

Bogdan Wilamowski, John Regnier, Aleksander Malinowski

wilam@ieee.org, regnier@uwyo.edu, olekmali@uwyo.edu

Dept. of Electrical Engineering  
University of Wyoming  
Laramie, WY 82071, USA  
Phone: (307) 766 2162 Fax: (307) 766 4444

**Abstract** - Computer network communications have become a popular and efficient means of computing. Most companies and research institutions use networks to some extent on a regular basis. This paper demonstrates the capability of computer-aided design (CAD) through computer networks. There are several benefits that make networked CAD tools desirable. A networked application can be used remotely through any network connection. Any operating system can be used to access a networked application, making the application operating system independent. Also the application can be made to run on a pay-per-use basis if licensing is desired, and much less installation time and configuration time is required because the application is located on one central machine. The Spice Intranet Package (SIP) is demonstrated as an example of such a networked CAD application. This particular application allows users to run SPICE simulations and view graphical analysis of electronic circuits through a network connection using a Web browser. The SIP application has many options that include simulation and analysis of Spice files, graphical analysis of data, online editing of Spice files, passwords with separate file areas for each user, and a user friendly graphical user interface.

## I. INTRODUCTION

Computer networks have become a popular and efficient means of computing. The following paper describes an actual implementation, and the programming technology used to develop a CAD tool, which can be used remotely through computer network connections [1]. The Spice implementation used in this presentation is just one example of a networked application. Several methods of computer network programming are available including Java and CGI (Common Gateway Interface). The focus here will be the advantages of network programming when used for Spice simulation and analysis. An application called the Spice Intranet Package (SIP) has been developed for use through Internet and Intranet networks. Internet web pages are available from all around the world, while Intranet networks provide connectivity for a smaller, more isolated domain like a company. The SIP provides an operating system independent interface which allows Spice simulation and analysis to be performed from any computer that has a web browser on the Internet or an Intranet. The SIP has a user-friendly GUI (Graphical User Interface) and features include password protection and user accounts, local or remote files for simulation, editing of circuit files while viewing simulation results, and

analysis of simulated data in the form of images or formatted text.

Several network-programming tools, which are available today, include Java, CGI, ActiveX, JavaScript, VBScript, HTML, and PERL. During software development it is important to justify which part of the software should run on the client machine and which part should run on the server. The CGI is quite different from writing Java applets in this aspect. Applets are transferred through a network when requested and execution is performed entirely on the client machine that made the request. In CGI much less information has to be passed to the server and the server executes instructions based on the given information and sends the results back to the local machine that made the request. Fig. 1 shows the program component division and data flow in a network based application.

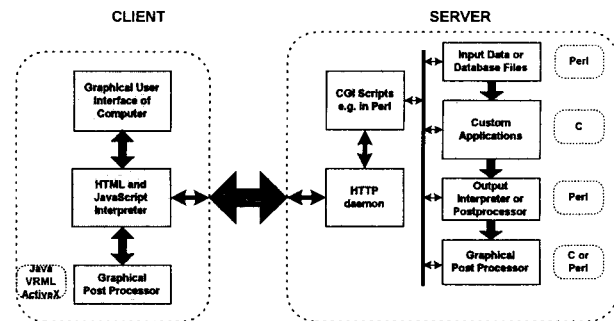


Fig. 1. Data flow in a network-based application.

In the case of the Spice Intranet Package it only makes sense to use CGI for the Spice simulation because it would be impossible to send the Spice engine through the network every time it was requested and this would be extremely slow. Java technology could also be used for functions like generating and manipulating graphs and implementing the graphical user interface on the client side. The SIP program currently incorporates CGI, PERL, HTML, and JavaScript. These technologies are briefly described below.

The CGI programming allows for dynamic web page generation in a web browser based on user selections in the initial page displayed in the web browser. Communication between a CGI program and the web browser is accomplished through a network connection between the web browser and a computer running an HTTP (Hypertext

server which it receives a request to process information from a web browser. The server then decides if the request should be granted and if the CGI program actually exists. If the authorization is secured the server executes the CGI program and returns the results to the web browsers that requested the processing.

PERL is a programming language especially suited for CGI network programming and commonly used for text processing and system shell programming. Sometimes a PERL program is called a PERL script because it is an interpreted language and is not compiled like a C program. A PERL script is similar to a Unix shell program. PERL has many features that make it well suited to processing CGI requests and generating HTML pages. HTML and JavaScript provide the front-end graphical user interface that allows a user to click a button or enter a circuit filename and start circuit simulations. Events in JavaScript are monitored and processing of information is initiated through features of these two technologies.

## II. SIP DESCRIPTION

The SIP software package is a Spice simulation and analysis program implemented in the network environment. A unique feature of the SIP program is that it is operating system independent. Anyone that has access to the Internet and a web browser, such as Netscape Navigator or MS Internet Explorer, can run a Spice simulation and view the results graphically from anywhere in the world using any operating system. A network model of the SIP software package is shown in Fig. 2.

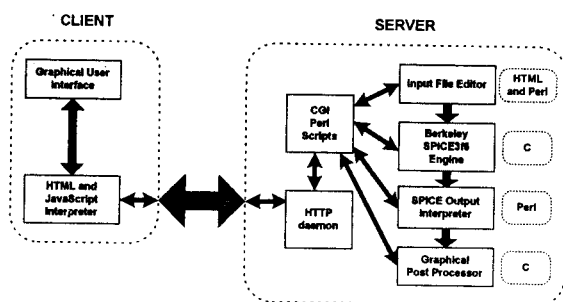


Fig. 2. Spice intranet package network model.

A server is configured to accept requests from web browsers through network connections. The server processes the request for Spice simulation or analysis and returns the results to the requesting web browser as an HTML document. The graphical analysis is embedded in the HTML as an image or returned as formatted text. Fig. 2 shows the flow of information and the distinction from the client and the server. Some of the SIP features include:

- Simulation and analysis of Spice files stored on the server or on your local machine.
- Graphing of DC/AC/Transient analysis data from a remote or local file. The data used for graphing is generated with the Spice ".PRINT <AC|DC|TRAN> ..." command.

- and scale.
- One copy of the Spice engine runs on a server and many users can access the SIP program simultaneously.
- Password protection and separate file areas for each user password.
- Editing of personal input and output files stored on the server.
- Multiple windows open at a time to allow viewing or editing of circuit files while viewing simulation results.
- Analysis output can be specified as a GIF image, or raw text containing the data points.

The current versions of SIP are written both for the Unix and the Windows NT operating systems. (Respectively: <http://atlantis.uwyo.edu/~regnier/sip> and <http://nn.uwyo.edu/sip-html/>.) The two versions differ only slightly because of the operating system independence of the programming languages being used. PERL for example is available for Windows NT as well as the Unix system and JavaScript and HTML are totally platform independent. The image generation is currently done with the gnuplot program and netpbm utilities.

## III. SIP EXAMPLES

The graphical user interface is shown in Fig. 3. It is necessary to select the radio button for **REMOTE** or **LOCAL** files depending on which you are using. This decision specifies whether the simulation and graphing programs will use the remote or local file names that you entered. If you use local files you have to save the output from simulation in a file on your local machine, and then enter that name in the **Local OUT Filename** edit box in order to plot the data. As an example of the use of the SIP program, consider the MOS Inverter in Fig. 4.

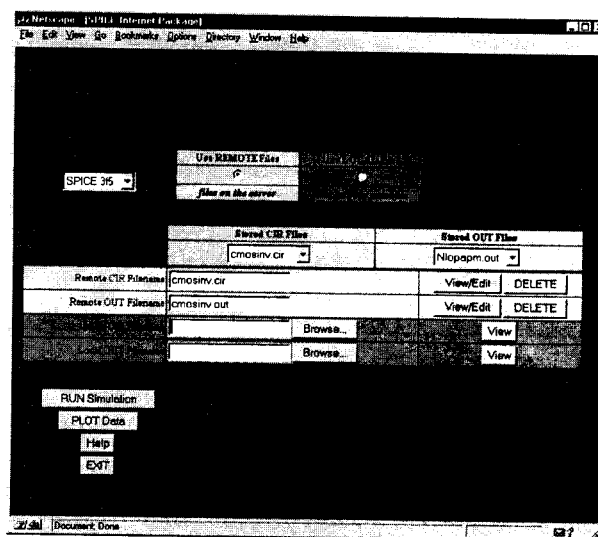
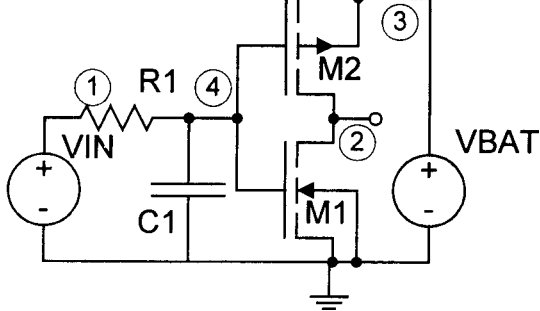


Fig. 3. Graphical user interface for SIP package.



```

Simple CMOS Inverter
.MODEL NMOSMOD NMOS LEVEL=2 VTO=0.75
+ KP=25u GAMMA=0.75 PHI=0.6 LAMBDA=0.03
.MODEL PMOSMOD PMOS LEVEL=2 VTO=-0.75
+ KP=8u GAMMA=0.75 PHI=0.6 LAMBDA=0.03
M1 2 4 0 0 NMOSMOD L=3u W=10u
M2 2 4 3 3 PMOSMOD L=3u W=20u
R1 1 4 100k
C1 4 0 1pF
VBAT 3 0 DC 5V
VIN 1 0 AC 1mV DC 2.3V
+ EXP(0.5 4.5 25u 8u 80u 8u)
.TRAN 0.5u 150u 0 1u
.PRINT TRAN V(1) V(2)
.DC VIN 0 5 0.05
.PRINT DC V(1) V(2)
.AC DEC 5 1k 10MEG
.PRINT AC VM(1) VM(2)
.END

```

Fig. 4. Simple CMOS inverter: circuit diagram (upper) and editing window with Spice input file (lower).

The Spice3 input file in Fig. 4 describes the MOS circuit in terms of the Spice3 language. The Spice2 and Spice3 input code significantly differs from other popular Spice versions as PSPICE for example. All differences between various Spice programs are well described in [2]. Notice in the Spice code there are commands for the Spice simulator to analyze the dc sweep, the ac analysis, and the transient response of the circuit and to print the output of V(1) and V(2).

To enter a Spice input file select the filename from the CIR drop down box or type the name into the *CIR Filename* edit box and then press the **VIEW/EDIT** button. After entering or modifying the input file press **SAVE CHANGES** in the edit window to save the changes to the file. After entering the input file and saving it, the simulation is then run by selecting the **RUN SIMULATION** button. The output from the simulation is displayed in the output window. If remote files were used the output is also saved to the file in the *Remote OUT Filename* edit box automatically. Next generate a graph of the output data by pressing the **PLOT DATA** button. A window will open allowing you to select what variables you want to plot and other customization variables as shown in Fig. 5. Select **PLOT DATA** again and the analysis is returned to the output window as shown in Fig. 6. To close all the windows and return to the original web browser window press the **EXIT SIP** button. There is

program. It is also possible to save the output data in a space delimited ASCII file so a high quality graph can be generated by MS Excel or other plotting packages.

Fig. 5. Plotting configuration screen.

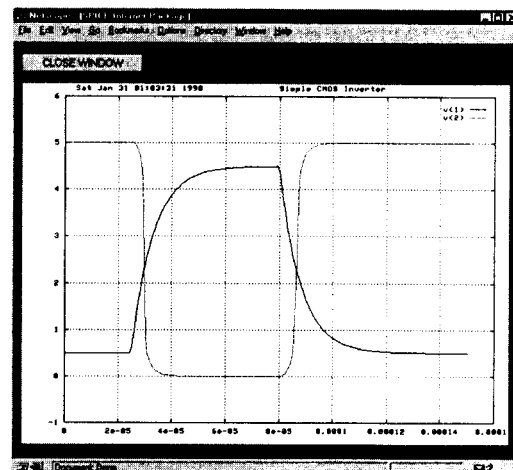


Fig. 6. Results of a transient analysis of CMOS inverter.

OPAMP shown in Fig. 7, which uses the macrodome of OPAMP as described in [2]. Also in this case all three types of analysis are performed. Fig. 8 shows both the transient response of the amplifier and the AC analysis. Note two types of distortions one due to rail voltage limitation and another due to the slew rate limitation. One unique feature is ability to copy the resulted graphics into reports or documents as it is demonstrated in Fig. 8. The SIP package uses the compressed GIF graphics formats, which are small and handy in comparison to the large bitmap files used in other Spice packages.

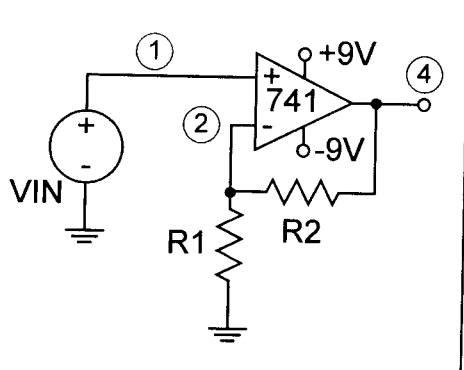


Fig. 7. Application of a non-ideal operational amplifier.

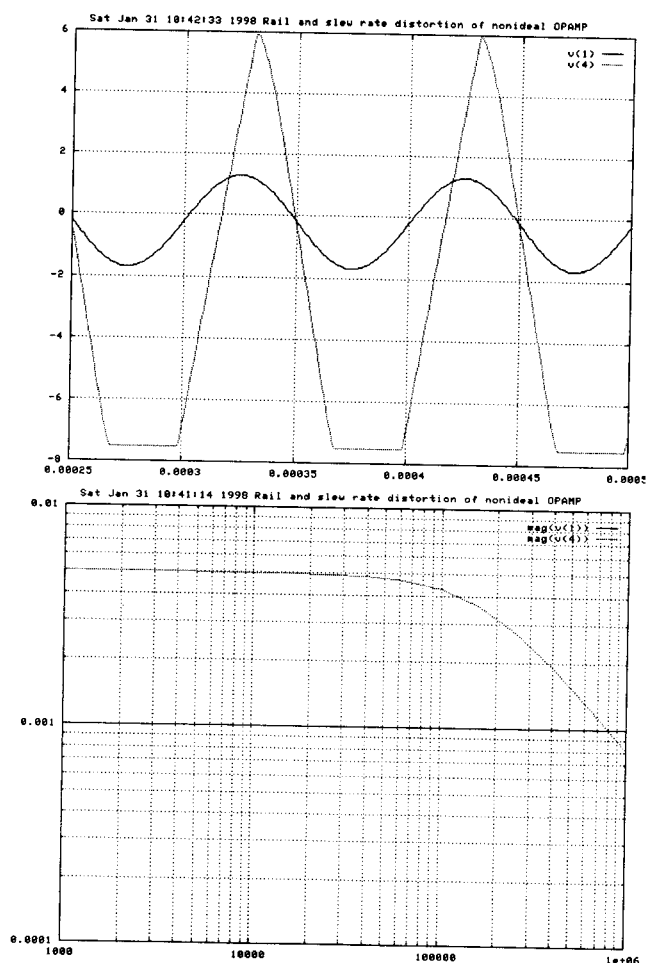


Fig. 8. Results of analysis of non-ideal operational amplifier: transient analysis (upper) and ac analysis (lower).

## V. CONCLUSION

The purpose of this presentation is to show how CAD tools can be used through the Intranet, and the SIP is just an example of using Spice. Several features make the Spice Intranet Package a desirable program for computer-aided engineering and design. Only one copy of the Spice engine needs to be installed and configured. One machine acts as the server and other machines can simultaneously access the Spice engine through network connections. Remote access to SIP allows users to run Spice simulations from any computer on the network, and that might be from home or another office in another building or town. Also the current Spice engine being used is Spice3f5 from Berkeley [4] which allows an unlimited number of transistors, unlike various "student versions" of Spice programs that are available.

User passwords are required because the SIP program is accessible from around the world and a heavy number of users would slow the server down noticeably. For undetermined time you can access the Spice Intranet Package and try some examples with the password "sip" at the following URL:

- <http://atlantis.uwyo.edu/~regnier/sip>
- <http://nn.uwyo.edu/sip/>

The Spice Intranet Package is successfully used in several classes at the University of Wyoming. Students appreciate a convenience of using it on any computer platform at the University or at home. It was designed originally for the Intranet, however it works also very well on Internet.

## VI. REFERENCES

- [1] Sweet W., Geppert L., "http:// It has changed everything, especially our engineering thinking", *IEEE Spectrum*, January 1997, pp. 23-37.
- [2] Wilamowski B. M. and Richard C. Jaeger, *Computerized Circuit Analysis Using Spice Programs*, McGraw-Hill, 1997.
- [3] Vladimirescu A., K. Zhang, A. R. Newton, D. O. Pederson, and A. SangiovanniVincentelli, *SPICE Version 2G6 User Guide*, Department of Electrical and Computer Science, University of California, Berkeley, CA, 1981.
- [4] Quarles T., A. R. Newton, D. O. Pederson, and A. SangiovanniVincentelli, *SPICE3 Version 3F5 User's Manual*, Department of Electrical and Computer Science, University of California, Berkeley, CA, 1994.
- [5] Mall L., T. Christiansen, and R. L. Schwartz, *Programming PERL*, O'Reilly & Associates, 2nd edition 1996.

# VLSI analog multiplier/divider circuit

Bogdan M. Wilamowski

Dept. of Electrical Engineering, University of Wyoming  
Laramie, WY 82071, USA  
wilam@uwyo.edu

**Abstract** - Intelligent systems require signal processing which can be done quickly and accurately. The commonly used digital approach has many advantages, but some disadvantages as well. Analog signal processing is in many cases much simpler, faster, and easier for performing parallel processing. In this paper a new accurate CMOS multiplier/divider circuit is presented. In contrary to previous solutions, which use the square law formula for MOS transistors operating in the strong inversion mode, the multiplier/divider circuits employ MOS transistor characteristics in the subthreshold conduction mode. It is shown that the circuit generates very accurate results. With some modification, the basic circuit can also be used as a four-quadrant multiplier, which may operate for both positive and negative input signals. The circuit itself may have many applications as multiplier, divider, signal squaring, square root calculations, frequency modulation, frequency doubling, etc. Most importantly it can be used for analog signal processing in many intelligent systems of industrial electronics.

## I. INTRODUCTION

Numerous applications of industrial electronics use intelligent computation. For example, many motor control systems require sophisticated computation. The intelligence is also involved in smart sensors, which are able to measure flux, and other electrical parameters just by analyzing currents and voltages on the supply terminals. Recent publications about intelligent system applications into industrial electronics are devoted mainly to the concept and sometimes simulation issues. Those publications, which describe hardware implementation, use computers or microcomputers for intelligent computation. The digital approach has many advantages, primarily it is flexible and easy to reprogram. At the same time these digital systems are rather complex and they require an analog to digital conversion at the front of the system and a digital to analog conversion at its end. Our surrounding world has an analog nature and it would be wise to perform all computation in analog fashion. The recent development of several new analog signal-processing circuits makes it so the analog signal processing becomes more visible for practical applications. Analog signal processing is usually much faster. Several computation processes can be done simultaneously and AD and DA conversion is not required.

Many analog processing units have been developed already. The operations which are the most difficult to implement are signal multiplication and division. Several solutions have been implemented already primarily using bipolar technology. The best known of these is the Gilbert four-quadrant multiplier.[1][2]. Many attempts have been made to construct an analog multiplier using the CMOS technology [3][4][5][6][7]. All those approaches assume the ideal characteristics of a simple Shichman-Hodges MOS transistor model where the drain current is given by:

$$\text{for } V_{DS} \leq V_{GS} - V_{th} \\ I_D = K [(V_{GS} - V_{th})V_{DS} - 0.5V_{DS}^2] (1 + \lambda V_{DS})$$

$$\text{for } V_{DS} \geq V_{GS} - V_{th} \\ I_D = 0.5K (V_{GS} - V_{th})^2 (1 + \lambda V_{DS}) \quad (1)$$

where

$$K = \mu C_{ox} \frac{W}{L} \quad (2)$$

Figure 1 shows the measured characteristics of NMOS transistors fabricated using 2 $\mu$ m MOSIS technology. Fig. 1(a) presents the drain current versus gate-source voltage in a linear scale and Fig. 1(b) shows the same characteristics drawn in a logarithmic scale. On Fig. 1(a) a parabolic function  $0.2(V_{GS} - 0.8)^2$  is also plotted with a dotted line. Note that the actual transistor characteristics for  $V_{DS} > V_{GS} - V_{th}$  have a different than parabolic shape. Therefore, the transistor model described by equation (1) is not accurate and all CMOS multipliers developed using this model are questionable.

## II. PROPOSED APPROACH

In this approach, instead of using the "square law" characteristics of MOS transistors in the strong inversion mode, the subthreshold conduction mode is used instead. Note on Fig 1(b) that the transistor characteristics for currents smaller than 10 $\mu$ A follows the exponential relationship for more than 6 decades. This is especially true for transistors with short channels. In this region transistor characteristics can be approximated by:

$$I_D = I_{ON} \exp\left(\frac{V_{GS} - V_{th} + \eta V_T}{\eta V_T}\right) \quad (3)$$

where for the measured device with  $L=2\mu\text{m}$   $\eta=1.43$  and  $I_{ON}=13\mu\text{A}$ . This part of the characteristics is used for constructing the multiplier-divide circuit.

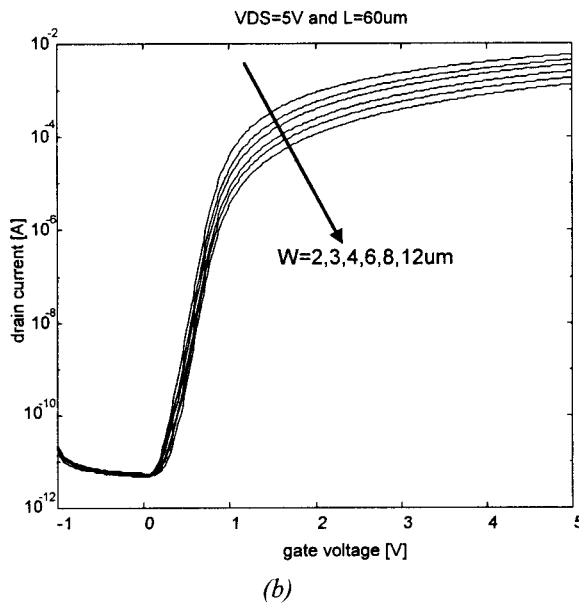
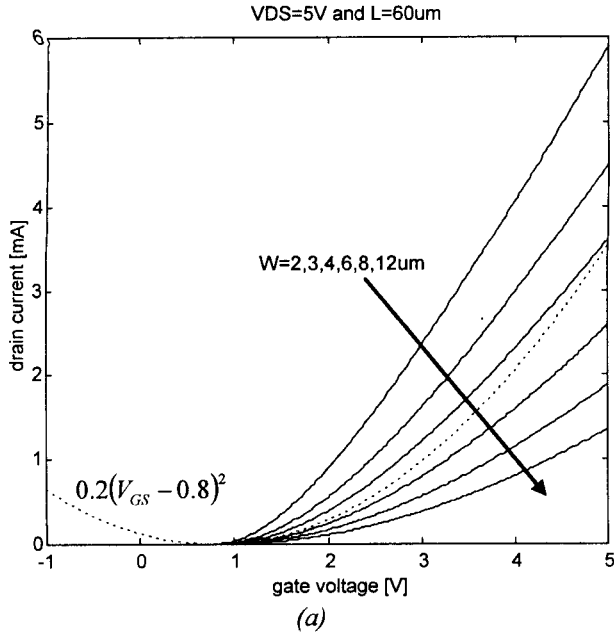


Fig. 1. NMOS transistor characteristics measured with HP4155 Semiconductor Parameter Analyzer for N-well  $2\mu\text{m}$  CMOS technology for transistor with channel width of  $W=60\text{mm}$  and channel length  $L=2,3,4,6,8,12\text{mm}$ . (a) linear scale and (b) logarithmic scale.

The concept of the multiplier-divider circuit is shown in Fig. 2(a). The actual implementation is shown in Fig. 2(b). The main difference between these circuits is that in the circuit on Fig. 2(b) gate-drain voltages for all transistors (with exception of M6, M7, and M9) are close to zero. This

way, the effect of channel length modulation is practically eliminated. Assuming the same sizes for transistors M1 to M4, ignoring the body effect, and using equation (3), one can find that

$$I_{D7} = I_{D4} = \frac{I_1 I_2}{I_3} \quad (4)$$

The threshold change for transistors M1 and M3 due to the body effect should be the similar since the gate-bulk voltages for those transistors are identical. The same is true for transistors M2 and M4.

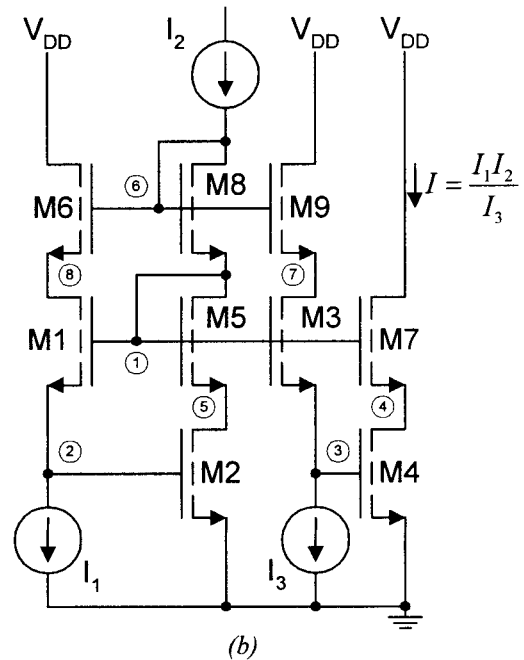
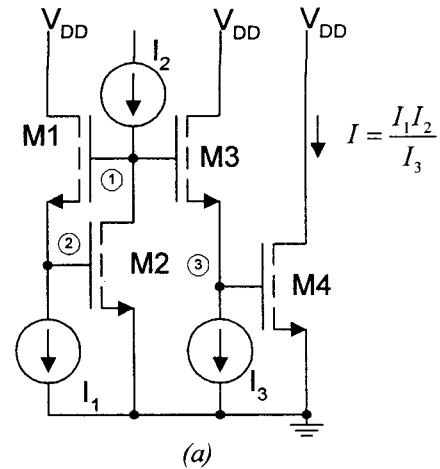


Fig. 2. Circuit diagram of multiplier/divider (a) concept diagram and (b) actual implementation.

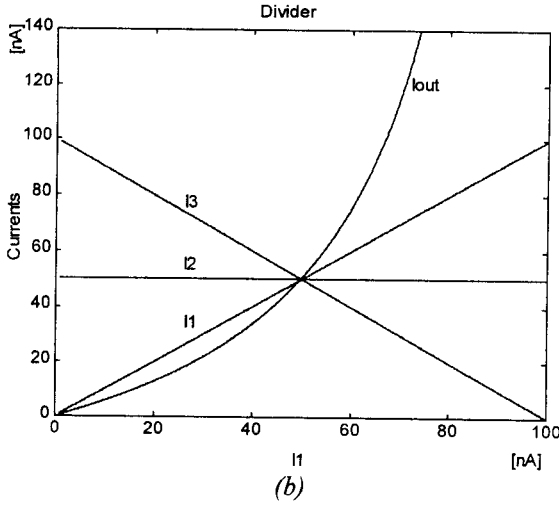
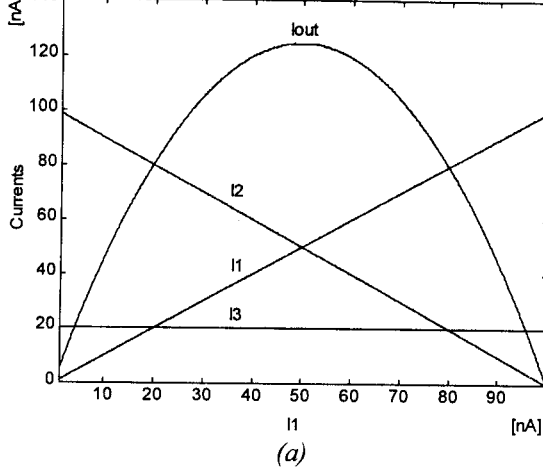


Fig. 3. The circuit of Fig. 2(b) as (a) signal multiplier and (b) signal divider.

Fig. 3 shows SPICE simulation results for the circuit from Fig. 2(b). In the case of Fig. 3(a) the circuit is used as a signal multiplier where current  $I_1$  is multiplied by  $I_2$ , and  $I_3$  is kept constant. In the case of Fig. 3(b) the circuit is used as a divider where  $I_1$  is divided by  $I_3$  and  $I_2$  is kept constant. For example, in Fig. 3(a)  $50\text{nA} \cdot 50\text{nA} / 20\text{nA} = 120\text{nA}$ , while in Fig. 3(b)  $50\text{nA} \cdot 50\text{nA} / 50\text{nA} = 50\text{nA}$ . Note that equation (4) is valid for all points in both graphs of Fig. 3. If  $I_1$  is equal to  $I_2$  the circuit can be used for squaring the signal. Fig. 4 illustrates the squaring action for the triangular waveform, where the output signal is  $\text{INP}^2/\text{BIAS}$ . Results of both Figures 3 and 4 follow equation (4) with high accuracy.

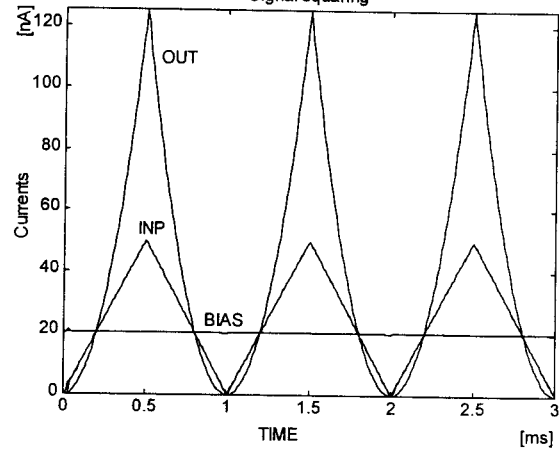


Fig. 4. Transient response of the squaring circuit for triangular waveform.

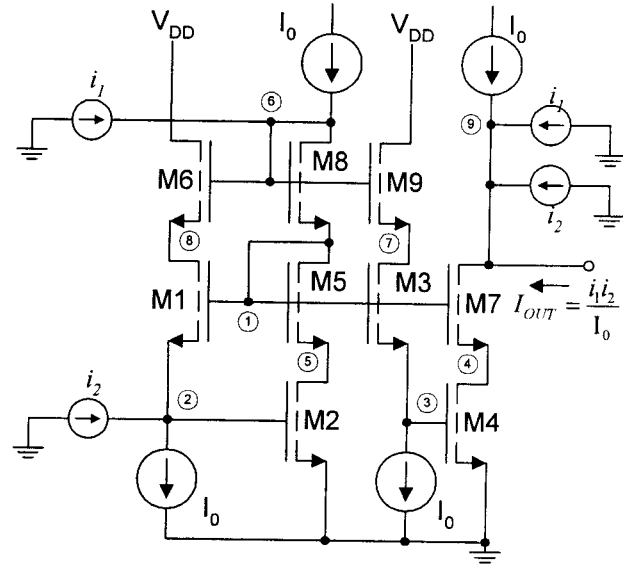


Fig. 5 Circuit diagram of four-quadrant multiplier

### III. FOUR-QUADRANT MULTIPLIER

The circuit presented in the previous section can only operate with a positive input current. However the same circuit, with but some modification, can be used as a four-quadrant multiplier. Let us set the same biasing currents  $I_0$  for all three inputs and let us add/subtract small current components to the first and second input. Equation (4) can be rewritten in the form:

$$I_{D7} = I_{D4} = \frac{(I_0 + i_1)(I_0 + i_2)}{I_0} = \frac{I_0^2 + I_0(i_1 + i_2) + i_1 i_2}{I_0} \quad (5)$$

$$I_{D7} = I_{D4} = I_0 + i_1 + i_2 + \frac{i_1 i_2}{I_0}$$

Note that in order to obtain the product of  $i_1 i_2$  from the drain current of M4, three current components  $I_0$ ,  $i_1$ , and  $i_2$  must be subtracted from the drain current of M7 as it is shown in the circuit diagram on Fig. 5. After the subtraction of  $I_0$ ,  $i_1$ , and  $i_2$ , equation (5) reduces to

$$I_{OUT} = \frac{i_1 i_2}{I_0} \quad (6)$$

and it is valid for both positive and negative values of currents  $i_1$  and  $i_2$ . The four-quadrant multiplier can be also used to multiply voltages. In this case the transconductance amplifier may be used to convert voltages into currents before multiplication.

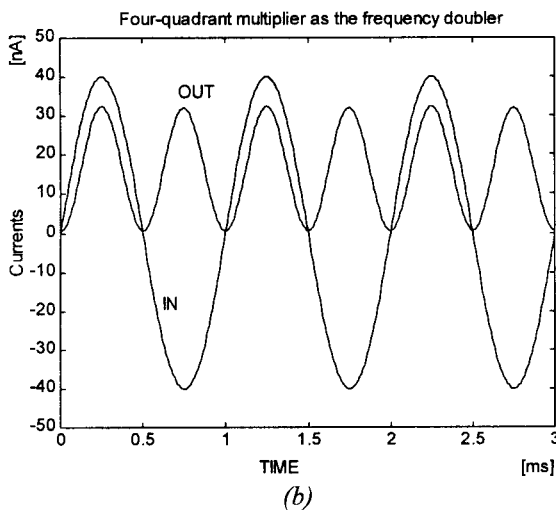
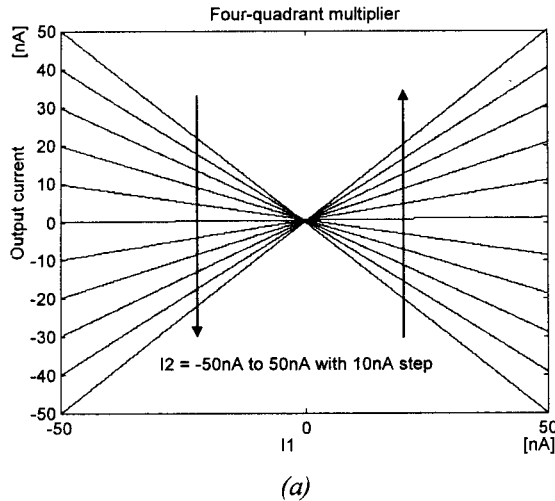


Fig. 6. Simulation results of the four-quadrant multiplier: (a) static characteristics and (b) multiplier used as the frequency doubler.

Fig. 6 shows the results of static and transient analyses of the four-quadrant multiplier circuit. In Fig. 6(a) the current  $i_1$  is swept from -50nA to 50nA, while current  $i_2$  is changed from -50nA to 50nA with a 10nA step. Note the almost ideal characteristics of the multiplier. Fig. 6(b) shows the same circuit used for signal squaring and that the output signal has double frequency of the input signal.

#### IV. CONCLUSION

The multiplier/divider circuit is presented. The circuit is characterized with very high accuracy over a large signal range. The proposed circuit operates in the

current mode, but voltage multiplication is also possible if transconductance amplifiers are used for voltage to current conversions. The original circuit operates correctly only with positive currents, but with minor modifications the circuit may operate also as a four-quadrant amplifier. The circuit itself may have many applications as multiplier, divider, signal squaring, square root calculations, frequency modulation, frequency doubling, etc. Most importantly it can be used for analog signal processing in many intelligent systems of industrial electronics.

#### V. REFERENCES

- [1] Gilbert B. "A New High-Performance Monolithic Multiplier Using Active Feedback," *IEEE Journal of Solid-State Circuits*, Vol. SC-9, pp. 364-373, December 1974.
- [2] Gilbert B. "A Precise Four-Quadrant Multiplier with Subnanosecond Response," *IEEE Journal of Solid-State Circuits*, Vol. SC-3, pp. 365-373, December 1968.
- [3] Liu S.I. and C. C. Chang, "CMOS Analog Divider and Four-Quadrant Multiplier Using Pool Circuits", *IEEE Journal of Solid-State Circuits*, Vol. SC-30, pp. 1025-1029, Sep 1995.
- [4] Liu S.I. and Y. S. Hwang, "CMOS Four-Quadrant Multiplier Using Bias Offset Crosscoupled Pairs" *Electronics Letters*, vol 29, No. 20. Pp 1737-1738, Sep 1993.
- [5] Liu S.I. and Y. S. Hwang, "CMOS Squarer and Four-Quadrant Multiplier" *IEEE Transactions on Circuit and Systems I*, Vol 42, No. 2, pp. 119-122, Feb 1995.
- [6] Song H. J. and C. K. Kim, "An MOS four-quadrant analog multiplier using two-input squaring circuits with source followers," *IEEE Journal of Solid-State Circuits*, Vol. SC-25, pp. 841-847, June 1990.
- [7] Wang Z. "A CMOS Four-Quadrant Analog Multiplier with Single-Ended Voltage Output and Improved Temperature Performance" *IEEE Journal of Solid-State Circuits*, vol. 26, No 9, pp. 1293-1301, Sep 1991.



# Identifying Cutting Sound Characteristics in Machine Tool Industry with a Neural Network

Yasuhiro Ota<sup>1</sup> and Bogdan M. Wilamowski<sup>2</sup>

<sup>1</sup>MAZAK Corporation, Development & Design Division, Aichi 480-01, JAPAN

*y-ota@eng.mazak.co.jp*

<sup>2</sup>University of Wyoming, Electrical Engineering Department, Laramie, WY 82071, USA

*wilam@uwyo.edu*

## Abstract

This paper presents a method for identifying cutting sound characteristics for machine tool industry based on a robust time-variant sound recognition system. The sound signal is compressed using linear prediction analysis method and then recognized by an artificial neural network. The procedure taken here is based on the following: (1) extraction of time-variant spectral features (i.e., raw data of sound), (2) characterization of each sample by observing the autocorrelation coefficients and reflection coefficients of the sampled data, and (3) training of an artificial neural network to identify extracted sound samples. The proposed technique is shown to be very effective, accurate, and powerful in performing sound data identification.

## 1. Introduction

Several approaches using artificial neural networks have been recently proposed for speech recognition technologies [1]-[5], [7], [8]. A lot of research in neural network computing techniques has been devoted towards improvement of classification performance [5]. Some examples include the minimum distance classifier (MDC), radial basis function network (RDFN), dynamic time warping (DTW), and hidden Markov models (HMM). Among them, the RDFN has been criticized for not resembling natural biological neurons. Furthermore, the DTW technique is not satisfied with speech recognition due to the lack of the ability to generalize and extensive computation requirements [3].

A major goal in modern manufacturing systems, particularly in machine tool industry, is to integrate machine tools which have the intelligence to look after themselves and their peripheral devices. In machine tool industry, users need to be always careful selecting optimal cutting environment and conditions to obtain accurate and smooth workpiece, or to meet high standards of product quality. Therefore, it has been troublesome and timely

operation to choose certain cutting conditions. Some of the contributing factors include spindle motor speed, chucking (workpiece holding) pressure, workpiece material, and tool sharpness. The machine operators typically adjust spindle motor speed and other contributing factors by carefully listening to cutting sound. Thus, this work has been a difficult task for novice operators, and even for experienced users, because it is generally hard for the machine operators to distinguish the cutting sound. The idea here is then to apply the technologies found in speech recognition systems to identification of cutting sound in machine tool industry.

In this paper, the following approach is used as an initial experimental work for an intelligent controller by identifying cutting sound data in machine tools: First, a number of sound samples with different cutting conditions are taken. Then, those sound samples are analyzed by observing their autocorrelation coefficients and reflection coefficients. With these data, a neural network is trained using a fast and reliable technique, modified error backpropagation (EBP) algorithm [6], for identification of cutting sound characteristics.

One major concern in sampling cutting sound data is that there exists relatively large background noise since machine tools are normally operated in a factory with some other heavy equipment. Speech recognition systems that perform well under no-noise environment usually experience serious performance degradation when noise is present. Therefore, several research works have been devoted to make a robust speech recognition system which performs well in noisy environments [7][8]. Although their performances were relatively satisfactory, their real-time applications are inferior to neural network approaches which possess parallel processing architecture. A robust speaker identification system based on a modified Kohonen algorithm [3] has several advantages for classification tasks, such as better performance and reduced network training time. However, this approach is only good for a single word, which is not sufficient for

continuous-speech or time-variant cutting sound samples. In order to overcome these problems, a robust time-variant sound identification system is introduced in this paper.

## 2. Analysis of Cutting Sound Samples

The data samples presented to the neural network were made by recording several cutting sound samples in a noisy environment, and a small microphone was mounted at close to a tool head of an NC lathe, *MAZAK- Slant Turn 20*. The data input was sampled at 11,025 Hz with a 16-bit digitizer. In this test, two cutting factors are considered, and these factors are the spindle motor speed and the chuck pressure. All other factors are kept identical for each recording sample. Figs. 1, 2, and 3 illustrate the raw recording data of cutting sound with different spindle motor speed. It has been found from the experiment that the autocorrelation coefficients and reflection coefficients are sensitive to the sound samples with different cutting conditions. Both the autocorrelation coefficients and the reflection coefficients or either of these two can be used for identification of each sound sample. Figs. 4, 5, and 6 show the autocorrelation coefficients of the raw data from Figs. 1, 2, and 3, respectively, while Figs. 7, 8, and 9 show the reflection coefficients of the raw data from Figs. 1, 2, and 3, respectively.

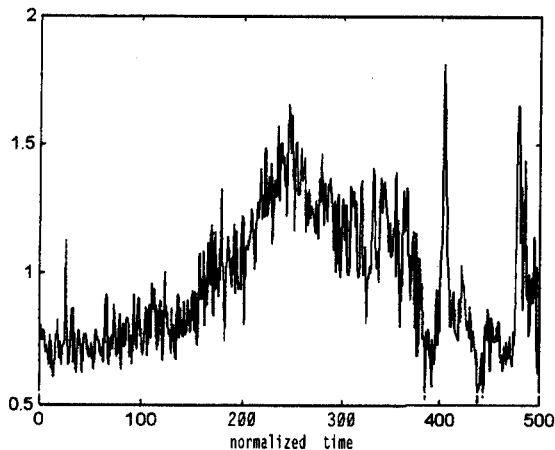


Fig. 1. Raw data of cutting sound with spindle speed of 800 rpm and chuck pressure of 20 kg/cm<sup>2</sup>.

As can be seen from the above figures, there is significant difference in autocorrelation coefficients as the spindle motor speed is varied. Thus, autocorrelation coefficients will be further used to analyze sound data with different chuck pressure. Figs. 10 and 11 show the raw recording data of cutting sound with different chuck pressure, and the corresponding plots of autocorrelation coefficients are

shown in Figs. 12 and 13, respectively.

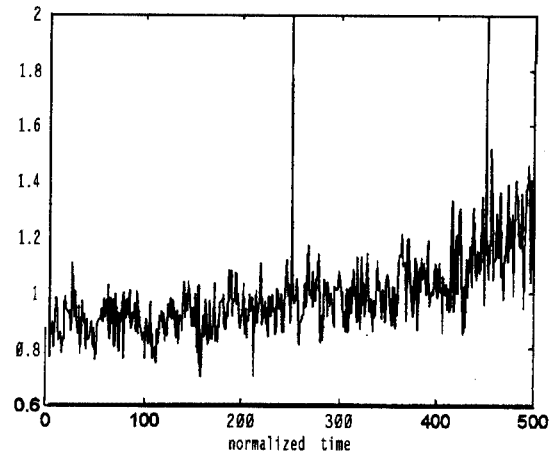


Fig. 2. Raw data of cutting sound with spindle speed of 2200 rpm and chuck pressure of 20 kg/cm<sup>2</sup>.

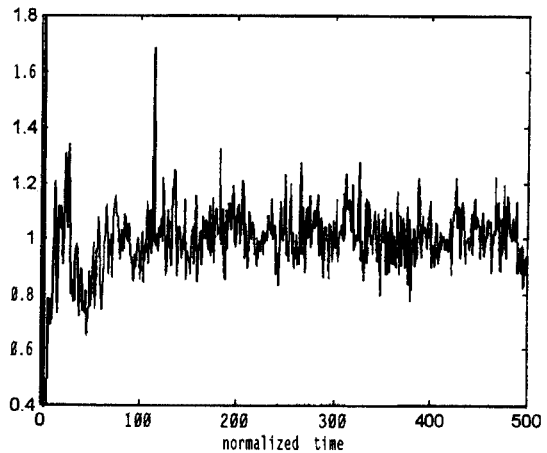


Fig. 3. Raw data of cutting sound with spindle speed of 2700 rpm and chuck pressure of 20 kg/cm<sup>2</sup>.

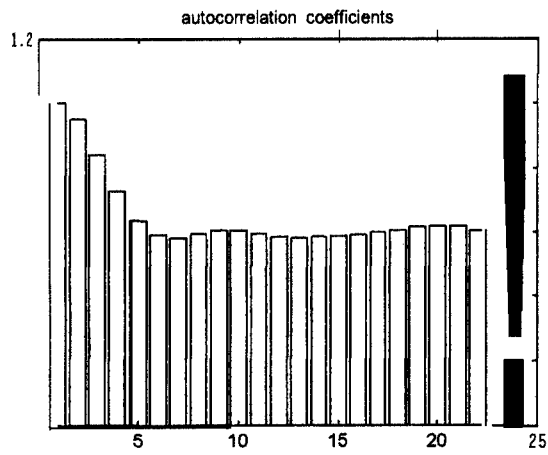


Fig. 4. Autocorrelation coefficients of the cutting sound in Fig. 1.

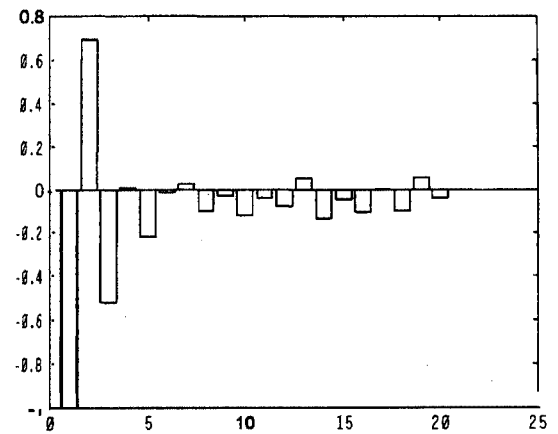


Fig. 7. Reflection coefficients of the cutting sound in Fig. 1.

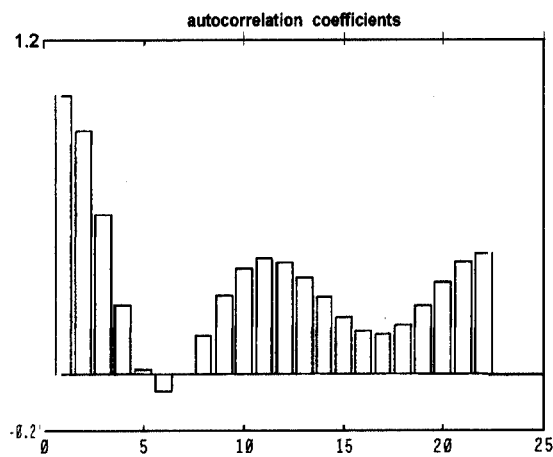


Fig. 5. Autocorrelation coefficients of the cutting sound in Fig. 2.

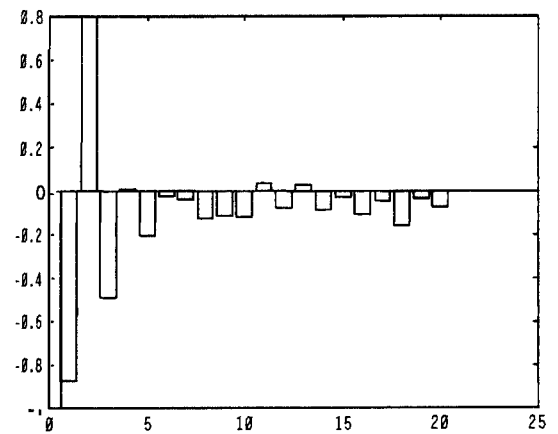


Fig. 8. Reflection coefficients of the cutting sound in Fig. 2.

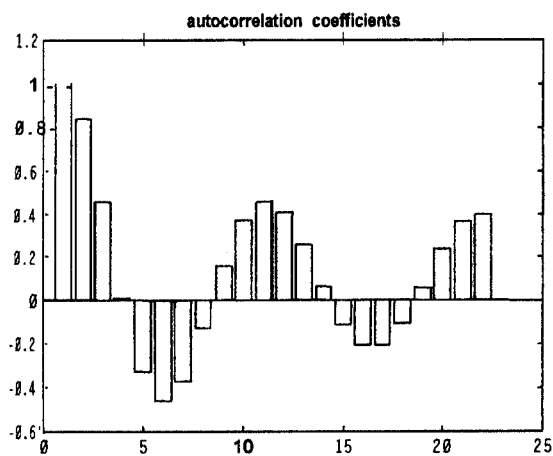


Fig. 6. Autocorrelation coefficients of the cutting sound in Fig. 3.

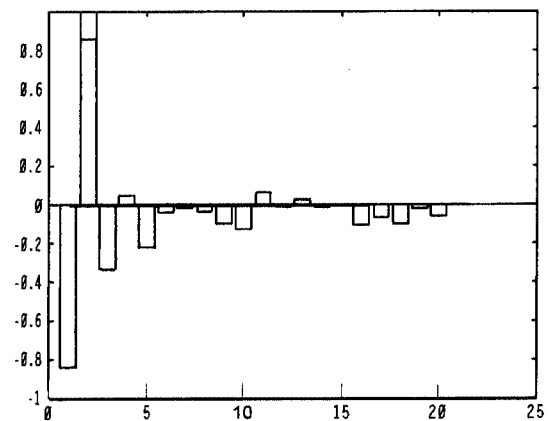


Fig. 9. Reflection coefficients of the cutting sound in Fig. 3.

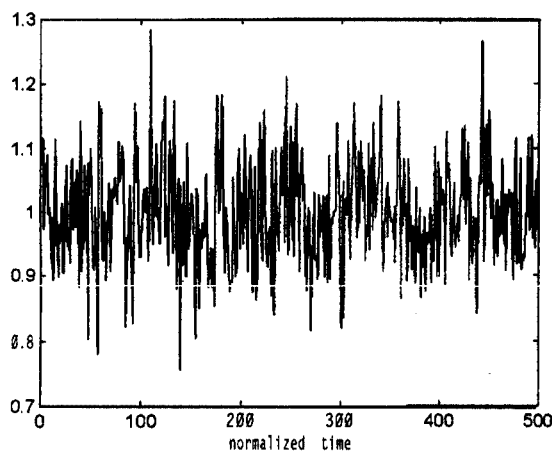


Fig. 10. Raw data of cutting sound with spindle speed of 2000 rpm and chuck pressure of 10 kg/cm<sup>2</sup>.

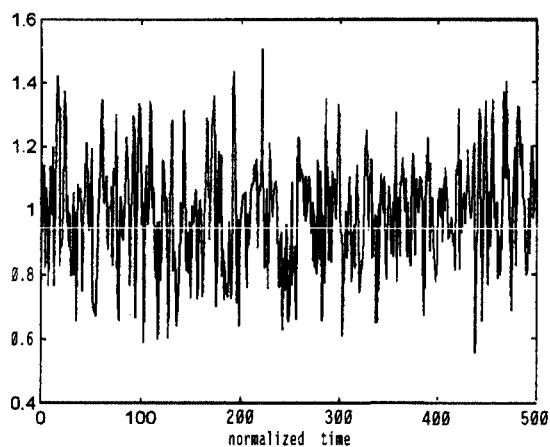


Fig. 11. Raw data of cutting sound with spindle speed of 2000 rpm and chuck pressure of 30 kg/cm<sup>2</sup>.

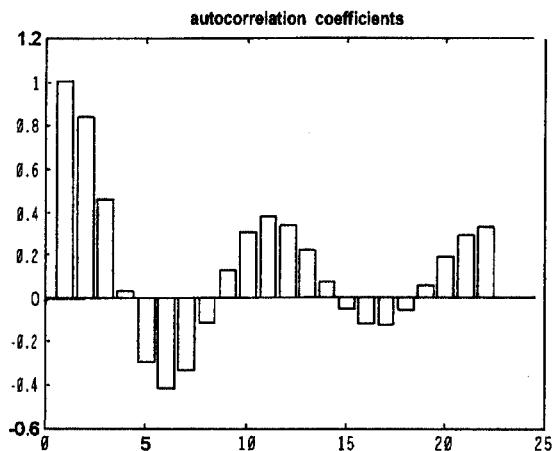


Fig. 12. Autocorrelation coefficients of the cutting sound in Fig. 10.

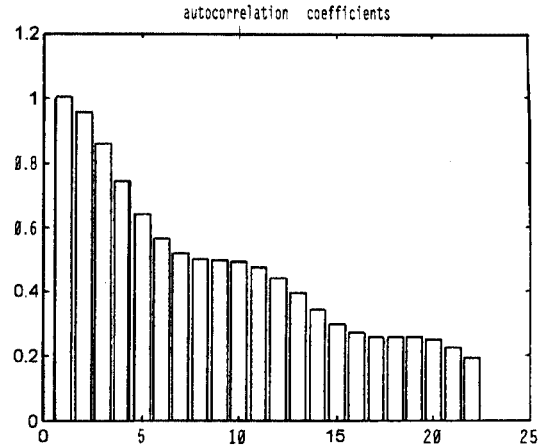


Fig. 13. Autocorrelation coefficients of the cutting sound in Fig. 11.

### 3. Sound Identification with Autocorrelation Method

In the linear prediction (LP) analysis of a sound segment consisting of  $N$  samples, the following  $p$ -th order all-pole transfer function is first assumed.

$$H(z) = \frac{C(z)}{D(z)} = \frac{G}{1 + \sum_{i=1}^p a_p(i)z^{-i}} \quad (1)$$

where the gain  $G$  is normally ignored to allow the parameterization to be independent of the signal intensity. With the filter transfer function described in Eq. (1), its corresponding difference equation to synthesize the sound samples  $c(n)$  is obtained as

$$c(n) = -\sum_{i=1}^p a_i c(n-i) + Gu(n) \quad (2)$$

It can be noted from Eq. (2) that  $c(n)$  is predicted as a linear combination of the previous  $p$  samples. The LP technique gives a procedure for separating the linear-system components (the first term in Eq. (1)) and the excitation-source (the second term) and of the sound production model. However, in order to achieve this, an all-pole model for the linear system is assumed. The all-pole coefficients,  $a_i$ , are calculated from the sound signal on the basis of a least-squares fit between the observed-signal values and the values linearly predicted from the preceding samples.

The autocorrelation method and the covariance method are two standard methods of solving for the predictor coefficients [9] [10]. When the sound signals are applied to this filter as their input, it outputs the LP error signal,  $e(n)$ , whose  $n$ -th order sample is then given by

$$e(n) = c(n) + \sum_{i=1}^p a_p(i)c(n-i) \quad (3)$$

In the LP analysis, the LP coefficients are computed by minimizing the total-squared value of the estimation error, where the summation range depends upon which of the two methods, the autocorrelation method or the covariance method, is used. With the autocorrelation method, the summation ranges from negative infinity to positive infinity, which indicates that the sound signal is available for all time. This method is computationally simpler than the covariance approach. Furthermore, unlike the covariance method, it assures that all the poles of  $H(z)$  lie within the unit circle, guaranteeing the stability of the estimated all-pole filter.

Since the cutting sound data is time-variant, an accurate set of predictor coefficients is adaptively determined over short intervals (10-30 msec) during which time-invariance is assumed. Deeply related to the autocorrelation coefficients,  $a_p(i)$ , are the reflection coefficients,  $r_p(i)$ . Both sets of coefficients can be computed from the sound data using the Levinson-recursion algorithm [11]. The relation between the autocorrelation coefficients and the reflection coefficients is described as

$$r_p(p) = a_p(p) \quad (4)$$

and

$$a_p(i) = a_{p-1}(i) + k_p(i)a_{p-1}(p-i) \quad (5)$$

#### 4. Neural Network Training with a Modified EBP Algorithm

A modified error backpropagation algorithm [6] is used for the network training to identify the sound data. This algorithm is advantageous to other commonly used network training techniques. The traditional error backpropagation (EBP) algorithm has been criticized for its unacceptable slow convergence rate. On the other hand, the modified EBP algorithm sustains a rapid convergence rate also for the cases when the Lavenberg-Marquardt (LM) method [9] fails. In addition, the convergence rate with the modified EBP is nearly independent on the choice of initial synaptic weights. This success is brought from the two major improvements. First, in the output layer a modified gradient is used, instead of an actual gradient, as described in [10][11]. The second improvement is that the synaptic weights in the hidden layer are updated using the modified regression algorithm [12].

The cutting sound samples are first normalized so that the average magnitude becomes zero and that the standard deviation is one. Then, the samples are also normalized with time-scale. For the autocorrelation coefficients and the reflection coefficients, 22 autocorrelation coefficients and 20 reflection coefficients are computed using the Levinson algorithm [13]. For each sound pattern, 10

recordings are made for the network training. Once the network is trained, it is used to identify a cutting condition, in this case, the spindle motor speed and chuck pressure. For the network training, the number of hidden-layer neurons is matched with the number of sound patterns. The output neuron at the output layer performs a simple logical-OR operation, and each layer is augmented with one additional input for biasing. Hence, the network architecture for a two-input pattern, for instance, can be visualized as shown in Fig. 14.

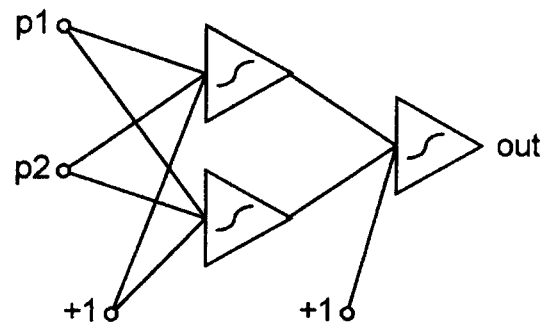


Fig. 14. The neural network architecture for a case with two input patterns.

#### 5. Conclusions

A robust time-variant sound identification system using a new neural network method has been described. In the network training process with the modified EBP algorithm, distant samples are treated as outliers of the signals and rejected so that the radius of the training clusters can be kept smaller. By applying and extending the speaker recognition work [3] which has 98-99.9% accuracy of identification for a single word, sound data that includes time-variant information can be provided effectively.

The tested results indicate that the proposed technique is promising to give an accurate identification of sound samples, even with noisy background, and the proposed algorithm is able to maintain near 100% accuracy. This described work is currently further extended and under the investigation to obtain optimized cutting conditions using neural networks. In addition, it is of interest to investigate sound characteristics including other contributing factors, such as tool sharpness and coolant condition, as the system inputs to achieve better performance. From this initial study to identify the cutting sound, it is expected that the machine operators may be able to monitor tool life by distinguishing the cutting sound using this sound recognition system. Perhaps, the machine users can train the network by supplying a set of sound samples with

different tool sharpness (tool life) so that the system can warn the operators for tool maintenance or tool exchange, once the tool becomes dull and need a new one for creating smooth and accurate cutting.

*Networks (WCNN'95)*, Washington DC, USA, vol. 1, pp. 687-690, July 17-21, 1995.

## References

- [1] T. Matsui and S. Furui, "Speaker Recognition Technology," *NTT Review*, vol. 7, no. 2, pp. 40-48, 1995.
- [2] A. Wold and J. Wroldsen, "Person Identification by Neural Networks and Speech Processing," *Proceedings: World Congress on Neural Networks (WCNN'94)*, vol. 4, pp. 562-567, June 1994.
- [3] K. Vieira, B. M. Wilamowski, R. Kubichek, "Speaker Identification Based on a Modified Kohonen Network," *Proceedings: International Conference on Neural Networks (ICNN'97)*, vol. 4, pp. 2103-2106, June, 1997.
- [4] B. J. Lee, "Parallel Neural Networks for Speech Recognition," *Proceedings: International Conference on Neural Networks (ICNN'97)*, vol. 4, pp. 2093-2097, June, 1997.
- [5] Y. Bennani and P. Galliwari, "Connection Approaches for Automatic Speaker Recognition," *Proceedings: ESCA Workshop on Automatic Speaker Recognition*, pp. 95-102, 1994.
- [6] B. M. Wilamowski, "Modified EBP Algorithm with Instant Training of the Hidden Layer," accepted for the *Industrial Electronic Conference (IECON'97)*, New Orleans, November 9-14, 1997.
- [7] D. Mansour and B. Juang, "The Short-Time Modified Coherence Representation in Noisy Speech Recognition," *Proceedings: IEEE-ASSP*, December, 1989.
- [8] M. Wang and S. Young, "Speech Recognition Using Hidden Markov Model Decomposition and a General Background Speech Model," *Proceedings: IEEE-ASSP*, 1992.
- [9] J. D. Markel and A. H. Gray Jr., Linear Prediction of Speech, Springer-Verlag, Berlin Heidelberg, New York, 1976.
- [10] K. T. Assaleh, R. J. Mammone, M. G. Rahim, and J. L. Flanagan, "Speech Recognition Using the Modulation Model," *Proceedings: IEEE Int'l Conf. Acoustic, Speech, and Signal Processing*, vol. 2, pp. 664-667, April, 1993.
- [11] T. Parsons, Voice and Speech Processing, McGraw-Hill, 1986.
- [12] M. T. Hagen and M. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Trans. on Neural Networks*, vol. 5, no. 6, pp. 989-993, 1994.
- [13] L. Torvik and B. M. Wilamowski, "Modification of the Backpropagation Algorithm for Faster Convergence," *Proceedings: Workshop on Neural Networks (WNN'93)*, San Francisco, USA, pp. 191-194, November 7-10, 1993.
- [14] L. Torvik and B. M. Wilamowski, "Modification of Gradient Computation in the Back-Propagation Algorithm," *Proceedings: Intelligent Engineering Systems Through Artificial Neural Networks, ANNIE 93 Conf.*, St. Louis, USA, November 14-17, 1993, vol. 3, pp. 175-180.
- [15] T. J. Andersen and B. M. Wilamowski, "A Modified Regression Algorithm for Fast One Layer Neural Network Training," *Proceedings: World Congress on Neural*