INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Intelligent Autonomous Inductively Coupled Plasma Instrumental Operation

by

Douglas P. Webb

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August 1996

Department of Chemistry
McGill University
Montreal, Quebec
CANADA

ABSTRACT

The development of a framework for the automated analysis of inductively couple plasma atomic emission spectroscopy is presented. Some of the research that lead to current state of this framework is presented. A small expert system that uses information about the current sample to generate a line search strategy which minimizes the number of emission lines which need to be measured, and avoids spectral overlaps when possible. A program is presented that evaluates the minimum number of spectral windows required to perform elemental analysis by ICP-AES, given a certain spectral window width. A method with the potential for rapidly ascertaining the physical properties of the sample matrix is presented. This system has the potential to help reduce sample introduction related system failures. Finally, three optimization algorithms are compared in their ability to optimize ICP-AES performance, from this an optimization module was developed for inclusion in the automated analysis framework.

RÉSUMÉ

Le développement d'une structure pour l'analyse automatisée par spectroscopie d'émission atomique à plasma inductif est présentée. Certains aspects de la recherche ayant mené à l'état actuel de cette structure sont traités. Un système expert simple utilise des informations relevant de l'échantillon courant pour générer une stratégie de recherche de raies d'émission qui minimise le nombre de raies mesurées et évite le chevauchement spectral si possible. Un programme est proposé pour l'évaluation du nombre minimal de fenêtres spectrales requises pour effectuer une analyse élémentaire par spectroscopie d'émission atomique à plasma inductif, selon une largeur de fenêtre donnée. Une méthode offrant la possibilité d'établir rapidement les propriétés physiques de la matrice d'échantillon est décrite. Cette méthode pourrait potentiellement contribuer à réduire l'occurence de pannes liées à l'introduction de l'échantillon. Enfin, trois algorithmes d'optimisation sont comparés quant à leur abilité à optimiser la performance d'analyse par spectroscopie d'émission atomique à plasma inductif; un module d'optimisation développé à la suite des résultats obtenus est incorporé à la structure d'analyse automatisée.

CONTRIBUTIONS TO ORIGINAL KNOWLEDGE

1) An initial framework for undertaking automated analysis using an ICP-AES has been developed.
2) An expert system for the selection of emission lines for use with a scanning inductively coupled plasma spectrometer was developed.
3) A study to determine the minimum number of spectral windows of various sizes required to analyze a variety of multi-element samples by emission spectroscopy was undertaken.
4) A study to determine physical sample characteristics by acoustic signature was undertaken.
5) A comparison of three popular optimization techniques and their performance on various inductively coupled plasma atomic emission response hypersurfaces is presented.

ACKNOWLEDGMENTS

I would like to express my thanks to Professor Eric D. Salin for his keen insight and support throughout this project. Our frequent discussions never failed to turn over a new stone underneath which knowledge was hidden.

Two other individuals have made substantial contributions to this project and deserve both credit and my thanks. Jan Hamier spent six months developing rules for our calibration methodology selection expert system, and helped greatly in defining the overall structure of the spectral analysis control loop as it now stands. Wayne Branagh wrote software that encapsulated the DESKview interface used by our software and extended my instrument interface code to a level where it was more useful and extensible. Furthermore, he designed the first draft of the cork board object that is used to pass data among the various portions of our software. Our discussions constantly made me rethink the structure of my work and the work is better for it.

I would like to thank everyone else in Prof. Salin's group for editing assistance and consistently thought provoking discussion.

I would also like to thank the following individuals:

Dr. J. F. Power who contributed greatly to my acoustic experiments.

Dr. Laurent Blain for answering my naive questions about the ICP.

Financial support in the form of Post-Graduate Awards from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Le Fonds pour la Formation de Chercheurs et l'Aide a la Recherche (FCAR) are greatly appreciated.

Perhaps most importantly, I'd like to thank my wife who supported me throughout the many trials that led to this work being finished.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 2

SELECTION OF SPECTRAL WINDOWS FOR PLASMA ATOMIC EMISSION SPECTROMETRY

CHAPTER 3

A LINE SELECTION EXPERT SYSTEM FOR CONTROL OF SCANNING INDUCTIVELY COUPLED PLASMA ATOMIC EMISSION SPECTROMETERS

CHAPTER 4

ICP SAMPLE MATRIX IDENTIFICATION BY ACOUSTIC SIGNATURE

CHAPTER 5

THE AUTONOMOUS INSTRUMENT: CONCEPT TO PROTOTYPE

CHAPTER 6

THE AUTONOMOUS INSTRUMENT: A DESIGN

CHAPTER 7

OPTIMIZATION OF INDUCTIVELY COUPLED PLASMA ATOMIC EMISSION SPECTROMETERS

TABLE 2. LIST OF FIGURES

CHAPTER 1

CHAPTER 2

SELECTION OF SPECTRAL WINDOWS FOR PLASMA ATOMIC EMISSION SPECTROMETRY

CHAPTER 3

A LINE SELECTION EXPERT SYSTEM FOR CONTROL OF SCANNING INDUCTIVELY COUPLED PLASMA ATOMIC EMISSION SPECTROMETERS

CHAPTER 4

ICP SAMPLE MATRIX IDENTIFICATION BY ACOUSTIC SIGNATURE

CHAPTER 5

THE AUTONOMOUS INSTRUMENT: CONCEPT TO PROTOTYPE

CHAPTER 6

THE AUTONOMOUS INSTRUMENT: A DESIGN

CHAPTER 7

OPTIMIZATION OF INDUCTIVELY COUPLED PLASMA ATOMIC EMISSION SPECTROMETERS

CHAPTER 1 - INTRODUCTION

North America is likely to experience a shortage of technically competent personnel in the near future (1986 Government of Canada Manpower Projections for the 1990's). If this shortage is coupled with the increasing environmental consciousness and demand for quality control in government and industry alike, there will be an increased demand for analytical services. The number of individuals qualified to operate today's analytical instrumentation will not keep pace with this new demand. One way to try and bridge this gap is to automate as many steps involved in these analytical analyses as is practical. However, simple automation, physically removing humans from the loop, will not be sufficient to produce competitive analytical laboratories. Automation without intelligent guidance will lead to systems which will be limited in their ability to operate autonomously and still produce high quality analytical results. Automation must be coupled with an intelligent decision making capability. Intelligent automation is comprised of several components: (1) the accumulation of information, (2) the ability to make decisions based on this information, and (3) the ability to take actions based on these decisions.

It is my belief that the most significant advances in analytical instrumentation in the next 5-10 years, as far as the end user is concerned, will come in the realm of improvements in instrumental control software. Among the most significant advances will be the integration of intelligent on-line data analysis and intelligent automation into this instrumentation. The benefits to the end user will be:

1. Reduced training costs. The instrument will be "smarter", therefore the requirement for highly skilled manpower will be reduced. Furthermore, the instrument will be easier to use, and the instrumental control software could be written so as to educate and train the instrument operator. The instrument will be able to automatically handle all but the most difficult analyses, freeing an expert to supervise several instruments or perform other functions. Because routine analyses can be performed without the need for

operator intervention, the expert need only intervene when a non-routine condition arises. This should make the working environment more interesting.

2. Less instrument "down time". The instrument will monitor itself, identify conditions that are likely to cause instrument failure and avoid these conditions where possible

3. Better analyses. Modern instrumentation produces data at a phenomenal rate, often as high as 1000 points/minute. No human operator, expert or not, can keep up with this data flow. The instrument will perform analyses in a consistent manner and will continually monitor the results of the analyses it performs. Using a rulebase derived from training sets or the input of experienced operators, deviation from expected behaviour will be noticed immediately, and remedial actions can be taken in 'real-time'.

4. Higher sample throughput. Autosamplers and computers never get tired, therefore instrumental working hours will be extended.

It was decided that Inductively Coupled Plasma Atomic Emission Spectrometry (ICP-AES) would be used as the target system in our development of an intelligent instrument control system for the following reasons:

1. Operating conditions are variable and dramatically affect the results of instrumental analyses. These instrumental operating conditions are now under computer control on many commercial instruments.

2. Achieving consistently good analytical results with an ICP requires operator expertise.

3. There are ample sources of information about the sample:

   a) Prior knowledge: usually operator supplied information.

   b) Acquired knowledge: information generated by the instrument itself.

4. Modern ICP-AES systems generate data at a rate that is impossible for a human operator to monitor in real time. Failure to monitor a sample in real time may result in errors which will require that the sample be rerun - if it has not been completely consumed.

1.2 EMISSION SPECTROMETRY

1.2.1 History

Analytical chemistry is the science of identifying and quantifying different materials based on their physical properties. Colour is one of the most important of these physical properties. The human eye can perceive colour in the wavelength region 400 to 700 nm. As early as 1758, Margraf observed that when salts of sodium and potassium were introduced into a flame, characteristic colors were observed. In 1820, Faunhofer used a grating made of closely spaced thin wires to generate a spectrum from white light. In the early 1800's, Herschel used a prism to determine that the spectrum emitted by a flame contained bright lines and dark gaps between these lines. Herschel and Ritter discovered that light was not limited to the range that the human eye can distinguish. They discovered that 'invisible' light was being emitted both below the red end of the spectrum, and above the violet end. In 1840, Draper photographed the solar spectrum and thereby confirmed the existence of emission lines extending well into the ultraviolet. Bunsen and Kirchhoff built the first 'spectroscope' in 1859. This instrument allowed more accurate observations of the spectral lines emitted by flames. They discovered that each element emitted light at a set of characteristic wavelengths. By the late 1870's Rowland had developed concave metal and glass gratings that allowed high dispersion of incident radiation.

1.2.2 Detection Systems

Since Draper first captured the solar spectrum photographically, techniques advanced to the point that an entire visible spectrum could be captured on a single photographic plate; however developing photographic plates was awkward and quantitative reproducibility was poor. In 1935, the first single-stage photomultiplier tube was reported by Iams and Salzberg[1]. This device introduced the concept of a detector with gain (which in this case was about a factor of eight). The photomultiplier tube (PMT) as we know it today was first commercially available in 1941[2]. These detectors had a gain of about 10^7 and peak quantum efficiency of about 12% at 400 nm. A constraint of photomultiplier detectors is that when used in conjunction with a

dispersive systems like those developed by Roland they can only be used to examine a narrow wavelength region at a time. Several solutions to this problem appeared: 1) Instruments with many PMTs spread across the Roland circle allowed simultaneous acquisition of up to about 60 narrow wavelength regions; 2) Instruments containing two or three PMTs and a mechanism for rapidly changing the orientation of the grating allow examination of any wavelength albeit sequentially. In the 1970's solid state, multichannel devices started to appear. These devices allowed the simultaneous capture of spectral information at many wavelengths. An Echelle spectrometer using a solid state detector to simultaneously capture an entire spectrum between 150 and 800 nm was prototyped as early as 1983[3]. These advances in solid-state detector technology have two effects: 1) Large spectral regions can now be acquired simultaneously so spectral interferences can be identified and corrected for. 2) The amount of information acquired cannot be evaluated and processed in real time by a human, no matter how skilled.

1.2.3 Inductively Coupled Plasma Atomic Emission Spectrometry

The advancement of atomic spectral analysis has depended on the development of emission sources hot enough to atomize and excite samples of analytical interest. Flames were the first atomic emission sources. Later arcs and sparks appeared. These methods all allowed the quantitative determination of low levels of analyte in a variety of sample matrices. None of these sources was markedly superior to the others in the analysis of all types of samples. When selecting an emission source, a selection was made depending on the relative importance of detection limit, dynamic range and freedom from chemical interferences.

In the 1940's Babat succeeded in generating the first sustaining induction heated plasma at atmospheric pressure[4,5]. The inductively coupled plasma (ICP) was first applied to analytical studies independently by Greenberg[6] and Fassel[7] in 1962. By 1969, atomic analysis using the ICP had advanced to the point where the detection limits attained were two orders of magnitude better than had previously been achieved using other techniques[8].

ICP-AES is now a well established technique for elemental analysis in a variety of applications[9,10,11,12]. The ICP is a popular atomic

emission source, because it exhibits many characteristics of the ideal source: little or no background emission, sufficient energy to excite all elements, and, if an argon plasma is used, a chemically inert environment. For this reason, ICP-AES systems are now available from a wide variety of instrument manufacturers, and ICP-AES is the instrumentation of choice for many types of elemental analysis.

1.3 ARTIFICIAL INTELLIGENCE

1.3.1 Definition:

Artificial Intelligence might be categorized as the study and design of computer systems that exhibit behaviour that is associated with intelligence in humans. Some fields of study that currently fall under the broad heading of AI are:

- Natural language understanding
- Robotics and motion planning
- Learning and inductive inference
- Vision
- Theorem proving
- Game playing
- Search
- Knowledge representation
- Expert systems

1.3.2 What is an AI Technique?

Artificial intelligence problems are very diverse. Many appear to be unrelated to each other except that they are very complex and difficult problems to solve. However, AI problems have one unifying attribute: they require *knowledge* to solve. Knowledge has several unfortunate qualities:[13]

- It is voluminous
- It is often hard to characterize accurately
- It is constantly changing

An AI technique might be defined as a computational method that exploits knowledge to solve an otherwise intractable problem.

1.3.3 History:

1.3.3.1 Beginnings:

It is generally agreed that the first practical implementation in the field of artificial intelligence (AI) was in 1947, when Arthur Samuel of IBM created a program which not only played checkers, but actually learned from its experiences, and improved its level of play.[14]

In 1950 Allan Turing presented his definition for machine intelligence.[15] This criteria is known as the Turing test and is still used as a benchmark against which intelligent machines are measured. In summary, the Turing test states that when a machine can carry on a conversation, and thereby convince a human that the machine is human, then the machine can be said to exhibit intelligence.

In 1956 many of the early luminaries in the budding field of AI attended a conference at Dartmouth. The term 'Artificial Intelligence' first appeared in print at this conference (the term is attributed to John McCarthy of Stanford University). The premise of the conference was that "every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it." Aside from founding the field of AI, the conference's primary contribution was the discovery that AI is a very difficult field of research.

1.3.3.2 New Algorithms:

The early years of AI research were largely focused on game-playing and puzzle solving. Many fundamental search techniques were pioneered in this period. The *state space search*[16] appeared during this period. This technique assumes that the problem can be formulated in terms of: 1) a starting state, 2) a termination test, which will allow the detection of problem resolution, and 3) a set of operations that can be applied to the current state of the problem. Unfortunately, for many problems the state space is very large, and the number of possible states expands exponentially as problem resolution progresses. If a brute force search technique like *breadth-first* or *depth-first*[16] search is used combinatorial explosion can quickly outstrip the computational abilities of even the fastest computers.

Given that an exhaustive search is impractical for large search spaces, techniques for directing the search were required. In the late

1960's and early 1970's, the most prominent techniques developed for searching involved using domain specific knowledge to direct the search across the state space. Primarily these techniques relied on "rules of thumb" (*heuristics*) and were not guaranteed to find a solution as an exhaustive search algorithm would. Examples of such heuristics are *hill climbing*[17] and *best-first*[16] search mechanisms. Both depend on an evaluation function to rate the positions in the state space. Based on these evaluations the next search position is determined. This search mechanism is unlike the blind, brute force techniques that merely progress from one state to the next.

Research directed at the problem of knowledge representation was complimentary to the development of knowledge directed search techniques. Knowledge representation comprises two parts: Data structures for storing information and procedures that allow 'intelligent' manipulation of these data structures to perform inferences. Some of the most widely used knowledge representation techniques were developed during the 1960's and early 1970's, including *production rules*[18], *associative nets*[19] and *frames*[20].

1.3.3.3 A Change in the Direction of Knowledge Programming:

From the mid 1960's through the early 1970's, there were many attempts to develop psychological models of how humans solve problems. Much emphasis in the AI community was focused on developing programs that 'understood' the domains in which they operated, and in modeling the way in which humans solved simple problems like puzzles. Unfortunately it is very difficult to determine directly if a computer and a human are solving problems the same way, even if the same result is obtained. Understanding requires the ability to 1) represent knowledge about a domain and reason using that knowledge, 2) recognize analogies or equivalencies between different representations of the same situation, 3) the ability to learn in a non-rote fashion that requires integration of new information with existing information[21]. It is arguable that a single program has yet to be developed which does all of these things well.

The 1980's and 1990's have been characterized by a keen sense that the AI effort did not solve the general understanding and cognitive modeling problems in the late 60's and the 1970's despite an enormous

effort and much hoopla. The search for general problem solving methodologies has been all but abandoned, and the emphasis on 'understanding' is not as all-pervasive as it once was. Many of today's programs explicitly represent domain specific knowledge which is accessed by the program directly rather than using sophisticated general purpose inference mechanisms. In recent years several programs have appeared whose performance of non-trivial tasks exceeds that of all but a few humans.

1.3.4 Expert Systems

An expert system might be defined as a program capable of representing and reasoning about some (usually narrow) knowledge rich domain, with a focus on solving problems and giving advice. Expert systems development has been one of the most successful branches of the artificial intelligence effort, having produced several programs which work at a high level of performance in non-trivial problem domains.

1.3.4.1 Historical Development

In the 1960's Allen Newell and Herbert Simon developed a mechanism for representing knowledge as a series of situation-action rules called productions. [18] These rules have the form:

IF situation is A THEN take action B

This work was generalized in a program called P, which was a vehicle for exploring different ways of representing heuristics as a set of explicit rules separate from the rule processing mechanism.[22] Because production rules make it easier to understand and modify systems containing large amounts of knowledge, they have been used in several large applications like MYCIN[23] which is an expert system developed at Stanford to diagnose bacterial infections, PROSPECTOR[24] an expert system developed to aid geologists searching for ore deposits, and the chemical analysis expert system DENDRAL[25].

In 1965, Feigenbaum *et al*[25] began work on DENDRAL, one of the first expert systems. DENDRAL's task was to determine the molecular structure of an unknown organic compound by interpreting both mass spectral and nuclear magnetic resonance data. While the older problem solving systems used heuristic search, DENDRAL was one of the first programs to use an explicit representation of domain specific knowledge

to help reduce the scope of its search space to a manageable size. DENDRAL is one of the success stories in expert systems (and by extension artificial intelligence). DENDRAL has reached the state of a production prototype and is being used at Stanford University to assist in the determination of molecular structures.

1.3.4.2 What are Expert Systems?

Detailed information about what expert systems are and how they are built is widely available.[21,26] An expert system usually consists of three main components: knowledge base; inference engine; and user interface, as illustrated in Figure 1.

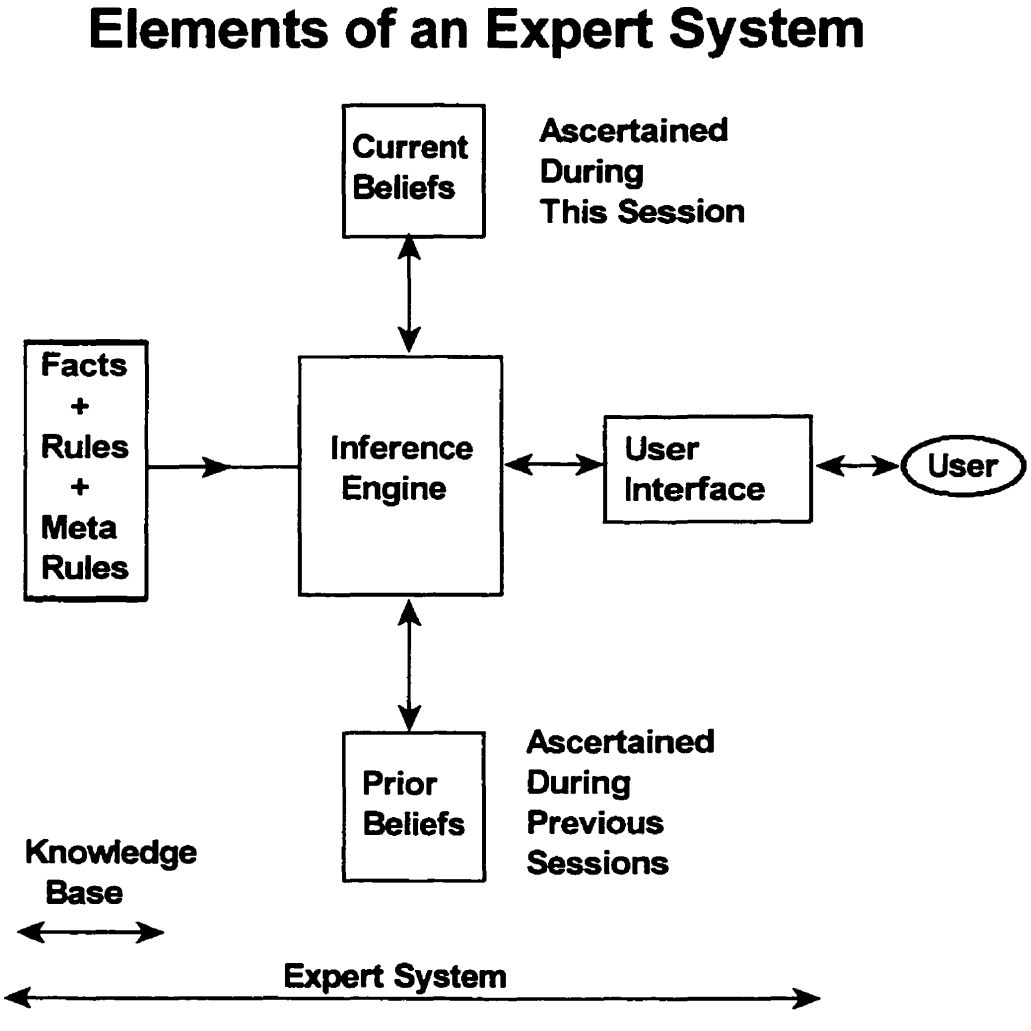


Figure 1 - The composition of an Expert System

1) **Knowledge base**: A knowledge base contains information about the domain (problem area) in which the expert system will be functioning. The performance of the expert system depends heavily on the completeness and accuracy of the knowledge base. Knowledge must be encoded in such a way that it can be efficiently stored on computer, manipulated by the inference engine (described below), and ideally, understood by the expert. Several formalisms for encoding knowledge have been proposed, some of the more popular formalisms include *production rules*[18], *associative nets*[19], *frames*[20], *structured objects*[27] and the *'well-formed formulas'* of predicate calculus[28]. Most expert systems in use today use one or more of these knowledge representation schemes.

2) **Inference Engine:** An inference engine is the portion of the program which manipulates the facts and rules which are stored in the knowledge base in an attempt to reach conclusions (resolve problems). The structure of the inference engine is highly dependent on the structure of the knowledge base. For example, if the knowledge is stored in the 'well-formed formulas' of predicate calculus, then the inference engine might consist of a theorem prover. There are two basic ways that an inference engine can function: it can start with collections of facts which are known and attempt to determine what conclusions can be reached given those facts (this is called data-driven search and uses a forward chaining search mechanism[29].) Alternatively, an inference engine can start with one or more hypotheses and attempt to determine which if any is correct (this is called goal-driven search and uses a backward chaining search mechanism[29].)

3) **(User) Interface:** The interface to the expert system can usually be divided into two parts: There must be a mechanism by which the expert can enter facts and encode rules, and then test the knowledge base. The interface must make it possible (and ideally easy) to query the knowledge base. Included in the user interface are functions to prompt for or acquire information, explain why information is needed, and explain how a conclusion was reached. In some systems, the user interface to the expert system may be part of the development system, where the knowledge base is built and tested, but may be absent from the final program.

The process of building the expert system is often called knowledge engineering and is considered 'applied artificial intelligence'[30].

1.3.4.3 Expert System Development Tools

There are several types of tools that can be used to build expert systems, or programs with some expert system capabilities:

1) Traditional procedural languages
2) Languages developed specifically for symbolic processing
3) Expert system shells

Procedural Languages

Traditional procedural languages (specifically C, but also Ada and Pascal) offer great flexibility in the development of software tools, however they have no built-in capabilities which facilitate the development of programs that depend heavily on symbolic computation as do expert systems and many other AI applications. Programs written in procedural languages do, however, tend to require fewer computational resources than similar systems written in the so-called AI languages. Expert systems are often prototyped in declarative languages like LISP or PROLOG, then translated into C for the final production version in an attempt to increase execution speed. The widespread use of these languages, C in particular, means that toolkits (collections of routines that can be incorporated directly into programs under development) that can perform almost any type of computation can be purchased.

Declarative Languages:

In 1958 the first mainstream AI programming language, LISP, appeared. LISP was introduced by John McCarthy at Stanford, and is still by far the most popular AI language in North America. In North America, 95% of the best known AI applications are written in LISP[29]. LISP was designed with powerful symbol manipulation capabilities and a high degree of flexibility which have made it very popular with the North American AI community.

In the early 1970's the PROLOG programming language was developed. It was perhaps the most important AI development tool since the development of LISP.. PROLOG was developed by Robert Kowalski and Maarten van Emden at Edinburgh, and first implemented by Alain Colmerauer at Marseilles. PROLOG is important because it is the first

language that "encourages the programmer to describe situations and problems, not the detailed means by which the problems are to be solved". This allows "the programmer to be more concerned with knowledge and less concerned with algorithms that exploit the knowledge"[31].

LISP and PROLOG both have built in symbol manipulation capabilities which make then better suited to the development of many types of knowledge based systems than their procedural brethren. Furthermore, PROLOG has a backward chaining inference mechanism and a very efficient pattern matching strategy built in. These built-in capabilities further speed the development of programs, like expert systems, that depend on them. The drawback to developing in these languages is the dearth of available third party software tools as compared to their more popular procedural counterparts. Applications developed in these languages usually suffer from a need for greater computational resources, and, in the case of PROLOG, reduced portability of the final code as compared to better standardized languages like C.

Expert System Shells:

Expert system shells are programs that speed the development and implementation of expert systems by providing many of the tools that are needed. An expert system shell is an expert system without any embedded knowledge. Expert system shells are much like spreadsheets for knowledge[32].

In 1980 EMYCIN, one of the first expert system shells, appeared[33]. EMYCIN provided a rule format which was concise and easy to read, a system design tool for adding and editing rules, a built-in backward chaining mechanism, an interface for consultation with the end user, and TEIRESIAS a knowledge editor which was designed to help maintain large databases.[34]

EMYCIN had many capabilities which are now standard on most of today's better expert system shells:

A mechanism for entering rules - with syntax checking
Ambiguity resolution
A rule interpreter

Debugging tools

An explanation facility

Expert system shells/tools vary widely in flexibility and cost. The most powerful tools like Inference Corp's ART, and Intellicorp's KEE, cost more than $50,000 for a development system, but they provide the developer with great flexibility and many built-in tools. The plethora of lower cost microcomputer based expert system development tools tend to be less flexible but more than sufficient for many expert system applications.

1.3.4.4 Expert Systems in Chemistry

Wade *et al*[35] and deMonchy *et al*[36] provide a good overview of the technology and description of the jargon that is associated with expert systems and describe some potential applications in the chemistry laboratory. The field of chemistry has, from the very beginning, embraced expert systems, and expert system technology is currently being applied in many different branches of chemistry[26,37]. These expert systems: use the output of triple quadrupole mass spectrometers (TQMS), to tune the TQMS[38]; assist in the design of complex organic synthesis[39]; use x-ray diffraction data to infer atomic structure[40]; determine metal concentrations using atomic absorption spectrometry output[41], and much more.

1.4 INSTRUMENTAL OPTIMIZATION

Systemic optimization is an area that has been developing rapidly in the last few years as can be seen in recent chemometrics reviews [42,43]. There are several fundamental types of optimization.

1.4.1 Optimization Techniques

Exhaustive Search (Grid Search). This technique involves performing evenly spaced experiments which characterize the entire surface to be optimized as illustrated in figure 2. The precision of the peak finding depends on the spacing experiments. For example, to characterize a 2-dimensional surface to the 5% precision level would require 100 experiments. This technique guarantees that the global optimum is found, but for many-dimensional spaces where a physical

experiment must be performed to generate a response, this technique is impractical, due to the large number of experiments required.

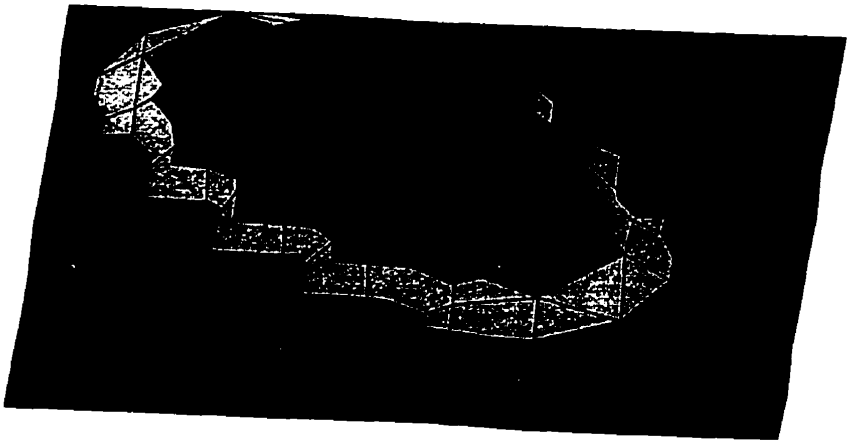


Figure 2 - Exhaustive search

Axial Iteration (Iterative Univariate Optimization). This technique is a classical one variable at a time technique as illustrated in figure 3. The technique keeps all but one variable fixed, while changing the remaining variable until the optimum response value for the changing variable is found. This procedure is repeated for each variable in succession. One iteration of this technique is complete when each variable has been altered once. This technique is known to have difficulty finding the optimal response value when the response surface contains a ridge which isn't parallel to any of the response surface's axes. For many dimensional spaces the number of experiments required to perform even a single iteration can be prohibitive.

Figure 3 - Axial iteration

Model Based Approach. In domains where the factors affecting the response are well understood mathematically, it is sometimes possible to run system simulations that allow rapid optimization without the need to perform physical experimentation, for example x-ray phase analysis[44]. The various techniques that fall under the general heading Monte Carlo optimization [45,46,47] are the preeminent model based optimization techniques.

Surface Fitting. Surface fitting techniques usually depend on interpolation to determine the nature of the response surface near the optimum. This technique usually requires a large number of experiments and some prior knowledge about the response surface being fitted. This method is impractical for spaces with more than two dimensions.[48]

Simplex. In areas of chemistry where the factors affecting the response are not sufficiently well characterized to apply model based techniques, optimization is implemented as one form or another of experiment based hill-climbing methodology such as the simplex first developed by Spendley *et al*[49]. The basic simplex methodology depended on prior knowledge about the response surface to achieve good performance, therefore Nelder and Mead[50] developed a self-adapting version of the simplex which performed better on widely varying response surfaces. This modified simplex was poorly designed to handle conditions where the optimum was near the edge of the response surface, and it exhibited degraded performance in the presence of noise in the experimental responses. Work by Routh *et al*[51] and Betteridge *et al*[52,53] has resulted in improvements that address these problems.

Gradient Methods. For continuous response surfaces, gradient methods calculate first or second-derivatives for each variable to determine the direction of greatest improvement in the response, then all variables are moved in the direction that improves the response. When using first derivative techniques, the speed with which the optimum is found is highly dependent on the selection of the starting points. An important refinement of the first derivative technique is the second derivative technique proposed by Fletcher and Powell[54]. This technique

uses a combination of parabolic extrapolation and second derivatives to locate the response surface optimum in an efficient manner.

1.4.2 Optimization for Inductively Coupled Plasma

Broekaert *et al* have stated that, in general, ICP-AES is difficult to optimize[55]. Attempts have been made to optimize ICP-AES using univariate methods[56], Nelder and Mead style simplexes[57], a super modified simplex[58] and gradient methods[59]. Werner and Friege examined in depth what criteria should be used when optimizing ICP to assure high quality analytical results[60].

1.5 TOWARDS AN AUTONOMOUS ICP ATOMIC EMISSION SPECTROMETER

Analytical spectroscopy in the 1990's and beyond will be characterized by instruments that can produce greater quantities of analytical results per unit time. Specifically, multichannel detection systems on available ICP-AES spectrometers can generate as many as 250 000 data points per sample, which dictates the use of intelligent data interpretation technology. Furthermore, there is a greater demand for reliable and inexpensive analysis than ever before, due to increasing ecological awareness and the increasing requirement for quality control in industry. In addition to these factors there is likely to be a dearth of highly qualified personnel to perform these analyses. The logical corollary to these facts is: these needs can only be met by judiciously applying advanced information processing and automation technology.

As stated previously, ICP-AES is an excellent vehicle for the study and development of this technology because of the existence of commercially available instruments which have the parameters that affect analytical results under computer control. Furthermore, ICP-AES requires expertise in instrumental operation to achieve consistently high quality analytical results.

Goals for the autonomous instrument are:

1) To recognize types of samples similar to those that have been previously analyzed. This could be useful for identifying problem sample types, for which an optimal analysis technique had been previously determined. So, if a new sample is encountered which

appears to be similar to a known sample, the operating conditions and constraints for the known sample could be applied to the analysis of the new sample.

2) To recognize instrumental operating behaviour indicative of a problem such as organic solvents which can 'blow out' the plasma. Using the recognition techniques mentioned above it should be possible to link remedial actions to previously evaluated problem conditions.

3) To optimize instrumental operating conditions to maximize the signal to noise ratio, maximize signal to background ratio, or achieve optimal results based on any other desired figure of merit.

4) Select a good analytical methodology using available information.

5) To identify and prevent conditions such as a blocked nebulizer or sample injection tube, both of which bring analysis to an immediate halt until the offending component can be cleaned or replaced.

1.6 PROPOSED METHODOLOGY

Initially a series of experiments to determine what hardware and software techniques could best be applied to our project were to be performed. Additionally studies were to be made examining what self-diagnostic capabilities are inherent in the ICP. Subsequently, the development of a small working prototype was planned to assist in the analysis of data usage, and data processing requirements. This work would then be used as a basis for the development of an efficient and flexible software architecture which could be used as a framework for the construction of the various low level modules that would comprise the working system.

Subsequent to the development of the prototype, the construction of this framework was proposed. Once the framework was in place the various supporting subsystems (databases, user interface, and a linkage supporting the use of rule-based knowledge programming) were to be developed.

Once the support structures were completed, the development of one major system component was to be undertaken. Consultation with

industry indicated that instrumental optimization is considered both important and difficult. Therefore, it was decided that automatic optimization of the instrumental operating parameters would be studied.

1.7 THESIS OUTLINE

The remainder of this thesis is structured as follows;

Chapter 2 describes our search for ways to maximize the spectral information that could be acquired with either a photodiode array detector based ICP emission instrument, or a photomultiplier tube based instrument with a moveable refractor plate.

Chapter 3 describes an expert system designed to determine the analytical emission line most likely to furnish the best analytical result, even in the presence of potential spectral interferences.

Chapter 4 describes an attempt to use the acoustic signature of a sample undergoing nebulization to identify sample type and thereby avoid sample introduction induced ICP system failure.

Chapter 5 Describes the initial thought process involved in developing a truly autonomous ICP instrument and some of the prototyping work done.

Chapter 6 Describes steps taken to revise and improve the initial design and our work to add calibration method selection to the autonomous instrument.

Chapter 7 describes a study which compares the performance of three popular optimization techniques on both discrete and smooth ICP-AES response surfaces using three different objective functions.

Chapter 8 describes work that still needs to be done to bring this system to a fully functional state.

1.8 CONTRIBUTIONS TO THIS THESIS

Dr. J.F. Power suggested many of the experiments which make up Chapter 4.

Dr. Guy Legere helped write the power spectrum code used to extract frequency information from the raw experimental data in the experiments that became Chapter 4. Dr. Legere's contribution to the data processing source code was less than 10% of the total.

Dr. Wayne Branagh developed a C++ object which encapsulated the functionality of my original C based DESKview interface. He also wrote the original version of the corkboard object which was an invisible but important part of the AJAX project's data handling scheme. He also made additional minor contributions to the user interface. Dr. Branagh's contributions comprised less than 5% of the original DOS-based AJAX source code.

Mr. Jan Hamier contributed the calibration methodology ruleset and the original implementation of an expert system using it (written in 1st Class), as well as helping to ascertain what prior knowledge should be extracted from the operator, and the original pre-calibration methodology. Additionally he wrote the original draft of the paper (about 20% of which appeared in the final revision.) These contributions all appear in Chapter 6. The work was part of his undergraduate thesis.

Dr. E.D. Salin made contributions as follows:

- The seed idea for the work that appeared in chapter 2, 70% of the writing/editorial work for the final paper.
- The seed idea for the work that appeared in chapter 3, 50% contribution to the development of the decision tree that appears in figure 2 of chapter 3 and about 40% writing/editorial work for the final paper.
- Suggestions for one or two experiments that appeared in chapter 4.
- A 50% contribution to the definition of the problems faced in developing the autonomous instrument and methods proposed to overcome them in chapter 5, and about 50% writing/editing work for the final paper.
- A 25% contribution to the revised framework described in chapter 6 and a 25% writing/editorial work for the final paper.
- Minor editorial contributions to chapter 7.

Anything not specifically attributed to one of these five people was done by myself.

[1] H.E. Iams and B. Salzberg, *Proc. IRE*, **23**, pp 55-64, 1935.

[2] V.K. Zworykin and J.A. Rajchman, *Proc. IRE*, **27**, pp 558-566, 1939.

[3] M.B. Denton, H.A. Lewis, and G.R. Sims, Multichannel Image Detectors, Volume 2, Yair Talmi (Ed.), American Chemical Society, Washington D.C., 1983.

[4] G.I. Babat, *Vastn. Elektroprom.*, **2**, p 1, 1942.

[5] G.I. Babat, *J. Inst. Elect. Eng.*, **94**, p 27, 1947.

[6] S. Greenberg; I.L. Jones and C.T. Berry, *Analyst*, **89**, p 713, 1964.

[7] R.H. Wendt and V.A. Fassel, *Anal. Chem.*, **37**, p 920, 1965.

[8]G.W. Dickson, V.A.Fassel, *Anal.Chem.*, **41**, p1021, 1969.

[9] J.D. Ingle and S.R. Crouch, Spectrochemical Analysis, Prentice Hall, New Jersey, 1988.

[10] A. Montaser and D.W. Golightly, Ed.; Inductively Coupled Plasmas in Atomic Spectrometry, VCH, New York, 1987.

[11] P.W.J.M. Boumans, Ed.: Inductively Coupled Plasma Emission Spectroscopy - Part 1: Methodology, Instrumentation and Performance, Chemical Analysis series vol. 90, John Wiley and Sons, Inc, New York, 1987.

[12] P.W.J.M. Boumans, Ed.: Inductively Coupled Plasma Emission Spectroscopy - Part 2: Applications and Fundamentals, Chemical Analysis series vol. 90, John Wiley and Sons, Inc, New York, 1987.

[13] Elaine Rich, Artificial Intelligence, McGraw-Hill, New York, 1983, p 5.

[14] A.L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers", Computers and Thought, E.A. Feigenbaum & J. Feldman (Eds.), McGraw-Hill, New York, 1963.

[15] Allan Turing, "Computing Machinery and Intelligence",*ibid.*

[16] A. Barr and E.A. FeigenBaum, The Handbook of Artificial Intelligence, Volume I, Addison Wesley, Don Mills, Ontario, 1981.

[17] A. Barr and E.A. FeigenBaum, The Handbook of Artificial Intelligence, Volume II, Addison Wesley, Don Mills, Ontario, 1982.

[18] A.Newell and H.A. Simon, Human Problem Solving, Prentice-Hall, Englewood Cliffs NJ, 1972.

[19] M.R. Quilian, "Semantic Memory", Semantic Information Processing, M. Minsky (Ed.), MIT Press, Cambridge MA, 1968.

[20]M. Minsky, "A framework for representing knowledge", The Psychology of Computer Vision, P.H. Winston (Ed.), 1975.

[21] Peter Jackson, Introduction to Expert Systems, 2nd Ed., Addison-Wesley, Don Mills, Ontario, 1990.

[22] D.A. Waterman, Artificial Intelligence, **1**, pp 121-170, 1970.

[23] E.H. Shortliffe, Computer-based Medical Consultations: MYCIN, North-Holland, New York, 1976.

[24] R.O. Duda; J. Gaschnig; P.E. Hart; K. Konolige; R. Reboh; P. Barrett and J. Slocum, Development of the PROSPECTOR consultation system for mineral exploration. Final Report, SRI Projects 5821 and 6415, SRI International, Inc., Menlo Park, Calif.

[25]R.K. Lindsay, B.G. Buchanen, E.A. Feigenbaum and J. Lederberg, Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project, McGraw-Hill, New York, 1980.

[26] D.A. Waterman, A Guide to Expert Systems, Addison-Wesley, Don Mills, Ontario, 1986

[27] N.V. Findler, ed., Associative Networks, Academic Press, New York, 1979.

[28] R.A. Kowalski, Logic for Problem Solving, North-Holland, Amsterdam, 1979.

[29]E. Charniak and D. McDermott, Introduction to Artificial Intelligence, Addison-Wesley, Don Mills, Ontario, 1984.

[30] E.A. Feigenbaum, "The Art of Artificial Intelligence: themes and case studies of knowledge engineering", Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1977, pp1014-1029.

[31] Patrick H. Winston, PROLOG Programming for Artificial Intelligence, Ivan Bratko, Addison-Wesley, Don Mills, Ontario, 1986.

[32] GOD personal communication.

[33] W.J. van Melle, System Aids in Constructing Consultation Programs, UMI Research Press, Ann Arbor Michigan, 1981.

[34] R. Davis and D. Lenat, Knowledge-Bases Systems in Artificial Intelligence, McGraw-Hill, New York, 1980.

[35] A.P. Wade; S.R. Crouch; D. Betteridge, *TRAC*, **7**, p 358, 1988.

[36] A.R deMonchy; A.R. Foster; J.R. Arretteig; L. Le; S.N. Deming, *Anal. Chem.*, **60**, p 1355, 1988.

[37] B.A. Hohne and T.H. Pierce, Ed., Expert Systems in Chemistry, American Chemical Society, Washington, DC, 1989.

[38] C.M. Wong; R.W. Crawford; J.C. Munz and T.P Kehler, "Application of artificial intelligence to triple quadrupole mass spectrometry", *Proceedings of the IEEE Nuclear Science Symposium*, San Francisco Ca, October 1983.

[39] H. Gelernter; G.A. Miller; D.L Larson and D.J Berndt, "Realization of a large expert problem-solving system, SYNCHEM2: a case study", *Proceedings of the First Conference on Artificial Intelligence Application*, IEEE Computer Society, December 1984.

[40] R. Englemore and T. Allan, "Structure and function of the Crysalis system", *Proceedings IJCAI-79*, 1979.

[41] W.R. Browette and M.J. Stillman, *Prog. Analyt. Spectrosc.*, **12**, p 73, 1989.

[42] S.D Brown; T.Q. Barker; R.J. Larivee; S.L. Monfre; H.R. Wilk, *Anal. Chem.*, **60**, p 252R , 1988.

[43] S.D. Brown. *Anal. Chem.*, **62**, p 84R , 1990.

[44] C.B. Gonzalez, *Acta Cryst.*, **A43**, p 622, 1987.

[45] O.A. Guell and J.A. Holcombe, *Anal. Chem.*, **62**, p 529A , 1990.

[46] I.O. Bohachevsky; M.E. Johnson and M.L. Stein, *Technometrics*, **28**, p 209, 1986.

[47] J.H Kalivas; N. Roberts and J.M. Sutter, *Anal. Chem.*, **61**, p 2024, 1989.

[48] S.B, Smith; M.A. Sainz and R.G. Schleicher, *Spectrochm. Acta, Part B*, **42**, p 323, 1987.

[49] W. Spendly; G.R. Hext; F.R. Himsworth, Technometrics, **4**, p 441, 1962.

[50] J.A. Nelder; R. Mead, Comput. J., 7, p 308, 1965.

[51] M.W. Routh; P.A. Swartz; M.B. Denton, Anal. Chem., **49**, p 1422, 1977.

[52] D. Betteridge; A.P. Wade; and A.G. Howard, *Talanta*, **32**, p 709, 1985.

[53] D. Betteridge; A.P. Wade; and A.G. Howard, *Talanta*, **32**, p 732, 1985.

[54] R. Fletcher and M.J.D. Powell, *Computer Journal*, **6**, p 163, 1963.

[55] J.A.C Broekaert; F. Leis and K. Laqua, Developments in Atomic Plasma Spectral Analysis, R.M. Barnes Ed., Heyden and Sons, London, p84, (1980).

[56] S.S. Bergman and J.W. McLaren, *Appl. Spectrosc.*, **32**, 1978, p 372.

[57] L. Ebdon, *ICP-Information Newsl.*, **7**, p 513, 1982.

[58] M.R. Cave; R.M. Barnes and P. Denzer, *ICP-Information Newsl.*, **7**, p 514.

[59] R.J Thomas and J.B. Collins, *Spectroscopy*, **5**, p38, 1990.

[60] P. Werner and H. Friege, *Appl. Spectrosc.*, **41**, p 32, 1987.

CHAPTER 2

Initial conceptualization for the autonomous instrument project indicated that spectral information not normally included in the spectral result set of standard polychromator would be required. A question we wanted an answer to was whether the work we were planning would only be applicable to rapid scanning monochromator systems. This study was undertaken to determine how much spectral information could be acquired using either a traditional polychromator with carefully placed detectors and a refractor plate in place, or spectrometers containing a small number of linear diode array based detectors.

Possible fallout of this study is the prospect of a polychromator that could use reduced numbers of photomultiplier tubes and still offer the spectral coverage required for even demanding environmental analysis, or a system using a small number of array detectors to capture all the spectral information required to perform comprehensive analysis.

The manuscript was published as:

SELECTION OF SPECTRAL WINDOWS FOR PLASMA ATOMIC EMISSION SPECTROMETRY

D.P.Webb and E.D. Salin
Department of Chemistry
McGill University
801 Sherbrooke St. W.
Montreal, P.Q.
Canada, H3A 2K6

Talanta, Vol 37, No. 1, 1990, pp33-38

ABSTRACT

A program has been developed to evaluate the number of spectral windows required to perform elemental analysis by ICP-AES, given a certain spectral window width. The data indicate that photodiode array systems (sequential slew scan) with as few as 7 or 8 acquisitions (windows) might be viable. With smaller windows, approaching in size those used by direct-reading spectrometers, it appears that conventionally designed spectrometers with photomultiplier tube detectors could use between 29 and 37 windows to determine 59 elements. This indicates that a general purpose direct reading spectrometer may be feasible.

INTRODUCTION

We have discussed elsewhere the performance of linear photodiode arrays with respect to their potential for atomic emission spectrometry[61,62,63,64]. Thought linear arrays continue to increase in size, there is no indication that this simple format, so well suited to simple grating systems, will reach a length and pixel count sufficient for plasma emission spectrometry with a single device. This is unfortunate, because these devices can provide simultaneous background spectral information which can be of enormous value, particularly when primitive (one and two point) spectral stripping schemes will fail. We have often wondered how a slew scanning (multiwindow) system might perform. Our previous work[62] indicates that photodiode arrays (PDA's) can give performance comparable to photomultiplier tube (PMT) detectors, although longer integration times may be necessary for unintensified systems. The real questions seem to be, 1) How many windows would be needed and 2) How wide should these windows be? Certainly the combined concerns of window width and number of pixel elements required would dictate the length and consequently the cost of the system.

We recently acquired a direct reading spectrometer equipped with a refractor plate for background correction purposes. The spectral window accessed by this system varies with wavelength, but is approximately 0.18 nm. We investigated the possibility of using a conventional system of this type for multielement analysis within a given window. Our hope was that a scanning capability combined with a judicious choice of compromise windows might allow the manufacture of systems designed to solve specific problems with a significant reduction in the number of "channels" (exit slit assemblies). Alternately, a conventional multi-channel spectrometer with broad applicability could be developed. Compared with a photo diode array based system the spectral windows used in photomultiplier based systems are of course much smaller, however the methodology of investigation is the same. We report below the results of an investigation of spectral windows for both types of instrument.

EXPERIMENTAL

We set about this investigation using the PROLOG language. PROLOG is a logic based language and well suited to questions like "How many 20 nm windows are needed to get the best lines of elements X, Y and Z?". In addition to its logic based orientation, it has inherent database-management facilities which were useful in the program development. We used a compiler version of PROLOG, Turbo PROLOG, rather than the more common interpreter form, because of the speed advantage expected from a compiled program. We utilized a somewhat modified version of the atomic spectral line database published by Winge et al.[65] as well as information provided by P.W.J.M. Boumans[66]. Though the database may not be as complete as some versions, it was quite adequate given the limited searching that we had in mind. A program called Windows-I was developed initially to investigate this problem. The program was provided with several sets of elements randomly selected from the list in Table 1. The program was also provided with a Degradation Factor[62] and a Window Width (in nm). All these parameters were read in from a file to allow convenient batch processing. Several of the parameters will be discussed in more detail below.

Ag	Al	As	Au	B	Ba	Be	Bi	C	Ca	Cd
Ce	Co	Cr	Cu	Fe	Ga	Ge	Hg	In	Li	Mg
Mn	Mo	Nb	Ni	P	Pb	Pt	Sb	Se	Si	Sn
Sr	Ta	Te	Th	Ti	Tl	U	V	W	Zn	Zr

Table 1. List of elements used

When initiated, the program reads in the selected set of elements. The Degradation Factor is used to extract the analytical lines of the various elements into the line set. It simply delineates the maximum level of detection limit degradation allowed during the line selection process.[62] For example, with a Degradation Factor (DF) of 3 and an element whose best line has a detection limit (DL) of 1 ng/ml, all the element's lines with detection limits of DL*DF = 3 ng/ml or less would be

read into the database. For simplicity, in our experiments, a given Degradation Factor has been applied to all elements in the set.

After the full database has been read in and pruned as described above, the window selection procedure begins. The time required for a given computation depends on both the number of lines (variable depending on the DF) and the number of elements. The relationships can be estimated by examining the times in Table 2 for various numbers of lines and windows. All computation times reported were based on times obtained using a 10 MHz 80286 CPU based system with zero wait state memory. When computations were carried out on other 8088 or 80286 CPU machines the values were corrected for the speed difference based on a calibration program run on both machines. When the element set consisted of 15 or more elements, the search time became excessive. In several cases we simply were not able to get data values even though we ran continually for 7 days on a 25 MHz 80386.

Windows-I utilizes a method of search for which PROLOG is particularly well configured. This is a "breadth first" method. It will search first for any combination of lines which will satisfy the requirements in a single window. Then it will try two windows, then three . . . *etc.* The search strategy utilized by Windows-I is exhaustive and so ensures that the best solution at any number of windows, if one exists, is found.

	Windows-I	Windows-II
5 elements, 10nm, DF3	$2.5x10^1$	$8.9x10^0$
10 elements, 10nm, DF3	$5.7x10^3$	$3.2x10^1$
15 elements, 10nm, DF3	N/A	$1.1x10^2$
20 elements, 10nm, DF3	N/A	$2.2x10^2$
25 elements, 10nm, DF3	N/A	$3.9x10^2$
30 elements, 10nm, DF3	N/A	$6.6x10^2$
5 elements, 10nm, DF10	$2.1x10^1$	$2.2x10^1$
10 elements, 10nm, DF10	$4.7x10^3$	$8.6x10^1$
15 elements, 10nm, DF10	N/A	$2.5x10^2$
20 elements, 10nm, DF10	N/A	$5.0x10^2$
25 elements, 10nm, DF10	N/A	$7.9x10^2$
30 elements, 10nm, DF10	N/A	$1.4x10^3$
5 elements, 20nm, DF3	$1.7x10^1$	$1.1x10^1$
10 elements, 20nm, DF3	$8.8x10^2$	$2.8x10^1$
15 elements, 20nm, DF3	$1.8x10^4$	$1.1x10^2$
20 elements, 20nm, DF3	N/A	$2.0x10^2$
25 elements, 20nm, DF3	N/A	$3.7x10^2$
30 elements, 20nm, DF3	N/A	$7.1x10^2$
5 elements, 20nm, DF10	$3.4x10^1$	$2.2x10^1$
10 elements, 20nm, DF10	$7.6x10^2$	$8.1x10^1$
15 elements, 20nm, DF10	$4.4x10^4$	$2.4x10^2$
20 elements, 20nm, DF10	N/A	$4.7x10^2$
25 elements, 20nm, DF10	N/A	$8.5x10^2$
30 elements, 20nm, DF10	N/A	$1.6x10^3$

Table 2. List of Times (Seconds) for Windows-I and Windows-II

As the data in Table 2 indicate, processing time becomes excessive for Windows-I when larger data sets are involved. These times are obviously unacceptable if one wishes to operate in real time. To overcome this problem a second program called Windows-II was developed, which we called the heuristic program, because it used an intelligent search strategy rather than the brute force, "test all possible combinations", approach used by Windows-I.

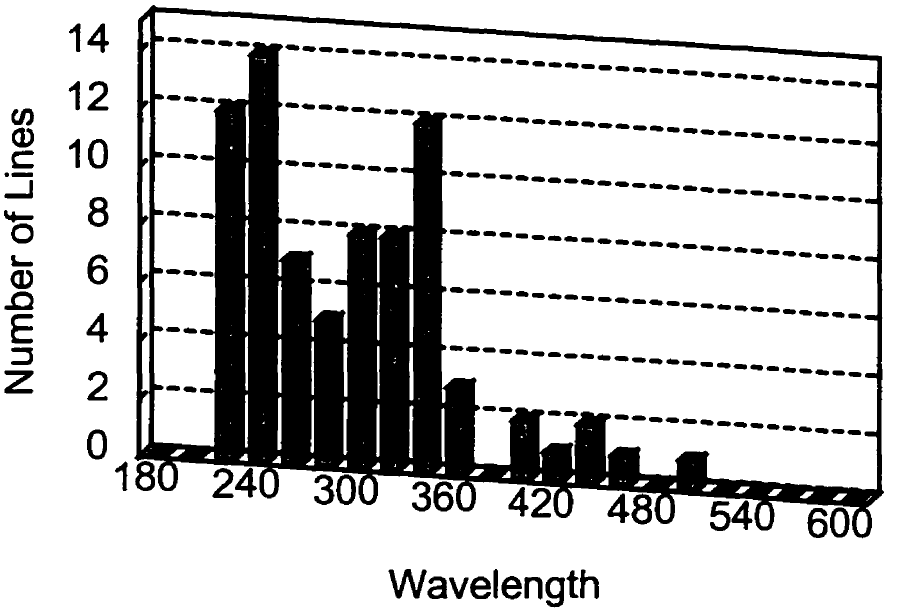


Figure 1. Lines for elements in Table 1 with detection limits less than 0.1 ng/ml.

The histogram of Fig. 1 indicates that the vast majority of the lines we would be interested in fall in a narrow spectral range (200-360nm). This suggested to us that an efficient and acceptable, but not necessarily perfect, solution to the window selection problem might be found by simply selecting windows based on the number of elements (elemental lines) which each window eliminated from the list of elements. In other words, whichever spectral window contains the most elements becomes the first window. The second window is the one which finds the most elements remaining in the line set. . . *etc.* This is the approach used by Windows-II. This algorithm is well suited to a recursive approach and encouraged us in our decision to use PROLOG again for Windows-II.

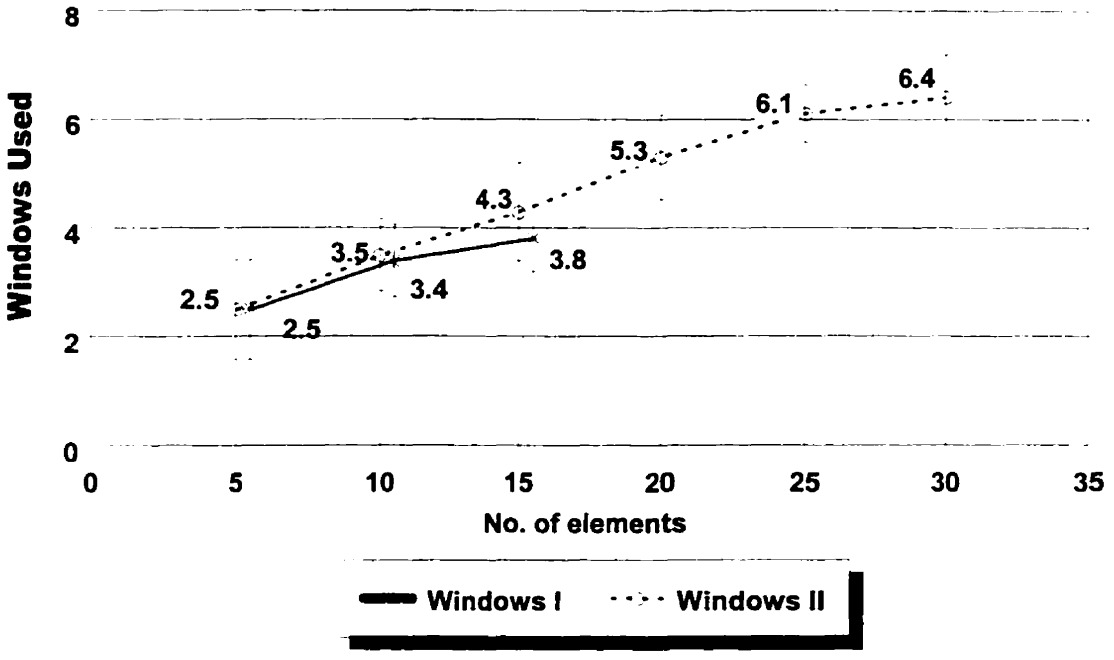


Figure 2. Number of 20 nm windows needed with DF of 3. In Figs. 2-5 the error bars represent one standard deviation

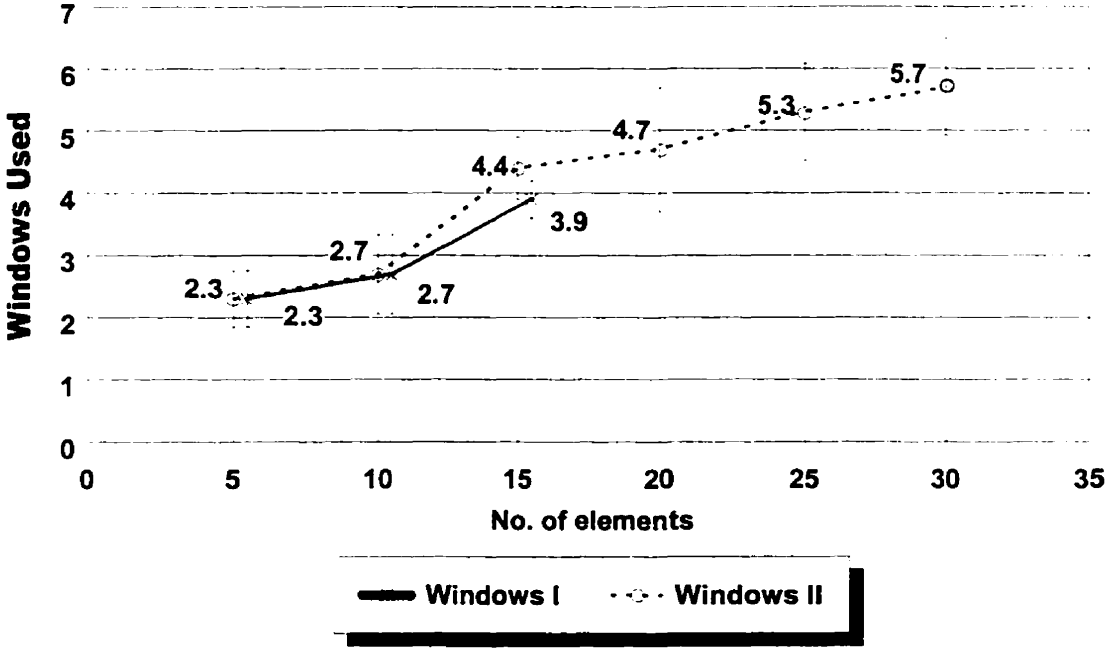


Figure 3. Number of 20 nm windows needed with DF of 10.

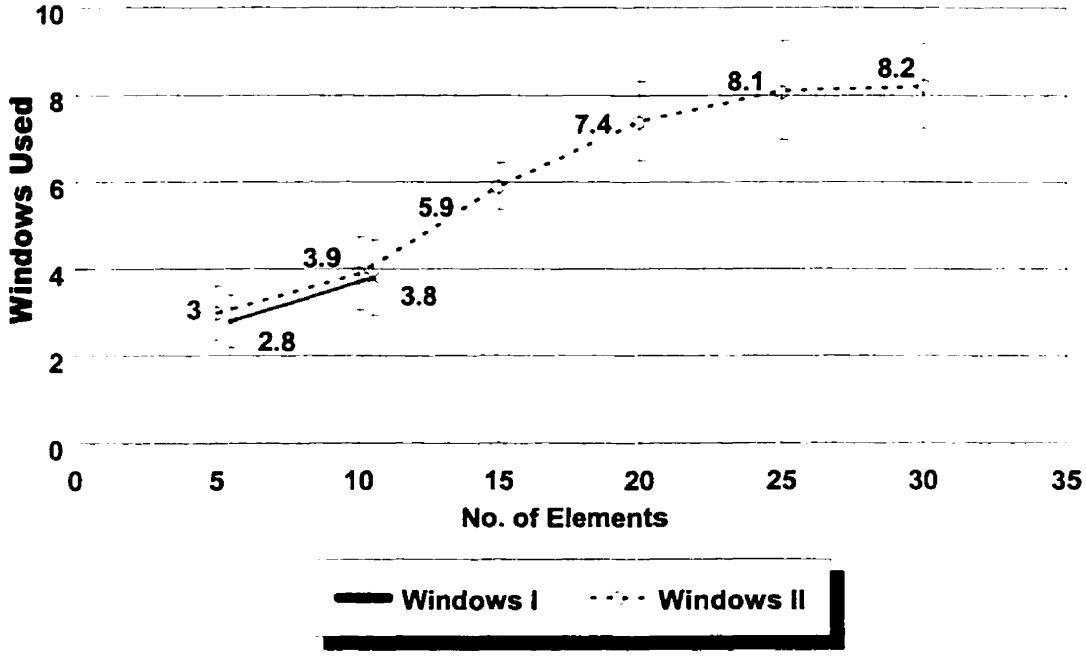


Figure 4 Number of 10 nm windows needed with DF of 3.

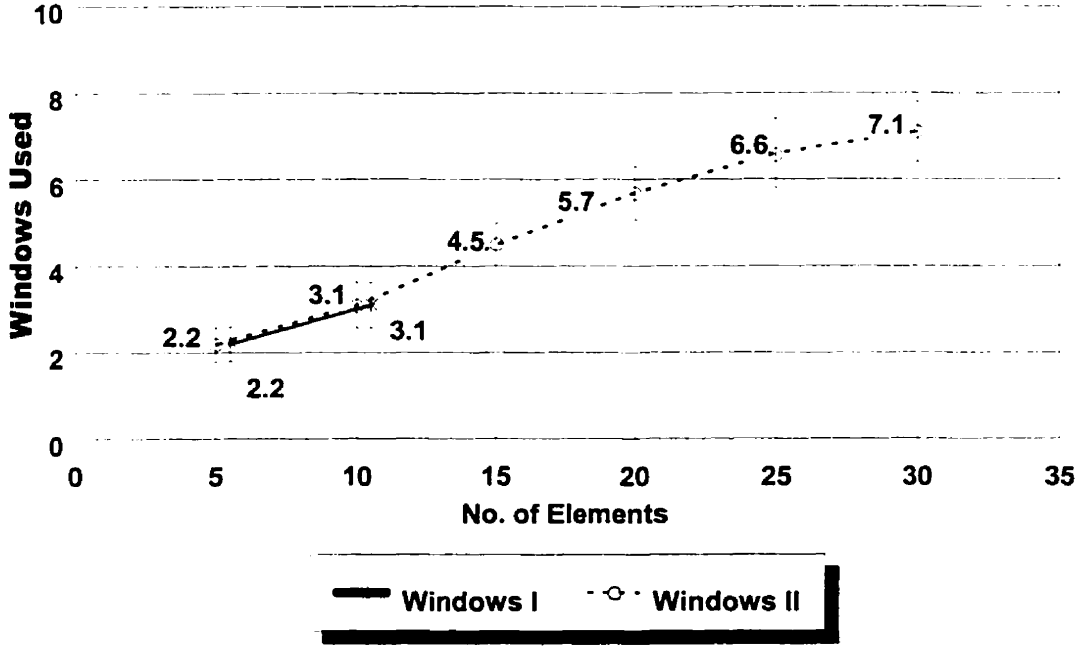


Figure 5. Number of 10 nm windows needed with DF of 10.

RESULTS AND DISCUSSION

As the data in Table 2 indicate, the scheme used for Windows-II is much faster, indeed, it seems to be fast enough to implement on a fast

microcomputer for real time operation of a windowed slew scan system. One must keep in mind that there are at least two common 32 bit CPUs (80386 and 68020) now used in microcomputers which operate at 25 MHz. Data from Windows-I and Windows-II are plotted together in Figures 2-5. The lower set of data has been slightly offset to the right to allow the error bars to be seen. The data from Windows-I runs are limited to very small data sets due to the computation time required. It is encouraging to note that the data provided by Windows-II runs are very similar or identical to those of Windows-I, indicating that our heuristic approach was good. The reader should keep in mind that Windows-I always finds the best solution. Windows-II uses a faster approach which will find a solution but not necessarily the best.

As the data in Figures 2-5 indicate, the number of required windows does not always increase linearly as the number of elements increases; however, it certainly has linear characteristics up to 25 elements. As would be expected, the lines/window ratio increases as the DF increases, simply because more lines become available, thereby enhancing the probability of solving the problem with fewer windows. We found the comparison between the data in Figs. 2 and 3 to be particularly interesting. Using a 1000 element linear array, one would have a spacing of 0.02 nm between element lines. There appears to be little advantage to using a Degradation Factor of 10. A Degradation Factor of 3 would allow most 30 element spectral analysis problems to be solved with 7 windows. By the use of a rapid scanning drive system and assuming that 10-sec integrations would be adequate, over half the analyses could be completed in about 70 sec. If the resolution must be doubled, then the data in Figs. 4 and 5 becomes applicable. Even in this situation, almost half of the problems can be solved with 8 windows.

To obtain similar quality data (with similar information for background correction) in the region of a single spectral line with a direct reading spectrometer specifically configured for those elements might reasonably take from 30 to 60 sec if 60 spectral positions (0.03 nm between positions) were used with 0.5 or 1 sec integration at each position. It should be noted that the 60 positions and 0.18 nm window are typical to our spectrometer system. The time required for a conventional scanning system to cover 30 elements could then be

estimated at 900 sec. If almost zero translation time and only 10 sec per element are assumed, the time spent would still be 300 sec. A scanning windowed system would offer flexibility in wavelength selection and would appear to have some advantage with respect to throughput at moderate to high light levels. It is probable, however, that an unintensified photodiode array operated in a typical linear dispersion system will not have a sufficient flux to allow 10 sec integration times for low light levels in the shorter wavelength regions of the spectrum[62].

This brings forth the question, "What can conventional multichannel PMT based systems with spectrum shifters do?" The Windows-II program was used for all this work, so superior solutions may be possible. The window width was set to three different values, 1.8, 0.4 and 0.18 nm. A 0.4 nm window is used in a modified Jarrell Ash system[67] which drives the galvanometer through a larger angle. The value of 0.18 nm is provided by the normal Jarrell Ash software on the Model 61 system. The wider 1.8 nm spectral coverage is possible with the same system by simply placing a thicker refractor plate in the system. Using only the 1.8 nm windows and a variety of 30 elements sets, with a Degradation Factor of 3, one can determine 30 elements in an average of 13.4 windows with a standard deviation of 1.2. The maximum number of windows found to be necessary was 16. If a Degradation Factor of 10 can be accepted, the average number of windows required is 10.9 with a standard deviation of 1.2. The maximum found necessary was 13. The question then arises, "How many windows does one need to do all the elements?". In this case we consulted several experts about elements that were monitored for environmental impact or used in common industrial processes. We arrived at a list of 59 elements. The elements added to our original database (Table 1) are listed in Table 3. The complete data set is listed in Table 4. The data indicate that as few as 17 windows are needed to solve this problem with a Degradation Factor of 10, or 21 with a Degradation Factor of 3. Major line overlaps will not be a problem in our determination because the closest major analytical lines in any of the windows were those of As (228.812) and Cd (228.802). We have been able to resolve this overlap on our system using the technique of multiple linear regressions[68].

Cs	Gd	Ho	Ir	K	La	Na	Os	Pd	Rb
Re	Rh	Sc	Sm	Y					

Table 3. List of Additional Elements

Window width, nm	No. of windows DF3	DF10
1.8	21	17
0.4	34	26
0.18	37	29

Table 4. Number of Windows Required for 59 Elements

These computations suggest to us that it should be possible to build a general purpose 59 element system that would satisfy the needs of many users. We have not in our work considered the spectral overlaps that can occur from minor lines of major elements in the matrix, a factor which might increase the number of windows, nor have we considered the use of the second order, which might reduce the number of windows by causing a given spectral line to appear in two places on the focal plane.

REFERENCES:

[61] S.W. McGeorge and E.D. Salin, Anal.Chem., **57**, pp 2740-2743, 1985.

[62] S.W. McGeorge and E.D. Salin, Spectrochim. Acta, **40B**, pp 435-445, 1985.

[63] S.W. McGeorge and E.D. Salin, Spectrochim. Acta, **41B**, pp 327-333, 1986.

[64] S.W. McGeorge and E.D. Salin, Spectrochim. Acta, **38B**, pp 633-638, 1986.

[65] R.K. Winge, V.J. Peterson, V.A. Fassel, Appl.Spectrosc., **33**, pp 206-219, 1979.

[66] Personal Communication consisting of 450 major lines. P.W.J.M Boumans. 1986.

[67] Personal Communication. Guy Legere. Technical Services Laboratory, Toronto, Ontario, Canada, 1988.

[68]Lyne Gervais and E.D. Salin. Atomic Spectral Deconvolution by Multiple Linear Regression, Internal research memo.

CHAPTER 3

If a rapid scanning monochromator were to be used as the platform for our autonomous instrument system a great deal of flexibility is gained if the instrumental system, rather than the operator, has the ability to select the emission lines that should be analyzed. Automated instrumental line selection has the advantage that a relatively inexperienced operator, or an expert operator with a very complex sample matrix at hand can avoid selecting lines that are likely to exhibit spectral overlaps. Furthermore, if a spectral overlap is unavoidable, then it would be useful to have a system that could automatically identify such cases and perform spectral subtraction where possible. Furthermore, because the analysis of each line is expensive in terms of time, and sample used, a system that minimizes the number of lines that need to be analyzed would be a valuable asset to the automated instrument.

The manuscript was published as:

A LINE SELECTION EXPERT SYSTEM FOR CONTROL OF SCANNING INDUCTIVELY COUPLED PLASMA ATOMIC EMISSION SPECTROMETERS

Douglas P. Webb and Eric D. Salin*
Department of Chemistry
McGill University
801 Sherbrooke St. W.
Montreal, P.Q., Canada
H3A 2K6
(514) 398 6236

Journal of Analytical Atomic Spectroscopy, Vol 4, December 1989, pp. 793-796.

ABSTRACT

An expert system for control of a rapid scanning spectrometer to perform atomic emission spectrometry has been developed using the PROLOG language. The expert system can be "taught" the elemental composition of a given sample type. It uses this information (when available) to generate a line search strategy which minimizes the number of lines which need to be measured. The rules used by the system can be observed during system operation to allow a trace of the logic. Reading of the initial data base takes no more than 6 seconds and then decisions are made in less than 1 second per line.

INTRODUCTION

As the power and complexity of modern instruments increases, the level of expertise required of the operator can also increase dramatically. Fortunately, the reduction of the cost of small computers has enabled much of the operational expertise to be programmed into the instrument by the manufacturer. While these programs have minimized the mechanical expertise required of the operator, very few have attempted to interact with the data to make "intelligent" or "expert-like" decisions on the operation of the instrument. Recent software releases of "artificial intelligence" languages have enabled microcomputers to utilize software tools that had previously been available only on larger computer systems. Symbiotically, the power of microcomputer systems has increased dramatically in the last few years to the point where the speed of processing approaches that of the minicomputer of several years ago. These developments have led to the generation of this microcomputer based, prototype expert system for the control of a rapid scan spectrometer.

The expert system was developed in what has been described as a "Fifth Generation Language", PROLOG, or more specifically Turbo PROLOG[69], which contains many of the features found in the Clocksin and Mellish[70] implementation of the language. As one advances upward from machine code programming to assembler level programming through the "higher level" languages of BASIC, APL, FORTRAN and Pascal which are used in the scientific community, one finds that each level generally requires less programming to carry out a specific function. Eventually, of course, all instructions are converted to machine code before execution, but the higher level languages offer ease of use to the programmer, and this usually turns into a faster completion of the programming assignment. PROLOG offers distinct advantages over other languages for the solution of many problem types. Its name means PROgramming in LOGic. The specific type of logic used is a subset of predicate calculus. The format of a PROLOG program is similar to the AND and OR conditions used in electronic circuit design or while searching databases such as STN (Chemical Abstracts on Line). PROLOG searches through its set of facts and rules (relationships) to

ascertain whether a given statement (the goal) is correct. PROLOG offers a number of advantages over traditional high level languages. The first advantage is that PROLOG has a built in search strategy called "backward chaining"[71]. PROLOG also provides powerful database management tools. If a search strategy other than backwards chaining is desired, it can be implemented in PROLOG; however, the efficiency of execution will be lower than one would obtain using backward chaining.

PROLOG has been used extensively in computer science and cognitive science, however its applications to date in chemistry have been limited. In this area, PROLOG has been applied to pattern recognition problems[72], IR spectrometry[73,74] synthesis methodology recommendations[75] and chromatographic separation enhancement[76]. Since it is a logic based language, rather than the traditional procedural type of language, it is not surprising that it has been used to reason about chemical systems[77,78].

Many expert systems have been developed in other languages. While these cover a wide variety of topics, several are particularly noteworthy in the context of this paper. Brunnee *et al*[79] and Wong and Crawford[80] reported an expert system developed in LISP for the control of a mass spectrometer, another form of rapid scanning instrument, and Fielden *et al*[81] described an expert system for use in electroanalysis. It should also be mentioned that there is a growing body of work on process control and fault diagnosis using expert systems[82].

EXPERIMENTAL

Design Considerations

A prototype expert system has been developed to directly control the selection of lines of rapid scan spectrometers. The system is, at present, generic. It is not configured to operate with any specific spectrometer system. For ease of use, the system will be referred to as Linex (for LINe EXpert). Some technical details are provided in Table 1.

Computer–	
AST Model 140R	AST Research Inc.
(IBM-PC AT type)	Irvine, Ca., U.S.A.
CPU:	80286 (10 MHz)
NPU:	80287 (10 MHz)
Memory:	1 MByte no wait state memory (10 MHz)
Mass Storage Used:	40 MByte Winchester rigid disk, with 28 ms average access time.
Software–	
Turbo PROLOG V1.1.	Borland International Scotts Valley, Ca., U.S.A.
Compiled Code Size:	150 kBytes
Data Base Size:	511 Lines for 48 Elements

Table 1. Technical details of hardware and software.

While it could operate in a multitasking environment conveniently on the 80386 generation computers, the system was developed to run on an independent IBM-PC under MS-DOS. A version of Linex that actually controls a spectrometer will have three separate software modules: (1) The user interface, which acts as intermediary between the system operator and the instrument interface. (2) The expert system, which selects spectral lines. (3) The instrument interface, which controls the actual instrumental hardware. Only the instrument interface is specific to any particular instrument manufacturer. Up to this point, prototypes have been developed for only the first two modules.

Many spectrometer systems now have their own internal controller which handles hardware operation, data acquisition and primitive processing. An external computer usually provides mass storage, high level processing, graphics and a color user interface. It would be possible to implement Linex on many such external computers, however the expert system was developed in a "stand alone" configuration for several reasons: (1) it is easier to prototype, (2) it will be easier to retrofit to existing systems and (3) a suitable spectrometer/computer system was not available for testing. Our assumption has been that a distribution of intelligence and responsibilities would be generally superior from a performance standpoint, and that the spectrometer system will find wavelengths correctly and perform background correction. The expert system is designed to order the selection of lines so as (1) to minimize the number of lines that must be measured and (2) to reduce the possibility of using a line subject to a spectral overlap which cannot be resolved by the spectrometer system.

Linex was developed in PROLOG (Turbo PROLOG, Borland International) for several reasons: (1) Borland's version of this language can be compiled and consequently is suitable for distribution in a form (compiled) which cannot easily be corrupted (2), as a compiler, it operates at a relatively high speed compared to interpreter versions of PROLOG, (3) it has some useful features (*e.g.*, graphics, interfaces to other languages) which make it more flexible than conventional PROLOGs. PROLOG was used rather than an expert system shell because of the flexibility provided by a language. A brief description of shells and the problems involved in their use may be gleaned from the work of Wade *et al*[83]. PROLOG was selected over other programming languages due to it's inherent structure (built in database management and search mechanism), which makes it well suited to programming problems of this type.

Linex is designed to minimize the number of line acquisitions required for a given analysis. This decision was implemented with the expectation that the acquisition of information about a line from the spectrometer system, and not the computation involved in determining the next appropriate line, would be the rate determining step in analyses. While this may not be appropriate for very rapid scanning systems, it

was felt to be appropriate for a generic system. Therefore, Linex tries to get as much information as possible from the operator before starting acquisitions. The information consists of maximum concentration levels expected for the various elements based on any knowledge that the operator may have. This information is stored in a "template." Linex can then use the approximate concentrations to estimate the intensities of the various elemental lines. If the predicted intensity is below detectable limits then one would not expect these lines to appear in the spectrum. Linex uses this information to reduce the number of lines which are used in the "run time" database. For example, if the template stated that the sample type had 10 ppb or less of Fe, then only 4 lines would be kept in the run time database, because only 4 Fe lines have detection limits at or below 10 ppb. This would eliminate 14 lines in our limited database and even more if a more complete database were used. For a commonly evaluated sample type, a standard template would normally be developed. General purpose templates can also be developed. Any template can be modified for the operator's specific sample type, and new templates can be developed. A template encompassing 60 elements can be generated in about 10 minutes by filling in a table.

A "complete" table of lines, intensities and detection limits called FULLIST is accessed after the template has been created. FULLIST is based on published information[84]. Any spectral line database with the same type of information can be used. Detection limits are used to calculate relative intensities. On a "real" system the intensities would be determined once empirically since different systems and even different conditions can affect performance.

Program Logic

There are two possibilities for analysis by Linex, Full Search and Partial Search. Only the Partial Search is now operational. The Partial Search involves the use of a template as described above. The Full Search can be easily implemented by simply having the spectrometer examine the major line of each element to get an approximation of the maximum possible concentration of each element. A template could then be generated using these approximate values. The system then continues in the manner used for Partial Search employing the Full

Search values template. The bandpass of the spectrometer is an input parameter. Linex uses bandpass to determine whether or not two spectral lines overlap. This leaves the system quite flexible at this stage of development, because it allows the emulation of a variety of spectrometers.

Setup of the template and resolution is accomplished in a windowed environment. Simulated spectrometer operation results in the appearance of two windows. One window is the "operator" interface which allows the operator to trace program progress and record results. The second window is a mock instrument interface, which allows the user to simulate instrument response. This allows the operator to evaluate the program logic. The expert system tells the instrument (instrument interface) which wavelength to seek, and the instrument (instrument interface) is expected to respond with a background corrected intensity. This intensity and all previous intensities are used in conjunction with the template information to make decisions about which wavelength should be accessed next. Figure 1 contains a sample interaction between the expert system and the machine interface (which is simulated by the user). Table 2 contains a list of the spectral lines pertinent to the example.

EX: Rule(2) - We must look to see if Cd's most intense unobstructed line is present, because if it is, then we can directly calculate the concentration of Cd. **[A]**
MI: Look for Cd's line at wavelength 361.051. What is the intensity? (0.0 for no line) ***0***
EX: Rule(3) - We must look for Cd's most intense line. If nothing is seen then we know that the concentration of Cd is below the detectable limit.**[B]**
MI: Look for Cd's line at wavelength 214.438. What is the intensity (0.0 for no line) ***36***
EX: Rule(4) - We must look for Cd's 2nd most intense line and if the measured ratio between lines 1 & 2 are comparable to the documented values then it is assumed that neither line is interfered with and the concentration of Cd can be calculated directly from the intensity of line 1.**[C]**
MI: Look for Cd's line at wavelength 228.802. What is the intensity (0.0 for no line) ***24***
EX: Rule(8) - Line 1 was definitely interfered with, line 2 may not have been. Look for Cd's 3rd most intense line and if the 2:3 intensity ratio is comparable to know values, then, as previously indicated, the concentration of Cd can be determined.**[D]**
MI: Look for Cd's line at wavelength 226.502. What is the intensity (0.0 for no line) ***12***
EX: Rule(9) - Direct measurements, and ratios have failed therefore an attempt will be made to strip away a single element which overlaps Cd.**[E]**
EX: - - STRIP - - Attempt to discern concentration of As so we can strip it from and overlap with Cd.**[F]**
EX: Rule(2) - We must look to see if As's most intense unobstructed line is present, because if it is then we can directly calculate the concentration of As.**[A]**
MI: Look for As's line at wavelength 197.197. What is the intensity (0.0 for no line) ***3***
EX: The concentration of As is known to be 1.96 ppm

EX: Rule(10) - Cd is interfered with so the intensity of the interfered with line must be measured so the strip can be done. As interferes with Cd at wavelength 228.802.**[G]**
EX: Cd's line at 228.802 nm has already been measured. The concentration of Cd is known to be 24.8 ppm.**[H]**
As 1.96 ppm **[I]**
Cd 24.8 ppm

Figure 1. Printout of a sample session with Linex. This example is a determination of Cd using a full database (*i.e.*, no template). EX is the explanation facility information. MI is the machine interface information and includes wavelength selection information requests for input (observed intensity). Operator input is in bold italics. At the end of the operation the goal is reached and the searched for concentration (Cd) is printed out with those determined incidentally (As). This example illustrates one possible path through the flow chart in Fig. 2. The approximate location in the flow chart (Fig. 2) is marked with the character in the brackets.

Cd lines in order of intensity	Relative Intensity	Interferent
214.483 nm	1.0	Pt 214.423 nm Sb 214.486 nm
228.802 nm	0.926	As 228.812 nm
226.502 nm	0.735	Te 226.555 nm
361.051 nm	0.011	No interferent

Table 2. Cd. Lines and interferences in the example database. These interferents are based on our database of only 511 lines.

Linex is designed to minimize propagation of error. Given the template information it first tries to find a line for a given element which will be unobstructed (*i.e.*, no spectral overlap) by any of the other elements at the concentrations listed in the template. If it finds an unobstructed line, it then makes a determination and prints out the concentration. If an unobstructed line is not found, then Linex attempts to use ratios of the major lines of the analyte to do a quantitative analysis. This is done first because this method, if successful, has no propagation of error. If this method fails then a spectral strip is attempted. Figure 2 contains a partial flow chart of the logic path. The algorithm is in fact recursive, and therefore requests for a spectral strip actually push the current state of search on the stack and start a new query at the 'Has This Element Already Been Analyzed?' box.

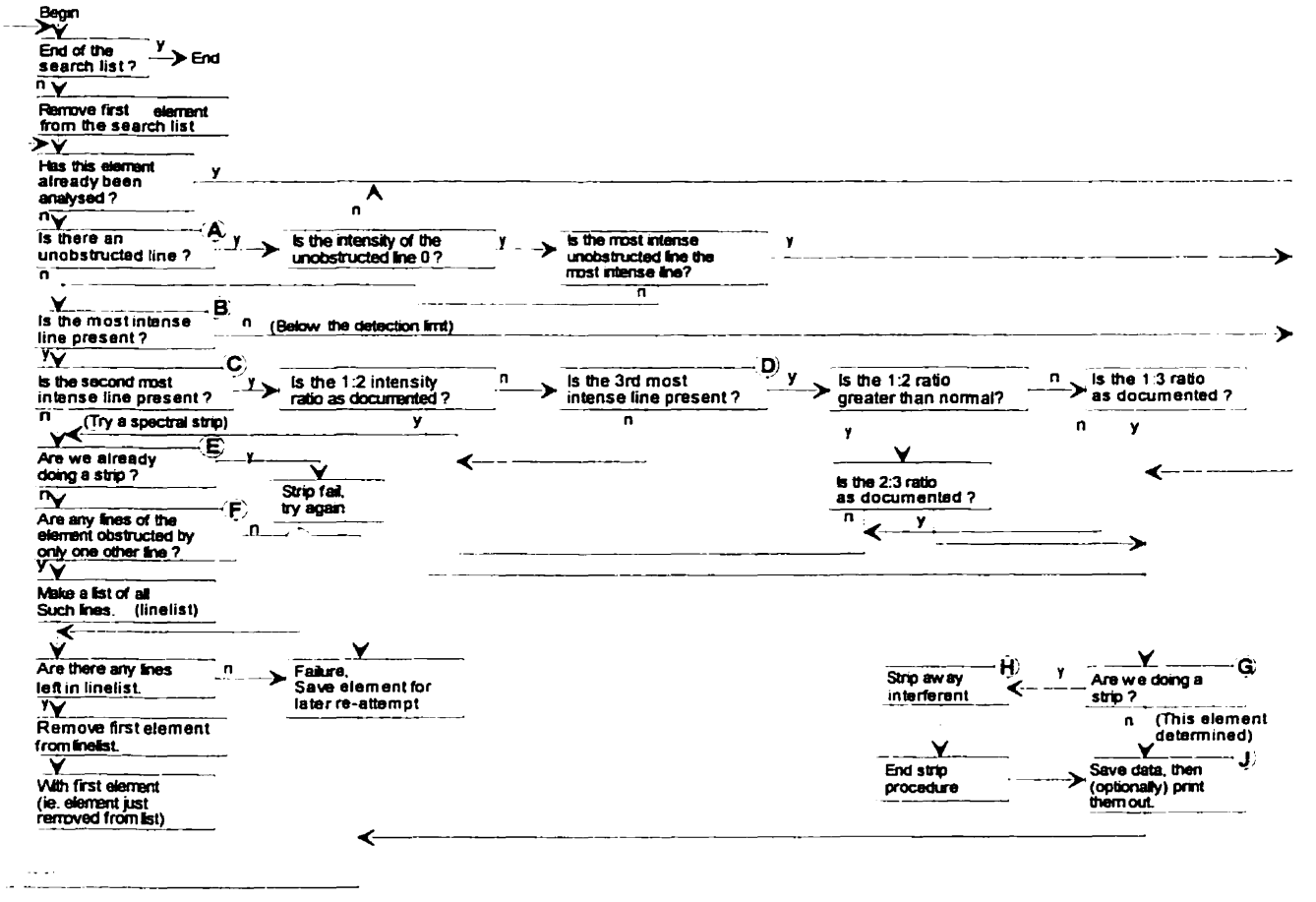


Figure. 2 Partial flow-chart of Linex logic path

To minimize propagation of error, only one strip is allowed on a given line, so Linex searches the database for a line which has only one possible elemental overlap. If it finds such a line, then it first checks to see if the interfering element can be determined unambiguously using the rules above. If the interfering element itself cannot be determined without a strip, then the procedure is terminated. An element not determined on the first pass might be noted, then retried after all other elements had been determined, because the working spectral line database will likely be reduced as elemental concentrations are determined, allowing a single element spectral strip where none was available before. This feature is not currently implemented.

The system proceeds so rapidly that to monitor the system's operation without intentionally slowing it down it was necessary to have Linex print the rules while the system is operating (rather than display

them on the screen). It was determined that the most convenient way to test the system is to modify the spectral data base to generate situations which forced the various rules to be used.

Currently elemental concentrations are determined in alphabetical order. Ordering to reduce the lineset as rapidly as possible, or determine elements who have the greatest likelihood of an unobstructed line might result in fewer lines being analyzed - this has not been implemented.

RESULTS AND DISCUSSION

At the beginning of each search Linex uses the applicable template to reduce the full data set of spectral lines to the set which is applicable. The remaining set is maintained in memory for rapid access. The reduction of the Full List, which is stored on a hard disk (40 ms average access), takes approximately 6 s in the worst case and 3 s in the best case. After this, the disk is not accessed.

At this point of development, the instrument response is simulated by the operator. After the ENTER key is pressed, the system responds within 1 s with either an answer or a command to record another line. Given that we expect to use a system which will require at least 1 s to find a line and perform the necessary measurement, this computation speed is considered adequate. It should be pointed out that the data base used for preliminary work only includes the major lines of 48 elements. A more realistic run time data base would include all the likely interfering lines around each of the major analytical lines. This might require the use of a more powerful CPU (80386) and/or the use of additional heuristics (rules) to reduce the run time line set as rapidly as possible. For example, one might wish to determine the level of Fe, a common interferent in certain sample types, at the start of the analysis and then undergo a further pruning of the data base.

The program was relatively easy to implement given that it took only 250 hours of programmer time. It has combined heuristics conveniently with the simple arithmetic necessary for the spectral strip and concentration calculations. The use of a language, rather than an expert system shell, has provided the flexibility necessary to do a task which might be inconvenient or impossible for a software package which was not designed for instrument control. The use of a language founded in logic has allowed the program to be easily structured in a logical manner. This was much more efficient than generating similar logical structures in a language not specifically designed for the purpose. The developmental process was dominated by the concern "how would an expert do it?" rather than the concern "how can this be conveniently programmed?" The logical structure has aided the debugging process significantly. The powerful database features of PROLOG have

eliminated or drastically reduced database organization requirements, thereby minimizing one of the major time consumers in a programming task of this type.

It is hoped that this will serve as a cornerstone of what will someday be fully automated scanning spectrometers which will self calibrate, self optimize, diagnose hardware problems, and initiate analytical methodologies when necessary. The real time response is adequate for many experiments, and the relative ease of implementing the system in PROLOG makes apparently difficult logic based problems much more feasible. Turbo PROLOG and many other PROLOGs now have extensive mathematical functions available which have not been available in traditional PROLOGs. This should further widen the applicability of this language to instrumental problems.

ACKNOWLEDGMENTS

The authors would like to acknowledge the initial prototyping work done by Lisa Kerr. The authors thank the Canadian Natural Sciences and Engineering Research Council for support under OPG 1126. D.P.W. acknowledges scholarship support from the Province of Quebec FCAR program.

REFERENCES:

[69] Clocksin, W. F.; Mellish, C. S., Programming in PROLOG, Springer-Verlag, Berlin, 1981.

[70] Turbo PROLOG Owner's Handbook, Borland International, California, 1986.

[71] de Monchy, A. R.; Forster, A. R.; Arretteig, J. R.; Le, L.; Deming S. N., *Anal. Chem.*, **60**, p 1355, 1988.

[72] Kleywegt, G. J.; Van't Klooster, H. A., *Trends Anal. Chem.*, **6**, p 55, 1987.

[73] Postma, G. J.; Vandeginste, B. G. M.; Van Halen, C. J. G.; Kateman, *Trends Anal. Chem.*, **6**, p 27, 1987.

[74] Luinge, H. J.; Kleywegt,G. J.; Van't Klooster, H. A. ; Van der Maas, J. H., *J. Chem. Inf. Comput. Sci.*, **27**, p 95, 1987.

[75] Moseley, C. Warren; LaRoe, W. D.; Hemphill, C. T., *ACS Symp. Ser.*, **306**, p 231, 1986.

[76] Gunasingham, H.; Srinivasan, B.; Ananda, A. L., *Anal. Chim. Acta*, **182**, p 193, 1986.

[77] Lyall, A.; Hammond, P.; Brough, D.; Glover, D., *Nucleic Acids Res.*, **12**, p 633, 1984.

[78] Darvas, F.; Futo, I.; Szeredi, P., *Int. J. Bio-Med. Comput.*, **9**, p 259, 1978.

[79] Brunnee, C.; Jung, G.; Markwardt, U.; Pesch, R.; Wollnik, H., *Am. Lab. (Fairfield, Conn.),* **18**, p 102, 1986.

[80] Wong, C. M.; Crawford, R. W., *Int. J. Mass Spectrom. Ion Processes*, **60**, p 107, 1984.

[81] Fielden, P. R.; Carr, R. N.; Oduoza, C. F., *Anal. Chem. Symp. Ser.*, **25**(Electrochem., Sens. Anal.), p 55, 1986.

[82] Goff, K. W., *Proc. - Refin. Dep. Am. Pet. Inst.*, **65**, p 487, 1986.

[83] Wade, A. P.; Crouch, S. R.; Betteridge, D., *Trends Anal. Chem.*, **7**, p 358, 1988.

[84] Winge, R. K.; Peterson, V. J.; Fassel, V. A., *Appl. Spectrosc.*, **33**, p 206, 1979.

CHAPTER 4

Sample introduction is arguably the weakest link in the ICP-AES sample analysis operation. Nebulizers can clog either entirely, or perhaps more insidiously, partially. The plasma can be blown out if, for example, a sample containing ethanol or methanol is introduced into the plasma and the R.F. power is set too low. In the later case the system must recognize that the plasma is out and restart the system. If the nebulizer is completely blocked operator intervention would be required - an undesirable eventuality in an autonomous system.

It would be useful, especially for an autonomous ICP system, to be able to determine sample properties that might lead to such a sample introduction induced system failure. Ideally this determination should be made early enough that remedial action could be taken to either clear the nebulizer or prevent the plasma from blowing out.

One mechanism by which a surprising amount of information about sample introduction and to some degree about the sample itself can be obtained is to listen to sample nebulization.

The manuscript was published as:

ICP SAMPLE MATRIX IDENTIFICATION BY ACOUSTIC SIGNATURE

Douglas P. Webb, Joan F. Power and Eric D. Salin*
Department of Chemistry
McGill University
801 Sherbrooke St. W.
Montreal, P.Q., Canada
H3A 2K6
(514) 398 6236

Applied Spectroscopy, Vol 46, 1992, pp. 1362-1369.

INTRODUCTION:

As part of our effort in the development of intelligent instrumentation, we are in the process of developing an autonomous inductively coupled plasma atomic emission spectrometry (ICP-AES) system. This system is to be a stepping stone to ICP mass spectrometry (ICP-MS) and other analytical systems. Systems which are designed to run with little or no operator intervention must evaluate what type of operating conditions should be used for each sample to achieve accurate results as rapidly as possible. To do this they must obtain information which describes the type of sample. Autonomous systems suffer from the handicap that they cannot make some types of observations that are simple for humans; "The sample is cloudy", or "The sample is viscous". One of the difficulties inherent in an autonomous system is that it has only two sources of information about the analyte and its matrix: 1) An initial operator sample description. 2) Spectral signal. The operator may have no information on the sample's composition. This lack of knowledge can impair the system's ability to avoid problems. For example, high salt samples can cause nebulizer blockage, or organic matrices can extinguish the plasma. Both of these types of problems can manifest themselves very rapidly. If a system is dependent on spectrometry to pinpoint these hazards, it may already be too late by the time they are identified. Information about the sample matrix is important in setting the optimal operating conditions for ICP-AES, but it is even more important in ICP-mass spectrometry because information about the solvent may allow prediction of some of the types of spectral interference effects which will appear. ICP-MS is also subject to deposition on the skimmer cone, as well as memory effects if the concentrations of solvent are inappropriate. What follows is a method with the potential for rapidly ascertaining the physical properties of the sample matrix.

While electrical, thermal, magnetic and optical characteristics of matter have long been under investigation by analytical chemists, recently one of the oldest and more easily recognized properties of physical phenomena has been enjoying renewed interest: acoustic

emission. Adrian Wade et al[85] have recently presented a report documenting some of the current analytical work being done in the field of acoustic emission. Acoustic measurements have been applied to the measurement of the reactions occurring in the inductively coupled plasma[86]. More recently acoustic emission measurements have been used as an internal standard to normalize the atomic emission signal or laser enhanced ionization signal from a laser-generated plume in ICP systems which uses laser ablation as a sample introduction technique.[87,88,89,90] Some recent acoustic studies in ICP have focused on audio-frequency-noise in the plasma and attempts to reduce it.[91,92]

Sample introduction has been described as the Achilles heel of modern ICP-AES[93]. We will discuss here a technique which allows the monitoring of the sample introduction for inductively coupled plasma spectrometry by the acoustic emission of the nebulizer/spray chamber system. The technique we propose might be used either alone or in conjunction with other techniques, to give some *a priori* information on the sample matrix. This information could be used to modify operating conditions for optimal results and prevent sample introduction system failure.

Experimental:

The ICP system

A Thermo Jarrell-Ash (TJA) Atomscan 2400 was used for the study. It is a 27 MHz ICP system which was operated at 1.25 kW for the experiments with a Plasma Gas flow of 16 l/min unless otherwise stated. All gas flows were argon. A Watson-Marlow 503U peristaltic pump was used to introduce sample at approximately 2.4 ml/min with a TJA fixed-crossflow nebulizer. The spray chamber and torch are standard TJA, and are illustrated in Figure 1. Equipment and software items are listed in Table 1.

Measurement electronics

An electret condenser tie-clasp microphone was used as the acoustic transducer for these experiments. The microphone was connected to the spray chamber drain as illustrated in Figure 1.

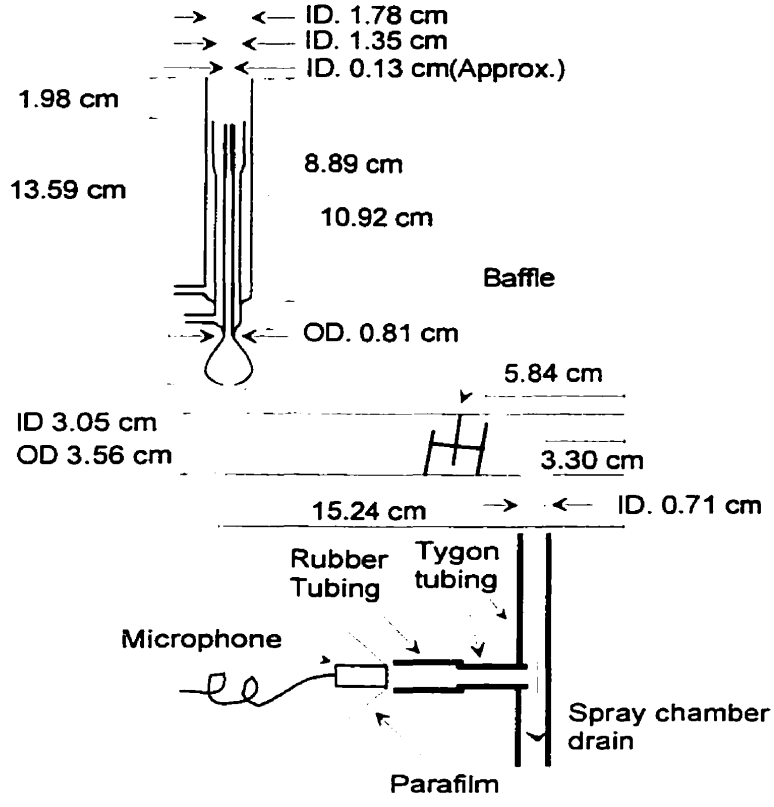


Figure 1. Dimensions and geometry of the torch and spray chamber assembly used for most experiments, and geometry of the microphone connection.

Stretched parafilm was used to protect the microphone from corrosive solvents. The system was tested both with and without the stretched parafilm present, and while the intensity was reduced by 10-20% with the parafilm present, the power spectrum did not change substantially. The microphone has a flat response between 150 and 3000 Hz, and a slightly decreasing response to 50Hz. The output signal from the microphone was amplified by an in-house designed and built two-stage amplifier whose gain could be varied between about 100 and 1000 by changing resistors. Gain was varied to keep the signal as high as possible without going out of the analog to digital converter's (ADC) range. The circuit was used to convert the output from the microphone into the +10V to -10V bipolar signal which was suitable for the Tecmar W/TM-40 PGL data acquisition board which was attached to our micro

computer based data acquisition system. The data acquisition board is a sample and hold amplifier with a 12 bit ADC coupled to a 16 MHz, 80386-based, IBM compatible microcomputer. Data were acquired at 4000 Hz unless otherwise stated.

When preparing our data acquisition system, a Krohn-Hite model 3700 filter (a 4-pole Butterworth filter[94]) was initially incorporated between the amplifier and the ADC, and later between the microphone and the amplifier. This unit has two filters, both low and high pass, each having a -24dB/octave roll-off. It was used to ensure that we were obtaining unaliased noise power spectra. Lower and upper cut-off frequencies of 1 Hz and 1000 Hz were used. We found that the signals with and without the filter in place were identical to within experimental error, so the filter was not used for any of the experiments described here.

A variable frequency waveform generator was used to determine if the amplifier circuit was attenuating the signal at the higher frequencies which were examined. We found that an input signal at 100Hz was amplified 8% more than a 1000Hz signal of equal amplitude. For the purposes of these experiments this was deemed acceptable because relative peak heights at different frequencies are not being compared.

Computer Calculations

The noise power spectra were calculated using Fourier Transform routines in Labcalc version 2.2 (Galactic Industries Corporation). Time series data were 4096 points each unless otherwise stated, and 25 such sets of data were individually transformed and averaged in the frequency domain to obtain the final spectrum. Because the gain of the amplifier was varied between certain series of experiments, ordinate values are not calibrated and amplitudes should not be directly compared except where they appear on the same graph or is otherwise specifically stated.

Component -	*Model -*	*Supplier -*
Spectrometer	Model Atomscan 2400	Thermo Jarrell-Ash, Franklin, MA, USA
Nebulizer	Fixed-crossflow	Thermo Jarrell-Ash
Spray Chamber & Torch	Model 2400 type	Thermo Jarrell-Ash
Pump	Watson-Marlow 503U	Fisher Scientific, Pittsburgh, PA, USA
Microphone	Electret Condenser Model ME4153	Electrosonic, Willowdale, ON, Canada
Analog to Digital Converter	Tecmar W/TM-40 PGL	Scientific Solutions, Cleveland, OH, USA
Analog Signal Filter	Model 3700	Krohn-Hite, Avon, MA, USA
Operational amplifiers	Motorola LF355J	Electrosonic
Laboratory Film	Parafilm "M"	American National Can Greenwich, CT., USA
Power supply	Powerace, Model 103	P. A. Products, Painesville, OH, USA
Data Analysis Software	Lab Calc Version 2.2	Galactic Industries Corporation, Salem, NH, USA

Table I - Description of equipment and software

Results And Discussion

Wideband Frequency Spectra

Initial experiments focused on determining where in the acoustic spectrum to look for useful information about the condition of the nebulizer and the spray chamber/torch assembly. Initially, data were acquired at 8000 Hz so that high frequency information could be evaluated. It was determined, by inspection of the power spectra, that the information content dropped to near zero above 1300 Hz and that most of the useful information was below 800Hz. Thereafter we confined our investigations to below 800Hz. There appeared to be two frequency ranges of interest. The area between 450-800 Hz varies substantially

between sample types and with varying operating conditions. Under most operating conditions 2 peaks were observed when no sample was aspirated: 525Hz and 640Hz (hereafter referred to as Peak A and Peak B respectively). The zone below 200 Hz seems to give useful general noise level indications, however, the microphone response is not documented below 50Hz and is non-linear below 150Hz. Therefore, any information presented here which falls in this range is qualitative. It should be reiterated that all measurements are using argon gas, and that nebulizer systems which use different sample gases might expect appropriate differences.

Signal Reproducibility

Whether the power spectra one is measuring have any diagnostic utility depends on reproducibility. Therefore, an experiment was undertaken to measure signal reproducibility over a 45 minute period. Another analysis was conducted 22 hours later, with identical operating conditions. The results illustrated that the signal is reproducible if experimental conditions remain invariant.

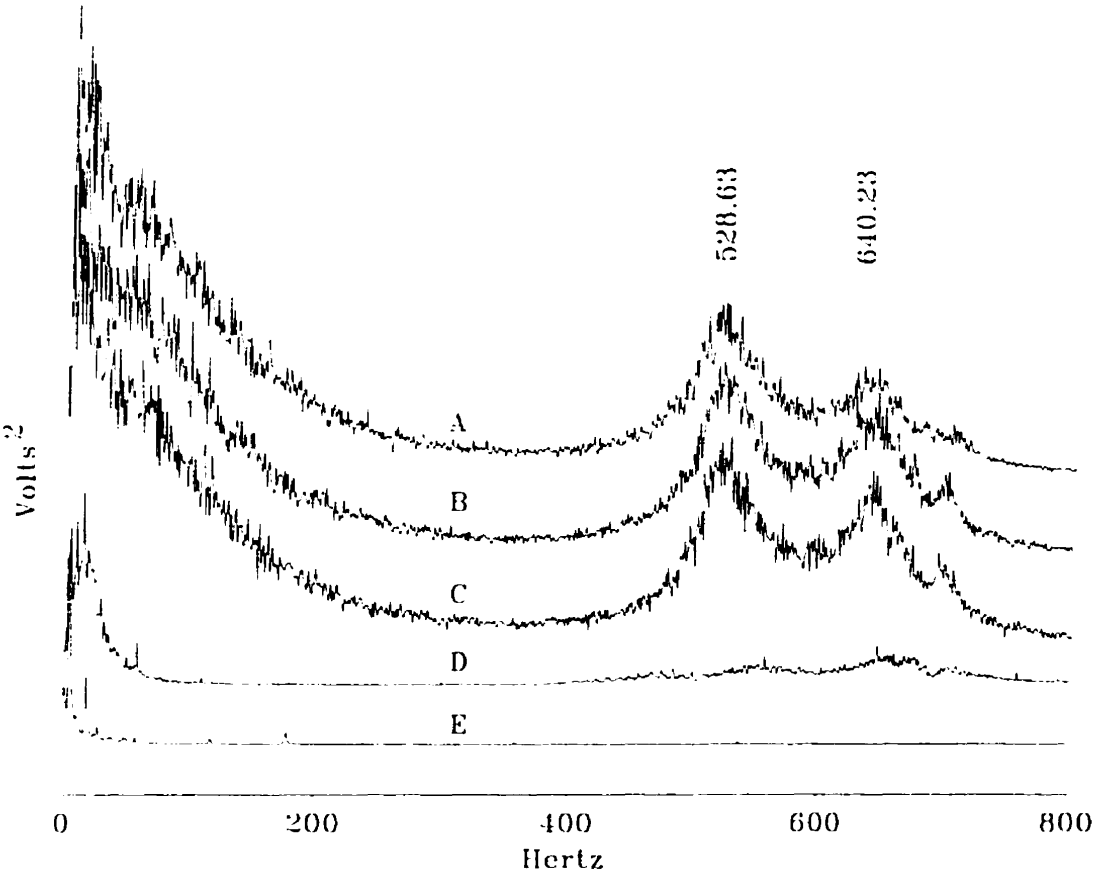


Figure 2. Acoustic power spectra, the spectra are offset for illustration purposes. A) All acoustic signal sources on. 16 l/min Plasma Gas, 0.55 l/min Sample Gas, 0.8 L/min, Auxiliary Gas at 0.8 l/min, 1.25 kW plasma, ventilation connected, no sample aspirating. B) Plasma Off. 16 l/min Plasma Gas, 0.55 l/min Sample Gas, 0.8 L/min, Auxiliary Gas at 0.8 l/min, no sample aspirating. C). Sample Gas only, at 0.55 l/min. D) Plasma Gas only, at 16 l/min. E) All gasses, ventilation, and the plasma shut off.

The Source of the Signal

Figure 2, trace A illustrates the spectral features as they appear typically with no sample aspirating. A number of experiments were run in which the various possible sources of acoustic signal were quantified. Figure 2 illustrates the findings. It was determined that, by a large margin, the greatest source of signal was the Sample Gas (either with or without sample aspirating). The Plasma Gas also made a contribution to the acoustic signal in the 500-750Hz range. This was attributed largely to resonances generated in the torch. The inductively coupled plasma had only a small impact on the signal in the 600-700Hz range as is

illustrated in Figure 2, trace C. This may be because the plasma is physically distant from the microphone, and there is an intervening baffle in the spray chamber. Auxiliary Gas, the RF generator, and the spectrometer exhaust ventilation had negligible influence on the acoustic signals.

When distilled water is introduced, the peak positions are not measurably affected, but the acoustic response appears damped at both Peaks A and B. Other types of sample matrices may shift or reduce the amplitude of these peaks, depending on their physical properties, as will be discussed later.

The nebulizer is acting as an acoustic source. The acoustic waves then encounter the resonant cavities of the spray chamber and the torch, and, from this interaction the majority of the acoustic phenomena are derived. Therefore changes in the geometry of the resonant chambers and changes in the power spectrum of the acoustic driver should be expected to affect the observed acoustic signal.

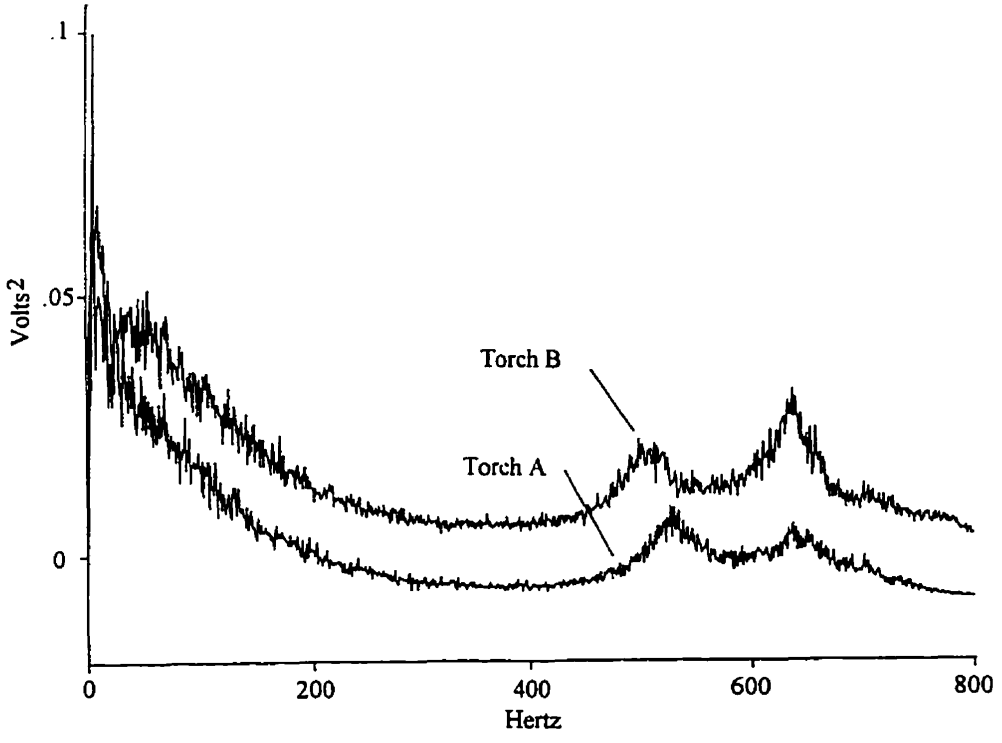


Figure 3. Torch acoustic signal comparison. Operating conditions are identical.

Effect of Torch and Spray Chamber Geometry on Signal

Two torches which varied primarily in injector tip diameter and geometry, and torch length were used. One torch (A) had an inject tube which pinched from about 3 mm to about 1 mm diameter right at the tip, while the other (B) pinched gradually about 1 cm from the tip, and the first torch was about 1 cm taller than the other. Experiments were run to confirm expected changes in spectral response caused by the different geometries. Figure 3 shows that there are indeed substantial differences in peak amplitude for the Peak B and significant differences in position for Peak A.

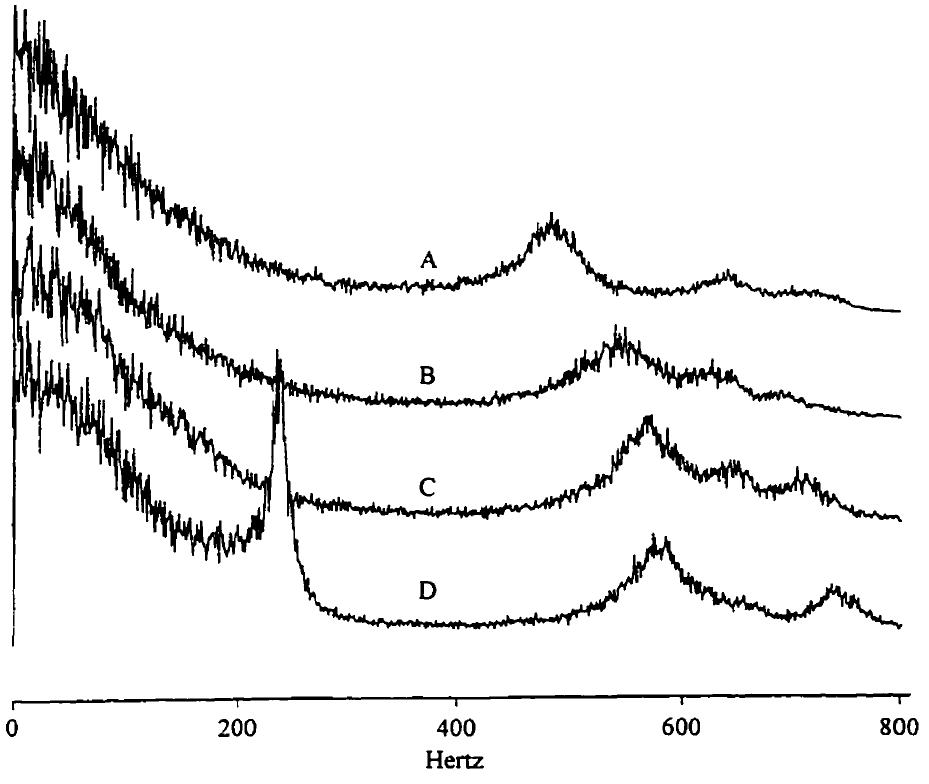


Figure 4. Torch removed. Trace A is a power spectrum of data acquired when a plate with a 1 mm hole was covering the spray chamber exit aperture. For trace B the hole was 1.6 mm, for trace C the hole was 3.2 mm. For trace D no plate covered the spray chamber exit aperture. Note: The feature appearing at about 250 Hz on Trace D is reproducible, but as of yet unexplained.

The torch was then removed, so that the sample gas was expelled from the spray chamber exit aperture, and was run with no sample

aspirating. Figure 4 shows the acoustic responses observed when a fixed plastic plate with a variable diameter hole was fixed over the spray chamber exit aperture. Under these conditions, acoustic resonances shift as a function of the changing spray chamber exit aperture geometry, which causes changes in the acoustic impedance of the system.[95]

In another set of experiments, the torch assembly was left in place, and sample was aspirated with the baffle in several different positions in the spray chamber. The position of the baffle has a great effect on the amplitude of the acoustic emission of the system when water is being aspirated. The intensity of both Peaks A and B appear to be positively correlated with baffle proximity to the nebulizer. Similar results were obtained when no sample was aspirated.

In summary, the geometry of the torch and the spray chamber are important in determining the intrinsic acoustic resonances of the system, but these factors should be invariant for a specific design. Therefore, under constant operating conditions, spectral changes in the output power spectrum will be due to changes in the acoustic emission spectrum of the driver, which in this case is the nebulizer.

Sample Gas Influence on the Signal

Experiments were performed which varied the sample gas flow rate in the absence of sample aspirating to determine the influence of the sample gas flow rate on the acoustic signal. It was found that the position of the various peaks remained approximately constant, but that signal intensity increased with increasing sample gas flow rate.

Effect of Solutes in Aspirated Water on Peak Position

Several phenomena were observed when solutes dissolved in water were aspirated. Generally, low concentrations had no measurable effect, for the solutes we examined. However, when concentrated acids (HNO_3, HCl, H_2SO_4 at 11, 10 and 18 normal respectively) were aspirated it was observed that the low frequency (below 200Hz) noise was higher than with water blanks With H_2SO_4, the low frequency noise was an order of magnitude greater (Figure 5). It was also observed that when acid concentration was reduced to what might reasonably be expected for an

analytical sample, 2-5 N, this very high noise was no longer evident, and there was no difference from the water blank with our instrumentation. Therefore, this methodology appears to have no useful diagnostic capability for determining acid matrix.

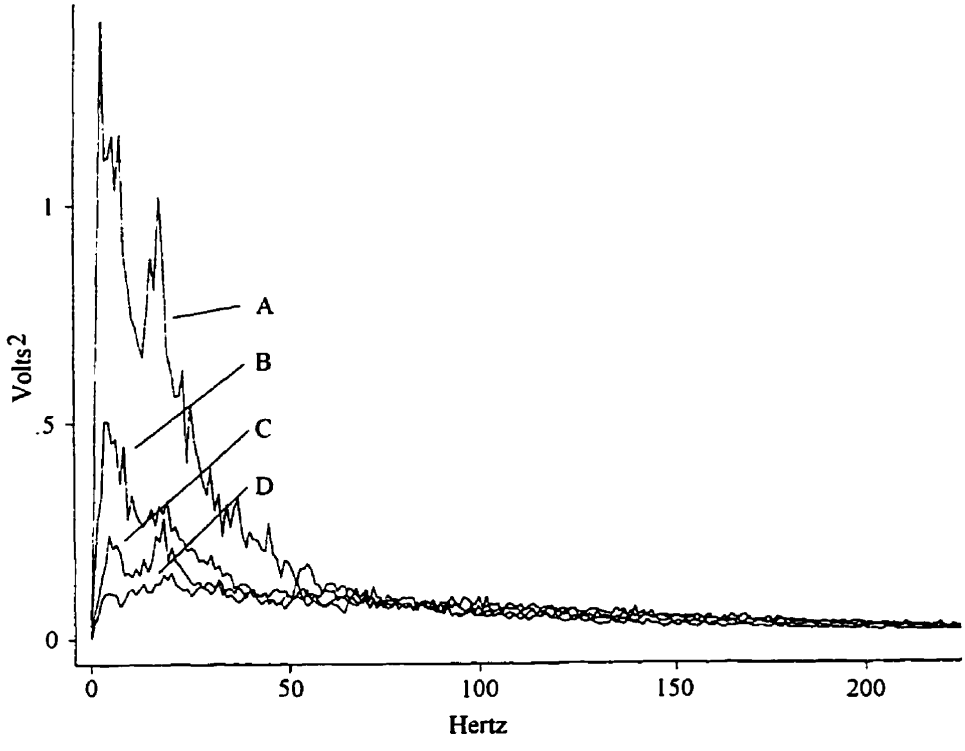


Figure 5. High Acid, high noise. Trace A is 18N H_2SO_4, trace B is 10N HCl , trace C is 11N HNO_3, Trace D is distilled water.

As with acids, moderate concentrations (1000 ppm) of commonly dissolved salts ($ZnSO_4$, $CaCO_3$) showed no obvious deviation from the water blank. However, when more concentrated (10,000-50,000 ppm) solutions were aspirated, Peak A was almost completely damped (Figure 6). It is not unreasonable in geological applications, for example, to have solutions which contain 1% dissolved solids. This technique could provide a useful predictive tool for indicating when high concentrations of dissolved solids are present in a sample. It is conceivable that a semiquantitative evaluation of total dissolved solids might be possible for highly concentrated samples.

Effect of Solvent on Peak Position

When non-aqueous solvents were employed the position of Peak A was always affected, and Peak B was often affected. Initially we attempted to ascribe peak position to a single physical parameter of the solvent. It was quickly demonstrated that viscosity was not an important factor affecting peak position. Peak position was found to be unaffected by running the three aforementioned acids which have highly varying viscosities.

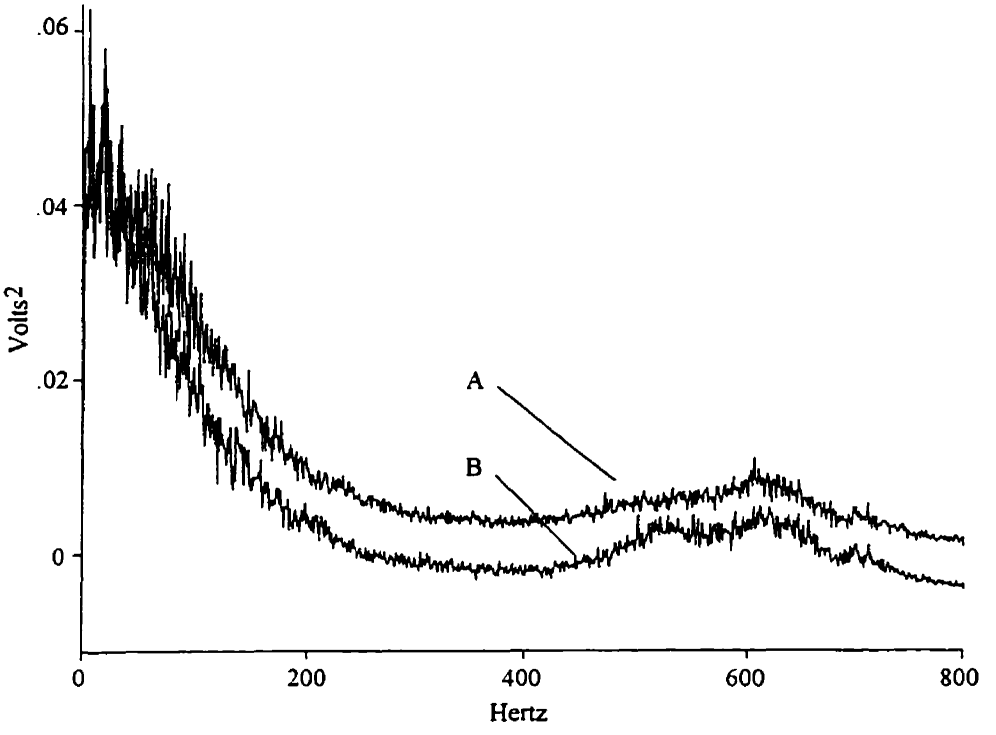


Figure 6. High Salt. Trace A is 10000 ppm $CaCO_3$, and is offset 0.005 volts^2 for clarity, trace B is distilled water.

It was suggested that the saturated aerosol environment present in the spray chamber might be affecting the speed of sound. To test this hypothesis a tuning fork of known resonant frequency was positioned above the top of the torch so that the sonic wave would have to travel through the spray chamber to reach the microphone. If any measurable speed of sound changes occurred, the peak due to the tuning fork should shift under varying conditions in the spray chamber. There was in fact no measurable shift in the tuning fork peak position, as can be seen in Figure 7, therefore spectral shifts are not explainable by a change in the

speed of sound in the medium. This might be expected, because the speed of a 400Hz wave in air varies only about 1/4% between 0% and 100% humidity at 20 degrees Celsius[96], and it might be assumed that other solvents and aerosols would have effects in the same range. This change would be too small to measure reliably with our instrumentation.

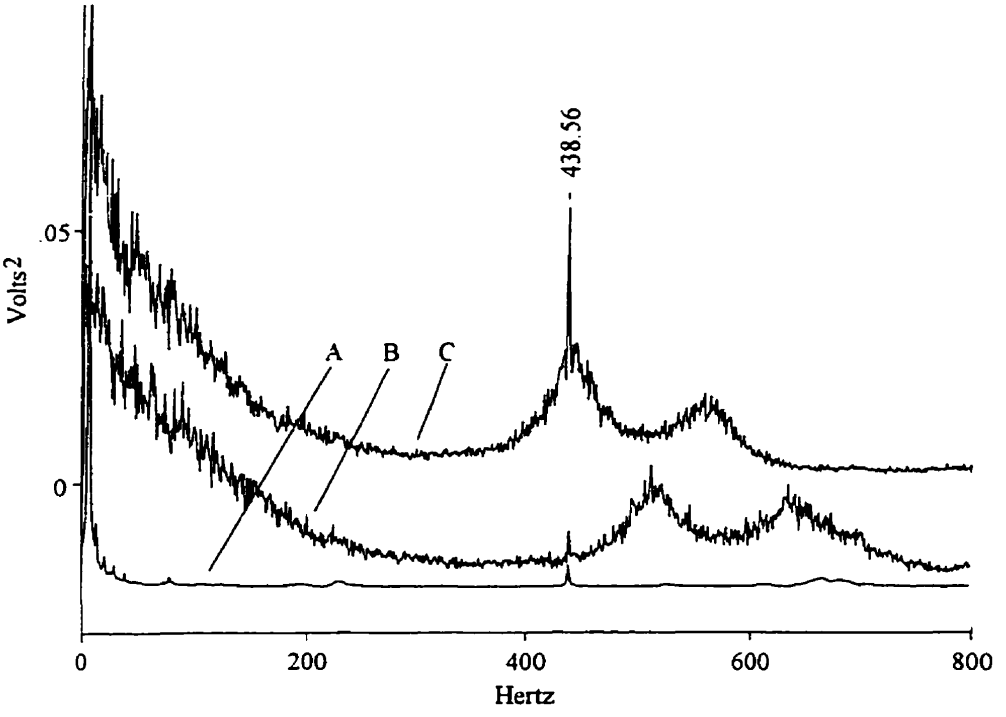


Figure 7. Acoustic signal for tuning fork/speed of sound experiments. Trace A is still air (no gas flow), trace B is no sample aspirating with Sample Gas at 0.6 l/min and Torch Gas 16 l/min, trace C is CCl_4 aspirating (same gas flow rates as B), trace C is offset 0.02 $Volts^2$ for clarity.

It was felt that solvent properties were causing the aerosol particle size distribution to change, and this was in turn affecting the observed acoustic emission spectrum. Successive dilutions of water with methanol were then attempted, and later diethyl ether, hexane, carbon tetrachloride, acetone and ethanol were examined in an attempt to determine the correlation between peak position and surface tension (Figure 8). Later, an attempt was made to correlate peak position with the Evaporation Factor defined by Boorn and Browner,[97] which is a measure of the physical properties which influence solvent evaporation and takes into account such factors as diffusion coefficient of the solvent vapor, temperature, saturated vapor pressure, surface tension and

density. We can say that there is a rough linear relationship between evaporation factor and peak position (slope -0.12, with a correlation coefficient of 0.58). CCl_4 does not fit into this attempt to explain peak position well. One could compute a polynomial fit between surface tension and peak position. CCl_4 fits poorly into this model as well.

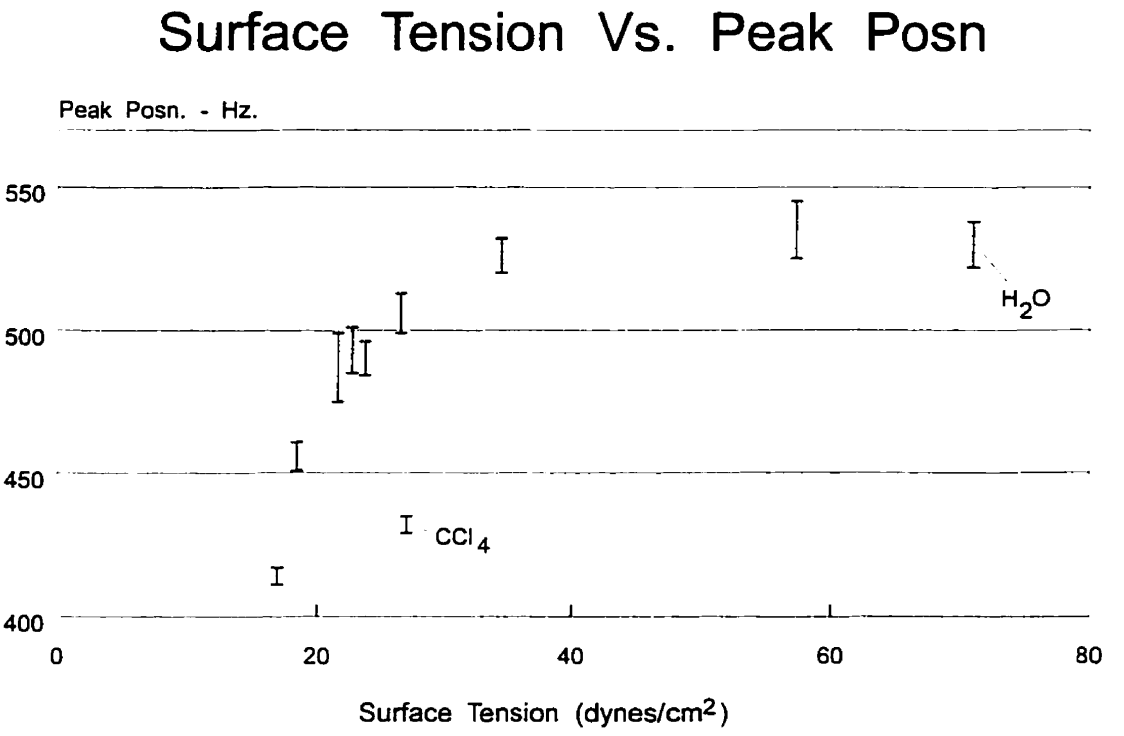


Figure 8. Surface Tension vs. Peak Position.

Despite the fact that neither of these two models explains the data quantitatively, they may perhaps yield information which would enable qualitative analysis of the acoustic signature of a solvent and make some predictions about its properties. Certainly, the data indicate that the solvent itself might be identified if it had been previously cataloged, although solvent mixtures might make this more difficult.

Conclusions:

The acoustic emission from the nebulizer/spray chamber assembly produces a spectrum which is affected by factors which influence droplet size. The emission peaks appear to be resonances associated with the spray chamber and the torch.

Any measurements in a spray chamber assembly of this type must be conducted in an environment where the chamber and torch geometries are invariant. This however should not be a serious constraint, as this will almost always be the case in a commercial spectrometer.

One might reasonably hope to have some information about the sample matrix before one has even begun to acquire spectroscopic data. This could allow one to take preventative measures should the sample require special procedures to generate meaningful analytical results. If the signal were acquired using a digital signal processor it is reasonable to assume that this type of acoustic analysis could be ongoing during the time that atomic spectroscopists usually allow for spray chamber clearout, 45 to 200 seconds, depending on the application and hardware being used[98,99]. With a digital signal processor it should be possible to make a good qualitative evaluation of the sample in 5-10 seconds.

This technique could be used to monitor the operation of the nebulizer, so that a quick diagnosis could be arrived at in the event that the nebulizer begins to clog, allowing remedial action to be taken. It could also give a qualitative evaluation of the nebulization process itself, as was observed with very high acid samples, high acoustic noise might be a warning that the sample is nebulizing poorly. One might also use this technique to identify organic solvents, once again allowing a timely change in instrument operating conditions, as appropriate.

As the effect of the plasma has relatively little influence on the measured acoustic signal in this hardware configuration, it is reasonable to assume that this type of technique would be applicable to other systems that use nebulization for sample introduction.

REFERENCES:

[85] Wade, A.P.; Sibbald,D.B.; Bailey, M.N.; Belchamber, R.M.; Bittman, S.; McLean, J.A.; Wentzell, P.D., Anal. Chem., **63**, p 497A, 1991.

[86] Belchamber, R.M. and Horlick, G., Spectrochim. Acta, **37B**, p 17, 1982.

[87] Dittrich, K.; Niebergall, K.; Wennrich R., Fres. Z. Anal. Chem., **328**, p 330, 1987.

[88] Chen, G. and Yeung, E.S., Anal. Chem., **60**, p 2258, 1988.

[89] Pang, H. and Yeung, E.S., Anal. Chem., **61**, p 2546, 1989.

[90] Pang, H.; Wiederin, D.R.; Houk, R.S.; Yeung, E.S., Anal. Chem., **63**, p 390, 1989.

[91] Davies, J. and Snook, R.D., J. Anal. At. Spectrom., **2**, p 27 (1987).

[92] Easley, S.F.; Monnig, C.A.; Hieftje, G.M., Appl. Spectrosc., **45**, p 1368 (1991).

[93] Browner, R.F.; Boorn, A.W., *Anal. Chem.*, **56**, p 787A (1984).

[94] Horowitz, P. and Hill, W., *The Art of Electronics*, (Cambridge University Press, New York, 1986).

[95] Merhaut, J., *Theory of Electroacoustics*, (McGraw-Hill International Book Company, Toronto, 1981).

[96] Evans, L.B. and Bass, H.E., *Tables of Absorbtion and Velocity of Sound in Still Air at 68F (20C)*, AD-738 576, available from the National Technical Information Service, U.S. Department of Commerce.

[97] Boorn, A.W. and Browner, R.F., Anal. Chem., **54**, p 1402, 1982.

[98] Dobb, D.E. and Jenke D.R., Appl. Spectrosc., **37**, p 380, 1983.

[99] Wohlers, C.C., ICP Inf. Newsl., **3**, p 37, 1977.

CHAPTER 5

We've now performed some preliminary studies to determine to what ICP emission instruments our work might apply, what mechanism might be used to select spectral wavelengths that our system should analyze to get the maximum amount of useful spectral information in the minimum time. We've also explored a novel approach to obtaining diagnostic information about the sample introduction process.

The next step involved in the development of the autonomous instrument is to set out the goals for our system, and determine what subsystems will be needed to meet these goals. Once the required subsystems have been identified a prototype framework should be developed to house the subsystems, so that assembly of a working prototype can begin.

The manuscript was published as:

THE AUTONOMOUS INSTRUMENT:
CONCEPT TO PROTOTYPE

Douglas P. Webb and Eric D. Salin
Department of Chemistry
McGill University
801 Sherbrooke St. W.
Montreal, P.Q., Canada
(514) 398-6231

***Intelligent Instruments & Computers*, Vol 9, Sept/Oct 1991, pp. 185-193.**

INTRODUCTION

While some of us view with amazement the introduction of microprocessors into everything from pipettes to mass spectrometers, it seems undeniable, laboratory instrumentation is becoming automated. It would seem obvious that the next step would be the introduction of intelligence into this modern automated instrumentation. A modern production laboratory usually consists of a powerful ensemble of instruments, computers and software. The instruments are often networked to a LIM (Laboratory Information Management) system so that all the tests done on a sample can be combined into a single report. A human may evaluate the collated data or a report may simply be generated automatically. Given this high degree of automation, one might wonder what more could reasonably be needed.

To appreciate the need for "intelligence" in instruments or a LIM, one need only take a closer look at the laboratory from the human side. Automated instruments often generate data at a rate which far exceeds the human bandwidth. In addition, the sheer mass of data may be too large to be grasped by a human under any circumstances. Despite the current level of instrument automation, it still takes a long time to train technologists to operate instruments. The training time of data analysts can vary, but may be even longer. To make matters worse, manpower projections indicate shortages of technically trained people starting in the mid 1990s. Compound the manpower shortages with budgets that seem to be shrinking, and we have a situation which cries out for the next logical step in instrument design, autonomous intelligent instruments.

While one can certainly see the need for an intelligent program monitoring LIMs data, that is a post analysis processing problem which we will not discuss here. Instead, we will examine the problem of the instrument itself. We have attempted to address this by studying what might be considered a typical problem, inductively coupled plasma (ICP) atomic emission spectrometry. The instrumentation in this area is in many ways typical of that in the analytical laboratory today. The ICP can be run automatically with a menu driven program from a small computer

and can be connected to an autosampler. This allows large numbers of samples to be analyzed unattended, and reports can be generated automatically. What's wrong? Nothing! It simply doesn't go far enough. There are several more things the instrument should do: 1) It should monitor the samples being analyzed in real time to determine if the analysis is likely to be invalidated due to the combination of the operating conditions and the sample type. If possible, it should take corrective action, and 2), it should determine if the sample is likely to compromise the instrument's ability to continue functioning properly. The first of these requirements is, by far, the most interesting. It implies that the instrument should automatically adjust itself to provide both accurate and precise results. This goes far beyond simply getting numbers from the instrument.

THE GENERIC VIEW

From the software design and the analyst's point of view, it is interesting to look for features of expert human operators that might be generic and consequently, when developed in software, allow that software to be used by various instrument families (Table 1).

Recognition of samples
Recognition of situations
Self optimization
Choose methodology
Learn

Table 1: Generic Instrument Goals

Some of these deserve elaboration. To keep a focus, we will use examples from ICP-AES.

If a sample is recognized to be of a certain type (*e.g.* geological), that may dictate that the instrument's operating parameters must be changed for optimal performance. A broader scope for the recognition of samples might extend to the answering of questions such as "Is it poisonous?" However, these can be answered by an intelligent LIMS system if the data from the instrument is accurate and precise.

The real time recognition of a situation is somewhat more subtle, because it is not always clear what the measured parameters will be. Consider a case in which an operator introduces a sample from a batch into the plasma and notes a bright yellow tail flame. Examination of the client data reminds the operator that this is plant effluent data collected from a coastal area in which the tides often cause sea water to be mixed with the water effluent. The operator then remembers that the analyte detection thresholds required by the client are high. The last time that this situation was encountered the sample was diluted by a factor of 100. Here the situation recognition parameters would be the symbolic (rather than numeric) plasma description "bright yellow tail flame" along with

the client name and sample collection point. Situation recognition could save the repetition of hours of work previously done.

Self optimization is clearly a worth goal and needs no further elaboration at this point. We will present an example below of how a methodology might be selected. The goal of learning is somewhat all encompassing. It describes what one generally expects of an intelligent entity, the ability to improve on the receipt of new information (unsupervised learning) or after instruction (supervised learning). Rather than programming situations, optimizations and sample type characteristics, we would prefer that the system learn these by itself. In the case of numeric data, this does not seem unreasonable, as we will demonstrate below. Means and standard deviations, with which we are all familiar, can serve as a basis for description. Even raw symbolic data such as that mentioned above can be conveniently processed to provide rules for intelligent behavior [100].

Keeping in mind the lofty goals of Table 1, our laboratory has embarked on a project to develop an autonomous intelligent instrument. We have selected the field of atomic emission spectrometry for several reasons. The first consideration was the "Goldilocks" test [101]. "Was it too hard, too easy, or just right?" When viewed as a whole, the task appears terribly complex, but viewed as an assembly of subproblems (see Figure 1) to be solved by different experts, it seems reasonable. The second consideration , the telephone test [101], is also applicable. In other words, could most problems be solved by an expert over the telephone in less than half an hour? If not then perhaps the problem is too difficult for an expert system to tackle. There is also the "stone soup" [101] effect to consider. Studying a decision making methodology can often result in a clarification and simplification of the entire process, yielding a better human understanding.

EXPERT SYSTEMS

One of our primary tools is the "expert system". Expert systems are programs which attempt to produce decisions or actions which are similar to those provided by human experts. The basic concepts of expert systems have been discussed previously by Deming [101], so we will provide only a limited background in the basics here. While expert

systems have not yet been highly visible in the public eye, they are being used extensively in the business world. A great deal of expert system development work is being done in the process control area, and the technology is now moving into the science domain for knowledge intensive problem solving [102].

The construction of expert systems has been drastically simplified by pioneer work in the Artificial Intelligence community. This work has led to the discovery that knowledge, usually facts and rules, can be conveniently separated from the program that uses the knowledge. This separation has led to the development of Expert System Shells which provide the knowledge manipulation program (usually called an Inference Engine) which manages knowledge presented in a standard format. Expert system shells come in a variety of flavors, but the most common is the Rule Based System. These shells make it easy for an expert (even a non programmer) to read, understand and modify the knowledgebase. Since Deming[101] has discussed these in some depth, we will only do one example here which is specific to our work.

There are some important characteristics of expert systems which need to be mentioned. First, expert systems lack the broad base of knowledge and relationships which we call "common sense". This is a general characteristic of computer programs, but has special implications for expert systems as we shall see: expert systems exhibit expertise only in a narrow domain. What this means for practical purposes is that if an expert system encounters conditions which it is not designed to deal with, results may be unpredictable unless great care is taken when developing the system.

A second characteristic common to many expert systems is an 'explain' facility. This facility can be used to explain why a question is being asked or why a given recommendation was made. This can provide, in essence, a trace of the decision making process. While this allows convenient debugging of the system, its most important use may be in educating users of the system.

Many expert systems provide an easy method to modify both the rules and the knowledgebase. In some cases, this feature may be valuable. In other situations, it will be critical that users be "locked out" from this facility to preserve the integrity of the system. Some expert

system shells provide "truth maintenance" facilities which can act as a safeguard to prevent the insertion of "illogical" or contradictory rules. In any event, modification of the knowledge base must be done with some care in any large expert system.

Expert systems are not general purpose tools, as indeed, many of our tools ranging from data base managers to graphics packages are not general purpose. It has been suggested by Ester Dyson, a noted AI industry analyst, that many of the modern large software systems can be viewed as "raisin bread", in which the AI (artificial intelligence) components will be the raisins and traditional programming methodologies will be the bread. We will return to this analogy later in our discussion.

EXPECTATIONS

When evaluating autonomous or "expert" instruments it is useful to consider the advantages that might accrue from various viewpoints. From the manager's viewpoint, what are the goals or benefits of an autonomous system?

Reduce training costs

An autonomous instrument should be able to run with an absolute minimum of human intervention. Moreover, by using the Explain facilities of the expert system, an operator should be able to learn the logical processes used in the various subsystems. Eventually, the operator should understand the "thought process" sufficiently well to extend it for special situations.

Lower skill manpower

Since the instrument requires very little sophistication from it's operator, it should be possible to use operators with a lower level of technical skill than is presently required. The instrument should be sufficiently versed in diagnosing it's own problems that it can isolate them and lead the operator through a correction procedure. The diagnosis and correction phase may take advantage of graphical displays to aid the operator in locating and correcting the fault.

Less Instrument Down Time

The autonomous instrument could monitor it's own raw material consumption relative to existing supplies, thereby insuring a minimum of human like "Oops! I forgot to order the X" mistakes. Preventative maintenance will be scheduled well in advance at a time convenient to both the human operator and the work schedule. The self diagnosis capability, possibly linked to the manufacturer by modem (computer telephone communication), will allow minor breakdowns to be resolved quickly. Major breakdowns will be efficiently diagnosed and repair personnel will arrive with the appropriate parts.

Full Utilization

Since the autonomous instrument will perform a series of self checks and sample checks, it can be run without constant attendance by human operators. If instrumental "life threatening" situations occur, the instrument will shut itself down. If unresolvable analytical problems occur, the data file will be flagged and the instrument will move on to the next sample. Knowing that the instrument will not destroy itself, the manager can allow the instrument to operate on a 24 hour basis.

The primary advantage for a technically skilled operator is a relief from tedium. An operator is called only to solve problems which are beyond the capacity of the expert system(s). This allows better utilization of the operator's time and the work place should become much more interesting.

The field of artificial intelligence/expert systems has moved away from the study of "toy" problems into the real world. Evidence for this can be seen in the enormous growth of AI research in the area of robotics as researchers seek the fires of reality to temper their tools. Rather than simulating a plasma spectrometer, we are following the general trend in AI and starting with a real system. The prime characteristic of the instrument must be a high degree of automation. We are presently using a Thermo Jarrell Ash Model 25 rapid scan spectrometer which allows rapid access to spectral lines as well as automated control of operating parameters.

SOFTWARE DESIGN

It is interesting to follow the evolution of our concept of the intelligent instrument through several years, because it reflects a maturation in our appreciation of the nature of the problem.

Our original design concept is illustrated in Figure 1. We began by building the system in a traditional "bottom up" approach. We anticipated that a family of expert systems/procedural problem solving modules were needed. This was in recognition of one of the known limitations of expert systems: they must be narrow in scope. We felt that we could better craft the "higher" levels of intelligence, the Preliminary Diagnosis and Data Evaluation Modules after testing the lower level modules. The configuration presented in Figure 1 is typical of the "procedural" approach to programming which has been popular for many years. Higher level procedures concentrate on a more global view of the problem, using lower level subroutines to handle specific subproblems, these subroutines may in turn call still lower level subroutines to handle even more specific subproblems. Each of these main modules are discussed briefly below.

It's interesting to note that most of the modules match a specific goal for the generic system, *i.e.*, a module was developed to handle the task outlined in each box in Figure 1. For the first prototype, each module was constructed using relatively simplistic rules or mathematical techniques with the expectation that they would be refined after an evaluation phase indicated where effort was needed. This iterative approach to expert system development is common since it is often difficult to predetermine where a high degree of "intelligence" will be needed.

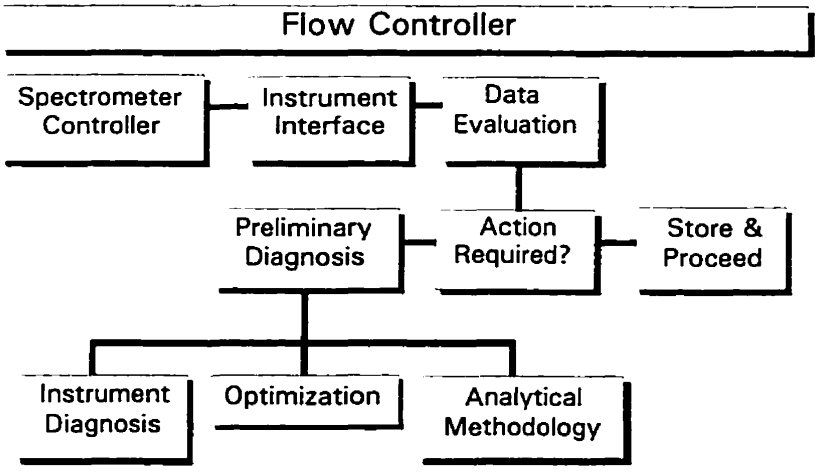


Figure 1 Intelligent Instrument expert system family.

The **Instrument Interface** was the first module to be developed. It was essential because it allowed the other modules to acquire data from the instrument. It was designed to take the equivalent of generic type commands like "Get the Calcium concentration from the 422.7 line" and translate them into instrument specific code. The only expert system developed for this module was LINEX [103], a line selection expert system. As a practical compromise between speed and software development time during this prototyping stage, the Instrument Interface works by generating "command" files which are then read by a modified version of the manufacturer's software specifically developed for these experiments. The manufacturer's software (TSPECSA) and our software (AJAX) live under a multitasking operating system called DESKview and communicate via files and a small program called REMOTE as illustrated in Figure 2. It is interesting to note that this module and the Diagnosis modules are the only two modules which would not be generally applicable to ICP scanning spectrometers.

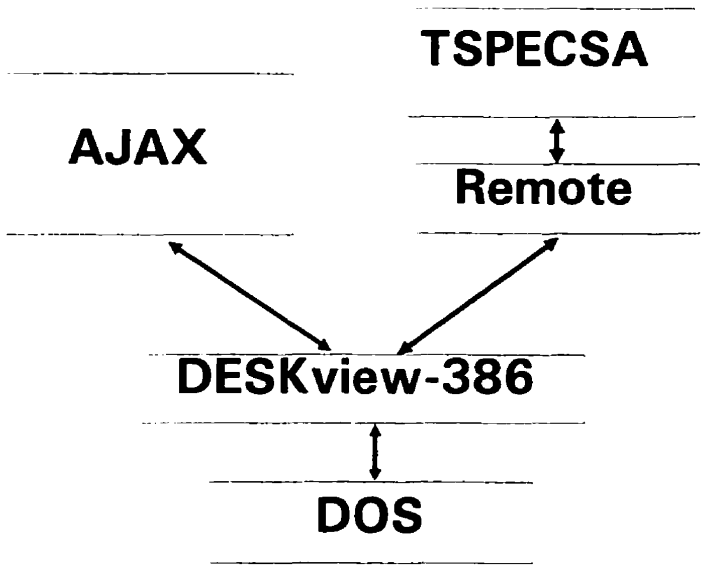


Figure 2. DESKview-386: a solution to the DOS problem.

The **Data Evaluation Module** was designed to perform two functions: (1) check the Signal-to-Noise ratio and TSPECSA supplied diagnostic codes to make certain that the instrument was running properly and to (2) recognize samples. In keeping with the simplistic theme of the first prototype, the Range method of sample recognition was developed (Figure 3). A training set was used to provide a mean and standard deviation for each element. A range was designated to be 3 standard deviations on either side of the mean for that element in the training set. This methodology has a number of advantages. Calculation is fast and requires minimal storage space. This is an acceptable assumption for a prototype and some quality control situations where the number of types of samples to be analyzed may be small, but in a general analysis laboratory this would not necessarily be true. This is a unsophisticated technique who's most serious flaw, obvious even at the prototyping stage, is its inability to place a measure of goodness on "fit" between a sample and the sets to which it may belong. A sample type with many elements may then actually form a super set of the elements required for membership to multiple other classes thereby providing a situation in which a sample can belong to multiple classes. It also performs poorly with overlapping classes.

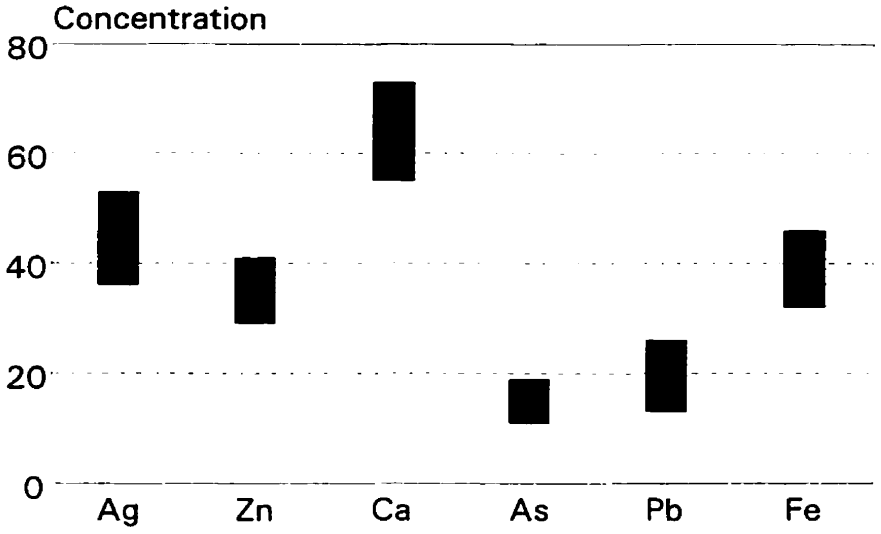


Figure 3. Simple pattern recognition: range method

If the **Sample Recognition** component of the **Data Evaluation Module** does not require action and the Signal-to-Noise is satisfactory, then the data would be stored. This would be the most common occurrence. If, however, there appeared to be a problem, control would pass to the **Preliminary Diagnosis** module which would decide whether the problem should be handled by an "**Instrument Diagnosis**" or "Optimization" module. The **Optimization Module** would then decide whether a methodological solution or an adjustment of instrument parameters was most appropriate.

A modified simplex[104] approach was selected as an appropriate starting point for optimization. This technique offers simplicity but suffers from a number of limitations. Perhaps the most important of these is the lack of a learning facility. A simplex always starts without having learned anything from previous simplexes. A second weakness is that the simplex procedure does not provide any indication of the importance of the various parameters used. This suggests that a second method like principle components analysis could be used to try and determine the important parameters so that the optimization process itself can be optimized.

The problem of methodological selection is best expressed in a symbolic format. This led us to a rule based approach for this section. A typical rule based system separates the knowledge (rules) from the way

the knowledge is processed (by the inference engine). As the reader can see below this leads to a very readable format, an important advantage of rule based systems.

A typical methodological rule set might contain the following:

IF sum of concentrations > 10,000 ppm OR
(Na OR K OR Mg OR Ca) > 3,000 ppm
THEN condition is HIGH SALT

IF condition is HIGH SALT AND
FOR each element of analytical interest
the ELEMENT CONCENTRATION > 20 * LIMIT OF DETECTION
THEN dilute

The dilution factor can then be calculated from the estimated concentration of the element closest to its detection limit and a suitable limit of quantitation.

The **Instrument Diagnosis Module** was included primarily for the sake of completeness. It is one system which can probably be best developed by a manufacturer. A manufacturer is most likely to have the information in house which can optimize a diagnosis/repair process. In fact, modern instruments do not seem to actually "break" often, but may have an Achilles' Heel. For inductively coupled plasma spectrometry, the Achilles' Heel is often the sample introduction system[105]. Nebulizers are a particularly weak point. With certain nebulizers, clogging can occur, thereby completely disabling the system. For this reason, we developed the specific portion of the **Instrument Diagnosis Expert System** which would detect failure in the sample introduction system. It was work in this area and the **Methodology Selection Expert** which led to a major change in our approach.

A QUESTION OF INFORMATION

Our perception of how an instrument control system should operate was influenced strongly by a traditional perspective on expert systems, programs that act like experts and talk to humans. Our original vision is illustrated in Figures 4 and 5. We had envisioned a scheme where an expert system would replace the operator or act as a buffer between the operator and the spectrometer. To replace the operator, the expert system would do the same types of things which the operator normally does. In fact, an autonomous instrument should operate quite differently. It's really a matter of perspective. Figure 6 more correctly expresses how one must view the problem. From the programming perspective, the expert system should not try to do the things in the same fashion that a human operator does, because it does not have the same sensory inputs. The expert system should use the sensory inputs it has (large amounts of spectral information) to its advantage.

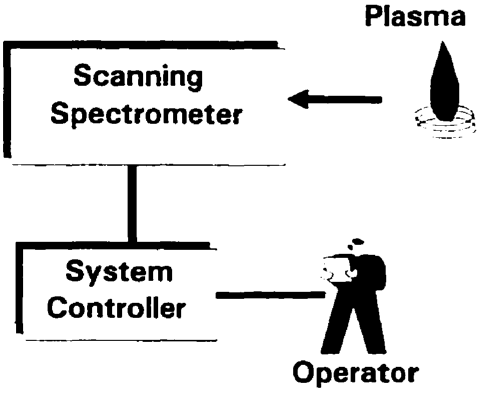


Figure 4

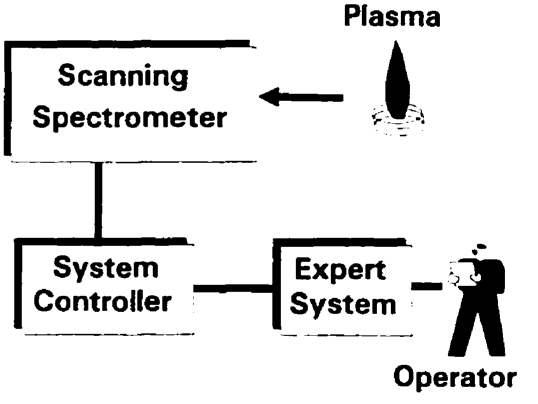


Figure 5

Our work on the Methodology and Instrument Diagnosis Modules had provided the clues. Where was the information coming from to tell the expert system that the sum of the concentrations was greater than 10,000 ppm? Where was the information coming from that would tell the expert system that the nebulizer was clogged? If a human operator were

running the instrument, a high salt condition might be obvious from a brighter plasma. Similarly, a blocked nebulizer might be noted any number of ways by a human operator, most of them visual.

It's axiomatic that intelligent decisions need information. While there is an initial temptation to attempt to use sensors to mimic human capabilities by using acoustic and imaging sensors, as might be suggested by the scheme proposed in Figures 4 and 5, this temptation should be avoided by thinking as we have illustrated in Figure 6, from inside the instrument. Additional senses should be added only if the existing senses cannot provide the necessary information.

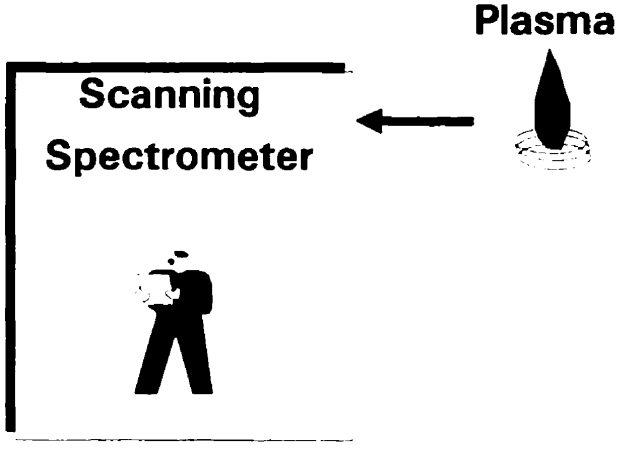


Figure 6

The need to substitute for human senses led to the first major change from conventional unattended plasma spectrometer operating procedures, the institution of a Preliminary Scan. The Preliminary Scan, in it's most simplistic form, looks rapidly through the spectrum for a major line of all the common elements. This feature is available in many spectrometers as a "semiquantitative scan." In addition to general element concentration information, two other pieces of information are sought during the Preliminary Scan. These are the intensities of the Hydrogen Beta line (486.1 nm) and the carbon line (193.1 nm). The carbon line can indicate the presence of organics, which may require alteration of the plasma operating conditions. The Hydrogen line may have multiple uses: high concentrations of easily ionizable elements, and blocked (or partially blocked) nebulizers can alter the character of this line, as Figure 7 indicates. It can serve as an indicator of nebulizer

blockage and may also serve to warn of changes in excitation conditions[106]. The information is used automatically to determine whether special operating conditions are required for the rest of the analysis cycle for that sample. A simplistic presentation of the use of Preliminary Scan information is provided in Figure 8. The two obvious prices of a Preliminary Scan are time and sample consumption, and this bring us to our next major change.

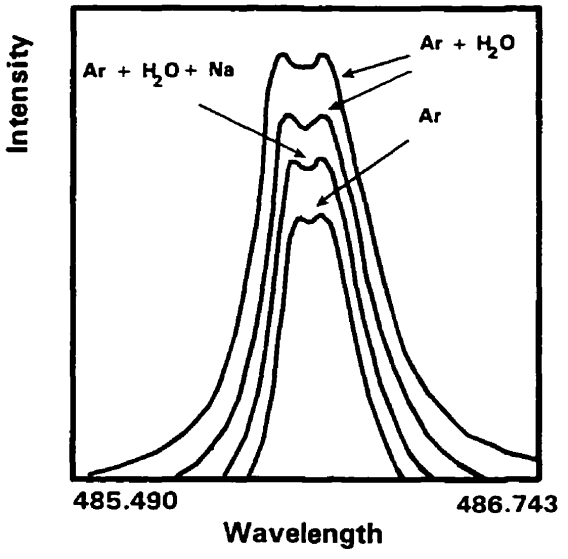


Figure 7 The Hydrogen β Line

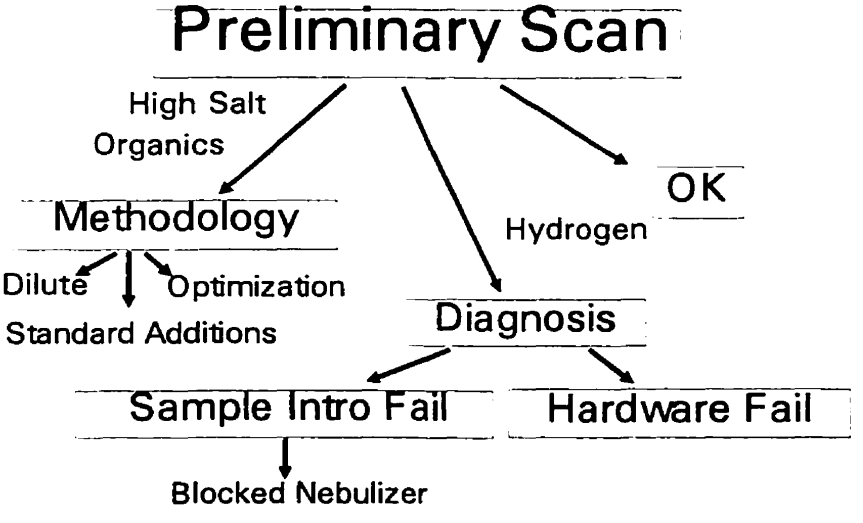


Figure 8 Preliminary scan and where it may lead.

A CHANGE IN STRUCTURE

As we worked with Preliminary Scan data, it became clear that several radical changes were necessary. We realized that the traditional procedural structure was insupportable. There were simply too many possibilities to program realistically in that manner. Maximum flexibility in the order of utilization of the analytical problem solving resources (optimization, sample methodology selection, instrument hardware diagnosis, *etc.*) is required. Any order of resource utilization should be possible, and it should ultimately be possible to examine methodological options, for example, repeatedly if necessary. This type of flexibility is difficult to implement using traditional procedural programming. Declarative programming and object oriented programming are better suited to solving this type of problem. The second change necessary was the demand that the operator provide certain basic information. This is the type of information which would be obvious to the operator, but not to the machine through it's senses. The absolute minimum information consists of the type of analysis to be run: quality control (Q.C.), unknown sample or reference material.

Figure 9 shows quite a different top level of the program than that indicated in Figure 1. Now the initial routing is determined by the type of analysis. A Q.C. sample is analyzed without wasted time or sample as illustrated in Figure 10, only a deviation from specifications causes the system's intelligence to come into play. Even then, as long as the instrument is operating properly, little needs to be done.

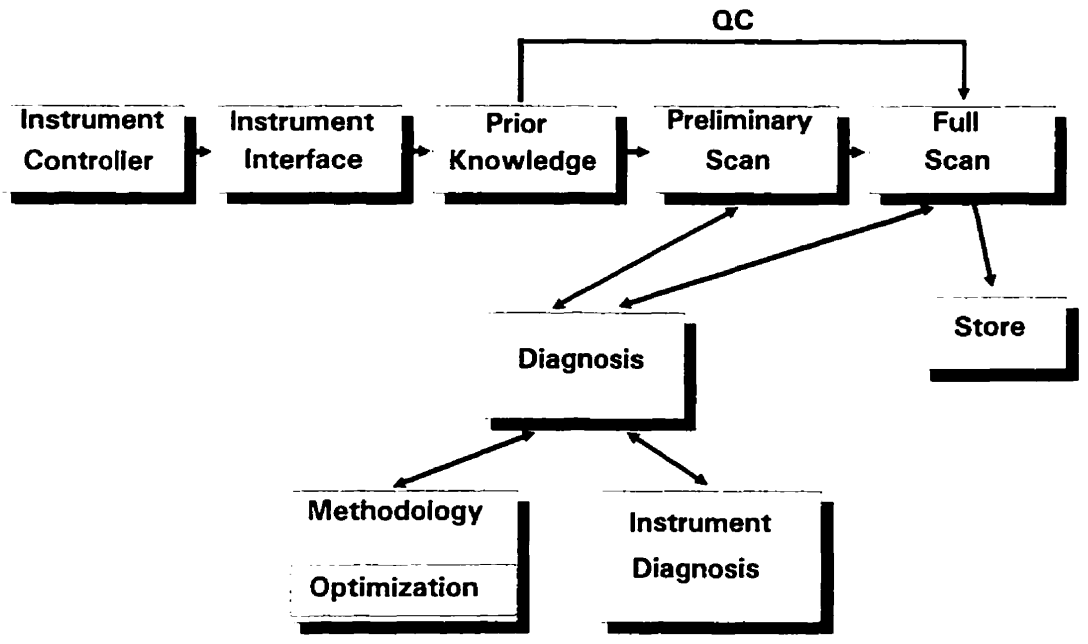


Figure 9 Top level view of the program.

If an unknown material is to be analyzed, then a Preliminary Scan may be run as suggested by Figure 10. Now numerous possibilities arise (see Figure 11). If there is a high salt level, then nebulizer clogging may be a concern. Dilution may be a remedy if concentrations are high. Even if a high solids nebulizer is present, dilution may be desirable. If concentrations are low, dilution is not

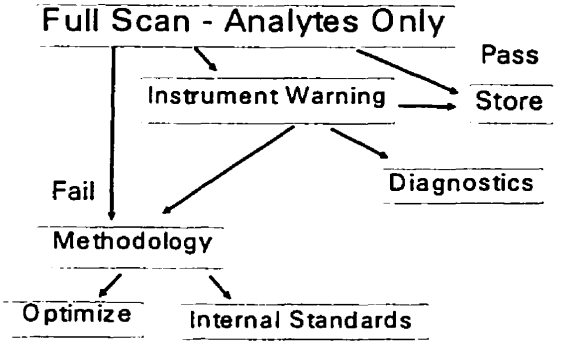


Figure 10. Routing

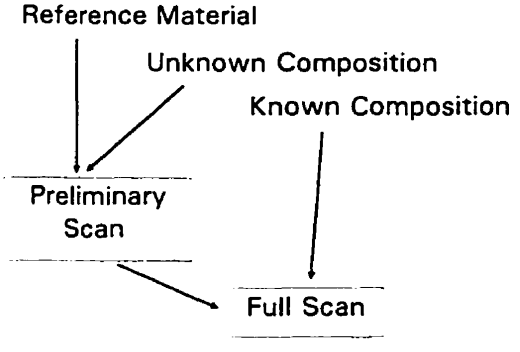


Figure 11. Unknown Composition

a viable alternative. Then the expert system must consider the relative merits of parameter optimization, perhaps increasing the forward power,

or using a methodological cure like standard additions, which can have its own problems. The solution appears to be expressed best in a series of IFs, ANDs, ORs, and BUTs, in other words production rules. The situation can become further complicated when one considers the possibility of having several conditions appear simultaneously, for example HIGH SALT and carbon. What then? Once again, this is something that has already been addressed in the AI community under the general heading of fuzzy variables[101,107]. Fuzzy variables are generally used to express a degree of confidence in a given input variable or conclusion. For example to the question, IS THIS A LAKE WATER SAMPLE? An operator might reply YES 60%. This could then affect the weighting (confidence) of any conclusions based on that piece of information. Similar fuzzy methodology can be used to make choices in cases where inputs could cause competing branches at decision points.

Overall one must say that the situation is far too complex to describe completely in a conventional procedural manner. Consequently, we have lost our ability to express the nature of the programs adequately using the simple flow charts that we employed throughout this paper. These diagrams continue to be valuable, because they are a simplification, but one continues to encounter the "oh! But what if . . ." situations that reality mandates.

LEARNING

An important characteristic of the generic expert instrument, Table 1, is the ability to learn. But, learn what? It would be especially valuable if the instrument could learn how to run a given type of material. Normally a reference material of known composition, is used and methodologies or instrument parameters are adjusted so that calibration with standards provides the correct concentrations for the reference material. In this case the learning procedure **appears** to be relatively easy. The optimization module can minimize the difference between the reference material and standards by using either simplex procedures to adjust instrument parameters or testing methodological solutions. Here we find that the optimization module must employ a combination of rules and an algorithm. We discovered this when the system took the expedient measure of simply cutting off the sample introduction pump, thereby reducing all signals to zero and satisfying the criteria of finding conditions in which the reference materials and standards produced the same signal. It is also important to appreciate that a combination of methodologies and instrument optimization is not only possible but may, in some cases, be essential. Figure 12 illustrates the approximate thought pattern that would be used but does not indicate the cycling between the options that may be necessary nor does it hint at some of the fuzzy decisions that might be called for.

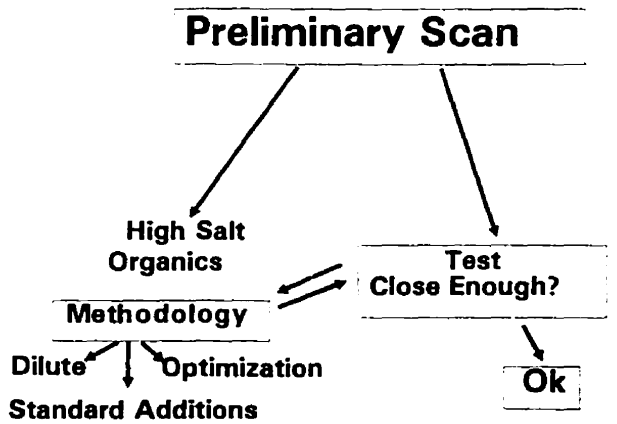


Figure 12 Though process for learning how to analyze an unknown sample.

TOOLS

Finally as a result of our attempts to write almost all of the system using procedural programming in the 'C' language and the difficulties (described above) that arose, we are more than ever convinced that no one programming tool or methodology will easily solve all of our problems. The raisin bread approach will have to be embraced as much as possible. Find the tool that solves each problem best and link the resulting modules together in a way that allows the various modules to interact with each other as easily as possible. If traditional procedural programming solves that problem (as it seems to for optimization) then use it, but don't try to force a square peg into a round hole. If object oriented or declarative programming is called for, use the correct tool. Similarly, Deming[101] propounded a gem of wisdom which bears repeating: "Don't underestimate the amount of time you can spend (or waste) on expert systems". This leads to the not surprising fact that anyone trying to tackle a problem of the type described here, *i.e.*, a series of small integrated subproblems, should become familiar with as broad an array of programming tools and methodologies as possible, so as to select the optimal tool for each subproblem.

THE NEXT STEP

Now that we have developed the programming strategy necessary to assemble the autonomous instrument system, we are looking forward to enhancing the simplistic approaches used to solve some of the various module's tasks. We are planning to study the possibility of improving the existing rudimentary, simplex based, optimization system. Some sort of system whereby information gained in earlier optimizations on the same or a similar response surface could be used to speed subsequent searches for optimal conditions. More advanced sample identification based on disjoint modeling techniques and/or nearest neighbor will also be investigated.

ACKNOWLEDGMENTS

This work was funded by an operating grant from the Canadian National Sciences and Engineering Research Council. D.P.W. would like to acknowledge scholarship support from NSERC. The authors thank Thermo Jarrell Ash for the loan of the spectrometer which was used for this research.

REFERENCES:

[100] Salin, E.D.; Winston, P.H., *Anal. Chem.*, **64**, p 49A, 1992.

[101] deMonchy, A.R.; Forster, A.R.;, Arretteig, J.R.; Le, L.;, Deming, S.N.; *Anal.Chem.*, **60**, p 1355A, 1988.

[102] Borman, S., C&E News, February 19, p 20, "New QSAR Techniques Eyed for Environmental Assessments", 1990.

[103] Webb, D.P.; Salin, E.D., *J.Anal.At.Spectrom*, **4**, p 793, 1989.

[104] Morgan, S.L.; Deming, S.N., *Anal. Chem.*, **46**, p 1170, 1974.

[105] Browner, R.F.; Boorn, A.W., *Anal Chem.*, **56**, p 787A, 1984

[106] Botto, R.I., *Spectrochim. Acta*, **40B**, p 397, 1985.

[107] Otto, M., *Anal. Chim. Acta*, **235**, p 169, 1990.

CHAPTER 6

Our prototype design for the autonomous instrument has identified most of the subsystems that need to be present in an autonomous ICP-AES instrumental system. However, an important general problem was left unaddressed in this prototype: how will the autonomous instrument determine what calibration methodology should be used on any specific sample type. The answer of course is to add a rulebase that will allow calibration methodology selection. This rulebase must take into account constraints determined by interrogating the operator about the sample and the results required. This in turn points to the requirement for a system that could extract "prior knowledge" from the operator wherever possible.

Additionally, in the original framework limited provision was made for long term information storage. A more advanced database subsystem needed to be integrated into the framework.

Furthermore the prototype framework was difficult to implement in practice, so a refined framework had to be developed which lends itself more readily to an implementation using the development tools of our choice (in this case C and C++). It is also important that the framework be modular enough that individual components of the whole subsystem can be implemented and integrated without the requirement for major system wide revisions.

The manuscript was published as:

THE AUTONOMOUS INSTRUMENT:
A DESIGN

Douglas P. Webb
Jan Hamier
Eric D. Salin

Department of Chemistry
McGill University
801 Sherbrooke St. W.
Montreal, P.Q., Canada
H3A 2K6
(514) 398-6236

Trends in Analytical Chemistry, Vol. 13, 1994, pp. 44-53.

ABSTRACT

A design of software for an autonomous atomic emission instrument is presented with an emphasis on a subsystem for automatic selection of calibration methodology for inductively coupled plasma atomic emission spectrometry (ICP-AES) is described. The suitability of the 1st-Class expert system shell for development and final implementation of this system is discussed. The calibration methodology system is described in the context of ongoing research into the development of autonomous ICP instrumentation.

INTRODUCTION

We are designing an artificial intelligence based software system which will ultimately be able to duplicate many of the abilities of an expert human instrument operator. We call the system the Autonomous Instrument Expert System. If developed in a sufficiently flexible manner, the core of this work may well be applicable to various types of analytical instrumentation. The attributes this system should have are listed in Table 1.

1) General sample recognition
2) Recognition of important situations which may arise during the course of analysis
3) Self Optimization
4) The ability to learn from experience
5) The ability to choose the appropriate methodology

Table 1: Requirements for the Analytical Analysis System

We have discussed a number of these features and concepts previously[108]. This paper will address the fifth requirement in some detail. The other aspects are under study separately and, in some cases, have been studied extensively. Optimization[109,110,111,112] and sample recognition (pattern recognition[113,114]) are fields which have been extensively researched. Situation recognition covers a broad range of possibilities. For example, line overlaps are a problem which can be predicted and largely avoided by the use of programs like LINEX[115,116], an expert system for the selection of analytical lines. Some spectrochemical and hardware problems can be identified by observing the behavior of non-analyte lines like the carbon line (193.1 nm) and the hydrogen β–line (486.1 nm)[108]. For example, the hydrogen β–line intensity can indicate changes in excitation conditions or denote sample introduction failure.

One of the primary building blocks in the development of this autonomous analytical instrument is the expert system. Expert systems are programs which generate decisions or perform actions at the level of an expert. An important general limitation in expert

systems is the requirement that they work in a very limited knowledge domain. While expert systems may have knowledge in depth, they do not have knowledge in breadth and certainly lack "common sense".

There are three main types of tools for developing expert systems: 1) languages, 2) toolkits, and 3) expert system shells. These tools are well described elsewhere[117,118]. We decided to use an expert system shell for this project because of the anticipated rapid development time as compared to language based development and the reduced cost and training time as compared to toolkit based development. 1st-Class Expert was selected because of its ability to manipulate DBase III+ files, its good technical support options, flexible search strategies and ability to translate its rulebases into C or Pascal code which could then be compiled.

The work of Lahiri and Stillman[119] and Bridge *et al*[120] has shown that expert systems have promise in atomic absorption spectrometry and high-performance liquid chromatography. Inductively coupled plasma atomic emission spectrometry (ICP-AES) is a well established analytical technique for the determination of many elements. ICP-AES provides a uniquely powerful combination of excellent detection limits, wide linear dynamic range and relative freedom from matrix effects. This combination makes ICP-AES the premier method of choice for elemental analysis of many sample types. The multielement analysis capability of ICP and a throughput rate of about one sample per minute provides a capability unmatched in flame or furnace instrumentation. This analytical capability also leads to an information load which exceeds the ability of a human operator to follow in real time. While it is impossible for a human operator to follow ICP-AES operation in real time, many factors may affect the way an instrument should be operated during an analysis. The need for real time feedback involving both pattern recognition and knowledge based decisions makes expert systems desirable if not essential. The reader will note that we have not addressed the question of post processing, another area ripe for expert system technology.

The development of an expert system usually requires an expert. Our expertise lies primarily in the area of ICP-AES. We have

worked with a sequential scanning ICP spectrometer which allows great latitude with respect to the choice of spectral lines which can be examined, including diagnostic lines like those of argon, hydrogen and carbon, but if many lines are examined, a sequential system is slow. Any autonomous instrument control system must take advantage of the strengths of the system to which it is integrated. Numerous types of samples can cause several kinds of analytical failure if not handled properly. For example, very high salt concentrations can lead to sample introduction failure by blocking the nebulizer or the injector tube. An automated system can use feedback to avoid or correct for sample induced problems which may arise. Several different types of analytical methodologies which can be applied to problems which may arise during sample analysis: standard additions, and internal standards, for example. An autonomous system must be able to select which methodology is applicable.

EXPERIMENTAL

Hardware:

An IBM PC-AT with an Intel Inboard 80386 accelerator board was used (although the expert system itself will run on any compatible with at least 640 Kb of RAM). To operate properly, the expert system must be run from a hard disk due to the space required for the various data and reference files.

The software is designed to be general purpose, but is presently configured for a Thermo Jarrell Ash AtomScan 25 coupled to a TJA 300 autosampler, both of which are designed for fully computer controlled operation.

Software:

The control structure portion of the system and the calibration methodology selection modules were written in the 1st-Class[121] expert system development tool. Turbo Pascal[122] was used for the sample information acquisition module (called AJAXDB.EXE) and for all data treatment sub modules. Both the spectrometer and the autosampler are controlled by software provided by the instrument manufacturer (TSPECSA version 5.02). Due to the limited compatibility between the spectrometer control software and the other software used, the expert system described herein does not yet interface with the spectrometer. It is our intent to incorporate this work into the program called AJAX which is described in our previous work[108] which does interact directly with the spectrometer. The entire computer system can be run under MS-DOS version 3.3 or higher.

Design Considerations:

This software was designed with the intention that it should be able to handle quality control, reference and unknown materials of all types. As such the design is open ended and an ongoing research project. It is our expectation that when this system interacts with real-world samples of different types that shortcomings in the original design will be exposed. Therefore, the system was designed in a

modular, object oriented fashion so that rearrangement of modules and addition of new modules should be relatively painless. Thus far the system has been tested on relatively simple standards in aqueous solutions.

DEFINITIONS AND VOCABULARY

Terminology

It is important that the reader share a common vocabulary with us. There are three general types of samples for which we have designed the system: quality control (QC), reference materials, and unknowns. Quality control is a common type of analysis in which repeated analyses of very similar samples are performed. The objective of a quality control analysis is to determine whether each sample's composition falls within a predefined range. Reference materials are of known composition and are used in our system to "teach" the software to run that type of material. The learning technique will be discussed in another paper. An unknown is a sample whose composition is not well characterized. Each of these types of samples must be handled differently.

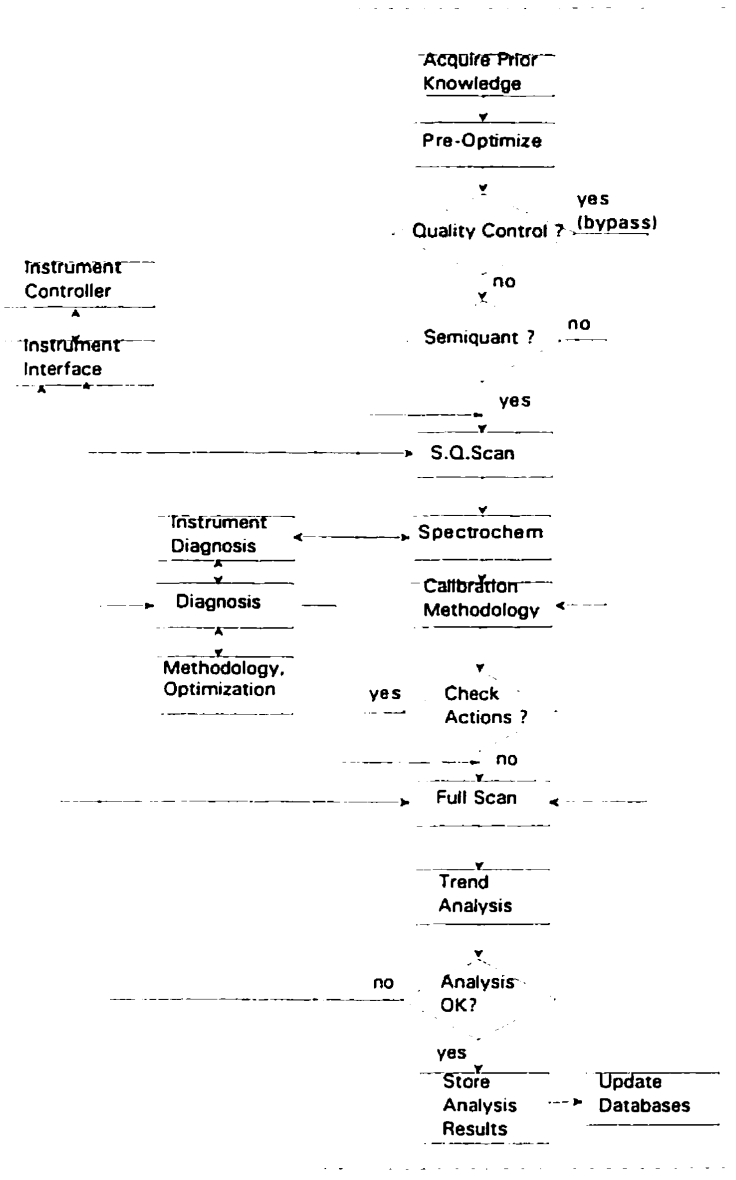


Figure 1. Algorithmic flow-control diagram (approximate) for the autonomous instrument software. The portion discussed in this paper appears within the dotted box.

We have broken the general problem of ICP analysis into sub problems. Each of these sub problems appears as a box in Fig. 1. Some of these problems, *e.g.*, the Instrument Interface, are best solved using traditional algorithmic programming techniques. Other problems are best solved using expert systems. The focus of this paper is the general top level control structure and the calibration methodology selection expert system in specific. In many respects,

the calibration methodology is a major piece of the puzzle, as will become evident.

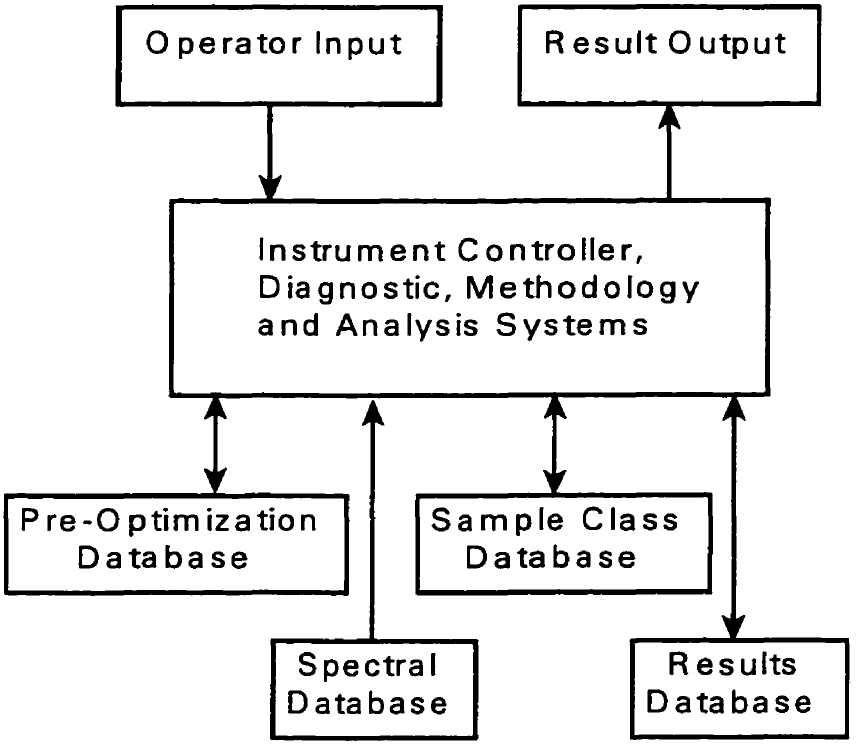


Figure 2. Databases used by the autonomous instrument, and the flow of data within the program.

Ultimately, information exchange between the spectrometer, the system's supporting databases (Fig. 2), and full operational control of both the spectrometer and the spectrometer subsystems (autosampler, autodiluter, gas flow controller, *etc.*) should all be handled automatically. The system should be able to perform the analysis autonomously once a minimal amount of information has been obtained from the operator. This information allows the system to choose the proper calibration methodology, optimize the operational parameters, and identify potential problems associated with the analysis of a given sample. If problems are encountered during sample analysis, the system should determine corrective actions and apply them prior to the final sample analysis. These are the

minimum requirements if the entire ICP-AES system is to be operated with maximal reliability and without constant human supervision.

What Information Does an Autonomous Expert System Need?

It is desirable to have as much information as possible about the sample before an analysis. From our viewpoint, two types of information about a given sample exist: Prior Knowledge, which is information known about the sample before any analysis is carried out, and Acquired Information, which can be obtained during the sample analysis process. Prior Knowledge consists of information like the volume of the sample to be analyzed. It is acquired by prompting the operator to enter information as described below. Acquired Information is information which cannot be reliably determined without spectral analysis, like the fact that the concentration of the analyte is above the upper linear range of the instrument. Although a Full Scan (see below) is also a source of Acquired Information, Acquired Information is usually obtained using a Semiquantitative Scan, which is a fast scan which consumes less time and sample. The Semiquantitative Scan is much less "expensive" than a Full Scan, which is run with the longer integration times required to achieve optimal signal-to-noise ratios. In some analysis scenarios a significant fraction of the knowledge required to assure accurate sample analysis cannot be extracted from Prior Knowledge. For example, spectral overlaps may not always be readily discerned from Prior Knowledge. Analyte concentrations which will saturate the detector pose a similar problem.

The Semiquantitative Scan

The Atomscan 25 has a Semiquantitative Scan option which allows rapid determination of the approximate concentrations of 37 elements without the need to calibrate each element. A Semiquantitative Scan uses very short integration times, and a single pass for each element, therefore accuracy will not be high, and noise will be higher than with a Full Scan. Semiquantitative Scans cannot use internal standards or do inter element corrections in the case of a spectral overlap. (One minor problem remains to be overcome however: At this time the Semiquantitative Scan and its data are not

readily accessible by the automatic spectrometer control software.) Additionally, for a Semiquantitative Scan we look at the carbon line and the hydrogen β–line to generate diagnostic information. These semiquantitative results are neither very precise nor accurate, yet they allow the identification of potential problems arising from spectral line overlap, concentrations out of linear range, high total salt level, high concentration of easily ionizable elements, sample introduction failure, *etc.* These problems can only be identified using spectral information: Acquired Information. Once identified, these problems can be solved before the Full Scan is carried out, thus leading to improved quality of results.

The Full Scan

A Full Scan is an analysis of all the analytical lines necessary to determine the sample concentrations of the elements of interest, but only those elements. This is the traditional analytical mode. The Full Scan must be done with a sufficient number of repeats and integration times that are long enough to ensure analytical results are as accurate as possible. The Full Scan will be the single most costly part for most analyses, and therefore an effort is made to do only one Full Scan.

Databases

An important alternative source of information which must be considered is previous analyses. Figure 2 illustrates the database our system will use, with the arrows indicating the flow of information in the Autonomous Instrument Expert System. Just as skilled operators learn from experience, so should our system, but this is only possible if information is stored in accessible databases as it is acquired. The spectral database can be used to anticipate and avoid spectral line overlaps. The Pre-Optimization database is used to select operating conditions which will favor high accuracy without the need to optimize the instrument for every sample. The sample class database is used to improve the accuracy of the Semiquantitative Scan. The results database is used for trend analysis, and to enable a novel calibration methodology we call Precalibration (described below).

RESULTS AND DISCUSSION

It is important to point out the key goals of the Autonomous Instrument Expert System. The primary goal is to achieve the desired accuracy. An important secondary goal is to minimize the expense of the analysis. Expense is a composite of several factors: instrument time and the use of system resources. A Semiquantitative Scan is expensive and should not be done unless it's necessary. Our decisions are based on the assumption that a Full Scan will be even more expensive than a Semiquantitative Scan. Therefore a Full Scan should be carried out only once per sample if at all possible. Computing power is cheap and fast. So, whenever possible, computing resources should be applied in lieu of instrumental resources.

Functional Overview of the Top Level:

Explanations regarding the operational aspects of the top level of the system are given in an order corresponding to the flow chart provided in Fig. 1.

Acquire Prior Knowledge

This first module (a separate Pascal program called AJAXDB) is designed to extract Prior Knowledge from the operator. This module has several interesting features worth noting: First, it prompts the user with a data entry screen which requests fundamental information such as where the sample came from, who it came from (company or individual), the sample volume, how much time (money) is to be spent analyzing the sample, what level of accuracy is required and whether the sample is a QC, reference material or an unknown (see Fig. 3). These facts should be known about every sample. Other attributes are requested such as sample matrix information, but if this information is unknown, this does not adversely affect the analysis procedure. This module has some error checking built in so that inconsistent entries are automatically brought to the operator's attention. For example, a sample volume of zero milliliters is flagged as an error. As illustrated in Fig. 3, a menu is available for each option except sample volume. A customer name/sample type option

menu will contain a list of all sample types previously handled, other menus present a list of all legal options.

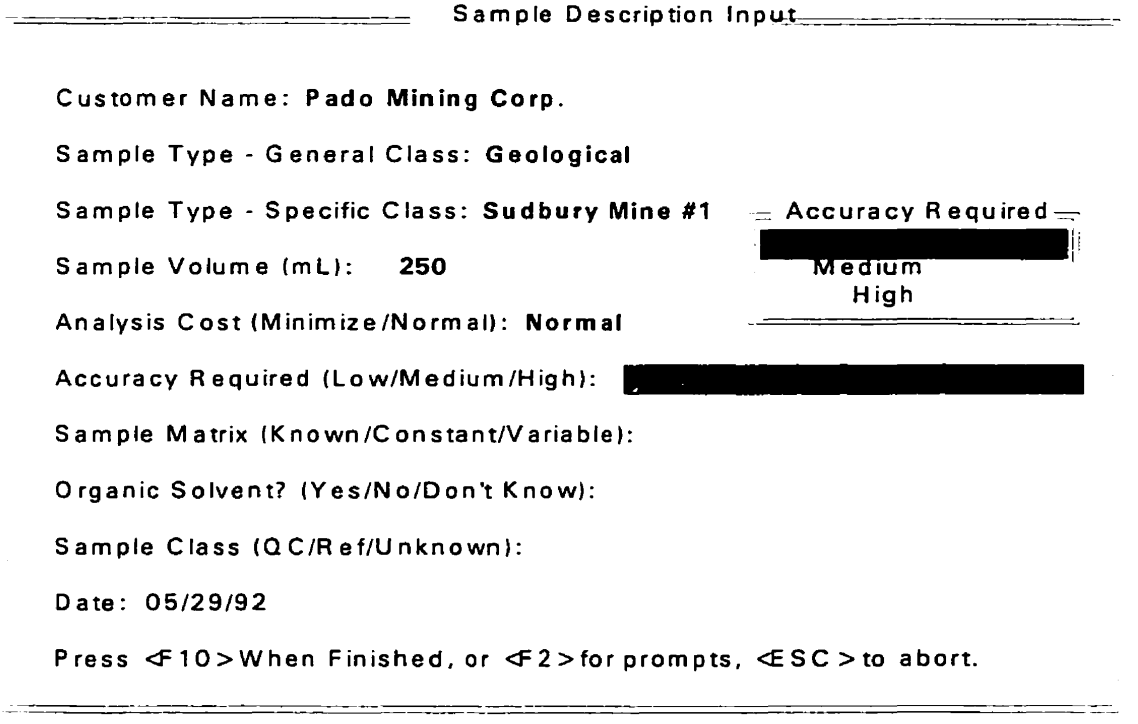


Figure 3. Sample operator input data screen.

Once the error checking step has been successfully performed, the input data is stored: in a DBase III+ compatible database (the sample class database) for future access and in a temporary text file used to pass the Acquired Information to Calibration Methodology module.

Pre-Optimization

If the current sample is similar to a previous sample, a "pre-optimization" of the operational parameters can be done. The concept of the Pre-Optimization is rather simple: After each Full Scan, the operational parameters (forward power, observation height, gas and sample flow rates etc.) are stored on disk in a database called the Pre-Optimization database (see Fig. 2). If the expert system "recognizes" a sample as "very similar" to one previously analyzed, then the same operational parameters are restored with the justification that if they were the "optimal" parameters for the previous sample then it is very

probable that they will be nearly optimal for a similar sample. This approach will increase the likelihood that very little optimization will be needed to obtain the best analysis possible in the shortest time.

QC?

If the sample is a QC sample then several assumptions are made: The sample matrix is consistent and the samples should be analyzed as fast and cheaply as possible without compromising reliability. Therefore, the main body of the control system is bypassed and a Full Scan is run immediately. Some error checking is still required. The signal to noise ratio for the Full Scan results must be monitored so that the results can be said to be meaningful. Trends in sample readings should be watched so that if drift or more serious problems (like partial nebulizer blockage) are occurring they will be detected and remedial action can be taken.

Semiquant?

Next, a decision must be made as to whether a Semiquantitative Scan should be done before deciding on a calibration methodology. Factors influencing this decision are: what sort of accuracy is required, how much time is allotted to the analysis of each sample, and how much sample is present. If the accuracy required is low, for example, then a Semiquantitative Scan alone might be sufficient. If the accuracy required is high, then a Semiquantitative Scan will always be performed, because the system must always be certain that no unexpected spectrochemical effects reduce the accuracy of the analysis. (Consequently, high accuracy is the slowest and most costly mode of operation, but it will always yield the most accurate results possible.) A Semiquantitative Scan will use up some sample, so if only 5 ml were present to begin with, then we could not do a Semiquantitative Scan.

Semiquantitative Scan

The Semiquantitative Scan module directs a request to the Instrument Interface Module to carry out a Semiquantitative analysis as described previously. If a Semiquantitative Scan is performed then the results are stored in a database. To get a more accurate Semiquantitative Scan, a refinement is possible when running similar

samples. A Full Scan is carried out on almost all samples, and it is likely that deviations from the correct concentration in the Semiquantitative Scans is a systematic function of the sample matrix (and may be different for each element in the sample), the results (in concentration units) can be used to "correct" the approximate calibration provided by the Semiquantitative Scan for a sample of the same type. This correction is done by dividing the Full Scan concentrations by the corresponding Semiquantitative Scan concentrations to obtain a Correction Factor associated with each line observed. This Correction Factor is stored on disk and can subsequently be used to "correct" the concentrations reported by a Semiquantitative Scan by multiplying the Semiquantitative Scan concentrations with the appropriate set of Correction Factors. The gain in accuracy should lead to better quality decisions taken by the expert system based on the information from the Semiquantitative Scan.

Spectrochem

The Spectrochem module is designed to identify all obvious spectrochemical problems, suggest corrective actions and generate a list of calibration methodologies which are incompatible with the sample. For example, a sample concentration outside the range of the calibration curve will eliminate the consideration of the external calibration methodology. Spectrochemical problems are those which have either a spectroscopic origin (line overlap etc.) or a chemical origin (matrix effects and concentration related problems), or a combination of the two.

As discussed previously, some spectrochemical and hardware problems can be identified by observing the behavior of non-analyte lines like the carbon line and the hydrogen β–line. If there is an indication that the spectrometer might be malfunctioning or on the verge of failure, control will be passed to the Instrument Diagnosis Module which will attempt to confirm the problem and take corrective action.

Should any spectrochemical problem be detected, the Spectrochem module would not only identify it (them) but also

suggest corrective actions. To continue with our example, if the concentration of the analyte is out of the concentration range covered by the sample calibration curve, then dilution might be recommended. In this case, the Check Actions? module would be activated to confirm the effectiveness of the remedy.

This module also looks to see if spectrochemical information exists that would suggest that the use of one (or more) calibration methodologies will not yield acceptable results. For example, if the Semiquantitative Scan indicates a sample of sufficient complexity that a good blank is unlikely then standard additions should be flagged as unusable.

One of the advantages of the modular design approach emerges in the Spectrochem module, as existing data treatment programs (like LINEX[115]) can be incorporated as sub-modules. To give more flexibility to such sub-modules, reference data (like a table of atomic emission lines and their intensities) would be provided in a database called the spectral database (Fig. 2). NOTE: This module is under development.

Calibration Methodology

First, this module checks for any contradictions between the Prior Knowledge acquired from the operator and the results (if any) from the Semiquantitative Scan. For example, if the operator indicated that the solvent was not organic but the results of the Semiquantitative Scan indicate otherwise then prior knowledge is corrected.

The main part of the Calibration Methodology module is a large 1st-Class knowledge base. (Fig. 4) This module accepts the knowledge entered by the user and returns a prioritized list of "recommended", "allowed", and "forbidden" calibration methodologies. This rule uses a prioritization mechanism as an alternative to fuzzy logic [123,124] or the use of certainty factors [125]. Each leaf on the rule tree in Fig. 4 has either 'error', which indicates a condition which should not occur, or a result code, for example 'R8B'. These result codes are simply commands to look up a prioritized calibration

methodology list which was developed previously by an expert. Several examples appear in Table 2.

Result Code	Methodology Recommendations
R2A	Standard Additions, Single Spiking - Forbidden
	Standard Additions. Multi-Spiking - Forbidden
	Internal Standards - Allowed
	External Calibration Curve - Forbidden
	Precalibration - Recommended
R16B	Standard Additions, Single Spiking - Forbidden
	Standard Additions. Multi-Spiking - Recommended
	Internal Standards - Forbidden
	External Calibration Curve - Forbidden
	Precalibration - Forbidden
R31A	Standard Additions, Single Spiking - Forbidden
	Standard Additions. Multi-Spiking - Forbidden
	Internal Standards - Allowed
	External Calibration Curve - Recommended
	Precalibration - Forbidden

Table 2 - Result Codes for the Calibration Methodology Rulebase

Certainty factors and fuzzy logic can be used to weight the likelihood that an alternative is the best one. These factors can be propagated with the execution of each rule. When the final rule is executed the alternative with the best value is chosen as the result. Alternatively, a complete set of all possible pathways can sometimes be generated if the rulebase is small enough, and the best alternative assigned manually for each pathway. The advantage of the former is that it allows the rulebase to be expanded easily and can be used for very large rulebases. The latter however has the advantages that it is easier to debug and to alter the final result based on any given input. We have used the latter 'rigid' approach.

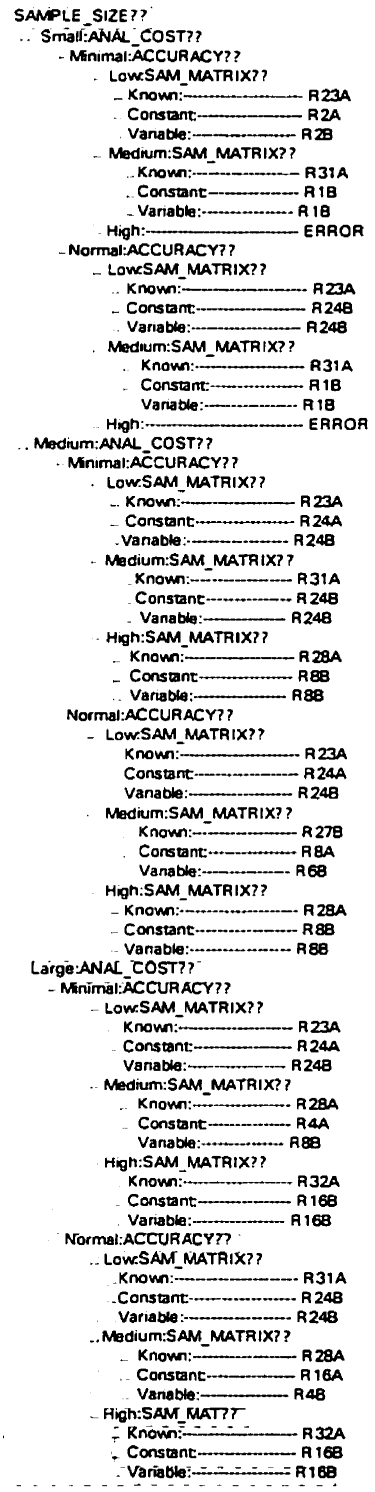


Figure 4. Calibration methodology selection rule, as generated by the 1st-Class expert system shell. The decision path to a sample result is surrounded by the dashed line.

The use of various reference databases (see . 2) allowed a novel low accuracy calibration methodology to be considered. The

Precalibration methodology uses calibration data that have been established for a previously run sample. This technique allows calibration to be bypassed so that the final results are obtained more quickly. The sample is analyzed immediately without calibrating the instrument. The recycled calibration curve might not correspond exactly to the sample analyzed, however this is irrelevant since the accuracy required is "low".

An example will show how this module works. The data in Table 3 will be considered sample operator input. Because the sample volume is quite large no options are precluded due to lack of sample. This is an unknown sample and the accuracy desired is high so a Semiquantitative Scan will be run to get a general idea of the sample composition. The Spectrochem module will then look for problems as described above.

Customer Name: **Pado Mining Corp.**
Sample Type - General Class: **Geological**
Sample Type - Specific Class: **Sudbury Mine No. 1**
Sample Volume (ml): **250**
Analysis Cost (Minimize/Normal): **Normal**
Accuracy Required (Low/Medium/High): **High**
Sample Matrix (Known/Constant/Variable): **Variable**
Organic Solvent ? (Yes/No/Don't Know): **No**
Sample Class (QC/Ref/Unknown): **Unknown**
Number of Samples: **30**
Date: 02/07/92

Table 3 - Sample Data Note: Operator input appears in **Bold**.

The Calibration Methodology module will then execute its calibration selection knowledge base and a decision path like the following will result: (The rule path used is surrounded by a dotted line in Fig. 4.)

IF Sample_Class is Unknown AND
Sample_Size is Large AND
Analysis_Cost is Normal AND
Accuracy_Required is HIGH AND

Sample_Matrix is Variable THEN

Multi-spiking Standard Additions is the preferred calibration methodology

Single Spiking Standard Additions is disallowed

Internal Standards is disallowed

External Calibration Curve is disallowed

Precalibration is disallowed

Precalibration (described earlier) is not considered because the accuracy desired is not Low. External calibration curve is not feasible because the matrix is listed as variable, so, in theory, a matrix matched calibration curve would be needed for each and every sample. Selection of appropriate internal standards will be difficult because the matrix is variable. Multi-spiking standard additions is selected over single spiking because of the desire for the highest accuracy and the fact that we are not attempting to minimize cost, multi-spiking allows confirmation that the signal is proportional to concentration (a fundamental assumption of Technique of Standard Additions).

Check Actions?

This module, if activated by the Spectrochem module, decides whether the success of any corrective actions taken should be checked to ensure that there are no more spectrochemical problems. If for example a sample was sufficiently concentrated that it had to be diluted before analysis, this module might want to confirm that the concentration is now low enough that it could be successfully determined. Factors influencing this decision are: sample volume, time to be allocated to the analysis of each sample, and accuracy required. In the event that a check is to be made the system branches back to the Semiquantitative Scan module.

Full Scan

The Full Scan module directs a request to the Instrument Interface module to carry out a full quantitative analysis of the type described previously. The analytical results and the operational parameters (forward power, observation height, gas flow rates and

sample flow rates, etc.) are then saved on disk. The results are used to calculate correction factors for the Semiquantitative Scan and are saved in the Sample Class database (Fig. 2) while the operational parameters are stored in the Pre-Optimization database (Fig. 2).

Trend Analysis

This module is specifically designed to identify systematic trends in system response during QC runs. This is particularly important, because it is the only way to identify systematic failures or instrumental degradation when analyzing QC samples, because the usual checks have been bypassed for the sake of lower cost (i.e. higher throughput).

Analysis OK?

This module checks to see if any instrumental failure is indicated by the result from the Full Scan or if the signal to noise ratio for the sample analysis was poorer than some predefined threshold value. In either of these cases, control is passed to the diagnosis module which determines what course of action is most likely to resolve the problem. There are two possibilities at this point. The Diagnosis module can elect to optimize instrument parameters (e.g. power, flow rates),and then rerun the sample or it can enter a detailed Instrument Diagnosis module which usually starts with running a standard solution.

CONCLUSIONS

An important point of discussion is the validity of the choice of 1st-Class as the expert system shell used to construct various portions the system. 1st-Class has the primary advantage of being easy to use for the development of "IF-THEN" type production rules, using either forward or backward chaining[117]. Moreover, the rules are built and modified with relative ease by using the built-in rule editor. This rule editor allows an expert system constructed in 1st-Class to be easily modified during the development cycle and/or customized by the user to better suit his/her needs.

On the other hand, iteration and mathematical operations are tedious in 1st-Class. This difficulty mandated that raw information (spectral/numerical) had to be translated, by means of programs written in a high level language, into "facts" that could be related by simple "IF-THEN" statements in the corresponding 1st-Class rule. Also, 1st-Class has limited built-in capability for providing "on line" justifications of the various decision steps taken by the expert system as the acquired data is processed. This lacuna makes the expert system appear to the end user as a "black box" into which data is fed at one end and a result is obtained at the other end without any justifications.

While 1st-Class was used to develop the prototype top level expert system, it became clear that 1st-Class could not easily cope with the requirements of a run-time version of the expert system. Despite the difficulties associated with the use of 1st-Class, the approach used is entirely valid and has some interesting advantageous characteristics: first, the concept of the Semiquantitative Scan allows the system to be as independent from the operator as possible by allowing critical information to be deduced by the system without a large loss of time and second, the identification of spectrochemical problems and Pre-Optimization can reduce the costs associated with an inappropriately configured Full Scan. Our current plan is to take the existing rule set and implement it in a programming language like 'C' or PROLOG (or both) to gain flexibility. This should be eased by the fact that, as mentioned earlier,

1st-Class Expert allows its rulebases to be converted to 'C' source code.

It is our contention that the design presented herein should allow the expert system to carry out the entire analysis of a series of samples without intervention by the operator unless some serious problem that requires physical intervention occurs.

Constant updating of both Pre-Optimization and Sample Class databases after each analysis simplifies the task of a) the operator, by allowing him/her to browse through a database of information on previously run samples and/or automatically prompting him with information on any sample that matches the sample class of his current sample and b) the Autonomous Instrument Expert System, by providing it with the operational parameters and calibration factors from a previously run sample that matches most closely the characteristics of the current sample (an automated Pre-Optimization of the analysis parameters).

The modular design of the Autonomous Instrument Expert System used allows specific expert subsystems to be incorporated as components. Development of the components can, to a large extent, be done independently of the rest of the system. This type of design is well suited to a project of this magnitude.

ACKNOWLEDGMENTS

This work was funded by an operating grant from the Canadian National Sciences and Engineering Research Council. D.P.W. would like to acknowledge scholarship support from NSERC. The authors also thank Thermo Jarrell Ash for the loan of the spectrometer which was used for this research.

REFERENCES:

[108] Webb, D.P. and Salin E.D., *Int. Instr. & Comp.*, Sept./Oct, p 185, 1991.

[109] Morgan, S.L. and Deming, S.N., *Anal. Chem.*, **46**, p 1170, 1974.

[110] Routh, M.W.; Swartz, P.A.; Denton, M.B., *Anal. Chem.*, **49**, p 1422, 1977.

[111] Betteridge, D.; Wade, A.P.; Howard, A.G., *Talanta*, **32**, p 709, 1985.

[112] Betteridge, D.; Wade, A.P.; Howard, A.G., *Talanta*, **32**, p 723, 1985.

[113] Wold, S., *Pattern Recognition*, **8**, p 127, 1976.

[114] Sjöström, M., Kowalski, B.R., *Anal. Chim. Acta*, **112**, p 11, 1979.

[115] Webb, D.P. and Salin E.D., *J. Anal. At. Spectrom.*, **4**, p 793, 1989.

[116] Webb, D.P. and Salin, E.D., *Spectrochim. Acta*, **47B**, p E1587, 1992.

[117] deMonchy, A.R.; Forster, A.R.; Arreteig, J.R.; Le, L.; Deming, S.N., *Anal. Chem.*, **60**, p 1355A, 1988.

[118] Settle, F.A. Jr., *Can. Chem. News*, **44**, p 12, March 1992.

[119] Lahirir, S. and Stillman, M.J., *Anal. Chem.*, **64**, p 283A, 1992.

[120] Bridge, T.P.; Williams, M.H; Fell, A.F., *Anal. Proc.*, **25**, p 43, 1988.

[121] "1st Class Reference Manual", AI Corp, Waltham, MA, 1989.

[122] Borland International, "Turbo Pascal User's Guide", Scotts Valley, CA, 1990.

[123] Otto, M., *Anal. Chim. Acta*, **235**, p 169, 1990.

[124] Zadah, L.A., *CHEMTECH*, **17**, p 340, 1987.

[125] Bratko, I., "PROLOG Programming for Artificial Intelligence", Addison-Wesley Publishing Co., Don Mills, ON, pp 347-354, 1986.

CHAPTER 7

A framework has now been designed to support the autonomous instrument system. The next phase in it's development is to build the major outstanding subsystems. One subsystem whose inclusion in the autonomous instrument would be desireable is that of a robust general purpose optimizer. Furthermore, the development of such a system could offer the opportunity to learn about optimizing ICP, which is deemed to be an important feature by industry and is also deemed to be moderately difficult.

OPTIMIZATION OF INDUCTIVELY COUPLED PLASMA ATOMIC EMISSION SPECTROMETERS

ABSTRACT

A study was conducted to determine the best way to optimize the performance on an inductively coupled plasma atomic emission spectrometer (ICP-AES). Six spectroscopic lines from the elements Ca, Mn and Zn were used for this study, including both hard and soft lines, as well as atomic and ionic lines. Optimization of both discrete and continuous surfaces were examined because while many spectrometer systems have a continuous (or nearly continuous) range of parameter control, some (the Thermo Jarrell Ash Model 25 rapid scanning ICP, for example) can only configure the instrumental operating parameters in large discrete steps. Three optimization algorithms are compared in their ability to optimize ICP-AES performance. General recommendations for optimization of ICP-AES systems are presented.

On discrete surfaces the composite modified simplex was found to achieve a better response than the modified simplex, and on smooth surfaces the Davidon, Fletcher & Powell algorithm outperformed both simplexes in this regard. However, the selection of the best optimization algorithm is not obvious, but depends on the relative importance placed on the various optimization performance characteristics.

INTRODUCTION

The Problem

Many papers have been written in recent years describing how to optimize response surfaces, as can be seen in the volume of material listed in recent reviews.[126,127] Many of these articles discuss optimizing odd surfaces which have little or no correspondence to the typical ICP-AES response surface - which, for single elements are sometimes monotonic and often have relatively broad features. Broekaert *et al* have stated that, in general, inductively coupled plasma atomic emission spectrometry (ICP-AES) is difficult to optimize[128]. However, particularly when operating near the limit of detection, ICP instrumentation should be periodically optimized to ensure that the highest quality results are being obtained. Sample analysis is a relatively expensive and time consuming task in ICP and therefore we desire to achieve optimal or near optimal response in the minimum possible number of experiments. For this reason a moderate number of papers have been written describing how one algorithm or another has been used to optimize sample introduction or overall ICP performance.[129,130,131,132] There are however no definitive papers which compare the performance of the most suitable optimization algorithms at the task of optimizing ICP performance.

The Algorithms

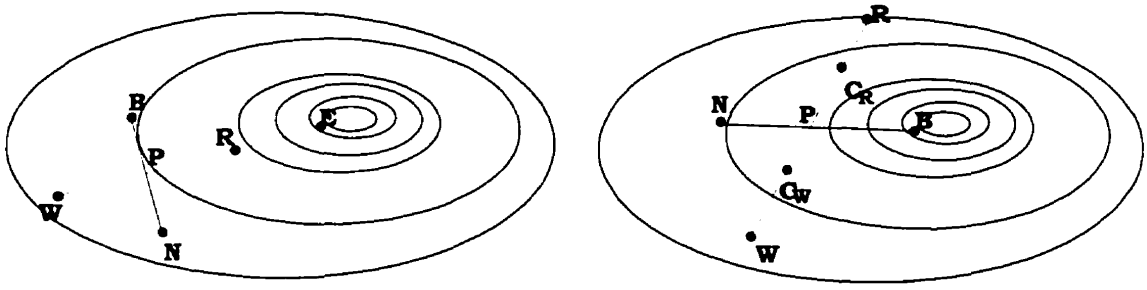
This paper will compare and contrast the performance of the modified simplex[133], composite modified simplex (CMS)[134,135], and a variant of the Davidon-Fletcher-Powell (DFP) optimization algorithm[136] as implemented by Thomas and Collins.[132]

The Modified Simplex

The Morgan and Deming[133] modified simplex was included in this study because it is widely referenced and simple in concept and therefore widely used. The main points of this algorithm are illustrated in Figure 1. In this algorithm K+1 starting points are selected, (where K is the dimensionality of the hypersurface being optimized) evaluated and ranked, in this case the points B(est), N(ext best) and W(orst). A ray is drawn from W through P, the centroid of the K best points in the simplex, to R an equal distance beyond P. Point R is evaluated, and if the response at R is better than B then the ray is extended the same

distance again to E which is also evaluated. The K+1 best points are kept and then reranked, and the process starts again. If the response at R is worse than that at N, then the simplex contracts half way towards P, to point C_R. If the response at R is worse than at W, then the simplex contracts to half way between P and W, to point C_W. Again the K+1 best points are reranked and the process starts again.

Eventually the size of the simplex, or the rate of improvement of the simplex will decrease below a predetermined threshold and the process is stopped.



Modified Simplex Step1 **Modified Simplex Step2**

Figure 1 - The modified simplex

The Composite Modified Simplex

The CMS was included in this study because it incorporates features to address many of the limitations of the modified simplex: better handling of optima near a surface boundary by truncating the ray from W to either R or E at the surface boundary if necessary; the ability to better handle a failed contraction (which happens if the response at C_R was worse than R, or the response at C_W is worse than the response at W); the ability to perform a lagrangian interpolation to an optima internal to the simplex and therefore accurately and rapidly contract when necessary.

An example of this last characteristic is illustrated in Figure 2. After determining that the response at C_R is better than at both R and W it is inferred that a maxima lies on the line segment between R and W. A lagrangian interpolation is performed to estimate the position of this optima (in Figure 2, the point L.)

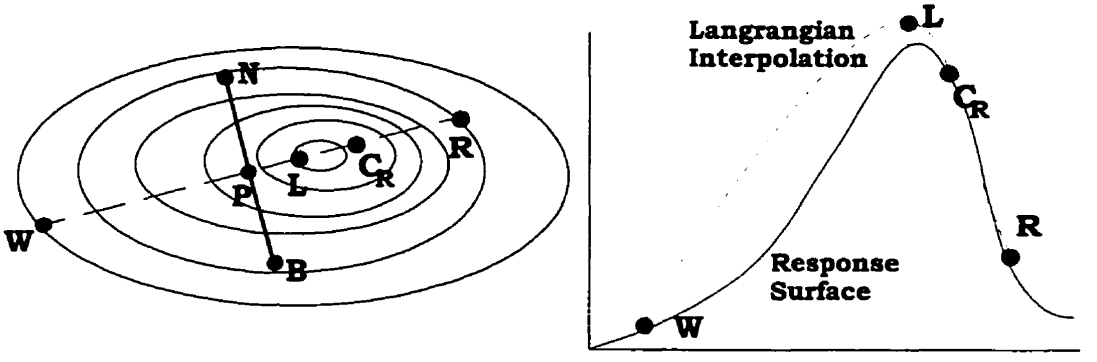


Figure 2 - Composite modified simplex - lagrangian interpolation.

The Thomas & Collins DFP Variant

The DFP algorithm was included in this study because of its availability in a commercial ICP product.[132] The DFP commences with a starting point S (usually in the middle of the hypersurface). Then 2×K additional points (where K is dimensionality) are evaluated which are ± a fixed percentage of the surface dimension (a two dimensional example is shown in Figure 3 part A.) In each dimension the optima is then estimated by parabolic fit. The vector sum of these estimates, point R is then evaluated, as is point E, twice the distance from point S (as shown in Figure 3 part B.). Then point P is estimated, again by parabolic fit. The procedure is started again with P as the new center point and a span one-half that used in the previous iteration.

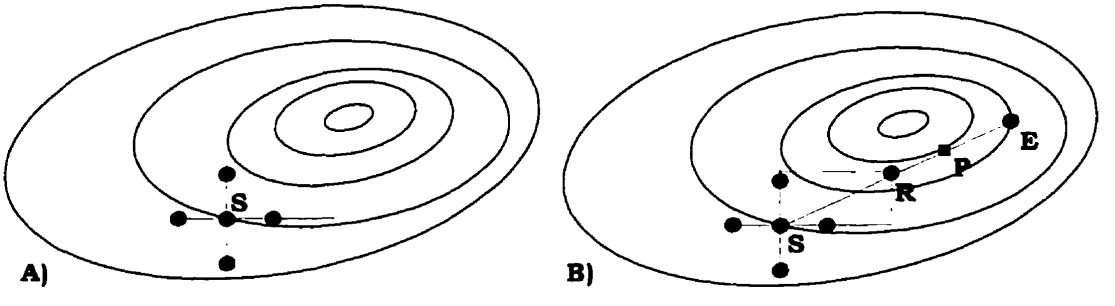


Figure 3 -One iteration of the Thomas-Collins variant of the DFP optimization algorithm

Because the DFP algorithm requires that the derivative of the optimization surface be available at arbitrary points, this algorithm is unsuitable for use on a discrete surface and was therefore not used for that portion of the study.

Algorithms based on exhaustive search, axial iteration, and surface fitting were not consider for inclusion, because they all require an excessive number of experiments to achieve the optima with reasonable precision.[132]

Objective Functions

Three different objective functions were used to optimize ICP performance: signal to noise ratio (SNR) or % relative standard deviation (%RSD); signal to background ratio (SBR) and criteria IV as defined by Werner and Friege.[137] SNR was selected because it is an often quoted and very useful objective function directly related to the precision of the measurements. Because %RSD, the inverse of SNR, is easier to extract from the raw instrumental data used in this study it is presented in the results instead of SNR. SBR ratio was selected because optimization on this objective function has been reported to give the best limits of detection.[138] Werner and Friege's criterion IV (CIV) (see Figure 4) was included because it's developers assert that optimizing a plasma using it leads to better precision than SBR alone.[137]

$$c_{opt} = \frac{0.01 \times c_q}{SBR(c_q)} \times \left[RSD(I_q) + RSD(I_o) \right]$$

Figure 4 - Werner and Friege's criteria IV. Where c_q is concentration of the element, I_q is gross intensity the signal at a wavelength for an element of concentration c_q, and I_o is the background intensity.

Issues not Addressed

Certainly there are several other objective functions that might be considered when optimizing an ICP: minimum ionization interference[139], and spectral interference reduction being examples. These objective functions are not addressed in this paper.

The compromising objective functions required for multi-element optimization are not explored, because they have been examined in some detail elsewhere.[140,141,142]

EXPERIMENTAL

Instrumentation

A Thermo Jarrell-Ash (TJA) Atomscan 2400 (a rapid scanning 27 Mhz ICP-AES) running an argon plasma was used to acquire data for this study. A Watson-Marlow 503U peristaltic pump was used to regulate liquid flow to the sample introduction system. Sample was introduced using a TJA fixed-crossflow nebulizer. The spray chamber and torch are standard TJA.

Elements/Lines Being Optimized:

To keep the study's scope manageable, 6 lines were selected for this optimization study. The analytical lines being optimized are shown in Table 1.

Element	Wavelength	Ionic/Atomic	Hard/Soft[143]
Ca	396.847nm	Ionic	Hard
Ca	422.673nm	Atomic	Soft
Mn	279.482nm	Atomic	Soft
Mn	257.610nm	Ionic	Hard
Zn	213.856nm	Atomic	Hard
Zn	202.551nm	Ionic	Hard

Table 1 - Elements and Emissions lines used

These lines were chosen because they represent a cross section of the types of lines that are typically used in analyses. In the following study, all elements were determined at a concentration of 1 ppm which varied from about 20 times the instrumental detection limit (IDL) for Zn 202.551 to about 100 times the IDL for Mn 257.610.

Parameters to be Optimized:

There are 6 operating parameters to optimize in ICP-AES spectrometry[144]: sample or injector gas flow rate, auxiliary gas flow rate,

torch or plasma gas flow rate, viewing height, forward power, and sample solution feed rate. Optimization will invariably proceed faster if fewer parameters need to be optimized. Nebulizer design and physical properties or the chemical nature of the sample dictate the sample feed rate required to achieve optimal nebulization.[144] Since these are sample specific we have elected to omit them from the optimization study. Carpenter and Ebdon have shown that auxiliary and plasma gases have relatively little effect on overall plasma performance[140] and therefore the decision was made to adopt compromise conditions for these parameters as well. This leaves sample gas flow rate, viewing height and forward power.

Surface Optimization

Because of the large number of experiments that were to be performed in this study a method for simulating an ICP response surface was essential. We decided to perform a coarse but exhaustive surface mapping of the three dimensional response surface defined by sample gas rate, viewing height and forward power.

Experiments were performed at all combinations of the operational settings listed in Table 2 (210 experiments in all) with measurements made at each of the lines listed above.

R.F. Power	0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0 Kilowatts
Sample Gas Rate	0.4, 0.45, 0.5, 0.55, 0.6, 0.65 L/min
Viewing Height	11,13,15,17,19 mm above the load coil

Table 2 - ICP operating parameters and settings used

There are instruments that have discrete and large steps for their operating parameters (the TJA Atomscan 25, for example.) This discrete response surface is used as is to emulate this type of system. Many instruments have continuously (or nearly continuously) adjustable operating parameters and therefore required points not actually part of our experimental data set. These points were generated using polynomial interpolation.[145] The primary drawback of interpolation is that it does not model the real response surface very closely in some cases, yielding exaggerated minima and maxima.

Sampling Methodology

Because we wanted to give each algorithm many chances to succeed (or fail) in optimizing this single surface, initial optimization points were modified in such a way as to present as many different views of each surface as possible. The simplex methods are sensitive to starting orientation on the hyper surface being optimized, so the tetrahedron delimited by the simplex starting points was rotated in each of the three dimensions in $2\pi/5$ increments for a total of 125 unique starting orientations. This is conceptually similar to the rotation that Burgess used to "shake up" the simplex and thereby avoid failed contractions.[146] Starting orientation has less influence on the behavior of the DFP algorithm. Therefore, the actual starting center point of each optimization was offset in each of the three dimensions in increments of 7.5% of the dimensional size, for a total of 125 starting locations. So for each configuration, 125 optimizations on each of the 6 surfaces were performed. Results reported here will be the average of the 125 optimizations except where otherwise indicated.

Optimization Parameters

The starting size of the simplexes were varied between 5 and 45% of the response surface dimension, and the starting size of the DFP optimizations were varied between 5 and 25% of the response surface, each in 9 equal steps. This was done to determine the optimum starting size of each type of optimization.

Endpoint Criteria

The simplex based algorithms were allowed to proceed until they had approached the optima to within a fixed precision. That level of precision was deemed to have been reached when the simplex had contracted to a predefined percent of the hypersurface being optimized. Both 5% and 10% were used to see if the extra experiments required to reach a precision of 5% resulted in a significant improvement in the response, and what the cost was in terms of time. The DFP algorithm was allowed to perform a fixed number of iterations as described by Thomas and Collins[132]. One, two and three iterations were tried. Each iteration using a span one half that of the previous iteration. Each iteration requires 9 experiment in a three dimensional hypersurface, and

is therefore expensive in terms of time. These three iterations were tried to determine whether the extra experiments yielded a noticeable improvement in the response.

Software used

The bulk of the software used in this study was custom code written in C++, with the exception of the interpolation routines that are those of Press *et al.*[145] Summary data and most of the charts were generated using Quattro Pro for Windows[147].

RESULTS AND DISCUSSION

Before presenting results and recommendations it must be emphasized that the behavior of individual optimization algorithms vary in efficacy depending on the shape of the response surface, and that the shape of the response surfaces vary significantly depending on the objective function being used. For example, the S/B surfaces associated with the analytical lines used in our study invariably have broad features and in fact are often nearly monotonic (See Figure 4).

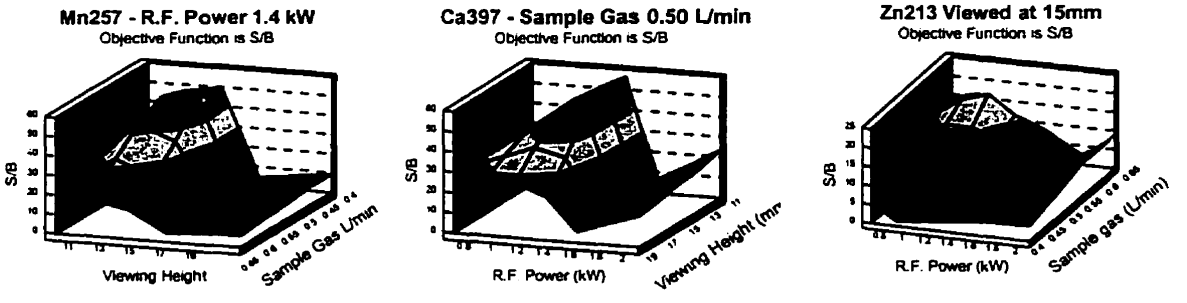


Figure 5 - 3 Typical SBR surfaces. (Each of the surfaces shown here is actually 1 slice in a 3 dimensional hypersurface.)

The %RSD response surfaces were observed to often exhibit more complex features (See Figure 5). CIV surfaces lie somewhere in between in terms of feature complexity.

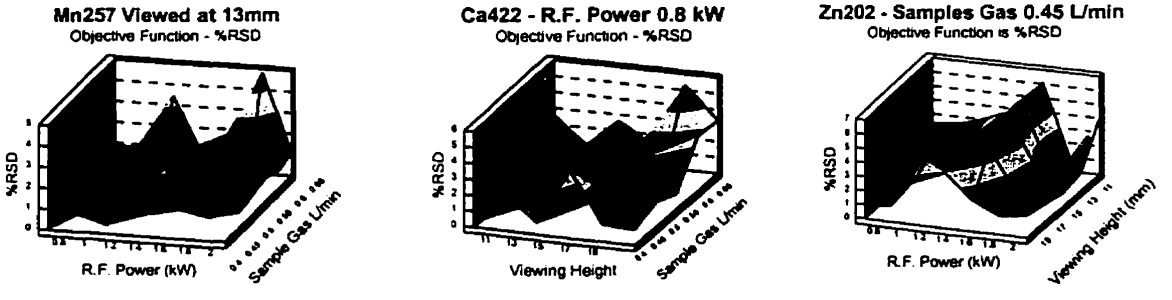


Figure 6 - 3 Typical %RSD surfaces. (Each of the surfaces shown here is actually 1 slice in a 3 dimensional hypersurface.)

The first step in deciding which optimization algorithm will best suit the needs of the experiment is to be sure that we have each

algorithm tuned to perform at it's best. The variable that we can tune is the starting size of the simplex or in the case of the DFP algorithm the span of the first iteration. Tables 3 and 4 show the modified simplex and its average performance at the starting sizes used in this study and the number of steps it took to achieve this performance.

Note: Both the %RSD and CIV hypersurfaces were minimized while the SBR surface was maximized.

	%RSD Response Surface				SBR Response Surface				CIV Response Surface			
Starting Simplex Size	5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria	
	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	1.1	6.0	1.1	6.0	18	6.0	18	6.0	5.8*	6.0	5.8	6.0
0.1	0.78	7.9	0.81	5.9	19	7.8	19	5.9	4.2	7.9	4.2	5.9
0.15	0.64	12	0.68	6.5	25	15	22	7.0	2.9	13	3.2	6.6
0.2	0.58	13	0.61	8.7	27	18	25	12	2.2	15	2.4	9.9
0.25	0.54	15	0.57	11	27	19	26	13	1.8	17	2.0	12
0.3	0.50	15	0.53	12	27	19	25	13	1.7	17	2.0	12
0.35	0.49	16	0.51	12	27	20	26	13	1.7	18	1.9	12
0.4	0.49	17	0.51	12	27	20	26	14	1.6	18	1.8	13
0.45	0.46	17	0.48	13	28	21	27	15	1.5	19	1.6	13

Table 3 - Modified Simplex. Discrete Surface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. * All CIV values are $e10^{-5}$

The values in the *Resp.* column are the average response achieved by that algorithm. Small numbers indicate a better result for the %RSD and CIV hypersurfaces, while a bigger number is better on the SBR hypersurface. #*Steps* is the number of experiments (on average) required to achieve that response. Normally the desired result is to achieve the best response while performing the fewest number of experiments possible, however there is always a positive correlation between number of experiments performed and response improvement. Therefore the operator must select a compromise which requires the minimum number of experiments to achieve an acceptable response.

	%RSD Response Surface				SBR Response Surface				CIV Response Surface			
Starting Simplex Size	5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria	
	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	0.66	5.5	0.61	5.5	18	5.7	18	5.7	10*	5.5	10	5.5
0.1	-0.07	10	0.24	5.6	26	17	21	5.7	-0.74	13	6.6	5.4
0.15	-0.15	14	0.14	6.5	27	19	23	7.5	-2.8	15	3.3	6.4
0.2	-0.02	15	0.11	9.0	29	20	27	12	-3.9	16	-0.25	9.5
0.25	-0.10	17	0.014	11	29	21	27	12	-4.4	17	-0.85	11
0.3	-0.12	17	-0.020	12	29	21	27	13	-4.2	18	-0.93	12
0.35	-0.16	17	0.017	12	29	21	26	13	-3.8	18	-0.30	12
0.4	-0.21	17	-0.033	12	29	22	27	14	-6.0	19	-1.3	13
0.45	-0.13	17	-0.013	13	28	22	26	14	-5.6	19	-2.2	14

Table 4 - Modified Simplex. Interpolated Surface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. *All CIV values are $e10^{-5}$ NOTE: Because all surfaces are interpolated, artifacts like negative %RSD and CIV become possible.

Tables 5 and 6 show the composite modified simplex and it's average performance at the starting sizes used in this study.

	%RSD Response Surface				SBR Response Surface				CIV Response Surface			
Starting Simplex Size	5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria	
	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	1.1	4.0	1.1	4.0	18	4.0	18	4.0	5.8*	4.0	5.8	4.0
0.1	0.61	8.6	0.81	4.5	20	8.5	19	4.4	3.4	9.1	4.3	4.5
0.15	0.53	13	0.61	8.0	25	15	24	8.3	2.0	14	2.9	8
0.2	0.48	16	0.52	13	27	17	26	15	1.4	16	1.7	14
0.25	0.44	17	0.47	15	27	17	26	15	1.2	17	1.3	15
0.3	0.43	18	0.45	16	27	17	26	15	1.2	17	1.3	15
0.35	0.42	19	0.44	17	27	18	26	16	1.2	17	1.2	16
0.4	0.42	19	0.43	17	27	17	26	16	1.1	17	1.2	15
0.45	0.39	18	0.40	17	26	17	26	15	1.1	17	1.2	16

Table 5 - Composite Modified Simplex. Discrete Surface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. *All CIV values are $e10^{-5}$

	%RSD Response Surface				SBR Response Surface				CIV Response Surface			
Starting Simplex Size	5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria	
	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	0.63	5.7	0.61	5.7	18	5.9	18	5.8	10*	5.6	10	5.6
0.1	-2.0	14	0.33	6.0	26	18	21	5.8	-2.1	17	7.2	5.7
0.15	-2.0	18	0.088	8	27	21	23	8.5	-4.2	19	3.2	7.7
0.2	-1.2	20	-0.026	13	29	21	27	14	-4.8	21	-2.1	14
0.25	-2.0	23	-0.12	16	29	22	28	15	-8.2	22	-3.2	16
0.3	-2.4	24	-0.20	17	29	23	28	16	-5.7	23	-2.9	17
0.35	-2.0	24	-0.095	18	29	24	29	16	-6.0	23	-2.3	17
0.4	-3.6	25	-0.22	19	29	24	29	16	-8.0	25	-3.8	19
0.45	-3.3	25	-0.20	20	28	23	27	16	-6.5	23	-2.8	17

Table 6 - Composite Modified Simplex. Interpolated Surface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. *All CIV values are $e10^{-5}$ NOTE: Because all surfaces are interpolated, artifacts like negative %RSD and CIV become possible.

Table 7 shows the DFP optimization and it's average performance at the starting sizes used in this study. Results with a 3 iteration end point criteria showed only slight improvement over the 2 iteration at a cost of 9 additional experiments.

	%RSD Response Surface				SBR Response Surface				CIV Response Surface			
Starting Span	1 Iteration		2 Iterations		1 Iteration		2 Iterations		1 Iteration		2 Iterations	
	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	0.18	10	0.56	19	20	10	22	19	8.8*	10	2.3	19
0.075	0.08	10	-0.69	19	20	10	23	19	6.3	10	-2.3	19
0.10	0.12	10	1.1	19	21	10	23	19	4.9	10	1.2	19
0.125	-0.06	10	-1.7	19	22	10	24	19	2.7	10	-4.9	19
0.15	-0.08	10	-1.2	19	22	10	24	19	2.4	10	-2.4	19
0.175	-0.01	10	-3.2	19	23	10	25	19	2.1	10	-3.8	19
0.20	-0.06	10	-4.5	19	23	10	25	19	1.0	10	-8.3	19
0.225	-0.04	10	-3.0	19	23	10	26	19	-0.61	10	-8.4	19
0.25	-0.09	10	-2.1	19	24	10	26	19	-0.40	10	-7.8	19

Table 7 - DFP Optimization. Interpolated Surface, 1 and 2 iteration end point criteria. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. *All CIV values are $e10^{-5}$

From these tables we can draw some conclusions:

Optimal Starting Sizes:

1) If optimizing using either type of simplex and either %RSD or CIV as the objective function for discrete or interpolated (smooth) hypersurfaces the starting size of the simplex should be large, with the best results in this study being uniformly achieved with a starting simplex size that was 40-45% of the hypersurface. Results almost as good could be achieved using a 20-25% starting simplex size, and this would usually reduce the number of experiments required by 15-30%.

2) On the SBR discrete or interpolated surfaces both types of simplex exhibited slightly better results with a starting simplex size of 20-35% of the hypersurface size. As with the %RSD hypersurface a starting size of 20-25% could be used to achieve good results with a reduced number of experiments.

3) The DFP algorithm performed best with a starting size 20-25% the size of the response surface.

Discrete Surfaces

1) The composite modified simplex almost always performed as well or better than the Modified Simplex in terms of finding the best response, showing a particular advantage on the more complex %RSD and CIV hypersurfaces.

2) When the 10% end point criteria was used the CMS performed 15-40% more experiments to achieve this improved response.

Smooth (interpolated) surfaces

1) On the more complex %RSD and CIV surfaces the two iteration DFP optimization achieved better responses than both simplexes and used about 20% fewer experiments than the CMS and only slightly more than the modified simplex.

2) On the relatively monotonic SBR surfaces the CMS and the modified simplex both achieved better responses than the DFP, and did so while requiring fewer experiments than the 2 iteration DFP.

3) While the number of steps the simplex algorithms require to reach the endpoint can vary significantly from run to run, the DFP always requires the same number of steps (which is a function of the number of iterations being performed.)

End Point Criteria Revisited

1) For discrete surfaces the benefits of setting the end point criteria smaller than half the step size on the discrete surface are quite minimal.

2) For the DFP optimization the 2nd iteration is highly worthwhile, but further iterations are expensive and yield rapidly diminishing returns.

3) On smooth surfaces the response achieved by the simplexes invariably improves with additional experiments, and it might be advantageous to change the end point criteria to a percent improvement per experiment threshold, rather than a 5 or 10% precision threshold.

The Effect of Noise

The ICP-AES is not a noise free environment, %RSD's in the range of 0.5 to 2% are quite common, and under adverse conditions can reach higher.

In an attempt to simulate this noise, five, ten and fifteen percent noise was imposed on the surfaces to see if it impeded the optimization algorithms ability to achieve a good response, and, in the case of the simplex algorithms, incurred a penalty in terms of the number of experiments required to satisfy the endpoint condition.

	SBR Response Surface							
Noise Imposed	0% Noise		5% Noise		10% Noise		15% Noise	
Starting Simplex Size	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	18	6.0	18	5.7	18	5.7	18	5.7
0.1	19	7.8	26	17	26	16	26	16
0.15	25	15	27	19	28	18	27	18
0.2	27	18	29	20	29	20	29	20
0.25	27	19	29	21	29	20	29	20
0.3	27	19	29	21	29	21	29	21
0.35	27	20	29	21	29	21	29	21
0.4	27	20	29	21	29	21	29	21
0.45	28	21	29	22	29	22	29	22

Table 8: Modified Simplex with 0, 5, 10, and 15% noise imposed on the response surface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. A 5% end point criteria was used.

	SBR Response Surface							
Noise Imposed	0% Noise		5% Noise		10% Noise		15% Noise	
Starting Span	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps	Resp.	#Steps
0.05	22	19	22	19	21	19	21	19
0.75	23	19	23	19	22	19	22	19
0.1	23	19	23	19	23	19	22	19
0.125	24	19	24	19	24	19	24	19
0.15	24	19	25	19	25	19	25	19
0.175	25	19	25	19	25	19	25	19
0.2	25	19	25	19	25	19	26	19
0.225	26	19	26	19	26	19	26	19
0.25	26	19	26	19	26	19	26	19

Table 9 - DFP Optimization with 0, 5, 10, and 15% noise imposed on the response surface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. 2 iteration end point criteria was used.

Tables 8 and 9 show the effect of this noise on the modified simplex and DFP algorithms (CMS behaved similarly to the modified simplex). All three algorithms were relatively unaffected by the noise. It is likely that this is because the slope on the response hypersurface is, in most places, large. Relatively small amounts of noise do not impact optimization performance.

	Modified Simplex				CMS				DFP			
Starting Simplex Size	5% end point criteria		10% end point criteria		5% end point criteria		10% end point criteria		1 Iteration		2 Iterations	
	Resp.	S.D.	Resp.	S.D.	Resp.	S.D.	Resp.	S.D.	Resp.	S.D.	Resp.	S.D.
0.05	18	1.7	18	1.8	18	1.8	18	1.7	20	5.5	22	4.2
0.1	26	2.4	21	2.2	26	2.5	21	2.2	20	4.4	23	4.0
0.15	27	2.3	23	2.9	27	2.8	23	3.2	21	4.7	23	4.2
0.2	29	2.2	27	2.7	29	2.3	27	2.8	22	4.6	24	4.6
0.25	29	2.0	27	2.3	29	2.3	28	2.4	22	3.9	24	4.2
0.3	29	2.0	27	2.5	29	2.3	28	2.6	23	3.6	25	4.4
0.35	29	2.4	26	2.5	29	2.2	29	2.3	23	3.5	25	3.9
0.4	29	2.3	27	2.5	29	2.3	29	2.7	23	3.4	26	3.5
0.45	28	2.5	26	2.6	28	3.0	27	3.2	24	3.4	26	3.4

Table 10 - SBR Interpolated Surface. Starting Spans for the DFP were 5%-25% of the response hypersurface. Resp. is the average response achieved by 125 optimizations for each of the 6 emission lines. S.D. is the Average of the Standard Deviations for all six emission lines.

Variability

Another performance characteristic that should be considered is not just the average performance, but also variability in that performance. Table 10 shows that the DFP algorithm has a higher variability in its performance for the SBR surface. Results are similar for the other two hypersurfaces. Adding a third iteration to the DFP did not reduce this variability significantly.

CONCLUSION

On discrete surfaces the CMS usually achieved a better optima than the modified simplex, but took significantly more experiments to get there. On more complex smooth (interpolated) surfaces the DFP optimization usually did better than either simplex, however it's performance showed the greatest variability. For the simplexes setting the end point criteria to 5% always yielded better results than 10%, but usually at a significant cost in terms of experiments performed.

The conclusion that must be drawn is that the optimization algorithm that should be selected depends greatly on the relative importance placed on the various performance characteristics: achieving the best optima; minimizing time (and cost); and variability of results.

ACKNOWLEDGMENT

This work was funded by an operating grant from the Canadian National Sciences and Engineering Research Council. D.P.W. would like to acknowledge scholarship support from NSERC. The authors thank Thermo Jarrell Ash for the loan of the spectrometer which was used for this research.

REFERENCES:

[126] Brown, S.D.; Barker, T.Q.; Larivee, R.J.; Monfre, S.L.; Wilk, H.R., *Anal. Chem.*, **60**, 1988, p 252R.

[127] Brown, S.D.. *Anal. Chem.*, **62**, 1990, p 84R.

[128] Broekaert, J.A.C.; Leis, F. and Laqua, K., Developments in Atomic Plasma Spectral Analysis, Barnes, R.M. Ed., Heyden and Sons, London, (1980), p84.

[129] Bergman, S.S. and McLaren, J.W., *Appl. Spectrosc.*, **32**, 1978, p 372.

[130] Ebdon, L., *ICP-Information Newsl.*, **7**, 1982, p 513.

[131] Cave, M.R.; Barnes, R.M. and Denzer, P., *ICP-Information Newsl.*, **7**, p 514.

[132] Thomas, R.J and Collins, J.B., *Spectroscopy*, **5**, 1990, p38.

[133] Morgan, S.L. and Deming, S.N., *Anal. Chem.*, **46**, 1974, p1170.

[134] Betteridge, D.; Wade, A.P.; and Howard, A.G., *Talanta*, **32**, p 709, 1985.

[135] Betteridge, D.; Wade, A.P.; and Howard, A.G., *Talanta*, **32**, p 732, 1985.

[136] Fletcher, R. and Powell, M.J.D., *J. Comput.*, **6**, p 163, 1963.

[137] Werner, P. and Friege, H., *Appl. Spectrosc.*, **41**, p 32, 1987.

[138] Terblanche, S.P.; Visser, K. and Zeeman, P.B., *Spectrochim. Acta*, **36**, p 293, 1981.

[139] Moore, G.L.; Humphries-Cuff, J. and Watson, A.E., *Spectrochim. Acta*, **39B**, p 915, 1984.

[140] Carpenter, R. and Ebdon, L., *J. At. Anal. Specrom.*, **1**, p 265, 1986.

[141] Leary, J.J.; Brokes, A.E.; Dorrzapf Jr., A.F., and Golightly, D.W., *Appl. Spectrosc.*, 6, p 37, 1982.

[142] Sartoros, C. and Salin, E.D., personal communication.

[143] Boumans, P.W.J.M. and Lux-Steiner, M., *Spectrochim. Acta, Part B*, **37**, p 97, 1981.

[144] Boumans, P.W.J.M. (Ed.), ICP Emission Spectroscopy, Part 1, John Wiley and Sons, Inc, New York, p 202, 1987.

[145] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., Numerical Recipes in C, 2nd Ed., Cambridge University Press, New York, 1992.

[146] Burgess, D.D., *Anal. Chim. Acta*, **181**, p 97, 1986.

[147] Borland International, Quattro Pro User's Guide, Scotts Valley, CA, 1993.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK.

We succeeded in establishing baseline requirements for an autonomous ICP-AES instrumental system. During the course of the project a framework was developed that could accommodate our goals for an autonomous ICP and also be implemented. I think the optimization subsystem developed is production quality. Because it's written in relatively portable C++ it should be able to migrate with the project however it may evolve. Several of the other subsystems (notably line selection, instrumental diagnostics and calibration methodology selection) are good springboards for the development of production grade subsystems. The nature of the problems surrounding development of these subsystems have been clarified to varying degrees. The primary limitations on the line selection subsystem described herein is its implementation in an old DOS-based PROLOG, and the lack of a good line width model for judging overlaps. The calibration ruleset needs to be either exported from First Class to C code or implemented in C/C++ so it can be more readily integrated with the rest of the project.

The project had some unpleasant surprises at the last. There were some unexpected and severe incompatibility problems with the TSPECSA software that was used to control the Thermo Jarrell-Ash spectrometer. This prevented performing the optimization study described in chapter 7 on live response surfaces, and prevented use of the optimization subsystem once integrated into our framework. The only way to circumvent this problem is to discard the TSPECSA software entirely and develop instrumental control software over which we have complete control.

While several significant subsystems were produced in the course of developing the autonomous instrument framework presented here, several significant problems were encountered while trying to integrate the various components. Some of the components are implemented in PROLOG, several in C/C++ and even a few in Pascal. Despite

manufacturers claims to the contrary, integrating modules from these disparate tools in a DOS environment is all but impossible, and imposes severe practical constraints on the system especially in terms of memory management and intra module communication. The best solution to this problem is to abandon DOS as the target operating system for this product. An operating system that allows easier development tool independent integration of modules should be chosen. Microsoft™ Windows, Windows 95, Windows NT and IBM™ OS/2 all meet this requirement through the use of dynamic link libraries.

While a fast and functional B-Tree based database subsystem was developed for this project, the adoption of a commercial database engine should be seriously considered so that scalability and maintenance concerns do not become an issue in the event that very large databases are ever handled by this system. Most commercial database packages offer simpler database development than our custom system, and pre-built data inspection and database maintenance tools

Pattern recognition has only been addressed superficially in this project thus far. Sample recognition is a difficult problem, and this aspect of the project in particular requires a great deal of work to become a useful part of the system. Avenues that provide promise are k-nearest neighbor and Bayesian systems of pattern recognition. Neural networks have shown promise in pattern recognition and should also be investigated.

The system diagnostics examined in the course of this work address only the identification of immediate problem conditions. Additional research could be devoted to plasma diagnostics for monitoring long term instrumental performance trends. This might lead to the ability to predict and therefore prevent ICP system failures due to long term component degradation.

When I first began work on this project, I envisioned a new type of instrument that would appear and suddenly change the way ICP instruments were used in the lab. Limited optimization capabilities have already appeared on commercial ICPs. Failure prediction capabilities are starting to make their way out of the lab. I now believe that the appearance of autonomous instruments will be a more gradual process, as these new capabilities are grafted onto existing ICP system designs.

APPENDIX A - GLOSSARY OF TERMS

Backwards Chaining - Is one of the two mechanisms by which an inference engine processes its knowledge base in an attempt to answer questions. An inference engine can start with one or more hypotheses and attempt to determine which if any is correct (this is called goal-driven search.) For example, a query might be "What are professor A's research interests?" This might lead to the following chain of rule execution.

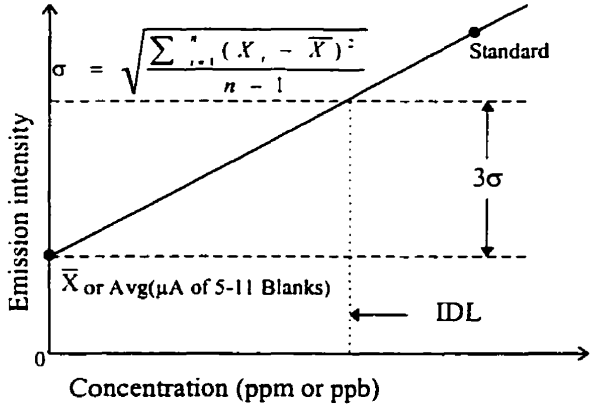
1) For each research project in our rulebase find the graduate student working on it.
2) For each graduate student found in step 1 find out if his/her supervisor is professor A.
3) *Answer:* The research projects of any of professor A's students are professor A's research interests.

Combinatorial explosion - The classic example of combinatorial explosion is the traveling salesman problem. A salesman has to visit a client in each of K cities and he wants to determine the shortest route to visit them each once. The only way to guarantee finding the best result is to try all possible routes. As K becomes even fairly large this problem becomes unsolvable.

Degradation Factor - When determining what spectral lines could be used to determine optimal spectral windows we sometimes included spectral lines which were not the most sensitive for each element. The threshold we used for line inclusion was a DF. A DF of 3 meant that elemental lines with detection limits up to 3 times that of each element's most sensitive spectral lines were included in the database used for spectral window determination. A DF of 1 would mean including only the most intense lines from each element.

Detection Limit - In this thesis, when we refer to detection limit we are referring to Instrument Detection Limit. To determine the IDL, the blank is usually analyzed 5-11 times to determine the standard deviation of the blank signal (σ). A standard of known concentration is then analyzed to calibrate the σ value in ppm or ppb. Three times this value defines the Instrument Detection Limit.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n - 1}}$$



Forward Chaining - Is one of the two mechanisms by which an inference engine processes its knowledge base in an attempt to answer questions. Forward chaining starts with a collection of facts which are known and attempt to determine whether the goal or answer can be reached given those facts (this is called data-driven search.) For example, a query might be "What are professor A's research interests?" This might lead to the following chain of rule execution.

1) The research projects of any of professor A's students are professor A's research interests.
2) Find out which graduate students have professor A as a supervisor.
3) *Answer:* Find out which projects each of the students found in step 2 are working on.

Frames - (Sometimes called schemas) Frames are a formalism for storing data and the relationships between that data. For example, a *room schema* might contain information about things that are likely to be found in a room like. Such a schema might indicate the likely presence of doors and windows, and information about those structures placement in the room, for example, windows usually do not touch the floor and doors are rarely in the ceiling or floor.

Hypersurface - A hypersurface is just a surface in any number of dimensions, where a surface is usually considered to apply only to two dimensional spaces.

Hill climbing - A hill climbing algorithm is one that uses some mechanism to determine gradient of the response hypersurface it is currently spanning and proceeds along the steepest gradient towards the optima

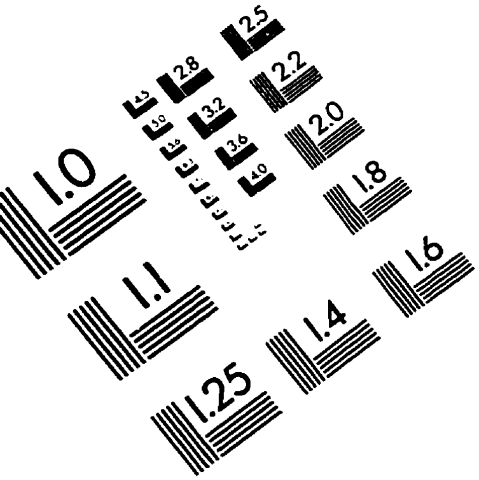
Inference Engine - An inference engine is routine which manipulates the facts and rules which are stored in the knowledge base in an attempt to reach conclusions (resolve problems). The structure of the inference engine is highly dependent on the structure of the knowledge base. For example, if the knowledge is stored in the 'well-formed formulas' of predicate calculus, then the inference engine might consist of a theorem prover. Inference engines have two mechanisms for progressing, either forward chaining or backwards chaining.

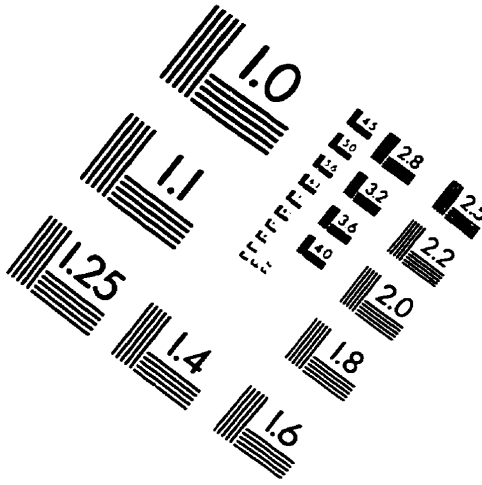
Knowledge Base - A collection of rules and facts about a problem domain. This collection of information is used by an expert system's inference engine as the basis for answering queries. A knowledge base can either be populated manually by a knowledge engineer who 'extracts' domain information from an expert, or in some instances generated automatically from raw data by induction systems like ID3 or C4.5.
Meta-Rules - In some cases a knowledge base may contain Meta Rules, which are knowledge about knowledge rules. These types of rules are used guide the inference engine's search in an effort to make it more efficient.

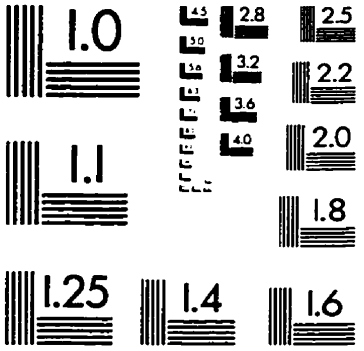
Production rules - A production rule is an IF-THEN construct for holding rules. For example:
IF (A is the son of B) THEN (B is the parent of A).
IF (B is the son of D) AND (C is the son of D) THEN B and C are siblings.

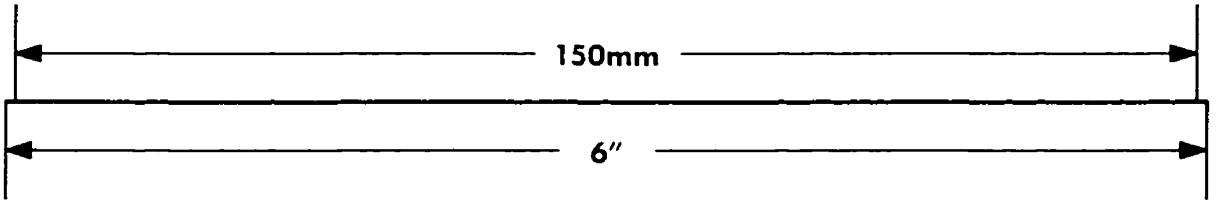
Spectral Stripping - Sometime two spectral lines appear so close together that they cannot normally be spectrally resolved using commercially available spectrometers. For example, the Cd 214.483nm and the Sb 214.486nm lines would appear to be indistinguishable. If in our polychromater the only line at we could use to determine the Cd concentration was 214.483, and if the Sb concentration is high enough to effect the apparent intensity at the Cd 214.483 line we could determine the contribution from Sb by determining the intensity of another Sb line. Using documented line intensity ratios we could then 'strip' away Sb's contribution at 214.486 by subtracting the intensity as determined by ratio.

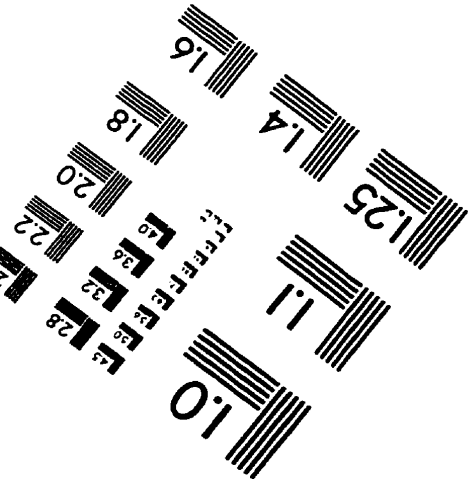
IMAGE EVALUATION
TEST TARGET (QA-3)

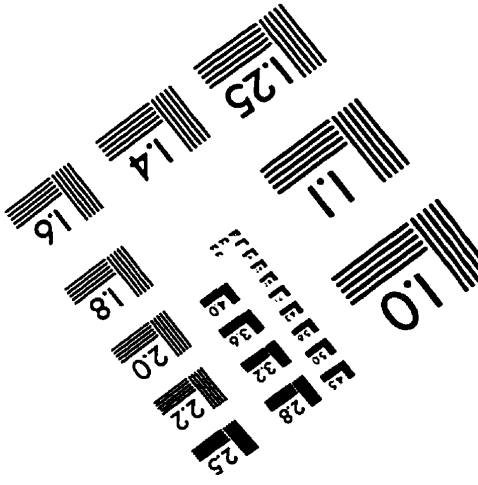












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989