SIMULATION OF AN ALGORITHM FOR THE
ACTIVE CONTROL OF COMBUSTION NOISE

by

Jean-Yves Capelle

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:

[Signatures]

J. Robert Mahan, Chairman

Uri Vandsburger

Larry A. Roe

Richard J. Roby

December, 1990
Blacksburg, Virginia
SIMULATION OF AN ALGORITHM FOR THE
ACTIVE CONTROL OF COMBUSTION NOISE

by
Jean-Yves Capelle
J. Robert Mahan, Chairman
Mechanical Engineering

(ABSTRACT)

The simulation of an algorithm for the active control of combustion noise generated by a turbulent flame produced by an open burner is developed in this thesis. The simulation includes (1) an autoregressive model of real combustion noise, (2) a feedback loop based on the "observer" method, (3) a model of the transfer function between the acoustic driver and the sensor through the flame, and (4) a method to take into account the time-delay due the calculation of the algorithm. A practical implementation of the control strategy is also proposed.

An attenuation of up to 40 dB is obtained in the 0 - 3000 Hz band, decreasing with the time-delay required for creation of the feedback signal. The influence of the order of the autoregressive model is studied, and it is shown that better results are obtained by increasing the order. The choices for the location of the activator and for the type of sensor are investigated and discussed. Further analytical research on the method of computing the feedback signal in the presence of time-delay is identified.
Acknowledgements

I would like to express my appreciation to my advisor, Dr. J. Robert Mahan, for his advice and guidance during this research.

I would also like to thank my committee members, Dr. Uri Vandsburger, Dr. Larry A. Roe and Dr. Richard J. Roby for their support and helpful suggestions.

I am grateful to the company SNECMA, France, for sponsoring my studies here, and especially to Mr. Michel Desaulty, Research Division in Combustion, Villaroche Center, for his support.

Many thanks to Dr. Harry H. Robertshaw, Dr. Alfred L. Wicks and Dr. Christopher R. Fuller for their advice during my research.

I would like to thank all my friends for the good time I had at Blacksburg; Andreas, Remi, Ari, Laurent, Jeff, Cathy, Greg and many others.

I would like to express special thanks to my family to be so close to me when I was so far.

Finally, I greatly thank Sylvie for everything, and many, many other things...!
# Table of Contents

1.0 Introduction .............................................................................. 1

1.1 Review of Previous Work ......................................................... 2

1.2 Goals of the Present Research .................................................. 8

2.0 Autoregressive Model for a Deterministic Signal ......................... 11

2.1 Theory ................................................................................... 11

2.2 Example .................................................................................. 14

3.0 Initial Remarks for the Control of Combustion Noise ..................... 16

3.1 On the Use of an Autoregressive Model ...................................... 16

3.2 Use of the LMS method ......................................................... 17

4.0 Basis of the Observer Method .................................................. 20

4.1 Formulation of the Control Loop ............................................. 20

5.0 Application of The Observer Method as a Controller ..................... 26

5.1 Method ............................................................................... 26

Table of Contents iv
List of Illustrations

Figure 1. Cancellation of Repetitive Noise [1].......................... 3
Figure 2. Block Diagram of the LMS Method............................. 5
Figure 3. Sketch (a) and performance (b) of the Dines' controller..... 9
Figure 4. Discrete Values of a Signal..................................... 12
Figure 5. LMS Control with an AR Model................................. 19
Figure 6. Block Diagram of a General System............................ 22
Figure 7. Block Diagram of the Observer Method........................ 24
Figure 8. Control of One Harmonic Signal, Time Domain.............. 31
Figure 9. Control of One Harmonic Signal, Frequency Domain........ 32
Figure 10. Control of One Harmonic Signal + Random Noise, Time Domain ... 33
Figure 11. Control of One Harmonic Signal + Random Noise, Frequency Domain 34
Figure 12. Control of Five Harmonic Signals, Time Domain............. 36
Figure 13. Control of Five Harmonic Signals, Frequency Domain....... 37
Figure 14. Control of Combustion Noise at 25600 Hz, Frequency Domain.. 39
Figure 15. Control of Combustion Noise at 12800 Hz, Frequency Domain... 40
Figure 16. Schematic of an Entire Representation of the Control System. 42
Figure 17. Block- Diagram of an Entire Control System.................. 43
Figure 18. Block-Diagram of an Entire Control System with Physical Elements ... 45
Figure 19. Complete Block-Diagram of an Entire Control System......... 46
Figure 20. Sketch of the Experimental Burner [20]. ........................................ 50
Figure 21. Experimental Result [20] and Model of a Flame Transfer Function. ... 51
Figure 22. Control of Combustion Noise with Time Delay (n = 1, p = 1) .......... 63
Figure 23. Control of Combustion Noise with Time Delay (n = 2, p = 1) .......... 64
Figure 24. Control of Combustion Noise with Time Delay (n = 3, p = 1) .......... 65
Figure 25. Control of Combustion Noise with Time Delay (n = 1, p = 2) .......... 66

List of Illustrations
List of Tables

Table 1. Time-Delay Relative to the Number of Coefficients n (IBM 3084) . . . . 56
List of Symbols

**English**

A  matrix representing a system; relates the vector $X_{k+1}$ to the vector $X_k$ (-).

$a_i$  coefficients for an autoregressive model (-).

B  matrix representing the influence of the input $U_k$ on the vector $X_{k+1}$ (-).

C  matrix representing the relationship between the vector $X_k$ and the output $y_k$ of the system in a control scheme (-).

D  matrix for the vectorial representation of an autoregressive model (-).

d  signal produced by a system (-).

$\Delta t$  sample interval (s).

$F_c$  discretization frequency (Hz).

G  gain of an amplifier (-).

$G_a$  amplitude of the gain of an amplifier (-).

H  observer column matrix in a control scheme, relating the error ($y_k - \hat{y}_k$) to the estimate $\hat{X}_{k+1}$ (-).
\( h_i \)  components of the matrix \( H \) (-).
\( I \)  matrix identity (-).
\( i \)  index of the time step (-).
\( k \)  index of the time step (-).
\( l \)  refers to the order of a regressive model and of an autoregressive model (-).
\( m \)  index of the time step (-).
\( n \)  order of the regressive model and order of the autoregressive model (-).
\( p \)  order of the step delay (-).
\( Q(z) \)  complex polynomial (-).
\( r \)  refers to the order of a regressive model and of an autoregressive model (-).
\( T \)  model of the flame transfer function (-).
\( T^{-1} \)  inverse of \( T \) (-).
\( U_i \)  discrete value of the input in a control scheme (-).
\( v \)  signal input in the acoustic driver (-).
\( v_k \)  discrete value of the signal \( v \) (-).
\( \hat{v} \)  estimate of the signal \( v \) (-).
\( \hat{v}_k \)  discrete value of the signal \( \hat{v} \) (-).
\( W^p \)  coefficient for a regressive model with the Least Mean Square method (-).
\( X \)  reference signal for the Least Mean Square method (-).
\( X_k \)  discrete value of the signal \( X \), or vector of different independent variables of a system in the control scheme (-).
\( \hat{X}_k \)  estimate of \( X_k \) in the control scheme (-).
\( x \)  time series, real signal produced by a system (-).
\( x_k \)  discrete value of the signal \( x \) (-).
\( Y \)  signal produced by a system (-).
\( Y_k \)  discrete value of the signal \( Y \) (-).
y  signal produced by a system (-).
\( y_k \)  discrete value of the signal \( y \) (-).
\( \hat{Y} \)  estimate of the signal \( d \) or \( Y \) (-).
\( \hat{Y}_k \)  discrete value of the estimate \( \hat{Y} \) (-).
\( \hat{y} \)  estimate of the signal \( d \) or \( y \) (-).
\( \hat{y}_k \)  discrete value of the estimate \( \hat{y} \) (-).
\( z \)  complex variable (-).

Greek

\( \alpha_k \)  discrete value of the random parameter in an autoregressive model (-).
\( \varepsilon \)  error signal which is the difference between the real signal \( Y \) (or \( d \)) and its estimate \( \hat{Y} \) (-).
\( \varepsilon_k \)  discrete value of the signal \( \varepsilon \) (-).
\( \frac{1}{\omega_\varepsilon} \)  time constant of an amplifier (s).
1.0 Introduction

It is well known that the noise produced by flow systems increases with combustion, and that the greater the turbulence is, the noisier are the systems. Turbulence is needed to reduce the size of a combustion chamber by improving the mixing of fuel and air with the hot products, and so combustion noise is inevitable. But the pressure oscillations which occur at low frequency can produce structural damage, among other problems. Therefore, special design of the system leading to limitation in its dynamic behavior can avoid disastrous mechanical consequences. Active acoustic control of the combustion noise may contribute to reduce these consequences. This thesis describes the simulation of an active control scheme. This control scheme is developed for an open burner, in order to simplify the problem.

The presentation of this thesis is divided in four major points. First, a general review of what has been done in the field of active control of noise, and especially in the case of combustion noise produced by an open burner, establishes the basis of this study. Then the principles for the active control of combustion noise as described in this thesis are developed, followed by a numerical simulation which applies these principles.
Finally, a more accurate simulation, taking into account more details in the feedback loop for the active control, is described and demonstrated with real combustion noise.

1.1 Review of Previous Work

Active noise control involves the reduction or cancellation of unwanted sound by using acoustic drivers to modify the acoustic field. Many ways exist to accomplish this, depending on the nature of the unwanted sound.

Synchronizing has been used effectively in the active minimization of periodic noise such as that produced by turboprop engines. In this method, a synchronizing signal (taken from the rotation of the axle of the engine for instance) is sent to the active control system. Therefore, the system gets an accurate prediction of the next period of the signal from the current period. This predicted signal is then injected out of phase into the signal path to reduce the noise.

Chaplin [1] used this method to reduce diesel engine noise, as shown schematically in Fig. 1. In the first step, the microprocessor takes into account the residual sound which gives a measure of how accurate the prediction of the exhaust noise is. Thereafter, once the residual sound has been minimized, the parameters for the prediction are fixed and the microphone can be removed. The same principle is also used for the active control of the noise produced by the exhaust system of an automobile [2,3]. The same synchronizing method is also used to reduce the blade passing tone of centrifugal fans [4], with the speaker synchronized with the fan rotation. Reductions in the level of blade tone component of up to 23 dB were observed. Other methods of control have been demonstrated without requiring periodicity of the noise, as explained later on.
Figure 1. Cancellation of Repetitive Noise [1].
Adaptive filtering is the implementation of a model of the unwanted sound, which is periodically adapted to track changes in the system. Adaptive filtering is the most generally effective method of active noise control. It has the advantages of automatic minimization and automatic adjustment in the system over time.

An adaptive filter is created by using two microphones. The first microphone is placed upstream (with respect to sound propagation) from the loudspeaker to sample the unwanted signal. The other microphone is placed downstream of the speaker to sample the error signal. The error signal is simply the result of the addition of the unwanted signal with the control signal. It should ideally be zero. The adaptive filter creates a model of the unwanted signal and uses it to drive the loudspeaker. The information obtained by the error microphone is fed back to the adaptive filter and the model is adapted to minimize the new error.

Ross [5,6,7] who developed this adaptive control method, obtained a reduction of about 20 dB in the 25-350 Hz band. This same method has been applied to the computer simulation of the sound control in a duct [8], and to the experimental control of vibration in a beam [9].

Most adaptive filtering methods use the so-called Least Mean Square (LMS) algorithm to minimized the error signal, as described by Sommerfeldt [10] for a mechanical system. Implementation of the LMS method is shown schematically in Fig. 2.

In Fig. 2, a system (an engine exhaust system for instance) driven by an input signal and producing the signal $d$ is considered. This system is modeled with a digital filter generating an estimate $\hat{y}$ of the signal $d$. This signal $\hat{y}$ is created by means of a regressive model, using a reference signal $X$ which is entirely correlated with the input (it could be the input itself), even if we do not know the transfer function between the input and this reference signal.
Figure 2. Block Diagram of the LMS Method.
The coefficients $W_i$ used in the regressive model are time dependent, because they change in order to minimize the error signal $e$ between the signal $d$ and the estimated signal $Y$.

Let $\Delta t$ be the sample interval such that $Y = Y(i\Delta t)$, $X_j = X(j\Delta t)$, the coefficient $W_i = W(i\Delta t)$ and $e = e(i\Delta t)$. Then the digital filter generates $Y$ using the equation of order $n$,

$$Y_t = W_1^iX_i + ... + W_n^iX_{i-n}, \quad (1.1.1)$$

where

$$e_t = Y_t - d_t. \quad (1.1.2)$$

The LMS method minimizes the error signal $e$ by adapting in time the coefficients $W_i$. This algorithm uses an estimate of the differential of the square of the error [10], and yields

$$W_r^{i+1} = W_r^i - \mu(2e_iX_i), \quad 1 < r < n, \quad (1.1.3)$$

where $\mu$ is an experimental parameter to influence the convergence of the system. Therefore, the coefficients of the digital filter "adapt" themselves in the time domain in such a way that the error is minimized. The restriction for the stability and convergence of this method is that the reference input $X$ must not contain any information from the output $Y$; otherwise, the value of the term $(e,X)$ in Eq. (1.1.3) will be modified.

In the case of adaptive filtering described above, the system being controlled has the advantage of being amenable to active control. That is, there is sufficient time, before the noise recorded by the first microphone (described earlier in the adaptive filtering paragraph) reaches the anti-sound source (a loudspeaker for example), for the digital filter to calculate an estimate of the control signal. In the case of flame noise, there is
no such advantage, because no reference signal is available other than the flame noise itself. Therefore, there is insufficient time for the computer to calculate an appropriate control signal.

Since turbulence is essentially a random process, the combustion noise generated by a turbulent open burner will cover a wide spectrum of frequencies. Bragg [11] assumes for the nature of combustion noise that the radiated acoustic power is proportional to the mean square of the rate of change of volume evolution by combustion heat release. His resulting model, which predicts that the broadband noise will be centered around 500 Hz for typical values of the parameters, is too simple to predict combustion noise for the wide range of parameters encountered in practical burners.

In 1987, Mahan [12] developed a theoretical model for the combustion noise produced by an open burner. He assumes that the flame is a statistically stationary compact noise source. This implies a physical model in which a stream of turbulent eddies of varying size enters the combustion zone in a random order, but with a statistical size distribution. Mahan assumed that the initial diameters of the eddies were normally distributed about a specified mean value with a specified standard deviation. A spectral shape typical of those observed for actual open turbulent flames is obtained by means of this model. This model has been improved [13,14] to take into account a continuous phenomenon when the eddies enter the combustion zone. While Mahan's models seem to predict the combustion noise spectrum, they are incapable of producing a pressure time series suitable for use as a reference signal for one of the active noise control strategies discussed above. The interested reader is referred to Mahan's NASA Contractor Report [15] for an extensive review of the combustion noise literature up to 1984.

To control turbulent combustion noise, we have only the combustion noise itself. We know that the most significant characteristic of combustion noise detected in the
acoustic far field is its predominately low-frequency (below 3000 Hz) content. Dines [16] describes an interesting development in which the light emission from an open flame was monitored and used as a reference signal to characterize the simultaneously produced combustion noise. A controller adapted this signal to generate, at a speaker near the burner, the antidote of the combustion noise, using an error signal taken from a microphone in the far field. Figure 3, which summarizes the operation and performance of the device, is taken from Dines’ Cambridge doctoral dissertation.

He obtained a 10 dB attenuation over the band 300-700 Hz with a peak attenuation of 17 dB at 300 Hz. Because of the relative location of the burner and the speaker, the attenuation can be obtained only in a certain region where the sound waves coming from the two sources interfere. Moreover, the speaker must generate the same acoustic energy as the open burner. In the case of an engine this approach would require an excessively powerful speaker. We are looking for a method which may be suitable in this kind of configuration, but which would not require such a high-powered control signal.

1.2 Goals of the Present Research

The main assumption for this thesis is that the combustion noise has some deterministic content. The large eddies produced by an open burner depend on the diameter of the burner and the flow rate for example. These large eddies are related to the low frequency content of the combustion noise. This has been studied by Mahan and his students [13,14].

The purpose of the work reported in this thesis is two-fold:
Figure 3. Sketch (a) and performance (b) of the Dines' controller: ADC, analogue-to-digital converter; DAC, digital-to-analogue converter; PM, photomultiplier; A is the anti-sound source; $F_0$ is the transfer function of the anti-aliasing filters; L is the light signal from the flame; S is the sound output from the flame; $T_1$ and $T_2$ are the transfer functions between the flame and the photomultiplier, and the flame and the microphone, respectively; $T_3$ is the transfer function between the anti-sound loudspeaker and the microphone; $T_d$ is the desired transfer function, $T_d = -ST_1LRT_2T_3$ [16].
• Specification of a configuration for the control actuator, burner, error-signal sensor and computer, which is more compact and exerizes control in a more uniform space, and which is able to work in a hot zone as in the combustion chamber of an aircraft engine

• to find a method of active control taking into account the previous point, and to perform a simulation of the control of combustion noise using a computer.

Bearing in mind that only the combustion noise itself is available as a reference, we begin with the study of a digital filter based on an autoregressive model of the acoustic signal produced by a turbulent flame.
2.0 Autoregressive Model for a Deterministic Signal

The purpose of this chapter is to describe the theory for an autoregressive model for a deterministic signal, and to give an example to illustrate this theory.

2.1 Theory

The fundamental premise of the autoregressive (AR) model is that a process, represented by a time series $x_i$, can be modeled based on the previous values of this process. Thus, it assumed that the time series is at least partially deterministic. Discrete values of a signal are considered, as shown in Fig. 4.

The quantity $\Delta t$ is the sample interval such that $x_i = x(i\Delta t)$. We can write

$$x_i = a_1x_{i-1} + a_2x_{i-2} + \ldots + a_nx_{i-n} + a_i,$$

(2.1.1)
Figure 4. Discrete Values of a Signal.
where \( \alpha \) is a random value to account for statistical error in prediction. Equation (2.1.1) is referred to an AR model of order \( n \).

We introduce the vectorial notation, with \( N \) greater than the order \( n \),

\[
\begin{bmatrix}
    x_l \\
    \vdots \\
    x_{l+N}
\end{bmatrix}
= 
\begin{bmatrix}
    x_{l-1} & \cdots & x_{l-n} \\
    \vdots & \ddots & \vdots \\
    x_{l+N-1} & \cdots & x_{l+N-n}
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    \vdots \\
    a_n
\end{bmatrix}
= [D] 
\begin{bmatrix}
    a_1 \\
    \vdots \\
    a_n
\end{bmatrix}
\tag{2.1.2}
\]

The quantity \( N \) can be equal to \( n \), but the results for the coefficients will be more accurate for a large \( N \), because \( N \) represents the number of points which are going to be averaged, as shown in Eq. (2.1.3). Therefore, the random noise included in the time series \( x \), will be reduced by use of this average, with \( N \) sufficiently large. The order of the matrix \( D \) is \((N+1)n\), that is, the matrix \( D \) is not square. Therefore, in order to get the coefficients, we use the following technique. The first step is to multiply by the transpose of matrix \( D \), and we get a square matrix for the system

\[
\begin{bmatrix}
    l + N \\
    \vdots \\
    l + N
\end{bmatrix}
= 
\begin{bmatrix}
    \sum_{j=1}^{l+N} x_j x_{j-1} \\
    \vdots \\
    \sum_{j=1}^{l+N} x_j x_{j-1}
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    \vdots \\
    a_n
\end{bmatrix}
\tag{2.1.3}
\]

For the second step, the coefficients can be obtained by inversion of Eq. (2.1.3) and by using the time series, \( x \).
The interest of parametric models like this one has been intense in recent years because of their potential for absence of leakage (problem encountered in frequency analysis when the discretization frequency is not a multiple of the different frequencies of the signal). Moreover, they have an extremely high-frequency resolution, independent of the sampling interval, because the calculation of the coefficients of the AR model takes into account this sample interval, through the values of the time series used in Eq. (2.1.3).

There are two significant disadvantages to the application of this parametric modeling technique. The most unfavorable of these is the problem of estimation of the model order $n$ which is the most accurate. This order must be chosen at the beginning of the analysis. If the order is too low, the resulting spectrum estimates neglect some part of the signal content. If the order is too high, the resulting spectrum estimates add some nonexistent components to the true signal content. The model order estimation requires subjective judgment and some guesswork based on the experience.

The second disadvantage is that the parametric modeling technique is more computationally complex than techniques based upon Fast Fourier Transform (FFT) analysis. Signal data need to be processed at speeds commensurate with that of data acquisition; in other words, the analysis must be performed in real time. This problem will decrease with the availability of more powerful computers.

2.2 Example

Consider the trivial signal $y(t) = e^{−t}$. We want to obtain an autoregressive model of this deterministic signal. We could use the method described in the previous chapter.
in which data representing the signal $y(t)$ are created, but in this case, a more direct method may be used. We write

$$y[i\Delta t] = D_y[(i - 1)\Delta t], \tag{2.2.1}$$

and using the expression of $y(t)$, we obtain $D = e^{-\alpha}$. 
3.0 Initial Remarks for the Control of Combustion Noise

The purpose of this chapter is to make some remarks on the use of an autoregressive model and on the use of the LMS method for the control of combustion noise.

3.1 On the Use of an Autoregressive Model

Because, by definition, a truly random noise cannot be predicted, we will be able to control only the deterministic components of the combustion noise. In the case of a noise produced by a machine, or a vibrating beam, we know the source of the noise, which corresponds to some driving signal (as explained in Chapter 1). In our case, we have some parameters like the flow rate, the burner diameter, the flame speed and the expansion ratio which influence the combustion noise. These different factors have been used in a series of models developed by Mahan and his students [12,13,14], and the
overall sound power level is well predicted. But we do not yet have a model for the pressure disturbance versus time based on these parameters which is adequate for use as a reference signal for an adaptive control system, as described in Chapter 1. Therefore, we have decided to use an autoregressive model of the combustion noise. This model is obtained by the method described in Chapter 2, with a burner creating the combustion noise and a microphone or a photomultiplier providing the needed time series.

3.2 Use of the LMS method

The LMS method described in Chapter 1 is commonly used in the acoustic laboratory of the Mechanical Engineering Department at Virginia Tech. Although we do not have a reference signal, we could modify the LMS algorithm to take into account an autoregressive model. But the remark made in Chapter 1 about the stability of this method, and the results given below show that we cannot use the LMS method with this kind of model.

Consider the trivial signal \( y(t) = \tau \). We call the estimate of this signal \( \hat{y}(t) \). In a discrete formulation, with a sample interval \( \Delta t \) such that \( \dot{y}_i = \dot{y}(i\Delta t) \), we write an AR model of \( \dot{y}(t) \)

\[
\dot{y}_i = a_1\dot{y}_{i-1} + a_2\dot{y}_{i-2},
\]

(3.2.1)

and the discrete-time expression of \( y(t) \) is written

\[
y_i = i\Delta t.
\]

(3.2.2)
Then using Eq. (3.2.2) to find the coefficients $a_1$ and $a_2$, we obtain

$$a_1 = 2, \quad a_2 = -1.$$  

(3.2.3)

In Fig. 5, we see that whatever the initial conditions for the estimate $\hat{y}$ are, the estimation of the signal $y(t)$ diverges. This proves that we cannot use the LMS method with an autoregressive model of the combustion noise which will have some harmonic content, referring to the assumption chosen for this work that the combustion noise has some deterministic content. Therefore, we must look for another method for the active control of combustion noise.
Figure 5. LMS Control with an AR Model: symbol (A) represents the application of the LMS method with the exact initial conditions ($\hat{y}_0 = 0, \hat{y}_1 = \Delta t$); symbol (B) represents the application of the LMS method with different initial conditions ($\hat{y}_0 = 1, \hat{y}_1 = 1$).
4.0 Basis of the Observer Method

The purpose of this chapter is to describe the basis of the observer method, which is the core of the control loop.

4.1 Formulation of the Control Loop

Bearing in mind that signal processing will lie at the heart of the control scheme, and that we are dealing with the mathematical formulation of an estimate of the control signal, we should describe the method for discrete signals. We use the representation of the equations defining a system in the z-plane (see Appendix A).

Consider a system which can be represented by the mathematical formulation of its evolution in time,

\[ X_{k+1} = AX_k + BU_k \]  \hspace{1cm} (4.1.1)

and
\[ y_k = CX_k. \] (4.1.2)

The quantities \( X_k \) and \( X_{k+1} \) are vectors of the different independent variables of the physical system at time steps \( k\Delta t \) and \( (k+1)\Delta t \), respectively. As a common example, the two components of these vectors \( X_k \) and \( X_{k+1} \) could be the spatial position and the speed of a mechanical device. The quantity \( A \) is the matrix modeling the evolution in time of the system, relating the vector \( X_{k+1} \) to the vector \( X_k \). For the above example, \( A \) represents the matrix of the mathematical system relating the speed and the position of the mechanical device at step \( (k+1)\Delta t \) with the speed and the position at step \( k\Delta t \). The variable \( U_k \) represents the input to the system, which could be a driving force for the example. The input \( U_k \) is a single value, whereas the variable \( X_k \) is a vector. Therefore, the input is multiplied by a column matrix which gives the weight of this input for every component of the vector \( X_k \). The variable \( y_k \) represents the output of the physical system, which could be the speed in the above example. Therefore, the matrix \( C \) represents the relationship between the output \( y_k \) and the vector \( X_k \). In the case of the example, \( y_k \) is equal to the second component of the vector \( X_k \). Figure 6 is a block diagram of such a system.

In fact, because we model the physical system, estimates \( \hat{X}_k, \hat{X}_{k+1} \), and \( \hat{y}_k \) of the real variables \( X_k, X_{k+1} \), and \( y_k \) are used. We use the same notation for the matrix \( A \) because it represents a model of the system. Therefore, \( A \) is already an estimate of the real system. In order to obtain as accurate an estimate of the output \( \hat{y}_k \) as possible, the error \( y_k - \hat{y}_k \) is used in a feedback loop with a matrix \( H \) to estimate \( \hat{X}_k \). But what is this matrix \( H \) and how is it determined? This question is answered in the next paragraph. The error \( y_k - \hat{y}_k \) represents the validity of the mathematical model \( A \) relative to the physical system. For example, an error of zero means that \( A \) is a perfect model.
Figure 6. Block Diagram of a General System.
The purpose of the feedback loop is to "inform" the model of the importance of this error through the matrix $H$. In other words, this difference represents how accurately the mathematical system represented by $A$ and $H$ "observe" the system. Thus this method is called an observer method. Just as the matrix $B$ relates the input $U_k$ to the estimate $\hat{X}_{k+1}$, the matrix $H$ relates the error $y_k - \hat{y}_k$, which is a single value, to the estimate $\hat{X}_{k+1}$. In the case of the earlier example, the components of the matrix $H$ will represent the weights for the error $y_k - \hat{y}_k$ in order to modify the estimate of the position and the speed, $\hat{X}_{k+1}$. This new feedback loop is written mathematically

$$\hat{X}_{k+1} = A\hat{X}_k + BU_k + H(y_k - \hat{y}_k),$$

(4.1.3)

and its block diagram is shown in Fig. 7.

It can be shown [17] that the fastest matrix $H$ in terms of tracking and converging speed to the true signal $y_k$ is obtained with a so-called "deadbeat" observer, which means that the matrix $H$ is chosen such that

$$\det(zI - A + HC) = Q(z) = z^n,$$

(4.1.4)

with $n$ the order of the system. Because we are dealing with the $z$-transform, the roots of the polynomial $Q(z)$, which represents the determinant of the transfer function between the real signal $y_k$ and the estimator $X_k$ [17], must be inside the unit circle in the $z$-plane to ensure the stability of the system. Furthermore, reference 17 establishes that the matrix $H$ can be written

$$H = \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} = \mathcal{Q}(A) \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

(4.1.5)

Basis of the Observer Method
Figure 7. Block Diagram of the Observer Method.

Basis of the Observer Method
with $Q(A) = A^*$. 

We are really concerned about the speed with which this method tracks the signal, because in the case of combustion noise the resulting signal $y_k - \hat{y}_k$ must be minimized. This error will ultimately represent the difference at the sensor location between the deterministic component of the disturbance entering the flame zone which leads to combustion noise and the pressure fluctuation imposed by the acoustic driver. Therefore, the estimator $\hat{y}_k$ must follow as closely as possible the true signal $y_k$. 

The basic idea of this observer method is to provide an adaptive model of the real system, the matrix $A$ with the feedback loop through the matrix $H$, which gives better results than the matrix $A$ alone.
5.0 Application of The Observer Method as a Controller

The purpose of this chapter is to develop a method based on the observer method described in Chapter 4 to perform an active control of any signal. This method is then tested for harmonic signals and for real combustion noise.

5.1 Method

We use an autoregressive model of open combustion noise. This model is found by using a data processing analyser, which produces the coefficients $a_1, \ldots, a_n$, from burner data obtained with a microphone or a photomultiplier. The order of the regression $n$ must be guessed, as explained in Chapter 2.
In agreement with the parametric techniques developed in Chapter 2, we have assumed that some components of the combustion noise are deterministic. The autoregressive model is valid only for a deterministic signal.

Finally, we have an AR model of the noise produced by an open turbulent flame,

\[
\hat{x}_{k+1} = a_1 \hat{x}_k + \ldots + a_n \hat{x}_{k+1-n},
\]

(5.1.1)

with \( \hat{x} \) representing the pressure fluctuation and \( k \) and \( n \) defining the time step; that is, \( \hat{x}_k = \hat{x}(k) \Delta T \).

Using vectorial notation,

\[
\hat{X}_{k+1} = \begin{bmatrix}
\hat{x}_{k+1} \\
\vdots \\
\hat{x}_{k+2-n}
\end{bmatrix}
\quad \text{and} \quad
\hat{X}_k = \begin{bmatrix}
\hat{x}_k \\
\vdots \\
\hat{x}_{k+1-n}
\end{bmatrix},
\]

(5.1.2)

and we can write \( \hat{X}_{k+1} = A \hat{X}_k \), where

\[
A = \begin{bmatrix}
a_1 & a_2 & \ldots & a_{n-1} & a_n \\
1 & 0 & \ldots & 0 & 0 \\
0 & 1 & \ldots & \ldots & \ldots \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
0 & \ldots & \ldots & 0 & \ddots \\
0 & \ldots & \ldots & 1 & 0
\end{bmatrix}.
\]

(5.1.3)

In our case, using the same notation as in the observer method of Chapter 4, the output variable of the system is \( \hat{y} \), which is the first component of the vector \( \hat{X}_k, \hat{x}_k \), and which can be written

*Application of The Observer Method as a Controller*
\[ \hat{y}_k = C\hat{X}_k \quad \text{with} \quad C = (1,0,\ldots,0). \quad (5.1.4) \]

As discussed in Chapter 4, the matrix \( H \) can be calculated using Eq. (4.1.5), and the final formulation for the model is given by

\[ \hat{X}_{k+1} = A\hat{X}_k + H(y_k - \hat{y}_k), \quad (5.1.5) \]

which is identical to Eq. (4.1.3) except that now there is no input \( U_k \).

In the case of combustion noise, we are interested in the result \( y_k - \hat{y}_k \), which now represents the addition of the true combustion noise pressure signal \( y_k \) to the contribution due to the control signal \( -\hat{y}_k \) produced by an acoustic driver. In this formulation we have temporarily set the flame transfer function between the acoustic driver to unity. We can notice that at step \( (k+1) \) for the components of the vector \( X_{k+1} \), all the previously computed values \( \hat{x}_k, \ldots, \hat{x}_{k+2-n} \) are modified by the term \( H(y_k - \hat{y}_k) \). These new values, \( \hat{x}_k, \ldots, \hat{x}_{k+2-n} \), will be used in calculating the vector \( X_{k+2} \) for the next step, in particular the component \( \hat{x}_{k+2} \). This new series facilitates tracking the true value, \( x_{k+2} \).

The program OBS2 (Listing 4 in Appendix C) performs this function.

### 5.2 Application to Harmonic Signals

This section deals with the application of the control method to one harmonic function, with or without a random component added, and to a sum of harmonic functions.
5.2.1 For One Harmonic Function

Consider the function \( x(t) = \cos(\omega t) \). In discrete notation, we have \( x_k = x(k \Delta t) = \cos(k \omega \Delta t) \). The order of the regression is assumed to be two, because the function can be completely described by its frequency and its amplitude. So it can be written, for \( k \geq 0 \),

\[
x_{k+2} = a_1 x_{k+1} + a_2 x_k.
\]  
(5.2.1.1)

Using the complex notation for the cosine and introducing it into Eq. (5.2.1.1), with \( \beta = \omega \Delta t \), we get for every \( k \)

\[
e^{j\beta(k + 2)} + e^{-j\beta(k + 2)} = a_1(e^{j\beta(k + 1)} + e^{-j\beta(k + 1)}) + a_2(e^{j\beta k} + e^{-j\beta k}) \]  
(5.2.1.2)

and

\[
e^{j\beta k}(e^{j2\beta} - a_1 e^{j\beta} - a_2) + e^{-j\beta k}(e^{-j2\beta} - a_1 e^{-j\beta} - a_2) = 0. \]  
(5.2.1.3)

Therefore,

\[
e^{j2\beta} = a_1 e^{j\beta} + a_2, \]  
(5.2.1.4)

and

\[
e^{-j2\beta} = a_1 e^{-j\beta} + a_2, \]  
(5.2.1.5)

which yields

\[
a_1 = \frac{\sin(2\omega \Delta t)}{\sin(\omega \Delta t)} \quad \text{and} \quad a_2 = -1. \]  
(5.2.1.6)
When this AR model is used with the feedback control scheme of Fig. 7, the harmonic function is completely suppressed, as shown in Fig. 8. The same results are shown in Fig. 9 in the frequency domain. We use the same reference (20 μPa) to obtain the dB scale. The noise at frequencies other than 1000 Hz is due to the precision limitation of the computer. There is about a 130 dB difference between the peak at 1000 Hz and the minimum value for the uncontrolled signal, which roughly represents the precision available with the computer. For the signal with control, because all the content is of the same order and near zero, we obtain -250 dB. This same level would be obtained for the uncontrolled signal at frequencies other than 1000 Hz if there was no large value (90 dB) in the frequency domain. The same remark concerning the precision of the computer applies for results in Figs. 11 and 13.

In order to be more realistic, we now take into consideration that combustion noise has significant random content and add a random component whose amplitude is ten percent of the harmonic function amplitude. We use a random number generator to create a random signal such that the ratio between its amplitude to that of the input signal is ten percent. When we do this we find that the control system cannot drive the random component of the signal to zero, as shown in Figs. 10 and 11. This follows logically from the definition of a random signal; i.e., that it is unpredictable.

5.2.2 For a Sum of Harmonic Functions

Now consider the signal \( x(t) = \sum_{i=1}^{5} \cos(\omega_i t) \). The maximum order of the regression which must be considered is ten (five harmonic functions using two coefficients each). Because of the possible relationship among the frequencies (i.e., whether or not \( \omega_i / \omega_j \), \( i \neq j \), is rational [18]), the order is in fact less.
Figure 8. Control of One Harmonic Signal, Time Domain: signal $x(t) = \cos(1000t)$, regression order $n = 2$, discretization frequency is 20 kHz. The dark line is the signal with control. A small peak near the zero axis can be noticed for the first step due to the initial conditions, and then the result equals zero.
Figure 9. Control of One Harmonic Signal, Frequency Domain: signal $x(t) = \cos(1000t)$, regression order $n = 2$, discretization frequency is 20 kHz.
Figure 10. Control of One Harmonic Signal + Random Noise, Time Domain: signal $x(t) = \cos(1000t) + .1(\text{random})$, regression order $n = 2$, discretization frequency is 20 kHz.
Figure 11. Control of One Harmonic Signal + Random Noise, Frequency Domain: signal $x(t) = \cos(1000t) + .1(\text{random})$, regression order $n=2$, discretization frequency is 20 kHz.
The order is found by testing different models of different order in the control method, and then using the one which gives the best result. The program USEDFT (Listing 2 in Appendix C) gives the acoustic power for each frequency band of 1000 Hz width, so we choose the number of coefficients which gives the lowest total level of power. The program COEF (Listing 3 in Appendix C) gives the coefficients of the regression. As can be seen in Figs. 12 and 13, the same kind of result can be obtained for a sum of harmonic functions as for a unique harmonic function. There is a transition period during which the control signal has to track the harmonic signal, because of the initial conditions. Then, the signal with the control drops to zero. The same problem noticed earlier with one harmonic function concerning the precision of the computer and its influence on the result, for frequencies other than the peak frequencies, also exists here. An attenuation of about 90dB is obtained for each of the harmonic signals. New peaks show up in the controlled signal, and we do not really understand them. But even with these new peaks the effect of the control is important.

5.3 Application to Real Combustion Noise

A small open burner of 200 mm length with a 5 mm diameter is operated with oxygen and methane. The flow rates were not recorded because we only wanted a typical combustion noise spectrum. We use two files of real combustion noise obtained by recording typical combustion noise produced by this open burner, digitized at two different discretization frequencies, 25,600 Hz and 12,800 Hz. These frequencies are sufficiently high that we can get a good representation of the combustion noise, at least in the range 0-10,000 Hz for the low sampling frequency and 0-5,000 Hz for the high
Figure 12. Control of Five Harmonic Signals, Time Domain: regression order $n = 6$, discretization frequency is 20 kHz. The dark line is the signal with control. A transition period is noticed at the beginning followed by a drop of the signal with control to the zero axis.
Figure 13. Control of Five Harmonic Signals, Frequency Domain: regression order \( n = 6 \), discretization frequency is 20 kHz.
one. These frequency ranges are defined by the requirement of more than two samples to represent a harmonic function with discrete data. For example, with a frequency of 12,800 Hz, a discretization frequency higher than 25,600 Hz is needed to get more than two samples per period.

The coefficients of the autoregressive model are found using the method described in Chapter 4. It was found that the more coefficients that were used, the less power remained in the error signal, which represents the combustion noise with control. But for more than ten coefficients, the attenuation does not improve significantly. With twenty coefficients, the improvement relative to the results with ten coefficients is less than ten percent over the range 0-12,800 Hz. Therefore, only ten coefficients were used.

The results obtained in the frequency domain are shown in Figs. 14 and 15. An attenuation of more than 40 dB is obtained in the band of interest to combustion noise, 0 - 3,000 Hz. Note that there is a slight increase in noise level at higher frequencies however. The results are even better with the highest sampling rate used in Fig. 14. This is logical because more information about the combustion noise is obtained with a higher discretization frequency, and therefore the control system tracks the combustion noise more accurately. The increase in noise level at high frequencies seen in both figures may be due to the relatively low number of samples used to obtain the FFT of the signal in time domain. Only 2000 samples were available for both the combustion noise and the combustion noise with control. We can expect better results with a larger number of samples which will have the effect of reducing the influence of "computer noise" and the influence of real random noise in the original signal.
Figure 14. Control of Combustion Noise at 25600 Hz, Frequency Domain: regression order $n = 10$, discretization frequency is 25,600 Hz.
Figure 15. Control of Combustion Noise at 12800 Hz, Frequency Domain: regression order $n = 10$, discretization frequency is 12,800 Hz.
6.0 Complete Simulation of Control of Combustion Noise

The purpose of this chapter is to describe more precisely the complete system including the burner and the control device.

6.1 Description of a Complete Simulation

Up to this point, simulation of the combustion noise and its control was performed considering the principle of anti-sound cancelling the sound. In a practical application of the technique the situation is more complicated. A system consisting of a burner and a feedback control system is shown schematically in Fig. 16, and Fig. 17 represents the block-diagram of the control system, with the control notation used in Chapter 4. The block-diagram in Fig. 17 represents the way the simulation is performed by the computer. For every step in time, relative to the discretization frequency, the computer
Transfer function $T_0$
Amplifier (flame) + time delay due to the travel of the acoustic waves

Figure 16. Schematic of an Entire Representation of the Control System.
Figure 17. Block-Diagram of an Entire Control System: this diagram represents the way the simulation of the whole system burner-controller is performed by the computer.
calculates $\hat{X}_{k+1}$ using the matrix A, the previous vector $\hat{X}_k$, the matrix H and the error $\varepsilon_k$. Then the first component of the vector $\hat{X}_k$, which represents the output $\hat{y}_k$, is delayed (it takes the value of $\hat{y}$ obtained at step $k+1-p$) to take into account the delay due to the calculation of the algorithm. Then, using the inverse of the model of the flame transfer function, the signal input to the acoustic driver is found. For the simulation, we also use the model of the flame transfer function to simulate the real burner. Figure 18 is the block-diagram of the control system relative to the physical elements of a real burner. This means that Fig. 18 is slightly different from Fig 17, in order to more closely represent a real experiment, with the real flame transfer function taken into account. Figure 19 is the block-diagram of the control system and the real burner. In this case we also illustrate the air and fuel flows, and the burner with the real flame transfer function $T_b$ as blocks. In Fig. 19 we emphasize the fact that the interaction between the control signal and the combustion noise occurs upstream of the flame. Then the result is summed and amplified by the real flame transfer function $T_b$ before arriving at the photomultiplier. At this later location, we still can consider a summation of the combustion noise at the photomultiplier $y_s$ and of the control signal passed through the real flame transfer function $T_b\hat{y}_{s-p}$, which may be equal to $\hat{y}_{s-p}$, if the model T of the real flame transfer function is perfect. Therefore, Fig. 17 and Fig. 19 are equivalent for the physics of the control system, except that in Fig. 17 we use the model T instead of the real transfer function $T_b$ used in Fig. 19. The point A in Figs. 18 and 19 represents the same location.

In the initial simulation of the control, we were considering an acoustic driver creating the anti-sound signal in the near field of an open burner, but at an unknown position. This configuration raises some problems. First, due to the relative positions of the source and the anti-sound source, there is a non-negligible phase shift between the two signals at the sensor location, and this phase shift is frequency dependent.
Figure 18. Block-Diagram of an Entire Control System with Physical Elements: this diagram represents the way the controller would perform in a real situation, with the real flame transfer function.
Figure 19. Complete Block-Diagram of an Entire Control System: this diagram represents the way the controller would perform in a real situation, with the real flame transfer function. The air and fuel flows, and the burner are illustrated with blocks.
This means that the attenuation achieved will be location dependent. Second, this configuration requires the same power for the anti-sound source as that produced by the primary source, which means that a large external amplifier would be required. Moreover, this configuration is not possible in an aircraft engine, for instance.

In order to address these problems, we locate the acoustic driver inside the burner, as shown in Fig. 16. We thus profit from the research done by Mc Manus, et al. [19] on the excitation of an unstable flame created by a rearward-facing step in the mixture flow. Referring to Mc Manus’ analysis, a weak acoustic excitation interacts with the mixture flow, increasing the system performance (higher heat release for the same dimension) and decreasing the magnitude of the acoustic waves generated by this turbulent flame. In our case, we have a small circular slot at the edge of the burner, such that the low frequency disturbance created by the acoustic driver interacts with the flow as in the case of Mc Manus’ experiment. The phase of the disturbance at the acoustic driver and at the edge of the burner is approximately the same because of the low frequency and the relatively small dimension of the burner. This gives us the same location for the two sources of sound and anti-sound. The combustion noise disturbance and the control signal created by the acoustic driver interact inside the flame, and then a common radial wave representing the result of this interaction goes from the burner to the sensor. Referring to the results of section 7.1 concerning the amplification of the flame, an amplification of 20 dB in the range 0-500 Hz between the speaker and the sensor can be expected. Therefore, the size of the required speaker can be estimated. In the range 0-500 Hz, the overall acoustic power of the combustion noise is 0.66 W. Using the 20 dB amplification, we need a speaker able to produce only 7 mW in the 0-500 Hz band. This value represents an order of the power required for the acoustic driver.

For the choice of a sensor, we follow the lead of Hurle, et al. [21] and Dines [16] and we choose a photomultiplier with CH or C filters. Dines proved that the light emission
due to a turbulent flame is strongly correlated with the concomitant combustion noise. A coherence of about .98 was obtained in the 50-2000 Hz band. Therefore, the use of photomultiplier has two advantages:

- the time delay due to the travel of the acoustic wave from the flame to the sensor is neglected, because the signal now travels at the speed of light. We gain a speed factor of roughly one million between the types of signal.

- The use of a photomultiplier is more feasible in an engine aircraft, where fiber optics can be used to transmit the light from the combustion chamber to a remote location where the photomultiplier can operate. Moreover, the photomultiplier will be less sensitive to disturbances from other sound sources and vibrations which can exist in an engine.

Finally, in order to be more realistic in our simulation of the control of combustion noise, three points need to be considered:

1. the computer calculating the appropriate control signal,

2. the time delay due to the calculation of the computer,

3. the modeling of the transfer function between the control signal and the sensor, which includes the acoustic driver, the flame and the time delay to reach the sensor.

The first point has already been discussed in the previous chapters, but the two other points have to be taken into account to create the correct control signal. These two points are taken up in the remaining chapters.
7.0 Simulation with the Flame as a Noise Amplifier

In this chapter the flame transfer function is modeled and its implementation in the control loop is described.

7.1 The Flame Transfer Function

It has been shown [20] using an open burner operating on an air-propane mixture that the flame acts as an amplifier for an acoustic signal input. Figure 20 is sketch of this burner taken from reference 20. A hot wire signal taken one inch (2.54 cm) below the flame front is used as the input and a photomultiplier signal is used as a measure of the acoustic output. The typical results for the transfer function are shown in Fig. 21. This transfer function is defined as the complex ratio of the output signal (in complex notation) to the input signal (in complex notation). In our case, this transfer function corresponds to the complex ratio of the output signal from the photomultiplier to the control signal input in the acoustic driver. For future experiments, we will need to check
Figure 20. Sketch of the Experimental Burner [20].

Simulation with the Flame as a Noise Amplifier
Figure 21. Experimental Result [20] and Model of a Flame Transfer Function.
the transfer function of the flame for the open burner in the appropriate range of frequencies (0 - 3000 Hz), and then model the result numerically.

A simplification of the transfer function (dashed line in Fig. 21) has been used in the current simulation. A first-order gain has been assumed with a slope of 20 dB/decade, a "corner" frequency \( \omega_c \) of 150 Hz, as shown in Fig. 21, and an amplitude \( G_0 \) of 1.E-3. We consider the same reference 2.E-5 Pa for the experiments and the model. Because the transfer functions corresponds to the ratio of voltages representing the input and output signal, we do not have any unit for \( G \). In other words, this transfer function \( G \) can be written in complex notation

\[
G = \frac{G_0}{1 + j \frac{\omega}{\omega_c}} ,
\]  

with \( G_0 \) the amplitude of the gain, \( \omega \) the frequency and \( 1/\omega_c \) the time constant of a hypothetical amplifier. The differential equation whose solution leads to Eq. (7.1.1) may be written

\[
\frac{1}{G_0} \frac{dy}{dt} = -\omega_c y + \omega_c v ,
\]

with \( y \) representing the amplification of the signal \( v \).

The solution of Eq. (7.1.2) is well known and can be written

\[
y(t) = y(t_0)e^{-\omega_c(t-t_0)} + G_0\omega_c \int_{t_0}^{t} v(\tau)e^{-\omega_c(t-\tau)}d\tau .
\]

In discrete notation, with time step \( \Delta t \), we approximate \( v(t + t_i) \) as \( v(t + \Delta t) \) for \( 0 < t_i < \Delta t \). Therefore, Eq. (7.1.3) becomes
\[ y_{k+1} = e^{-\omega \Delta t} y_k + (G_0 \omega_e v_{k+1}) \int_0^{\Delta t} e^{-\omega \lambda} d\lambda \]  

\[ y_{k+1} = e^{-\omega \Delta t} y_k + G_0 (1 - e^{-\omega \Delta t}) v_{k+1}. \]  

Equation (7.1.5) is used in the simulation to account for the flame transfer function.

### 7.2 Implementation in the Simulation

The principle of the calculation performed by the computer, in order to create the correct control signal for the acoustic driver, is to estimate what should be the resulting signal component at the location of the photomultiplier due to the control signal, and to calculate the control signal needed for the acoustic driver to obtain this estimate. This calculation can be done using the inverse of Eq. (7.1.5).

A more complex transfer function between \( v \) and \( y \) would have to be analyzed in the z-domain to make sure that the inverse transfer function is stable. In other words, all the poles of the inverse would have to be inside the unit circle. In the present case, however, the inverse transfer function is simple and does not have any poles.

Finally, we use the method described in Chapter 5 to get the coefficients of the autoregressive model for the variable \( \hat{x} \) or, using the notation of Chapter 4, for \( \hat{y} = C \hat{x} = \hat{x} \), which represents the estimate of the combustion noise at the
photomultiplier. Therefore, we estimate the control signal \( \hat{v} \) needed to feed the acoustic driver with the inverse transfer function of the flame discussed before; that means

\[
\hat{v}_{k+1} = \frac{1}{G_0(1 - e^{-\omega_0 \Delta t})} (\hat{\gamma}_{k+1} - \hat{\gamma}_k e^{-\omega_0 \Delta t}).
\] (7.2.1)

The results for the simulation with real combustion noise are exactly the same as before in Section 5.3 since \( TT^{-1} = 1.0 \) (refer to Fig. 17). Now we can create the control signal we need to input to the acoustic driver. The results are the same because the calculation of the signal \( \hat{v} \) is independent of the calculation of the control signal \( \hat{\gamma} \) at the photomultiplier and because we use the model of the flame transfer function to calculate \( \hat{v} \) and also to simulate the flame transfer function. The main difference between the simulation and a real experiment is in the use of a model of the flame transfer function in the simulation, compared to the real transfer function in the experiment. The program SIMUL (Listing 5 in Appendix C) performs the simulation with the model of the flame transfer function.
8.0 The Time Delay

The purpose of this chapter is to estimate the time delay due to the calculation of the algorithm, to derive the new equations for the feedback matrix H, and to give some results for one and two step delays.

8.1 Estimation of the Time Delay due to the Calculation of the Algorithm

Experiments and applications are envisioned in which microprocessors should be used, because they are more specific and thus are potentially faster than the IBM 3084 (25 mips) or the IBM 3090 (45 mips) used for the simulation. Moreover, the use of machine language can reduce the delay introduced by the computer.

The computation time delay is necessarily related to the number of coefficients we use in the autoregressive model, such that the time delay increases with the number of
coefficients. Therefore, a compromise has to be made between a large number of coefficients for the autoregressive model, which increases the accuracy of the model, and a small number of coefficients, which reduces the time delay in the calculation of the control signal. By "step delay" we mean the time delay between two steps of calculation for the discrete signal. For example, considering a computer able to calculate the control signal using an AR model with ten coefficients in 39.06 $\mu$s, we obtain a time delay of 39.06 $\mu$s, which represents one step delay for a discretization frequency of 25,600 Hz. As stated earlier, a higher discretization frequency $F_d$ should give better results, especially at high frequency. But a higher $F_d$ means a smaller step delay (in $\mu$s). Therefore, if we double $F_d$ for example, the number of step delays doubles for the same time delay for the calculation. Then if, for example, one time delay of 39.06 $\mu$s corresponds to one step delay for a discretization frequency of 25,600 Hz, this same time delay corresponds to two step delays for a discretization frequency of 52,000 Hz. Therefore, a compromise must also be made between a high discretization frequency $F_d$, which increases the accuracy of the control, and a low value for $F_d$, which reduces the number of step delays relative to a fixed time of calculation. Moreover, in taking into account a time delay in the equations, we reduce the efficiency of the control, as shown in the next section.

By using the calculation which takes into account the model of the flame transfer function discussed in Chapter 7, some estimate of the order of the delay is found, corresponding to an IBM 3084. The results of this study are given in Table 1.

<table>
<thead>
<tr>
<th>Number of Coefficients</th>
<th>Time Delay ($\mu$s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>180</td>
</tr>
<tr>
<td>10</td>
<td>480</td>
</tr>
<tr>
<td>20</td>
<td>1650</td>
</tr>
</tbody>
</table>
These values are divided by two with the IBM 3090. The step delay is equal to 39.06 μs for a discretization frequency of 25,600 Hz. This means that the computer needs 39.06 μs to calculate the control signal, between the time when it receives the input of the error ε to the time when it outputs the control signal at the acoustic driver \( \hat{v} \). If we compare the values obtained with the IBM 3090 with the step delay, we obtain a delay of two steps with five coefficients. We write \( p \) the number of step delay.

### 8.2 Equations for the Feedback Matrix \( H \)

We follow the lead of the method described in reference [17] to derive the new equations. With the same notations as used in Chapter 5 and taking into account the step delay \( p \), Eqs. (5.1.4) and (5.1.5) become

\[
\hat{y}_k = C \hat{x}_k \quad \text{with} \quad C = (1,0,...,0) \tag{8.2.1}
\]

and

\[
\hat{x}_{k+1} = A \hat{x}_k + H(\hat{y}_k - \hat{y}_{k-p}) . \tag{8.2.2}
\]

In the z-domain, Eqs. (8.2.1) and (8.2.2) yield

\[
\hat{y}(z) = C \hat{x}(z) \tag{8.2.3}
\]

and
\[(zI - A)\hat{X}(z) = Hy(z) - z^{-p}HC\hat{X}(z) , \quad (8.2.4)\]

and Eq. (8.2.4) can be rewritten

\[ (z^{p+1}I - z^pA + HC)\hat{X}(z) = z^pHy(z) . \quad (8.2.5)\]

Now we have the transfer function between the signal representing the real combustion noise at the photomultiplier, \(y(z)\), and the model of the combustion noise, \(X(z)\). In reference 17 it is explained how to choose the column matrix \(H\) such that the poles of the transfer function are inside the unit circle, to satisfy the stability requirements and to optimize the speed of convergence. In our case, the equations are different, but the purpose is the same. We need to find \(H\) such that the zeros of the polynomial function \(\det(z^{p+1}I - z^pA + HC)\) are inside the unit circle, and as close as possible to the origin in order to speed up the convergence.

We do not have a general method to obtain \(H\), therefore, we use a “brute force” method to obtain acceptable results. With the previous method neglecting step delay, we always obtain the first three components of \(H\) inside \([-5, +5]\), and the other components are equal to zero. Therefore, when we include the step delay we assume the value of each component of the column matrix \(H\) to be more than \(-5.0\) and less than \(+5.0\). Then the zeros of the polynomial function are checked for the different possible values of \(H\). We finally choose the matrix \(H\) which gives us zeros inside the smallest circle of radius less than one. This method is time-consuming, and there is no assurance that we have obtained the optimum solution, because we make some assumptions and we impose only one condition (minimal radius). Moreover, experience indicates that the difficulty of finding values of the matrix \(H\) using this method increases with the number of step delays and the number of coefficients. Therefore, we will give the results for one
step delay, which represents 39.06 \( \mu s \) allocated for the processor to calculate the control signal, with one, two and three coefficients for the autoregressive model of the real combustion noise; and for two step delays, which represents 78.12 \( \mu s \) of calculation time, with only one coefficient. The general equation for the determinant is derived in Appendix B. The program CALCULH (Listing 7 in Appendix C) implements the brute force method.

### 8.3 Results for One Step Delay

This section presents the results of the simulation for one step delay with one, two and three coefficients for the AR model of the combustion noise. The block diagram shown in Fig. 17 illustrates the time delay for every case of this simulation.

#### 8.3.1 For One Coefficient

We choose one coefficient for the autoregressive model. Therefore, Eq.(8.2.5) becomes for one step delay

\[
(z^2 - az + h)\hat{x}(z) = zhy(z) .
\]  

In this case, the factor \( a \) is always on the order of unity, and so we can find the optimum value of \( h \) without using the brute force method because it involves solution of only a second-order polynomial equation,
Let $z_1$ and $z_2$ be the roots of Eq. (8.3.1.2). Then we know that $z_1 z_2 = h$ and $z_1 + z_2 = a$ and, considering the cases of complex or real roots, a simple analysis gives the optimum $h$: $h = a^2/4$. This value is used in the simulation, and the results are represented in Fig. 22 for the control of the real combustion noise. An average attenuation of 7.5 dB is obtained over the 0-1000 Hz band. Most of the attenuation is obtained in the 0-500 Hz band, as shown in Fig. 22.

### 8.3.2 For Two Coefficients

In this case, we have to consider the polynomial function

$$z^3 - a_1 z^2 + (h_1 - a_2)z + a_2 h_2 = 0.$$  \hspace{1cm} (8.3.2.1)

Because it is a third-order equation, the brute force method is used, and the smallest radius obtained which enclosed all the roots is 0.678. The result for the control of the real combustion noise is shown in Fig. 23. An average attenuation of 9.4 dB is obtained over the 0-1000 Hz band. An improvement of the attenuation can be noticed by comparison with the case with one coefficient described in Section 8.3.1. Moreover, most of the attenuation is now obtained in the 0-1000 Hz band, compared with the 0-500 Hz band in the case of one coefficient. The lower attenuation at the low frequencies (less than 500 Hz) obtained with two coefficients, relative to the results with one coefficient, can be explained only by the different number of coefficients. When we use two coefficients instead of one, the global attenuation improves, but it does not mean that the results improve for each frequency.
8.3.3 For Three Coefficients

Now we consider the equation

\[ z^4 - a_1 z^3 + (h_1 - a_2)z^2 + (a_2 h_2 + a_3 h_3 - a_4)z + a_4 h_2 = 0. \]  \hspace{1cm} (8.3.3.1)

The smallest radius obtained which enclosed all the roots is 0.801. The result for the control of the real combustion noise is shown in Fig. 24. An average attenuation of 16 dB is obtained over the 0-1000 Hz band, and we obtain an attenuation of 15.5 dB over the 0-1500 Hz band. An improvement of the attenuation can be noticed by comparison with the case with two coefficients described in Section 8.3.2. A slight increase in higher frequencies can be noticed. This increase can be due to the same problem encountered and discussed in Section 5.3.

The attenuation in the 0-1000 Hz band evidently improves with the order \( n \) of the AR model of the combustion noise, as illustrated by the evolution of the dark line in Figs. 22, 23, and 24.

8.4 Results for Two Step Delays

8.4.1 For One Coefficient

Referring to Appendix B, we consider the equation

\[ z^3 - a_1 z^2 + h_1 = 0. \]  \hspace{1cm} (8.4.1.1)
The smallest radius obtained which enclosed all the roots is 0.65. The result for the control of the real combustion noise is shown in Fig. 25. An average attenuation of 4 dB is obtained over the 0-1000 Hz band, with most of this attenuation being obtained in the 0-500 Hz band. Referring to Figs. 22 and 25, we notice a decrease of the average attenuation from 7.5 dB to 4 dB of the combustion noise with control when one coefficient for the AR model is used with one and two step delays, respectively.

To summarize, Figs. 22, 23, and 24 illustrate the increase of the attenuation with one, two and three coefficients for the AR model and for one step delay, and Figs. 22 and 25 illustrate the decrease of the attenuation with one and two step delay for one coefficient for the AR model.
Figure 22. Control of Combustion Noise with Time Delay ($n = 1, p = 1$): Frequency Domain. Regression order $n = 1$, one step delay $p = 1$, discretization frequency is 25,600 Hz.
Figure 23. Control of Combustion Noise with Time Delay (n = 2, p = 1): Frequency Domain. Regression order n = 2, one step delay p = 1, discretization frequency is 25,600 Hz.
Figure 24. Control of Combustion Noise with Time Delay (n = 3, p = 1): Frequency Domain. Regression order n = 3, one step delay p = 1, discretization frequency is 25,600 Hz.
Figure 25. Control of Combustion Noise with Time Delay (n = 1, p = 2): Frequency Domain. Regression order n = 1, two step delay p = 2, discretization frequency is 25,600 Hz.
9.0 Summary and Conclusion

Our simulation of the control of real combustion noise includes several elements:

1. an autoregressive model of the combustion noise,
2. a feedback loop based on the "observer" method, which is the core of the control,
3. a model of the transfer function of the flame,
4. a method to take into account the time delay required to calculate the control signal, and
5. a plan for practical implementation of the method.

In taking into account every step of the control system except the time delay, an attenuation of about 40 dB is obtained in the 0 - 3000 Hz band by simulation using real combustion noise. We also have shown than the real combustion noise we used contains determinant components at low frequencies, because we were able to attenuate them. This confirms the initial assumption that the combustion noise has some deterministic content.

Because the method to find the right coefficients for the feedback loop, in the case of the time delay, is time-consuming and does not ensure an optimum, a simulation was
for the autoregressive model, and for two step delays with one coefficient. One step delay is relative to the discretization frequency, and for the simulation with 25,600 Hz, the step delay is equal to 39.06 μs. For example, considering a computer able to calculate the control signal using an AR model with ten coefficients in 39.06 μs, we obtain a time delay of 39.06 μs, which represents one step delay for a discretization frequency of 25,600 Hz.

The results and remarks for the simulation taking into account the time delay due to the calculation of the algorithm are the followings:

1. the attenuation is about 11 dB in the 0 - 3000 Hz band with three coefficients (n = 3) and one step delay (p = 1), and better results can be expected with more coefficients for the autoregressive model,

2. in using high performance microprocessors designed for signal processing, there is potential for reducing the time delay. Therefore, the results corresponding to no time delay might be considered as the limiting case,

3. in practical applications, the transfer function between the acoustic driver and the photomultiplier, and the coefficients for the autoregressive model, must be obtained experimentally.

Finally, a method has been conceived to control the combustion noise generated by an open burner by creating an anti-noise input with an acoustic driver, and by exploiting the amplification of the flame itself.

The following suggestions are made for further research in this area:

1. experimental verification of this method should be undertaken,

2. a mathematical treatment may exist which gives a general method for finding the optimum value of the feedback coefficients, without using a brute force method,

3. the simulation will improve with the use of an experimental transfer function between the speaker and the photomultiplier,
4. Some general method should be derived starting from this point, which will be applicable to combustion noise in the case of an enclosed burner, in order to be closer to the engine aircraft configuration.

A final remark may be made concerning this work to suggest further research in this area. We tried to reduce the noise produced by a turbulent flame with an open burner, using an anti-sound source. But the design of the burner itself might be another means to obtain the same result. For example, Bitterlich [22] was able to do this in the case of a natural draft burner, as was Lamaucusa [23] in the case of a four cylinder engine. In other words, a systematic approach involving interactive design of the burner and its control may optimize the results.
References


Appendix A. The z-Transform

A.1 Definition

Let \( x(n) \) be a discrete sequence \( (n \geq 0) \). Then the z-transform of this sequence is

\[
X(z) = \sum_{n=0}^{\infty} x(n)z^{-n},
\]

(A1)

where \( z \) is a complex variable, \( z = u + iv \). This complex function is defined only on the convergence domain of the series. If the limit of \( \left| \frac{x(n+1)}{x(n)} \right| \) when \( n \) goes to infinity exists and is equal to a real \( L \), the convergence domain is outside a circle of radius \( L \) and center \( 0 \).

As an example consider the unit step function

\[
x(n) = 1 \quad n \geq 0.
\]

(A2)
We obtain

\[ X(z) = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}}, \quad (A3) \]

which converges for \(|z| > 1\).

### A.2 Inverse of the z-transform

If \(X(z)\) is the z-transform of the sequence \(x(m)\), we obtain

\[ x(m) = \frac{1}{2\pi i} \oint_X X(z)z^{-m-1} \, dz, \quad (A4) \]

where \(C\) is the border of a domain included in the convergence domain, with \(0\), the origin, inside. Equation (A4) can be derived from the following steps:

We have the identities

\[ \oint_C z^{-1} \, dz = 2\pi i \quad (\text{trivial for a circle}), \quad (A5) \]

and

Appendix A. The z-Transform
\[ \int_C z^m dz = 0 \quad \text{if} \quad m \neq -1. \quad (A6) \]

We can write

\[ \int_C X(z)z^{m-1} dz = \int_C \left[ \sum_{q=0}^\infty x(q)z^{m-q-1} \right] dz = \sum_{q=0}^\infty x(q) \int_C z^{m-q-1} dz, \quad (A7) \]

and using Eqs. (A5) and (A6), we obtain

\[ \int_C X(z)z^{m-1} dz = x(m) 2\pi i, \quad (A8) \]

which leads directly to Eq. (A4).

### A.3 Linearity

Let \( X(z) \) be the z-transform of the sequence \( x_1(m) \), and let \( X_2(z) \) be the z-transform of the sequence \( x_2(m) \). Then the z-transform of \( ax_1(m) + bx_2(m) \) is

\[ X(z) = \sum_{m=0}^\infty (ax_1(m) + bx_2(m))z^{-m} = a \left[ \sum_{m=0}^\infty x_1(m)z^{-m} \right] + b \left[ \sum_{m=0}^\infty x_2(m)z^{-m} \right] \quad (A9) \]
or

$$X(z) = aX_1(z) + bX_2(z), \quad (A10)$$

which proves the linearity of the $z$-transform.

### A.4 Delay

Consider the sequence $x(m)$ ($m \geq 0$; otherwise $x(m) = 0$) with its $z$-transform $X(z)$. Also consider the sequence $y(m)$ defined as $y(m) = x(m-k)$, where $k$ is the delay, with its $z$-transform $Y(z)$. Then we have

$$Y(z) = \sum_{m=0}^{\infty} y(m)z^{-m} = \sum_{m=0}^{\infty} x(m-k)z^{-m} = \sum_{q=0}^{\infty} x(q)z^{-q-k},$$

with $q = m-k$ and $x(k) = 0$ for $k$ less than 0. So we obtain

$$Y(z) = z^{-k} \sum_{q=0}^{\infty} x(q)z^{-q} = z^{-k}X(z). \quad (A11)$$
Appendix B. The Time-Delay Equation

The time-delay equation is given by \( \det(z^{p+1}I - z^p A + HC) = 0 \), as derived in Chapter 8. We develop the determinant for an order \( n \) of the autoregressive model and for \( p \) step delays. We call the determinant \( D_n^p \). We have

\[
D_n^p = \begin{bmatrix}
(z^{p+1} - a_1z^p + h_1) & (-a_2z^p) & \cdots & \cdots & ( -a_nz^p) \\
(h_2 - z^p) & z^{p+1} & 0 & \cdots & 0 \\
h_3 & ( -z^p) & z^{p+1} & \cdots & \cdots \\
\cdots & 0 & ( -z^p) & \cdots & \cdots \\
h_n & 0 & \cdots & \cdots & 0 \\
\end{bmatrix}
\]

\( (B1) \)

We develop this determinant starting with the last column, and we obtain

\[
D_n^p = z^{p+1}D_{n-1}^p + \cdots + (-1)^{n+1}( -a_nz^p)E_{n-1}^p ,
\]

\( (B2) \)
with

\[
E^p_{n-1} = \begin{bmatrix}
(h_2 - z^p) & z^{p+1} & 0 & \ldots & 0 \\
(h_3) & (-z^p) & z^{p+1} & \ldots & \ldots \\
h_4 & 0 & (-z^p) & \ldots & \ldots \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
h_n & 0 & \ldots & \ldots & (-z^p) \\
\end{bmatrix} \tag{B3}
\]

We can derive the exact expression of \(E^p_{n-1}\) by recursion, and the result is

\[
E^p_{n-1} = (h_2 - z^p)(-z^p)^{n-2} + (-1)^n \sum_{j=3}^{n} h_j z^{np+j-2} \tag{B4}
\]

Therefore, we have the recursion equation for \(D^p_n\)

\[
D^p_n = z^{p+1}D^p_{n-1} + (-1)^n a_n z^p E^p_{n-1} \tag{B5}
\]

which yields the general equation

\[
D^p_n = D^p_2 z^{(n-2)(p+1)} + \sum_{k=4}^{n+1} (-1)^k a_{k-1} E^p_{k-2} z^{(n+2-k)p + n + 1 - k} \tag{B6}
\]

with Eq. (B4), and with

\[
D^p_2 = z^p(z^{p+2} - a_1 z^{p+1} - a_2 z^p + h_1 z + h_2) \tag{B7}
\]

Appendix B. The Time-Delay Equation 78
Finally, Eqs. (B4), (B6), and (B7) give the result for the determinant in the case of $p$ step delays.
Appendix C. Program Listings

C.1 Listing 1: Subroutine DFT

This subroutine is used in the program USEDFT to perform the Discrete Fourier Transform of any function.

```fortran
C       ==================================================================
C       COMPUTATION OF THE DFT FOR DIFFERENT DATA.
C       WE OBTAIN THE AMPLITUDE OF THE DFT FOR EACH
C       FREQUENCY (ALSO FOR F = 0)
C       ==================================================================

C
SUBROUTINE DFT(N,VAR,DFTAB)
C
  REAL*4 PI,X1,X2,VAR(1550),DFTAB(800)
  INTEGER N
  PARAMETER (PI=3.14159)
C
  DO 160 I = 0,(N/2)
     X1 = 0
     X2 = 0
     DO 150 J = 0,N-1

Appendix C. Program Listings
80
```
\[ X_1 = X_1 + \text{VAR}(J + 1) \cdot \cos(J \cdot 1^2 \cdot \pi / \text{FLOAT}(N)) \]
\[ X_2 = X_2 + \text{VAR}(J + 1) \cdot \sin(J \cdot 1^2 \cdot \pi / \text{FLOAT}(N)) \]

150 \text{CONTINUE}
\text{DFTAB}(I + 1) = X_1^*X_1/\text{FLOAT}(N^*N) + X_2^*X_2/\text{FLOAT}(N^*N)

160 \text{CONTINUE}
\text{RETURN}
\text{END}
C.2 Listing 2: Program USEDFT

This program gives the frequency domain representation of any function. It also gives the energy content per band.

C  **********************************************************************
C  **********************************************************************
C  ** THIS PROGRAM USES THE DFT ROUTINE FOR                         
C  ** DIFFERENT FUNCTIONS                                          
C  WE CALCULATE THE SOUND POWER LEVEL                              
C  BY BANDWIDTH                                                      
C  **********************************************************************
C  **********************************************************************
C
PROGRAM USEDFT
REAL*4 T,PI,FD,Y,Z,X(1500),SXX(750)
REAL*4 FLIMIT,ENERGY(20),FLS,FLB,TOTAL
INTEGER NDATA,IND,NB
PI = ACOS(-1.)

WRITE(6,*)'NUMBER OF DATA IN FILE INDFT 
& FOR THE FUNCTION (< 1500) ?'
READ(5,*) NDATA
WRITE(6,*)'FREQUENCY OF DISCRETIZATION ?'
READ(5,*) FD
OPEN (UNIT = 2,FILE = 'INDFT')
DO 20 I = 1,NDATA
READ(2,*) X(I)
20 CONTINUE
CALL DFT(NDATA,X,SXX)

C
FLIMIT = FD/2.
NB = INT(FLIMIT/1000.)
DO 30 I = 1,NB + 1
ENERGY(I) = 0.
30 CONTINUE
DO 40 I = 0,(NDATA/2)
FIN0 = I/(FLOAT(NDATA)/FD)
DO 50 J = 1,NB + 1
FLS = J*1000.
FLB = (J-1)*1000.
IF ((FIN0.LE.FLS).AND.(FIN0.GE.FLB)) THEN
ENERGY(J) = ENERGY(J) + (2.*SXX(I+1))*FD/FLOAT(NDATA)
ENDIF
50 CONTINUE
40 CONTINUE
TOTAL = 0.
DO 55 J = 1,NB + 1
TOTAL = TOTAL + ENERGY(J)
55 CONTINUE

C
OPEN (UNIT = 1, FILE = 'OUTDFT')
WRITE(1,*), 'FOR THE DATA, THE AUTO SPECTRUM IS:
WRITE(1,*)
DO 60 J = 1,NB
WRITE(1,*) 'ENERGY BETWEEN ',(J-1)*1000,' AND ',J*1000,' HZ : ',
&ENERGY(J)
60 CONTINUE
WRITE(1,*) 'ENERGY BETWEEN ',NB*1000,' AND ',FLIMIT,' HZ : ',
&ENERGY(NB + 1)
WRITE(1,*) 'OVERALL ENERGY BETWEEN 0 AND ',FLIMIT,' HZ : ',TOTAL
WRITE(1,*)
WRITE(1,*), 'FREQUENCY 10*LOG GXX '
WRITE(1,*)
DO 100 I = 0,(NDATA/2)
WRITE(1,1000) I/(FLOAT(NDATA)/FD),10*LOG(2.*SXX(I+1))/4.E-10
&/LOG(10.)
100 CONTINUE
1000 FORMAT(1X,F12.5,2X,F12.5)
END
C.3 Listing 3: Program COEF

This program gives the coefficients for the autoregressive model of any function. The number of coefficients can be chosen.

```
C **************************************************************
C CALCUL DES COEFFICIENTS DE REGRESSION
C LINEAIRE, A PARTIR D'UN FICHIER DES
C VALEURES TEMPORELLES DE X ET Y
C DOUBLE PRECISION.
C THE TOLERANCE IS LOW
C **************************************************************

C PROGRAM COEF
C ---------------------------------------------------------
C WE DECIDE THE ORDER OF THE REGRESSION
C WITH NDX = ORDER + 1, NIND = ORDER < 50
C ---------------------------------------------------------
C
C ---------------------------------------------------------
C WE MUST HAVE NPOINTS = LDX + NIND, BECAUSE WE HAVE N POINTS
C ---------------------------------------------------------
C
REAL*8 XYMEAN(51)
REAL*8 X(1000,51), B(50,1), R(51,51)
REAL*8 COV(51,51), SCPE(1,1)
C ---------------------------------------------------------
C WE MUST HAVE LESS THAN 1000 POINTS
C ---------------------------------------------------------

REAL*8 VALX(1000)
INTEGER N,NIND,INCD(1,1)
C NUMBER OF POINTS N
WRITE(6,'(A)') 'NUMBER OF POINT FOR THE DATA (N <= 1000) ?'
READ(5,*) N
C
WRITE(6,'(A)') 'DESIRED ORDER OF THE REGRESSION?'
```
READ(5,*), NIND
NDEP = 1.D0
NDX = NIND + NDEP
LDCOV = NIND + NDEP
LDSCPE = NDEP
INTCEP = 1.D0
LDB = INTCEP + NIND
LDR = INTCEP + NIND
LDX = N + 1.D0 - NDX

C
CALL REGRES(NDEP,NDX,LDCOV,LDSCPE,INTCEP,LDB,LDR,LDX,
& N,NIND,XYMEAN,X,B,R,COV,SCPE,VALX)
END

C
C ......................................................
C ......................................................
C SUBROUTINE REGRES FOR THE CALCULATION OF THE COEFFICIENTS
C OF THE REGRESSION
C ......................................................
C ......................................................
C SUBROUTINE REGRES(NDEP,NDX,LDCOV,LDSCPE,INTCEP,LDB,LDR,LDX,
& N,NIND,XYMEAN,X,B,R,COV,SCPE,VALX)
C
INTEGER INCD(1,1)
REAL*8 XYMEAN(NIND + NDEP)
REAL*8 X(LDX,NDX), B(LDB,NDEP), R(LDR,LDR)
REAL*8 COV(LDCOV,NIND + NDEP), SCPE(LDSCPE,NDEP)
REAL*8 VALX(LDX + NDX - 1)
EXTERNAL DCORVC, DRCOV, AMACH
IFRQ = 0.D0
IWT = 0.D0
MOPT = 0.D0
ICOPT = 1.D0
NROW = LDX
NVAR = NDX

C
OPEN(UNIT = 1, FILE = 'INCOEF')
DO 10 I = 1, LDX + NDX - 1
READ(1,*) VALX(I)
10 CONTINUE
DO 20 I = 1, LDX
DO 20 J = 1, NDX
X(I,J) = VALX(i + J - 1)
20 CONTINUE
C
CALL DCORVC (0, NROW, NVAR, X, LDX, IFREQ, IWT, M0PT, ICOPT, XYMEAN,
&            COV, LDCOV, NCID, I, NOBS, NMISS, SUMWTF)
C
TOL = 1.D-1*AMACH(4)
CALL DRCOV (INTCENP, NIND, NDEP, COV, LDCOV, XYMEAN, SUMWTF, TOL,
&                B, LDB, R, LDR, IRANK, SCPE, LDSCEPE)
C
OPEN (UNIT = 2, FILE = 'OUTCOEF')
DO 50 I = 1, LDB
WRITE(2, *) B(LDB-I+1, I)
50 CONTINUE
RETURN
END
C.4 Listing 4: Program OBS2

This program performs the simulation of the control of any signal, without an amplifier or time delay included. This is the application of the observer method.

C ********************
C EXAMPLE OF THE OBSERVER METHOD
C FOR THE CONTROL OF A SIGNAL
C WITH AN AR MODEL
C WE CREATE 2 FILE FOR THE RESULTS
C SIGNAL AND DIFFERENCE
C DOUBLE PRECISION
C ********************
C
PROGRAM OBS2
REAL*8 DELTA(2000), SIGNAL(2000), A(70)
REAL*8 RATE, RANDOM(2000), MEAN, BIG, DEL, MOY, SXX, SXXS
REAL*8 FD, SUM, PI
REAL*8 H(70), PLAN(70, 70), L(70, 70), LN(70, 70), C(70, 70), CN(70, 70)
INTEGER N, N1, M, NUM
EXTERNAL RNSET, DRNUN, DMRRRR, DLINRG
PI = DACOS(-1.0D0)
BIG = 0.0D0
MOY = 0.0D0
C
C ........................
C INPUT OF THE DATA
C ........................
C
WRITE(6,*) 'THE FILE FOR THE SIGNAL MUST BE "SIGNAL"
WRITE(6,*) 'NUMBER OF POINT FOR THE SIGNAL DATA (N <= 2000)?'
READ(5,*) NUM
C
WRITE(6,*) 'FREQUENCY OF DISCRETIZATION FD IN HZ?'
READ(5,*) FD
WRITE(6,*) 'RATE OF NOISE ? (PUT .1 FOR 10%)'
READ(5,*) RATE
WRITE(6,*) 'ORDER OF THE REGRESSION ?'
READ(5,*) N
OPEN(UNIT = 2, FILE = 'OUTCOEF')
DO 40 I = 1, N
   READ(2,*) A(I)
40 CONTINUE
WRITE(6,*) 'AT WHICH STEP DO YOU WANT TO CONSIDER & THE BIGGEST DIFF. AND THE AVERAGE (LESS THAN 2000) ?'
READ(5,*) M

C
C *************************************************************************
C CALCULATION OF THE RANDOM VARIABLE
C WITH AN AVERAGE OF ZERO
C *************************************************************************
C
MEAN = 0.D0
CALL RNSET(5520314)
CALL DRNUN(NUM, RANDOM)
DO 50 I = 1, NUM
   MEAN = MEAN + RANDOM(I)
50 CONTINUE
MEAN = MEAN/DFLOAT(NUM)
DO 60 I = 1, NUM
   RANDOM(I) = RANDOM(I)-MEAN
60 CONTINUE

C WE USE SXX.
C *************************************************************************
C
SXXS = 0.D0
OPEN (UNIT = 1, FILE = 'SIGNAL')
DO 61 I = 1, NUM
   READ(1,*) SIGNAL(I)
   SXXS = SXXS + SIGNAL(I)*SIGNAL(I)
61 CONTINUE
SXXS = SXXS/DFLOAT(NUM)
SUM = DSQRT(2.D0*SXXS)
DO 70 I = 1, NUM
   RANDOM(I) = 2.D0*RATE*SUM*RANDOM(I)
70 CONTINUE
C
C  ------------------------------
C  CREATION OF THE SIGNAL + NOISE
C  ------------------------------
C
DO 80 I = 1, NUM
   SIGNAL(I) = SIGNAL(I) + RANDOM(I)
80  CONTINUE
C
C  ------------------------------
C  CALL TO THE CALCULATION OF THE ESTIMATOR
C  ------------------------------
C
   CALL PARAM (N, A, L, L, N, C, CN, PLAN, H)
   CALL TIME (N, NUM, PLAN, H, SIGNAL, DELTA)
C
C  ------------------------------
C  END OF THE PROGRAM, RESULTS
C  ------------------------------
C
DO 180 I = M, NUM
C  WE CALCULATE THE RATE OF DELTA TO THE AMPLITUDE OF THE SIGNAL
C  (ROUGHLY = SUM)
   DEL = DABS(DELTA(I))/SUM
   MOY = MOY + DEL
   IF (BIG.LT.DEL) BIG = DEL
180  CONTINUE
   MOY = MOY/DFLOAT(NUM-M+1)
C
   OPEN (UNIT = 3, FILE = 'INDFT')
   DO 200 I = 101, NUM
   WRITE(3,1000) DELTA(I)
200  CONTINUE

   OPEN (UNIT = 4, FILE = 'OBSRES2')
   WRITE(4,*)'OBSERVER METHOD FOR SIGNAL + NOISE'
   WRITE(4,*)'RATE OF NOISE :: RATE'
   WRITE(4,*)'ORDER OF THE REGRESSION :: N'
   WRITE(4,*)'AVERAGE BEGINING AT STEP :: M'
   WRITE(4,*)'AMPLITUDE :: SUM'
   WRITE(4,500) BIG, MOY
   WRITE(4,*)
   WRITE(4,*)'TIME DELTA'
   WRITE(4,*)
DO 201 I = 1, NUM
    WRITE(4,1000) (I-1)/FD, DELTA(I)
201    CONTINUE
500 FORMAT(1X,'BIGGEST DIFFERENCE : ','F10.5,' AVERAGE : ','F10.5)
1000 FORMAT(1X,2(F10.5))
END

C
C ........................................................................
C ........................................................................
C
C SUBROUTINE OF THE CALCULATION OF THE PARAMETERS
C PARAM
C ........................................................................
C ........................................................................
C
C ************
C WE CREATE THE MATRIX PLAN AND WE CALCULATE
C THE OBSERVER MATRIX H
C ************
C
C SUBROUTINE PARAM (N,A,L,LN,C,CN,PLAN,H)
C
C REAL*8 A(70)
C REAL*8 H(N), PLAN(N,N), L(N,N), LN(N,N), C(N,N), CN(N,N)
C INTEGER N
C
C
C ************
C INITIALIZATION OF THE MATRIX A AND L
C ************
C
C DO 1000 I = 1, N
C PLAN(I,I) = A(I)
C L(I,I) = A(I)
1000 CONTINUE
C DO 1010 I = 2, N
C DO 1010 J = 1, N
C K = J + 1
C IF (I.EQ.K) THEN
C PLAN(I,J) = 1.D0
C L(I,J) = 1.D0
C ELSE
C PLAN(I,J) = 0.D0
C 1010 CONTINUE
L(I,J) = 0.D0
ENDIF
1010 CONTINUE
C
C ........................................................................
C CALCULATION OF THE OBSERVABILITY MATRIX C
C AND THE CURRENT MATRIX L = L*A TO OBTAIN A**N
C ........................................................................
C
C(I,I) = 1.D0
DO 1020 J = 2,N
C(I,J) = 0.D0
1020 CONTINUE
DO 1030 K = 1,N-1
DO 1040 J = 1,N
C(K+1,J) = L(I,J)
1040 CONTINUE
CALL DMRRRR (N,N,L,N,N,N,PLAN,N,N,N,LN,N)
DO 1050 J1 = 1,N
DO 1050 J1 = 1,N
L(I1,J1) = LN(I1,J1)
1050 CONTINUE
1030 CONTINUE
C
C ........................................................................
C NOW L = A**N
C ........................................................................
C ........................................................................
C CALCULATION OF THE INVERSE OF C
C THE INVERSE IS PUT IN C
C ........................................................................
C
CALL DLINRG (N,C,N,C,N)
C ........................................................................
C CALCULATION OF THE OBSERVER VECTOR H
C H = LAST COLUMN OF (A**N)*(C**-1)
C HERE WE OBTAIN LAST COLUMN OF L*C
C ........................................................................
C
CALL DMRRRR (N,N,L,N,N,N,C,N,N,N,CN,N)
DO 1060 I = 1, N
H(I) = CN(I, N)
1060 CONTINUE
RETURN
END

C

SUBROUTINE OF THE CALCULATION OF THE ESTIMATOR
TIME

SUBROUTINE TIME (N, NUM, PLAN, H, SIGNAL, DELTA)
REAL*8 H(N), PLAN(N, N)
INTEGER N, NUM

C

INITIALIZATION OF THE ESTIMATOR

DO 2000 I = 1, N
ESTX(I) = 1.0D0
NEWX(I) = 1.0D0
DELTA(I) = SIGNAL(I) - ESTX(I)
2000 CONTINUE

C

CALCULATION OF THE ESTIMATOR
FOR EACH STEP

DO 2010 I = N, NUM - 1
DO 2020 J = 1, N
NEWX(I + 1 - (J - 1)) = H(J) * DELTA(I)
DO 2020 K = 1, N
NEWX(I + 1 - (J - 1)) = NEWX(I + 1 - (J - 1)) + PLAN(J, K) * ESTX(I - (K - 1))
2020 CONTINUE

DO 2030 J = 1, N
ESTX(I + 1 - (J - 1)) = NEWX(I + 1 - (J - 1))
2030 CONTINUE
   DELTA(I+1)=SIGNAL(I+1)-ESTX(I+1)
2040 CONTINUE
C
   RETURN
END
C.5 Listing 5: Program SIMUL

This program performs the simulation of the control of any signal, with a first order system transfer function included.

C ****************************
C EXAMPLE OF THE OBSERVER METHOD
C FOR THE CONTROL OF A SIGNAL
C WITH AN AR MODEL AND AN
C AMPLIFIER.
C WE CREATE 2 FILE FOR THE RESULTS
C DOUBLE PRECISION
C ****************************
C
PROGRAM SIMUL
REAL*8 DELTA(1000),SIGNAL(1000),ESTX(1000),ESTY(1000)
REAL*8 A(70)
REAL*8 RATE,RANDOM(1000),MEAN,BIG,DEL,MOY,SXX,SXXS
REAL*8 FD,FC,GAIN,SUM,PI
REAL*8 H(70),PLAN(70,70),L(70,70),LN(70,70),C(70,70),CN(70,70)
INTEGER N,M,NUM
EXTERNAL RNSET,DRNUN,DMRRRR,DLINRG
PI = DACOS(-1.D0)
BIG = 0.D0
MOY = 0.D0
C
C ------------------------
C INPUT OF THE DATA
C ------------------------
C
WRITE(6,*)'THE FILE FOR THE SIGNAL MUST BE "SIGNAL"
WRITE(6,*)'NUMBER OF POINT FOR THE SIGNAL DATA (N <= 1000) ?'
READ(5,*) NUM
C
WRITE(6,*)'FREQUENCY OF DISCRETIZATION FD IN HZ ?'
READ(5,*) FD
WRITE(6,*) 'RATE OF NOISE? (PUT .1 FOR 10%)'
READ(5,*) RATE
WRITE(6,*) 'ORDER OF THE RECURSION?'
READ(5,*) N
OPEN(UNIT = 2, FILE = 'OUTCOEF')
DO 40 I = 1, N
READ(2,*) A(I)
40 CONTINUE
WRITE(6,*) 'CHARACTERISTIC FREQUENCY FC FOR THE AMPLIFIER (IN HZ)?'
READ(5,*) FC
WRITE(6,*) 'GAIN FOR THE AMPLIFIER?'
READ(5,*) GAIN
WRITE(6,*) 'AT WHICH STEP DO YOU WANT TO CONSIDER
& THE BIGGEST DIFF. AND THE AVERAGE (LESS THAN 1000)?'
READ(5,*) M
C
C **********************************************/
C CALCULATION OF THE RANDOM VARIABLE
C WITH AN AVERAGE OF ZERO
C **********************************************/
C
MEAN = 0.D0
CALL RNSET(5520314)
CALL DRNUN(NUM, RANDOM)
DO 50 I = 1, NUM
MEAN = MEAN + RANDOM(I)
50 CONTINUE
MEAN = MEAN/DFLOAT(NUM)
DO 60 I = 1, NUM
RANDOM(I) = RANDOM(I) - MEAN
60 CONTINUE
C **********************
C WE USE SXX.
C **********************
C
SXXS = 0.D0
OPEN (UNIT = 1, FILE = 'SIGNAL')
DO 61 I = 1, NUM
READ(1,*) SIGNAL(I)
SXXS = SXXS + SIGNAL(I)*SIGNAL(I)
61 CONTINUE
SXXS = SXXS/DFLOAT(NUM)
\[ SUM = DSQRT(2.0*SXXS) \]
DO 70 I = 1,NUM
RANDOM(I) = 2.0*RATE*SUM*RANDOM(I)
70   CONTINUE
C
C ........................................
C CREATION OF THE SIGNAL + NOISE
C ........................................
C
DO 80 I = 1,NUM
SIGNAL(I) = SIGNAL(I) + RANDOM(I)
80   CONTINUE
C
C ........................................
C CALL TO THE CALCULATION OF THE ESTIMATOR
C ........................................
C
CALL PARAM(FD,FC,N,AL,LN,C,CN,H,PLAN)
CALL TIME(FC,GAIN,FD,NUM,N,H,PLAN,SIGNAL,DELTA,ESTX,ESTY)
C
C ........................................
C END OF THE PROGRAM, RESULTS
C ........................................
C
DO 90 I = M,NUM
C WE CALCULATE THE RATE OF DELTA TO THE AMPLITUDE OF THE SIGNAL
C (ROUGHLY = SUM)
DEL = DABS(DELTA(I))/SUM
MOY = MOY + DEL
IF (BIG.LT.DEL) BIG = DEL
90   CONTINUE
MOY = MOY/DFLOAT(NUM-M+1)
C
OPEN (UNIT = 3, FILE = 'INDFT')
DO 100 I = 101,NUM
WRITE(3,600) DELTA(I)
100  CONTINUE
OPEN (UNIT = 4, FILE = 'SIMURES2')
WRITE(4,'(A)') 'OBSERVER METHOD FOR SIGNAL + NOISE'
WRITE(4,'(A)') 'RATE OF NOISE ::RATE
WRITE(4,'(A)') 'ORDER OF THE REGRESSION ::N
WRITE(4,'(A)') 'AVERAGE BEGINING AT STEP ::M

Appendix C. Program Listings
WRITE(4,*) AMPLITUDE : 'SUM
WRITE(4,500) BIG,MHOY
WRITE(4,*)
WRITE(4,*') DELTA'
WRITE(4,*)
DO 110 I = 1,NUM
WRITE(4,600) DELTA(I)
110 CONTINUE
500 FORMAT(1X,'BIGGEST DIFFERENCE : ',F10.5,' AVERAGE : ',F10.5)
600 FORMAT(1X,F10.5)
END

C
C ..............................................
C ..............................................
C
C SUBROUTINE OF THE CALCULATION OF THE OBSERVER PARAMETERS
C "PARAM"
C ..............................................
C ..............................................
C
SUBROUTINE PARAM (FD,FC,N,A,L,LN,C,CN,H,PLAN)
C
REAL*8 FD,FC,WC,PI,A(70)
REAL*8 H(N),PLAN(N,N),L(N,N)
REAL*8 LN(N,N),C(N,N),CN(N,N)
INTEGER N
PI = DACOS(-1.D0)
WC = 2.D0*PI*FC
C
C ..............................................
C WE CREATE THE MATRIX PLAN AND WE CALCULATE
C THE OBSERVER MATRIX H
C ..............................................
C ..............................................
C INITIALIZATION OF THE MATRIX A AND L
C ..............................................
C
DO 1000 I = 1,N
PLAN(I,I) = A(I)
L(I,I) = A(I)
1000 CONTINUE
DO 1010 I = 2, N
DO 1010 J = 1, N
K = J + 1
IF (I.EQ.K) THEN
   PLAN(I,J) = 1.D0
   L(I,J) = 1.D0
ELSE
   PLAN(I,J) = 0.D0
   L(I,J) = 0.D0
ENDIF
1010 CONTINUE
C
C........................................................................
C CALCULATION OF THE OBSERVABILITY MATRIX C
C AND THE CURRENT MATRIX L = L*A TO OBTAIN A**N
C........................................................................
C
C(I,I) = 1.D0
DO 1020 J = 2, N
C(I,J) = 0.D0
1020 CONTINUE
C
C(I,J) = L(I,K)*PLN(K,J)
DO 1030 K = 1, N-1
DO 1040 J = 1, N
C(K+1,J) = C(K,J)
1040 CONTINUE
C
CALL DMRRR (N,N,L,N,N,N,PLAN,N,N,N,L,N,N)
DO 1050 I1 = 1, N
DO 1050 J1 = 1, N
L(I1,J1) = LN(I1,J1)
1050 CONTINUE
1030 CONTINUE
C
C---------------------------------
C NOW L = A**N
C---------------------------------
C
C........................................................................
C CALCULATION OF THE INVERSE OF C
C THE INVERSE IS PUT IN C
C........................................................................
C
CALL DLINRG (N,C,N,C,N)
C
C -----------------------------
C CALCULATION OF THE OBSERVER VECTOR H
C H = LAST COLUMN OF (A**N)*(C**-1)
C HERE WE OBTAIN LAST COLUMN OF L*C
C -----------------------------
C
C CALL DMRRRR (N,N,L,N,N,N,C,N,N,N,CN,N)
DO 1060 I = 1,N
H(I) = CN(I,N)
1060 CONTINUE
C
RETURN
END
C
C -----------------------------
C -----------------------------
C
C SUBROUTINE OF THE SIMULATION IN TIME DOMAIN
C "TIME"
C -----------------------------
C -----------------------------
C
C SUBROUTINE TIME (FC,GAIN,FD,NUM,N,H,PLAN,SIGNAL,DELTA,
&ESTX,ESTY)
C
REAL*8 FC,GAIN,FD,WC,PI,ACU1,ACU2,USETIME
REAL*8 H(N),PLAN(N,N)
REAL*8 SIGNAL(1000),NEWY(1000),DELTA(1000),ESTX(1000),ESTY(1000)
INTEGER NUM,N
C
PI = DACOS(-1.D0)
WC = 2.D0*PI*FC
C
C -----------------------------
C INITIALIZATION OF THE ESTIMATOR
C -----------------------------
C
DO 2000 I = 1,N
ESTY(I) = 1.D0
NEWY(I) = 1.D0
DELTA(I) = SIGNAL(I)-ESTY(I)
2000 CONTINUE
C
C WE INITIALIZE ESTX(I) WITH 1
C
ESTX(I) = 1.D0
DO 2010 I = 1,N-1
ESTX(I + 1) = (ESTY(I + 1) - DEXP(-WC/FD)*ESTY(I))/
&(GAIN*(1.D0 - DEXP(-WC/FD)))
2010 CONTINUE
C
C ---------------------------------------------
C CALCULATION OF THE ESTIMATOR
C FOR EACH STEP
C ---------------------------------------------
C
DO 2020 I = N,NUM-1
DO 2030 J = 1,N
NEWY(I + 1-J-1) = H(J)*DELTA(I)
DO 2030 K = 1,N
NEWY(I + 1-J-1) = NEWY(I + 1-J-1) + PLAN(J,K)*ESTY(I-K-1)
2030 CONTINUE
DO 2040 J = 1,N
ESTY(I + 1-J-1) = NEWY(I + 1-J-1)
2040 CONTINUE
C
C ---------------------------------------------
C CALCULATION OF THE SIGNAL X TO
C INPUT IN THE SPEAKER AND WHICH
C WILL BE AMPLIFIED. WE USE A MODEL
C OF THIS AMPLIFIER.
C ---------------------------------------------
C
ESTX(I + 1) = (ESTY(I + 1) - DEXP(-WC/FD)*ESTY(I))/
&(GAIN*(1.D0 - DEXP(-WC/FD)))
DELTA(I + 1) = SIGNAL(I + 1)-ESTY(I + 1)
2020 CONTINUE
C
RETURN
END
**C.6 Listing 6 : Program TOTAL**

This program performs the simulation of the control of any signal, with a time delay included. The value of the feedback coefficients are input; that is, they are evaluated by a "brute force" method. This is an example for one step delay and three coefficients.

```c
C    ******************************
C    EXAMPLE OF THE OBSERVER METHOD
C    FOR THE CONTROL OF A SIGNAL
C    WITH AN AR MODEL AND A TIME DELAY
C    WE CREATE 2 FILE FOR THE RESULTS
C    SIGNAL AND DIFFERENCE
C    DOUBLE PRECISION
C    ******************************
C
PROGRAM TOTAL
REAL*8 DELTA(2000),SIGNAL(2000),A(50)
REAL*8 RATE,RANDOM(2000),MEAN,BIG,DEL,MOY,SXX,SXXX
REAL*8 FD,F(50),AMP(50),SUM,PI
REAL*8 H(50),PLAN(50,50),L(50,50),LN(50,50),C(50,50),CN(50,50)
INTEGER N,N1,M,NUM
EXTERNAL RNSET,DRNUN,DMRRRR,DLINRG
PI = DACOS(-1.0D0)
BIG = 0.0D0
MOY = 0.0D0

C
C    -----------------------------
C    INPUT OF THE DATA
C    -----------------------------
C
WRITE(6,*)'THE FILE FOR THE SIGNAL MUST BE 'SIGNAL''
WRITE(6,*)'NUMBER OF POINT FOR THE SIGNAL DATA (N <= 2000) ?'
READ(5,*) NUM

C
WRITE(6,*)'FREQUENCY OF DISCRETIZATION FD IN HZ ?'
```
READ(5,*) FD
WRITE(6,*)'RATE OF NOISE ? (PUT .1 FOR 10%)'
READ(5,*) RATE
WRITE(6,*)'ORDER OF THE REGRESSION ?'
READ(5,*) N
OPEN(UNIT = 2,FILE = 'OUTCOEF')
DO 40 I = 1,N
READ(2,*) A(I)
40 CONTINUE
WRITE(6,*)'AT WHICH STEP DO YOU WANT TO CONSIDER & THE BIGGEST DIFF. AND THE AVERAGE (LESS THAN 2000) ?'
READ(5,*) M
C
C ***********************************************
C CALCULATION OF THE RANDOM VARIABLE
C WITH AN AVERAGE OF ZERO
C ***********************************************
C
MEAN = 0.D0
CALL RNSET(5520314)
CALL DRNU(NUM,RANDOM)
DO 50 I = 1,NUM
MEAN = MEAN + RANDOM(I)
50 CONTINUE
MEAN = MEAN/DFLOAT(NUM)
DO 60 I = 1,NUM
RANDOM(I) = RANDOM(I)-MEAN
60 CONTINUE
C ***********************************************
C WE USE SXX.
C ***********************************************
C
SXXS = 0.D0
OPEN (UNIT = 1,FILE = 'SIGNAL')
DO 61 I = 1,NUM
READ(1,*) SIGNAL(I)
SXXS = SXXS + SIGNAL(I)*SIGNAL(I)
61 CONTINUE
SXXS = SXXS/DFLOAT(NUM)
SUM = DSQRT(2.D0*SXXS)
DO 70 I = 1,NUM
RANDOM(I) = 2.D0*RATE*SUM*RANDOM(I)
70 CONTINUE
C
C -----------------------------------
C CREATION OF THE SIGNAL + NOISE
C -----------------------------------
C
DO 80 I = 1, NUM
     SIGNAL(I) = SIGNAL(I) + RANDOM(I)
80 CONTINUE
C
C -----------------------------------
C CALL TO THE CALCULATION OF THE ESTIMATOR
C -----------------------------------
C
CALL CALCUL(NUM, N, SIGNAL, A, PLAN, DELTA)
C
C -----------------------------------
C END OF THE PROGRAM, RESULTS
C -----------------------------------
C
DO 180 I = M, NUM
C WE CALCULATE THE RATE OF DELTA TO THE AMPLITUDE OF THE SIGNAL
C (ROUGHLY = SUM)
     DEL = DABS(DELTA(I))/SUM
     MOY = MOY + DEL
     IF (BIG.LT.DEL) BIG = DEL
180 CONTINUE
     MOY = MOY/DFLOAT(NUM-M+1)
C
OPEN(UNIT = 3, FILE = 'INDFT')
DO 200 I = 101, NUM
     WRITE(3,1000) DELTA(I)
200 CONTINUE
C
OPEN (UNIT = 4, FILE = 'OBSRES2')
WRITE(4,*) 'OBSERVER METHOD FOR SIGNAL + NOISE'
WRITE(4,*) 'RATE OF NOISE : RATE'
WRITE(4,*) 'ORDER OF THE REGRESSION : N'
WRITE(4,*) 'AVERAGE BEGINING AT STEP : M'
WRITE(4,*) 'AMPLITUDE : SUM'
WRITE(4,500) BIG, MOY
WRITE(4,*)
WRITE(4,*) 'TIME    DELTA'
WRITE(4,*)
DO 201 I = 1, NUM
WRITE(4,1000) (1-I)/FD, DELTA(I)
201 CONTINUE
500 FORMAT (1X,'BIGGEST DIFFERENCE : ', F10.5, ' AVERAGE : ', F10.5)
1000 FORMAT (1X,2(F10.5))
END

C

C ..............................................................
C ..............................................................
C
C SUBROUTINE OF THE CALCULATION OF THE ESTIMATOR AND SO ON...
C
C ..............................................................
C ..............................................................
C
C ******************
C WE CREATE THE MATRIX PLAN AND WE INPUT
C THE OBSERVER MATRIX H
C ******************
C
C SUBROUTINE CALCUL (NUM,N,SIGNAL,A,PLAN,DELTA)
C
REAL*8 H(50),A(50),PLAN(N,N)
INTEGER N, NUM
C
C ............................
C INITIALIZATION OF THE MATRIX A AND L
C ............................
C
DO 90 I = 1, N
PLAN(I,1) = A(I)
90 CONTINUE
DO 100 I = 2, N
DO 100 J = 1, N
K = J + 1
IF (I.EQ.K) THEN
PLAN(I,J) = 1.0D0
ELSE
PLAN(I,J) = 0.0D0
100 CONTINUE
ENDIF

100 CONTINUE
C
C INPUT OF THE OBSERVER VECTOR H
C
C
H(1) = 0.53385D0
H(2) = 0.454336D0
H(3) = 0.373555D0
C
C
C INITIALIZATION OF THE ESTIMATOR
C
C
DO 150 I = 1, N
ESTX(I) = 1.D0
NEWX(I) = 1.D0
Y(I) = 1.D0
DELTA(I) = SIGNAL(I) - Y(I)
150 CONTINUE
C
C
C CALCULATION OF THE ESTIMATOR
C FOR EACH STEP
C 1 STEP DELAY FOR DELTA
C
C
DO 160 I = N, NUM - 1
DO 170 J = 1, N
NEWX(I + 1 - (J - 1)) = H(J) * DELTA(I)
DO 170 K = 1, N
NEWX(I + 1 - (J - 1)) = NEWX(I + 1 - (J - 1)) + PLAN(J, K) * ESTX(I - (K - 1))
170 CONTINUE
DO 171 J = 1, N
ESTX(I + 1 - (J - 1)) = NEWX(I + 1 - (J - 1))
Y(I + 1) = NEWX(I + 1)
171 CONTINUE
C
C
C DELTA TAKES INTO ACCOUNT THE 1 STEP DELAY
C
C
DELTA(I+1) = SIGNAL(I+1) - Y(I)
160 CONTINUE
C
RETURN
END
C.7 Listing 7: Program CALCULH

This program gives the coefficients of the column matrix H and the radius of the roots of the input polynomial function. We need to choose a domain for the value of the coefficients.

C
C **********************************************************
C WE LOOK FOR THE VALUE OF THE
C PARAMETERS H(I) FOR GIVEN COEFFICIENTS
C A(I) AND A GIVEN STEP DELAY P, SUCH
C THAT ALL THE ZEROS OF THE DETERMINANT
C ARE INSIDE THE CIRCLE OF RADIUS R (< 1)
C **********************************************************
C

PROGRAM CALCULH
REAL*8 ERRABS,ERRREL,RADIUS,ACU1,ACU2,USETIME
REAL*8 A(3),H(3),CZ(4),H1,H2,H3
COMPLEX*16 F,Z(4),ZINIT(4),Z1,Z2,Z3,Z4
INTEGER INFO(3),ITMAX,NGUESS,NKNOWN,NEW,INDEX,N
INTEGER IH1,IH2,IH3
COMMON/ VAR/H1,H2,H3,A
C
EXTERNAL F,ZANLY,CPUTIME
C
DATA ZINIT/4*(-1.D0,1.D0)/
Z1 = (0.D0,0.D0)
Z2 = (0.D0,0.D0)
Z3 = (0.D0,0.D0)
Z4 = (0.D0,0.D0)
C
ERRABS = 0.001D0
ERRREL = 0.001D0
NKNOWN = 0
NEW = 4
NGUESS = 4
ITMAX = 100

C
C WRITE(6,*)'NUMBER OF COEFFICIENTS'
C READ(3,*) N
N = 3
OPEN(UNIT = 1, FILE = 'OUTCOEF')
DO 5 I = 1, 3
READ(I,*) A(I)
5 CONTINUE
C
RADIUS = 1.0
H(1) = 0.0
H(2) = 0.0
H(3) = 0.0
OPEN(UNIT = 2, FILE = 'OUTH1')
DO 10 IH1 = -15, 0
DO 10 IH2 = -15, 15
DO 10 IH3 = -15, 15
C CALL CPUPTIME(ACU1, I1)
H1 = 1.0*D*DFLOAT(IH1)
H2 = 1.0*D*DFLOAT(IH2)
H3 = 1.0*D*DFLOAT(IH3)
CALL DZANLY (F, ERRABS, ERRREL, KNOWN, NNEW, NGUESS, ZINIT, & ITMAX, Z, INFO)
DO 20 I = 1, 4
CZ(I) = CDABS(Z(I))
20 CONTINUE
INDEX = 0
DO 30 I = 1, 4
IF(CZ(I).LT.RADIUS) INDEX = INDEX + 1
30 CONTINUE
IF(INDEX.EQ.4) THEN
H(1) = H1
H(2) = H2
H(3) = H3
RADIUS = MAX(CZ(1), CZ(2), CZ(3), CZ(4))
Z1 = Z(1)
Z2 = Z(2)
Z3 = Z(3)
Z4 = Z(4)
ENDIF
CALL CPU TIME(ACU2,I2)
IF(I1.NE.8.AND.I2.EQ.0) USE TIME = ACU2-ACU1
WRITE(6,*)'USED TIME FOR ONE LOOP IS = ',USE TIME
GOTO 110
CONTINUE
WRITE(2,*)'SOLUTION'
WRITE(2,*)'H1 = ',H1,' H2 = ',H2
WRITE(2,*)'H3 = ',H3
WRITE(2,*)'RADIUS = ',RADIUS
WRITE(2,*)'Z1 = ',Z1
WRITE(2,*)'Z2 = ',Z2
WRITE(2,*)'Z3 = ',Z3
WRITE(2,*)'Z4 = ',Z4
END
COMPLEX FUNCTION F(Z)
COMPLEX*16 Z
REAL*8 H1,H2,H3,A(3)
COMMON/ VAR/H1,H2,H3,A
F = Z**3-A(1)*Z**2+(H1-A(2))*Z+A(2)*H2
F = Z**4-A(1)*Z**3+(H1-A(2))*Z**2+(H2*A(2)+H3*A(3)-A(3))*Z+
& A(3)*H2
RETURN
END
Vita

The author was born on April 5, 1966, in Creil, France. He did his preparatory classes at the Lycée Saint Louis, Paris, France; then he passed the entrance examination for the Ecole Centrale de Paris, France. He received his engineering degree and his Diplôme d’Études Approfondies (DEA) in Thermophysics and Combustion from Ecole Centrale de Paris in July, 1989. He continued his graduate studies at Virginia Polytechnic Institute and State University in August, 1989, and completed his Master of Science in Mechanical Engineering in December of 1990.

Jean-Yves Capelle