A SYSTEMATIC PROFILE/FEATURE-BASED INTELLIGENCE FOR SPECTRAL SENSORS

Michael C. Vogt

Energy Systems Division
Argonne National Laboratory
Argonne, IL 60439

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

October 2000

Submitted to SPIE Intl. Symposium on Environmental & Industrial Sensing, Boston, MA, Nov. 5-8, 2000, sponsored by SPIE.

*Work supported by the U.S. Department of Energy under Contract W-31-109-Eng-38.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

A systematic profile/feature-based intelligence for spectral sensors

*Michael C. Vogt

Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL, 60439 USA

ABSTRACT

Argonne National Laboratory (ANL) has been creating a special-purpose software-engineering tool to support research and development of spectrum-output-type [chemical] sensors. The modular software system is called SAGE, the Sensor Algorithm Generation Environment and includes general-purpose signal conditioning algorithms (GP/SAGE) as well as intelligent classifiers, pattern recognizers, response accelerators, and sensitivity analyzers. GP/SAGE is an implementation of an approach for delivering a level of encapsulated intelligence to a wide range of sensors and instruments. It capitalizes on the *generic* classification and analysis needed to process most profile-type data. The GP/SAGE native data format is a generalized one-dimensional vector, signature, or spectrum. GP/SAGE modules form a computer-aided software engineering (CASE) "workbench" where users can experiment with various conditioning, filtering, and pattern recognition stages, then automatically generate final algorithm source code for data acquisition and analysis systems.

SAGE was designed to free the [chemical] sensor developer from the signal processing allowing them to focus on understanding and improving the basic sensing mechanisms. The SAGE system's strength is its creative application of advanced neural computing techniques to response-vector and response-surface data, affording new insight and perspectives with regard to phenomena being studied for sensor development.

Neural networks, sensors, algorithms, chemical, spectrum-response

RECEIVED

NOV 08 2000

OSTI

1. INTRODUCTION

1.1 Origins from support technologies

ANL's Cermet Microsensor program has investigated and developed many support technologies to compliment the [physical and chemical] sensors and instruments it has produced. These support technologies include special-purpose microelectronics, advanced [including nano-technology] fabrication methods, and a systematic approach to "intelligent" sensor signal processing¹.

A popular goal of both basic and applied research is to identify new and improve upon current physical phenomenon used as the heart of a chemical sensor². However, significant improvements can be made to the usefulness of existing sensors by simply improving upon the sensor *signal processing*. The availability of high-performance low-cost data acquisition systems has changed the development of sensors by helping to remove artificial constraints on the types and numbers of signals collected and used from the sensor³. Many researchers are expanding the numbers and type of parameters they employ for differentiating chemical signatures^{4,5}, even identifying some past methodologies as now appropriate for "universal detectors"^{6,7}. A common property observed in most of these sensors and instruments has been a *vector* or *profile*-type output response as shown in Figure 1.

Figure 1. Example chemical sensor response vectors

* Correspondence: Email: vogt@anl.gov, WWW website for Energy Systems Division, <http://www.es.anl.gov>

In these vectors, common *measured* parameters [ordinates] include induced electrical current, absorption, or impedance while common *reference* parameters [abscissa] include applied potential, spectral wavelength, or frequency. The vectors can be one-dimensional with an implicit abscissa, or two-dimensional with an explicit abscissa. The response profiles display comprehensive, full-duration, components as well as features that represent specific, short-duration, reactions⁸. The different chemical reaction phenomenon responsible for features in different responses are not as important as the similarity displayed in the features they produce [a peak is a peak, a slope change is a slope change].

All of these disparate signal responses and components benefit from similar types of processing, but tools for this type of processing are few, proprietary, and often specialized⁹. To address this need for tools a suite of modular processing algorithms has been developed, collected as an approach, and deployed in a special-purpose computer aided software engineering (CASE) tool. With this tool, the sensor researcher can be well-staged to generate an appropriate signal processing algorithm quickly, allowing them to focus on the physical improvements.

1.2 P/F-BI approach and SAGE

The approach is called Profile/Feature-Based Intelligence (P/F-BI) and the software tool is called the Sensor Algorithm Generation Environment (SAGE). P/F-BI is a strategy to systemize the identification and use of features in a sensor's response profile (a single vector or multi-vector surface) to provide response analysis and to provide an external monitoring and/or control system with significant additional information regarding conditions experienced by the sensor. This intelligence is encapsulated in an interface making many already available sensors "smarter". Smart sensors have been recognized as an important benefit to process controllers because they remove the burden of individual device calibration and drift correction from the upper level controls. Smart sensors promote simpler and more robust systems with more manageable and dependable components. While the P/F-BI and SAGE approach is applicable to common process control temperature and pressure-type mechanical sensors, its development has been focused more on difficult research areas where sensor advances have been slow, that of chemical sensors.

1.3 ChemSAGE

ChemSAGE combines a suite of neural-based signal processing algorithms, along with a strategy for their application, to chemical sensor response data. ChemSAGE has been demonstrated with data from a variety of sources including Fourier transform infrared (FTIR) spectroscopy, gas chromatography (GC), mass spectrometry (MS), voltammetry (CV), neutron activation analysis (NAA), and is suitable for most types of spectrometric analyses. The SAGE strategy has also been scaled to multi-dimensional data and has been used to generate algorithms for multi-spectral and hyper-spectral *imagery* (MSI and HSI) processing.

While P/F-BI has been used for chemical sensor signature processing, it may be more readily introduced with a more common sensor such as a temperature probe used as a thermometer.

1.4 Oven thermostat example

In a system such as an oven controller, the 'state' of the oven can be observed from a single measurement and the heating adjusted up or down accordingly to maintain a desired temperature. Decades of experience in engineering has shown that significant improvements can be made to the way heating is applied if the history and rate-of-change of the oven temperature is known.

Controllers of this kind are often referred to as PID (proportional-integral-derivative) controllers. PID controllers produce feedback components for a control system as a function of the signal's own strength, based upon a signal's past response, and based upon a sensor's response rate-of-change. The integral and derivative components of the signal depend upon having a record of the sensor's [recent] past response. The historic trends and instantaneous rate-of-change of the oven temperature can also be described by "features" in the temperature time-response profile. Similar feature and feature-relationship information is available in spectrum-type signals where one parameter is measured with respect to another controlling variable. The controlling variable is often time but can be wavelength, voltage, or frequency. The main requirement is that the signal profile be captured (digitized) in a vector for subsequent processing. The sampling window can be moved along the controlled variable axis, continuously updating the vector being processed. With A/D devices and CPUs operating at multi-megahertz sampling speeds, even large windows can be processed quickly enough to supply real-time error-correction and control signals. Thus, extracting integral and derivative control information becomes a pattern recognition problem.

Furthermore, important additional features such as threshold crossings, excessive rates-of-change, and repeated behaviors can also be observed and identified lending the monitoring controller the ability to take even more complex and intelligent control actions to prevent accidents or erroneous commands. For example, exceeding the desired heating rate for a ceramic object in a kiln. Problems like this represent a continuous output taken a window at a time and are solved by sliding the windows and repeating the pattern recognition.

1.5 Medical thermometer example

A different problem that illustrates a similar solution would be a medical thermometer. The common use of the thermometer is to capture a single simple measurement and use it to test for threshold crossings. But much more complete systemic diagnosis could be supported by capturing that same patient temperature over some period of time, even a short one, or through the course of physical activity.

This temperature profile would provide the patient's reaction to a significant number of bodily functions and conditions such as digestion, sleep, and over-exertion, allowing for a review of health with respect to average temperature, recovery rate, and time and duration of normal bodily functions. The identification and classification of rises and falls in body temperature are again, simple pattern recognition problems, and shapes, including rates-of-change and integrated areas are indicative of specific events and can be/should be used to augment diagnoses. Problems like this generate a response vector as a one-time snap-shot and, if obtained in a rigorous standardized fashion, can be processed and categorized and/or analyzed as a whole.

Pattern recognition routines and advanced signal processing as described earlier are regularly included in high performance control systems, but such systems are often not available nor appropriate for developmental or experimental sensors and instruments. This can extend seconds worth of processing into many person-hours. Such unneeded delays handicap sensor development and stunt the associated research.

This paper will describe the major components in the SAGE workbench and will illustrate these with a sample of different data types and applications.

2. APPROACH AND IMPLEMENTATION

2.1 Introduction to the SAGE environment

The Sensor Algorithm Generation Environment is a collection of signal processing algorithms coded, ported, or interfaced into a single environment to aid chemical sensor researchers in standardizing their signal processing. The system began in a UNIX environment with modules written in C/C++ and FORTRAN. The current implementation is rewritten in the MathWorks MATLAB environment version 5. It is portable and operates in UNIX (SGI IRIX) and in Microsoft Win95/98/NT/2000. Few changes will be required to port to a MacOS environment as well. The system integrates components from commercial MATLAB extensions, ported legacy code, and interfaced specialized routines. Users can experiment with the application of various pre and post processing routines, filters, and pattern recognition modules. The standard signal handled is a 1-D vector representing FTIR, GC, MS, MSI, HSI, CV, NA and AA, or any other similar formatted sensor signal.

2.2 SAGE user's guide

The *SAGE Users' Guide and Programmers' Reference Manual* describes both the design and operation of the main modules that make up the SAGE environment. It is further supported by a set of case studies that illustrate the types of data for which SAGE has currently been designed to develop algorithms. The modules represent the common stages in signal processing. They include a Preprocessor module, a Digital Filtering module, a Pattern Recognition module, and a Multivariate Calibration module.

2.3 SAGE theory of operation

In the current implementation, SAGE is a collection of graphical user interface panels with menus and buttons¹⁰. The user fills in various visual control boxes with values and can examine their partially-transformed data before committing the settings. Completing each stage writes appropriate MATLAB scripts off to an algorithm file where the script can be reviewed and/or exported as C/C++ code [Java soon].

2.4 SAGE preprocessor module

Although there is no set data format for SAGE input, it is expected that the data conform to a 1-D vector shape. It can have any number of sample points, but only can be single dimensional. This does not represent much of a restriction as the standard chemical data from most instruments fits this easily. Shown in Figure 2 is a sample of data types used in SAGE demonstration analyses during its initial development.

Figure 2. Vector-based chemical data used during SAGE development

Options exist (shown in Figure 3) to allow moderate data transformations such as cropping, scaling, normalizing, re-sampling, and simple mathematical manipulation of the signals with the option to save the transformed signal under a different file designation extension.

Figure 3. SAGE preprocessor module GUI

2.4 Digital filtering module

The Digital Filtering Module shown in Figure 4 allows the user to design and apply several classic Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) digital and analog filters to gathered signals. It uses a graphical environment to poll the users for design criteria, then displays the original wave form, the applied filter, and the filtered wave form. Classic linear

phase FIR filters include FIRLS, Remez and Kaiser. Classic IIR filters include Butterworth, Chebyshev I & II, Bessel, and Elliptical. These can be further divided into high pass, low pass, band pass, and band reject types. The filter specifications are set using editable fields where sampling rate, passband and stopband frequencies, amount of ripple in the passband, and attenuation in the stopband are adjusted. The classic filters are made available to model analog filtering that might be applied to some sensor designs. (For sensors that produce strictly digital data, these analog approximations of ideal filters are unnecessary.)

Figure 4. Digital filtering module GUI

Figure 5. FFT sub-module

sized problems. RBFNs can be built as generic function approximators, but generally require more memory and execution time than BPNs in exchange for shorter learning times and accuracy.

Probabilistic Neural Networks

The inclusion of the probability distributions in the processing, and the effective application of a nearest-neighbor search in the pattern layer, make the PNN a neural network implementation of the classic Bayesian classifier. The Bayesian classifier has the advantage of a rigorous mathematical defense and validation. The PNN is strictly a classifier network. The modifications needed to allow analog output far outweigh the benefits and complicate the mathematical defense. SAGE's original PNN implementation preceded the availability of PNNs in the current MATLAB Neural Network Toolbox. The code was ported over version of an academically-developed PNN written in C++. This version has been replaced in favor of uniform MATLAB CNN implementation.

2.6 Multivariate calibration module

At the time of this writing, the only operational multivariate calibration is integrated into the neural pattern recognition module with the analysis options. Both the BPN and RBFN types allow for binary outputs (suitable for classification only problems) or full linear output values that can be trained for (calibrated) analog numeric values. It is common to not introduce unneeded work into separating the calibration from the pattern recognition in these algorithms.

3. EXAMPLE [RESULTS]

Interaction with much of the SAGE environment can be best explained through examples. The following cases introduce the SAGE interfaces and illustrate the types of data that have been worked with during SAGE development.

3.1 Preprocessor and digital filtering modules examples

In the Preprocessor and the Digital Filtering Modules a standard file-selection scrollable window can be used to navigate the file system to identify the desired vector file to be processed. SAGE modules expect a 1-D signal vector (or collections of vectors as a surface) with no constraints on length. When the file is selected, then the desired display window can be chosen and the plotting color (red, green, blue, etc.) set. The signatures can be re-sampled, off-set, scaled, and normalized. Filters can be built and applied, allowing the user to inspect both the filter and the sample data vector. The Preprocessor and Digital Filter Modules are shown in Figures 7 and 8.

Figure 7. Preprocessor module with neutron activation data, zoomed-in and re-sampled

Figure 8. Digital filter module with band pass filter and FTIR data

3.2 Pattern recognition examples

Several sub-modules are available to provide pattern recognition with and without calibrated analysis for a given type of data. Samples shown have been processed for both *chemical* and *spatial* information content with the hyperspectral data (Figure 9) cast back into a spatial format to illustrate future applications.

Figure 9. RBFN sub-module set up for classification network (with HSI data)

Additional examples of hyperspectral imagery illustrate the future direction SAGE is being taken. Algorithms developed by the SAGE environment can be ported to existing spatial analysis software to allow the chemical analysis to be part of the spatial analysis and classification of an image. This is appropriate because the data, in the form of spectra that serves as chemical signature input to SAGE algorithms, often comes from remotely sensed imagery. It is a natural progression then to develop not only automatic chemical spectra classification algorithms, but to present that new data as similarly formatted imagery from which a scientist or analyst can make decisions. An example of work already completed is illustrated in Figure 10.

Figure 10. SpatialSAGE algorithm for spectral/spatial classification

4. FUTURE RELATED WORK

Early P/F-BI and SAGE work involved integrating legacy signature processing code with a selected commercial development environment (MATLAB). Current improvements have been made to many of the MATLAB toolboxes that allow them to replace the non-MathWorks legacy code. Additionally, MathWorks has added several new toolboxes including a Data Acquisition Toolbox that completes the systems and allows full cycle data acquisition and analysis algorithms to be generated. The 2001 release of ChemSAGE will include the data acquisition capability and will target code generation in C/C++ for high-speed DSP boards as well as Java for select microcontrollers. The next release of SAGE will also introduce an additional block-diagram type interface allowing the user to inspect the final algorithm in that format.

5. REFERENCES

1. Vogt, M. C., C. M. Klaus, M. Poulos, S. Crofts, 1997, *Sensor Algorithm Generation Environment (SAGE) Users Guide and Programmer's Reference Manual*, prepared by Argonne National Laboratory for the Department of Energy
2. Fraden, Jacob, 1993, *AIP Handbook of Modern Sensors, Physics, Designs, and Applications*, American Institute of Physics, New York, NY, pg. 535
3. Smyth, Malcom R., J. G. Vos, 1992, *Comprehensive Analytical Chemistry-Analytical Voltammetry*, Vol XXVII, Wilson and Wilson, Elsevier Science Publishing Company Inc., New York, NY, pg. 20, 34, 59
4. Kumazawa, Noriyuki, M.R. Islam, and M. Takeuchi, 1999, "Photoresponse of a Titanium Dioxide Chemical Sensor", *Journal of Electroanalytical Chemistry*, Vol. 472. Pp. 138-140
5. Diaz-Cruz, Jose Manuel, R. Tauler, B. S. Grabaric, M. Esteban, and E. Casassas, 1995, "Application of Multivariate Curve Resolution to Voltammetric Data. Part 1. Study of Zn(II) Complexation with Some Polyelectrolytes", *Journal of Electroanalytical Chemistry*, Vol. 393, pp. 7-16
6. Radomski, R., M. Radomska, M., Dankowski, K. Szajowska, and Z. Wisialski, 1995, "Microcomputer-Controlled Electrochemical Universal Meter", *Computer and Chemistry*, Vol. 19, No. 3, pg. 303
7. Salikhdzhanova, R. M.-F., A. I. Gorobets, and N. Ya. Petrova, 1994, "Automation of Polarographs and Voltammetric Analyzers", *Measurement Techniques*, Vol. 37, No. 7, pp. 835-838
8. Bard, Allen J., L.R. Faulkner, 1980, *Electrochemical Methods*, John Wiley & Sons, Inc, New York, NY, pp. 232-236
9. *BAS, HANDBOOK OF ELECTROANALYTICAL PRODUCTS*, 1997, Bioanalytical Systems Inc., West Lafayette, IN
10. *Graphics and GUIs with MATLAB*, 1996, Marchand, P., CRC Press, Boca Raton, FL
11. Stearns, S., R. David, 1996, *Signal Processing Algorithms in MATLAB*, Prentice Hall PTR, Upper Saddle River, NJ
12. *Signal Processing Toolbox Users Guide*, 1998, MathWorks, Inc., Natick, MA, pp. 2-2
13. *Wavelet Toolbox Users Guide*, 1997, MathWorks, Inc., Natick, MA, pp. 1-11
14. Freeman, J., D. Skapura, *Neural Networks, Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publishing Company, reading, MA, 1991, pp. 89-111
15. *Neural Computing Documentation Set*, 1995, NeuralWare, Inc, Pittsburgh, PA, pp. 83
16. *Neural Computing: A Technology Handbook for Professional II/Plus and NeuralWorks Explorer*, 1995, NeuralWare, Inc., Pittsburgh, PA, pp. 1-8, 63-86, 235-275
17. Masters, Timothy, 1993, *Practical Neural Network Recipes in C++*, Academic Press, Inc, Boston, MA, pp. 174-185
18. *MathWorks MatLAB Documentation Set (ver 4, ver 5)*, 1996, MathWorks, Inc., Natick, MA
19. *Neural Networks Toolbox Users Guide*, 1998, MathWorks, Inc., Natick, MA, pp. 10.7-10.10

This work was supported by the U. S. Department of Energy under Contract W-31-109-Eng-38.