CPRC CCRP

CANADIAN POLICE RESEARCH CENTRE

CENTRE CANADIEN DE RECHERCHES POLICIÈRES

TR-08-93
IMS Signal Processing

Dr. R.A. Goubran and Dr. H.M. Hafez
Systems and Computer Engineering
Carleton University

TECHNICAL REPORT

October 1993

NOTE: Further information about this report can be obtained by calling the CPRC information number (613) 998-6343

IMS SIGNAL PROCESSING

FINAL REPORT

December 1992

Submitted to: Dr. André Lawrence
Institute for Aerospace Research
National Research Council Canada
Montreal Road, Building M-50
Ottawa, Ontario, Canada, K1A 0R6

By: Dr. R.A. Goubran and Dr. H.M. Hafez
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada, K1S 5B6

Under Contract Number 92-551 / 4037

Résumé

Dans ce rapport, on traite de la détection de la cyclotriméthylène-trinitramine (RDX) par spectrométrie de mobilité ionique (SMI). On évalue un certain nombre de techniques de traitement numérique des signaux post-ionisation, en vue d'améliorer la sélectivité et les seuils de détection de la SMI.

L'évaluation fait appel à des données de SMI obtenues expérimentalement, ainsi qu'à des pics SMI simulés. Les données expérimentales, qui ont été fournies par le Conseil national de recherches du Canada, représentent des échantillons de cuir, des échantillons de RDX, et du cuir combiné à diverses quantités de RDX. Les pics simulés ont été obtenus à l'aide de programmes élaborés dans le cadre de contrats (988-4783 R et 990-1266/1045). Ces programmes génèrent des pics semblables à ceux obtenus avec un SMI, auxquels on peut attribuer une valeur quelconque pour l'amplitude, l'écart-type, la séparation inter-pics et le rapport signal/bruit (rapport S/B). Ainsi, on peut déterminer très précisément, en termes de sensibilité et de sélectivité, les limitations de tout algorithme de détection des pics.

Les résultats indiquent que ce sont les méthodes de calcul différentiel qui offrent la meilleure sélectivité, avec des séparations inter-pics minimum décelables de 0,20 msec. Toutefois, leur sensibilité est médiocre en raison de leur susceptibilité au bruit. Ce sont les méthodes de corrélations croisées qui offrent la meilleure sensibilité, c.-à-d. le meilleur seuil de détection, les quantités minimales décelables de RDX étant de 0,01 nanogramme. Malgré leur faible seuil de détection, les méthodes de corrélation possèdent une sélectivité médiocre en raison de leur effet de lissage. Un algorithme multirésolutions, combinant les méthodes de calcul différentiel et les méthodes de corrélation et possédant les avantages de ces deux méthodes, a été mis au point à contrat. Dans cet algorithme, décrit dans ce rapport, on applique les méthodes de corrélations croisées au signal original, afin de déterminer les endroits où des pics risquent d'être observés. On applique ensuite des méthodes de calcul différentiel à l'intérieur de fenêtres temporelles spécifiques, en procédant à un lissage approprié aux limites.

Dans le rapport, on examine aussi dans quelle mesure les réseaux neuronaux peuvent être appliqués au problème de détection des pics en spectrométrie de mobilité ionique. On donne un bref aperçu des réseaux neuronaux et on examine de façon plus détaillée le réseau de Hopfield qui a été sélectionné comme celui dont la structure convient le mieux au problème de détection des pics SMI. Les résultats des simulations montrent que les réseaux neuronaux permettent d'obtenir une résolution dans le temps supérieure aux limites des techniques classiques. Les réseaux de Hopfield permettent de distinguer des pics dont la séparation est égale au produit écart-type du pic x 0,65, alors que la séparation obtenue avec les techniques classiques est égale à l'écart-type x 1. D'autres résultats permettent d'analyser la performance des réseaux neuronaux dans des cas où le rapport signal/bruit est faible.

ABSTRACT

This report addresses the detection of Cyclotrimethylenetrinitramine (RDX) using Ion Mobility Spectrometry (IMS). It evaluates a number of post ionization Digital Signal Processing (DSP) techniques for improving the selectivity and detection limit in IMS.

The evaluation is based on experimental IMS data as well as simulated IMS peaks. The experimental data, provided by the National Research Council of Canada, represents leather samples, RDX samples, and leather combined with various amounts of RDX. The simulated peaks were generated using the programs developed under contracts (988-4783 R), and (990-1266/1045). These programs generate IMS-like peaks with any desired amplitude, standard deviation, peak separation, and signal to noise ratio (SNR). This way, the limitations of any peak detection algorithm in terms of sensitivity and selectivity can be determined very precisely.

The results indicate that derivative methods provide the best selectivity, with minimum detectable peak separations of 0.20 msec. However, their susceptibility to noise results in poor sensitivity. Cross-correlation methods provide the best sensitivity, or detection limit, with minimum detectable RDX quantities of 0.01 nanogram. Despite their good detection limit, correlation methods suffer from a poor selectivity due to their smoothing effect. A multiresolution algorithm combining derivative methods with correlation methods, and providing the advantages of both, was developed under this contract and is described in this report. In the proposed algorithm cross correlation methods are applied to the original signal in order to determine potential peak locations. Derivative methods are then applied within specific time windows with proper smoothing at the edges.

The report also discusses the applicability of neural networks to the peak detection problem in IMS. A brief overview of neural networks is presented with an emphasis of the Hopfield network which was selected as the most appropriate structure for the IMS peak detection problem. Simulation results demonstrate that neural networks are capable of achieving a time resolution beyond the limits of conventional techniques. Hopfield networks are shown to resolve peaks separated by 0.65 times the peak's standard deviation, as opposed to one standard deviation using conventional techniques. Other results also analyze the performance of neural networks in low signal to noise cases.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

Cyclotrimethylenetrinitramine, or RDX, is a component of plastic explosives and its detection is important in a number of security screening scenarios. Ion Mobility Spectrometry (IMS) has recently emerged as a technology which allows the rapid detection of sub-nanogram quantities of RDX and other plastic explosive residues.

In IMS, ion diffusion and mobility are studied by measuring the time it takes an ion to drift a specified length in an applied electric field. Gaseous molecules are converted into product ions by collisions with reactant ions. Product ions are then accelerated by an electric field against the countercurrent of a drift gas. The individual ion mobility depends on ionic mass, shape, and charge distribution. Complete ion drift patterns are obtained by measuring the time of flight of the ions as they drift through the electric field at atmospheric pressure [2]. Because of its detection limit, fast response, and operation at atmospheric pressure, IMS has been successfully miniaturized into a compact detection system suitable for field use[3].

Although IMS exhibits a very high and distinctive response toward some nitroaromatics (e.g. TNT), nitroesters (e.g. EGDN), and nitramines (e.g. RDX), it has drawbacks namely, moderate resolution and low chemical signal-to-noise ratio. Consequently, DSP techniques are necessary in order to improve the confidence levels in the output of IMS-based explosive detectors.

Chapter 2 of this report describes the experimental set-up and experimental data files used throughout the work. A literature survey of the relevant background material is also presented, including a description of the IMS signal, an analysis of the peak shapes and noise, and a brief review of peak detection algorithms commonly used in IMS-based detection.

Chapters 3, 4, and 5 deal with the derivative, cross-correlation, and neural network methods, respectively. Each method is described and its limitations regarding the detection of RDX traces using IMS are established. Finally, the conclusions resulting from this work are summarized in chapter 6.

2.0 THE IMS SIGNAL

2.1 Introduction

When a gaseous mixture is injected into an Ion Mobility Spectrometer, the resulting output signal comprises peaks corresponding to the various compounds in the mixture, and noise. The main goal of any detection algorithm is to identify the various peaks in the IMS output signal with a high level of confidence. Any algorithm is rated by its detection limit, or sensitivity to noise, and by its selectivity, or ability to resolve closely located peaks. Both, the detection limit and selectivity in IMS can be improved in the ionization stage using analytical chemistry techniques or at the post ionization stage using Digital Signal Processing (DSP) techniques [1,6,7]. This report deals with signal improvement in IMS using post ionization DSP techniques.

Sections 2.2 and 2.3 describe the experimental set-up and the experimental IMS data files used throughout this work. Section 2.4 addresses the IMS signal corresponding to RDX. The peak shapes and the noise are analyzed. Finally, the various peak detection algorithms applicable to IMS signals are summarized in section 2.5.

2.2 Experimental Set-Up

The experimental IMS data used in this work were obtained using a Phemto-Chem 100 ion mobility spectrometer (PCP Inc. West Palm Beach, FL). The experimental parameters used to operate the instrument are shown in table 1. At each scan of the IMS an analog output signal, containing a number of noisy peaks embedded in noise, is generated. A Low-Pass filter connected at the output stage of the IMS rejects the out-of-band noise. Furthermore, In order to minimize the noise at this stage, several scans are performed, and the IMS output signal is averaged over these scans. The number of scans in most of the tests presented in this work was set at 2048. The averaging process is based on the fact that given that the IMS actual signal is deterministic, averaging a number of them will maintain the signal energy, whereas given that the noise is random, averaging it will tend to reduce it. The averaged signal is therefore a smoothed version of the original IMS output signal at each scan. The averaging is performed using a Nicolet 1170 signal averager.

The averaged IMS signal is then digitized using a linear Analog to Digital (A/D)

converter with 12 bits of precision. The digital data were then transferred to an IBM compatible 486 Personal Computer and to a SUN workstation (Sun Microsystems Inc., Mountain View, CA) for processing and analysis using several Fortran 77 and C programs, as well as the MatLab software package (Mathworks Inc. Sherborn, MA).

Cell Length	14 cm
Drift Length	10 cm
Carrier Gas (+) (Nitrogen Spiked with Dichloromethane)	200 mL/min
Drift Gas (Nitrogen)	600 mL/min
Inlet and Drift Temperature	200 °C
Drift Voltage	- 2700 Volts
Dwell Time	2 μs/Channel
Gate Width	0.2 msec
Delay Time (*)	9.1 msec
Pressure	Atmospheric

(+) RDX peak corresponds to RDX (Cl^-) of m/z 256
(*) Time between gate opening and start of data collection

Table 1. IMS (Phemto-Chem 100) instrument parameters

2.3 Experimental IMS Data Files

The results discussed in this work are based on the analysis of IMS output signals corresponding to various amounts of RDX and interfering compounds due to the presence of leather using the procedure described in [2]. A number of IMS data files were provided by the Trace Vapour Detection Section of the National Research Council of Canada. The first set of files, shown in figure 2.1 , represents RDX injections with quantities ranging from 0.01 up to 0.2 nanogram. The horizontal axis represents the drift time in msec, while the vertical axis represents the amplitude of the averaged IMS output signal, the higher the quantity of injected RDX the higher the peak value and the higher the signal to noise ratio (SNR).

A second set of files representing experimental IMS signals corresponding to leather and RDX was also provided by the National Research Council. Figure 2.2 shows pure leather and pure RDX, and Figure 2.3 shows leather with various amounts of RDX impurity.

Fig 2.1 RDX Samples With Various Injected Quantities

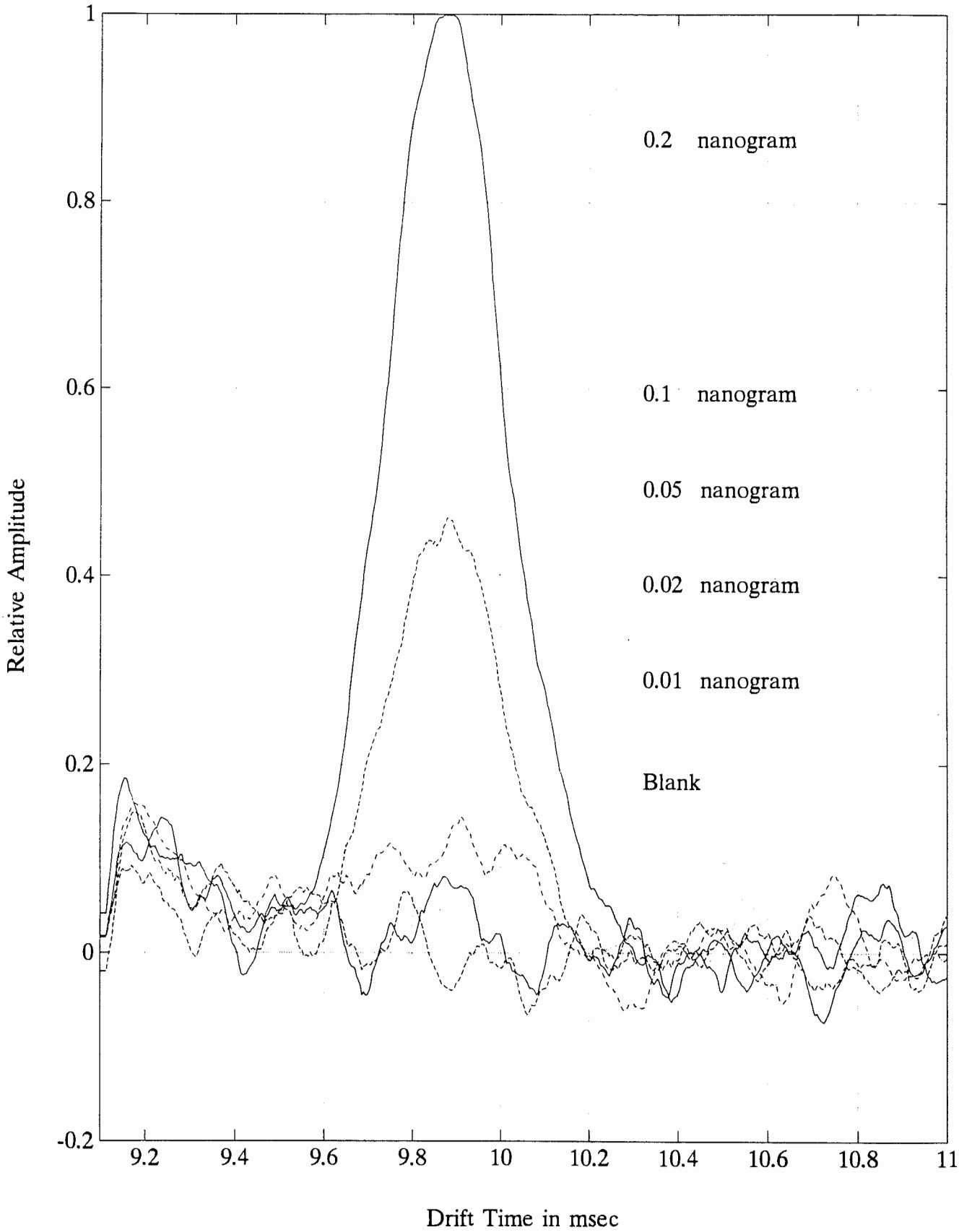


Fig 2.2 IMS Signal: Pure Leather and Pure RDX

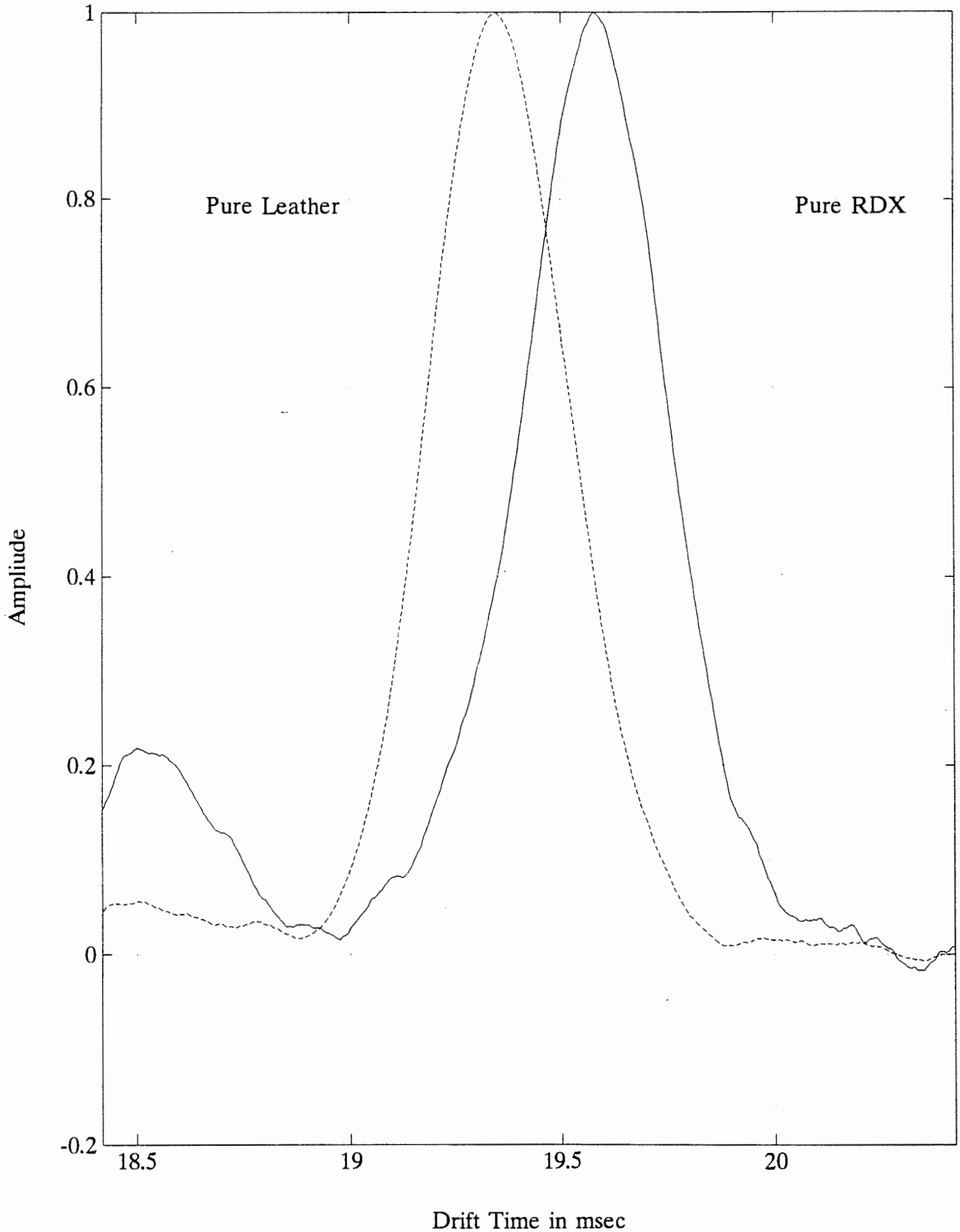
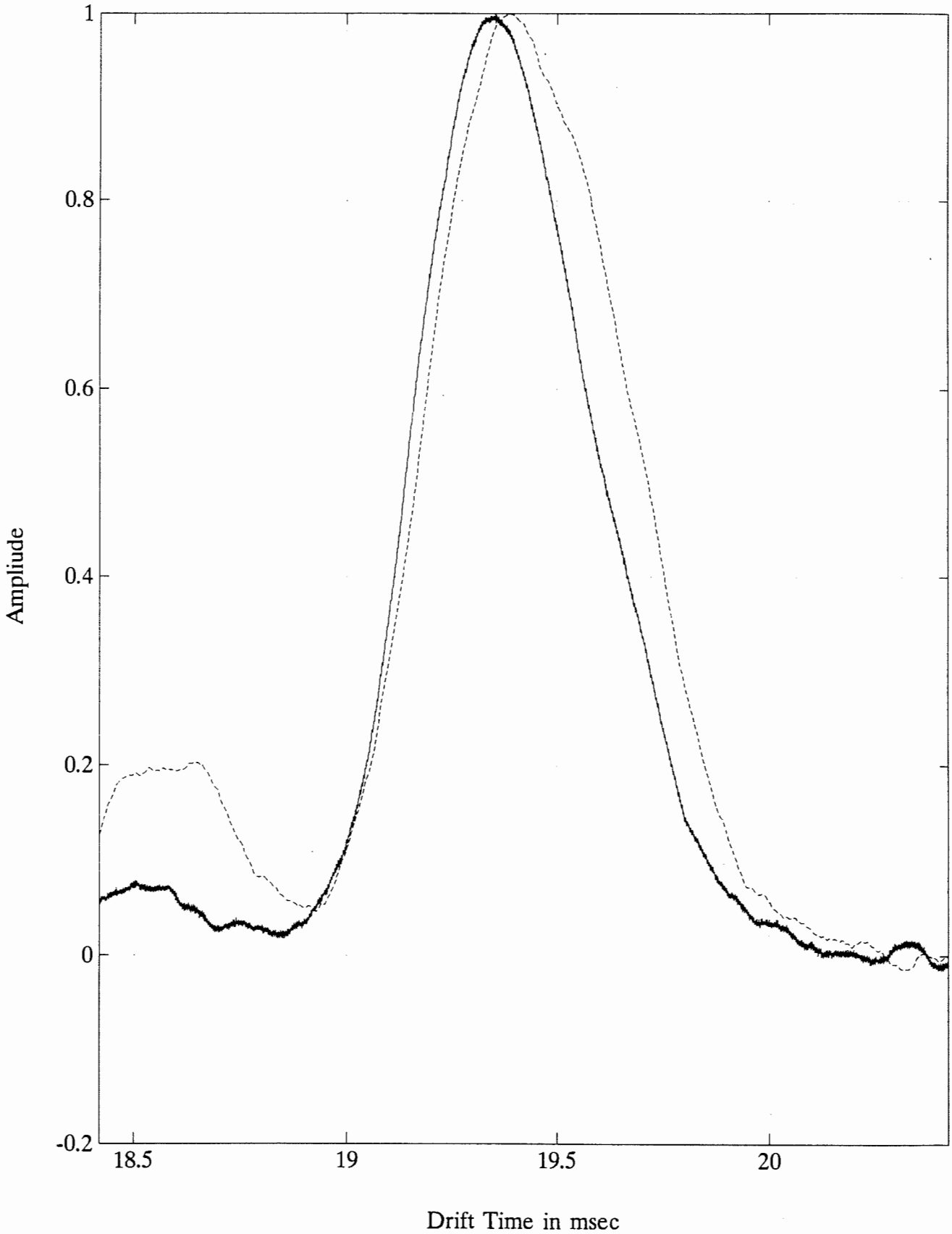


Fig 2.3 IMS Signal: Dominant Leather with RDX Impurity



2.4 IMS Signal Analysis, Modelling, and Simulation

Two main approaches are generally used in order to characterize and model IMS signals. The first one attempts to model the ion's behaviour in the IMS process and thus determine the IMS pulse shapes [4]. The resulting model is usually very complicated, and the introduction of simplifications leads to many inaccuracies. A different approach is to analyze experimental data obtained from actual IMS tests. Statistical information related to the peak shapes and noise is then extracted from the data and used to build the model. This approach, which was adopted in this work, leads to a more realistic simulation of IMS signals, however, its success is related to the amount of data being analyzed [8].

The analysis of the peaks in IMS signals [8] revealed that the peak shapes for the data being analyzed could be approximated by a Gaussian shape. In order to establish a quantitative measure to the finding, a program that generates a Gaussian peak, according to equation (1), was developed. For each experimental data file, the parameters of the Gaussian curve were optimized in order to achieve the best fit by minimizing the mean square of the error signal, defined as the difference between the experimental data and the ideal Gaussian curve.

$$y(t) = Y + \frac{A}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \qquad (1)$$

Where:

* σ is the standard deviation in *msec*
* A is the scaling factor or area under curve
* μ is the horizontal deviation or mean in *msec*
* Y is the vertical deviation
* t is the horizontal time scale in *msec*
* $y(t)$ represents the IMS peak signal

For the RDX IMS output files, with the mass of chemicals injected in these experiments (0.01 -0.2 nanogram), the standard deviation (σ) and the horizontal deviation (or mean μ) were found to be constant at 0.18 msec and 9.8 msec, respectively. The goodness of fit factor (τ) was define as:

$$\tau = \frac{\textit{Ideal Gaussian Curve Energy}}{\textit{Error Signal Energy}} \qquad (2)$$

The signal energy is calculated in a window of +/- 3 σ around the mean, thus ensuring that 99.74 % of the area under the curve is considered. The error signal is the difference between the ideal Gaussian curve and the experimental data curve; its energy is calculated using the same time window. Under these conditions, the Signal to Noise Ratio (SNR) for large quantities (ca. 20 nanogram) of RDX was found to be approximately 2500 (or 34 dB), meaning that the goodness of the Gaussian fit was 0.04 %.

It is important to emphasize that the error signal energy is a combination of errors due to the noise from the IMS process, thermal noise from the analog processing, quantization noise, as well as modelling errors due to the peaks shape not being perfectly Gaussian and fitting errors due to the fitting algorithm. In order to evaluate the Gaussian approximation for the peak shape, it is necessary to consider the modelling error only. This is achieved by minimizing as much as possible all the other sources of error. The IMS process error and the thermal noise are minimized by considering the curves with high chemical concentrations, whereas the fitting error is minimized by reducing the step size of the fitting algorithm.

Autocorrelation tests of the noise in IMS signals result in no clear repetitive patterns which indicates that the noise is random. Windowing and Fast Fourier Transform (FFT) tests indicate that the noise is bandlimited due to the low pass filtering performed at the output stage of the IMS. Finally, amplitude distribution tests of the noise signal indicate that it is Gaussian. It is therefore concluded that the noise present on IMS signals is random, bandlimited, and Gaussian.

Based on the previous findings an IMS signal simulator was developed. The simulator is capable of artificially generating IMS-like signals with any desired peak parameters, peak separation, and SNR. The signals are then used to test the detection limit and the selectivity of peak detection algorithms.

2.5 Summary of IMS Peak detection algorithms

The aim of IMS is to determine the various compounds existing in a given mixture. This is achieved by detecting the various peaks existing in the IMS output signal using a peak detection algorithm. The main problems facing any of these algorithms are the presence of noise and overlapping peaks.

Small quantities of a component result in a low energy peak whose amplitude is close to the amplitude of the noise (i.e low SNR). The minimum peak level that could be detected by a given algorithm determines the detection limit of such algorithm. The detection threshold of an algorithm could be lowered by varying its parameters. However this could result in false detection where the algorithm interprets a noise spike as an actual peak. The detection limit of a given algorithm could therefore be increased at the expense of false detection (or false alarm). The selectivity of a given algorithm is determined by the minimum peak separation that could be correctly detected by the algorithm. The resolution is a function of the peak separation, the standard deviation of each peak, the relative amplitude of the peaks, as well as the signal-to-noise ratio. These parameters were evaluated in reference[1] with respect to the second derivative algorithm. The detection limit and the selectivity of any detection algorithm should therefore be optimized based on the expected IMS peaks in the application, the signal-to-noise ratio, as well as the desired detection level, selectivity, and probability of false detection.

Several peak detection algorithms have been proposed [6,7,8,11] for resolving peaks in IMS and other similar situations. The most widely used algorithms, the derivative methods, and the cross correlation methods, are described and evaluated in detail in chapters 3 and 4 of this report. Another promising algorithm, using the Hopfield neural network, is evaluated in chapter 5. This section presents other possible peak detection algorithms that have been reported in the literature or have been tested in previous work by the authors. Curve fitting methods and adaptive signal processing methods are described in subsections 2.5.1 and 2.5.2 respectively.

2.5.1 Curve Fitting Methods

In curve fitting methods [6,11] the signal is resolved into distinct bands. For each band, a best fitting polynomial of order (n) is calculated using the least squares method. The polynomial coefficients are then used to estimate the parameters of the peak in this specific band.

In [6] it is assumed that the peaks have a Gaussian shape, which could approximated by a quadratic equation. A linear least-squares fit to a quadratic equation is therefore implemented for each band. The coefficients of the quadratic best fit are then used to calculate the parameters of the Gaussian curve, namely the standard deviation, the mean, and the area. Other models for the signal, such as Lorentzian peak shapes, have been analyzed in [11].

In curve fitting methods, the order of the fitting polynomial is crucial. A low order polynomial will tend to smooth the original data, and hence remove most of the noise by on the other hand remove the slight variations in the signal due to overlapping peaks. As a result, the detection limit will be enhanced and the expense of selectivity. A high order polynomial increases the computational complexity of the best-fitting algorithm and allows more noise into the system. Curve fitting algorithms are generally very accurate as far as estimating the peak position.

2.5.2 Adaptive Signal Processing Methods

Most Digital Signal Processing (DSP) applications require a filtering operation in some way. Adaptive filtering is the central module in many applications; some examples include: acoustic echo cancelling in audio teleconferencing, electronic echo cancelling and channel equalization in telephony, noise cancelling using a reference signal, adaptive coding of speech signals (ADPCM, speech companding, and scrambling), adaptive control systems, and image processing. Moreover, fixed filtering could be considered as a special case of adaptive filtering. Therefore, we will consider the case of a Finite Impulse Response (FIR) transversal filter using the stochastic gradient algorithm for adaptation [14, 15].

Figure 2.4 shows a block diagram of the filter. The incoming sample is entered into a tap delay line. The filter output is a weighted sum of the delayed samples. Therefore,

$$y(t) = \sum_{i=0}^{N} c_i \cdot x(t-i)$$

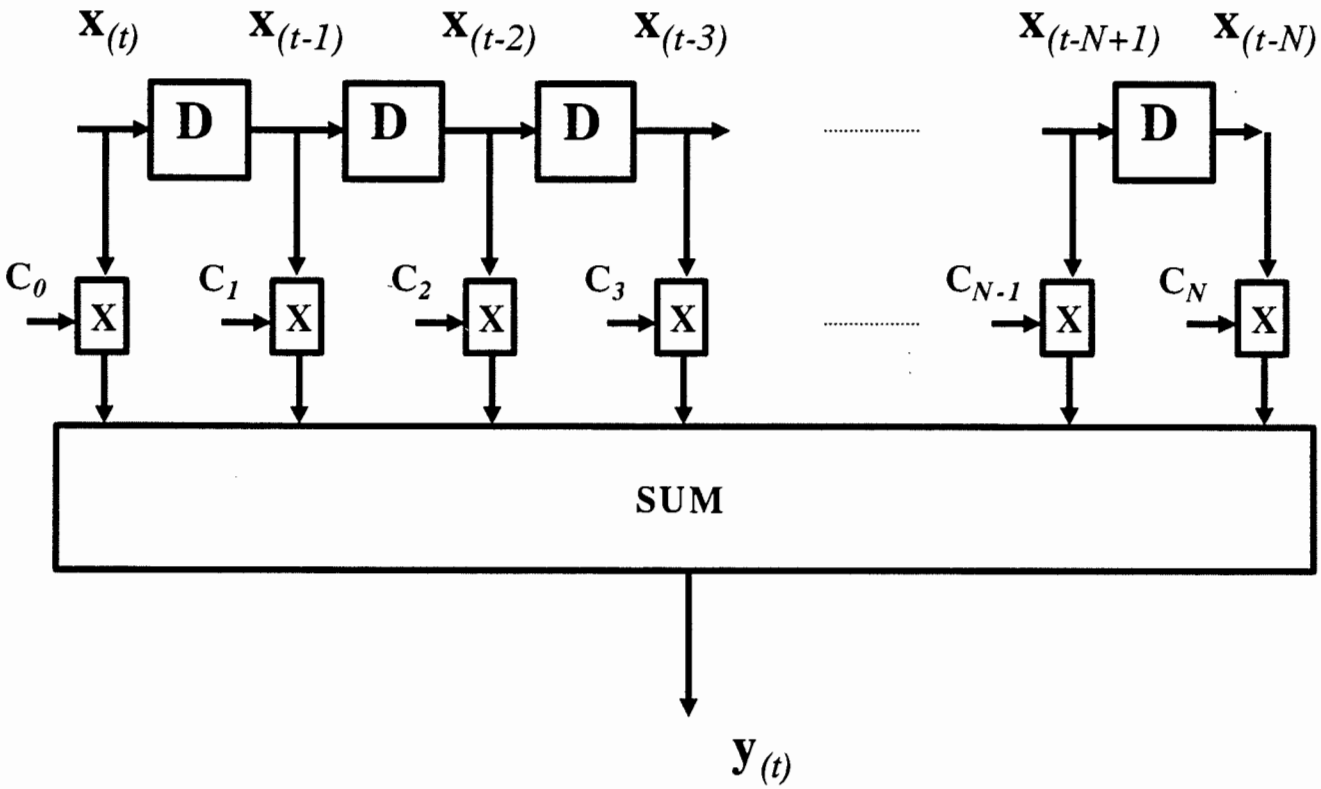


Figure 2-4 Finite Impulse Response (FIR) Filter

In a fixed filter the coefficients Ci's are constant, whereas in an adaptive filter the coefficients are updated according to the desired adaptation algorithm. The coefficients are therefore a function of time, and the ith coefficient at time t is written as $C_i(t)$. If we consider the simple Least Mean Square (LMS) or Stochastic Gradient adaptation algorithm, which is the most widely used one, a coefficient at time (t+1) is updated according to :

$$C_i(t+1) = C_i(t) + \beta \cdot e(t) \cdot x(t-i)$$

where,

* $C_i(t+1)$ is the updated coefficient of stage i
* $C_i(t)$ is the old coefficient of stage i
* $x(t-i)$ is the input at stage i
* β is the step size, which determines the speed of adaptation, or convergence
* $e(t)$ is the error term, defined as the difference between the desired signal and the estimated signal.

Adaptive algorithms such as the LMS can be used in order to improve the performance and limitations of a number of peak detection algorithms. For example, in the curve fitting methods [6,11] introduced in section 2.5.1 of this report, the LMS algorithm can be used for improving the fitting procedure. The LMS algorithm has also been used in [8,9] and in chapter 2 of this report in order to implement the best fitting Gaussian curve onto the experimental IMS data files and therefore determine the signal to noise ratio and other parameters of the files.

Another approach is to take advantage of some known information and accordingly estimate other parameters. For example if a substance with known drift time and known standard deviation is known to be in a mixture, this information could be used by an adaptive DSP subsystem in order to calibrate the measurement, automatically. The know signal could also be removed, or cancelled, from the IMS output and therefore improving the resolution of the remaining peaks.

Adaptive techniques could also be used for removing unwanted noise signals from the IMS output. For example, other sensors could be installed in order to detect possible noise sources such as vibration or interference. Adaptive noise cancellation techniques [14,15] could then be used in order to estimate the noise on the IMS signal and attempt to remove it. This approach has been successfully used in other applications such as adaptive acoustic noise cancellation and echo cancellation. Figure 2.5 shows a block diagram of a typical noise canceller. The primary input represents the signal obtained from the Ion Mobility Spectrometer, which includes the IMS information in addition to other noise sources, for example vibration related noise. The secondary input in this case represents the output of a vibration sensor. The adaptive filter would then attempt to model the transfer function between the primary and the secondary inputs and would therefore estimate the vibration noise on the primary input and attempt to remove it.

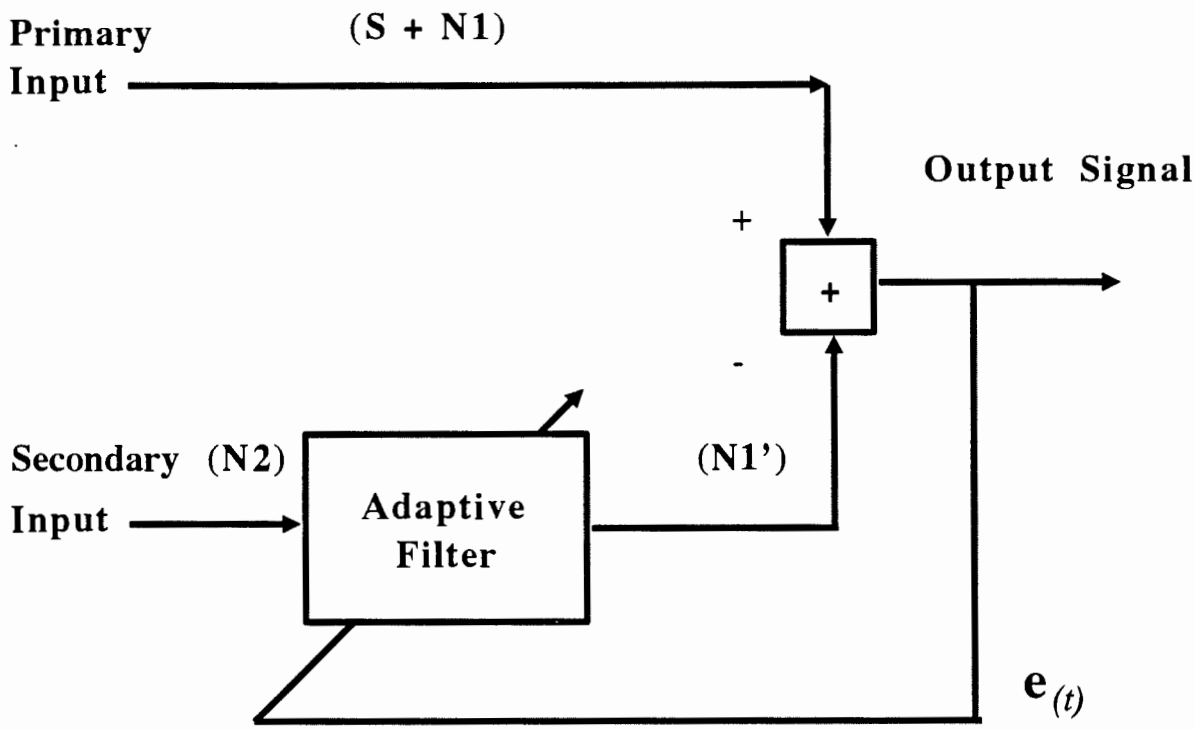


Figure 2-5 Adaptive Noise Cancellation

(Using Two Inputs)

3.0 DERIVATIVE METHODS

3.1 Introduction to derivative methods

Derivative methods are based on differentiating the IMS signal. Any variations in the slope of the original IMS signal results in large peaks in the second derivative. Therefore, slight slope variations in the original IMS signal due to weak or overlapping peaks can be detected using the second derivative method. Higher order derivatives, where the derivative of the IMS signal is again differentiated, emphasize any slight variations in the original signal.

However, any noise on the original IMS signal will also result in large peaks showing on the derivative curve. Derivative methods are therefore very susceptible to noise, and, any realistic second derivative algorithm has to be preceded by very precise filtering in order to remove the out-of-band noise. The performance of any second or higher order derivative algorithm depends critically on the stop band attenuation and the slope of the Low-Pass Filters (LPF) used.

Analog techniques have been widely used for implementing second derivative algorithms, however, the limited performance of the analog LPFs used limit their performance. In this work, second derivative algorithms were implemented using DSP, where more stringent filtering was performed.

The selectivity of the derivative methods is the highest among the various peak detection algorithms discussed in this report. However, due to their susceptibility to noise, derivative methods exhibit a high probability of false detection, unless their detection limit is reduced. In summary, derivative methods provide an excellent selectivity at the expense of a low detection limit or a high probability of false detection.

3.2 Second order derivative

A peak identification algorithm based on the second derivative method was developed [8]. Each Low Pass Filter (LPF) was implemented as a 130 taps Finite Impulse Response (FIR) transversal filter. The filter coefficients, calculated using the Remez Exchange algorithm, provide a cut-off frequency of 2.2 KHz and a stop band attenuation of -80 dB. It is important at this point

to emphasize that the performance reported in the later subsections depends critically on the shape, cut-off frequency, slope, and stop-band attenuation of the LPF. Tailoring the filter parameters to a specific experiment could improve the performance for this specific case, however, it might deteriorate the performance in other situations. For example, reducing the cut-off frequency of the filter slightly could improve the detection limit in the presence of one peak while reducing the selectivity. Each differentiator is implemented using a 24 taps transversal FIR configuration.

The IMS signal simulator described in section 2.4 was used to generate IMS signals exhibiting two peaks. The standard deviation of each peak was set at 0.18 msec, similar to the typical values of RDX obtained in the experimental tests. The peak separation was varied, as well as the amplitude ratio. Peak separations down to 0.27 msec in a high SNR environment and a 1:1 amplitude ratio were resolved. When the amplitude ratio was changed to 6:1, the minimum peak separation to be resolved was found to be 0.35 msec. The previous results depend very critically upon the standard deviation of the Gaussian peak and are therefore only valid for a peak standard deviation of 0.18 msec. The smaller the standard deviation, the better is the resolution. It is therefore possible to related the minimum resolvable peak separation to the standard deviation of the peak. The results indicate that second order derivative methods can resolve equal peaks separated by at least 1.5 of their standard deviation in a high signal to noise ratio (SNR) environment. In terms of sensitivity, single peaks were detected in an 18 dB SNR environment.

Figures 3.1 and 3.2 show the results of the application of second derivative methods to experimental IMS signals representing RDX. In each figure the solid line shows the IMS signal and the dotted line shows the second derivative. In figure 3.1, with an injected RDX quantity of 0.1 nanogram, the second derivative plot has a single distinct minimum in the peak region, which indicates a correct detection of the peak. When the injected RDX quantity is reduced to 0.05 nanogram, as depicted in figure 3.2, the second derivative plot shows several local minima in the peak area. This is an indication that the second derivative algorithm has failed and that the algorithm is confusing the RDX peak with noise.

Fig 3.1 RDX Sample and its Second Derivative

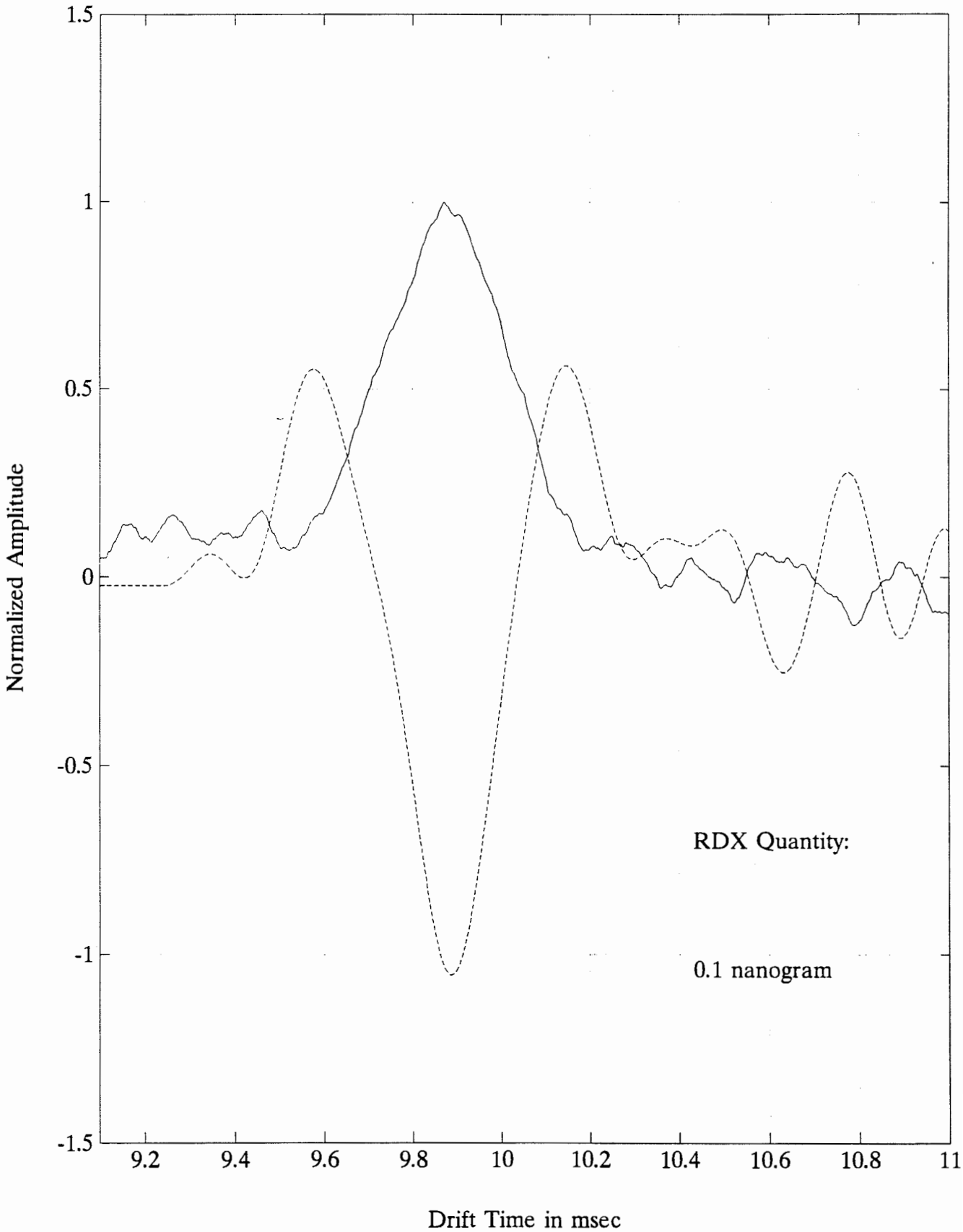
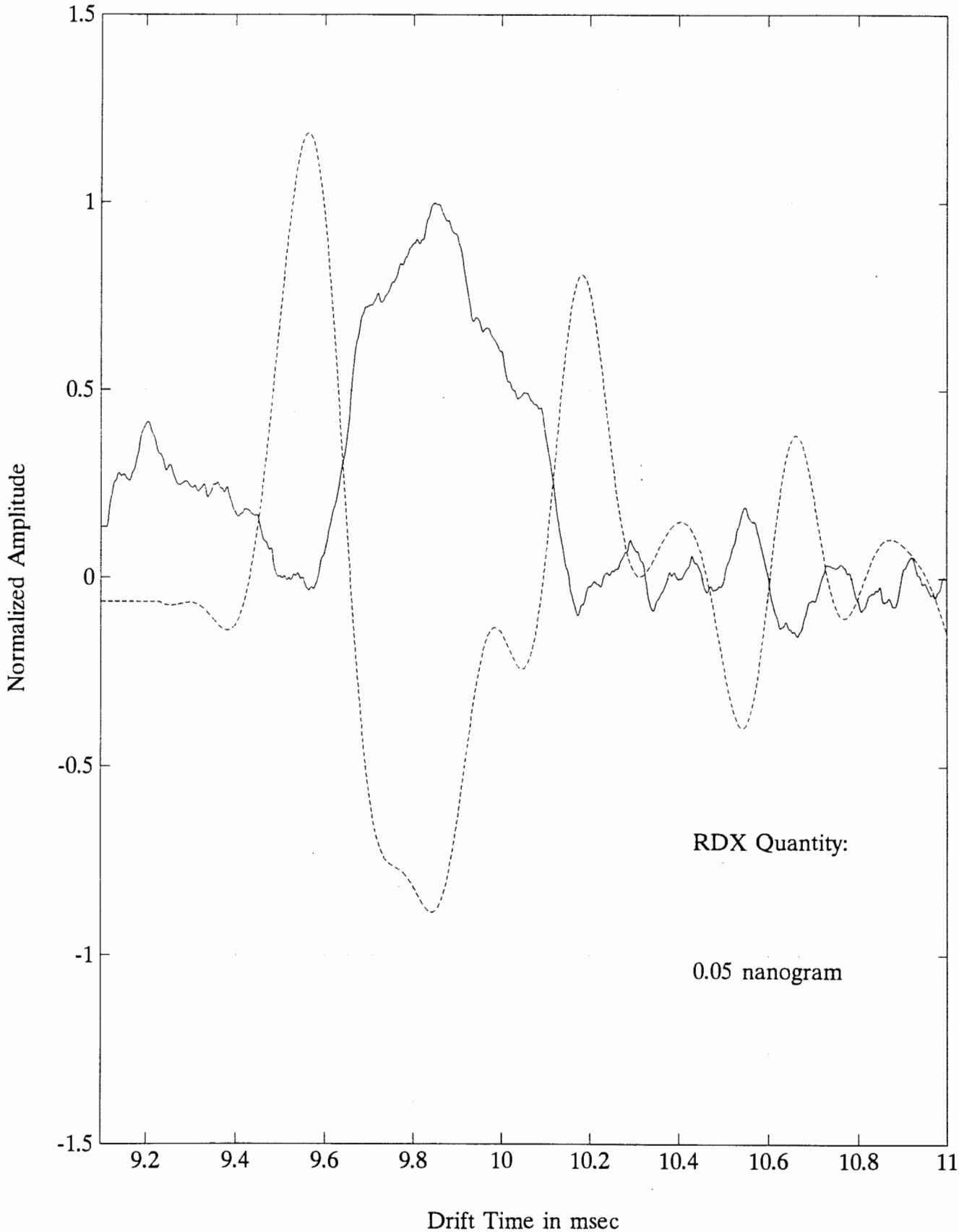


Fig 3 .2 RDX Sample and its Second Derivative



3.3 High order derivatives

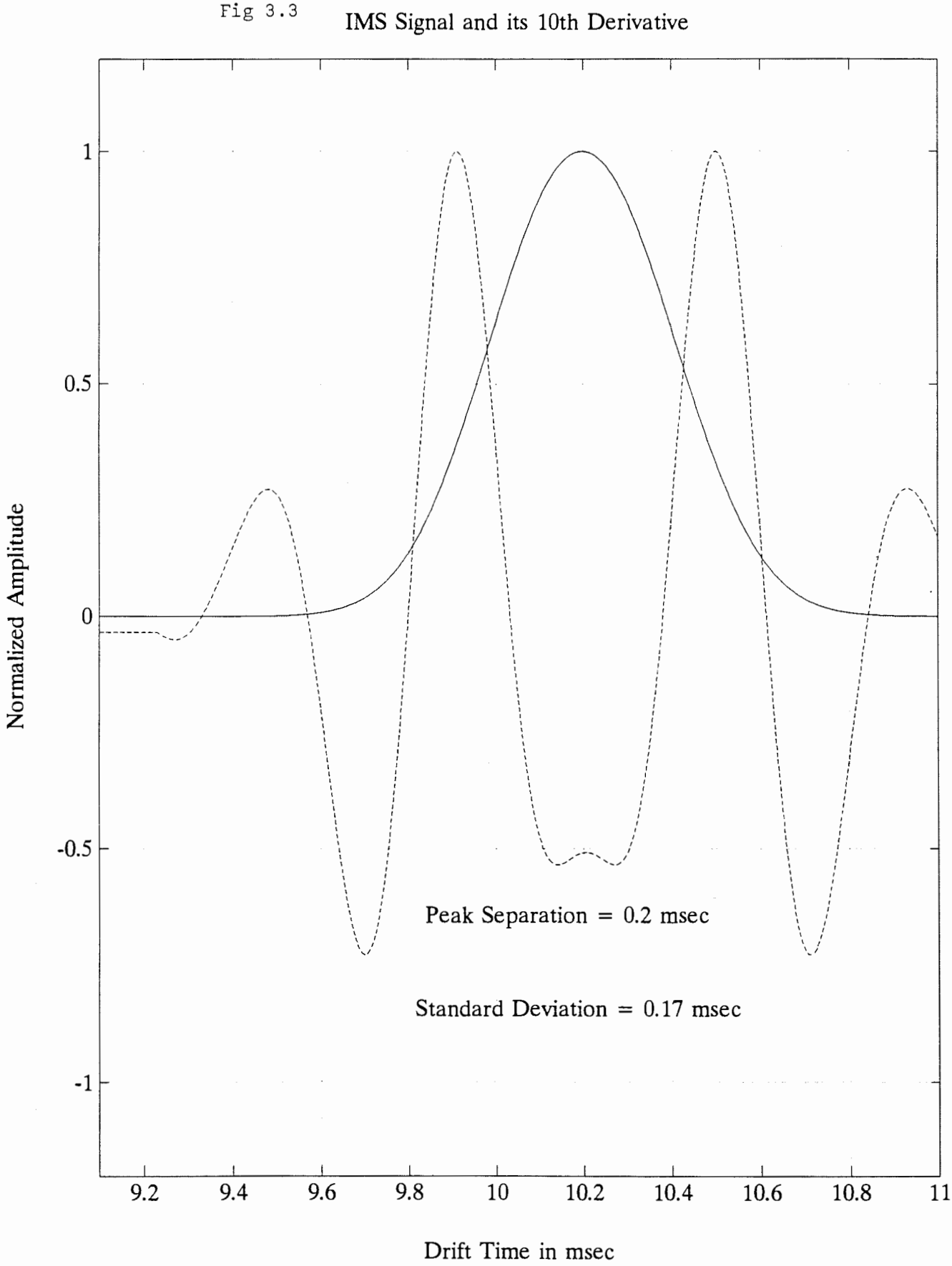
In higher order derivatives, the original IMS signal is again differentiated a number of times in order to emphasize and amplify any slight fluctuations in its slope. Each differentiation stage is followed by a LPF. An Nth order derivative is therefore generated by cascading N second derivative stages. Higher order derivatives are more susceptible to noise than second order derivatives [5], each differentiation stage decreases the SNR.

The results indicate that in the case of two Gaussian peaks with equal amplitude, equal standard deviation of 0.17 msec, and a peak separation of 0.23 msec, second derivative methods are not capable of detecting the peaks, whereas the 6th derivative shows, very clearly two distinct peaks. When the peak separation is reduced to 0.20 msec, neither the 2nd, 4th, nor 6th derivatives show any visible peaks, whereas when the 10th order derivative is used, figure 3.3, two distinct peaks can be observed. It is important to notice that the peaks in this experiment were pure peaks, with no additive noise. Therefore the results only determine the limitation of derivative methods in terms of peak separation. In practical cases any weak noise could result into unacceptable performance, as described in [5].

The resolvable peak separation is also a function of the standard deviation of the Gaussian peaks. In the previous experiments the standard deviation was set at 0.17 msec, a value similar to most chemicals considered. When the standard deviation is increased to 0.18 msec, the minimum resolvable peak separation, with no noise added, was found to be 0.21 msec.

Therefore, very high order derivatives are very powerful in terms of selectivity and resolution. However, in practical situations, their use makes the system very susceptible to noise which introduces unacceptable false detection probabilities. The poor detection limit of high order derivative methods restricts their use to high SNR environments. Nevertheless, such very high order derivatives could still be used in practical situations in conjunction with other peak detection algorithms such as cross-correlation methods.

Fig 3.3


IMS Signal and its 10th Derivative
Normalized Amplitude
1
0.5
0
-0.5
-1
9.2
9.4
9.6
9.8
10
10.2
10.4
10.6
10.8
11
Drift Time in msec
Peak Separation = 0.2 msec
Standard Deviation = 0.17 msec

4.0 CORRELATION METHODS

Given two sequences, or vectors x and y, each having a length (N), their cross-correlation function is a sequence of length (2 N - 1), and is defined as:

$$C(m) = \sum_{i=-N}^{i=N} x(i)\ y^*(i+m) \qquad (3)$$

The cross-correlation function is similar to a convolution operation for symmetrical signals. It shows how correlated the two sequences are for a relative shift of -N up to +N. When the two sequences are normalized the cross-correlation function is maximum when the two signals are identical and is zero when the two sequences are uncorrelated. If the two sequences, for example, are independent random numbers, or white noise, the cross-correlation function will be flat and will not show any peaks. If the sequence x represents a square peak embedded in white noise, while the sequence y represents a square peak similar to the one in x, the cross-correlation function will have a triangular peak at the position where the two squares coincide. The same case applies to a Gaussian peak embedded in white noise, such as the case in IMS. The optimum correlating sequence in that case would be a Gaussian shape.

In peak detection using cross-correlation, the output signal of the IMS is cross correlated with a given function. The choice of such function is a main factor in determining the system performance. Reference [7] analyzes three different functions, square, triangular, and Gaussian functions. It proves that with an even zero-area function the linear background component ($a_0 + a_1 t$) is completely filtered out, as opposed to the higher order components. Based on the RDX data files, this feature is very useful. It was found that all the RDX files have a large linear background component (a_1).

Since IMS peaks are Gaussian, the optimum cross-correlating function should also be Gaussian, with similar parameters. In this work, extensive simulation was undertaken in order to determine the optimal parameters of the cross-correlating Gaussian curve. Based on [7] an even zero-area curve is required. There are two parameters to be determined, the standard deviation of the curve and the base width. Given that most of the IMS data files analyzed throughout this work had a standard deviation around 0.16 msec, it was decided to experiment

with three sets of Gaussian curves, with standard deviations of 0.1, 0.16, and 2.00 msec. As far as the base width is concerned, it was thought that given that the data files have a span of about 2 msec, base widths of 2.0, 1.5, 1.0, and 0.5 msec would be considered which gives a total of 12 different cross-correlating functions.

Each one of the 12 cross-correlating Gaussian curves described before was cross-correlated with the experimental RDX files. It was found that with the 0.2, 0.1, and 0.05 nanogram quantities the algorithm was able to perform a correct detection with almost all cross-correlating curves. Proper selection of standard deviation and base width resulted in detecting concentrations down to 0.01 nanogram. Large values of standard deviation and base width tend to smooth the peaks, whereas small values extract the main peak but are also affected by the noise around it. Figures 4.1 and 4.2 show the effect of correlating the Gaussian curves (σ=0.10 msec BW=2 msec) and (σ=0.16 msec BW=2 msec) with an RDX file corresponding to a quantity of 0.01 nanogram.

Fig 4.1 Cross-Correlation with a Zero-Area Gaussian Curve

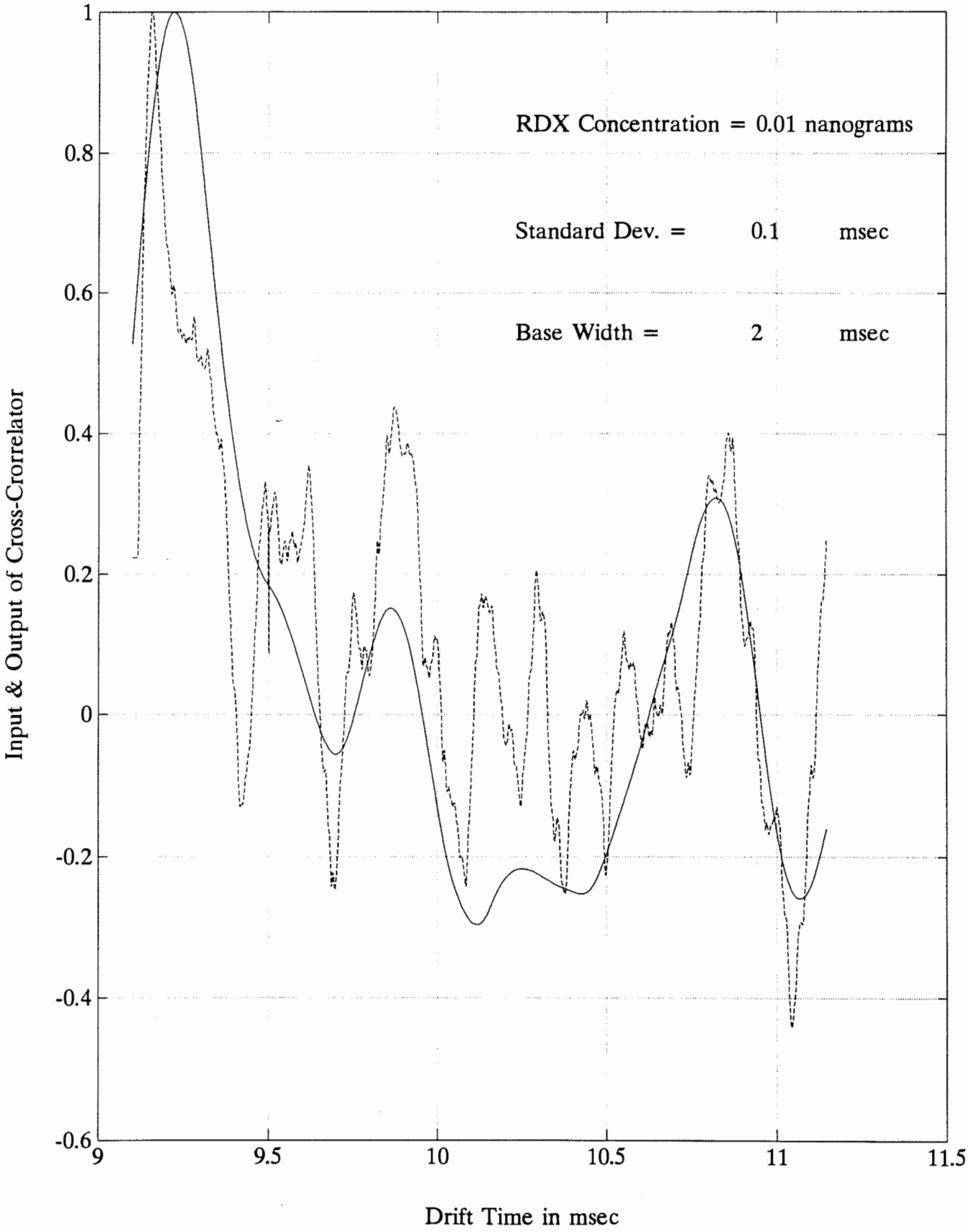
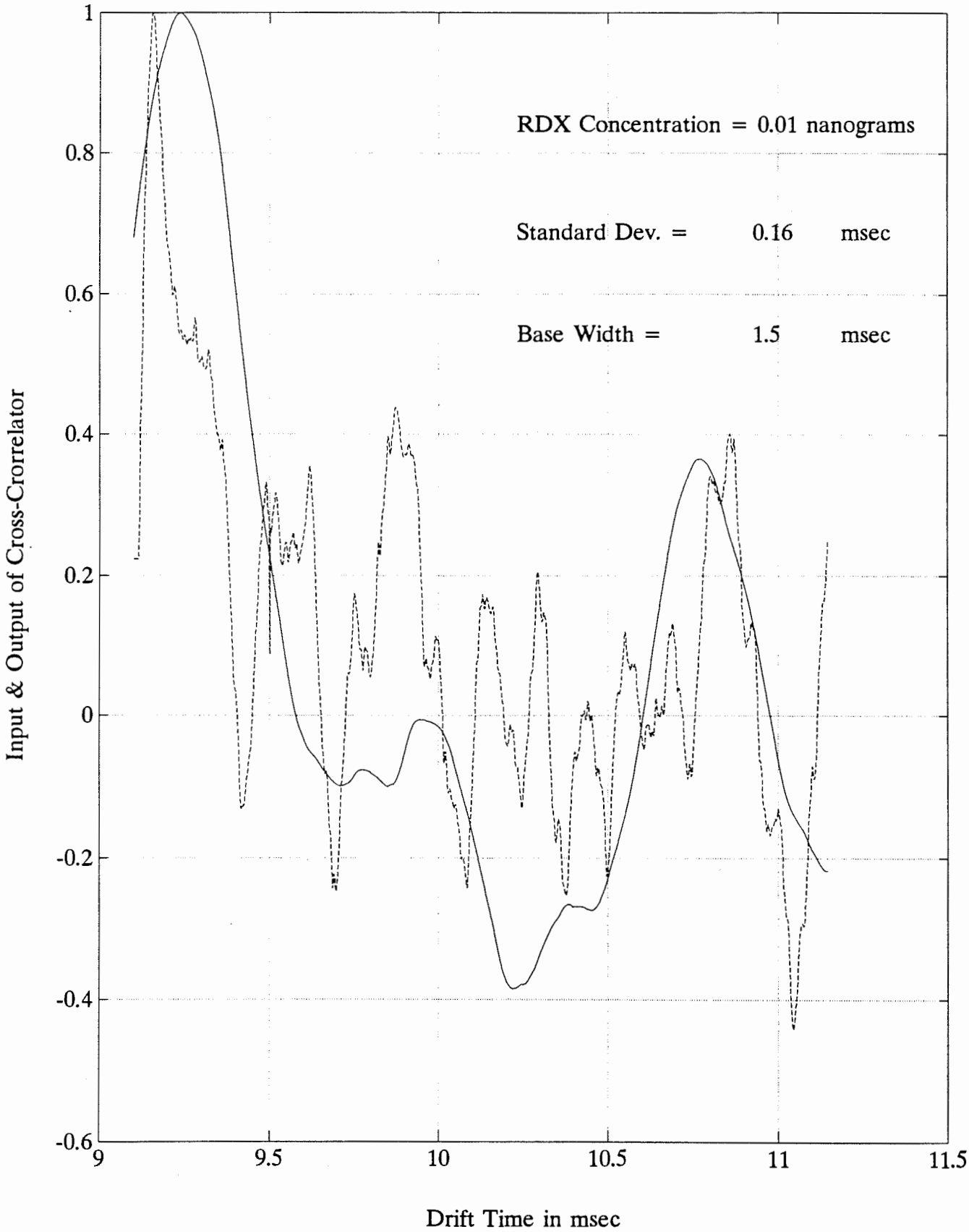


Fig 4.2 Cross-Correlation with a Zero-Area Gaussian Curve



5.0 SIGNAL DECOMPOSITION USING NEURAL NETWORKS

5.1 Introduction

The problem of detecting two or more closely spaced Gaussian pulses can be viewed as a signal decomposition problem. The objective is to observe a noisy recorded signal and decide on how many Gaussian pulses are in it, and also decide the amplitude, centre and variance of each detected pulse. If two pulses are separated by a sufficiently large time delay, the two peaks of the pulses will be visible and the detection problem is trivial. As the two pulses move closer to each other, their peaks will merge into a single peak. Under this condition, taking the first derivative of the recorded signal can be used to detect the presence of two pulses [1]. As the time delay between the pulses is further reduced, higher time derivatives may be needed to perform the detection. But, time derivatives enhance the noise, and higher derivatives result in a noisy signal which obscure the detection as this method reaches its resolution limit.

In this section, the use of neural networks to perform the detection is being proposed and analyzed. The neural network approach is different from the more conventional linear signal processing approaches, but it is based on the same mathematical basis. This section is organized as follow: In 5.2 the problem is formulated mathematically to illustrate the detection complexity when one is seeking a fine resolution. The problem is cast as a decomposition problem using Gaussian basis functions. Since closely spaced Gaussian pulses are not orthogonal, the analysis resulted in a set of cross-coupled equations, which are hard to solve analytically. In 5.3. a recursive approach is suggested to solve the equations. The recursive processing block diagram is very similar to a well known class of neural networks called "Hopfield Networks" [17-19]. In 5.4. we describe the Hopfield approach, and develop all equations needed to simulate the system. The details of an experimental simulation study of Hopfield network are given in 5.5.

5.2 Problem Formulation

Let r(t) be the noisy recorded signal:

$$r(t) = x(t) + z(t) \quad ; 0 \le t \le t_{max} \tag{4}$$

where x(t) is the signal of interest (SOI) and z(t) is additive noise. The recorded signal is sampled at a rate of 1/T samples per second to form the sequence {r(n); n=1,2,..,N}, where r(n)=r(t=nt) and $N=t_{max}/T$.

x(t) is assumed to be made up of Gaussian pulses of the form:

$$g(t;l,m,k) = a_l \exp\left[-\frac{(t-c_k)^2}{\sigma_m^2}\right] \tag{5}$$

where $\{a_l\}$, $\{c_k\}$ and $\{\sigma_m\}$ are sets of discrete amplitudes, centres and standard deviations; and

$$\begin{aligned} a_l &= l.\Delta a \quad ; l=1,2,..,L \\ c_k &= k.\Delta t \quad ; k=1,2,..,K \\ \sigma_m &= m.\Delta\sigma \quad ; m=1,2,..,M \end{aligned}$$

Δa, Δt and $\Delta\sigma$ are the amplitude, displacement and standard deviation resolutions, respectively.

Now, x(t) can be written as:

$$x(t) = \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{m=1}^{M} A_{lkm}\, g(t;l,k,m) \tag{6}$$

where the parameter $A_{lkm}=1$ if g(t;l,k,m) is present in the recording and equal zero otherwise. Our objective is to obtain the values of $\{A_{lkm}\}$ from the noisy record.

Substitute Eqn.(6) in (4) (written in a sampled form):

$$r(n) = \sum_{l=1}^{L} \sum_{k=1}^{K} \sum_{m=1}^{M} A_{lkm} g(nT; l, k, m) + z(n) \quad ; n=1,2,..,N \tag{7}$$

We can isolate any given parameter ,$A_{l^*k^*m^*}$, in Eqn.(7) by multiplying both sides of the equation by the reciprocal of g(Nt;l^*,k^*,m^*) and re-arranging the terms:

$$A_{l^*k^*m^*} = \frac{r(n)}{g(nT; l^*, k^*, m^*)} - \sum_{l \neq l^*} \sum_{k \neq k^*} \sum_{m \neq m^*} A_{lkm} \frac{g(nT; l, k, m)}{g(nT; l^*, k^*, m^*)} - \frac{z(n)}{g(nT; l^*, k^*, m^*)} \tag{8}$$

We can extend Eqn.(8) by summing over all terms in the recorded signal. This will result in a transform term plus a cross-coupling term and a noise term:

$$A_{l^*k^*m^*} = I_{l^*k^*m^*} - C_{l^*k^*m^*} + Z_{l^*k^*m^*} \tag{9}$$

where

$$I_{l^*k^*m^*} = input\ term = \frac{1}{N} \sum_{n=0}^{N} \frac{r(n)}{g(nT; l^*, k^*, m^*)} \tag{10}$$

$$C_{l^*k^*m^*} = cross\text{-}coupling = \frac{1}{N} \sum_{n=1}^{N} \sum_{l \neq l^*} \sum_{k \neq k^*} \sum_{m \neq m^*} A_{lkm} \frac{g(nT; l, k, m)}{g(nT; l^*, k^*, m^*)} \tag{11}$$

$$Z_{l^*k^*m^*} = noise\ term = -\frac{1}{N} \sum_{n=1}^{N} \frac{z(n)}{g(nT; l^*, k^*, m^*)} \tag{12}$$

Eqn.(9) defines the signal decomposition process, and illustrates its complexity.

If we ignore the noise term, Eqn.(9) will be reduced into a deterministic set of "LKM" cross-coupled equations in "LKM" binary unknowns; namely the set {A_{lkm}; l=1,2,..,L; k=1,2,..,K; m=1,2,..,M}. It is difficult to find an analytical solution to this set of equations. Instead, we propose to use a neural network approach to find the solution iteratively as explained in the section 5.3.

5.3 Recursive Solution

It is useful to think of the "LKM" equations as a multi-input/multi-output process as shown in Figure 5.1. The input vector is the N-dimensional recorded signal $\mathbf{R}=\{r(n); n=1,2,..,N\}$. The output is the binary vector $\mathbf{A}=\{A_{lkm}, l=1,2,..,L; k=1,2,..,K; m=1,2,..,M\}$. The input vector, **R**, is processed according to Eqn.(10) to produce the input-related terms. The output vector, **A**, is processed according to Eqn.(11) to produce the cross-coupled terms. The difference between the two sets of terms is a new value for the vector, **A**. Since **A** is binary, values of **A** computed in each cycle is quantized to 0 or 1 through a set of threshold devices.

The solution illustrated in Figure 5.1 is an iterative one. We could start the algorithm by setting the vector **A** to an initial value $\mathbf{A}(0)=\mathbf{0}$. Then, the cross-coupling terms will all be zeros and the first estimate of **A**, **A**(1), will be due to the input vector only. As we iterate **A** through the cross coupling processor, its value will change from cycle to cycle. When we hit the right decomposition, the vector **A** will remain unchanged.

The system structure shown in Figure 5.1 is similar to a well known neural network structure called Hopfield net [19]. In fact, a problem similar to the one considered in this work has been considered by Hopfield et.al.[20]. In the next section, a description of Hopfield network is given.

5.4 Hopfield Networks

Hopfield network is a member of an emerging signal processing technology called "neural networks". The word "network" in the name is there to signify the similarity between this type of processing and some biological neural systems. Although artificial neural networks are far inferior than their biological counterparts, the two types of networks share one important characteristic; that is the processing in done in parallel using a large number of simple, identical processors. This is in sharp contrast with today's computers and signal processing systems, which are invariably built around a single powerful processor called the Central Processing Unit (CPU). Another similarity between artificial and biological neural networks is that both learn their function by training rather than a sequential instructions imposed by an external entity. Again, the sequential instructions (or software) is the basis of operation of today's computers.

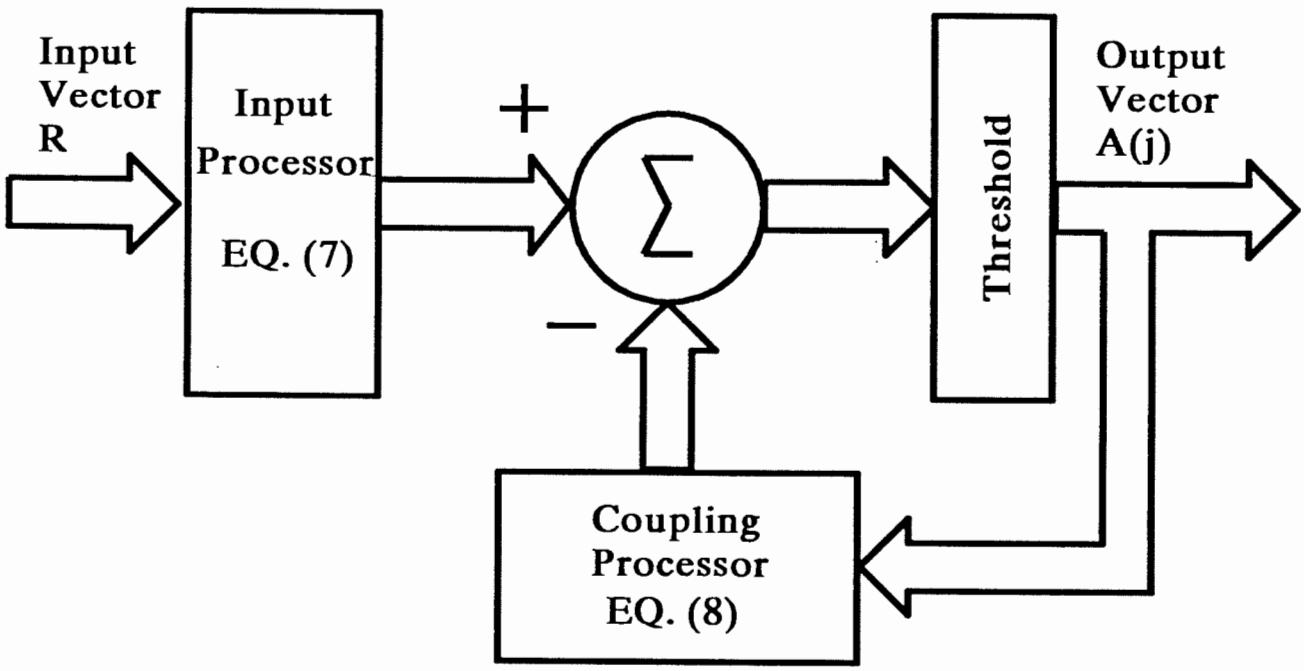


Figure 5.1- The Signal Decomposition Process

A more comprehensive discussion of neural networks is outside the scope of this report. Here, we will limit the discussion to Hopfield network only, and we will look at its structure as just a one possible way to solve a number of simultaneous equations.

Hopfield network has the re-current structure shown in Figure 5.2. The network consists of a single layer of Q simple identical processors called neurons. Each neuron adds up all of its inputs and compares the sum to a threshold value. If the sum is greater than the threshold, the neuron output is set to one, otherwise it is set to zero. Each neuron output is fed back to the inputs of all neurons except its own input. The feedback connections are called weights. For example, the feedback connection between the output of neuron k and the input of neuron m is the weight W_{km}. Consequently, each neuron receives (Q-1) feedback signals at its input, and the sum of these feedback signals is of the form $Y_k = \Sigma\ W_{nk}.A_n$. All the feedback connection weights and threshold values are either pre-computed or adaptively adjusted to solve an optimization problem which suits the application being considered. The neuron also receives signals from an external source through a fixed set of connections. The sum of inputs from the external source form a bias term which remains constant while the network iterates to find the best output vector. For the kth neuron, the input bias term, I_k is a function of R_n: $I_k=I_k(R_n)$. Then, the total input to the kth neuron is:

$$\begin{aligned} X_k &= \textit{feedback term} + \textit{bias term} \\ &= Y_k + I_k \\ &= \sum_{q=1}^{Q} W_{qk}A_q + I_k(R_n) \end{aligned} \qquad (13)$$

The next issue is how to determine the connection weights $\{W_{nk}\}$ and the relation between I_k and R_n?. The answer to this question provides the essence of Hopfield approach, which is simple but quite powerful.

Hopfield network finds a good solution to a stated problem by minimizing an artificial energy function, E, related to the problem. The energy, E, is a measure of how far the value of the output vector **A** is from an acceptable solution. The initial value of **A** corresponds to a high energy state (i.e. the solution is not acceptable). The energy, E, which is a function of the output

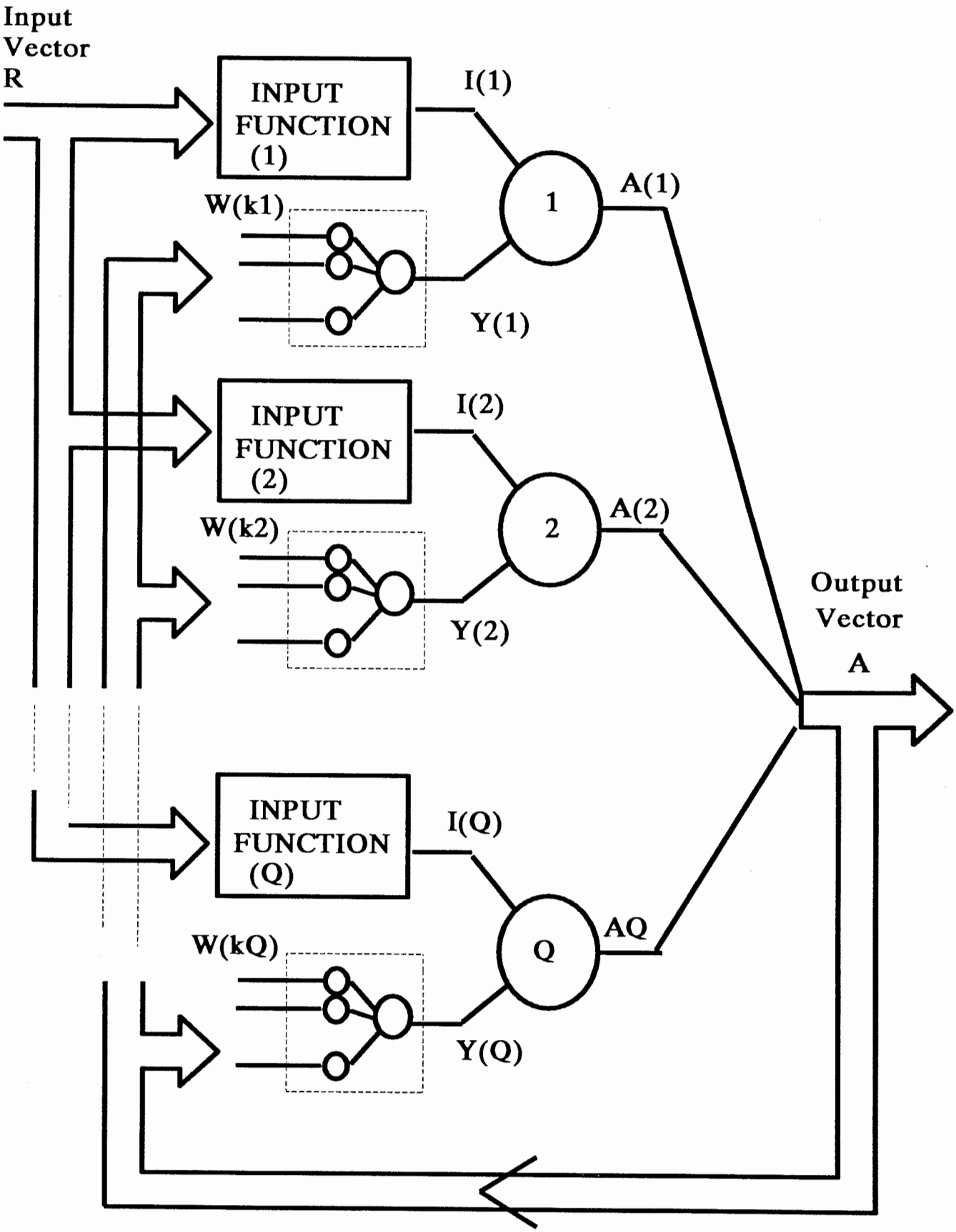


Figure 5.2- Hopfield Network

vector, **A**, is designed such that any change in **A** either reduces E or leaves it unchanged. Therefore, each iteration can only reduce the energy and bring the output closer to a good solution.

Selecting an appropriate energy function is the most critical step in designing the network. E is selected as a function of **A** to satisfy certain constraints, or certain description of an acceptable solution, and since **A** is a function of the connection weights, the weight can be computed from the energy function. We will demonstrate the procedure of selecting E and calculating the weights through our problem of signal decomposition.

In the model presented in section 5.2, the input vector, **R**, is a sampled version of a recorded signal with N samples. The output vector, **A**, is a binary vector. Each element in **A** is an indicator of the presence or absence of a Gaussian pulse with specific amplitude, centre and variance. There are LKM possible Gaussian pulses, and the Hopfield network should contains KLM neurons. Therefore, we will set the length of the output vector **A** to Q=KLM. The energy function of this problem will be selected to satisfy three requirements:

(1) If we construct a signal using the output vector, the difference between the constructed signal and the original recording should be small.

(2) All elements of the vector **A** are binary [0 or 1]

(3) The matrix of the feedback connections should have zeros on the diagonal.

The first two requirements are obvious. The third requirement is applicable to all Hopfield networks. It is a condition that guarantees the stability of the solution as explained in [2].

To satisfy the first condition, we create an energy term that represents the difference between the original and reconstructed signals:

$$E_1 = \sum_{n=1}^{N} \left[r(n) - \sum_{q=1}^{Q} A_q g(nT; q) \right]^2 \qquad (14)$$

The structure of E_1 satisfies the first condition, but it does violate the last condition, since squaring the summation inside the brackets results in diagonal terms. The second term of the

energy function E_2 is designed to cancel out the diagonal terms and at the same time favours a binary output:

$$E_2 = -\sum_{n=1}^{N}\sum_{q=1}^{Q} g^2(nT;q)\, A_q (A_q - 1) \qquad (15)$$

Since the range of A_q is between 0 and 1, E_2 will be minimum (equal to zero) when all A_q are either 0 or 1. Any other value of A_q (between 0 and 1) will result in a positive increase in E_2. Also notice that using $g^2(Nt;q)$ as coefficients produces $A_q^2 g^2(Nt;q)$ terms which cancel out the diagonal terms in E_1.

The energy function, E, is the sum of E_1 and E_2:

$$E = \sum_{n=1}^{N}\left\{\left[r(n) - \sum_{q=1}^{Q} A_q g(nT;q)\right]^2 - \sum_{q=1}^{Q} A_q (A_q - 1)\, g^2(nT;q)\right\} \qquad (16)$$

Eqn.(16) can be re-arranged into the following form:

$$E = \sum_{n}\sum_{q}\sum_{m\neq q} A_q A_m G_q G_m + \sum_{n}\sum_{q} A_q\left[G_q^2 - 2\, r(n)\, G_q\right] + R^2 \qquad (17)$$

where

$$R^2 = \sum_{n=1}^{N} r^2(n) \qquad (18)$$

and

$$G_q = g(nT;q) \qquad (19)$$

We note that the term R^2 is constant and independent of the output vector, **A**; so we may drop it without changing the optimization procedure. This gives us the final form of E as a function of the basis functions {g(nT;q)}, the input vector, **R** and the output vector, **A**:

$$E = \sum_{n}\sum_{q}\sum_{m\neq q} [G_m G_q] A_m A_q + \sum_{n}\sum_{q}\left[G_q^2 - 2\, r(n)\, G_q\right] A_q \qquad (20)$$

The next stage of this procedure is to relate the energy function as written in Eqn.(20) to the connection weights of Eqn.(13); namely, $\{W_{ij}\}$ and $I_k(R_n)$. We start by writing the energy function in terms of $\{Y_q\}$ and $\{I_q\}$ (refer to Eqn.(13) and Figure 5.2):

$$E = -\left[\sum_{q=1}^{Q} Y_q A_q + \sum_{q=1}^{Q} I_q A_q\right] \tag{21}$$

where

$$Y_q = -\sum_n \sum_{m \neq q} G_m G_q A_m \tag{22}$$

and

$$I_q = \sum_n [2r(n)\, G_q - G_q^2] \tag{23}$$

Comparing Eqn.(13) to (22) reveals that the feedback connection weights are:

$$W_{mq} = -\sum_n G_m G_q \tag{24}$$

The final step in this analysis is to show that the energy function as expressed in Eqn.(21) satisfies the objective of producing a correct output. To illustrate this point, we only need to show that any permissible change in **A** (from iteration to iteration) must lower E, because reducing E brings the output closer to the correct answer. Take the first derivative of both sides of Eqn.(21) with respect to one of the output, A_q:

$$\Delta E = -[Y_q + I_q]\, \Delta A_q \tag{25}$$

But, the new value of A_q is simply the sign of the term between brackets. Then, ΔA can either be zero, or it can have the same sign as $[Y_q+I_q]$. Consequently, ΔE can either be zero (i.e. no change in the energy level), or it can be negative (i.e. lower energy).

This concludes the analysis of Hopfield network as a method for decomposing a signal into a number of Gaussian basis functions. In the remainder of this report, we will report on a simulation study performed on the basis of the analysis given above to test the capability of Hopfield solution, and to compare the resolution of this techniques to that of a more conventional technique based on higher order derivatives.

5.5 Simulation Study

The simulation study will be presented in a step by step fashion, which would allow the reader to review the analysis and methodology of section 3.

Step 1: Select the resolution and time parameters

N = the length of the input vector **R** = 1024

This means that the input signal consists of 1024 points

Δt = time resolution = N/Z

where Z is the number of Gaussian basis functions. This means that the minimum distance between two consecutive Gaussian pulses is Δt points (which will normally be set equal to the standard deviation). Z is also the number of neurons in Hopfield model. Figure 5.3 illustrates the set of Gaussian basis pulses. The function of Hopfield net is to determine which of them are present in the input signal.

$\Delta\sigma$ = standard deviation resolution = 0

There is no standard deviation tolerance. There is only one standard deviation.

Δa = amplitude resolution = 0

Only one amplitude is allowed a=1

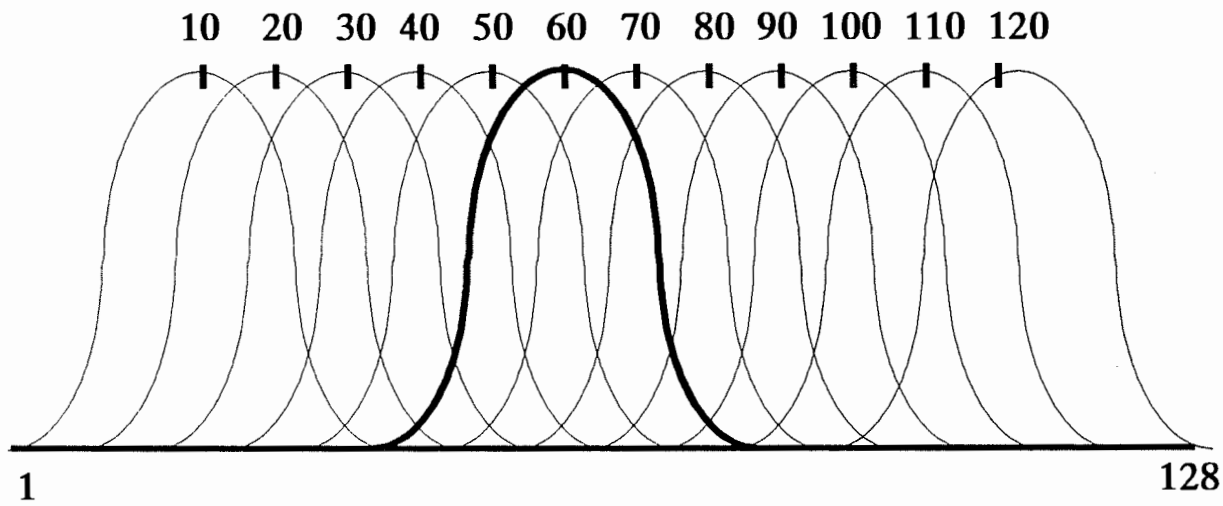


Figure 5.3 - The 12 Single-amplitude Single-variance Gaussian Basis Functions

In the illustrative examples shown here, Z will be set to 12, which means that $\Delta t=N/Z=85.33$ points. With this specific set of parameters, we only have 12 basis function (**Q=12**) of the form:

$$g(nT;k) = \exp\left[-\left(\frac{n-c_k}{85.33}\right)^2\right] \qquad (26)$$
$$\text{where} \;\; ; n=1,2,\ldots,1024$$
$$\ldots \;\; ; c_k=85,33*k$$

This also means that Hopfield network will have 12 neurons, one for each basis function.

Step 2: Compute the feedback connections

The weight connections $\{W_{ij}\}$ are calculated using Eqn.(24), and $\{G_k\}$ in Eqn.(24) are calculated according to Eqn.(22). The resulting equation is given below (Eqn.(27)). It should be noted the $W_{kk}=0$; i.e. the resulting matrix has zeros on the diagonal. The matrix has 12 x 12 elements, 12 of them are zeros. Also, the matrix is symmetric around the diagonal, which means that only 66 non-zero coefficients are to be calculated using Eqn.(27):

$$W_{ij} = -\sum_{n=1}^{1024} \exp\left[-\left(\frac{n-c_i}{85.33}\right)^2\right] . \exp\left[-\left(\frac{n-c_j}{10.24}\right)^2\right]$$
$$\text{where} \;\; i,j=1,2,..,12$$
$$,, \;\; c_i=85.33\,i$$
$$.. \;\; c_j=85.33\,j \qquad (27)$$

Step 3: Computing the input bias term

The input bias terms $\{I_q\}$ are calculated according to Eqn.(26), which is re-written here in details:

$$I_q = 2\sum_{n=1}^{1024}\left\{r(n)\exp\left[-\left(\frac{n-c_q}{85.33}\right)^2\right]-\exp\left[-2\left(\frac{n-c_q}{85.33}\right)^2\right]\right\}$$

$$\text{where} \quad ; \; c_q = q*85.33$$

While the feedback connections are calculated once for each set of basis functions (i.e. once per network), the feed-forward bias terms must be calculated to each new signal.

Step 4 The de-composition process

1 The signal $\{r(n)\}$ is applied to the input terminals, and the bias terms for this signal are computed as explained in step 3.

2 Initially the output vector $\mathbf{A}=\{A_q\}$ is set to zero; $\mathbf{A}(0)=\mathbf{0}$.

3 The first possible solution, $\mathbf{A}(1)$, is computed as $A_q=\text{sign}[I_q]$.

4 $\mathbf{A}(1)$ is fed-back through the feedback weights to form the $\{Y_q\}$ terms, which are added to the bias terms $\{I_q\}$ and the sum is used to generate the next estimate of $\mathbf{A}$ using: $A_q(2)=\text{sign}[I_q+Y_q]$.

5 Step 4 is repeated as many times as necessary until the vector $\mathbf{A}$ reaches an

equilibrium state: $\mathbf{A}(m+1)=\mathbf{A}(m)$. The equilibrium vector $\mathbf{A}(\infty)$ is the solution.

Test Signals:

The most critical step in preparing test signals is to simulate noise signals with the correct characteristics and to be able to accurately determine the signal-to-noise ratio (SNR). The noise, z(t), is assumed to be Gaussian and white. The power spectral density of the pre-filtered noise is flat with a height equal N_o, which extends from $f=-\infty$ to $+\infty$. In a sampled form, z(t) can be simulated as an un-correlated sequence of zero-mean Gaussian random numbers whose variance=N_o. z(t), as described above, has an infinite power. Much of this power is irrelevant to the detection problem at hand. Essentially, we must cut-out the frequency components in the noise spectrum that fall outside the effective bandwidth of the signal of interest. The relevant noise power is:

$$P_z = W \cdot N_o \qquad (29)$$

where W is the noise bandwidth.

A simple way to estimate the relevant noise is to compute the noise components that fall within a specified range around the peak. Normally, the noise is computed for a range of $+/- 3\sigma$ around the centre of the peak. In the case where more than one pulse is present and the distance between the two pulses is small (merged peaks) the same approximation remains good.

In the case of simulated data, the signal to noise ratio (SNR) is computed by first generating the signal (Gaussian pulses) and calculating their energy within $+/- 3\sigma$, and then generating the additive noise and computing its energy within the same range. Since we are only interested in the ratio between the signal and noise, the energy for either the signal or the noise can be calculated by adding up the squares of the discrete signals. In the case or real data, the signal and noise cannot be separated, and in this case we must estimate the SNR from the ratio of (signal+noise)/(signal only). The signal only part is estimated by the approximation of the signal as a Gaussian pulse with known parameters.

Test Signal Preparation Procedure:

Since all experimental data represent cases where the peak separations are greater the standard deviation, σ, we will use only simulated data to illustrate the improved resolution of the neural network technique.

1. Generate Gaussian pulses with different amplitudes, variances and centres and add them up to form a 1024 point signal vector: $\mathbf{x}=\{x(n);\ n=1,2,...,1024\}$.

2. Square each point in **x** and add up all the squares to obtain the signal energy.

3. Specify the desired SNR in dB's and convert it into a ratio $\gamma=10^{-SNR/10}$.

4. Determine the noise variance as: (signal energy)/(SNR)

5. Generate 1024 un-correlated samples of Gaussian noise with the correct variance (as calculated in step 4) and add them to **x** (sample by sample).

6. Filter the signal pulse noise vector (generated in step 5) in a lowpass filter which is wide enough such that it does not distort the signal but created correlated noise samples.

7. Estimate the filter delay (by numerical techniques) and compensate for the delay by back-shifting of the vector.

8. The result of step 7 is a vector with known SNR and partially correlated noise samples.

5.6 Results

This section describes the results of a series of tests designed to explore the advantages, disadvantages and performance limits of Hopfield method. The parameters used in the simulation are:

(1) Signal-to-Noise Ratio (SNR)

(2) Relative Amplitude: In the case of a single pulse, the amplitude is fixed at "1". In the case of two pulses, the amplitude of the two pulses are denoted by A1 and A2, where the "1" refers to the first (early) pulse and the "2" refers to the second pulse.

(3) Delay: The delay between the two pulses is either one standard deviation (denoted in the figure by "s") or less.

(4) Shift: Since the delays between the basis Gaussian pulses is fixed at "s", when the spacing between two pulses in the test signals is less than "s", one of the pulses will be shifted from its ideal centre location. The shift parameter refer to the displacement of a pulse from its corresponding basis function.

Sensitivity to Noise:

In this test one Gaussian pulse (with zero shift) is used. The noise level was increased gradually until the network failed to produce the correct output. The minimum SNR was found to be -3 dB. At this level, the noise power is twice the signal power. Reducing the SNR below this level resulted in an unreliable performance. Figure 5.4 shows the network input (top trace) and the output (bottom trace) at SNR=-3 dB. The horizontal axis is the delay in msec and the vertical axis is the relative amplitude. For illustrative purposes, the noise-free signal is superimposed on the noisy signal. Of course, only the noisy signal is fed to the network.

This result indicates that Hopfield network is very tolerant to noise, which means that even a minute quantity of compound could be detected by a neural network processor.

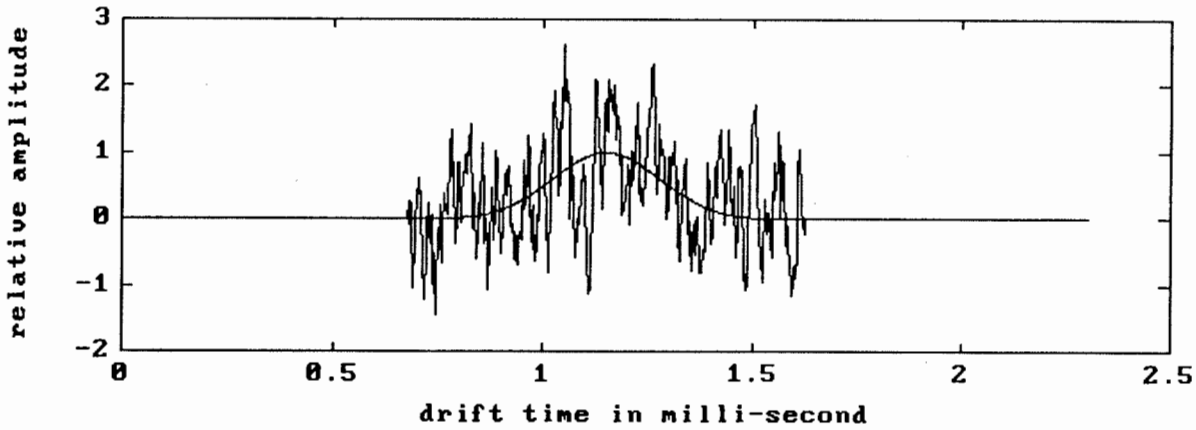
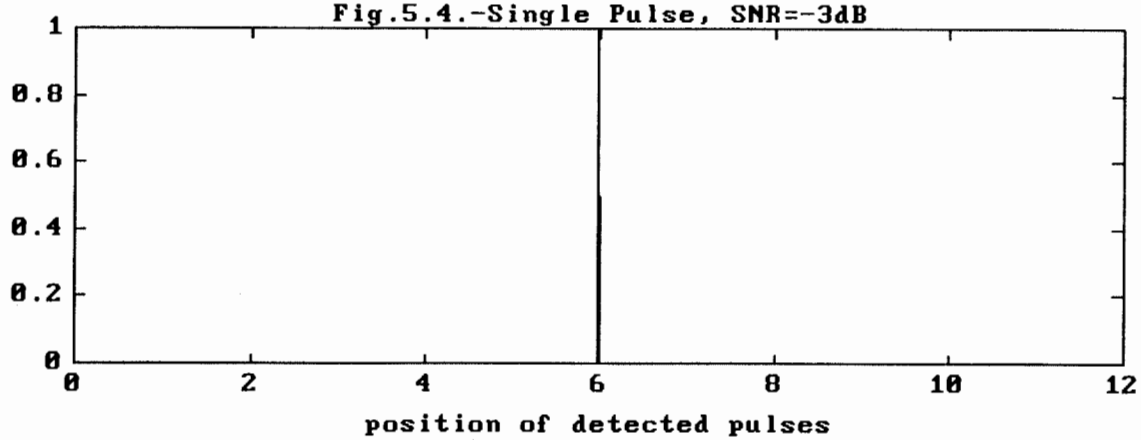


Fig.5.4.-Single Pulse, SNR=-3dB



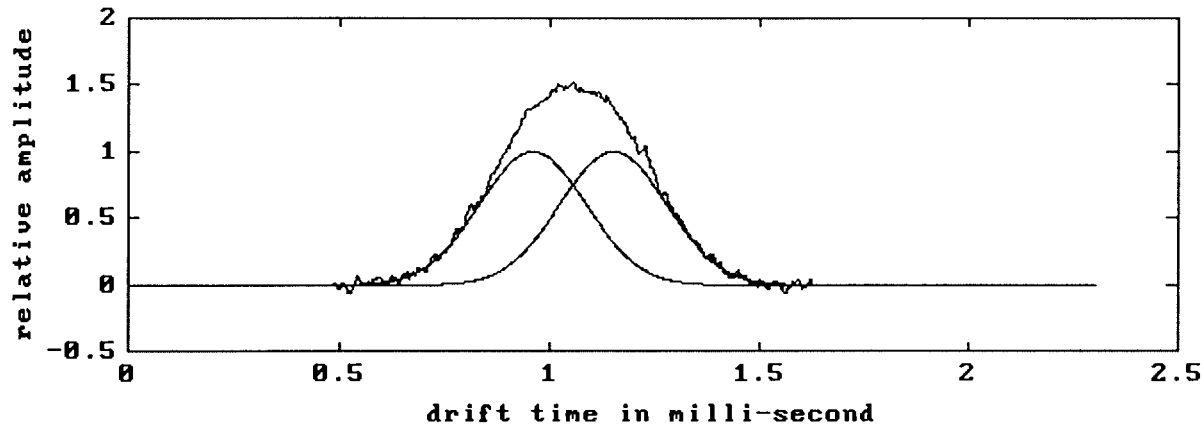
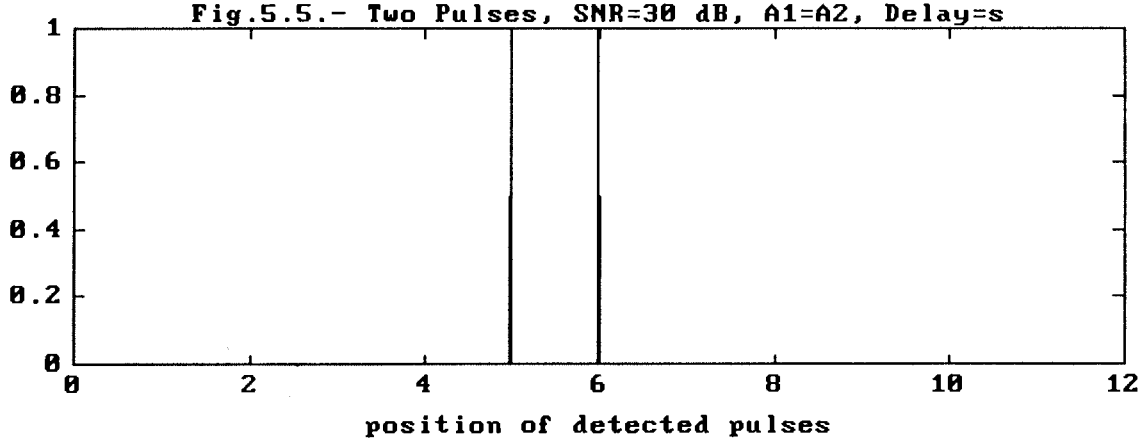


Fig.5.5.- Two Pulses, SNR=30 dB, A1=A2, Delay=s



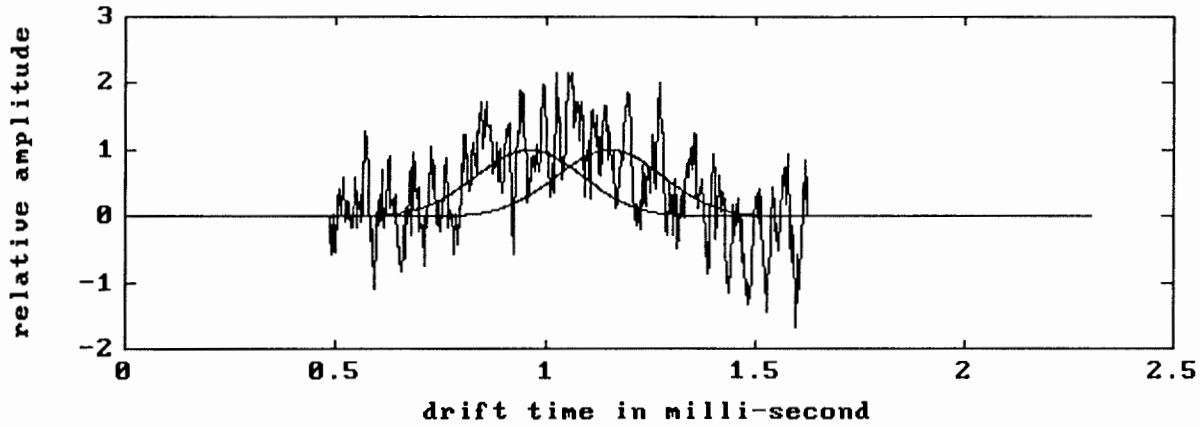
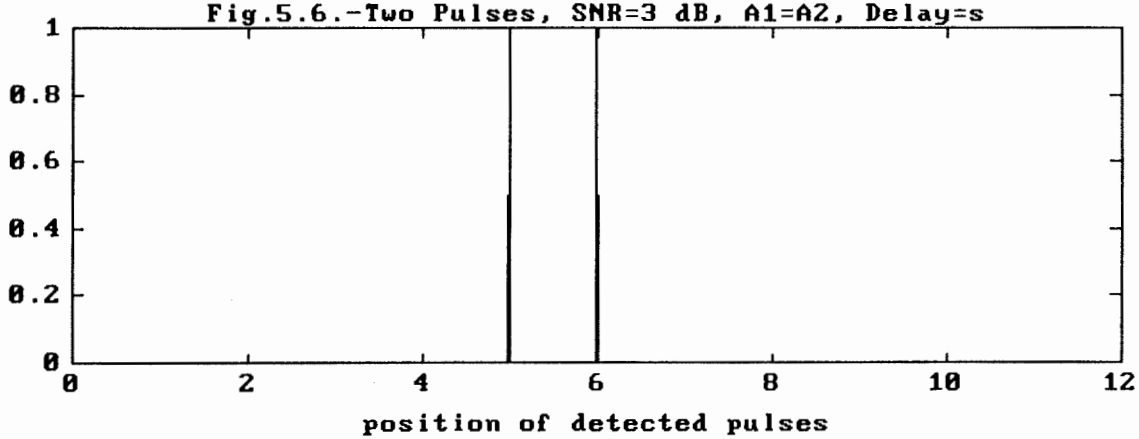


Fig.5.6.-Two Pulses, SNR=3 dB, A1=A2, Delay=s



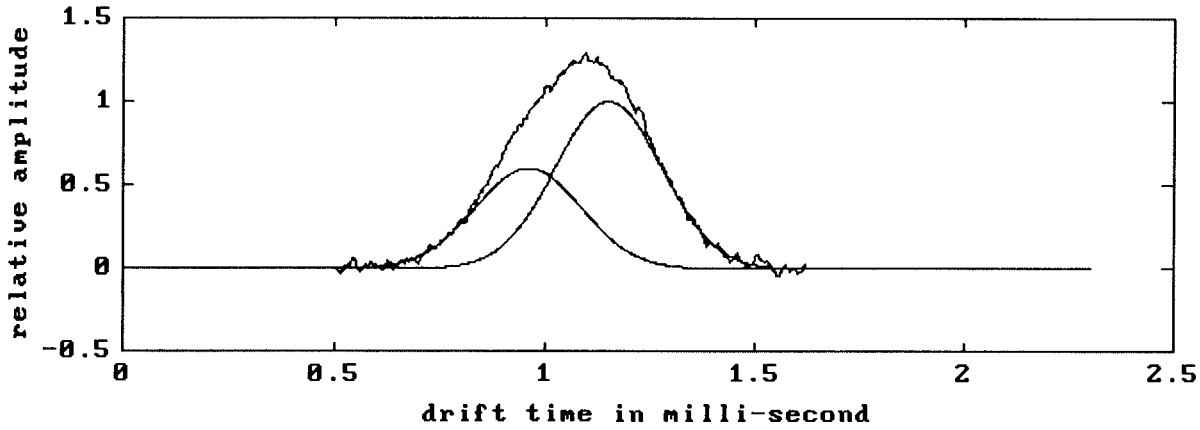
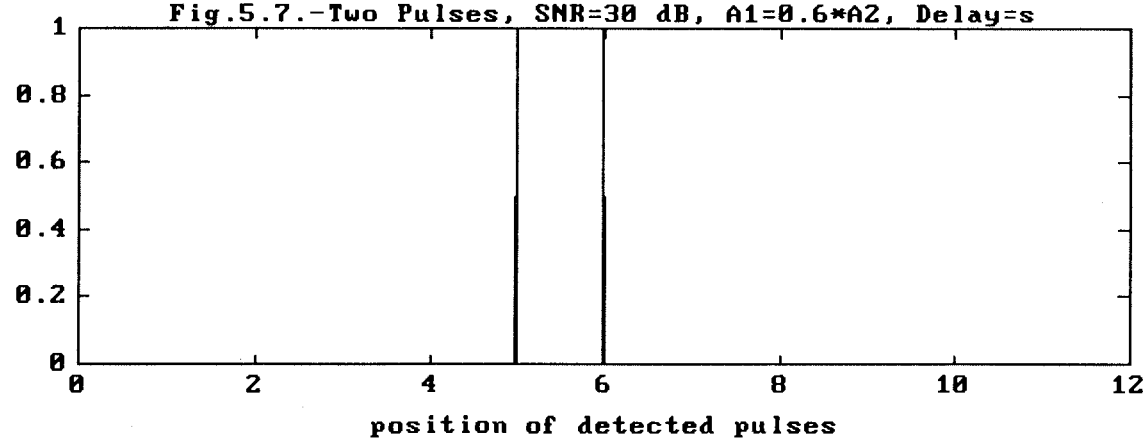


Fig.5.7.-Two Pulses, SNR=30 dB, A1=0.6*A2, Delay=s



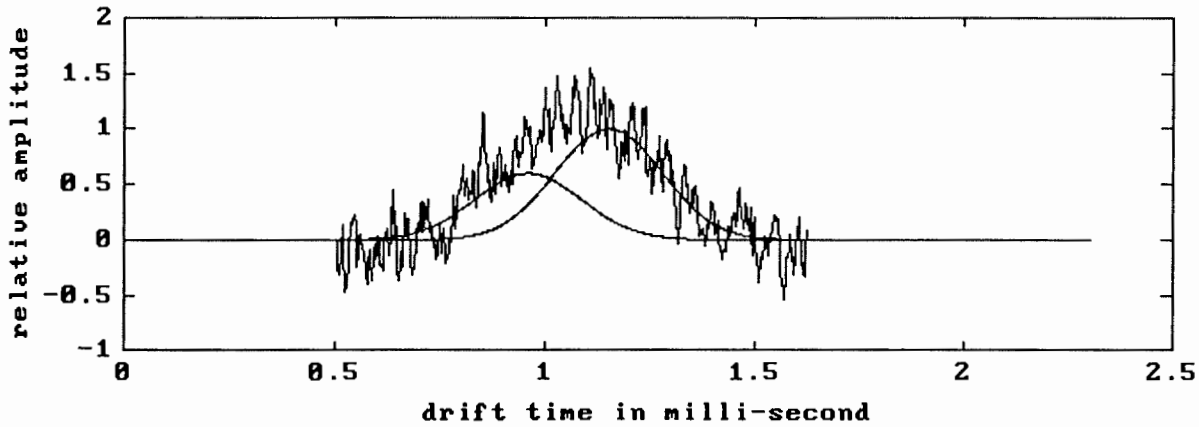
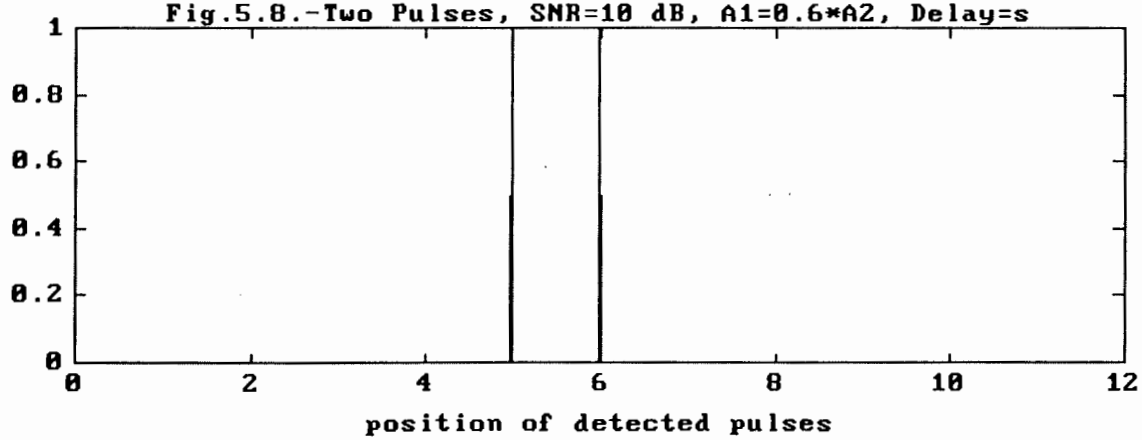


Fig.5.8.-Two Pulses, SNR=10 dB, A1=0.6*A2, Delay=s



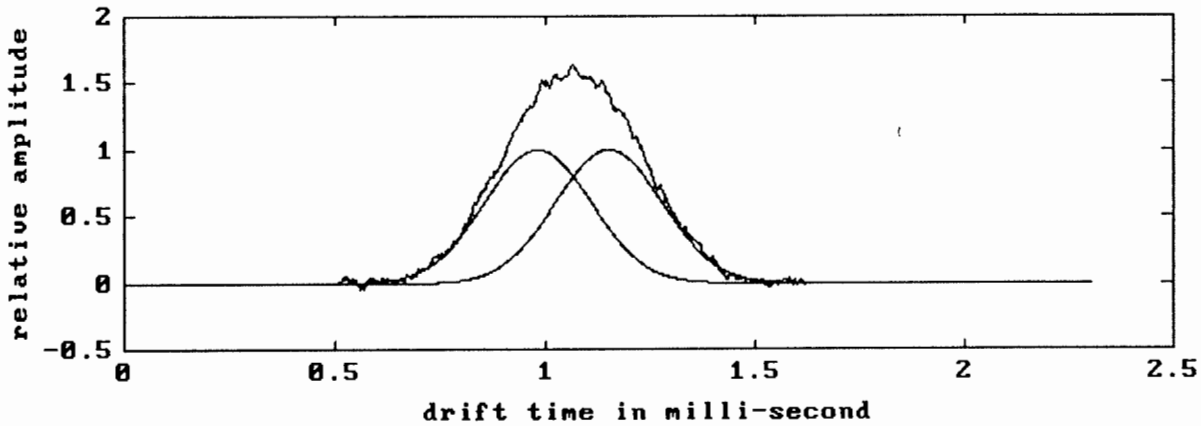
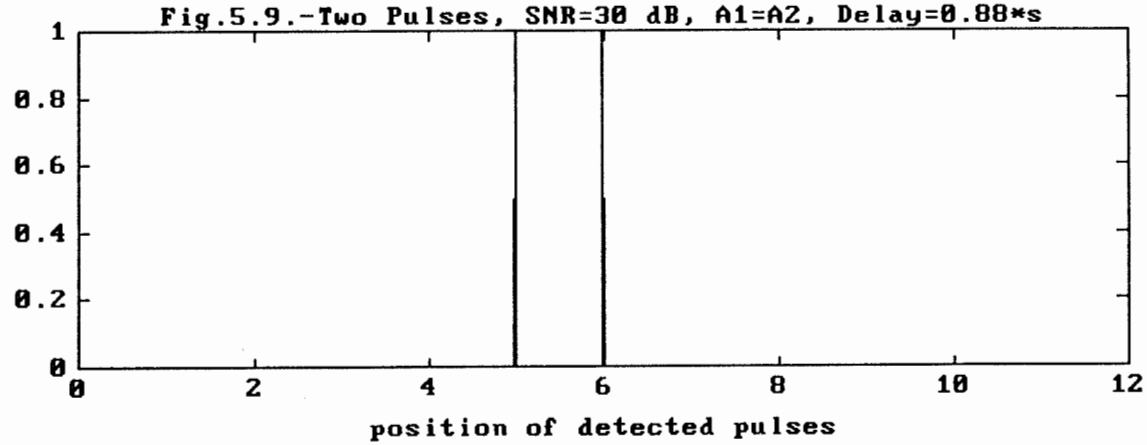


Fig.5.9.-Two Pulses, SNR=30 dB, A1=A2, Delay=0.88*s



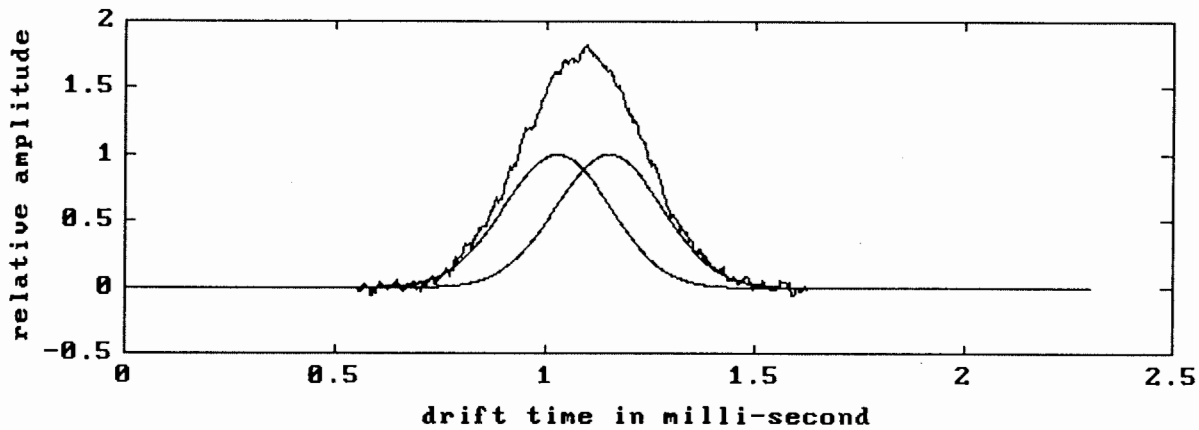
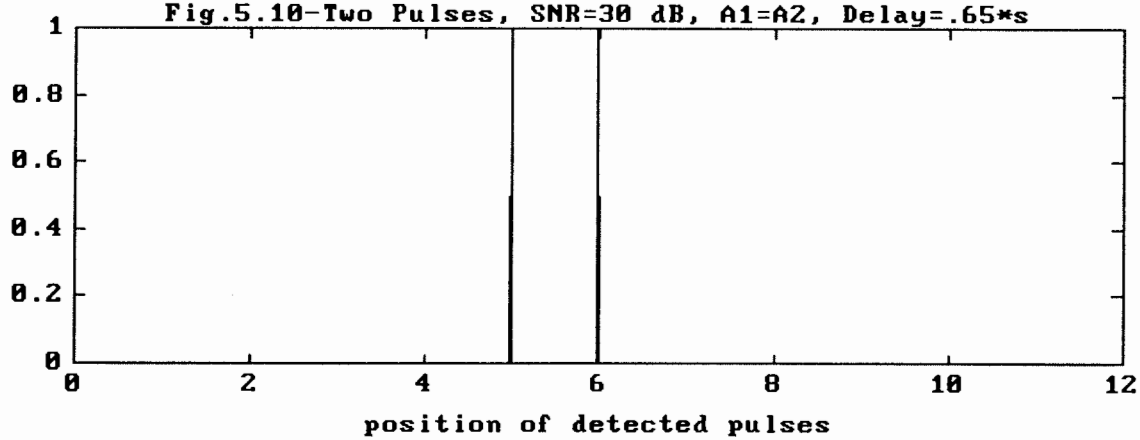


Fig.5.10-Two Pulses, SNR=30 dB, A1=A2, Delay=.65*s



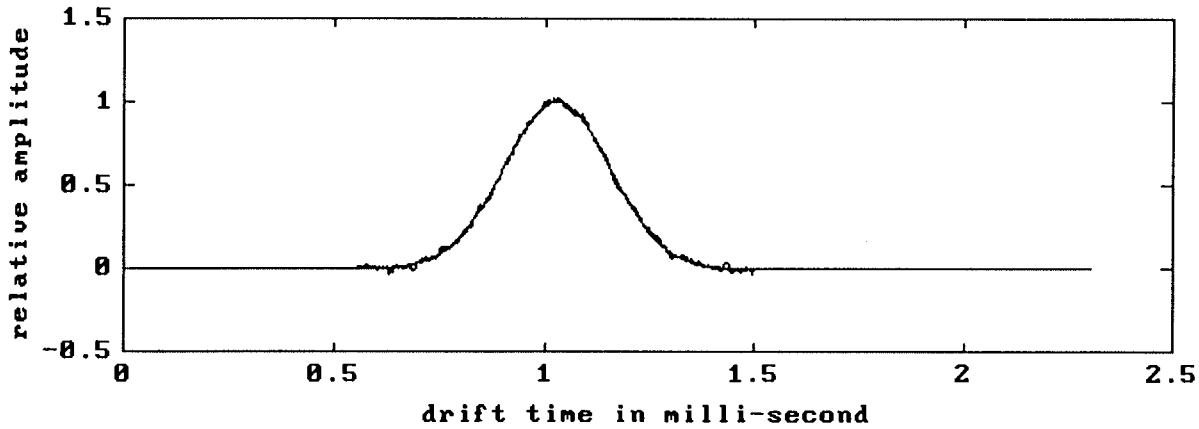
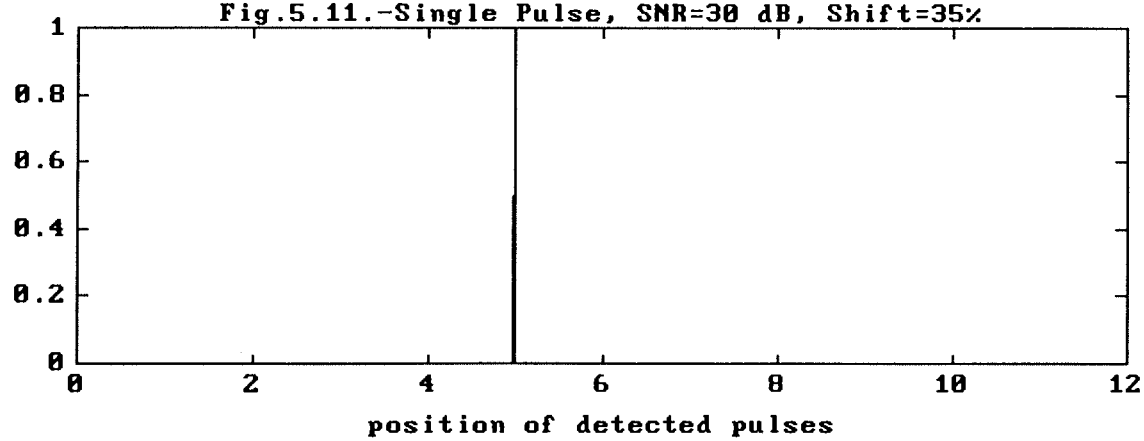


Fig.5.11.-Single Pulse, SNR=30 dB, Shift=35%



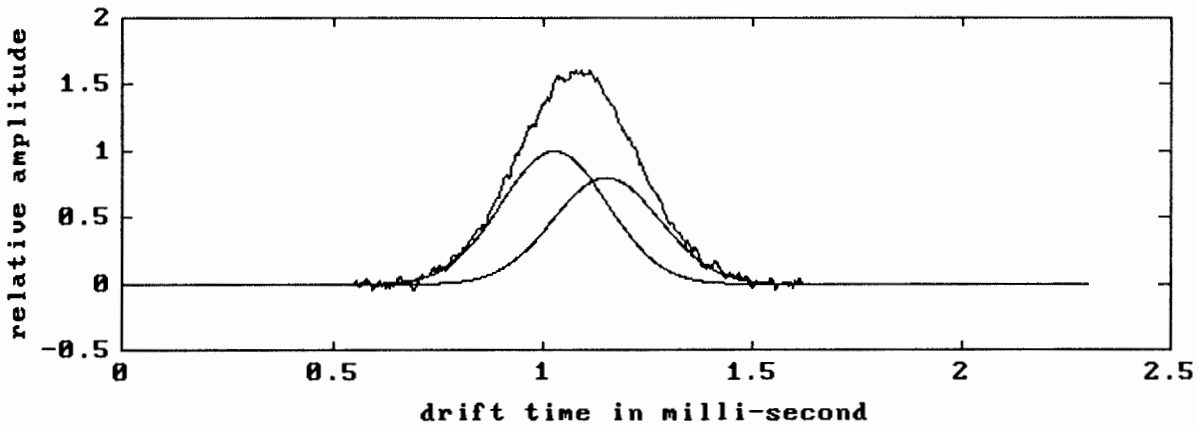
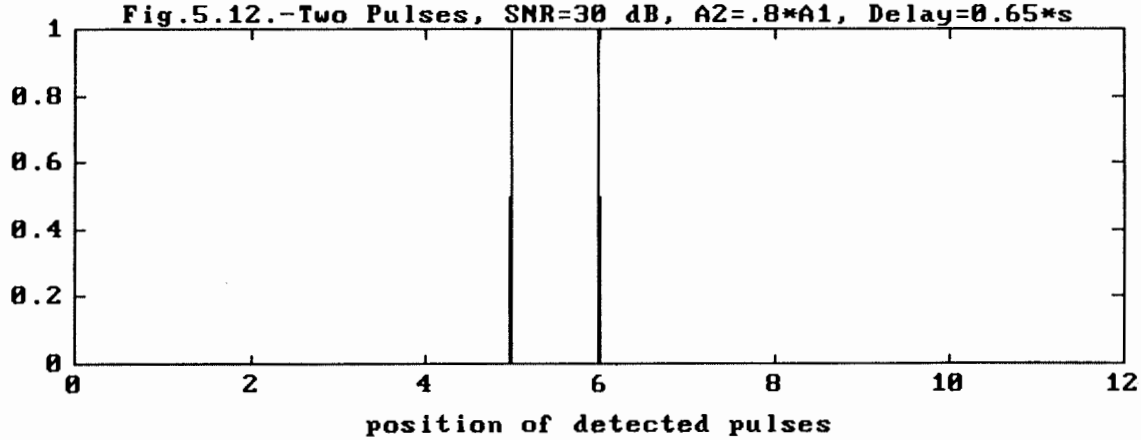


Fig.5.12.-Two Pulses, SNR=30 dB, A2=.8*A1, Delay=0.65*s



Relative Amplitude:

In this test, we used two pulses which are synchronized to the positions of their basis Gaussian pulses counterparts. The two pulses are separated by a fixed delay of one standard deviation. Figure 5.5. illustrates the case of high SNR and equal amplitude. The network has no problem detecting both signals. In Figure 5.6., the SNR was dropped to 0 dB (equal noise and signal power). Again, the detection was reliable, which confirms the high tolerance of Hopfield network to noise. Reducing the SNR below 0 dB resulted in unreliable performance.

Next, the amplitude of one pulse was reduced gradually until the network failed to produce the correct output. The limit of amplitude ratio was about 0.6. Figure 5.7. illustrates this limit case at high SNR. This level of amplitude resolution can still be achieved at lower SNR down to about 10 dB. It should be noted, however, that the test parameters interact with one another. For example, when the amplitude of one pulse is reduced, the detection became more sensitive to the noise level. We were able to establish that two pulses of equal standard deviation separated by a standard deviation and differ in amplitude by 0.6 can be detected when the SNR is 10 dB and higher. This case is illustrated in Figure 5.8. When the two pulses have the same amplitude, the minimum detectable SNR was 0 dB as was illustrated earlier by Figure 5.7.

Minimum Delay:

Since Hopfield technique relies on matching the input signal to a set of pre-defined Gaussian basis functions, the delay resolution cannot be better than half the distance between two consecutive basis functions (i.e. half the standard deviation). We have tried to store Gaussian basis functions separated by less than the standard deviation, but that did not work since the cross talk between the pulses was very high. Therefore, the best time resolution lies somewhere between 1 and 0.5 σ.

Figure 5.9. shows a case where the time resolution is less than the standard deviation. The SNR is high (30 dB), and the two pulses are equal in magnitude. The time separation reported in this figure is 0.88 s (note that in this section we use s and σ inter-changeably). Next, the delay between the two pulses was reduced in steps until the network failed to identify the

correct combination. This has established 0.65 s as a limit on the time resolution as shown in Figure 5.10.

It should be noted that when a single pulse is located mid-way between two basis pulses, the network incorrectly indicates the presence of the two basis pulses. To test the validity of the time resolution as reported in Figure 5.10, we repeated the test with only one pulse shifted by 35% from its corresponding basis pulse. The network correctly identified the output as that of a single pulse as shown in Figure 5.11.

Finally, we tested the combination of a minimum delay and an amplitude difference. The limit is illustrated by Figure 5.12, which is a minimum delay of 0.65 s at an amplitude ratio of 0.8

5.7 Conclusion

This simulation study demonstrates the fundamental advantage of neural networks which is the capability of achieving a time resolution beyond the limit of conventional techniques. The derivatives and correlation methods can not resolve two pulses separated by less than a standard deviation, σ, while Hopfield network can resolve pulses separated by 0.65σ. Another advantage of Hopfield network is its tolerance to a high level of noise. In the case of a single pulse, SNR as low as -3dB can be tolerated. In the presence of two pulses, the limit on the SNR ranges between 0 and 10 dB depending on other factors such as the amplitude ratio and the pulses separation.

In spite of these two great advantages, the neural network solution is not as obvious as it may appear in this study. For instance, Hopfield solution depends on a prior knowledge of the pulses standard deviation. This is essential since the technique relies on matching the input signal to a set of known basis function. Another disadvantage is the tendency of Hopfield network to produce spurious response when the pulses are separated by a large delay. In fact, the network has failed to recognize some cases which were easily identified by the derivative or correlation method.

This suggests that The neural network be incorporated in a more elaborate detection system. One that combines the advantage of the ultra high resolution of Hopfield network and the reliability of the derivative method. The following system is suggested:

- A pre-processing stage, where the IMS signal is filtered and detected by a derivative method. If the solution is evident, we may stop at this point, otherwise, we move to the second stage.

- The second stage is used when the derivative method fails to resolve two closely pulses (i.e. two pulses are suspected but the derivative method produces only one output). In this second stage, the parameters of the suspected compounds are used to calculate the biases and connection weights of a Hopfield network. In this stage, it may be necessary to change the standard deviation in steps and re-calculated the connection weights.

- A post processing stage, where the output of Hopfield network in response to several standard deviations in analyzed to confirm or deny the presence of two pulses.

7.0 CONCLUSIONS

IMS files with RDX quantities ranging from 0.2 down to 0.01 nanogram injected into the IMS were analyzed. It was revealed that RDX IMS files could be approximated by Gaussian peaks with a standard deviation of about 0.17 to 0.18 msec, and white noise, bandlimited to about 2.2 kHz. Other files representing leather contaminated by various amounts of RDX were also analyzed. Several peak detecting algorithms, such as derivative methods, cross-correlation methods, and neural network-based methods were evaluated.

The results indicate that using second derivative algorithms, peak detection was achieved down to 0.1 nanogram of RDX injection into the IMS. Lower quantities, for example 0.05 nanogram, introduced a high probability of false detection. Derivative methods were able to resolve leather peaks with RDX contamination. The selectivity of derivative methods was also determined. Using equal Gaussian peaks with a standard deviation of 0.17 msec each, peak separations down to 0.20 msec were resolved.

The work establishes the lower detection limit for the cross-correlation to be at 0.01 nanogram quantities. It reveals the possibility of combining a number of detection algorithms in order to improve both the detection limit and the resolution in IMS. A typical configuration would therefore include a number of detectors operating simultaneously, followed by a control system which monitors the behaviour of each detector, generates global parameters, and makes the final decision as to the existing peaks in a given mixture. A more specific configuration is to use cross-correlation methods in order to set a time window within which higher order derivative methods are activated.

A study of the applicability of using neural networks to the peak detection problem in IMS was carried. This simulation study demonstrates the fundamental advantage of neural networks which is the capability of achieving a time resolution beyond the limit of conventional techniques. The derivatives and correlation methods can not resolve two pulses separated by less than a standard deviation, σ, while Hopfield network can resolve pulses separated by 0.65σ. Another advantage of Hopfield network is its tolerance to a high level of noise. In the case of a single pulse, SNR as low as -3dB can be tolerated. In the presence of two pulses, the limit on the SNR ranges between 0 and 10 dB depending on other factors such as the amplitude ratio and the pulses separation.

In spite of these two great advantages, the neural network solution is not as obvious as it may appear in this study. For instance, Hopfield solution depends on a prior knowledge of the pulses standard deviation. This is essential since the technique relies on matching the input signal to a set of known basis function. Another disadvantage is the tendency of Hopfield network to produce spurious response when the pulses are separated by a large delay. In fact, the network has failed to recognize some cases which were easily identified by the derivative or correlation method.

It is therefore suggested that a combination of several peak detection algorithms be used simultaneously. Cross correlation methods, for example, could predict the existence of a potential peak. Derivative methods and neural network methods could then be used in order to improve the selectivity by estimating whether it is indeed a single peak or whether there are several superimposed peaks. This way, it is possible to maintain the excellent sensitivity and detection limit performance of cross correlation methods while achieving a selectivity performance similar to derivative and neural network methods. The proposed system could also offer a better detection confidence and robustness, with a computational complexity within the limits of real-time implementation using present technology.

8.0 ACKNOWLEDGEMENTS

The authors are grateful for the financial support provided by the Canadian Police Research Centre of the National Research Council of Canada (NRC), and wish to thank Tony Mungham from the NRC, and the students Mark Pichora, Matthew Yuen, and Nicodem Tilya from Carleton University for their technical assistance.

9.0 REFERENCES

1] A.H. Lawrence, R.A. Goubran, and H.M. Hafez, "Signal Improvement in Ion Mobility Spectrometry," Proceedings of the International Society of Optical Engineering Symposium, on Applications of Signal and Image Processing in Explosive Detection Systems, Boston, MA, Nov. 1992.

2] A.H. Lawrence, "Characterization of Benzodiazepine Drugs by Ion Mobility Spectrometry," *Analytical Chemistry*, vol. 61, no. 4, pp. 343-349, Feb. 1989.

3] A. Prini, A.H. Lawrence, and S. Laframboise, "Compact Digital Signal Average Ion Mobility Spectrometer (IMS) Signals," J. Phys. E Sci. Instrum., vol. 20, pp. 1422-1424, 1987.

4] M.L. Glasser, "Peak Shape Analysis for Ion Mobility Spectroscopy," Journal of Applied Physics, vol. 63, no. 10, pp. 4823-4831, May 1988.

5] T.C. O'Haver and T. Begley, "Signal-To-Noise Ratio in Higher Order Derivative Spectrometry," *Analytical Chemistry,* vol. 53, no. 12, pp. 1876-1878, Oct. 1981.

6] R.A. Caruana, R.B. Searle, T. Heller, and S.I. Shupack, "Fast Algorithm for the Resolution of Spectra," Analytical Chemistry, vol. 58, no. 6, pp. 1162-1167, May 1986.

7] F. Janssens, J.P Francois, "Evaluation of Three Zero-Area Digital Filters for Peak Recognotion and Interference Detection in Automated Spectral Data Analysis," *Analytical Chemistry,* vol. 63, no. 4, pp. 320-331, Feb. 1991.

8] R.A. Goubran, and A.H. Lawrence, "Experimental Signal Analysis in Ion Mobility Spectrometry," *International Journal of Mass and Ion Processes,* vol. 104, pp. 163-178, Jan. 1991.

9] R.A. Goubran, "IMS Signal Processing", Final Report, NRC contract Number 988-4743 R, March 1989.

10] R.A. Goubran, "Peak Detection of IMS Signals", Final Report, NRC contract Number 990-1266/1045, March 1991.

11] R.A. Caruana, R.B. Searle, and S.I. Shupack, "Additional Capabilities of a Fast Algorithm for the Resolution of Spectra," *Analytical Chemistry*, vol. 60, no. 18, pp. 1896-1900, Sep. 1988.

12] J.E. Roehi, "Microprocessor Controlled Chemical Detection and Alarm System Based in Ion Mobility Spectrometry", *IEEE Transactions on Industrial Electronics*, Vol. 32, No. 2, pp. 108-113, May 1985.

13] A. Prini, A.H. Lawrence, and S. Laframboise, "Compact Digital Signal Averager for Ion Mobility Spectrometer Signals", *J. Phys. E: Sci. Instrum.*, pp. 1422-1424, 1987.

14] A.V. Oppenheim, and R.W Shaffer, "Discrete-Time Signal Processing", Prentice Hall, 1989.

15] B. Widrow, and S.T. Stern, "Adaptie Signal Processing", Prentice Hall, 1985.

16] S. Haykin, Adaptice Filter Theory, Prentice Hall, 1987.

17] J.A.Freeman and D.M.Skapura, "Neural Networks: Algorithms, Applications and Programming Techniques", Addison Wesley 1991

18] J.J.Hopfield, "Neurons with Graded Response Have Collective Computation Properties Like Those of Two State Neurons", *Proceedings of the National Academy of Sciences 81*, May 1984, pp. 3088-92.

19] D.W.Tank and J.J.Hopfield, "Simple Neural Optimization Networks: an A/D Converters, Signal Decision Circuit and Linear Programming Circuit", *IEEE Trans. on Circuits and Systems*, 33(5), May 1986, pp. 533-41

20] Y.S.Abu-Mostafa and J.St.Jacques, "Information Capacity of the Hopfield Model", *IEEE Trans. on Information Theory*, 31(4), July 1985, pp.461-64