot b}$	$S_{yy}\gamma^2_{by}$	$S_{yy\cdot ba}\gamma^2_{cy\cdot ba}$
b ⇒ c ⇒ a	$S_{yy\cdot bc}\gamma^2_{ay\cdot bc}$	$S_{yy}\gamma^2_{by}$	$S_{yy\cdot b}\gamma^2_{cy\cdot b}$
c ⇒ a ⇒ b	$S_{yy\cdot c}\gamma^2_{ay\cdot c}$	$S_{yy\cdot ca}\gamma^2_{by\cdot ac}$	$S_{yy}\gamma^2_{cy}$
c ⇒ b ⇒ a	$S_{yy\cdot cb}\gamma^2_{ay\cdot cb}$	$S_{yy\cdot c}\gamma^2_{by\cdot c}$	$S_{yy}\gamma^2_{cy}$

It should be apparent that as the number of inputs increases linearly, the number of filter possibilities and thus the number of computations increases factorially. To use the information in Table 5.1, conditionals within the DSP algorithm must be met. The order of

the inputs classifies what the three filters should be for each particular bin. It should be emphasized that each of the quantities in Table 5.1 are frequency dependent.

Once the filters themselves are computed, they can be stored separately and used again to actually filter each reference input during control. This procedure brings up the question as to whether or not the COP filters can be continually updated. There are several reasons why an *adaptive* COP filter may be undesirable.

The basis of the COP design itself relies on the assumption that the signals used to design the filters (the inputs and the output) are stationary and ergodic. If they are not, the design will be less than optimal and probably inaccurate. This observation aside, suppose the design of the filter were to be updated once every minute (since it typically requires about 60 averages to define random stationary ergodic signals). During control, the adaptive filter attempts to minimize the error signal, driving the primary power contributors to zero.

If the error signal is minimized to near zero, its signal characteristics will not have the necessary information to compute the COP filters on subsequent iterations. For an error signal of zero, the COP filters will be designed based on the inputs and the zeroed error signal producing a set of filters which suppress all bins because of lack of coherence at any bin. Once this occurs, a coherent reference cannot be produced and the control algorithm will fail.

Using the same argument it may be possible to use another error sensor which is not the same sensor to be minimized. This is illustrated in Figure 5.18.

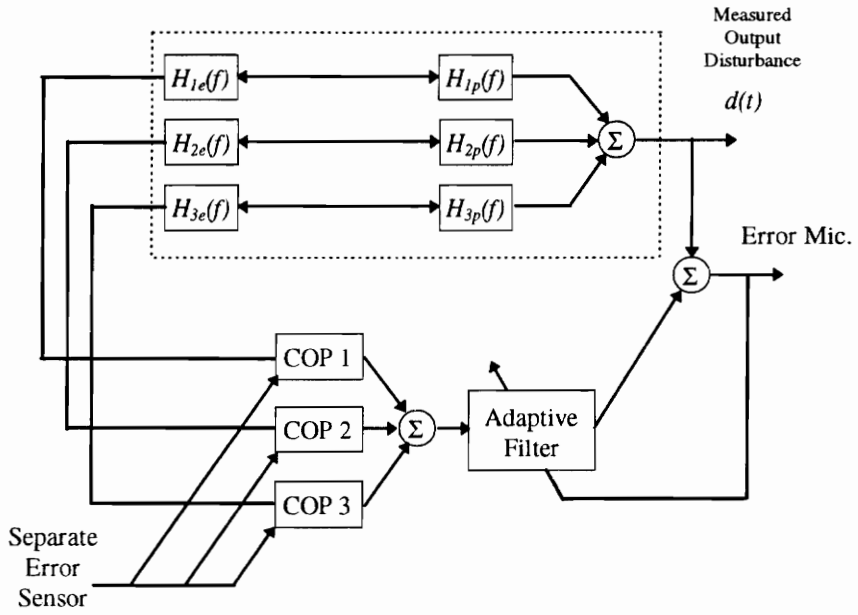


FIGURE 5.18 CONTINUALLY UPDATED COP FILTERING PROCEDURE

The main problems encountered with this type of arrangement exist in the determination of a coherent reference (satisfying Equation 5.4) as well as that of having an uncontrollable error sensor. If the separate error sensor were located in the same area as the error microphone but far enough to not be in total silence, it would be affected by the control speaker via spillover. The COP filters would then begin designing themselves based on the control signal (output of the adaptive filter) rather than on the error sensor and a positive feedback path would be introduced. In effect, the COP filtering algorithm would be calculating the coherence between the *control* signal and the input rather than the *error* signal and the input. Since the output of the adaptive filter is directly related to the input (in steady state) the coherence calculation would be bogus.

To avoid this problem one might suggest placing the separate error sensor far enough away from the original error sensor so as to not be affected by the control signal. It is fairly likely

that if this were done, the separate error sensor would hold little resemblance to the error signal. This is self defeating when trying to design a coherent reference based on Equation 5.4. The output signal would be different from the actual disturbance and thus the COP filters would incorrectly represent the contribution of each input to the separate reference signal rather than to the original disturbance.

If it were possible to obtain a signal which was highly coherent with the original *error* signal but remained unaffected by the control signal, the structure shown in Figure 5.18 could be realized and the COP filters could be continually updated. However, since the original design was intended to operate on stationary ergodic signals, updating of the COP filter will not be considered. Implementation of the COP filters will emulate that of the filtered-X and occur prior to control. Each of the COP filters will then be used on each reference signal followed by a summation and the control to error plant estimation.

COP Filter Each Input and Add

There are several ways in which the filtering of the input data can occur after the COP filter has been designed. Since the COP filter itself is designed first in the frequency domain, it is logical that block filtering of the input data would be desirable. There are several structures which can be used depending on the overall arrangement of DSP's and the control algorithm.

Consider first that the entire system is implemented on a single processor. Having designed the COP filters and performed the system identification (both remaining in the frequency domain) the most efficient methodology would be to implement an algorithm based entirely in the frequency domain. Figure 5.19 shows this type of structure.

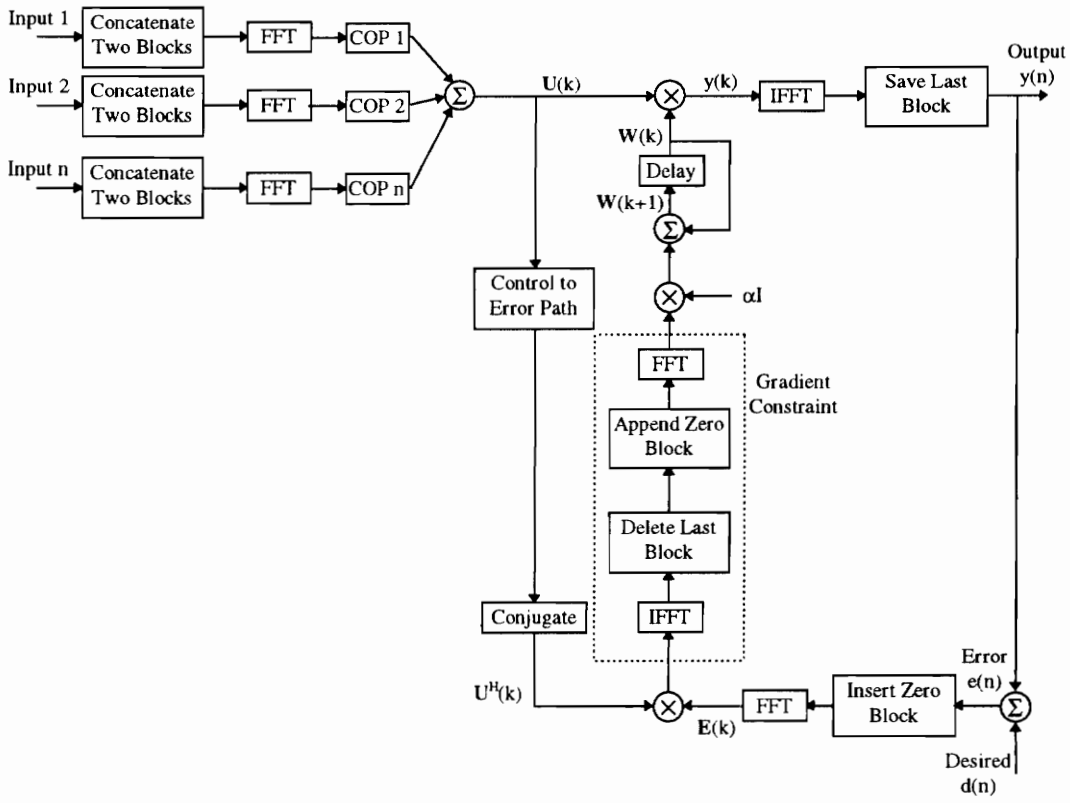


FIGURE 5.19 FREQUENCY DOMAIN COP FILTERING PROCEDURE

Each of the inputs are combined in the frequency domain and used to run the control algorithm also in the frequency domain. This procedure only adds one FFT operation for each additional input to the overall computations calculated in Table 3.1. N multiplication's must also be included when using the filtered reference (i.e. control to error path). Figure 5.19 represents the preferred (and most computationally efficient) method of COP filtering (after the filters have been designed).

One of the primary motivations in developing the COP filtering technique was to present a way to develop a coherent reference to be used in a SISO controller which was already designed and available. If this is the case the next most desirable implementation of the COP filtering procedure would be to retroactively fit a COP filtering algorithm. The most efficient method for doing this still lies in the frequency domain, independent of the nature of the adaptive control portion of the overall design. Figure 5.20 shows the general structure of this type of "add-on" hardware.

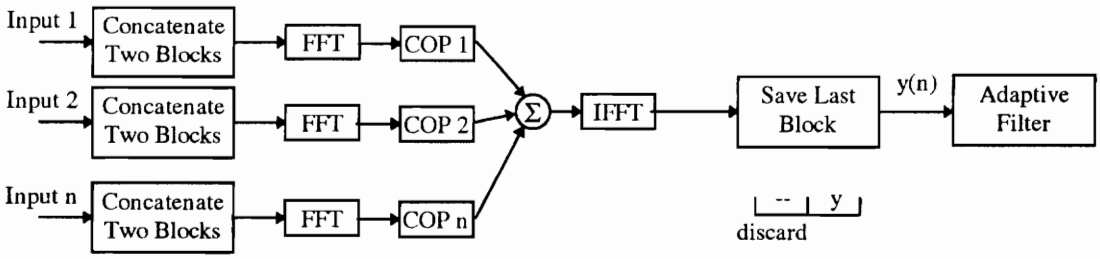


FIGURE 5.20 EXTERNAL COP FILTERING PROCEDURE

Each of the COP filtering procedures done in the frequency domain is performed as shown in the middle portion of Figure 3.22 but is represented as a single block here for brevity. Computational advantages of block filtering in general have been discussed. Regardless of the design of the adaptive filter, the creation of the coherent reference via frequency domain COP filtering is computationally advantageous.

A final structure which can be used represents the most off-line (and real time) computationally inefficient method for creating the coherent reference. If it were desired to filter each of the inputs in the time domain, it is possible to convert the frequency domain COP filters to equivalent FIR filters and sum the outputs, all in the time domain. As an additional piece of hardware, or a resident single DSP algorithm, converting the COP filters to the time domain wastes computation in the IFFT operations as well as the time domain vs. frequency domain expense. This method may be desirable if efficient FIR filter codes are already available as will be seen in Chapter 6.

Once the decision has been made concerning the overall structure of implementing the COP procedure, the control to error path must be determined in order to generate the

filtered reference signal. It is most desirable to perform this system identification using a white noise source in order to excite all the modes of the specified path. Typically however, the reference signal to be used in the control portion of the algorithm is used as the exciting signal during the system identification. The main reason for this is that only the very loud signals (tonals) will be controlled therefore those are the frequencies where the identification is most critical. If weights are wasted on modes that are never controlled, a poor system identification may result at other more critical frequencies.

In this light, the COP filtering procedure provides a benefit toward control to error path system identification performed using the reference signal. If the reference signal used during system identification is not correlated with the noise source, system ID at incoherent peaks will be accurate which will result in unstable control as explained in Section 3.3. If however the COP generated reference is used to perform system ID, it can be assured that proper identification of the path at the frequencies of primary interest will occur.

Perform Filtered-X Control

The final step in the COP procedure is to use the single reference to perform control. The actual control is no different from what has been explained in Chapter 3. The method must be chosen based on existing hardware and software as well as computational complexity. Clearly the entire procedure realized in the frequency domain is most efficient while varying degrees of time domain filtering add to the overall number of computations.

Several simulations of the effects of the COP procedure on coherence of the reference signal will now be explored. Since it has been shown that control is a function of coherence by Equation 5.4, strict emphasis will not be placed on effects on control performance until Chapter 6.

5.3.2 MATLAB SIMULATIONS

Continuing with the automobile example it is now possible to examine the coherence of the original sum as compared to the COP filtering technique. Figure 5.7 showed the coherence of the engine signal plus the fan signal to the interior noise (termed as the output). Using

the COP filters as defined by Equations 5.6, the FRF's of each of these filters are displayed in Figures 5.21 and 5.22. (Appendix A contains several of the MATLAB files used to generate these simulations).

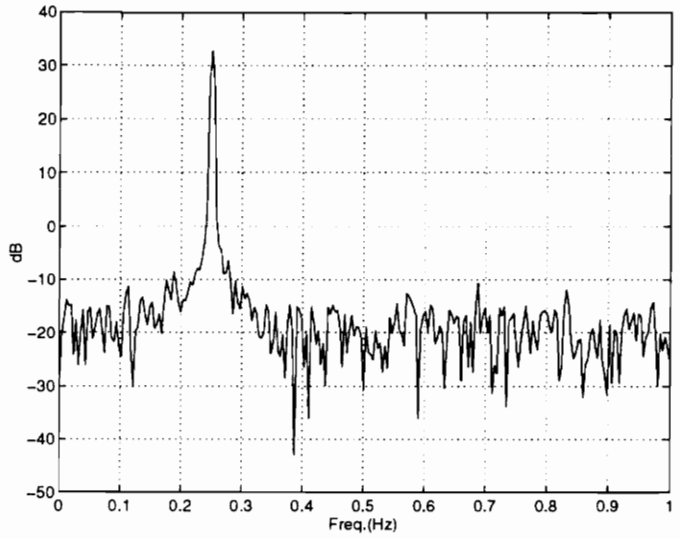


FIGURE 5.21 FRF OF COP FILTER FOR ENGINE NOISE

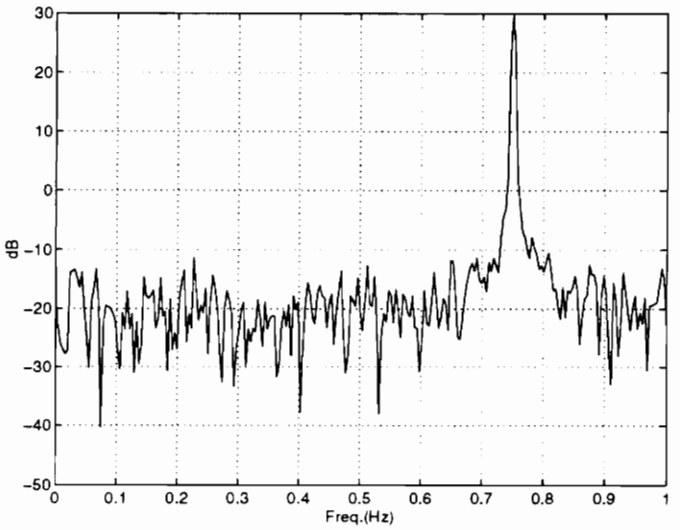


FIGURE 5.22 FRF OF COP FILTER FOR FAN NOISE

It is now evident that when the input signals are filtered and recombined, the incoherent components are suppressed into the noise of the original reference signals. Figure 5.7 illustrated the detrimental results of a simple summation of the non COP filtered signals while Figure 5.23 now shows the advantages of using the COP filtering technique.

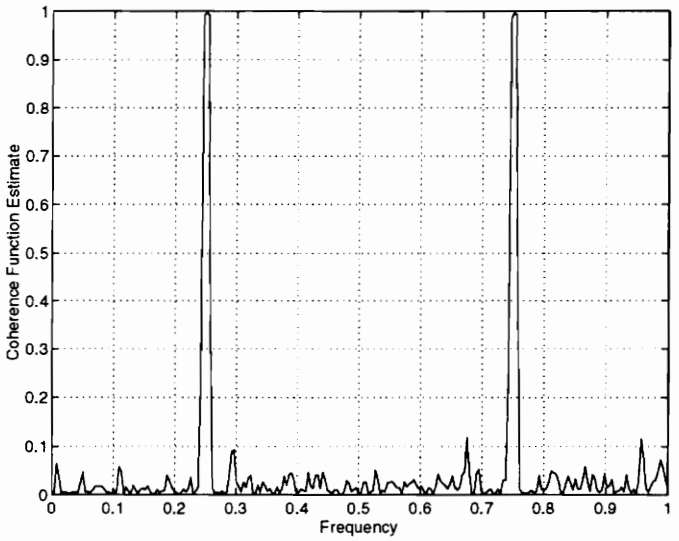


FIGURE 5.23 COHERENCE OF COP FILTERED REFERENCE TO OVERALL NOISE

It is evident that the COP filtered summation will perform much better than the simple summation for a SISO adaptive controller. A mathematical evaluation of the relationship between the COP filtered sum and the simple summation is necessary to show this for all cases. As will be shown, the number of possibilities are far too great to generalize the result with a simple answer, even for the two input case.

Turning to a more complicated and generalized example, consider a three-input single-output system where the first reference has low frequency coherent energy, the second has mid-band coherent energy and the third is primarily a high frequency contributor. Each of the three references however, contain energy at all frequencies causing the simple summation to become non-coherent. Figures 5.24, 5.25 and 5.26 show the power spectra of each of these three inputs. The bandwidths in each signal which *do not* contribute to the overall power of the output were amplified in the creation of the signals themselves and

should be apparent in each of the three spectra. This was done in order to decrease the coherence of the simple summation (by increasing the noise) and show that the COP filters are capable of determining coherent signals even when the noise is of a greater magnitude.

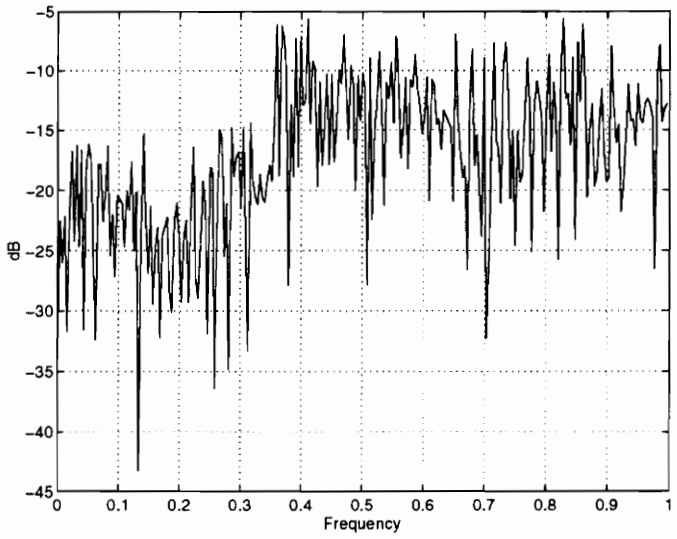


FIGURE 5.24 POWER SPECTRUM OF CANDIDATE REFERENCE 1

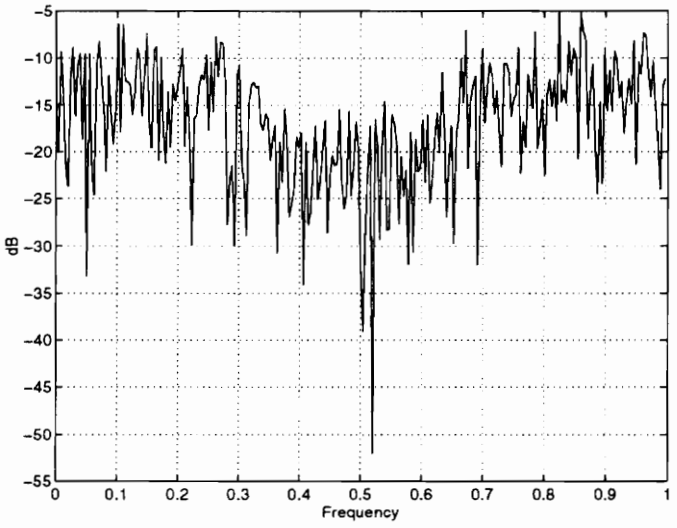


FIGURE 5.25 POWER SPECTRUM OF CANDIDATE REFERENCE 2

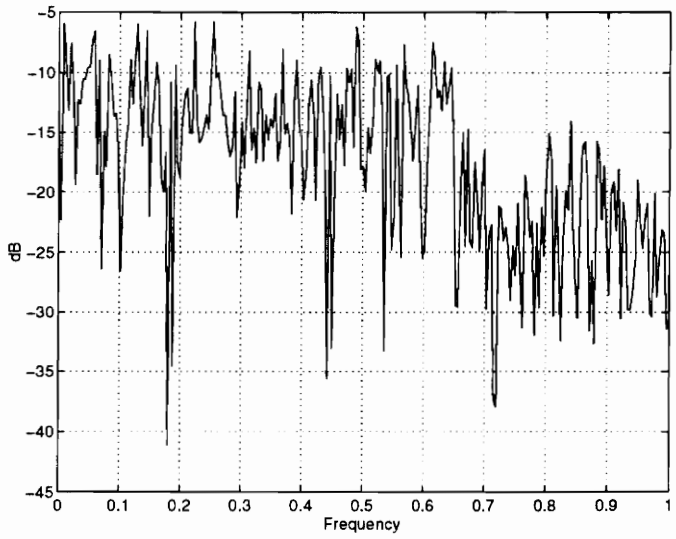


FIGURE 5.26 POWER SPECTRUM OF CANDIDATE REFERENCE 3

Using the output (not shown) which resembled a white noise source with contributions from the low, middle, and high bands of the three references, the COP filters were designed as specified in the previous section. Figures 5.27, 5,28, and 5,29 show the FRF of each of the COP filters for each of the respective inputs.

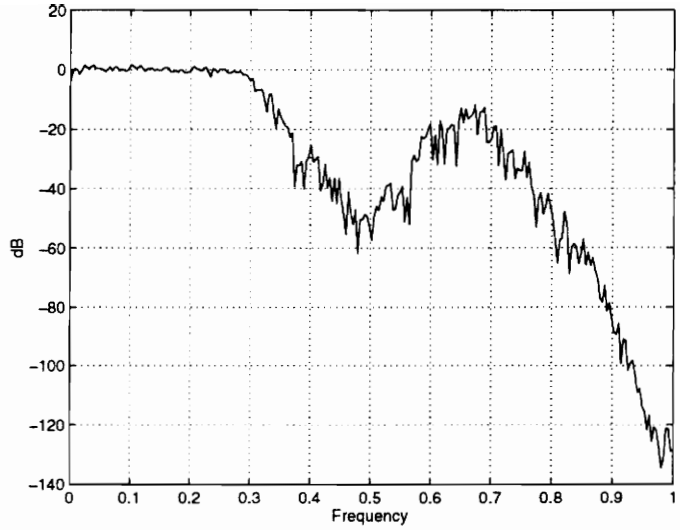


FIGURE 5.27 FRF OF COP FILTER FOR REFERENCE 1

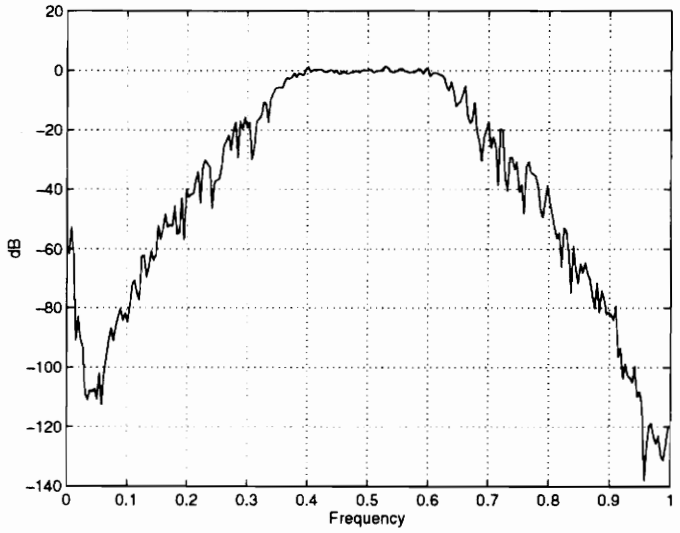


FIGURE 5.28 FRF OF COP FILTER FOR REFERENCE 2

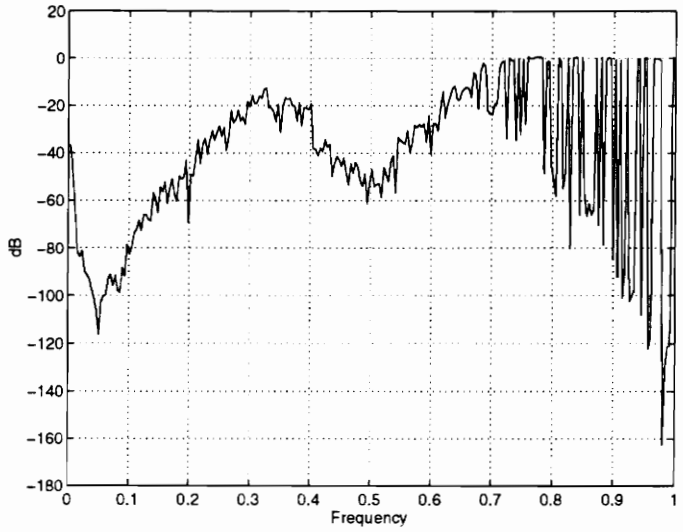


FIGURE 5.29 FRF OF COP FILTER FOR REFERENCE 3

Now, examining the reference signal for the simple summation, it should be evident that the noise in the bandwidths of the opposing references will overpower the coherent portions of each signal and result in an unacceptably small coherence. The COP filters in Figures 5.27, 5.28, and 5.29 show each of the signals' contributions to the overall output and can be examined with respect to the previous three signals to provide information regarding which portions were passed to the output in the simulation. Figure 5.30 shows the overall coherence of the simple summation of the original references linearly combined (Figures 5.24, 5.25, and 5.26), to the output.

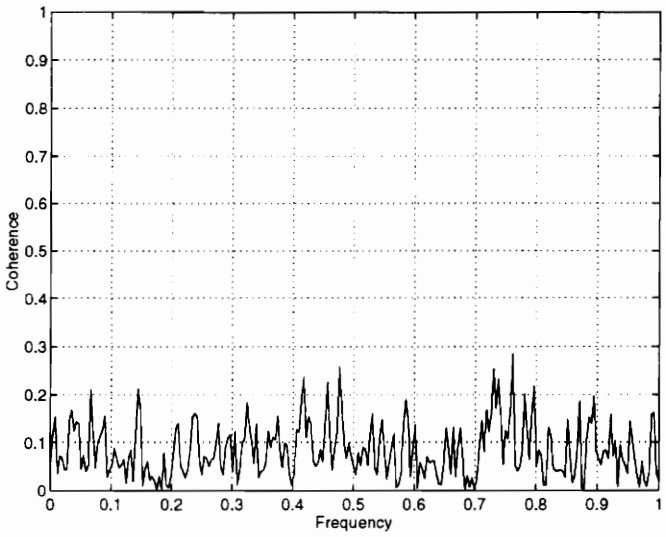


FIGURE 5.30 COHERENCE OF ORIGINAL SUMMATION

This is clearly unacceptable and the noise control engineer may incorrectly assume that none of the candidate references could be used in the adaptive control procedure. However, if a more careful approach were taken in using the COP filtering procedure, the coherence shown in Figure 5.31 would result.

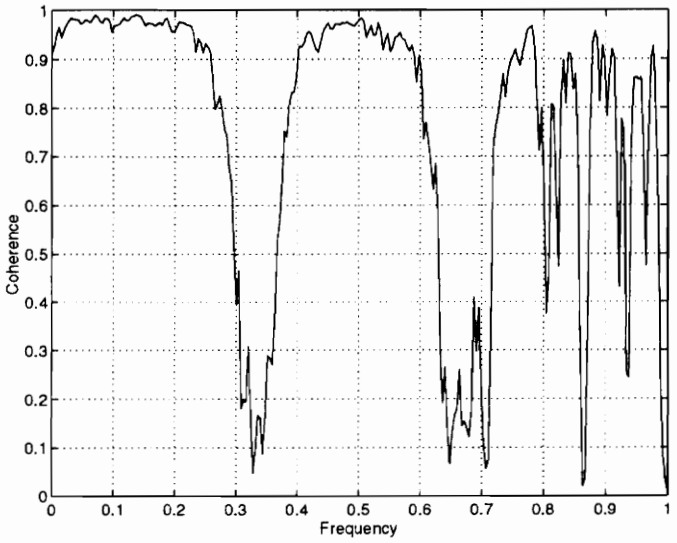


FIGURE 5.31 COHERENCE OF COP FILTER SUMMATION

The coherence for the COP output summation is less than perfect at a few frequencies due to the extraneous output *measurement* noise which was included to add an important practical consideration. It should also be taken under advisement that the design of the filters themselves are only an approximation of the optimal filter which would be based on an infinite number of averages. However, it is evident that the COP filter summation provides an exceptional increase in the coherence of the new reference signal to the output (disturbance to be canceled). This increase in coherence will be directly translated to an dramatic increase in controller performance. It is very likely that the original summation will fail to provide any control authority. This will be further examined in Section 5.4.1 and Chapter 6.

Having shown the benefits of the COP filtering procedure in creating a single coherent reference signal, it is now necessary to make the comparison with the simple summation in an analytical manner. Section 5.4 describes the relationship with the simple summation mathematically and presents some advantages and disadvantages of using the COP filtering procedure as opposed to the strict linear combination of candidate references.

5.4 COMPARISON WITH SIMPLE REFERENCE SUMMATION

5.4.1 SIMULATION RESULTS

Returning to the two tonal example illustrated in Figure 5.11, it has become clear that using the COP filtering procedure improves coherence of both tonals by reducing the noise in the opposing candidate reference signals. Until now, only analytical evidence suggests that an increase in coherence translates directly to an increase in control performance.

For the simulated case addressed in Section 5.3.2, it is now possible to illustrate the control performance based on coherence of the linear reference summation and the COP filtered reference summation as defined by the signals shown in Figure 5.32

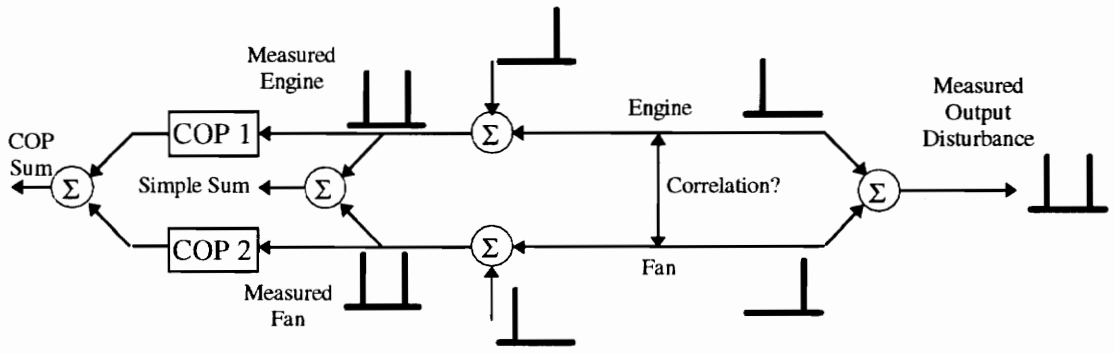


FIGURE 5.32 SIMPLE SUMMATION VS. COP FILTER DIAGRAM

Both the COP summation and the simple summation have power spectra that look like the measured output disturbance but only the COP signal has high coherence at each peak. The noise introduced in each signal following the originating source covers the coherence component upon simple linear combination. This can be seen in Figures 5.7 and 5.23 for the simple summation and the COP filter summation respectively.

The ultimate goal is of course to improve adaptive feedforward control performance by creating a coherent reference which has as many components of the output disturbance as possible. The COP filtering procedure will ensure this as long as a sufficient number of

references that adequately represent the output are used. Simply combining these signals in a linear summation will provide the control effectiveness as shown in Figure 5.33.

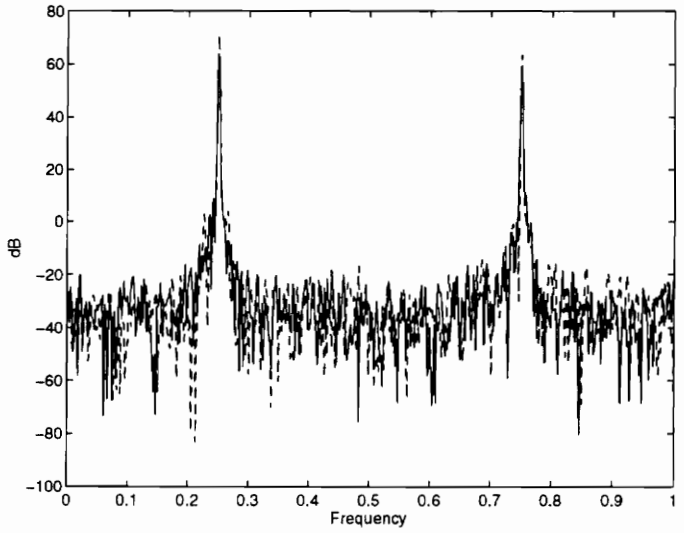


FIGURE 5.33 CONTROL PERFORMANCE OF SIMPLE REFERENCE SUMMATION

Both the plant and the converged controlled version of the disturbance are plotted together with the original disturbance as the dotted plot. The frequency domain LMS algorithm was used in conjunction with block filtering of the reference signal by the COP filters. After performing the COP filtering procedure on each of the references and combining them to form a single coherent reference, the control achieved in Figure 5.34 results.

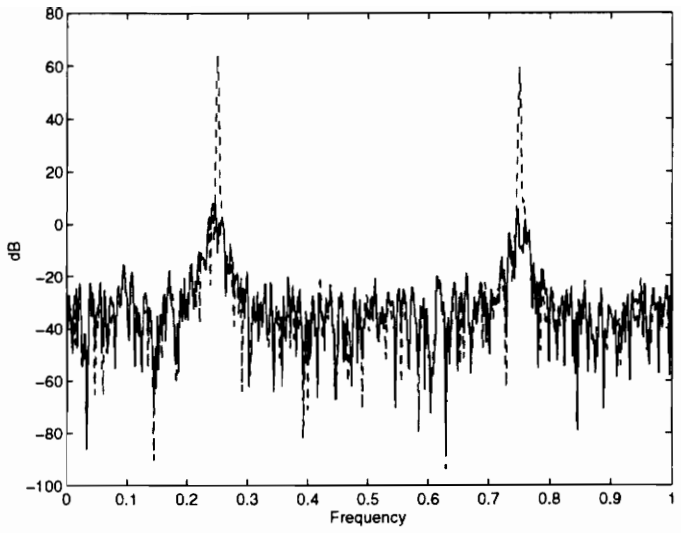


FIGURE 5.34 CONTROL PERFORMANCE OF COP REFERENCE SUMMATION

Clearly the control is much more acceptable and the performance increase parallels that which was expected from the estimated coherence function.

It should now be emphasized that only a single case for two tonals has been exclusively examined regarding the comparison of the simple summation to the COP filter summation. This special case was designed because it was easy to understand and effectively displayed the beneficial aspects of the COP filtering procedure. In addition, even though the COP filters are all designed to take into account any cross-correlation paths existing at the originating inputs, it is fairly difficult to accurately generate these types of signals either in a simulation or laboratory setting. Often the correlated paths exist physically such as through a source generating both a vibration and acoustic disturbance. In cases such as this, the COP filters will appropriately assign weighting to the frequency bins by estimating these cross paths.

No special effort has been made to include cross paths in the simulated signals as they won't directly affect the end result. Since the COP filters assign appropriate weighting based on including the cross paths, the possibility of having a cross path equal to zero is merely a special case as addressed in Chapter 4.

To create a more generalized comparison, Section 5.4.2 examines the special case of no cross path existing in the diagram of Figure 5.32. Analytical comparisons are made between the simple sum result and the COP filter sum result as various transfer functions are used to span a limited space of results.

5.4.2 ANALYTICAL COMPARISON

The strict mathematical comparison of the simple sum coherence to the COP sum coherence is quite involved and beyond the scope of this work. The initial development required for the comparison of the special case of the two-input single-output model will be addressed in an analytical manner. This development can serve as a launching point toward more detailed analyses which are required in order to present a strict result comparing the two methodologies.

Consider the two-input single-output case where it is assumed no cross path exists between the two originating inputs as shown in Figure 5.35.

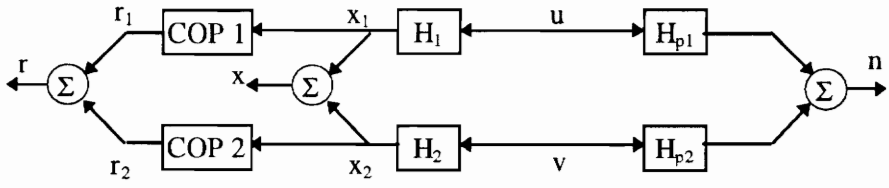


FIGURE 5.35 TWO-INPUT SINGLE-OUTPUT SPECIAL CASE COMPARISON

The COP filtering procedure is intended to increase the ordinary coherence of the signal r to n over that of x to n. It has already been shown for a few very specific cases that the results can be quite dramatic. Surprisingly however, there are some very rare cases where the COP filtered reference signal coherence can be slightly *reduced* as compared to the original reference. This is clearly an undesirable result and must be quantified as well as possible.

It should be immediately clear that for the case of two uncorrelated inputs (as shown in Figure 5.35) that there are six independent variables which can vary for different systems.

There are two "electrical" path transfer function denoted H_1 and H_2 which can take on theoretically any value. Next, the powers of the two originating sources, u and v, can also vary between zero and (theoretically) infinity. Finally, the plant transfer function (H_{p1} and H_{p2}) can also assume any possible value. With each of these variables having magnitude and phase terms (real and imaginary) which can also vary with frequency, the mathematical comparison of γ_{xn}^2 to γ_m^2 becomes exceedingly complex.

The first step in attempting to effectively compare the two methods involves developing mathematical expressions for the two coherence values in terms of the source variables. Consider first, the coherence between the simple summation x and the disturbance noise n. The general expression for the ordinary coherence is written because the overall system "appears" as a SISO system to the adaptive controller; whose only concern is the ordinary coherence of the input (reference) to the disturbance noise. General system relations are then substituted in until the relation exists in terms of the transfer functions and originating signals as shown in Equations 5.8.

$$
\begin{aligned}
\gamma_{xn}^2 &= \frac{|G_{xn}|^2}{G_{xx}G_{nn}} \\
G_{xn} &= G_{x_1 n} + G_{x_2 n} \\
&= H_1^* G_{un} + H_2^* G_{vn} \\
G_{xx} &= G_{x_1 x_1} + G_{x_2 x_2} \\
&= |H_1|^2 G_{uu} + |H_2|^2 G_{vv} \\
G_{nn} &= |H_{p1}|^2 G_{uu} + |H_{p2}|^2 G_{vv} \\
\gamma_{xn}^2 &= \frac{|H_1^* G_{un} + H_2^* G_{vn}|^2}{\left(|H_1|^2 G_{uu} + |H_2|^2 G_{vv}\right)\left(|H_{p1}|^2 G_{uu} + |H_{p2}|^2 G_{vv}\right)}
\end{aligned}
\qquad \textbf{(5.8)}
$$

It should be emphasized that each of the components of Equations 5.8 are functions of frequency but are not explicitly expressed as such in order to simplify the expressions. So

references that adequately represent the output are used. Simply combining these signals in a linear summation will provide the control effectiveness as shown in Figure 5.33.

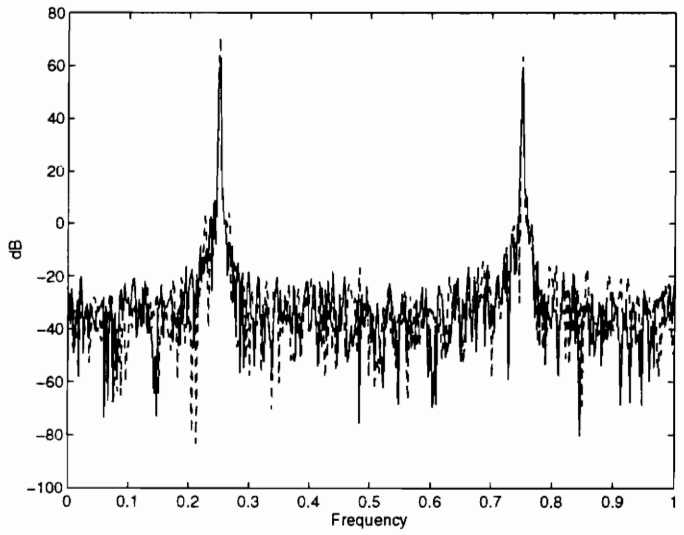


FIGURE 5.33 CONTROL PERFORMANCE OF SIMPLE REFERENCE SUMMATION

Both the plant and the converged controlled version of the disturbance are plotted together with the original disturbance as the dotted plot. The frequency domain LMS algorithm was used in conjunction with block filtering of the reference signal by the COP filters. After performing the COP filtering procedure on each of the references and combining them to form a single coherent reference, the control achieved in Figure 5.34 results.

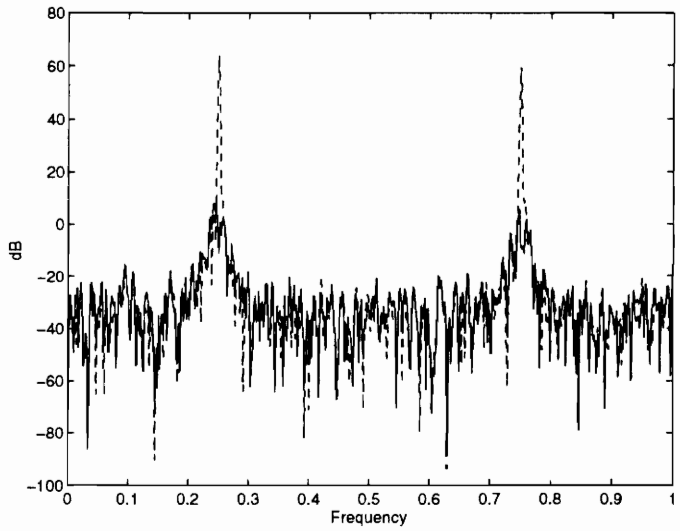


FIGURE 5.34 CONTROL PERFORMANCE OF COP REFERENCE SUMMATION

Clearly the control is much more acceptable and the performance increase parallels that which was expected from the estimated coherence function.

It should now be emphasized that only a single case for two tonals has been exclusively examined regarding the comparison of the simple summation to the COP filter summation. This special case was designed because it was easy to understand and effectively displayed the beneficial aspects of the COP filtering procedure. In addition, even though the COP filters are all designed to take into account any cross-correlation paths existing at the originating inputs, it is fairly difficult to accurately generate these types of signals either in a simulation or laboratory setting. Often the correlated paths exist physically such as through a source generating both a vibration and acoustic disturbance. In cases such as this, the COP filters will appropriately assign weighting to the frequency bins by estimating these cross paths.

No special effort has been made to include cross paths in the simulated signals as they won't directly affect the end result. Since the COP filters assign appropriate weighting based on including the cross paths, the possibility of having a cross path equal to zero is merely a special case as addressed in Chapter 4.

To create a more generalized comparison, Section 5.4.2 examines the special case of no cross path existing in the diagram of Figure 5.32. Analytical comparisons are made between the simple sum result and the COP filter sum result as various transfer functions are used to span a limited space of results.

5.4.2 ANALYTICAL COMPARISON

The strict mathematical comparison of the simple sum coherence to the COP sum coherence is quite involved and beyond the scope of this work. The initial development required for the comparison of the special case of the two-input single-output model will be addressed in an analytical manner. This development can serve as a launching point toward more detailed analyses which are required in order to present a strict result comparing the two methodologies.

Consider the two-input single-output case where it is assumed no cross path exists between the two originating inputs as shown in Figure 5.35.

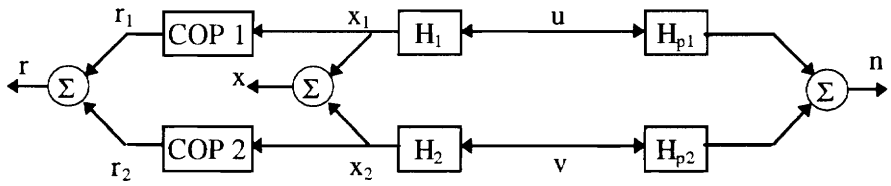


FIGURE 5.35 TWO-INPUT SINGLE-OUTPUT SPECIAL CASE COMPARISON

The COP filtering procedure is intended to increase the ordinary coherence of the signal r to n over that of x to n. It has already been shown for a few very specific cases that the results can be quite dramatic. Surprisingly however, there are some very rare cases where the COP filtered reference signal coherence can be slightly *reduced* as compared to the original reference. This is clearly an undesirable result and must be quantified as well as possible.

It should be immediately clear that for the case of two uncorrelated inputs (as shown in Figure 5.35) that there are six independent variables which can vary for different systems.

There are two "electrical" path transfer function denoted H_1 and H_2 which can take on theoretically any value. Next, the powers of the two originating sources, u and v, can also vary between zero and (theoretically) infinity. Finally, the plant transfer function (H_{p1} and H_{p2}) can also assume any possible value. With each of these variables having magnitude and phase terms (real and imaginary) which can also vary with frequency, the mathematical comparison of γ_{xn}^2 to γ_m^2 becomes exceedingly complex.

The first step in attempting to effectively compare the two methods involves developing mathematical expressions for the two coherence values in terms of the source variables. Consider first, the coherence between the simple summation x and the disturbance noise n. The general expression for the ordinary coherence is written because the overall system "appears" as a SISO system to the adaptive controller; whose only concern is the ordinary coherence of the input (reference) to the disturbance noise. General system relations are then substituted in until the relation exists in terms of the transfer functions and originating signals as shown in Equations 5.8.

$$
\begin{aligned}
\gamma_{xn}^2 &= \frac{|G_{xn}|^2}{G_{xx}G_{nn}} \\
G_{xn} &= G_{x_1n} + G_{x_2n} \\
&= H_1^* G_{un} + H_2^* G_{vn} \\
G_{xx} &= G_{x_1x_1} + G_{x_2x_2} \\
&= |H_1|^2 G_{uu} + |H_2|^2 G_{vv} \\
G_{nn} &= |H_{p1}|^2 G_{uu} + |H_{p2}|^2 G_{vv} \\
\gamma_{xn}^2 &= \frac{|H_1^* G_{un} + H_2^* G_{vn}|^2}{\left(|H_1|^2 G_{uu} + |H_2|^2 G_{vv}\right)\left(|H_{p1}|^2 G_{uu} + |H_{p2}|^2 G_{vv}\right)}
\end{aligned}
\qquad \textbf{(5.8)}
$$

It should be emphasized that each of the components of Equations 5.8 are functions of frequency but are not explicitly expressed as such in order to simplify the expressions. So

the ordinary coherence of the simple summation is a function of each of the transfer functions as well as the signal powers and cross-correlations.

Using the same methodology, expressions for the coherence of the COP filtered sum can be expressed in terms of the COP filters themselves. First consider the general expressions shown in Equations 5.9

$$\begin{aligned}
\gamma_m^2 &= \frac{|G_m|^2}{G_{rr} G_{nn}} \\
G_m &= G_{r_1 n} + G_{r_2 n} \\
&= (COP_1 H_1)^* G_{un} + (COP_2 H_2)^* G_{vn} \\
G_{rr} &= G_{r_1 r_1} + G_{r_2 r_2} \\
&= |COP_1 H_1|^2 G_{uu} + |COP_2 H_2|^2 G_{vv}
\end{aligned} \tag{5.9}$$

The new reference signals are written as functions of the originating signals in terms of the COP filters and the electrical and physical transfer functions which still exist. (Refer to Figure 5.35 to clarify signal definitions).

Recall that for the special case of uncorrelated originating signals, the coherent output power of each signal need not remove the effects of the opposing signal. In other words, the *partial* coherent output power is not necessary for the case of uncorrelated signals and the multiple coherence is defined by the sum of the *ordinary* coherences. In the same manner, the multiple coherent output power is defined as in Equation 4.61 without removing the effects of signal one from signal two. Therefore, by definition of the coherent output power filter design, the COP filters for this special case are the ordinary coherent output powers as shown in Equations 5.10.

$$COP_1 = \gamma_{x_1 n}^2 G_{nn} = \frac{\left|G_{x_1 n}\right|^2}{G_{x_1 x_1}}$$
$$COP_2 = \gamma_{x_2 n}^2 G_{nn} = \frac{\left|G_{x_2 n}\right|^2}{G_{x_2 x_2}} \qquad \textbf{(5.10)}$$

Substituting the expressions in Equations 5.9 and 5.10 into the definition of the ordinary coherence, it can then be rewritten as in Equation 5.11.

$$\gamma_m^2 = \frac{\left|\left(\frac{\left|G_{x_1 n}\right|^2}{G_{x_1 x_1}} H_1\right)^* G_{un} + \left(\frac{\left|G_{x_2 n}\right|^2}{G_{x_2 x_2}} H_2\right)^* G_{vn}\right|^2}{\left(\left|\frac{\left|G_{x_1 n}\right|^2}{G_{x_1 x_1}} H_1\right|^2 G_{uu} + \left|\frac{\left|G_{x_2 n}\right|^2}{G_{x_2 x_2}} H_2\right|^2 G_{vv}\right)\left(\left|H_{p1}\right|^2 G_{uu} + \left|H_{p2}\right|^2 G_{vv}\right)} \qquad \textbf{(5.11)}$$

To define the relationship between the simple reference summation and the COP filter summation, Equations 5.8 and 5.11 must each have a minimum identified and compared with each other for all possible values of H_1, H_2, H_{p1}, H_{p2}, G_{uu}, and G_{vv}. Solving this minimization problem is beyond the scope of this work but a numerical comparison has been developed in order to present a small number of the multitude of possibilities for the solution to these expressions.

Consider both Equations 5.8 and 5.11 existing as a function of frequency. Once the comparison between the two coherence values is made at a single frequency bin, it can be evaluated for all the bins in order to determine an overall improvement or deterioration as a result of using the COP filters. Therefore the numerical relationships will assume that all the values are known for a single bin and the coherences can be evaluated accordingly.

For the first case, consider H_{p1}, H_{p2}, G_{uu}, and G_{vv} are all set equal to unity while H_1 and H_2 are each varied from 0 to 100. Figure 5.36 shows the difference in coherence between the

simple summation and the COP filtered summation. Therefore, negative z-axis values indicate that using the COP filtering procedure reduces coherence over that of a simple summation while positive values indicate an improvement.

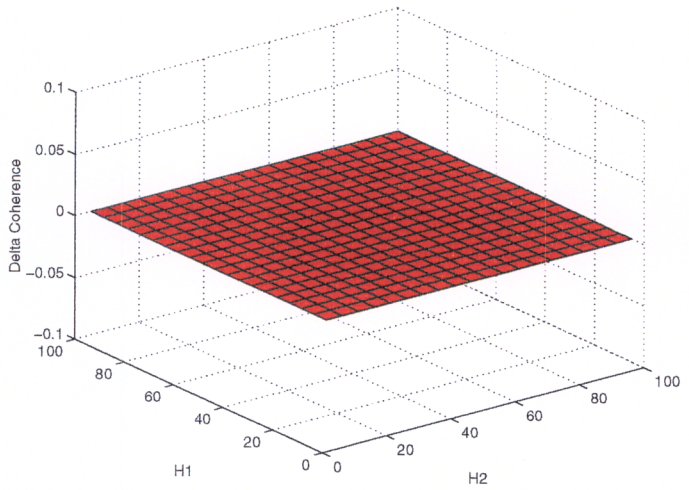


FIGURE 5.36 COMPARISON SURFACE FOR ELECTRONIC TRANSFER FUNCTIONS

For fixed values of the originating signal power and physical plant transfer functions, it is evident that variations in the electrical transfer functions alone do not result in any changes in the two summation coherences. Therefore, the COP filtering procedure provides no improvement (or hindrance) to the coherence function for this special case.

Continuing with the same analysis, consider variations in the originating signal powers G_{uu} and G_{vv} while the four transfer functions are set to unity. Figure 5.37 illustrates the difference in coherences for this special case.

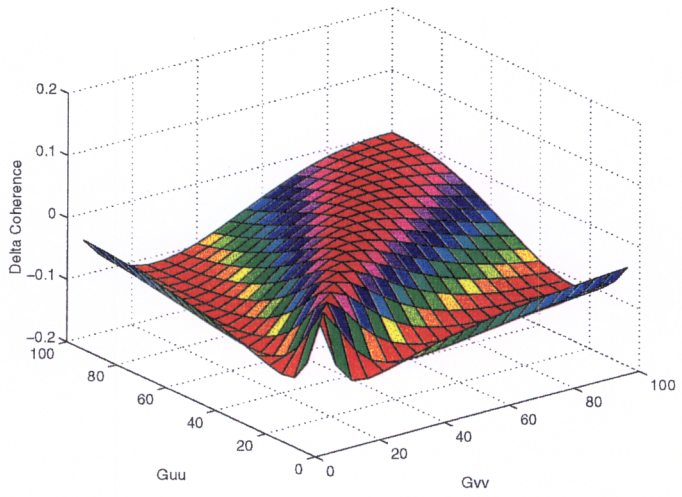


FIGURE 5.37 COMPARISON SURFACE FOR ORIGINATING SIGNAL POWERS

This case demonstrates that for some specific values of G_{uu} and G_{vv} the coherence of the COP filtered sum is *less* than that of the simple summation. This is an undesirable result in terms of the COP filtering procedure. However, the reduction is quite small and the effects on control are likely to be equally small. It is also unlikely that these specific cases will occur in practical applications but verification of that must remain for future work.

The final specific case for numerical comparison involves variations in the physical plant transfer functions while the electronic and originating signals are unity. Figure 5.38 illustrates this particular case.

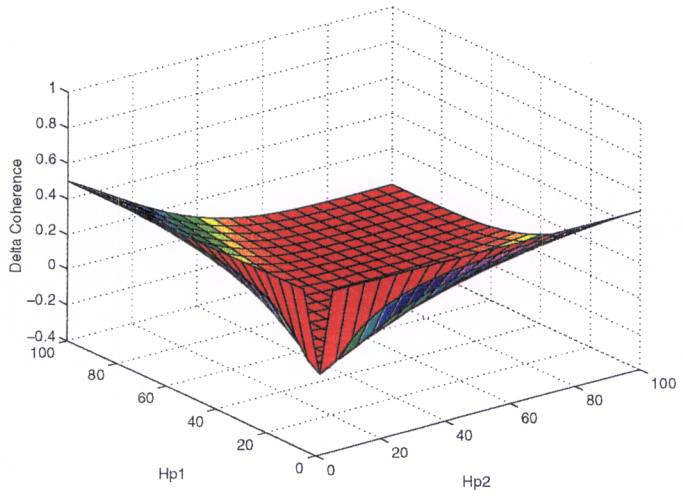


FIGURE 5.38 COMPARISON SURFACE FOR PHYSICAL PLANT VARIATIONS

It is now evident under which cases the COP filtering procedure can drastically improve performance. Variations in the physical plant transfer functions are far more common and are equally difficult to determine. The COP filter summation is shown to increase the overall coherence for nearly all variations in H_{p1} and H_{p2} from unity.

While these comparisons are educational they are far from all inclusive and cannot fully describe the improvements achieved by the COP filtering procedures. Either a mathematical solution or a method to plot (for all possible values) seven independent degrees of freedom is required.

In an attempt to further generalize this comparison, reconsider Figure 5.35. If G_{uu} and G_{vv} were set to unity, variations in both the electrical and physical transfer functions could take into account possible variations in the originating sources. In other words, any value G_{uu} could take on, can be emulated at the measured input and measured output by modifying H_1 and/or H_{p1} appropriately. Therefore it is possible to eliminate that particular degree of freedom by assuming each of the originating signals are unity and varying both the electronic and physical transfer functions accordingly.

Remaining in the comparison of the two ordinary coherence functions, are the four transfer functions, leaving the need for four independent axes. To investigate variations in all of these transfer functions for illustration on a single plot requires examination of the *difference* in the functions rather than the actual values they themselves assume. Suppose that H_1 takes on a constant value of 1250 and H_{p1} assumes a constant value of 500, each for a single frequency bin as before. Varying H_2 between 0 and 3000 provides a difference between H_2 and H_1 which varies between 1250 and -1750. This delta vector is adopted as the x-axis labeled "delta H's" in Figure 5.39. In the same manner, H_{p2} is varied from 0 to 3000 to provide a y-axis ranging from -2500 to 500. As before, the z-axis represents the difference in the simple summation and the COP filtered summation.

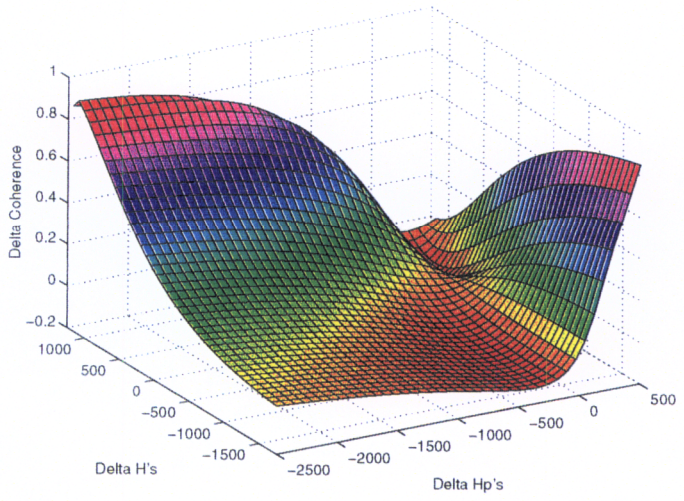


FIGURE 5.39 DELTA COMPARISON SURFACE FOR TRANSFER FUNCTION VARIATIONS

The particular range of values was chosen in order to illustrate the worst performance achievable by trial and error. The two local minima are representative points indicating the COP filter summation *reduces* performance over the simple summation. However, in the vast majority of the cases the COP filtering procedure *improves* performance over the simple summation, in some cases drastically.

To achieve a better vantage point of the surface shown in Figure 5.39, considered the rotated version in Figure 5.40.

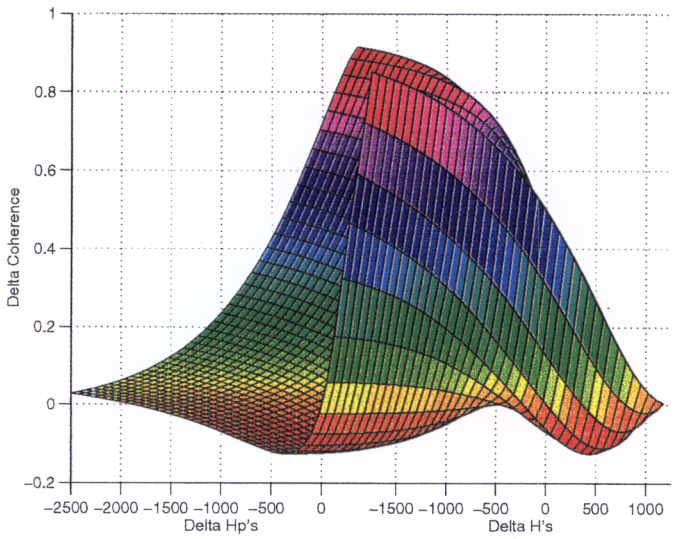


FIGURE 5.40 DELTA COMPARISON SURFACE FOR TRANSFER FUNCTIONS - SIDE VIEW

The two minima do not fall below a negative difference of 0.1. For the improvements achieved in all of the other possible cases, the COP filtering procedure is considered to drastically improve performance over the simple summation despite several small transgressions.

It should be emphasized that only the special case of two uncorrelated inputs was examined here with respect to numerical results. It is likely that consideration of a cross path will indicate further improvements in performance over that of the simple reference summation. It should also be noted that the COP filtering procedure itself provides a method of system identification that is not available using a simple reference summation. The COP filters can be examined following their design in order to indicate which input is contributing what power to the overall output.

Automating the procedure of MISO system identification is considered to be a valuable by product of the original intent of the COP filtering procedure. Not only will the engineer be allowed to examine the contribution of candidate references, but will also be alerted to cases where a poor reference signal has been chosen and should be discarded. The combination of online system identification with the development of a single, coherent reference signal

for SISO adaptive control, are also advantages which remain valid in the context of the conventional adaptive array.

5.5 COMPARISON WITH CONVENTIONAL ADAPTIVE ARRAYS

The adaptive array was originally developed in order to control an array of antennas adapted to track either a jamming signal or desired transmission in order to obtain the desired transmission. This (as has previously been noted) constitutes a different type of structure than the one which is applied to active noise control. Typically ANC problems have the ability to obtain a set of reference signals without approximating them from the error signal. The performance of the adaptive array structure designed with respect to ANC is now compared with performance of the COP filter procedure under identical conditions.

5.5.1 SIMULATION COMPARISON

Returning to the simple two-input single-output case of the automobile noise, an adaptive algorithm can be formulated as described in Section 3.4. As a matter of convenience, the adaptive array structure in terms of the automobile problem, has been recreated in Figure 5.41.

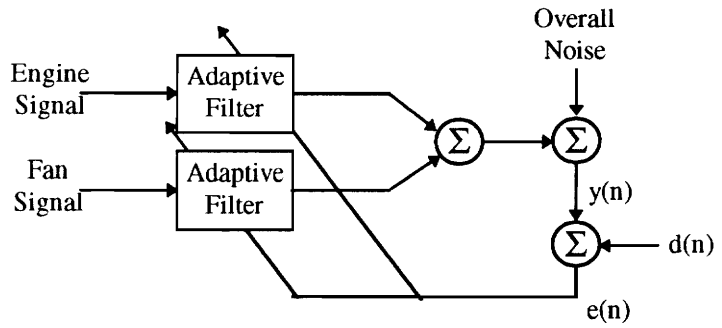


FIGURE 5.41 ADAPTIVE ARRAY FOR AUTOMOBILE NOISE EXAMPLE

Each of the adaptive filters in Figure 5.41 consists of an LMS algorithm as described in Chapter 4. The adaptive filter size is 128 and each has a fixed convergence rate of 0.0008. The time domain LMS algorithm was used to simplify the problem. However, identical results will be achieved by using the fully constrained frequency domain LMS algorithm.

The structure used for the COP filtering procedure is identical to that which is used in Figure 5.16. The two COP filters are designed prior to control and used to filter each of the reference signals. The apriori design of the COP filters does not present an implementation problem when taken in the context of the stationary ergodic assumption. The most favorable reference is determined based on the available signals and subsequently used for control. The cases where the reference and error signals vary with time are not considered.

The filtered-X version of the LMS, although not explicitly included in any of the following simulations, demonstrates a distinct computational advantage which the COP filtering approach maintains over the adaptive array. In practice, Figure 5.41 is not entirely accurate with respect to physical phenomenon. Because each of the reference signals used in each adaptive controller is utilized separately, they must each be separately filtered by the control to error path estimation as described in Chapter 3. The COP filtering method however, only requires a single reference to perform control and thus only a single filtering operation of the reference signal. Depending on the number of candidate references, the computational savings can be quite significant. For a filter size of N, the adaptive array structure has aN more computations resulting from the filtered reference signal alone (where a is the number of reference signals used exceeding one).

In order to ensure a "level playing field" for each case, identical signals were used for both the adaptive array and the COP filtering procedure. In addition, the adaptive controller used, following the COP filtered reference generation was designed using the same parameters as the adaptive array filters. Figures 5.42 and 5.43 show the error signal with respect to time (samples) for the adaptive array and the COP filtered reference. Even though the adaptive array has a smaller initial error, the converged values of both methods indicate that the COP filtering procedure has a lower overall error.

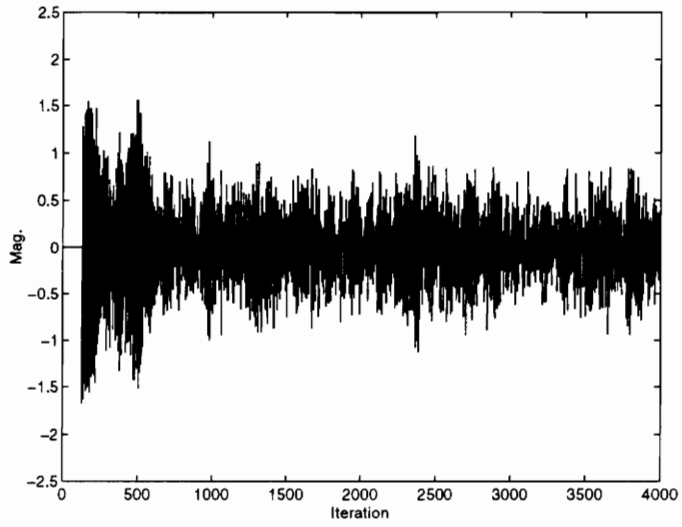


FIGURE 5.42 CONVERGED ADAPTIVE ARRAY ERROR SIGNAL

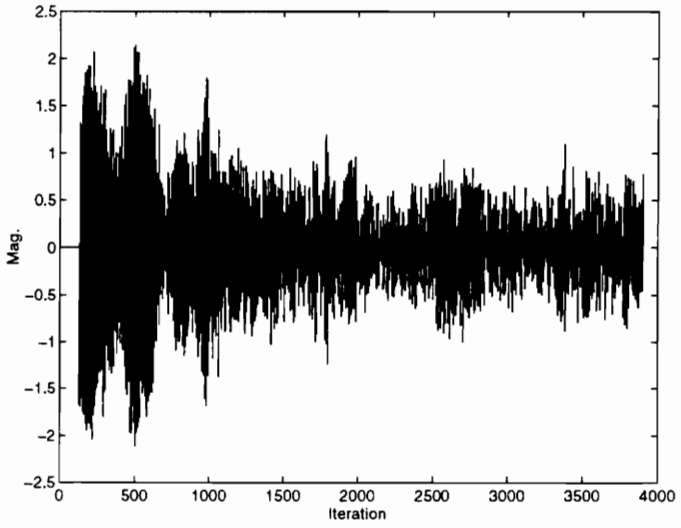


FIGURE 5.43 CONVERGED COP FILTERED ERROR SIGNAL

Postulations concerning why the error signals are slightly different will be addressed momentarily. To accurately compare the two methods, the most intuitive means lies in the examination of the controlled version of the plant (the error microphone signal) as related to the uncontrolled levels. Figures 5.44 and 5.45 compare the controlled and uncontrolled versions of each of the two methods.

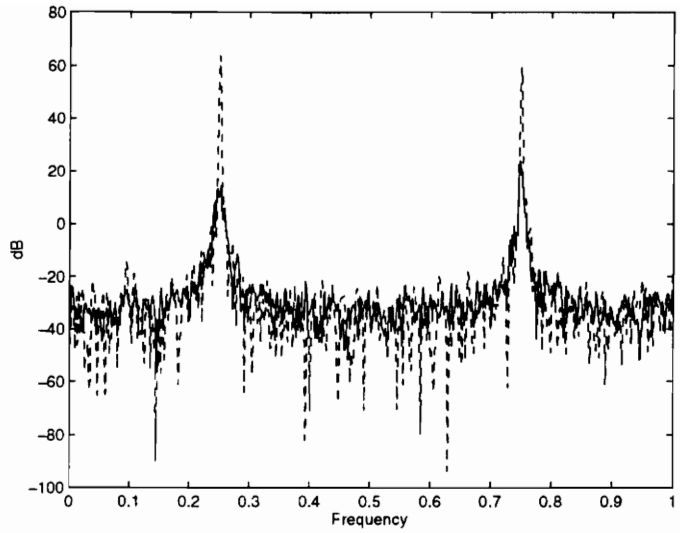


FIGURE 5.44 CONTROLLED AND UNCONTROLLED ADAPTIVE ARRAY

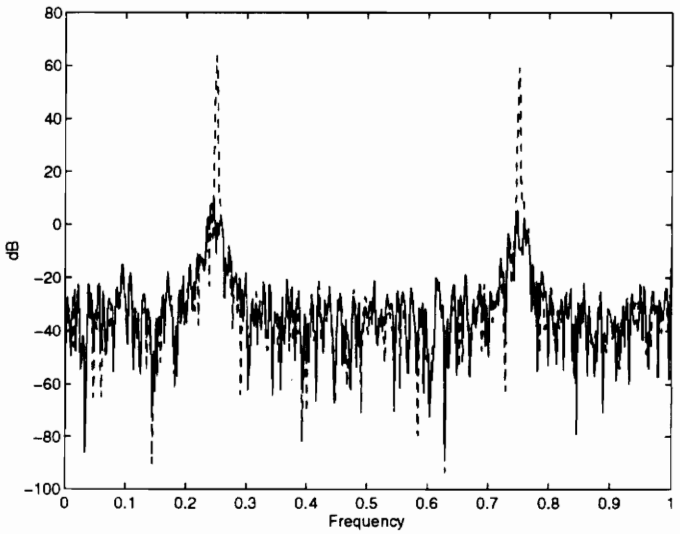


FIGURE 5.45 CONTROLLED AND UNCONTROLLED COP FILTERING

Figures 5.44 and 5.45 indicate that the COP filtering procedure improves performance over that of the adaptive array for this specific case. Figures 5.42 and 5.43 indicate that each filter has adequately converged to the result as delegated by the LMS algorithm. Judging by the unchanging error, 4000 iterations were deemed adequate for convergence. It should be emphasized that this particular example can be considered as a special case. As was seen for

the comparison of the COP filtering procedure with the simple summation, an infinite number of cases can exist and it is also likely that a scenario can be invented (or discovered) in which the adaptive array out performs the COP filter. It is predicted, however, that based on the expected value design of the COP filters and the expected value design of the optimal Wiener weights for the adaptive filter, a non-stationary case will easily satisfy this scenario.

The adaptive filters in the adaptive array are designed based on the LMS algorithm. As seen in Chapter 3, the optimal solution for the LMS adaptive filter converges to the Wiener weights. The inverse of the autocorrelation matrix times the cross-correlation matrix define the optimal weights for a given system. Each of these values are based on the expected value of an infinite amount of data. Anything short of that provides only an approximation of the actual filter weights. As was seen in Chapter 4 and the beginning of this chapter, the COP filters are designed using the same basic supposition. It therefore stands to reason that given enough data, the two should converge to the same filtering solution.

Elliot and Nelson in [2] provide a new metric for evaluating the performance of an adaptive array as opposed to the ordinary coherence. They state that the *multiple* coherence of the input reference signals to the disturbance defines the performance of the adaptive array. The multiple coherence was defined in Section 4.3 and studied in the context of the energy source identification problem in Section 4.4. Solving the coherent output power problem for a set of given inputs results in a single signal which has the same value of ordinary coherence to the disturbance as the numerous inputs have as the multiple coherence. Therefore the adaptive array and the COP filtering procedure should ideally converge to the same solution for stationary ergodic signals. As is seen in the next section this is not always the case, possibly due to the existence of local minima in the adaptive array performance surface.

5.5.2 TRANSFER FUNCTION COMPARISON

Given the optimal results of each of the two methods, results can be examined regarding the converged filter designs. Figure 5.46 shows the converged versions of each of the two methods where the adaptive filters have ceased updating because the error has reached a minimum value.

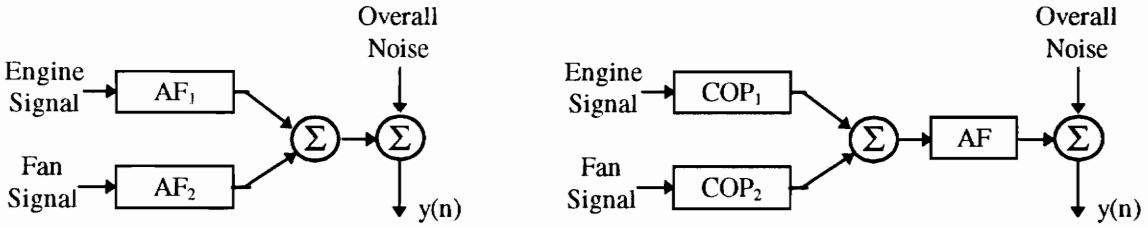


FIGURE 5.46 CONVERGED ADAPTIVE ARRAY AND COP FILTERING BLOCK DIAGRAMS

Continuing with the two tonal example discussed in the previous section, it can be assumed that based on the error signals of each control scheme, that all adaptive filters have adequately converged by iteration 3000. Given that the filters are now stationary, linear system rules can apply and simple block diagram algebra can be used to relate the two systems. Rearranging the COP filtered version of Figure 5.46 to emulate the adaptive array, Equations 5.12 should be obvious.

$$\begin{aligned} AF_1(\omega) &= COP_1(\omega)AF(\omega) \\ AF_2(\omega) &= COP_2(\omega)AF(\omega) \end{aligned} \qquad \textbf{(5.12)}$$

These relations should hold provided that each adaptive filter converges successfully to the optimal solution where no noise is present and y(n) is minimized. In effect, since the inputs to the two systems are identical and the overall noise to be canceled is identical, the end result must also be identical. For the converged adaptive filter solution, the adaptive array and the COP filtering technique should become equivalent.

As was seen in the previous section, this may not always occur. First compare the magnitude and phase of the first adaptive filter in the adaptive array to the magnitude and phase of the first COP filter time the transfer function of the SISO adaptive filter.

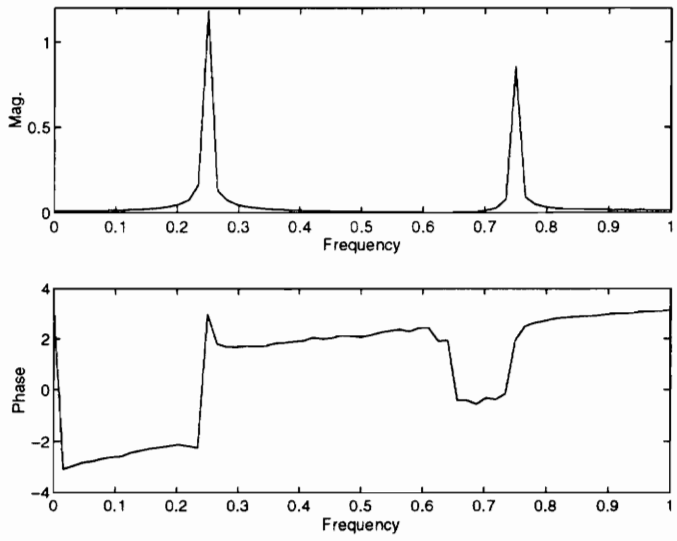


FIGURE 5.47 CONVERGED ADAPTIVE ARRAY INPUT 1 TRANSFER FUNCTION

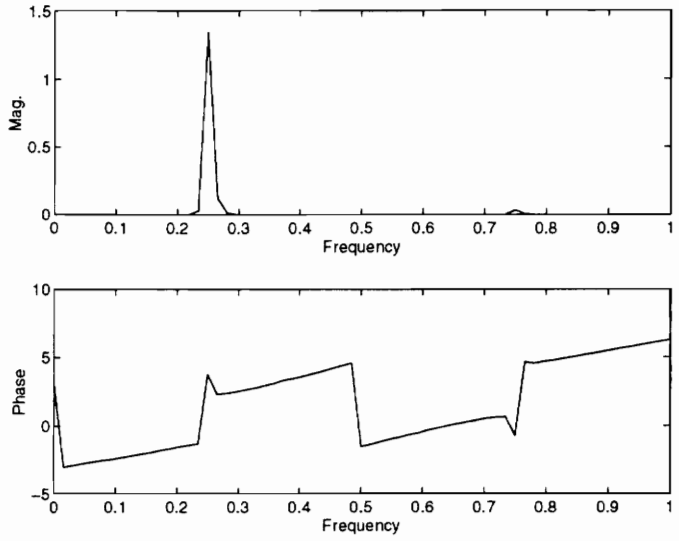


FIGURE 5.48 CONVERGED COP FILTERING INPUT 1 TOTAL TRANSFER FUNCTION

The result in Figure 5.47 is not surprising since the engine input has power in both tonal bins even though only one is coherent. Figure 5.48 shows that the COP filter was able to first separate the coherent portion of the engine before using the adaptive filter to apply control. This extra degree of freedom seems to provide a fail-safe against local minima which may be encountered by the adaptive array. The adaptive array filter has the task of both control *and* separating the coherent portion of the spectrum. These may become competing goals in certain situations and local (rather than global) minima may be encountered on the error surface(s).

The indication of Figures 5.47 and 5.48 illustrate that in both magnitude and phase of the (first tonal) controlling bins, the adaptive array and COP filtering procedure approach one another. Detailed analytical analysis of the true relationship between the two is beyond the scope of this work. One notable difference in the two magnitude plots of the transfer functions lies in the ability to perform a real time system identification based on the transfer functions of the converged filters. Before explaining why the adaptive array cannot offer this feature, consider the converged version of each of the second (fan) filters as shown in Figure 5.49 and 5.50.

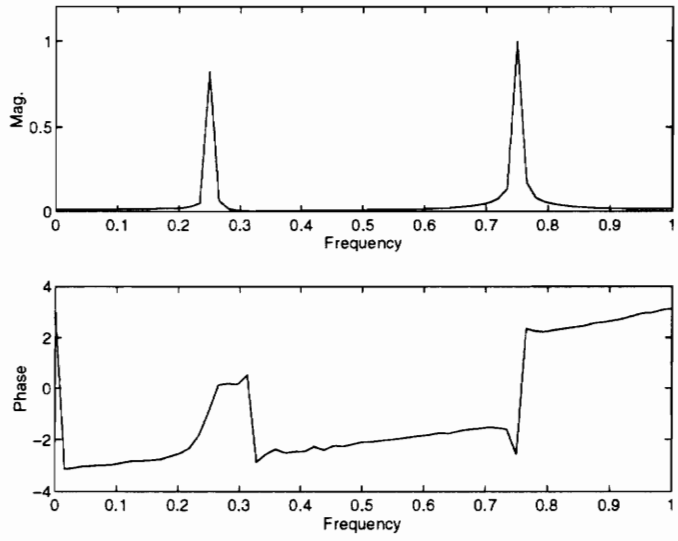


FIGURE 5.49 CONVERGED ADAPTIVE ARRAY INPUT 2 TRANSFER FUNCTION

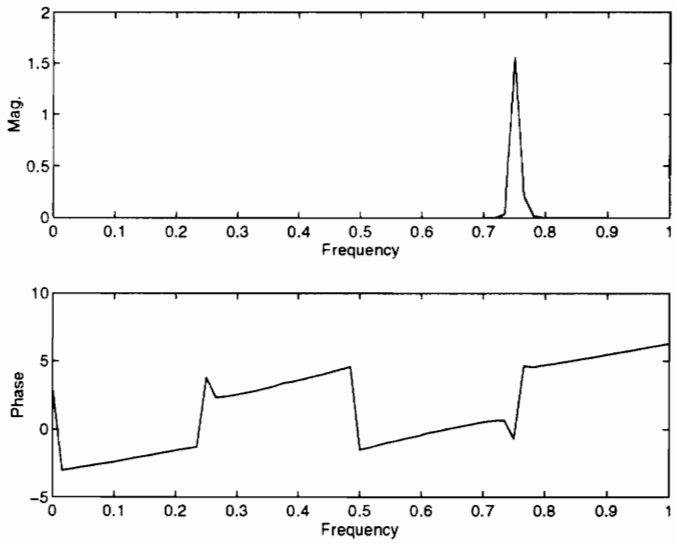


FIGURE 5.50 CONVERGED COP FILTERING INPUT 2 TOTAL TRANSFER FUNCTION

Independent of the discussion of local or global minima, the adaptive array is able to exercise notable control over the original noise as seen in Figure 5.44. The reason it is able to do this in spite of the fact that there is opposing incoherent noise in the second reference (at either tonal), is that the SNR requirement is satisfied by the adaptive filter.

Recall that there is a definable relationship between the SNR and the coherence of a signal. Once this exceeds a certain point (typically 6 dB or 0.8 coherence) a notable level of control can be achieved. The adaptive array's only goal is to minimize the error signal and if that can be achieved by a very small SNR then no other reinforcement is necessary. From a control perspective this is a satisfactory answer. However, observing the converged adaptive filter's weights will often provide an incorrect assessment of the individual reference's contributions to the overall output. These facts are evidenced by the four preceding figures.

While an extensive examination of the adaptive array versus the COP filtering technique has not been pursued, it is evident that they can achieve the same goal via different means. After the initial design of the COP filters, the two methods demonstrate a discrepancy from a computational standpoint. M adaptive filters in the adaptive array requires $M2N^2$ multiplication's to produce N output samples. However, the filtered reference must be included in any practical noise control implementation; this adds N^2 multiplication's for N output samples and each filter, or MN^2 total computations. The COP filtering technique requires MN^2 computations for candidate reference filtering and $2N^2 + N^2$ multiplication's for the filtered reference LMS algorithm. When compared, the adaptive array requires $\mathbf{3MN^2}$ multiplication's and the COP method requires $\mathbf{(3 + M)N^2}$ multiplication's. Therefore as long as M exceeds unity (which it always will for the non-trivial case), the COP filtering method is computationally advantageous.

An added benefit of the COP procedure is seen in the accurate system identification of each of the individual contributions of the references to the output. It is not evident that the adaptive array could even identify a reference that had no contribution to the output unless its adaptive filter were examined with respect to the other for each frequency bin. However, what is gained in stationary system identification is lost in system adaptability. If the noise field to be canceled is continually non-stationary and changes in the reference signal reflect that, the COP filters will become outdated and cease to provide a useful reference. However, the same comparison can be made by examining the filtered-X LMS. If the control to error path were to change during control, the system would become

unstable as the path estimation becomes increasingly outdated. Physical and environmental limitations will always plague any new design and certain performance assumptions will always be necessary.

Evaluation of performance of such non-stationary systems can no longer rely on the performance metric developed in Equation 5.4. The coherence function itself relies on the stationary ergodic assumption as does Wiener filter theory. The advantage of the fully adaptive arrangement is that it can redesign itself at a moderate rate for various changes in the plant. Keeping with the previous example of controlling two tonals, Chapter 6 now uses the stationary ergodic assumption to demonstrate the practical application of the COP filtering procedure in a controlled laboratory setting.

Chapter 6

Experimental Application of Results

6.1 OVERVIEW

Practical implementation of any simulated design almost always behaves differently than originally expected. Computer simulations must become far too complex to accurately represent all the physical and environmental variables which can possibly plague an experiment. Therefore unless a design is actually implemented and shown to be effective in a practical sense, questions of its validity will always arise.

This chapter discusses all of the hardware and software used, as well as the results obtained, in demonstrating the COP filtering technology as applied to a practical, real-world system. The example which has been carried through for the two-input case is slightly modified and generalized for implementation in a laboratory environment. Since it has been shown that the adaptive array performs at least similarly to the COP filtering design, the experimental comparison of these two methods was not performed. Focus remained on the comparison of the COP filtering method to that of the simple reference summation since the most apparent gains can be achieved by this design alternative.

The facilities used to demonstrate the COP filtering design were equipped with several previously designed, optimized pieces of DSP code. These included a single channel filtered-X LMS algorithm as well as some efficient FFT and FIR filter functions. Jeffrey Vipperman is responsible for writing the filtered-X LMS code used in performing the control portion of the COP filtering procedure while Ensigma Ltd. provided an assembly code, optimized DSP library containing the FFT and FIR filter functions used within the COP filtering code. The real time COP filtering design and implementation code is considered as purely original work.

6.2 EXPERIMENT DESIGN

6.2.1 EXPERIMENTAL SETUP

As mentioned earlier, the design of the experiment follows closely with the automobile example carried throughout the previous chapters. The overall setup was constructed as a two-input single-output COP filtered design using a SISO filtered-X LMS controller. The overview of this system layout is shown in Figure 6.1.

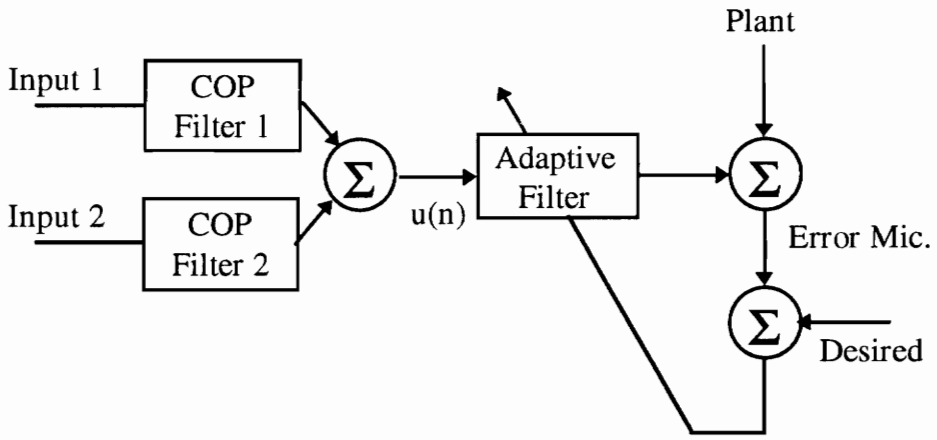


FIGURE 6.1 OVERALL EXPERIMENTAL DESIGN

This is similar to the system shown in Figure 5.16 with the exception of the illustration of the originating signal paths. In a laboratory setting the signals are typically created from signal generators. The luxury in this approach is found in the ability to develop signals which are coherent and incoherent at specific peaks. Knowing the benefits of the COP filtering technology, signals can be created which demonstrate the most useful implementation of the design. Demonstrating the design with user developed signals is sufficient in order to show that the COP filters will also work in a more practical application such as an automobile.

In order to create signals which were both realistic in light of practical applications and effectively demonstrated the benefits of COP filtering, four separate signal generators were used to create four types of signals. Choosing a sample rate of 2000 Hz for the entire

system, efforts were made to create signals which were similar to those used in the MATLAB simulations. Most effective active noise control occurs at frequencies below 1000 Hz. Doubling this frequency to give a sampling frequency of 2000 Hz allows the algorithm(s) to accurately resolve frequencies up to 1000 Hz.

"Input 1", "Input 2", "Plant", and the simple reference summation were the primary signals which needed to be artificially generated. Figure 6.2 illustrates the process used to create each of these signals via operational amplifier summing circuits.

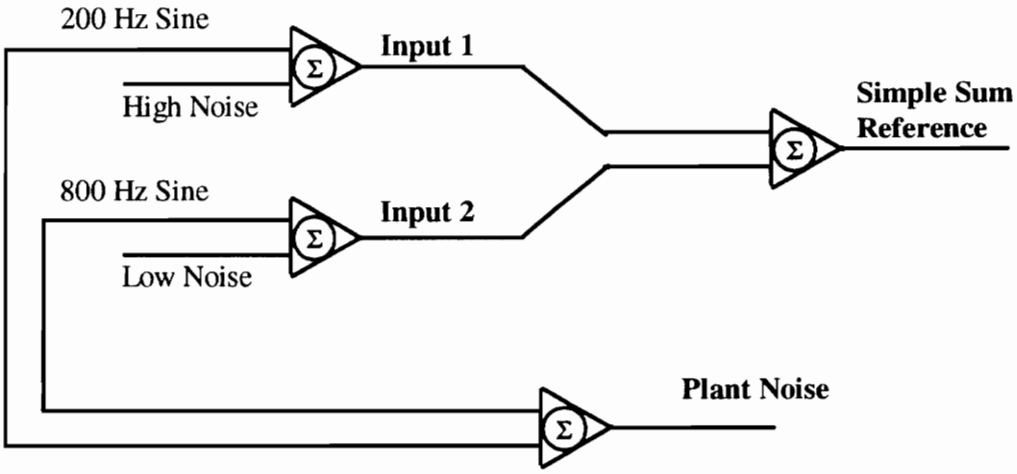


FIGURE 6.2 ARTIFICIAL SIGNAL GENERATION

Beginning with the plant noise it is seen that the combination of the 200 Hz sine wave and the 800 Hz sine wave will produce a two tonal complex. This is the actual sound which will be heard (and eventually canceled). This signal was taken from the op-amp summation and amplified to drive what was termed the "disturbance speaker". Since no practical piece of equipment was used (such as an automobile), the noise to be canceled was generated from an audio speaker. To monitor this noise, a Bruel and Kjaer (B&K) microphone and Hewlett Packard signal analyzer were used. Inserting the appropriate microphone scaling factor for conversion to sound pressure level, data collected from the error microphone signal is shown in Figure 6.3.

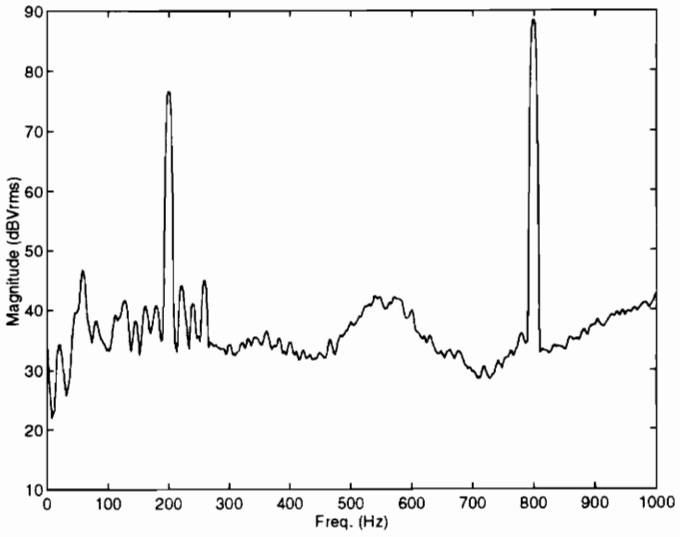


FIGURE 6.3 ERROR MICROPHONE POWER SPECTRUM

To summarize the hardware involved in creating the path from the plant signal to the signal analyzer, consider Figure 6.4.

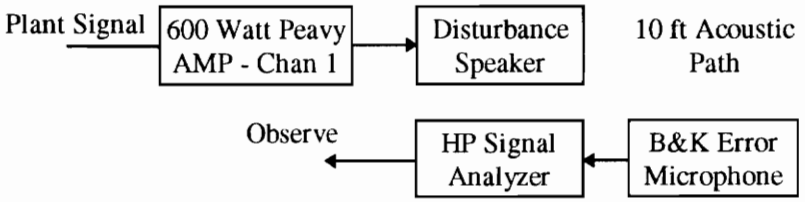


FIGURE 6.4 PLANT SIGNAL PATH

The B&K microphone is referred to as the error microphone in Figure 6.4 because ultimately it is used for the error signal in control of the plant. As will later be seen, it is also used for the COP filter design as the "output" signal and the error signal in identification of the control to error path for the filtered-X LMS control.

Continuing on with the generation of each of the reference inputs, input 1 represent the summation of a 200 Hz tonal and what is termed "high noise". The high noise is essentially white noise high pass filtered so as not to significantly overlap the tonal. Special care was taken in the adjustment of signal levels such that the noise had a higher amplitude in its

frequency range than the tonal had in the opposing reference signal. This was done in order to emphasize (and generalize) for the case where the coherent portion of the reference signal (tonal) had much less contribution to the overall bandpower than the incoherent noise. Figure 6.5 shows the power spectrum of the first reference signal, also known as "Input 1".

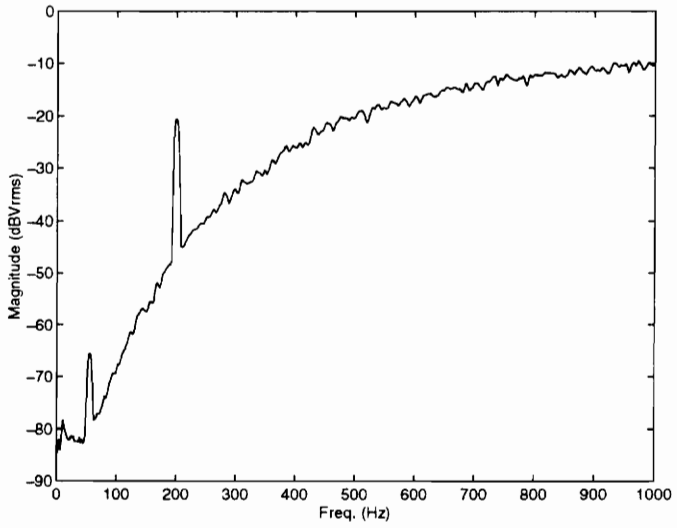


FIGURE 6.5 REFERENCE INPUT 1 POWER SPECTRUM

The relative levels of the tonal and the noise are more significant than the overall signal levels shown in Figure 6.5. The overall signal's peak amplitude was slightly under 10 Volts and the sine wave was set at approximately 0.2 Volts peak amplitude. Figure 6.5 is meant to illustrate the shape and relative levels rather than absolute levels which are less important.

Reconsidering Figure 6.2, it should now be apparent that the coherence between "Input 1" and the plant should be high for the 200 Hz tone and low elsewhere. This corresponds to the type of signals which were developed for use in the simulations of previous chapters. Figure 6.6 shows the coherence between these two signals as calculated by the HP signal analyzer for 30 averages.

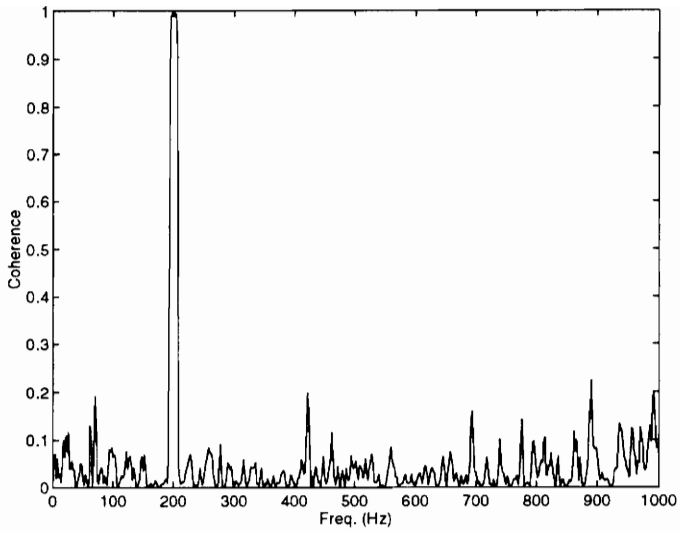


FIGURE 6.6 REFERENCE INPUT 1 TO PLANT COHERENCE

Continuing in the same manner, input 2 was developed using the same 800 Hz tonal used to generate the plant signal. For this reason a high coherence at 800 Hz is expected between the second reference signal and the plant but a low coherence is expected elsewhere. Combining the 800 Hz tonal with what is termed as low noise provides the same effect as seen in Figure 6.7, only reversed in the frequency spectrum. The low frequency noise was created using a white noise signal generator different from the one used to generate the high frequency noise for input signal 1. This was done in order to prevent unexpected coupling between the two signals. Recall the requirements developed for a well defined model. One of the criteria developed stated that the coherence between any two inputs should never equal unity. If this is the case, one of the inputs is unnecessary and should be removed. To avoid this discrepancy, separate white noise generators were used.

Figures 6.7 and 6.8 show the power spectrum and coherence of reference signal 2, respectively.

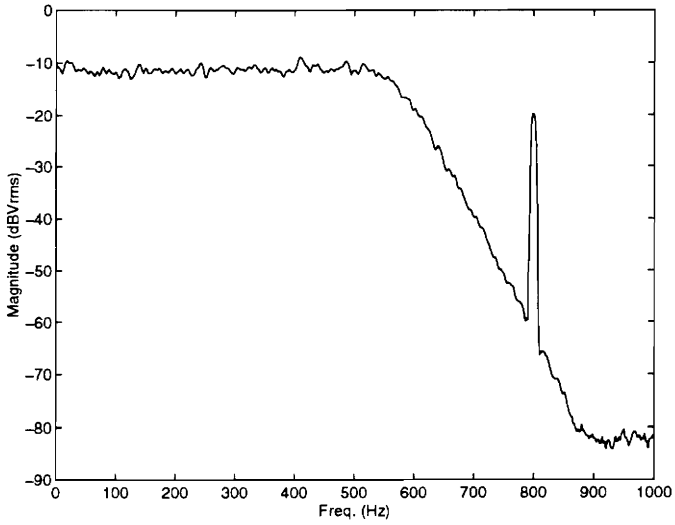


FIGURE 6.7 REFERENCE INPUT 2 POWER SPECTRUM

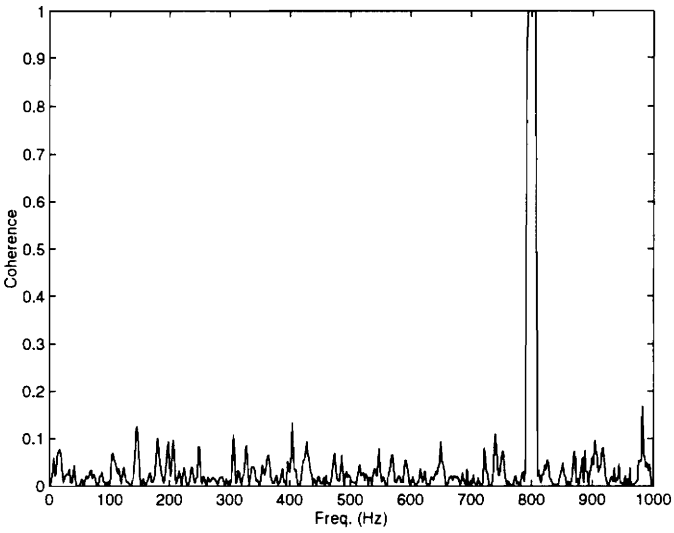


FIGURE 6.8 REFERENCE INPUT 2 TO PLANT COHERENCE

The levels of each of the signals used to generate the second reference signal are comparable to the levels used in the first reference. The noise signals opposing the tonals in corresponding frequency bins are approximately 10 dB greater than the tonal amplitudes.

The final signal which was generated via analog methods was the simple reference summation. This signal represents an alternative solution to the multiple reference problem which does not involve the added complexity of an adaptive array. As was mentioned previously, a single channel filtered-X LMS code currently exists in the laboratory where these experiments took place. Without completely rewriting a control code, the only solution to the multiple reference problem is to simply add the references. For the example which has been developed here, adding the first and second inputs (reference signals) results in the signal termed as the "simple summation". The power spectrum of this signal is shown in Figure 6.9.

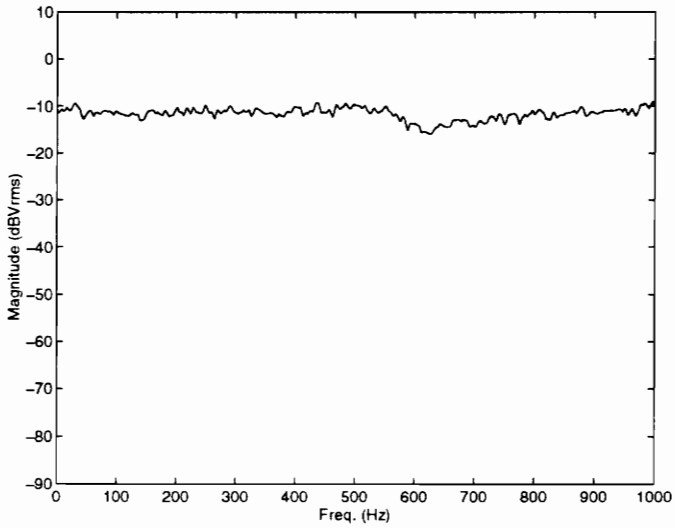


FIGURE 6.9 SIMPLE SUMMATION REFERENCE SIGNAL

After creating reference input 1 and reference input 2, the two signals were linearly combined to form the signal shown in Figure 6.9. As should have been expected based on the power spectrums of the two inputs, the summation shows no evidence of the two tonals due to the noise in each of the two opposing signals.

Having created all of the signals needed to perform the comparison between the COP filtering algorithm and the simple reference summation, the next steps include setting up the necessary hardware and software to run the experiment. Before proceeding with any

control analysis or results, a more detailed description of the hardware and software used in this experiment will be presented.

6.2.2 HARDWARE

As mentioned before, two separate digital signal processing systems were used in parallel real time operation to perform this experiment. This is of course not to say that it is not possible for a single DSP to perform all the computation necessary to implement the COP filtering procedures. Previously designed software made the control portion more feasible so that focus could be turned to developing the COP system on its own independent DSP. Reconsidering Figure 6.1, it is now necessary to illustrate in more detail, the hardware required in this experiment. Figure 6.10 shows a system level diagram of all of the hardware components used in the laboratory setup.

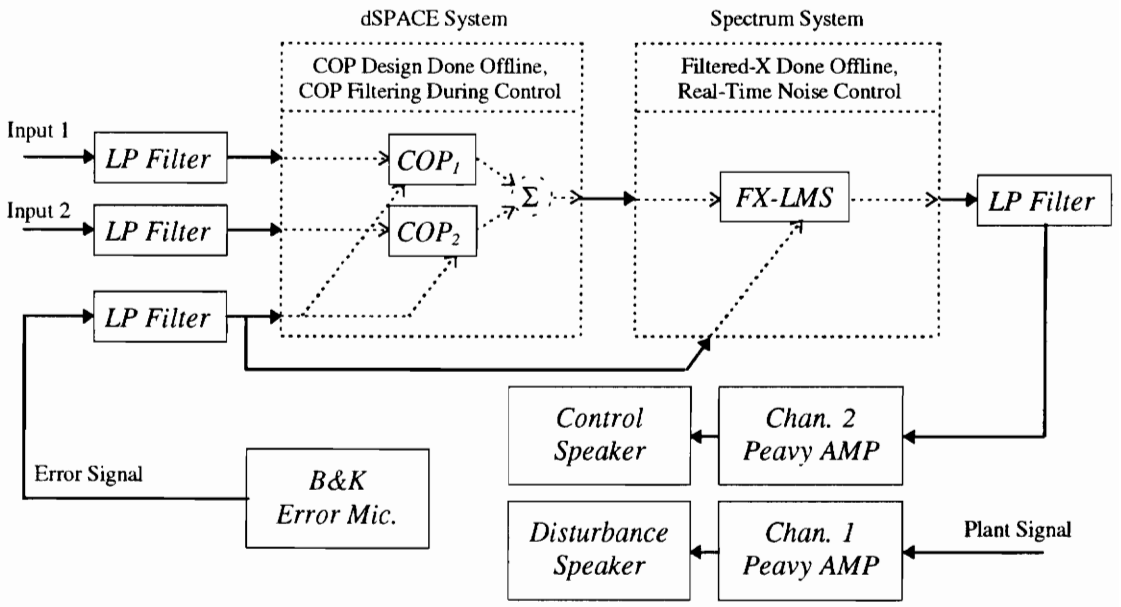


FIGURE 6.10 HARDWARE SETUP FOR COP EXPERIMENT

Each of the two DSP systems perform both an off-line and real time task. After off-line design of the COP filters, the dSPACE system applies the COP filtering procedure to each of the inputs and sums the result to create a coherent reference. The Spectrum DSP system must perform the filtered-X system identification off-line and the actual noise control is

performed in real time. (The dSPACE system is circumvented when the simple summation is used as the reference for the noise control system. The summation of the low pass filter outputs is fed directly into the Spectrum system).

The dSPACE system is a high end digital signal processing environment used primarily for prototyping of preliminary designs. The dSPACE system was installed on a 100 MHz Pentium PC which acts as a host to the DSP processor and a analog to digital (A/D) input output peripheral board. The DS1003 processor houses a TMS320C40 floating point DSP running at a 50 MHz clock rate with a 40 ns cycle time. There is 2 K x 32 bit on chip memory and 8 K x 32 bit true dual port memory for host to DSP communication. The card uses a single 16-bit PC/AT expansion slot on the host PC. Connected to the DSP is the DS2201 multiple input output board which also uses a 16-bit PC/AT slot. Its capabilities include 20 simultaneous sampling 12-bit analog to digital converter (ADC) channels and 8 double registered 12-bit digital to analog converter (DAC) channels.

The dSPACE system also includes several pieces of software which can be used in developing real time DSP applications. The Texas Instruments C-compiler must be used to generate on board executables from compiled C or Assembly language code. Also included in the dSPACE package are two programs which access dual port memory for real time interaction with on board codes. The Trace code acts as an oscilloscope and will plot, on screen, any global variable in real time. The Cockpit software allows real time alteration of global constants declared in the code effected by a user designed graphical user interface. Although these codes were used in the troubleshooting and development of the COP software, ultimately they were deemed unnecessary for the operation of the COP filters.

The Spectrum Signal Processing Inc. system was housed in a 486 PC operating at 33 MHz. The PC/C31 Real Time Application Board houses a TMS320C31 DSP processor at 33 MHz. The PC/16IO8 Multi-Channel Analog I/O board was also housed in the 486 PC. Its capabilities include 16 simultaneous sampling inputs and 8 DACs.

Each of the three inputs to the dSPACE system and the error signal input to the Spectrum system experience what are known as anti-aliasing filters. These low pass filters (denoted LP Filter in Figure 6.10) are required to ensure that no aliasing of the input signals occur upon sampling. It was previously mentioned that the sampling rate was set at 2000 Hz for each system causing the Nyquist frequency to become 1000 Hz. Each of these analog low pass filters were set with a cutoff frequency (-3 dB down point) of approximately 900 Hz. Since the filters used were 8 pole elliptical filters (by Frequency Devices), the roll off was sufficient enough to ensure only a very small amount of aliasing of the input signals directly surrounding the Nyquist frequency.

In addition to the three anti-alias filters, two other low pass filters were used in the system shown in Figure 6.10. The first of which (not shown) was a "smoothing filter" on the output of the dSPACE system. The DAC takes each individual output sample and performs a zero-order hold operation to give the signal analog continuity. The output then has frequency content due to the holding operation which should not be transmitted to the Spectrum system. An analog low pass filter set at the same cutoff frequency as the anti-alias filter removes the undesired high frequency content. Shown on the output of the Spectrum system is another low pass filter serving the same purpose of smoothing the zero order held output.

The only remaining hardware includes the power amplifier for the speakers and the speakers used to generate the acoustic noise. One channel of the Peavy amplifier was used to drive the control speaker while the other was used to generate the disturbance. The B&K microphone and associated amplifier were used for COP system identification, filtered reference system identification, and error signal during control. The HP analyzer (not pictured) was used to collect all data shown in this chapter.

6.2.3 CONTROL CODES

Two separate and specific pieces of control software were used throughout this experiment. The first, which was developed by Jeffrey Vipperman, was a single channel filtered-X LMS noise control algorithm running on the TI C31 system. The focus of this experiment

centers around the COP code developed for the TI C40 DSP processor. The COP code uses three non-original functions designed for the C30 and C40 DSP processors written in assembly language by Ensigma Limited.

Jeffrey Vipperman's code provides a DOS based user-interface which permits the engineer to proceed through a well defined course of events required for active noise control. First the convergence parameter (μ) is selected for the control to error path system identification necessary for performing the filtered-X algorithm. The system identification is done using the reference input as the signal to excite the modes for identification while the design is based on the LMS algorithm. Since the input signal cannot exceed a 10 Volt peak without saturating the ADC input, a convergence parameter of 0.01 is the fastest, stable value which can be used. Following system identification the user is again prompted to readjust the convergence parameter for control. Depending on the autocorrelation matrix of the input at the modes of desired control, the convergence parameter will be set arbitrarily small. The final step in the control process is the actual convergence of the LMS control code to a solution which minimizes the error. This is done as described in Chapter 3. Jeff's code is a *time domain* adaptive code.

The focus of this experiment is to analyze the effectiveness of the COP filtering procedure versus the simple reference summation on the overall control performance of an actual acoustic noise control system. The COP software developed to illustrate this is included in Appendix B. Many of the C functions are self explanatory and the code has been adequately commented. Three routines are used in the program which do not represent original work. A real FFT algorithm, a complex inverse FFT function, and an FIR filtering routine were designed by Ensigma Ltd. specifically for the C30 and C40 DSP processors. They are highly optimized assembly language codes which minimize instruction count and pipeline conflicts.

The general procedure in the COP filtering code begins with data collection. In order to design the filters, the size of the filter is selected and the number of averages used to calculate the coherent output powers is entered. For the experiment presented throughout

this chapter, a filter size of 256 with 41 averages was selected. This nearly fills all of the available RAM on the DSP. This particular operation could have been accomplished more efficiently by performing calculations on each ensemble and saving only the latest information and latest average. However, the code shown in Appendix B first collects all of the data from the "output" (or error signal), the first candidate reference, and the second candidate reference prior to any control. The second step in the COP filtering procedure is performed off-line.

The COP filters are designed based on the collected information and subsequently converted back to the equivalent time domain FIR filters. (Zero phase filters were assumed for simplicity). This conversion was performed in order to illustrate that the COP filters can be implemented in either the time or frequency domains. (The MATLAB simulations performed the filtering in the frequency domain). In addition, a highly efficient time domain filtering algorithm already existed via the Ensigma Ltd. C library.

After converting the COP filters back to the equivalent time domain FIR filters, the second of two interrupt service routines is used to filter each of the two inputs with the COP filters and add the results. The new reference is then sent out of a DAC channel to be used by the control code as the new COP filtered reference signal. The entire off-line processing procedure takes less than one second (as timed by observation). Each FFT operation for a 256 point FFT takes 0.306 ms while the IFFT operations take 0.549 ms. The FIR filtering takes approximately 12 μs for a 256 weight filter.

Examining each of the functions more carefully, it should be apparent as to what operations each of them perform. Beginning with *main*, the C40 board is first initialized and the interrupt service routine using timer "t0" is started. The *while* loop is only executed when the system state is satisfied; this occurs after all input samples have been collected. Once each of three input matrices have been populated, the function *MikesCOPDesign* is executed. What is returned are two vectors of the FIR filter weights of the COP filters. Finally, the second interrupt service routine is enabled. This uses the COP design to filter and add each of the inputs and sends the new reference to the DAC channel one. The functions

executed by *MikesCOPDesign* are self explanatory and will not be reviewed here. Following is a summary of the results obtained from the comparison of the simple summation to the COP filtered reference.

6.3 RESULTS

Having developed each of the necessary signals and all of the software, the effects of COP filtering can now be examined in terms of active noise control. Recall the simple summation shown in Figure 6.9. Due to the incoherent noise in each original candidate reference signal at the opposing bin locations, the summation of the two references indicates no tonal content from either of the two original reference signals. When examining the ordinary coherence between the simple sum reference signal and the plant disturbance noise, it is seen that the noise overcomes each of the two coherent peaks and results in a low coherence for all frequencies. This is shown in Figure 6.11.

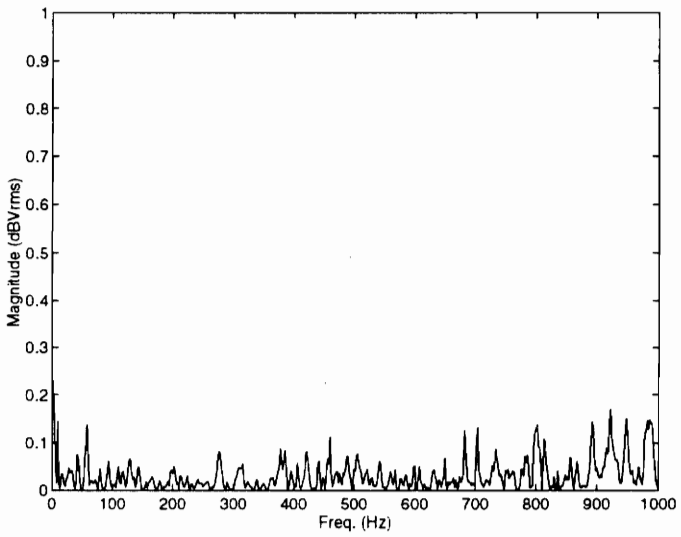


FIGURE 6.11 COHERENCE OF SIMPLE SUMMATION REFERENCE TO NOISE

According to the performance metric which has been established based on this particular coherence value, the performance of the adaptive controller should be unacceptable.

Using the linear summation from the op-amp circuit shown in Figure 6.2, the following procedures were then carried out. The system identification portion of the filtered-X LMS was performed using the simple sum reference signal and a convergence parameter of 0.01. Next, the convergence parameter for control was set to 0.0001 and an adaptive filter size of 128 weights was used. (This is the maximum filter size limitation of Jeff's code). Without performing any control, a power spectrum of the plant disturbance at the error microphone was obtained. Additionally, after convergence of the adaptive algorithm, using the simple reference sum, a second power spectrum at the error microphone was collected. These spectra are plotted together for comparison in Figure 6.12 where the uncontrolled plant is shown dotted.

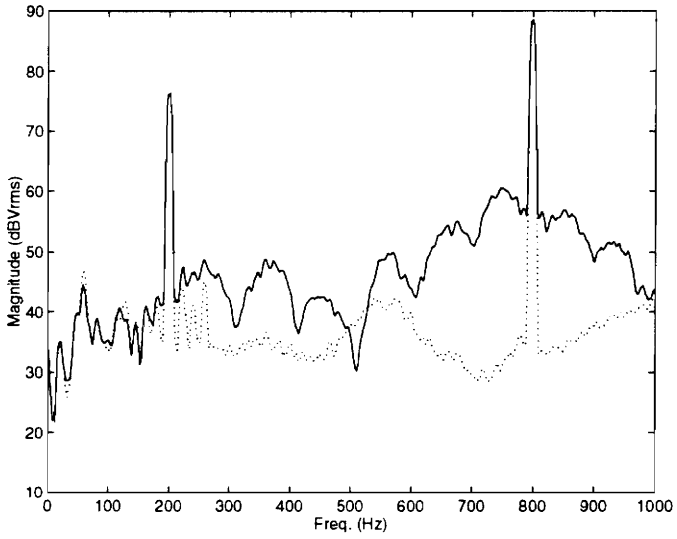


FIGURE 6.12 CONTROLLED AND UNCONTROLLED - SIMPLE SUM

The performance shown in Figure 6.12 is non-existent. No control authority of either of the tonals is exhibited and spillover of extraneous noise from the reference is quite significant. This case presents an excellent example of the frustration which may be encountered by the noise control engineer. Two independent signals, each with coherent components to the output, have been identified but cannot be used together in a SISO adaptive controller. A more generalized case of this problem has a correlation path between the two inputs. This is fairly difficult to generate in a laboratory situation, but

frequently occurs with real world systems. The COP filtering procedure is therefore under utilized for this particular problem, since the solution need not involve the partial coherence function.

Examining the COP filter design solution to this particular experiment, Figure 6.13 shows the power spectrum of the output of the dSPACE system, corresponding the newly created, COP filtered reference signal.

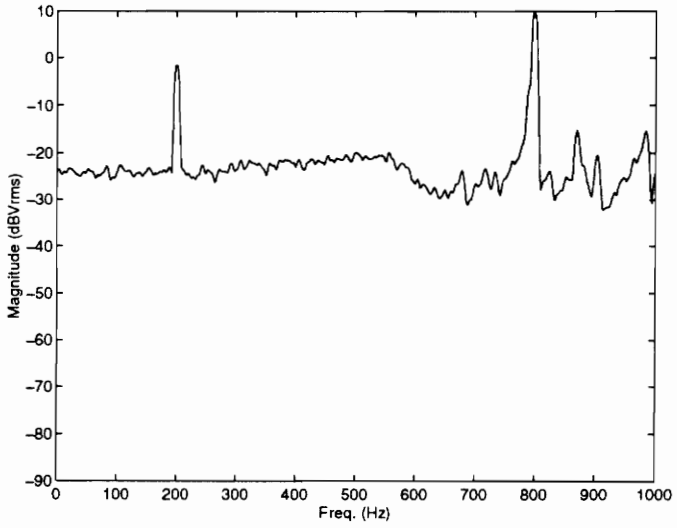


FIGURE 6.13 POWER SPECTRUM OF COP FILTERED REFERENCE SIGNAL

Examining the levels of each of the tonals now apparent in Figure 6.13, it is seen that they exceed the amplitude of the original tones by approximately 20 dB. In addition, the noise levels surrounding the tonals are reduced by approximately 15 dB. The COP filters have extracted the coherent portions of each signal and suppressed the incoherent portions so that upon linear combination, the signal represents the maximum coherence achievable from a single signal to the output, as shown in Figure 6.14.

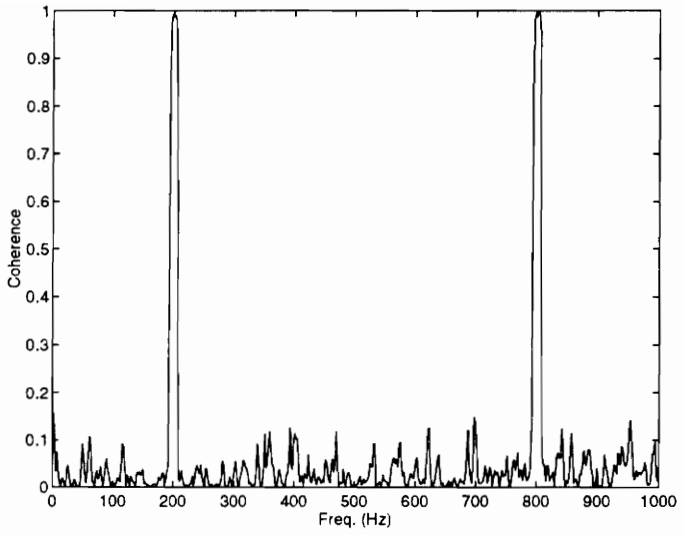


FIGURE 6.14 COHERENCE OF COP FILTERED REFERENCE SIGNAL

Based on the signal power in the plant disturbance signal, Figures 6.13 and 6.14 represent a reference signal which is coherent for all controllable frequencies.

After COP filter design and creation of the coherent reference, the Spectrum system was used to analyze control performance. The system identification portion of the algorithm was performed using the COP filtered reference. This is an important distinction which should be examined in detail. As was mentioned in Chapter 3, the best signal to use for system identification of the control to error path is typically white noise. The reason for this is so that *all* of the modes (or frequencies) within the sample period will be accurately identified. One difficulty which may be encountered with this approach can be explained in terms of the continuing two tonal example.

Suppose only 4 weights were available for creating an estimate of the control to error path in the above example. If a white noise signal was used for identification, it is likely that the four weights would attempt to distribute themselves across the entire bandwidth, identifying frequencies where control may not be necessary. However, if the reference signal (which ideally has all of the frequency components to be controlled) was used, the two tonals would receive the most emphasis in the system identification and the four weights would

likely redistribute emphasizing each of the two tonals. Since each frequency requires two weights to define the system (magnitude and phase), a better chance of identifying the paths of the more important frequencies exists when using the intended reference signal. This presents another argument in favor of using the COP filtering procedure. Using the simple sum reference may not provide adequate representation of the frequency content of the output noise and an unacceptable system identification will result.

Following the system identification portion of the filtered-X LMS, the convergence parameter was set to 0.0001 as in the simple reference summation experiment. The same filter size and plant amplitudes were used to ensure an equal comparison. Figure 6.15 shows the controlled and uncontrolled version of the COP filtered active noise control experiment.

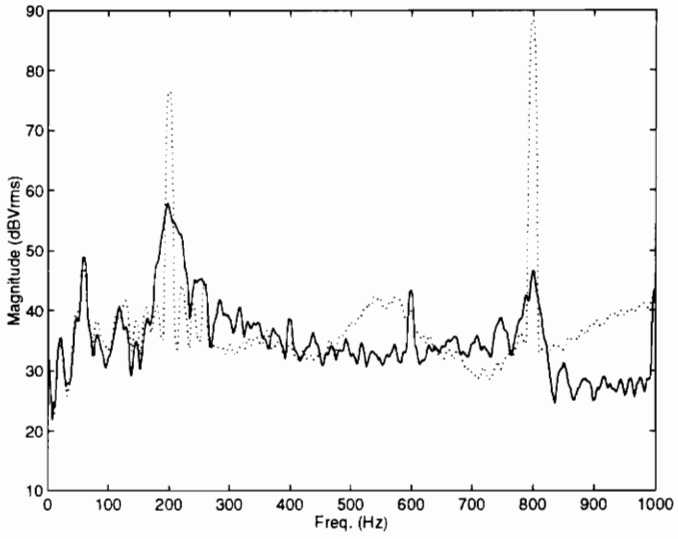


FIGURE 6.15 CONTROLLED AND UNCONTROLLED - COP SUM

Comparing the performance shown for the COP sum in Figure 6.15 with that of the simple summation in Figure 6.12, the differences are quite notable. Providing the controller with a coherent reference signal drastically improves performance on both of the tonals. As was expected, the spillover in adjacent frequencies is not completely eradicated. The primary reason for this is that the reference signal still contains power at frequencies which are not

being transmitted to the disturbance speaker. Finite length filters can only completely suppress frequencies on bin centers. Frequencies lying off bin centers are not directly controllable by the filter, but rather by any leakage exhibited from the bin center filter weights. This phenomenon is a fact of life and cannot be corrected by anything short of infinite length filters, controlling every frequency existing in the bandwidth of interest.

Nevertheless, the performance improvement exhibited by using the COP filtering method is quite notable. The filters, examined individually, also provide an indication of which reference is contributing to what frequencies of the output and in what power ratio. As was mentioned earlier, the COP filtering method is somewhat under utilized in this particular scenario. Coupling can exist between the inputs which would likely decrease performance of the simple sum, but has no effect on the COP filter performance. By calculating the coherent output powers, ordering the inputs appropriately by bin, and designing the filters, the COP design can drastically improve performance of multiple-input single-output adaptive control systems.

Chapter 7

Conclusions and Future Directions

7.1 RETROSPECTIVE

Nearly all physical systems, whether they are acoustical or vibrational, generate outputs that are a function of a *variety* of inputs. When pursuing adaptive feedforward control of these types of systems, it is necessary to have a signal which is representative of all of the frequency content present in the output. Since no single input will ever completely represent the output at every frequency (except in the simplest cases), measurement of the specific contributions of each of the inputs to the output is necessary.

Locating this *uncontrollable coherent reference signal* is typically one of the most difficult tasks in active noise or vibration control. If such a signal exists, performing the actual control within the constraints of physics is usually fairly simple. It was shown on several occasions that the coherence between the reference signal and disturbance noise is directly related to the control performance of the adaptive system. Solving the problem of finding a coherent reference signal represents the primary motivation for this work. Secondary benefits can be found in an automated approach to MISO system identification and correlation analysis of all available signals.

The recent and past literature in these areas is extensive. Many researchers have developed approaches to solving the MISO system problem in the context of energy source identification. Improvements and physical implementations of the energy source identification problem have been presented and verified. Adaptive feedforward control is also a very mature field with countless articles and papers in areas such as adaptive arrays, frequency domain adaptive control, and active noise control. Pertinent representative literature was thoroughly examined in the context of the original work presented herein.

Although the *idea* of taking into account MISO system dynamics in developing a coherent reference has been repeatedly conveyed, the concept of COP filtering has never been mentioned.

Adaptive filtering, as applied to active noise control, was presented in some detail including concepts such as the filtered reference and frequency domain filtering. Although many of the examples presented frequently revert back to ANC, the concepts and equations can be applied to a variety of platforms including active vibration control. Adaptive arrays represent a solution to the multi-source problem which is the motivation of this work. The existence of this particular structure of adaptive filters was not realized until after the development of the COP technology. Even though the two methods can solve the same problem, some notable structural and performance differences exist. Frequency domain filtering represents a computationally efficient method of convoluting a weight sequence with an input sequence to produce a filtered output. Since large filters are typically required to achieve broadband noise control, the frequency domain adaptive LMS presents a viable alternative to time domain noise control in relatively stationary cases.

The signal processing required to analyze the MISO energy source identification problem can become quite complicated when taking into account all the possible permutations which can exist in a large system. The method of determining the (partial) coherent output power represents a way to identify the specific contributions each input makes to the overall output. Examining the power spectrum of the coherent output power provides a visual system identification capability. The overall sum of the coherent output powers of each of the inputs will be exactly equal to the measured output power. If this is not the case, there are signals (or noise) contributing to the measured output which are not being measured as inputs. Automating this system identification tool is the first step in creating a COP filter.

The coherent output power (COP) filter is a digital filter specifically designed to extract only the portion of the respective input which is contributing to the output and amplify it accordingly. Each COP filter is designed in the frequency domain and has the same frequency response as the coherent output power for that respective input. The coherent

output power for a given input and a given frequency bin may actually be the *partial* coherent output power for that frequency depending on the direction of the cross path between the inputs. To avoid confusion, all filters were simply termed as "COP" filters. It is likely that at least some of the bins in each signal will be represented by the coherent, rather than partial coherent output power. The COP filtering procedure relies on the assumption that the signals being examined are all stationary and ergodic. Coherence calculations are meaningless otherwise.

The COP filter design is performed off-line prior to control, in the same way as the filtered reference system identification. Once the filters have been designed, pre-filtering of the references by the COP filters and linear combination of their outputs is performed. This preprocessing technique offers notable performance advantages over a simple linear combination of all candidate references. If an adaptive array structure cannot be constructed, or a closed SISO LMS controller already exists, the COP filtering procedure holds apparent advantages over the adaptive array as well. For systems which conform to the stationary ergodic assumption, the COP filtering approach can also offer significant real time computational savings over the adaptive array. These comparisons were made using primarily numerical techniques for some fairly specific cases. Much remains to be seen in the broader comparison of these methods.

The COP procedure was shown to be effective in both a simulation format and an experimental setting. Only one of many overall structures of the COP filtering procedure was implemented using entirely time domain filtering. Various levels of frequency domain filtering prove to offer drastic computational advantages over the conventional LMS algorithm. Much future work remains in order to classify the COP filtering procedure as a viable technology. As DSP's become more inexpensive and faster, it is likely that the COP procedure can find a market niche in the area of system identification, if not as a SISO adaptive control pre-filter.

7.2 FUTURE WORK

It is highly unlikely that any original research endeavor can be completely defined and analyzed by a single person in a short period of time. The COP filtering technology is no exception. While much work has been done to develop the idea and show its feasibility, many tasks still remain before it will be recognized as a viable technology. Even if the concept is useful in certain applications, it is likely that the ideas will pass into antiquity if a market driven approach to productization is not taken. The following paragraphs discuss the remaining tasks which must be examined before the COP filtering technology should be realized on a product scale.

Before conducting a market analysis for determining the demand of a COP filtering device, several technical issues must be addressed and resolved. Section 5.4 began an analysis of the COP filtering technique as compared to the simple reference summation for the uncorrelated, two-input single-output case. It was shown that many cases exist where the COP filtering design will offer drastic improvements over the simple summation arrangement. However, there are several cases where the COP technique reduces the coherence of the summation from that of the simple summation. Although this was shown using numerical techniques, a more analytical approach should be carried out in order to accurately define the relationships between the two methods. It should be sufficient to determine these relationships for the three input mutually correlated case but would be ideal to define it for the generalized multiple input case. Special numerical optimization techniques must be carried out and were beyond the scope of this study. However, the development of the equations for the simplest case represent the same analysis required for solving the more complex arrangements.

As with the simple summation, the COP technique needs to be analyzed as compared to the adaptive array. This analysis will require the *independence theory* used to find the minimum mean squared error given the weight error correlation matrix. Comparing the *actual* solutions arrived at by the adaptive array and the COP filtering techniques will illustrate the performance of each method as they deviate from the optimal or expected solution.

Intuitively, the adaptive array may not perform as well as the COP filtering technique. The preprocessing of the reference signals in the COP filtering method deterministically provide a single coherent reference signal which is used in a SISO adaptive controller. The adaptive filter in this arrangement is only required to perform control using the coherent reference signal. The adaptive array filters however, are required to both extract the coherent signals *and* provide control authority. According to Elliot and Nelson in [2], the adaptive array can perform according to the multiple coherence of the inputs to the output. If this is the case, the adaptive array will provide the same result as the COP filter procedure for control. However, it is expected that this performance can only be expected within a certain SNR limiting the coherence of each input with respect to the other. Therefore, as was seen in Section 5.5, the adaptive array weights cannot accurately represent the system identification, even if the control is acceptable. A more detailed mathematical analysis of these two methods and their converged solutions is required to draw any further conclusions. It is already apparent however, that the real time computation of the COP filter controlled plant presents a more computationally efficient method than the adaptive array.

In addition to an analytical analysis of the adaptive array with respect to the COP filtering technique, an actual performance comparison implemented on DSP hardware is suggested. Too often, simulation and analytical comparisons are used to draw wide ranging conclusions about the performance of certain inventions. Many times all of the aversive environmental variables which will inevitably plague an experiment cannot possibly be taken into account. It is always necessary to "bench test" and verify all suppositions with practical applications. Clearly this is the case for any product oriented developments but should also be considered in the pure research community.

Following experimental and analytical verification of the COP filtering methodologies, several design optimization tasks should be explored in order to optimize performance. The experimental C code used to demonstrate the COP performance was written in a very inefficient manner. The primary goal was to illustrate that the COP filtering technique can work in a practical setting as well as in simulation. One of the first main modifications which is suggested involves a complete restructuring of the calculations of averaged values.

The current method is to save all of the data before any calculations take place. It is suggested that the averaged values of the power spectra, cross-spectra, coherence, etc. be continually calculated in real time. This organization will conserve memory and allow an indefinite number of averages along with a great many more inputs. Only the current and averaged vectors will be required to be saved at any given time.

Having conserved memory, a further modification of the code in Appendix B involves conservation of computation. Frequency domain filtering was introduced in Chapter 3 and revisited in COP filter implementations discussed in Chapter 5. The actual implementation (as opposed to the calculation or design) of the COP filters is optimized by filtering in the frequency domain. While there are drawbacks to frequency domain *adaptive* filtering, the preprocessing of the reference signals do not suffer from the same non-stationary issues. The design of the COP filters requires that the signals be stationary and ergodic. However, the filtering and combination of the reference signals occurs during control and certain changes can occur in the plant which will be corrected by the SISO adaptive filter but will not be affected by the filtering of the references. Significant computational savings can be realized by (at least) performing the filtering of the reference signals in the frequency domain. The next step is to use a frequency domain adaptive algorithm. Assuming stationary signals, the update drawbacks of the frequency domain LMS will not affect performance but will improve computational complexity. It is therefore suggested that the filtering of the reference signals be performed in the frequency domain as well as the adaptive controller. Depending on stationarity of the control portion of the system, update advantages may supersede the gains achieve in computation; in this case the time domain LMS is suggested.

The next step necessary in verifying the feasibility of the COP procedure from a marketing perspective involves expanding the capabilities of the current experimental system. At the very least it is expected that three candidate reference possibilities will be required to present a useful product. It would be desired however, to maximize the potential of the DSP system by using all of the available inputs. The dSPACE system (C40) has 20 simultaneous sampling inputs while the Spectrum (C31) has 16. With one of each of those sets of

channels required for the error (or output) signal a maximum of 19 or 15 candidate references could be allowed. Since the number of computations increases factorially with the number of inputs, processor limitations must be examined. If it is assumed that all the signals are stationary (as is required), it is acceptable to go off-line between collecting ensembles in order to calculate the averaged quantities. (Lost data will be the same, in time, as the collected data so the information is identical). Practically speaking, it will also be cost effective to include both the COP procedure and the filtered-X LMS on the same processor. Computational limitations may be reached rather quickly if it is required to filter each reference signal *and* perform the LMS all in the same sample period. (Herein lies the advantage of parallel processing).

Finally, it will be necessary to examine the performance of the product in a variety of different scenarios before its true worth will be realized. The experimental procedure depicted in Chapter 6 represents a carefully controlled environment with prior knowledge of all of the signals involved. While the results were extremely promising, performance as applied to actual physical implementations will be required. This process is the first step in determining a market niche for such a product. If several "real life" applications can be identified where the COP filtering algorithm improves performance or offers a new and distinct advantage over current methods, it is likely that a product using this technology could be sold.

Two very distinct markets may exist for this type of technology. If a PC based system could be designed where the user installs a piece of GUI software, a system identification tool may be realized. The external board accepts any number of candidate references (inputs) and a single error signal (output). After running the system, on screen indication will show the individual contribution of each input with respect to the given output and provide suggestions for which references seem to be unnecessary. Independent of adaptive control, this piece of hardware would only design the COP filter shapes, never filtering or providing any output.

The second market would be for engineers who currently have a turnkey SISO adaptive controller but a host of candidate references. The system mentioned above would be expanded to include a filtering mode which creates the coherent reference after performing the system identification. It is likely that this incarnation is less attractive from a market standpoint unless distinct advantages over adaptive arrays are noted. Engineers interested in multiple reference control may pursue the adaptive array structure as it is more common and simple to implement.

A variety of possible implementations of the COP filtering technology exist in both the frequency domain and time domain. Several realistic applications and implementations of the COP filtering methodology also exist. Before pursuing the technology to the product stages however, several comparison analyses and feasibility studies remain. Based on this work, it is evident that the COP filtering ideas and designs can provide a variety useful tools for future development of system identification and effective adaptive feedforward control.

REFERENCES

[1] B. Widrow and S. Stearns, *Adaptive Signal Processing.* Prentice-Hall, Englewood Cliffs, N.J., 1985.

[2] P. A. Nelson and S. J. Elliot, *Active Control of Sound*, Academic Press Ltd., London, England, 1992.

[3] J.S. Bendat and A.G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*, 2nd Edition, John Wiley & Sons, Inc., New York, 1993.

[4] L.L. Beranek and I.L. Ver, *Noise and Vibration Control Engineering: Principles and Applications*, John Wiley & Sons, Inc., New York, 1992.

[5] W.R. Saunders and M.A. Vaudrey, "Active Noise Control Systems: Designing for the Auditory System," *Proceedings of Noise-Con 96*, pp.385-390, 1996.

[6] J.S. Bendat and A.G. Piersol, *Random Data: Analysis and Measurement Procedures*, Wiley-Interscience, New York, 1971.

[7] T.G. Beckwith, R.D. Marangoni, and J.H. Lienhard V, *Mechanical Measurements*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1993.

[8] J.S. Bendat, "Solutions for the Multiple Input/Output Problem," *Journal of Sound and Vibration*, v. 44 no. 3, pp. 311-325, 1976.

[9] J. Park and K Kim, "Source Identification Using Multi-Input/Single-Output Modeling and Causality Checking of Correlated Inputs," *Journal of Vibration and Acoustics*, v. 116, pp. 232-236, April 1994.

[10] S.Haykin, *Adaptive Filter Theory,* 3rd ed, Prentice-Hall, Upper Saddle River, N. J., 1996.

[11] A.V. Oppenhiem and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, N.J., 1989.

[12] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*, 3rd Edition, Prentice Hall, Upper Saddle River, N.J., 1996.

[13] L.B. Jackson, *Digital Filters and Signal Processing*, 3rd Edition, Kluwer Academic Publishers, Boston, Massachusetts, 1996.

[14] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing A Practical Approach*, Addison-Wesley Publishers Ltd., Wokingham, England, 1993.

[15] M. Dentino, J. McCool, and B. Widrow, "Adaptive Filtering in the Frequency Domain," *Proc. IEEE* vol. 66, pp. 1658-1659, Dec. 1978.

[16] A. Weiss and D. Mitra, "Digital Adaptive Filters: Conditions for Convergence, Rates of Convergence, Effects of Noise and Errors Arising from the Implementation," *IEEE Trans. Info. Theory* vol. IT-25, pp.637-652, Nov. 1979.

[17] S. S. Narayan and A. M. Peterson, "Frequency Domain Least-Mean-Square Algorithm," *Proc. IEEE* vol. 69, pp.124-126, Jan. 1981.

[18] F. A. Reed and P. L. Feintuch, "A Comparison of LMS Adaptive Cancellers Implemented in the Frequency Domain and the Time Domain," *IEEE Trans. on Circuits and Systems* vol. CAS-28, pp. 610-615, June 1981.

[19] J.J. Shynk, "Frequency Domain and Multirate Adaptive Filtering," *IEEE Signal Processing Magazine*, pp. 14-37, January 1992.

[20] J.G. Casali and G.S. Robinson, "Narrow-Band Digital Active Noise Reduction in a Siren-Cancelling Headset: Real-Ear and Acoustical Manikin Insertion Loss," *Noise Control Engineering Journal*, v. 42 no. 3, pp. 101-114, 1994.

[21] S. Laugesen and S.J. Elliott, "Multichannel Active Control of Random Noise in a Small Reverberant Room," *IEEE Transactions on Speech and Audio Processing*, v. 1 no. 2, pp.241-249, 1993.

[22] S.J. Elliott, I.M. Stothers, and P.A. Nelson, "A Multiple Error LMS Algorithm and its Application to the Active Control of Sound and Vibration," *IEEE Trasactions on Acoustics, Speech, and Signal Processing*, v. ASSP-35 no. 10, pp.1423-1434, 1987.

[23] A.J. Efron and L.C. Han "Wide-Area Adaptive Active Noise Cancellation," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, v. 41 no. 6, pp.405-409, 1994.

[24] R. Zelinski, "A Microphone Array with Adaptive Post-Filtering for Noise Reduction in Reverberant Rooms," *International Conference on Acoustics, Speech, and Signal Processing*, v. 2, pp. 2578-2581, 1988.

[25] J.B. Allen, D.A. Berkley, and J. Blauert, "Multimicrophone Signal-Processing Technique to Remove Room Reverberation from Speech Signals," *Journal of the Acoustical Society of America*, v. 62 no. 4, pp. 912-915, 1977.

[26] K. Martin and P. Vary, "Combined Acoustic Echo Cancellation Dereverberation and Noise Reduction: A Two Microphone Approach," *Annals de Telecommunications*, v. 49 no. 7-8, pp. 429-438, 1994.

[27] S. Laugesen and S.J. Elliott, "Multichannel Active Control of Random Noise in a Small Reverberant Room," *IEEE Transactions on Speech and Audio Processing*, v. 1 no. 2, pp.241-249, 1993.

[28] S.F. Boll and D.C. Pulsipher, "Suppression of Acoustic Noise in Speech Using Two Microphone Adaptive Noise Cancellation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. ASSP-28 no. 6, pp.752-753, 1980.

[29] K.U. Simmer, S. Fischer, and A. Wasiljeff, "Suppression of Coherent and Incoherent Noise Using a Microphone Array," *Annals de Telecommunications*, v. 49 no. 7-8, pp. 439-446, 1994.

[30] T. Fujii and S. Shimada, "Linear Combined Multichannel Adaptive Digital Filters," *Electronics and Communications in Japan*, Part 1 v. 71 no. 4, pp. 53-61, 1988.

[31] D. Guicking and M. Bronzel, "Multichannel Broadband Active Noise Control in Small Enclosures," *Proceedings of InterNoise 90*, pp. 1255-1258, 1990.

[32] M. Dentino, J. McCool, and B. Widrow, "Adaptive Filtering in the Frequency Domain," *Proceedings of the IEEE*, v. 66 no. 12, pp. 1658-1659, 1978.

[33] D. Mansour and A.H. Gray, Jr., "Unconstrained Frequency-Domain Adaptive Filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. ASSP-30 no. 5, pp. 726-734, 1982.

[34] X. Li and W.K. Jenkins "Convergence Properties of the Frequency-Domain Block-LMS Adaptive Algorithm," *Proceedings of the 28th Asilomar Conference on Signals, Systems, Computers*, Part 2 (of 2), pp. 1515-1519, 1994.

[35] F. A. Reed and P. L. Feintuch, "A Comparison of LMS Adaptive Cancellers Implemented in the Frequency Domain and the Time Domain," *IEEE Trans. on Circuits and Systems* vol. CAS-28, pp. 610-615, June 1981.

[36] S. S. Narayan and A. M. Peterson, "Frequency Domain Least-Mean-Square Algorithm," *Proc. IEEE* vol. 69, pp.124-126, Jan. 1981.

[37] N.J. Bershad and P.L. Feintuch, "Analysis of the Frequency Domain Adaptive Filter," *Proceedings of the IEEE*, v. 67 no. 12, pp. 1658-1659, 1979.

[38] J.J. Shynk, "Frequency Domain and Multirate Adaptive Filtering," *IEEE Signal Processing Magazine*, pp. 14-37, January 1992.

[39] S. Kijimoto, H. Shimojima, and A. Shibahara, "Frequency Domain Adaptive Algorithm for Active Noise Control," *Transactions of the Japan Society of Mechanical Engineers*, v. 62 no. 596, pp. 1297-1402, 1996.

[40] Q. Shen, "Frequency-Domain Multichannel Optimal Adaptive Algorithm for Active Control of Sound and Vibration," *Proceedings of the 1994 National Conference on Noise Control Engineering*, pp. 321-324, 1994.

[41] M.E. Deisher, "Practical Considerations in the Implementation of a Frequency-Domain Adaptive Noise Canceller," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, v. 41 no. 2, pp. 164-168, 1994.

[42] P. Koers, "Adaptive, Active Reduction of Periodic Noise Using a Frequency Domain LMS Algorithm," *Proceedings of Inter-Noise 86*, pp. 595-600, 1986.

[43] S.P. Applebaum, "Adaptive Arrays," *IEEE Transactions on Antennas and Propagation*, v. AP-24 no. 5, pp. 585-607, 1976.

[44] P.A. Kullstam, "Array Performance Optimization Using Projection Techniques," *National Telecommunications Conference Record* (New Orleans, LA), Session C-7, 1981.

[45] L.L. Hororwitz and K.D. Senne, "Performance Advantage of Complex LMS for Controlling Narrow-Band Adaptive Arrays," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. ASSP-29 no. 3, pp. 722-736, 1981.

[46] V.V. Popovskii and S.V. Kobin, "The Use of Adaptive AA in the Processing of Partially Coherent Signals," *Radioelectronics and Communications Systems (English Translation of Izvestiya Vysshikh Uchebnykh Z.)*, v. 29 no. 2, pp. 77-80, 1986.

[47] W.F. Gabriel, "Using Spectral Estimation Techniques in Adaptive Processing Antenna Systems," *IEEE Transactions on Antennas and Propagation*, v. AP-34 no. 3, pp. 291-300, 1986.

[48] I.J. Gupta and A.A. Ksienski, "Adaptive Antenna Arrays for Weak Interfering Signals," *IEEE Transactions on Antennas and Propagation*, v. AP-34 no. 3, pp. 420-426, 1986.

[49] J.S. Bendat, "System Identification from Multiple Input/Output Data," *Journal of Sound and Vibration*, v. 49 no. 3, pp. 293-308, 1976.

[50] C.R.S. Talbot, "Coherence Function Effects on Phase Difference Determination," *Journal of Sound and Vibration*, v. 39 no. 3, pp. 345-358, 1975.

[51] R.J. Alfredson, "The Partial Coherence Technique for Source Identification on a Diesel Engine," *Journal of Sound and Vibration*, v. 55 no. 4, pp. 487-494, 1977.

[52] M.E. Wang and M.J. Crocker "On the Application of Coherence Techniques for Source Identification in a Multiple Noise Source Environment," *Journal of the Acoustical Society of America*, v. 74 no. 3, pp. 861-872, 1983.

[53] J. Park and K Kim, "Source Identification Using Multi-Input/Single-Output Modeling and Causality Checking of Correlated Inputs," *Journal of Vibration and Acoustics*, v. 116, pp. 232-236, April 1994.

[54] N.L. Owsley, "Adaptive Data Orthogonalization," *Proceedings of the 1978 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 109-112, 1978.

[55] C.S. Lindquist and W.H. Haas, "Adaptive Correlation Filters Defined in the Frequency Domain," *Conference Record - Eighteenth Asilomar Conference on Circuits, Systems Computers.*, pp. 327-331, 1985.

[56] H. Subbaram and K. Abend, "Interference Suppresion Via Orthogonal Projections: A Performance Analysis," *IEEE Transactions on Antennas and Propagation*, v. 41 no. 9, pp. 1187-1193, 1993.

[57] M.J. Gelle, "An Adaptive Phase Processing Technique :pplied to the Recovery of a Coherent Reference," *1990 IEEE Military Communications Conference - MILCOM 90, Part 1 (of 3)*, pp.146-150, 1990.

[58] R.J. Patton, M. Miles, and P. Taylor, "Use of the Coherence Function for a Comparison of Test Signals for Frequency Domain Identification," *International Conference on CONTROL '91*, v. 1 no. 332, pp. 651-657, 1991.

[59] K. Lou, D.E. Lyon, and P.J. Sherman, "System Identification and Coherence Analysis in the Presence of a Harmonic Signal," Proceedings of the 9th International Modal Analysis Conference, Part 1 (of 2), pp. 728-734, 1991.

[60] R.T. Compton, Jr. , *Adaptive Antennas: Concepts and Performance*, Prentice Hall, Englewood Cliffs, N.J., 1988.

APPENDIX A: MATLAB SIMULATIONS

SECTION 3.2.2 SIMULATION

```
%MAIN FILE TO RUN signals.m and lmsacous.m
close all
clear
signals
order = 2;
mu = 0.1;
[w_hats,y_sig] = lmsacous(order,mu,p_plant,ref_inp,max_iter);
```

```
%signals.m
%GENERATE SIGNALS TO BE USED IN
%LMS ALGORITHM
%
%ref_inp = u or reference input
%p_plant = p or plant filter output

%F AND FS ARE IN HERTZ AND FS SHOULD BE > 2X AT LEAST

samp_freq = 20;
int_freq = 2;
max_iter = 400;

n = [1:max_iter];

%CREATE REFERENCE INPUT

ref_amp = 1.3;
ref_phase = 0.0;
ref_inp = (ref_amp*sin((2*pi*(int_freq/samp_freq)*n)+ref_phase))';

%CREATE DESIRED INPUT WITH SOME NOISE AT SAME FREQUENCY
%BUT DIFFERENT AMPLITUDE AND PHASE

noise = rand(size(1:max_iter));
noise = (noise - mean(noise));

p_plant_amp = 2.8;
p_plant_phase = 3.2;
p_plant = p_plant_amp*sin((2*pi*(int_freq/samp_freq)*n)-p_plant_phase);
```

```
function [w_hats,y_sig] = lmsacous(order,mu,plant_out,ref_inp,max_iter)

%LMS CODE FOR CANCELLING SIN WAVE
%USING M WEIGHTS

%LMS ALGORITHM NON-FILTERED X USING M WEIGHTS
%ref_inp AND w_hats MUST BE IN COLUMN FORM
%THE ref_inp IS u
%y-sig IS ERR MIC SUM OF CONTROL AND P_PLANT DISTURBANCE
%c_cont IS CONTROL OUTPUT
%d_desir IS 0

M = order;

w_hats = [zeros(size(1:M))]';
p_plant = plant_out;
d_desir = 0;

for index = M:max_iter

        c_cont(index) = w_hats'*flipud(ref_inp(index-M+1:index));
        y_sig(index) = p_plant(index) + c_cont(index);
        e_err(index) = d_desir - y_sig(index);
        w_hats = w_hats + mu*flipud(ref_inp(index-M+1:index))*e_err(index);

end
```

SECTION 3.3.2 SIMULATION

```
%GENERATE SIGNALS TO BE USED IN
%FILTERED X - LMS ALGORITHM GENERATE THESE
%FROM DRIVING A REFERENCE SIGNAL THROUGH A P2E (PLANT TO
%ERROR) FILTER
%AND THROUGH A C2E (CONTROL TO ERROR) PATH FILTER
%
%ref_inp = u or reference input
%p_plant = p or plant filter output
%filt_x = filtered x signal to drive lms algorithm

%F AND FS ARE IN HERTZ AND FS SHOULD BE > 2X AT LEAST

samp_freq = 16;
int_freq = 1.5;
max_iter = 2000;

n = [1:max_iter];

%CREATE REFERENCE INPUT

ref_amp = 1.3;
ref_phase = 0.0;
ref_inp = (ref_amp*sin((2*pi*(int_freq/samp_freq)*n)+ref_phase))';

%CREATE PLANT OUTPUT WITH SOME NOISE AT SAME FREQUENCY
%BUT DIFFERENT AMPLITUDE AND PHASE

p_plant_amp = 2.8;
p_plant_phase = 3.2;
p_plant = p_plant_amp*sin((2*pi*(int_freq/samp_freq)*n)-p_plant_phase);

%CREATE FILTERED REFERENCE SIGNAL FILTERED BY THE C2E PATH

num_c2e = 1;
den_c2e = real(poly([0.3 0.4 -0.6 -0.9 0.8+0.3*i 0.8-0.3*i]));
filt_x = filter(num_c2e,den_c2e,ref_inp);
```

```
%FILTERED -X LMS
M = 10;

w_hats = [zeros(size(1:M))]';
d_desir = 0;
mu = 0.0005;
```

```
for index = M:max_iter

        af_out(index) = w_hats'*flipud(ref_inp(index-M+1:index));
        temp = filter(num_c2e,den_c2e,af_out(index-M+1:index));
        c_cont(index) = temp(M);
        clear temp
        y_sig(index) = p_plant(index) + c_cont(index);
        e_err(index) = d_desir - y_sig(index);

%ONLY CHANGE THIS SIGNAL FOR FILTERED OR NON-FILTERED X
        w_hats = w_hats + mu*flipud(ref_inp(index-M+1:index))*e_err(index);

end
```

SECTION 3.5.3 SIMULATION

```
%MAIN M-FILE WHICH SETS UP SIGNALS AND PRESENTS A CHOICE
%FOR THE USER FOR CONSTRAINED, UNCONSTRAINED, OR CIRCULAR
%FREQUENCY DOMAIN LMS
clear
close all
signals
select = menu('choose method','constrained',...
        'unconstrained','circular');

if select == 1
        cfrqlms
elseif select == 2
        ufrqlms
else
        freqlms
end
```

```
%signals.m
%GENERATE SIGNALS TO BE USED IN FREQ DOM LMS ALGORITHM
%F AND FS ARE IN HERTZ AND FS SHOULD BE > 2X AT LEAST

samp_freq = 16;
int_freq = 1.33;
max_iter = 2000;

n = [1:max_iter];

%CREATE REFERENCE INPUT

ref_amp = 1.3;
ref_phase = 0.0;
ref_time = (ref_amp*sin((2*pi*(int_freq/samp_freq)*n)+ref_phase));

%CREATE DESIRED INPUT WITH SOME NOISE AT SAME FREQUENCY
%BUT DIFFERENT AMPLITUDE AND PHASE

%noise = rand(size(1:max_iter));
%noise = (noise - mean(noise));

plant_amp = 2.8;
plant_phase = 3.2;
plant = plant_amp*sin((2*pi*(int_freq/samp_freq)*n)+plant_phase);
```

```
%cfrqlms.m
%FREQ DOMAIN LMS CODE UTILIZING 5 FFT OPERATIONS AND THUS
%THE FULL GRADIENT CONSTRAINT.  OVERLAP SAVE IS USED ON
%BOTH THE CONVOLUTION AND THE CORRELATION.

%NUMBER OF WEIGHTS, THUS THE NUMBER OF INPUT SAMPLES = 2N
order = 63;
M = order + 1;

%NUMBER OF CORRECT (LINEARLY CONVOLUTED) OUTPUT POINTS
N = M/2;

%OPTIMAL BLOCK SIZE IS THE SAME AS N
L = N;

%M WEIGHTS and USE A CONSTANT MU (COULD BE TIME VARYING WITH
%INPUT POWER)
w_hats = (zeros(size(1:M)))';
mu = 300;
d_time = 0;

ref_time = ref_time;
max_blk = floor((length(ref_time)-L)/L);

for blk = 1:(max_blk-1)

        ref_freq(blk,:) = (2/M)*fft([ref_time(1+(blk-1)*L:(blk*L)) ,...
          ref_time((blk*L + 1):(blk*L + L))] , M);

        c_freq(blk,:) = (ref_freq(blk,:)'.*w_hats)';
         c_temp = real(ifft(c_freq(blk,:)));
         c_time(blk,:) = c_temp(1,L+1:M);

        y_time(blk,:) = plant( (1+(blk-1)*L) : (blk*L) )  +  c_time(blk,:);

        e_time(blk,:) = d_time - y_time(blk,:);
         e_freq(blk,:) = (2/M)*fft( [ zeros(size(1:32)),e_time(blk,:)] , M  );

        phi_temp = real(ifft(  conj(ref_freq(blk,:)).*e_freq(blk,:)  ));
        phi(blk,:) = phi_temp(1,1:L);
        w_hats = w_hats + mu*((2/M)*fft([phi(blk,:), zeros(size(1:N)) ],M))';

        clear c_temp phi_temp
end
```

```
%ufrqlms.m
%FREQ DOMAIN LMS CODE UTILIZING ONLY 3 FFT OPERATIONS AND
%THUS
%THE UNCONSTRAINED.  OVERLAP SAVE IS USED ON
%ONLY THE CORRELATION. THE CONVOLUTION OF Y = XW IS CIRCULAR

%NUMBER OF WEIGHTS, THUS THE NUMBER OF INPUT SAMPLES = 2N
order = 63;
M = order + 1;

%NUMBER OF CORRECT (LINEARLY CONVOLUTED) OUTPUT POINTS
N = M/2;

%OPTIMAL BLOCK SIZE IS THE SAME AS N
L = N;

%M WEIGHTS and USE A CONSTANT MU (COULD BE TIME VARYING WITH
%INPUT POWER)
w_hats = (zeros(size(1:M)))';
mu = 10;
d_time = 0;

ref_time = ref_time;
max_blk = floor((length(ref_time)-L)/L);

for blk = 1:(max_blk-1)

        ref_freq(blk,:) = (2/M)*fft([ref_time(1+(blk-1)*L:(blk*L)) ,...
          ref_time((blk*L + 1):(blk*L + L))] , M);

        c_freq(blk,:) = (ref_freq(blk,:)'.*w_hats)';
         c_temp = real(ifft(c_freq(blk,:)));
         c_time(blk,:) = c_temp(1,L+1:M);

        y_time(blk,:) = plant( (1+(blk-1)*L) : (blk*L) )  +  c_time(blk,:);

        e_time(blk,:) = d_time - y_time(blk,:);
         e_freq(blk,:) = (2/M)*fft( [ zeros(size(1:32)),e_time(blk,:)] , M  );

        w_hats = w_hats + mu*(conj(ref_freq(blk,:)).*e_freq(blk,:))';

        clear c_temp phi_temp
end
```

```
%frqlms.m
%FREQUENCY DOMAIN LMS CODE FOR CANCELLING SIN WAVE
%USING order WEIGHTS
%FREQ DOMAIN LMS ALGORITHM NON-FILTERED X USING M WEIGHTS
%ref_time AND w_hats MUST BE IN COLUMN FORM
%
%y-TIME IS ERR MIC SUM OF CONTROL AND PLANT DISTURBANCE
%c_freq and c_time IS CONTROL OUTPUT
%d_TIME IS 0
%NO OVERLAP SAVE IMPLEMENTED GIVES SMALL MISADJUSTMENT DUE
%CIRCULAR CONVOLUTION EFFECTS

order = 64;
M = order;
L = M;

w_hats = [zeros(size(1:M))]'; %COMPLEX VALUED WEIGHTS IN FREQ DOMAIN
d_time = 0;
mu = 10;

%SET UP A FFT MATRIX FOR FREQ DOMAIN VERSION OF REF_TIME
%EACH ROW IS THE FFT OF THE PAST L POINTS OF DATA

        ref_time = ref_time;
        max_k = floor((length(ref_time)-L)/L);

        for k = 1:max_k,

          ref_freq = (2/L)*(fft(ref_time( (1+(k-1)*L) : (k*L) ),L));
          ref_freq_mat(k,:) = ref_freq;
          clear ref_freq

        end

clear k
for k = 1:max_k,

%NOT A DOT PRODUCT, ONLY ELEMENT BY ELEMENT
        c_freq(k,:) = (w_hats').*ref_freq_mat(k,:);
          c_time(k,:) = real(ifft(c_freq(k,:)));

        y_time(k,:) = plant( (1+(k-1)*L) : (k*L) ) + c_time(k,:);

        e_time = d_time - y_time(k,:);
          e_freq(k,:) = (2/L)*(fft(e_time,L));
```

```
        w_hats = w_hats + (mu*conj(ref_freq_mat(k,:)).*e_freq(k,:))';

end
```

SECTION 5.3.2 and 5.4.1 SIMULATIONS

```
%two input case
%GENERATES SIGNALS FOR COHERENCE AND ADAPTIVE FILTER

Fs = 2;
iter = 30000;
npts = 512; % = ORDER OF FILTER

randn('seed',395633076)
z_1 = randn(size([1:iter]));
randn('seed',609335419)
z_2 = randn(size([1:iter]));
%randn('seed',1115880016)
z_3 = randn(size([1:iter]));

%MAKE BANDPASS FILTER DESIGN
%TYPE 1 FILTER IS PEAK AT FIRST TONAL
%TYPE 2 FILTER IS PEAK AT SECOND TONAL
%TYPE 3 FILTER IS PEAKS AT BOTH TONALS

pole_rad = 1.0;
pole_ang = 45;
zero_rad = 0.7;
zero_ang = 45;

pole_real = pole_rad*(cos(pole_ang*(pi/180)));
pole_imag = pole_rad*(sin(pole_ang*(pi/180)));
zero_real = zero_rad*(cos(zero_ang*(pi/180)));
zero_imag = zero_rad*(sin(zero_ang*(pi/180)));

numz_1 = 0.0001*poly([zero_real+i*zero_imag zero_real-i*zero_imag]);
denz_1 = poly([pole_real+i*pole_imag pole_real-i*pole_imag]);

pole_ang = 135;
zero_ang = 135;

pole_real = pole_rad*(cos(pole_ang*(pi/180)));
pole_imag = pole_rad*(sin(pole_ang*(pi/180)));
zero_real = zero_rad*(cos(zero_ang*(pi/180)));
zero_imag = zero_rad*(sin(zero_ang*(pi/180)));

numz_2 = 0.0001*poly([zero_real+i*zero_imag zero_real-i*zero_imag]);
denz_2 = poly([pole_real+i*pole_imag pole_real-i*pole_imag]);
```

```
numz_3 = conv(numz_1,numz_2);
denz_3 = 0.0001*conv(denz_1,denz_2);

%SIGNAL LEVELS ARE ADJUSTED TO GIVE A REASONABLE MU
%DEPENDENT ON THE INPUT CORRELATION MAT (HIGH CORR,HAS
%A HIGH MU)

input_1 = filter(1000*numz_3,denz_3,z_1);
input_2 = filter(1000*numz_3,denz_3,z_2);
input_1 = input_1(1000:length(input_1));
input_2 = input_2(1000:length(input_2));

%REFERENCE IS THE SAME AS THE INPUT-THE PLANT IS THE
%AMPLIFIED (NON PHASE SHIFTED) VERSION OF THE INPUT IN THIS
%TRIAL CASE
%THESE SIGNALS ARE USED BY THE ADAPTIVE CODE

ref_time = input_1 + input_2;
output_noise = 0.7*randn(size(1:length(ref_time)));
temp1 = filter(1000*numz_1,denz_1,z_1);temp2 = filter(1000*numz_2,denz_2,z_2);

plant = temp1(1000:length(temp1)) + temp2(1000:length(temp2)) + output_noise;
```

```
%NON ADAPTIVE DEMONSTRATION
%DESIGN FILTERS (FREQUENCY DEPENDENT) BASED ON
%PART COH OUTPUT POWER

[C_1_y] = cohere(input_1,plant,npts);
[C_2_y] = cohere(input_2,plant,npts);

%TAKE DIFFERENCE AND IF IT'S POSITIVE THEN THE
%FIRST INPUT IS THE SOURCE, IF IT'S NOT THE CROSS
%PATH GOES FROM INPUT 2 TO INPUT ONE.

direc = C_1_y - C_2_y;

C_1y_2 = parco2in(input_2,input_1,plant,npts,Fs);
C_2y_1 = parco2in(input_1,input_2,plant,npts,Fs);
G_yy = psd(plant,npts,Fs);
G_yy_1 = G_yy.*(1 - C_1_y);
G_yy_2 = G_yy.*(1 - C_2_y);


G_ycol1_direct = C_1_y.*G_yy;
G_ycol1_indir = C_1y_2.*G_yy_2;
```

```
G_ycol2_direct = C_2_y.*G_yy;
G_ycol2_indir = C_2y_1.*G_yy_1;

for index = 1:length(direc)

        if direc(index)>0
          G_ycol1(index) = C_1_y(index)*G_yy(index);
          G_ycol2(index) = C_2y_1(index)*G_yy_1(index);
        else
          G_ycol1(index) = C_1y_2(index)*G_yy_2(index);
          G_ycol2(index) = C_2_y(index)*G_yy(index);

        end

end
```

```
%FILTERING PROCESS USING WEIGHTS DETERMINED BY
%PCOP FILE-DO FILTERING IN FREQ DOMAIN

%REMEMBER NPTS IS ORDER OF FILTER

order = npts;
M = order;

%NUMBER OF CORRECT (LINEARLY CONVOLUTED) OUTPUT POINTS
N = order/2;

%OPTIMAL BLOCK SIZE SAME AS N
L = N;

w_hats_1 = [G_ycol1(1:L) fliplr(G_ycol1(1:L))];
w_hats_2 = [G_ycol2(1:L) fliplr(G_ycol2(1:L))];

max_blk = floor((length(ref_time)-L)/L);
input_1 = input_1(1:max_blk*L);
input_1_mat = reshape(input_1,L,max_blk)';

input_2 = input_2(1:max_blk*L);
input_2_mat = reshape(input_2,L,max_blk)';

for blk = 2:max_blk

        input_1_freq(blk,:) = (2/M)*fft([input_1_mat(blk-1,:) ,...
```

```
    input_1_mat(blk,:)] , M);

  nu_input_1_freq(blk,:) = (input_1_freq(blk,:).*w_hats_1);
  nu_input_1_time_all(blk,:) = real(ifft(nu_input_1_freq(blk,:)));
  nu_input_1_time(blk,:) = nu_input_1_time_all(blk,L+1:M);


  input_2_freq(blk,:) = (2/M)*fft([input_2_mat(blk-1,:) ,...
    input_2_mat(blk,:)] , M);

  nu_input_2_freq(blk,:) = (input_2_freq(blk,:).*w_hats_2);
  nu_input_2_time_all(blk,:) = real(ifft(nu_input_2_freq(blk,:)));
  nu_input_2_time(blk,:) = nu_input_2_time_all(blk,L+1:M);

end

nu_input_1_time = reshape(nu_input_1_time',1,blk*L);
nu_input_2_time = reshape(nu_input_2_time',1,blk*L);

nu_ref = nu_input_1_time + nu_input_2_time;
```

```
function [C_23_1] = parco2in(signal_1,signal_2,signal_3,npts,Fs)

%CALCULATES PARTIAL COHERENCE FUNCTION FOR THE
%TWO INPUT CASE ONLY, NEEDED FOR THE CALCULATION
%OF MULTIPLE COHERENCE FUNCTION
%
%READ LIKE COHERENCE OF SIGNAL 2 TO SIGNAL THREE WITHOUT
%THE EFFECTS OF SIGNAL 1
%
%[C_23_1] = parco2in(signal_1,signal_2,signal_3,npts,Fs)

  G_11 = psd(signal_1,npts,Fs);
  G_23 = csd(signal_2,signal_3,npts,Fs);
  G_21 = csd(signal_2,signal_1,npts,Fs);
  G_13 = csd(signal_1,signal_3,npts,Fs);

num = abs(G_11.*G_23 - G_21.*G_13).^2;

  G_22 = psd(signal_2,npts,Fs);
  G_33 = psd(signal_3,npts,Fs);
  C_1_2 = cohere(signal_1,signal_2,npts,Fs);
  C_1_3 = cohere(signal_1,signal_3,npts,Fs);
```

```
den = (G_11.^2).*G_22.*G_33.*(1-C_1_2).*(1-C_1_3);

C_23_1 = num./den;
```

```
function [G_13_2] = rescsd(signal_1,signal_2,signal_3,npts,Fs)

%ESTIMATES THE RESIDUAL CROSS SPECTRAL DENSITY
%FUNCTION USED IN CALCULATING PARTIAL COHERENCE
%WORKS FOR ANY THREE SIGNALS WHERE "FIRST AND OUTPUT"
%ARE THE FIRST TWO USING BENDAT NOTATION AND THE "SECOND"
%IS AFTER THE DOT TERM
%
%function [G_13_2] = rescsd(signal_1,signal_2,signal_3,npts,Fs)

G_13 = csd(signal_1,signal_3,npts,Fs);
G_12 = csd(signal_1,signal_2,npts,Fs);
G_23 = csd(signal_2,signal_3,npts,Fs);
G_22 = psd(signal_2,npts,Fs);

G_13_2 = G_13.*(1 - ((G_12.*G_23)./(G_22.*G_13)));
```

```
%three input case
%GENERATES SIGNALS FOR COHERENCE AND ADAPTIVE FILTER
%SCHEME AS DESCRIBED ON PAGE 41 OF NOTES

Fs = 2;
iter = 20000;
npts = 512; % = ORDER OF FILTER

%randn('seed',395633076)
z_1 = randn(size([1:iter]));
%randn('seed',609335419)
z_2 = randn(size([1:iter]));
%randn('seed',1115880016)
z_3 = randn(size([1:iter]));

%MAKE LOW PASS FOR 1, BANDPASS FOR 2 AND HIGHPASS FOR 3

[lo_num,lo_den] = butter(10,0.33);
[band_num,band_den] = butter(5,[0.33 0.66]);
[hi_num,hi_den] = butter(10,0.66,'high');

input_1 = filter(lo_num,lo_den,z_1) + 10*filter(band_num,band_den,rand(size(1:iter))) + ...
        10*filter(hi_num,hi_den,rand(size(1:iter)));
input_2 = filter(band_num,band_den,z_2) + 10*filter(lo_num,lo_den,rand(size(1:iter))) + ...
```

```
        10*filter(hi_num,hi_den,rand(size(1:iter)));
input_3 = filter(hi_num,hi_den,z_3) + 10*filter(lo_num,lo_den,rand(size(1:iter))) + ...
        10*filter(band_num,band_den,rand(size(1:iter)));

input_1 = input_1(1000:iter) - mean(input_1);
input_2 = input_2(1000:iter) - mean(input_2);
input_3 = input_3(1000:iter) - mean(input_3);

%REFERENCE IS THE SAME AS THE INPUT-THE PLANT IS THE
%AMPLIFIED (NON PHASE SHIFTED) VERSION OF THE INPUT IN THIS
%TRIAL CASE
%THESE SIGNALS ARE USED BY THE ADAPTIVE CODE

ref_time = input_1 + input_2 + input_3;
        output_noise = 0.7*randn(size(1:length(ref_time)));
        temp1 = filter(lo_num,lo_den,z_1);
        temp2 = filter(band_num,band_den,z_2);
        temp3 = filter(hi_num,hi_den,z_3);
  plant = temp1(1000:iter) + temp2(1000:iter) + temp3(1000:iter);
plant = plant - mean(plant);
```

```
%NON ADAPTIVE DEMONSTRATION
%DESIGN FILTERS (FREQUENCY DEPENDENT) BASED ON
%PART COH OUTPUT POWER

[C_1_y] = cohere(input_1,plant,npts);
[C_2_y] = cohere(input_2,plant,npts);
[C_3_y] = cohere(input_3,plant,npts);

G_yy = psd(plant,npts,Fs);
C_2y_1 = parco2in(input_1,input_2,plant,npts,Fs);
G_yy_1 = G_yy.*(1 - C_1_y);
C_3y_12 = parco3in(input_1,input_2,input_3,plant,npts,Fs);
G_yy_12 = G_yy.*(1 - C_1_y).*(1 - C_2y_1);
C_3y_1 = parco2in(input_1,input_3,plant,npts,Fs);
C_2y_13 = parco3in(input_1,input_3,input_2,plant,npts,Fs);
G_yy_13 = G_yy.*(1 - C_1_y).*(1 - C_3y_1);
C_1y_2 = parco2in(input_2,input_1,plant,npts,Fs);
G_yy_2 = G_yy.*(1 - C_2_y);
C_3y_2 = parco2in(input_2,input_3,plant,npts,Fs);
C_1y_23 = parco3in(input_2,input_3,input_1,plant,npts,Fs);
G_yy_23 = G_yy.*(1 - C_2_y).*(1 - C_3y_2);
C_1y_3 = parco2in(input_3,input_1,plant,npts,Fs);
G_yy_3 = G_yy.*(1 - C_3_y);
```

```
C_2y_3 = parco2in(input_3,input_2,plant,npts,Fs);

%CASE WHERE ORDER IS 123
G1_123 = C_1_y.*G_yy;
        G2_123 = C_2y_1.*G_yy_1;
         G3_123 = C_3y_12.*G_yy_12;

%CASE WHERE ORDER IS 132
G1_132 = G1_123;
         G2_132 = C_2y_13.*G_yy_13;
        G3_132 = C_3y_1.*G_yy_1;

%CASE WHERE ORDER IS 213
        G1_213 = C_1y_2.*G_yy_2;
G2_213 = C_2_y.*G_yy;
         G3_213 = G3_123;

%CASE WHERE ORDER IS 231
         G1_231 = C_1y_23.*G_yy_23;
G2_231 = G2_213;
        G3_231 = C_3y_2.*G_yy_2;

%CASE WHERE ORDER IS 312
        G1_312 = C_1y_3.*G_yy_3;
         G2_312 = G2_132;
G3_312 = C_3_y.*G_yy;

%CASE WHERE ORDER IS 321
         G1_321 = G1_231;
        G2_321 = C_2y_3.*G_yy_3;
G3_321 = G1_312;

for index = 1:length(C_1_y)

        if (C_1_y(index) > C_2_y(index)) & (C_1_y(index) > C_3_y(index)) &
(C_2_y(index) > C_3_y(index))
          %ORDER IS 123
          G_ycol1(index) = G1_123(index);
          G_ycol2(index) = G2_123(index);
          G_ycol3(index) = G3_123(index);

        elseif (C_1_y(index) > C_2_y(index)) & (C_1_y(index) > C_3_y(index)) &
(C_3_y(index) > C_2_y(index))
          %ORDER IS 132
          G_ycol1(index) = G1_132(index);
```

```
        G_ycol2(index) = G2_132(index);
        G_ycol3(index) = G3_132(index);

      elseif (C_2_y(index) > C_1_y(index)) & (C_2_y(index) > C_3_y(index)) &
(C_1_y(index) > C_3_y(index))
        %ORDER IS 213
        G_ycol1(index) = G1_213(index);
        G_ycol2(index) = G2_213(index);
        G_ycol3(index) = G3_213(index);

      elseif (C_2_y(index) > C_1_y(index)) & (C_2_y(index) > C_3_y(index)) &
(C_3_y(index) > C_1_y(index))
        %ORDER IS 231
        G_ycol1(index) = G1_231(index);
        G_ycol2(index) = G2_231(index);
        G_ycol3(index) = G3_231(index);

      elseif (C_3_y(index) > C_1_y(index)) & (C_3_y(index) > C_2_y(index)) &
(C_1_y(index) > C_2_y(index))
        %ORDER IS 312
        G_ycol1(index) = G1_312(index);
        G_ycol2(index) = G2_312(index);
        G_ycol3(index) = G3_312(index);

      elseif (C_3_y(index) > C_1_y(index)) & (C_3_y(index) > C_2_y(index)) &
(C_2_y(index) > C_1_y(index))
        %ORDER IS 321
        G_ycol1(index) = G1_321(index);
        G_ycol2(index) = G2_321(index);
        G_ycol3(index) = G3_321(index);

      else
        disp('this shouldn't have happened')
      end

end

G_ycol1 = abs(G_ycol1);
G_ycol2 = abs(G_ycol2);
G_ycol3 = abs(G_ycol3);
```

```
%FILTERING PROCESS USING WEIGHTS DETERMINED BY
%PCOP FILE-DO FILTERING IN FREQ DOMAIN

%REMEMBER NPTS IS ORDER OF FILTER

order = npts;
M = order;

%NUMBER OF CORRECT (LINEARLY CONVOLUTED) OUTPUT POINTS
N = order/2;

%OPTIMAL BLOCK SIZE SAME AS N
L = N;

w_hats_1 = [G_ycol1(1:L) fliplr(G_ycol1(1:L))];
w_hats_2 = [G_ycol2(1:L) fliplr(G_ycol2(1:L))];
w_hats_3 = [G_ycol3(1:L) fliplr(G_ycol3(1:L))];

max_blk = floor((length(ref_time)-L)/L);
input_1 = input_1(1:max_blk*L);
input_1_mat = reshape(input_1,L,max_blk)';

input_2 = input_2(1:max_blk*L);
input_2_mat = reshape(input_2,L,max_blk)';

input_3 = input_3(1:max_blk*L);
input_3_mat = reshape(input_3,L,max_blk)';

for blk = 2:max_blk

        input_1_freq(blk,:) = (2/M)*fft([input_1_mat(blk-1,:) ,...
          input_1_mat(blk,:)] , M);

        nu_input_1_freq(blk,:) = (input_1_freq(blk,:).*w_hats_1);
        nu_input_1_time_all(blk,:) = real(ifft(nu_input_1_freq(blk,:)));
        nu_input_1_time(blk,:) = nu_input_1_time_all(blk,L+1:M);

        input_2_freq(blk,:) = (2/M)*fft([input_2_mat(blk-1,:) ,...
          input_2_mat(blk,:)] , M);

        nu_input_2_freq(blk,:) = (input_2_freq(blk,:).*w_hats_2);
        nu_input_2_time_all(blk,:) = real(ifft(nu_input_2_freq(blk,:)));
        nu_input_2_time(blk,:) = nu_input_2_time_all(blk,L+1:M);
```

```
        input_3_freq(blk,:) = (2/M)*fft([input_3_mat(blk-1,:) ,...
          input_3_mat(blk,:)] , M);

        nu_input_3_freq(blk,:) = (input_3_freq(blk,:).*w_hats_3);
        nu_input_3_time_all(blk,:) = real(ifft(nu_input_3_freq(blk,:)));
        nu_input_3_time(blk,:) = nu_input_3_time_all(blk,L+1:M);

end

nu_input_1_time = reshape(nu_input_1_time',1,blk*L);
nu_input_2_time = reshape(nu_input_2_time',1,blk*L);
nu_input_3_time = reshape(nu_input_3_time',1,blk*L);

nu_ref = nu_input_1_time + nu_input_2_time + nu_input_3_time;
```

```
function [C_34_12] = parco3in(signal_1,signal_2,signal_3,signal_4,npts,Fs)

%CALCULATES PARTIAL COHERENCE OF ONE OF ANY
%THREE INPUTS TO THE OUTPUT FOR A THREE INPUT
%SYSTEM
%
%READ LIKE COHERENCE OF THREE TO FOUR WITHOUT THE
%EFFECTS OF SIGNALS ONE AND TWO (FOUR IS TYPICALLY THE OUTPUT)
%
%[C_34_12] = parco3in(signal_1,signal_2,signal_3,signal_4,npts,Fs)

        G_22_1 = rescsd(signal_2,signal_1,signal_2,npts,Fs);
        G_34_1 = rescsd(signal_3,signal_1,signal_4,npts,Fs);
        G_32_1 = rescsd(signal_3,signal_1,signal_2,npts,Fs);
        G_24_1 = rescsd(signal_2,signal_1,signal_4,npts,Fs);

G_34_12 = (G_22_1.*G_34_1)-(G_32_1.*G_24_1);

        G_33_1 = rescsd(signal_3,signal_1,signal_3,npts,Fs);
        G_44_1 = rescsd(signal_4,signal_1,signal_4,npts,Fs);
        C_23_1 = parco2in(signal_1,signal_2,signal_3,npts,Fs);
        C_24_1 = parco2in(signal_1,signal_2,signal_4,npts,Fs);

comb_den = ((G_22_1.^2).*G_33_1).*G_44_1.*(1-C_23_1).*(1-C_24_1);

C_34_12 = (abs(G_34_12).^2)./comb_den;
```

APPENDIX B: ORIGINAL C-CODE FOR ILLUSTRATING COP FILTERING USING A TI-C40 DSP

```
/*-------------------------------------------------------------------------------------------------------*/
/*

COP FILTERING PROCEDURE WRITTEN FOR dSPace SYSTEM USING A TI C40 DSP
PROCESSOR.

USES FIRST INTERRUPT SUBROUTINE TO COLLECT DATA
PAUSES WHILE COP FILTERS ARE CALCULATED
THEN USES SECOND ISR TO FILTER INCOMING DATA AND ADD
THE REFERENCES TO OUTPUT THE CORRECTED REFERENCE

WRITTEN BY Michael A. Vaudrey ON OR ABOUT November 11, 1996

*/
/*-------------------------------------------------------------------------------------------------------*/

/* INCLUDES */

#include <brtenv.h>
#include <ensigdsp.h>
#include <stdlib.h>
#include <math.h>

#define FFT_SIZE 256
#define AVGS 41

float Samp_Freq = 2048.0;
float u, v, w, in_1, in_2, junk, out, single_out_1, single_out_2;
int col_index_1 = 0;
int col_index_2 = 0;
int row_index_1 = 0;
int row_index_2 = 0;
int state = 1;


float input_x1[AVGS][FFT_SIZE] ;
float input_x2[AVGS][FFT_SIZE] ;
float eror_mic[AVGS][FFT_SIZE] ;

float cop_filt_1[FFT_SIZE];
float cop_filt_2[FFT_SIZE];
```

```
float *workspace_1, *workspace_2;
float filtered_output_1, filtered_output_2;


/* FUNCTION PROTOTYPES*/

void MikesCOPDesign(float *time_weights_1, float *time_weights_2);
void AutoSpec(float *Gxx, float **time_input_matrix, int rows, int columns);
void CrossSpec(float *Gxy, float **time_input_matrix_x,
        float **time_input_matrix_y, int rows, int columns);
void OrdCoherence(float *coherence, float *Gxy, float *Gxx, float *Gyy,
        int columns);
void PartialCoherence(float *part_coh_ab_c, float *Gaa, float *Gbb,
        float *Gcc, float *Gab, float *Gac, float *Gcb, int columns);
void ResidCSD(float *Gyy_x, float *Gyy, float *ordcoh_x_y, int columns);
void FreqWeightDesign(float *w_hats_1, float *w_hats_2, float *C_1_y,
        float *C_2_y, float *G_yy, float *PC_1y_2, float *PC_2y_1,
        float *G_yy_1, float *G_yy_2, int columns);
void FreqToFIR(float *w_fir_1, float *w_fir_2, float *w_fft_1,
        float *w_fft_2, int columns);
void VectorMagSqrd(float *output_data, float *input_vector, int columns);
void VectorConjugate(float *output_vec, float *input_vec, int columns);


/* USED FOR EXECUTION TIME CALCULATION */

#define TMR0 0
float exec_time;
unsigned long count0;


/* ERROR FLAG */

volatile int *error = (int *) (DP_MEM_BASE + DP_MEM_SIZE - 1);


/* SYNCHRONOUS INPUT SAMPLING MACRO */

 #define Input_Data(u, v, w)    \
 start_ds2201ad(DS2201_1_BASE);  \
 u = ds2201ad(DS2201_1_BASE, 1);   \
 v = ds2201ad(DS2201_1_BASE, 2);  \
 w = ds2201ad(DS2201_1_BASE, 3)
```

```
/* OUTPUT MACRO*/

#define Output_Chan1(out) ds2201da(DS2201_1_BASE, 21, out);



/*-------------------------------------------------------------------------------------------------------------*/
/* MAIN, RUNS ISR T0 THEN CALCULATES COP THEN ISR T1 FILTERS DATA      */
/*-------------------------------------------------------------------------------------------------------------*/

main()
{

 float DT = 1.0/Samp_Freq;

 init();
 *error = NO_ERROR;

 start_isr_t0(DT);


 /*BACKGROUND PROCESS GOES ONLY AFTER DATA IS TAKEN */

 while(*error == NO_ERROR)
 {

        if (state == 2)
        {

        MikesCOPDesign(cop_filt_1, cop_filt_2);

        col_index_2 = 0;
        state = 3;

        InitFir(&workspace_1, FFT_SIZE);
        InitFir(&workspace_2, FFT_SIZE);
        start_isr_t1(DT);
        }

 }


}
```

```
/*-------------------------------------------------------------------------------------------------------*/
/* INTERRUPT SERVICE ROUTINE FROM TIMER 0   COLLECT DATA                                                 */
/*-------------------------------------------------------------------------------------------------------*/

isr_t0()
{
  begin_isr_t0(*error);

  if ( (col_index_1 > (FFT_SIZE-1)) && (row_index_1 >(AVGS-1))  )
          {
          state = 2;
          disable_isr_t0();
          }

  if(col_index_1 > (FFT_SIZE-1))
          {
          row_index_1++;
          col_index_1 = 0;
          }

  Input_Data(input_x1[row_index_1][col_index_1],
          input_x2[row_index_1][col_index_1], eror_mic[row_index_1][col_index_1]);

  Output_Chan1(input_x1[row_index_1][col_index_1]);

  col_index_1++;

  end_isr_t0();
}



/*-------------------------------------------------------------------------------------------------------*/
/* INTERRUPT SERVICE ROUTINE FROM TIMER 1    MAKE OUTPUT DATA                                            */
/*-------------------------------------------------------------------------------------------------------*/

isr_t1()

{

  begin_isr_t1(*error);


/*THIS SECTION WAS USED TO MONITOR FILTER SHAPES - TROUBLESHOOTING
```

```
  if (col_index_2 > FFT_SIZE)
        col_index_2 = 0;

  single_out_1 = cop_filt_1[col_index_2];
  single_out_2 = cop_filt_2[col_index_2];

  Output_Chan1(single_out_1);
  service_trace();

  col_index_2++;
*/

/*DATA WAS READ IN AND FILTERED, ADDED AND SENT OUT */

  Input_Data(in_1, in_2, junk);

  Fir1(&filtered_output_1, in_1, cop_filt_1, FFT_SIZE, &workspace_1);
  Fir1(&filtered_output_2, in_2, cop_filt_2, FFT_SIZE, &workspace_2);


/*OUTPUT WAS MAGNITUDE SCALED BECAUSE THE INPUT SIGNALS USED
WERE IN THE TEN VOLT RANGE WHICH IS TOO HIGH TO PUT INTO SECOND
DSP */

  Output_Chan1(0.25*(filtered_output_1 + filtered_output_2));

  end_isr_t1();

}



/*-------------------------------------------------------------------------------------------------------*/
/* MAIN FILE THAT RUNS ALL THE OTHER ROUTINES TO DO COP DESIGN    */
/*-------------------------------------------------------------------------------------------------------*/

void MikesCOPDesign(float *time_weights_1, float *time_weights_2)

{

        float **time_input_ptr_x1, **time_input_ptr_x2, **eror_mic_ptr;
        float Gx1x1[FFT_SIZE/2], Gx2x2[FFT_SIZE/2], Gerer[FFT_SIZE/2];
        float Gx1x2[FFT_SIZE], Gx1er[FFT_SIZE], Gx2er[FFT_SIZE];
        float Gx2x1[FFT_SIZE], Gerx1[FFT_SIZE], Gerx2[FFT_SIZE];
        float OrCo_1_er[FFT_SIZE], OrCo_2_er[FFT_SIZE], OrCo_1_2[FFT_SIZE];
        float ParCo_1er_2[FFT_SIZE], ParCo_2er_1[FFT_SIZE];
```

```
float Gerer_x1[FFT_SIZE], Gerer_x2[FFT_SIZE];
float freq_weights_1[FFT_SIZE], freq_weights_2[FFT_SIZE];

int r;

time_input_ptr_x1 = (float **) malloc(AVGS * sizeof(float*));
for(r = 0; r<AVGS ; r++)
  time_input_ptr_x1[r] = input_x1[r];


time_input_ptr_x2 = (float **) malloc(AVGS * sizeof(float*));
for(r = 0; r<AVGS ; r++)
  time_input_ptr_x2[r] = input_x2[r];


eror_mic_ptr = (float **) malloc(AVGS * sizeof(float*));
for(r = 0; r<AVGS ; r++)
  eror_mic_ptr[r] = eror_mic[r];


AutoSpec(Gx1x1, time_input_ptr_x1, AVGS, FFT_SIZE);
AutoSpec(Gx2x2, time_input_ptr_x2, AVGS, FFT_SIZE);
AutoSpec(Gerer, eror_mic_ptr, AVGS, FFT_SIZE);

CrossSpec(Gx1x2, time_input_ptr_x1, time_input_ptr_x2, AVGS, FFT_SIZE);
CrossSpec(Gx1er, time_input_ptr_x1, eror_mic_ptr, AVGS, FFT_SIZE);
CrossSpec(Gx2er, time_input_ptr_x2, eror_mic_ptr, AVGS, FFT_SIZE);

free(time_input_ptr_x1);
free(time_input_ptr_x2);
free(eror_mic_ptr);

VectorConjugate(Gx2x1, Gx1x2, FFT_SIZE);
VectorConjugate(Gerx1, Gx1er, FFT_SIZE);
VectorConjugate(Gerx2, Gx2er, FFT_SIZE);

OrdCoherence(OrCo_1_er, Gx1er, Gx1x1, Gerer, FFT_SIZE);
OrdCoherence(OrCo_2_er, Gx2er, Gx2x2, Gerer, FFT_SIZE);
OrdCoherence(OrCo_1_2, Gx1x2, Gx1x1, Gx2x2, FFT_SIZE);

PartialCoherence(ParCo_1er_2, Gx1x1, Gerer, Gx2x2, Gx1er, Gx1x2, Gx2er,
  FFT_SIZE);
PartialCoherence(ParCo_2er_1, Gx2x2, Gerer, Gx1x1, Gx2er, Gx2x1, Gx1er,
  FFT_SIZE);

ResidCSD(Gerer_x1, Gerer, OrCo_1_er, FFT_SIZE);
ResidCSD(Gerer_x2, Gerer, OrCo_2_er, FFT_SIZE);
```

```
        FreqWeightDesign(freq_weights_1, freq_weights_2, OrCo_1_er, OrCo_2_er,
          Gerer, ParCo_1er_2, ParCo_2er_1, Gerer_x1, Gerer_x2, FFT_SIZE);


        FreqToFIR(time_weights_1, time_weights_2, freq_weights_1, freq_weights_2,
          FFT_SIZE);


}


/*-------------------------------------------------------------------------------------------------------*/
/* CALCULATES THE AUTOSPECTRUM GIVEN A DATA MATRIX                                                       */
/*-------------------------------------------------------------------------------------------------------*/


void AutoSpec(float *Gxx, float **time_input_matrix, int rows, int columns)

{

        float freq_mat[AVGS][FFT_SIZE];
        float freq_one_row[FFT_SIZE];
        float mag_sqrd[FFT_SIZE/2];

        int x,y;
        int r, c;



        for(r = 0; r < rows; r++)
        {
          FftReal(freq_one_row, time_input_matrix[r], FFT_SIZE);

          for(c = 0; c < columns; c++)
            freq_mat[r][c] = freq_one_row[c];
        }



/*MAGNITUDE SQUARED OPERATIONS */


        for (r = 0; r < rows ; r++)
         {
         VectorMagSqrd(mag_sqrd, freq_mat[r], FFT_SIZE);
```

```
        for(x = 0; x < (columns/2); x++)
          freq_mat[r][x] = mag_sqrd[x];
        }


/*MATRIX AVERAGING OPERATIONS */

      for(y = 0; y < (columns/2); y++)
        Gxx[y] = 0.0;

      for(y = 0; y < (columns/2); y++)
        {
          for(x = 0; x < rows;  x++)
            Gxx[y] = freq_mat[x][y] + Gxx[y];
        }

      for(y = 0; y < (columns/2); y++)
        Gxx[y] = (Gxx[y]) / ((float)rows);

}



/*-------------------------------------------------------------------------------------------------------------*/
/* CALCULATES THE CROSS SPECTRUM GIVEN TWO DATA MATRICES          */
/*-------------------------------------------------------------------------------------------------------------*/

void CrossSpec(float *Gxy, float **time_input_matrix_x,
        float **time_input_matrix_y, int rows, int columns)

{
      float freq_mat_x1[AVGS][FFT_SIZE];
      float freq_mat_x2[AVGS][FFT_SIZE];
      float freq_one_row_1[FFT_SIZE];
      float freq_one_row_2[FFT_SIZE];
      float conj_data[AVGS][FFT_SIZE];

      int r, c;
      int x, y;

      for(r = 0; r < rows; r++)
      {
        FftReal(freq_one_row_1, time_input_matrix_x[r], columns);
        FftReal(freq_one_row_2, time_input_matrix_y[r], columns);
```

```
        for(c = 0; c < (columns); c++)
          {
          freq_mat_x1[r][c] = freq_one_row_1[c];
          freq_mat_x2[r][c] = freq_one_row_2[c];
          }
      }


      /* CROSS SPECTRUM OPERATIONS */


      for(r=0; r < rows; r++)
      {
        for(c=0; c < (columns/2); c++)
          {
          conj_data[r][2*c] = (freq_mat_x1[r][2*c])*(freq_mat_x2[r][2*c])
                + (freq_mat_x1[r][2*c+1])*(freq_mat_x2[r][2*c+1]);

          conj_data[r][2*c+1] = (freq_mat_x1[r][2*c])*(freq_mat_x2[r][2*c+1])
                - (freq_mat_x1[r][2*c+1])*(freq_mat_x2[r][2*c]);

          }
      }

      /*MATRIX AVERAGING OPERATIONS */

      for(y = 0; y < columns; y++)
        Gxy[y] = 0.0;

      for(y = 0; y < columns; y++)
        {
          for(x = 0; x < rows;  x++)
            Gxy[y] = conj_data[x][y] + Gxy[y];
        }

      for(y = 0; y < columns; y++)
        Gxy[y] = (Gxy[y]) / ((float)rows);

}


/*-------------------------------------------------------------------------------------------------------------*/
/* CALCULATES THE ORDINARY COHERENCE GIVEN THE CROSS SPEC
AND TWO AUTO SPEC AS CALCULATED IN THEIR RESPECTIVE FUNCTIONS   */
/*-------------------------------------------------------------------------------------------------------------*/
```

```
void OrdCoherence(float *coherence, float *Gxy, float *Gxx, float *Gyy,
        int columns)

{
        int k;
        float output[FFT_SIZE/2];


        VectorMagSqrd(output, Gxy, columns);

        for(k = 0; k < (columns/2); k++)
          coherence[k] = (output[k]) / (Gxx[k] * Gyy[k]);


}


/*-------------------------------------------------------------------------------------------------------*/
/* VECTOR MAGNITUDE SQUARED FOR COMPLEX DATA IN THE FORM OF
REAL,IMAG,REAL,IMAG,... RETURNS A VECTOR HALF THE SIZE OF THE
ORIGINAL VECTOR                                                                    */
/*-------------------------------------------------------------------------------------------------------*/

void VectorMagSqrd(float *output_data, float *input_vector, int columns)
{


        int k;

        for(k = 0; k < (columns/2); k++)
        {

          output_data[k] = (input_vector[2*k] * input_vector[2*k])
            + (input_vector[2*k+1] * input_vector[2*k+1]);

        }

}



/*-------------------------------------------------------------------------------------------------------*/
/* PERFORMS THE CONJUGATE OPERATION ON A COMPLEX VECTOR AND
RETURNS A VECTOR OF THE SAME SIZE AS THE INPUT                                     */
/*-------------------------------------------------------------------------------------------------------*/

void VectorConjugate(float *output_vec, float *input_vec, int columns)
```

```
{
        /*ONLY TAKE COMPLEX DATA FORMAT AS AN ARGUEMENT */

        int i;
        for(i = 0; i < columns; i++)
          output_vec[i] = 0.0;

        for(i = 0; i < (columns/2); i++)
        {
          output_vec[2*i] = input_vec[2*i];
          output_vec[2*i+1] = -(input_vec[2*i+1]);
        }
}



/*-------------------------------------------------------------------------------------------------------*/
/* PARTIAL COHERENCE CALCULATION USING THE SHOWN INPUT DATA,
SHOULD BE INTERPRETED AS "PARTIAL COHERENCE OF SIGNAL A TO SIGNAL
B WITHOUT THE LINEAR EFFECTS OF SIGNAL C"                              */
/*-------------------------------------------------------------------------------------------------------*/


void PartialCoherence(float *part_coh_ab_c, float *Gaa, float *Gbb,
        float *Gcc, float *Gab, float *Gac,  float *Gcb, int columns)

{

        int k;
        float temp[FFT_SIZE], numer[FFT_SIZE];
        float denom[FFT_SIZE];
        float OrCo_c_a[FFT_SIZE], OrCo_c_b[FFT_SIZE], OrCo_a_b[FFT_SIZE];

        for(k = 0; k<(columns/2); k++)
        {
          temp[2*k]   = (Gab[2*k] * Gcc[2*k]) - (Gcb[2*k] * Gac[2*k]) +
            (Gcb[2*k+1] * Gac[2*k+1]);
          temp[2*k+1] = (Gab[2*k+1] * Gcc[2*k]) - (Gcb[2*k] * Gac[2*k+1]) -
            (Gcb[2*k+1] * Gac[2*k]);
        }

        VectorMagSqrd(numer, temp, columns);

        OrdCoherence(OrCo_c_a, Gac, Gcc, Gaa, columns);
        OrdCoherence(OrCo_c_b, Gcb, Gcc, Gbb, columns);
        OrdCoherence(OrCo_a_b, Gab, Gaa, Gbb, columns);
```

```
/* THE CONDITIONALS PREVENT INFINITY NUMBERS WHEN THE PARCO IS
UNDEFINED - I.E. THE MODEL MUST BE WELL DEFINED FOR THE PARTIAL
COHERENCE CALCULATION TO BE MEANINGFUL */

        for(k = 0; k< (columns/2); k++)
        {

          denom[k] = Gcc[k] * Gcc[k] * Gaa[k] * Gbb[k] * (1.0 - OrCo_c_a[k]) *
            (1.0 - OrCo_c_b[k]);

          part_coh_ab_c[k] = numer[k] / denom[k];

          if (OrCo_c_a[k] < 0.1)
            part_coh_ab_c[k] = (OrCo_a_b[k])/(1.0 - OrCo_c_b[k]);

          if (part_coh_ab_c[k] > 1.0)
            part_coh_ab_c[k] = OrCo_a_b[k];

          if (part_coh_ab_c[k] < 0.0)
            part_coh_ab_c[k] = 0.0;

        }

}

/*-------------------------------------------------------------------------------------------------------*/
/* RESIDUAL CROSS SPECTRAL DENSITY CALCULATION CAN ONLY BE USED
FOR YY WITHOUT ONE OR TWO BECAUSE THE OTHERS ARE DEFINED
DIFFERENTLY                                                                                          */
/*-------------------------------------------------------------------------------------------------------*/


void ResidCSD(float *Gyy_x, float *Gyy, float *ordcoh_x_y, int columns)
{
        int k;

        for(k = 0; k < (columns/2); k++)
          Gyy_x[k] = Gyy[k] * (1.0 - ordcoh_x_y[k]);
}




/*-------------------------------------------------------------------------------------------------------*/
/* THIS ROUTINE TAKES ALL THE PREVIOUSLY CALCULATED NECESSARY
DATA, DETERMINES THE POSSIBLE PATH DIRECTION AND CALCULATES THE
```

```
/* COHERENT OUTPUT POWER FOR EACH OF THE INPUTS AT EACH OF THE
SEPEARATE FREQUENCY BINS. THESE ARE DEFINED AS THE COP FILTERS IN
THE FREQ DOMAIN                                                    */
/*-------------------------------------------------------------------------------------------------------*/


void FreqWeightDesign(float *w_hats_1, float *w_hats_2, float *C_1_y,
        float *C_2_y,float *G_yy, float *PC_1y_2, float *PC_2y_1, float *G_yy_1,
        float *G_yy_2, int columns)

{
        int k;
        float path_direction[FFT_SIZE];

        for(k = 0; k < (columns/2); k++)
        {
          path_direction[k] = C_1_y[k] - C_2_y[k];

          if(path_direction[k] >= 0.0)
           {
             w_hats_1[k] = (C_1_y[k] * G_yy[k]);
             w_hats_2[k] = (PC_2y_1[k] * G_yy_1[k]);
           }
          else
           {
             w_hats_1[k] = (PC_1y_2[k] * G_yy_2[k]);
             w_hats_2[k] = (C_2_y[k] * G_yy[k]);
           }

        }

}


/*-------------------------------------------------------------------------------------------------------*/
/*THIS TAKES THE FREQUENCY DOMAIN COP FILTERS AND CONVERTS THEM
BACK TO THE TIME DOMAIN ASSUMING ZERO PHASE.  SOME SMALL PHASE
WILL RESULT BUT DOES NOT AFFECT ANSWER.                            */
/*-------------------------------------------------------------------------------------------------------*/

void FreqToFIR(float *w_fir_1, float *w_fir_2, float *w_fft_1, float *w_fft_2,
        int columns)

{

        int k, x;
```

```
        for(k = 0; k<(columns/2); k++)
          {
          w_fir_1[2*k] = w_fft_1[k];
          w_fir_1[2*k+1] = 0.0;

          w_fir_2[2*k] = w_fft_2[k];
          w_fir_2[2*k+1] = 0.0;
          }

        lfftInReal(w_fir_1, columns);
        lfftInReal(w_fir_2, columns);


}


/*-------------------------------------------------------------------------------------------------------*/
/*END                                                                                                    */
/*-------------------------------------------------------------------------------------------------------*/
```

VITA

Michael Vaudrey was born in Franklin, Virginia on April 26, 1972. After moving to Greenville, South Carolina and again to Columbia, he achieved the rank of Eagle Scout and graduated with high honors from Spring Valley High School. Mike spent his first two summers after high school graduation working as a manufacturing shop hand at Patterson Fan Company. Beginning his Bachelor's degree in Mechanical Engineering at Virginia Tech in 1990, he spent three semesters in school before starting a co-op program with Michelin Americas Research and Development Corporation. After four semesters of work and five more semesters of class, Mike graduated Magna Cum Laude in 1995. Immediately following graduation he began his graduate program in Mechanical Engineering also at Virginia Tech. Nineteen short months later he successfully defended his thesis. His future plans include marraige and following the teachings of a great philosopher who once said "Try not. *Do, do.* Or do not. There is no try."