Int. J. Engng Ed. Vol. 10, No. 4, pp. 337-354, 1994
Printed in Great Britain.

0742-0269/91 $3.00+0.00

An Interactive Graphic Environment for Numerical Control Lathe Training*

DAOH-TIARN ZHANG
Institute of Business Administration, National Chung Cheng University, Taiwan, ROC
JACOB JEN-GWO CHEN
Department of Industrial Engineering, University of Houston, Houston, TX 77204, USA

An interactive, intelligent simulation system, named NC-LATHE, is developed for NC lathe training. NC-LATHE allows the user to develop, edit and execute an NC lathe program interactively. Three embedded rule-based systems assist the user in selecting workholding devices and machining operations. An operations troubleshooter is used to diagnose several most frequently encountered operation problems during the execution phase with probable causes and recommendations. Ten cases with various machining operations have been used to illustrate the system in practical circumstances. The results indicate that NC-LATHE provides accurate and effective numerical control of most lathe operations. Twenty students were involved in evaluating the effectiveness of NC-LATHE as a training tool and programming aid. The results show that NC-LATHE can be used as a tool to assist the instructor and the student in teaching and learning NC programming and lathe operation. The result of user interface evaluation indicates that NC-LATHE provides a friendly and efficient study environment.

SUMMARY OF EDUCATIONAL ASPECTS OF THE PAPER

The paper describes software applications useful in the following engineering disciplines:

Industrial engineering, manufacturing engineering, mechanical engineering.

The paper is suitable for teaching/classwork/self-study for engineering students at the following levels:

Junior students and graduate students.

What aspects of your contribution are new?

To integrate the simulation, expert systems with NC part programming for NC lathe training

How is the material as presented to be incorporated in engineering teaching?

Students may get access to this system to learn the NC part programming, NC lathe operations

Have the concepts presented been tested in the classroom or in project work?

The system has been tested in the classroom and the result shows that this system is an effective training tool.

What conclusions have been drawn from the experience?

Students like to use computer simulation before operating the real NC lathe.

* Paper accepted 15 January 1994.

INTRODUCTION

INCREASING competition from foreign companies has forced American manufacturers to adopt NC machine tools and other NC equipment that enable them to control costs and improve quality. Statistics in the 1991 US Industrial Outlook show that the rate at which NC machine tools are adopted will continue to increase rapidly, both in machine tools numbers and as a share of total machine tool utilization in the future [1]. The rapid growth of NC machines is expected to increase the employment of numerical tool programmers and NC machine tool operators faster than any other occupations for the next decade [2, 3]. These statistics highlight the expected shortage of workers with NC skills and the importance of NC education and training.

Unfortunately, the NC equipment, material, operation and maintenance are expensive. A significant amount of space is often required to install the NC machines. Though NC machine tools are available, use of the NC machines might be restricted because of excessive costs and problems that could result from broken tools or damaged machine components. As a result, it becomes difficult to allow students to have hands-on laboratory experience in this field. When such a laboratory is not available, simulation is perhaps an effective alternative [4].

The primary objective of this research is to fulfil the above mentioned challenge by developing an interactive, intelligent graphic system named NC-LATHE to teach the basic principles of NC, NC programming and NC lathe operations to engineering and technology students.

AN OVERVIEW OF NC-LATHE STRUCTURE AND OPERATION

The NC-LATHE runs on the PC for numerical control lathe training and consists of the following six systems:

(1) lathe operation advisers (experts in NC lathes),
(2) program development system,
(3) lathe set-up system,
(4) execution system,
(5) manual data input (MDI) system,
(6) reporting system.

The overall structure of NC-LATHE is shown in Fig. 1. The NC-LATHE was developed under two software environments, EXSYS and C. The advisers system and operations troubleshooting adviser during the program execution are handled by the EXSYS environment while the other modules are developed under the C environment. The NC-LATHE is a menu-driven system with a graphical interface. This means that menus or lists of options are displayed that allow the user to use a mouse to select functions in NC-LATHE. Figure 2 shows the main menu with three visible components on the screen: the menu bar at the top, the window area in the middle and the status line at the bottom. The menu bar is the user's primary access to all system functions. Most of what the user sees and does in NC-LATHE happens in windows.

There are eight options on the menu bar: EDIT, ADVISOR, SETUP, DEVELOP, EXECUTE, MDI, HELP and QUIT. A message in the status line prompts the user to move the cursor and press the mouse button in order to select one of the above mentioned options.

Lathe operation advisors

Three lathe operation advisers are integrated with the program development system and the

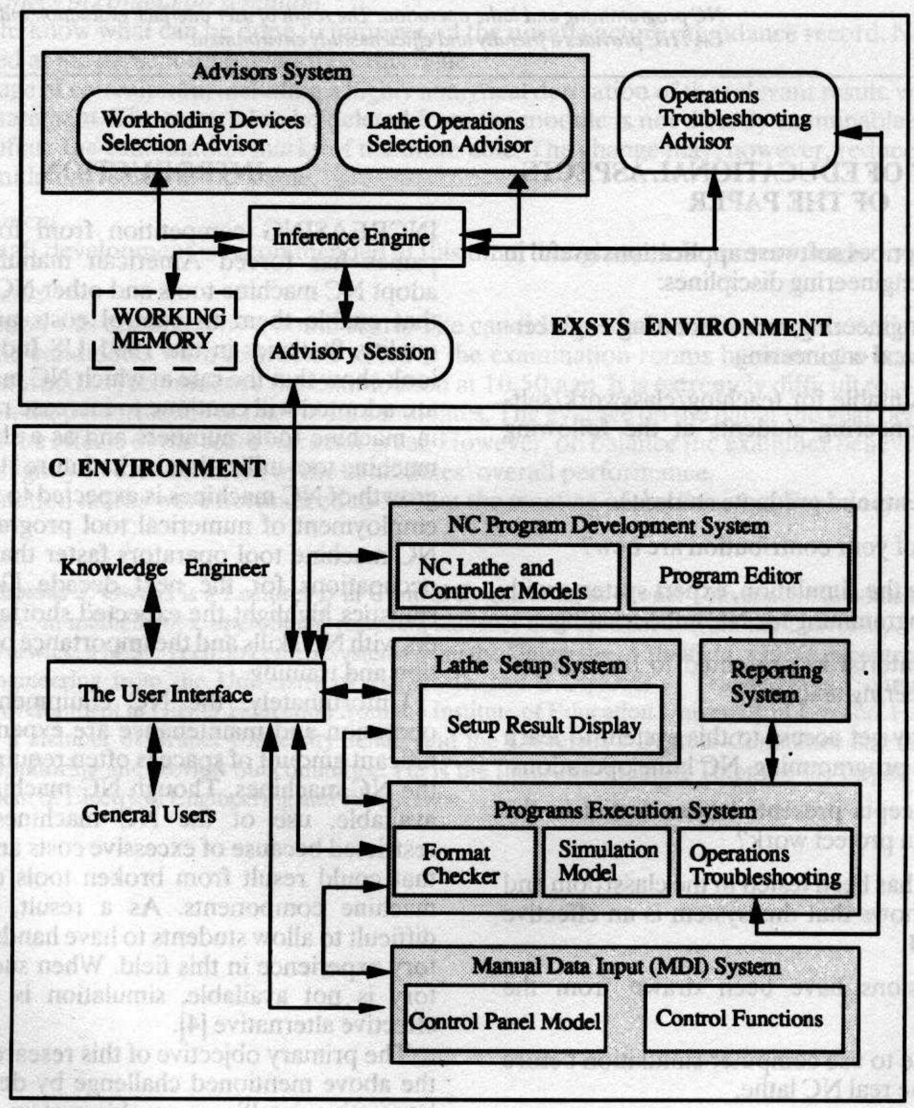

Fig. 1. The structure of NC-LATHE.

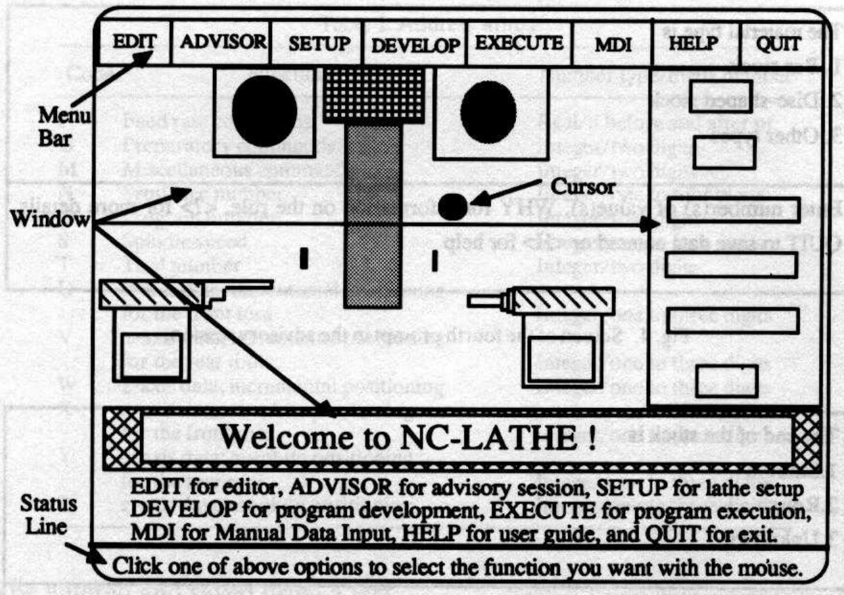

Fig. 2. Main menu.

execution system for programming assistance and operations troubleshooting. The workholding device adviser is designed to assist the user in the selection of workholding devices for lathe operations. An example of the production rule is shown as follows:

Rule 90
IF Workpieces are relatively long with respect to their diameters
AND Workpieces must be machined on both ends
AND Workpieces shape are round cross section
AND Workpieces do not need to be held more accurately than with chucks
AND Workpieces are long slender pieces between centers
THEN Lathe centers and mandrels is recommended

The operation selection adviser helps the user to select lathe operations. For the sake of illustration, let us assume that a stock bar with a diameter of 20 mm and length of 100 mm is held on NC-LATHE and is used to produce a shaft. The user wishes to determine the operation sequence in order to develop an NC program for machining the shaft. Clicking operations advisers on the menu bar with the mouse, Fig. 3 will appear on the screen.

After the user selects the machining operations selection knowledge base, another screen that provides the user with the text for the advisory session presentation appears. The user may press any key to start the advisory session for operations selection. The advisory session begins with a screen. This screen should list the values for the type of raw material (Fig. 4) and the user needs to type in the appropriate response.

Since a bar stock is usually held on the lathe, the user should type ⟨1⟩. The user will now be presented with a screen listing the values of the bar stock end (Fig. 5). Assume the response here is to type ⟨2⟩.

Once this has been done, the advisory session for this issue is finished and the user should be presented with a screen listing the text of the session

To run the lathe operations advisors, You should select one of the following two expert systems:

1. Workholding Devices Selection Knowledge Base.
2. Machining Operations Selection Knowledge Base.

Enter your choice (Default = 1):

Fig. 3. The screen of first prompt in the advisory session.

The material type is
1. Bar stock
2. Disc-shaped stock
3. Other types

Enter number(s) of value(s), WHY for information on the rule, <?> for more details, QUIT to save data entered or <H> for help

Fig. 4. Screen of the fourth prompt in the advisory session.

The end of the stock is
1.Smooth
2.Rough
3.Unknown

Enter number(s) of value(s), WHY for information on the rule, <?> for more details, QUIT to save data entered or <H> for help

Fig. 5. Screen of the fifth prompt in the advisory session.

completion. The results screen for this example is introduced in Fig. 6. As may be noted, this particular user should consider the facing operation in NC programming.

The operation troubleshooter helps the user to diagnose the most frequently encountered lathe operating problems during NC program execution with an explanation of probable causes and recommendations. An example of the production rule in the turning troubleshooting is given as follows:

Rule 42
IF Chatter and vibration are occurring
AND Poor surface finish is occurring
THEN Change the tool geometry [cf = 90%]
AND Change the feed rate [cf = 90%]
AND Change the depth of cut [cf = 90%]
AND Change the cutting speed [cf = 80%]
AND Change the tool angles [cf = 80%]
Reference: *Tool and Manufacturing Engineers Handbook*, 4th edn., pp. 8–76 and 8–77 (1983).

The lathe operating problems (e.g. tool failure, chatter, cutting tool problems and tooling set-ups) occurring in various machining operations (e.g. turning, milling and drilling) on the NC lathe were simulated. The execution phase of a program will be interrupted by showing the warning message on the screen if the user does not set up the lathe model properly. The user then accesses the troubleshooter for operations troubleshooting. After the troubleshooter makes a complete diagnosis of the operation's failure, causes and remedies are provided.

In order to make the lathe operation advisers easy to maintain, these expert systems are built separately. The knowledge in these expert systems is obtained from the public domain such as books and journals [5–10]. Currently, lathe operations advisers have a total of 390 rules. There are 25 rules for workholding devices selection and 65 rules for machining operations selection. The operation troubleshooting has a total of 300 rules (e.g. 105 rules for turning troubleshooting, 110 rules for milling troubleshooting and 75 rules for drilling troubleshooting).

The program development system

In NC-LATHE, an NC program can be prepared either manually or using the program development system. When a program is prepared

Values based on 0/1 system	**VALUE**
1. Operation choice is facing	**0.9**

All choices<A>, only if value>1<G>, Print<P>, Change and return<C>, rule used<line number>, Quit/save<Q>, Help<H>, Done<D>

Fig. 6. Results screen in the advisory session.

Table 1. Address words

Code	Function	Number type/digits number
F	Feed rate commands	Real/3 before and after pt.
G	Preparatory commands	Integer/two digits
M	Miscellaneous commands	Integer/two digits
N	Sequence number	Integer/one to four digits
O	Program number	Integer/four digits
S	Spindle speed	Integer/one to four digits
T	Tool number	Integer/two digits
U	*x*-axis data; incremental positioning for the front tool	Integer/one to three digits
V	*x*-axis data; incremental positioning for the rear tool	Integer/one to three digits
W	*z*-axis data; incremental positioning	Integer/one to three digits
X	*x*-axis data; absolute positioning for the front tool	Integer/one to three digits
Y	*x*-axis data; absolute positioning for the rear tool	Integer/one to three digits
Z	*z*-axis data; absolute positioning	Integer/one to three digits

manually, it can be entered and saved using a text editor or using the built-in NC-LATHE editing system. The program development system connects to the execution system for developing an NC program interactively. The system permits users to test and debug their programs line by line with respect to lathe behavior, part shape, collisions, interference and other information (e.g. spindle speeds, feed rates and coolant conditions). The program development system not only shows users whether the processing command they have typed would be successful but also saves error-free codes and deletes error codes automatically.

There are three basic formats that have been used in NC programming: fixed sequential, tab sequential and word address. Two of the formats, fixed sequential and tab sequential, are obsolete, that is, they are not used on new NC equipments [11]. Therefore, the ANSI/EIA RS-274-D word address format is adopted in NC-LATHE. There are 14 categories of NC code used for programming the target lathe (Table 1). Each category is identified by its address word and has a special meaning to the NC controller model.

The G, M, N, O, S, T, U, V, W, X, Y and Z codes take integers to make up a word (i.e. GO1, TO2). The remaining codes can take integers or decimals. Entering command data (codes) into a block in any order and any case (upper case and lower case) desired is permitted. Codes fall into two groups: modal or non-modal. Commands that stay in effect until changed or canceled are modal commands. One-shot commands are non-modal. Table 2 describes the various G and M codes that are applicable in NC-LATHE to perform a specific mode of operation. The specification of the major components of NC-LATHE are the following.

1. Collet/chuck dimensions: diameter 7 cm.
2. Clamp dimensions: thickness 1.5 cm.
3. Toolpost dimensions: width 2 cm, height 2 cm and length 8 cm.
4. Toolholder dimensions: width 1 cm, height 1 cm, and length 9.6 cm.

Table 2. Preparatory and miscellaneous commands

Code	Function
G00	Rapid travel (modal), sets the NC controller for rapid travel mode on the axis motion
G01	Linear feed rate travel (modal), sets the NC controller for linear motion at the programmed feed rate
G04	Pause or to establish time delay during which there is no machine motion
G98	Sets the NC controller for tool feeding in m.m.p.m.
G99	Sets the NC controller for tool feeding in millimeter per revolution (m.m.p.r.)
M00	Tells the NC controller to interrupt the program
M03	Starts the spindle rotation clockwise
M04	Starts the spindle rotation counterclockwise
M05	Stops the spindle rotation
M08	Turns on the coolant
M09	Shuts off the coolant
M10	Clamps the workpiece automatically
M11	Unclamps the workpiece automatically

5. Workpiece dimensions range: radius 0–3.2 cm and length 0–30 cm.
6. Tool size: tool no. 1 0.3 cm (width), tool no. 2 0.6 cm (width) and 0.2 cm (nose radius) and tool no. 3 0.4 cm (diameter) and 3 cm (length).
7. Tool number: 3 (tool no. 1 is a grooving and parting tool, tool no. 2 is a turning and side tool and tool no. 3 is a milling and drilling tool). The data for tool offsets are as folows: 159 mm on the *x*-axis and 200 mm minus the workpiece length on the *z*-axis for tool no. 1 163 mm on the *x*-axis and 198 mm minus the workpiece length on the *z*-axis (front end) or 168 mm on the *x*-axis and 198 mm minus the workpiece length on the *z*-axis (right side) for tool no. 2 and 158 mm minus the workpiece radius on the *x*-axis and 207 mm minus the workpiece length on the *z*-axis for tool no. 3.

The above mentioned specifications can be changed according to specifications and functions of the real NC lathe.

The NC controller model simulates most of NC control functions and software requirements of an NC controller. On the basis of an existing NC controller, the specifications of the controller model are designed as follows.

1. Controlled axes: two (*x* and *z*).
2. Travel ranges: 27 cm on the *x*-axis and 31 cm minus the workpiece length on the *z*-axis.
3. Linear units accepted: inch (in.) and millimeter (mm).
4. Positioning: both incremental and absolute.
5. Input resolution: 0.03937 in or 1 mm.
6. Output sensitivity: 0.03937 in. or 1 mm.
7. Maximum programmable dimensions: ±12.20 in. or ±310 mm.
8. Rapid travel: 5400 m.m.p.m., without linear interpolation.
9. Feed rate travel: 810 m.m.p.m., with linear interpolation.
10. Feed rate range: 1–6000 m.m.p.m. with an increment of 1 m.m.p.m.
11. Spindle speed is programmable from 1 to 6000 r.p.m.
12. The reference point: tool nos 1 and 2 are X240 to X270, tool no. 3 is Y240 to Y270.
13. Every point (pixel) on the screen is equivalent to 1 mm or 0.03937 in.

In NC-LATHE, the origin is pre-set at the toolpost axis and the tail of the toolpost. The controller model can accept both incremental and absolute positioning commands.

Based on above mentioned format and models, users could develop their NC programs. It causes the user to deal, in a realistic way, with lathe programming. Theoretically speaking, events to be programmed in NC-LATHE are unlimited. Practically, it depends on the available storage space in a computer.

The lathe set-up system

The NC-LATHE is capable of graphically simulating a lathe set-up. Just like operating a lathe, operators have to set up the lathe before starting an NC program. The lathe set-up system allows the user to simulate setting up the tool and workpiece before the lathe model executes its operation.

In the lathe set-up system, the user will be prompted by NC-lathe to provide information for a set-up simulation to be conducted. The prompts are shown below.

1. The initial position of cutting tools on the toolpost (front or rear side).
2. The initial position of clamps on the lathe bed (center or edge).
3. The radius of the workpiece (mm).
4. The length of the workpiece (mm).

The user is first prompted by two questions in order to set up the cutting tool and clamp properly. One question is to ask the user to insert three cutting tools in the front side or rear side of a toolpost. The other question is to ask the user to install the clamp on the center or edge of the lathe bed.

Then the user is instructed to enter the data of the size of the workpiece. In other words, the system allows the user to change the size of workpieces by giving the data of the workpiece's radius and length. The maximum radius is 32 mm. The length of the workpiece cannot be greater than 300 mm.

At the end of the set-up process, the lathe model will be displayed graphically on the screen (Fig. 7). the user can review the clamp positions, tool positions and the size of the workpiece with the screen display. The user may reset the lathe if necessary.

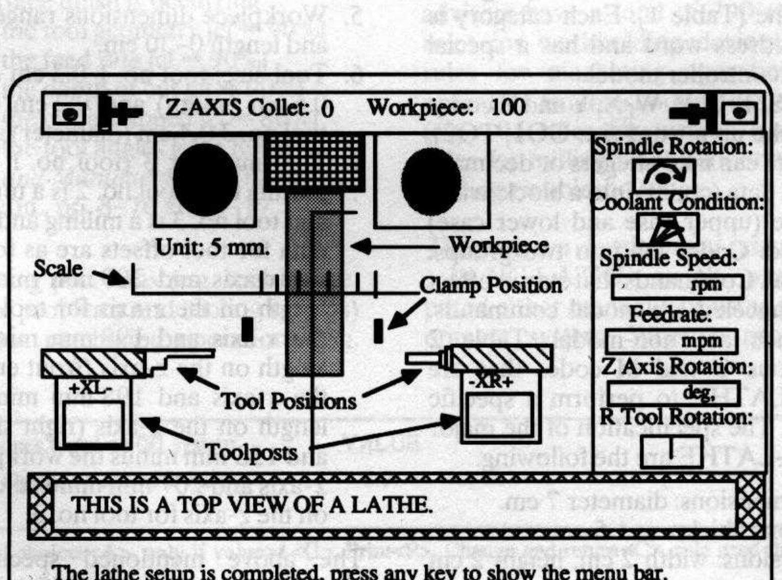

Fig. 7. The result screen of the lathe set-up.

The execution system

The execution system is linked to the program development system, MDI system, reporting system and lathe operation advisers. It requests the user to provide the file name of an existing NC program. After the file name has been entered, the system verifies and executes the program.

All output information is displayed on the screen during the execution phase of an NC program. Users will then be able to study demonstration programs provided by the system or debug their programs directly. The execution phase is designed to dynamically animate various operations of the lathe model and report various messages on the screen corresponding to an existing NC program. This system consists of a program format checker, a simulation model and demonstration programs.

The program format checker can verify the NC program. It reads the NC program and checks it via the embedded NC program format and syntax checking production rules. Then it will inform the reporting system to show the error message and also send a signal to terminate the execution phase for every bad code and operation error.

For example, the following seqauence, no. 10, in an NC program is designed to select tool no. 3 and set tool offset on the z-axis:

N10 TO3 G00 Z350 M10 S1500

The following rules will be activated and the following error messages will show on the screen to indicate that there are two errors in this code:

Rule 14

IF	Tool offset is set
AND	The value of Z-code specified >(310 mm − the workpiece length)
THEN	Show the following error message: Error message—the maximum travel range on z-axis = 310 mm minus the workpiece length. Please change your Z-code.

Rule 19

IF	The spindle speed is setting
AND	M03 or M04 is not specified yet.
THEN	Show the following error message: Error message—you forgot to set the spindle rotation direction! Please specify M03 (rotate clockwise) or M04 (rotate counterclockwise) before defining S-code.

The execution phase is in progress and the successful message is displayed when an NC program is error free. The format checker functions like a bridge that links both the execution and reporting system.

The simulation model accepts an existing NC program and graphically generates lathe operations which correspond to every processing instruction. It dynamically animates lathe operations on the screen in order to simulate the NC program in actual operation.

The simulated NC lathe operations 'duplicate' the behavior of the lathe by graphically displaying interactions among its components. This allows the user to see directly the sequence of events that will take place on an NC lathe. Both the cutting process and operation conditions are demonstrated on the screen. The simulation model possesses the following functions.

1. It lets users see the workpiece, tools and the processs of machining on the screen as it will occur on the NC lathe.
2. All aspects of machine motion are simulated. This includes animation of all moving axes, tool changes and auxiliary units such as clamp, coolant condition and workpiece geometry changes.
3. The animation is detailed enough to show all lathe operations and information such as spindle direction, spindle speed and feed rate.
4. Any collision that would occur between the tool, workholding device and auxiliary units, against any other part of the machine during the operation will be displayed.

This on-screen simulation shows users whether the NC program they have developed would be successful and makes users understand what an NC lathe does and how it performs its tasks during a lathe operation. An example of the screen display of simulation model is shown in Fig. 8.

Fourteen NC programs (Table 3) are included in the package to demonstrate the coordinate system of the lathe model, the offset of three cutting tools, the lathe model capabilities and basic lathe operations. They can be executed by the execution system to help the user understand the lathe model, controller model, program format and simulation model.

Table 3. A list of demonstration programs in NC-LATHE

Number	File name	Purposes of the demonstration program
1	Set-1	Demonstrate tool no. 1 offset
2	Set-2.1	Demonstrate tool no. 2 offset for turning
3	Set-2.2	Demonstrate tool no. 2 offset for facing
4	Set-3.1	Demonstrate tool no. 3 offset for milling
5	Set-3.2	Demonstrate tool no. 3 offset for drilling
6	Parting	Demonstrate cutting off and necking operations
7	Facing	Demonstrate facing operation
8	Plain	Demonstrate plain turning operation
9	Taper	Demonstrate paper turning operation
10	Profile	Demonstrate profile turning operation
11	Groove	Demonstrate grooving operation
12	Drill	Demonstrate drilling and milling operation
13	Sample-1	Demonstrate the cutting process of a shaft
14	Sample-2	Demonstrate the cutting process of a nut

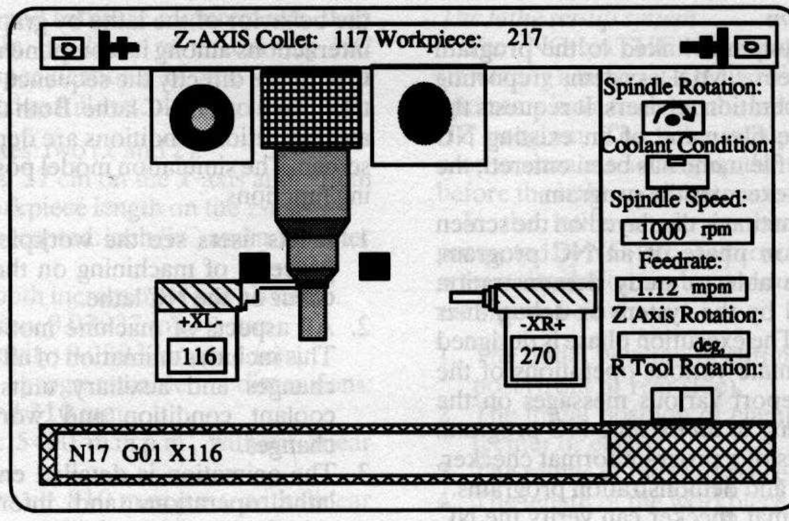

Fig. 8. A screen display of the simultation model for lathe operations.

The MDI system

The MDI system simulates manual data input control, a type of computer-based numerical control, designed for manual programming on the machine site. Users are allowed to operate the system via the control buttons displayed on the screen. The dynamic performance of the part lathe-controller can be simulated when the user manages the button figures. Simulated lathe operations are displayed on the screen when the user presses a button figure with the computer mouse.

The MDI system contains a control panel model with 22 control functions simulated. Table 4 describes the functions and their corresponding miscellaneous words.

In this system, the user enters the NC data into the NC controller model via the above mentioned controls. The NC controller model sends the data to the program execution system for 2-D color graphics views. These views produce the operation animation that corresponds to the NC data. The animation allows the user to see the execution process of an NC data and the outcome that will happen on the lathe model. The tool motion, the part machining process and operation conditions are also displayed on the screen. Consequently, users can make a visual check of the NC data which is provided by them. Such an on-screen simulation not only enables the user to understand the MDI control and its capabilities but also lets the user obtain NC data that may be used in developing NC programs. An example of MDI screen display is shown in Fig. 9.

The reporting system

The reporting system is connected to the NC program execution and verification system for reporting NC data and various messages to the user. The reporting system converts the data that comes from the program execution and verification system to prompt the user for every format, syntax or operation information during the execution of an NC program.

In NC-LATHE, a window is designed to display images and messages related to the lathe model, its status and the execution of NC programs. All typing actions will be listed at the bottom of the window. All lathe status information will be displayed on the right side of the window and all animation will be shown at the middle of the window.

In the reporting system, there is one status line to report general messages, warning messages and error messages (Fig. 10). The general message module offers the information regarding the system's status and the requests of input from the user such as what the systsem is doing or it reminds the user of basic keystrokes applicable at that moment in the active window. An example of the general messages used in NC-LATHE to remind the user is shown:

> Press 'Insert' key, then enter your NC data. Press 'enter' to execute the NC data after you enter one block, maximum 128 characters or numbers in each block, of the NC data.

The warning message alerts users of a condition that may affect their program. In the reporting system, warning messages are also displayed on the status line. An example of the various warning messages used in NC-LATHE to inform the user is shown:

> Your NC data file cannot open!

the error message module helps users to isolate and correct problems that arise in lathe programming. A programming error message refers to a line of bad codes in the user's NC program. An example of the various error messages used in NC-LATHE is shown:

> You must set G98 or G99 before setting F code!

Table 4. Function description for MDI controls

Control	Function	Corresponding miscellaneous words
FEEDRATE INC	An increase of the feed rate by 1 m.m.p.m.	M08
FEEDRATE DEC	A decrease of the feed rate by 1 m.m.p.m.	
COOLANT ON	To sprinkle the coolant on the workpiece	
COOLANT OFF	To stop the sprinkling of the coolant on the workpiece	
CLAMP ON	To support the workpiece	
CLAMP OFF	To release the clamp on the workpiece	
SPINDLE STOP	To stop the spindle rotation	M05
SPINDLE SPEED INC	An increase of the spindle speed by 1 r.p.m.	
SPINDLE SPEED DEC	A decrease in the spindle speed by 1 r.p.m.	
SPINDLE CW	To start the spindle rotation clockwise	M03
SPINDLE CCW	To start the spindle rotation counterclockwise	M04
JOG	An increase or decrease of the movement rate by 1 mm	
TRAVERSE	An increase or decrease of the movement rate by 10 mm	
−X	To move the front tool	(a) U − 1 with 'JOG' pressed. (b) U − 10 with 'TRAVERSE' clicked.
+X	To move the front tool	(a) U + 1 with 'JOG' pressed. (b) U + 10 with 'TRAVERSE' clicked.
−Z	To move the workpiece	(a) W − 1 with 'JOG' pressed. (b) W − 10 with 'TRAVERSE' clicked.
+Z	To move the workpiece	(a) W + 1 with 'JOG' pressed. (b) W + 10 with 'TRAVERSE' clicked.
−Y	To move the rear tool	(a) V − 1 with 'JOG' pressed. (b) V − 10 with 'TRAVERSE' clicked.
+Y	To move the rear tool	(a) V + 1 with 'JOG' pressed. (b) V + 10 with 'TRAVERSE' clicked.
TOOL T1	To select tool no. 1	TO1
TOOL T2	To select tool no. 2	T02
END	To quite MDI system	

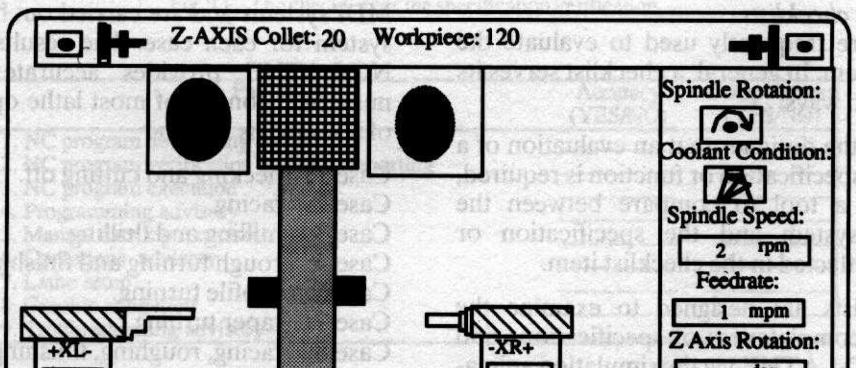

Fig. 9. An example of MDI screen display.

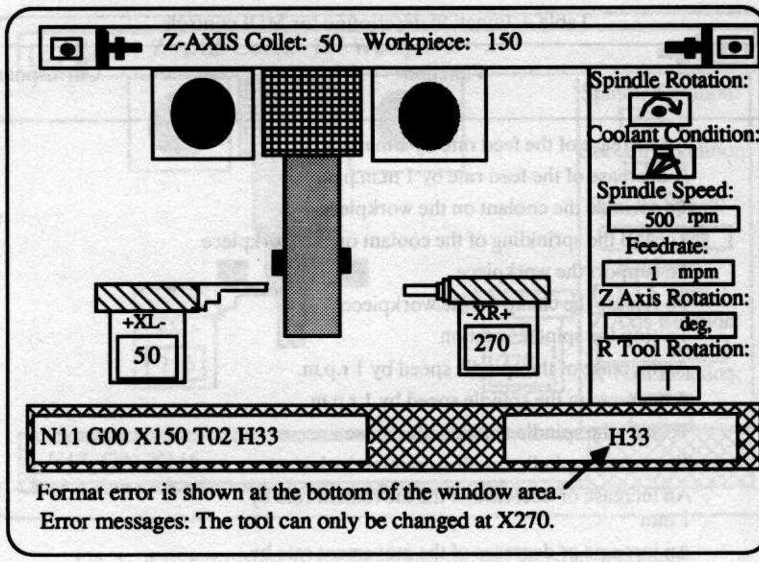

Fig. 10. An example of the reporting system.

VERIFICATION AND VALIDATION

Verification focuses on whether specifications have been implemented correctly. Most simulation languages possess the animation function which provides a media for verifying a simulation model by comparing the graphical animation with the design specifications. The purpose of validation is to test the agreement between the behavior of the model and that of the real system [12]. Since most simulation models contain both simplifications and abstractions of the real world, no simulation model is absolutely correct in the sense of a one-to-one correspondence between itself and the actual system [13].

Verification via checklists

Checklists are frequently used to evaluate the design of a system. In general, a checklist serves its functions in two ways:

(1) to remind the designer that an evaluation of a particular specification or function is required,
(2) to use as a tool to compare between the designed system and the specification or function reflected in the checklist item.

Two checklists are designed to examine the accuracy and completeness of specifications and functions of NC-LATHE via the simulation animation. In the first checklist (Fig. 11), item 1 verifies specifications of the program format, items 2–14 verify specifications of the controller model and items 15–23 verify specifications of the lathe model.

In the second checklist (Fig. 12), items 1–4 examine functions of the program development, items 5–8 examine the lathe operations and item 9 examines the operation troubleshooting. These checklists were used to evaluate NC-LATHE during and after the development of the system to ensure the quality of NC-LATHE. One expert also used these checklists to verify NC-LATHE after NC-LATHE was developed. The results were satisfactory.

Verification via visual examinations

Animation essentially provides a visual equivalent of a trace. With computer animation, NC-LATHE was visually verified whether it could correctly handle normal and special NC lathe operations and programming [14]. Ten cases with various machining operations on the lathe have been used to illustrate the system in practical circumstances. Numerical control programs are developed through program development and MDI systems and are carried out by the execution system for each case. The results indicated that NC-LATHE provides accurate and effective numerical control of most lathe operations. A list of ten cases is shown.

Case 1: necking and cutting off.
Case 2: facing.
Case 3: milling and drilling.
Case 4: rough turning and finishing.
Case 5: profile turning.
Case 7: taper turning.
Case 8: facing, roughing, finishing and cutting off.
Case 9: facing, roughing, finishing, grooving, drilling and cutting off.
Case 10: facing, roughing, finishing, taper turning and cutting off.

Validation via lathe operations

Because the specifications and functions of an available NC lather differed from those of NC-LATHE, the NC programs needed to be modified.

Specifications	Accuracy (YES/NO)	Completeness (YES/NO)
1. Addresses: C, F, G, M, N, O, S, T, U, V, W, X, Y, Z words		
2. Controlled axes: X and Z-Axis		
3. Travel ranges: 27 cm on the X-axis and 31 cm minus the workpiece length on the Z-axis.		
4. Linear units accepted: inch and mm.		
5. Positioning: both incremental and absolute.		
6. Input resolution: 0.03937 inch or 1 mm.		
7. Output sensitivity: 0.03937 inch or 1 mm.		
8. Maximum programmable dimensions: ± 12.2 in. or ± 310 mm.		
9. Rapid travel: 5400 mmpm, without linear interpolation.		
10. Feedrate travel: 810 mmpm, with linear interpolation.		
11. Feedrate range: 1 to 6000 mmpm with an increment of 1 mmpm.		
12. Spindle speed is programmable from 1 to 6000 rpm (revolution per minute).		
13. The reference point: tool No. 1 and 2 are X240 to X270, tool No. 3 is Y240 to Y270.		
14. Every point (pixel) on the screen is equivalent to 1 mm or 0.03937 inch.		
15. Collet / Chuck dimensions: Diameter 7 cm.		
16. Clamp dimensions: Thickness 1.5 cm.		
17. Toolpost dimensions: Width 2 cm, Height 2 cm, Length 8 cm.		
18. Toolholder dimensions: Width 1 cm, Height 1 cm, Length 9.6 cm.		
19. Workpiece dimensions range: Radius 0 - 3.2 cm, Length 0 - 30 cm.		
20. Tool size: Tool No.1 - 0.3 cm (width), Tool No.2 - 0.6 cm (width) and 0.2 cm (nose radius), Tool No.3- 0.4 (diameter) and 3 cm (length).		
21. Tool number: 3 (Tool No. 1 is a grooving and parting tool, tool No. 2 is a turning and side tool, and tool No. 3 i milling and drilling tool.)		
22. Tool changing is programmable: three tools No.1, No.2, and No.3 have T01, T02 and T11 tool change co respectively.		
23. Tool offsets: Tool No. 1 is 159 mm on the X-axis and 200 mm minus the workpiece length on the Z-axis. Tool Nc is 163 mm on the X-axis and 198 mm minus the workpi length on the Z-axis (front end) or 168 mm on the X-a and 198 mm minus the workpiece length on the Z-a (right side). Tool No. 3 is 158 mm minus the workpi radius on the X-axis and 207 mm minus the workpi length on the Z-axis.		

Fig. 11. The checklist for the specification verification.

Functions	Accuracy (YES/NO)	Completeness (YES/NO)
1. NC program developing and editing		
2. NC program verification and error reporting		
3. NC program execution		
4. Programming advisory		
5. Manual data input control		
6. Operations advisory		
7. Lathe setup		
8. Graphic animation		
9. Troubleshooting advisory		

Fig. 12. The checklist for the function verification.

An effective method of validating NC-LATHE was to compare its operations to that of the real NC lathe to ensure the performance of NC-LATHE in program formats, specifications of NC controller and lathe models. Therefore, six NC programs with different machine operations were simulated on NC-LATHE. These six programs were modified and operated according to specifications and functions of the real NC lathe, SpectroLIGHT II Lathe System.

The execution and operation of these NC programs on SpectroLIGHT II were observed and compared with the simulation on NC-LATHE. Parts machined on SpectroLIGHT II were also compared with parts displayed by NC-LATHE in terms of their shape and dimension. The results indicated that the peformance of NC-LATHE was equivalent to that of the real NC lathe. A list of six machining jobs executed both by NC-LATHE and SpectroLIGHT II is as follows and an example of comparison for profile turning is shown in Table 5.

1. Necking and cutting off.
2. Facing.
3. Plain turning.
4. Rough turning and grooving.
5. Profile turning.
6. Paper turning.

Evaluation of NC-LATHE as a training tool and performance aid

One of the major purposes of NC-LATHE is to assist in teaching and programming NC lathe operations. Hence, the most objective method of validating the NC-LATHE is via its applications and comparing the effectiveness of NC lathe training with and without NC-LATHE. An experiment is designed and conducted to evaluate the efficiency of NC-LATHE as a training tool for teaching and learning the NC lathe operations and programming.

As described in Fig. 13, 20 junior and senior students without any NC programming knowledge in the manufacturing processes class at University of Houston received a 4 h lecture in basic NC programming knowledge. All the students had computer and FORTRAN programming experience. The lecture materials included NC machine structure and components, NC procedures and NC words. one NC program is given as an example to the students.

The students were randomly divided into two groups. The experimental group was asked to enhance their NC programming skill by using NC-LATHE for 2 h to learn NC programming from the previously mentioned 14 demonstration programs (e.g. tool offset, plain turning, facing, drilling, etc.). The subjects were then requested to design two programs within 3 h for the following machining tasks:

Task one—facing, roughing, finishing, grooving and cutting off
Task two—facing, roughing, finishing, grooving, milling and cutting off.

The control group was asked to undertake a similar procedure but enhancing their skills by reading the lecture notes and books .

The performance of the two groups are compared in Table 6. The full score is 100 points and a score of 70 points is acceptable. All subjects in the experimental group passed the first problem and failed the second problem. All subjects in the control group failed both problems. Therefore, the percentage acceptance rate for the experimental and control groups was 50 and 0%, respectively.

The one-tailed Student t-test indicates that there is a significant difference $[|T(13.620)| > t_{19,99.5}(2.86)]$ between the two groups. The Student t-test was conducted with 19 degrees of freedom at a 0.005 significance level. This result suggests NC-LATHE is a more useful learning tool than a literature review for NC training. Another Student t-test was conducted to compare the degree of difficulty between problems 1 and 2. The results show that there is a significant difference $[|T(14.97)| > t_{19,99.5}(2.86)]$ between the two problems in terms of difficulty.

The control group was asked to repeat the previous experiment after training with NC-LATHE for 2 h just like the experimental group. Their performance was then graded and compared with their previous performance without using NC-LATHE training.

All students of the control group received acceptable grades in problem 1 but not in problem

Table 5. An example of system comparison between SpectroLIGHT II and NC-LATHE

SpectroLIGHT II	NC-LATHE	Comment
N10 M03	N10 G98 T02 M03 S790	The reference point of SpectroLIGHT II is X13 and Z60
N 11 G00 G90 T2 X10 Z50	N11 G00 X 130 Z80 M10	SpectroLIGHT II cannot accept both incremental and absolute positioning
N12 G01 X9 F5 M10	N12 G01 F5 W5 X108	
N13 G01 Z30	N13 G01 X113 W5	Entering codes into a block in any order and any case desired is permitted in NC-LATHE
N14 G00 X10 Z50	N14 G01 X117 W80	
N15 G01 X8 Z30	N15 G00 U3 W − 10	The tool offset of SpetroLIGHT II is 3 mm on the x-axis and 10 mm on the z-axis
N16 G01 Z31	N16 G01 U − 6 W10 M05	
N17 G00 X10 Z50	N17 G00 X270 M11 Z0 S0	SpectroLIGHT II cannot accept S-code
N18 G01 X7		In SectroLIGHT II, the sequence of words in a block must appear in the following order: N.G,T,X,Z,I,K,L,F,M and lower case is not permitted
N19 G01 Z32		
N20 G00 X10 Z50		
N21 G01 X6		
N22 G00 X10 Z50		
N23 G01 X5		
N24 G01 Z34		
N25 G00 X10 Z50		
N26 G01 X5 Z34		
N27 G01 X10 Z30 M11		
N28 G00 X13 Z60 M02		

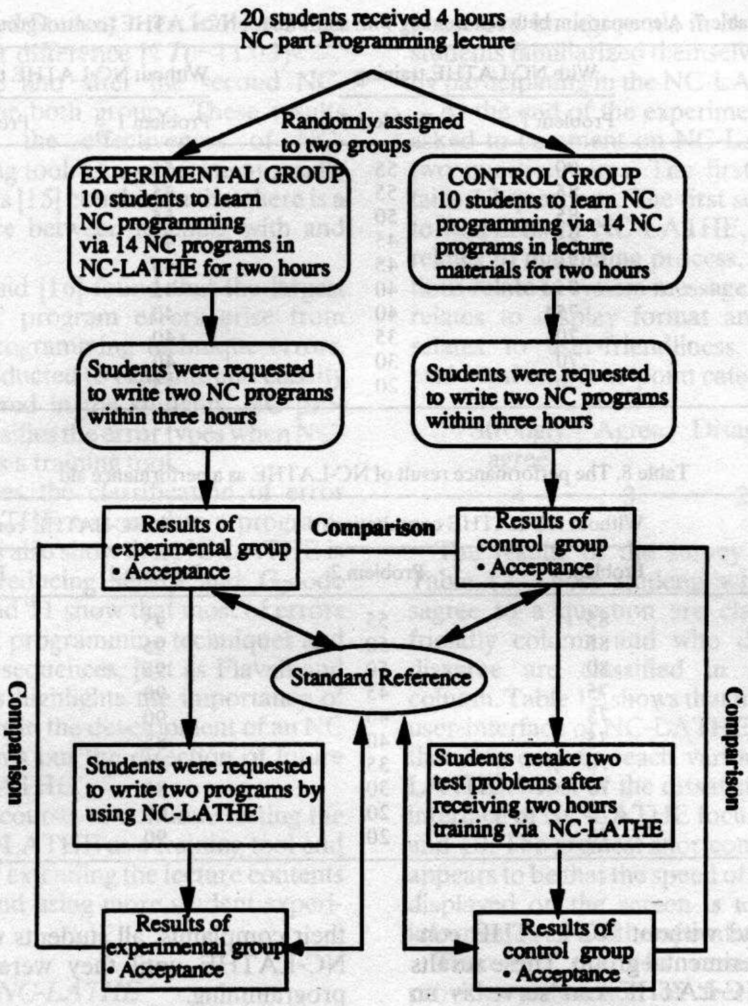

Fig. 13. The experimental procedure.

2 (Table 7). This means that the performance of the control group improved to a 50% acceptance level. The one-tailed Student *t*-test indicates that there is a significant difference $[|T(-15.90)| > t_{19,99.5}(2.86)]$ before and after receiving training via NC-LATHE for the control group. This result substantiates the conclusion obtained from Table 6 that NC-LATHE is an effective tool for NC training.

The experimental group was asked to solve two new problems with consultation to NC-LATHE. In other words, the subjects were allowed to use NC-LATHE as a performance aid to design two new tasks.

Table 8 shows that the percentage acceptance of the experimental group improved from 50 to 100%. The one-tailed Student *t*-test indicates that there is a significant difference $[|T(-7.72)| >$

Table 6. The performance result of learning efficiency

Subject	Experimental group		Control group	
	Problem 1	Problem 2	Problem 1	Problem 2
1	85	55	55	35
2	85	50	55	30
3	80	50	55	30
4	75	45	45	30
5	75	40	45	30
6	75	40	45	10
7	70	35	40	10
8	70	30	40	10
9	70	20	40	10
10	70	20	35	10

Table 7. A comparison between training with and without NC-LATHE (control group)

	With NC-LATHE training		Without NC-LATHE training	
	Problem 1	Problem 2	Problem 1	Problem 2
1	90	55	55	35
2	85	55	55	30
3	85	50	55	30
4	85	45	45	30
5	75	45	45	30
6	75	40	45	10
7	75	40	40	10
8	70	35	40	10
9	70	30	40	10
10	70	20	35	10

Table 8. The performance result of NC-LATHE as a performance aid

Subject	Without NC-LATHE consultation		With NC-LATHE consultation	
	Problem 1	Problem 2	Problem 1	Problem 2
1	85	55	95	90
2	85	50	95	90
3	80	50	95	90
4	75	45	90	90
5	75	40	90	85
6	75	40	85	85
7	70	35	85	85
8	70	30	85	80
9	70	20	80	75
10	70	20	90	70

$t_{19,995}(2.86)$] with and without NC-LATHE consultation for the experimental group. These results demonstrate that NC-LATHE can serve as an effective tool in helping students to develop NC programs.

Most students of both groups complained that the time allotted for learning NC-LATHE and developing NC programs was too short. To answer their complaints, all students were allowed to use NC-LATHE until they were familiar with NC programming.

After the training, two new NC programs were designed by the students. Both groups of students improved their performance to a 90% acceptance level (Table 9). Students 1–10 are in the experimental group and students 11–20 are in the control

Table 9. The performance result of second NC-LATHE learning

Subject	Before second training		After second training	
	Problem 1	Problem 2	Problem 1	Problem 2
1	85	55	95	90
2	85	50	95	90
3	80	50	95	90
4	75	45	90	90
5	75	40	90	85
6	75	40	85	85
7	70	35	85	85
8	70	30	85	80
9	70	20	80	65
10	70	20	90	60
11	90	55	95	90
12	85	55	95	90
13	85	50	95	90
14	80	45	90	85
15	75	45	90	85
16	75	40	90	85
17	75	40	85	80
18	70	35	85	80
19	70	30	80	65
20	70	20	80	65

group. The one-tailed Student *t*-test indicates that there is a significant difference $[\in T(-11.03)\in \frown t_{39,99.5}(2.71)]$ before and after the second NC-LATHE training for both groups. These results again demonstrate the effectiveness of NC-LATHE as a training tool. Our experiment results also support Taylor's [15] conclusion that there is a significant difference between teaching with and without simulation.

Flavell and Kincaid [16] found that the largest percentages of NC program errors arise from dimensional and programming technique errors. An analysis was conducted to examine and classify errors which occurred in the students' NC programs. Table 10 classifies the error types when NC-LATHE was used as a training tool.

Table 11 indicates the classification of error types when NC-LATHE was used as a programming aid. The results also show that NC-LATHE is more efficient in reducing set-up and G-code errors. Tables 10 and 11 show that most of errors were in dimensions, programming techniques and defining machining sequences, just as Flavell and Kincaid found. This highlights the importance of reducing these errors in the development of an NC program and it points out the direction of future extensions of NC-LATHE.

It is desirable, of course, to continue testing the effectiveness of NC-LATHE as a training tool and programming aid by extending the lecture contents and learning time and using more student experiments.

User-friendliness of NC-LATHE

The students involved in the application of NC-LATHE also made an evaluation of the user-friendliness of NC-LATHE. Fourteen of them possessed backgrounds in human factors. All the students familiarized themselves with NC-LATHE by participating in the NC-LATHE experiments.

At the end of the experiment, each student was asked to comment on NC-LATHE by filling out two questionnaires. The first questionnaire contains 13 questions. The first seven questions relate to functions of NC-LATHE, the eighth question relates to machining process, the next three questions relate to system message, the twelfth question relates to display format and the last question relates to user-friendliness. All answers were evaluated on a four-point category-ratio scale:

Strongly agree	Agree	Disagree	Strongly disagree
4	3	2	1

The results of the survey are summarized in Table 12. Those students who strongly agree or sagree to a question are classified in the user-friendly column and who disagree or strongly disagree are classified in the user-unfriendly column. Table 12 shows that most students find the user-interface of NC-LATHE is user-friendly and that it is easy to reach various functions of NC-LATHE. Most of the dissatisfaction with the user interface of NC-LATHE focuses on questions 2, 9 and 10. The greatest shortcoming of NC-LATHE appears to be that the speed of simulated animation displayed on the screen is too fast to catch the information of a lathe operation.

Therefore, the new version of NC-LATHE provides a function of 'freezing' the animation display by pressing the 'pause' key. The speed of animation can be reactivated by entering any key. Users also have the option to try out an NC

Table 10. Program errors comparison for NC-LATHE as a training tool

Causes	Experimental group		Control group	
	(Before training)	(After training)	(Before training)	(After training)
Dimension error	10	9	20	10
Programming technique	9	9	17	8
Function error	7	5	14	8
Tooling error	6	4	11	7
Programming set-up error	4	3	9	5
G-code error	2	2	5	1
Total error number	38	32	76	39

Table 11. Program errors comparison for NC-LATHE as a programming aid

Causes	Without using NC-LATHE	Using NC-LATHE	Error reduction rate (%)
Dimension error	10	3	70
Programming technique	9	4	56
Function error	7	1	86
Tooling error	6	1	83
Programming set-up error	4	0	100
G-code error	2	0	100
Total error number	38	8	79

Table 12. Responses of user interface evaluation

Survey question	User-friendly (strongly agree and agree) (%)	User-unfriendly (strongly disagree and disagree) (%)
1. Does the system provide enough information in preparing to perform a specific mode of operation (e.g. rapid travel, absolute position or incremental position)?	95	5
2. Does the system provide enough information in the tool movement (e.g. tool motion dimension in the X direction)?	60	40
3. Does the system provide enough information in the feed rate and spindle speed?	90	10
4. Does the system provide enough information in identifying the tool number and tool change?	90	10
5. Does the system provide enough information on the miscellaneous or auxiliary machining functions (e.g. spindle on/off, coolant condition)?	90	10
6. Does the system provide enough information in the workpiece movement?	80	20
7. Does the system provide enough information in the machine set-up?	90	10
8. Does the system demonstrate the machining sequence?	75	25
9. Does the system provide enough feedback?	65	35
10. Is the feedback easy to follow?	65	35
11. Does the system provide enough information in checking the part machining?	85	15
12. Do you like the system's display format?	80	20
13. Is the system easy to use?	85	15

program through the program development or MDI systems.

The second questionnaire contains 11 questions which relate to NC programming learning with NC-LATHE. All answers were evaluated on a five-point category-ratio scale:

No problem	Minor	Moderate	Serious	Very serious
1	2	3	4	5

The survey results are summarized in Table 13. It shows that most students do not have serious or very serious problems in NC programming with NC-LATHE except in defining the dimension words, machining sequences and tool settings. This supports the findings in tables 10 nd 11 that most student errors were in dimensions, programming techniques and defining machine sequences.

Flavell and Kincaid [16] found that eliminating programming errors may not produce an error-free program since many errors may occur during machine set-up and operation. Their conclusions highlight the need for a simulator on NC lathe operation which is the major contribution of NC-LATHE.

The outcome of applications show that NC-

Table 13. Responses of NC programming learning evaluation

Survey question	No problem (%)	Minor problem (%)	Moderate problem (%)	Serious problem (%)	Very serious problem (%)
1. Do you have a problem in defining the preparatory functic (G-word)?	40	55	0	5	0
2. Do you have a problem in defining the dimension words (Y, Z)?	5	15	40	35	5
3. Do you have a problem in defining the feed word (F-word)?	55	30	10	5	0
4. Do you have a problem in defining the spindle-speed word (S.word)?	55	35	10	0	0
5. Do you have a problem in defining the miscellaneous function word (M-word)?	20	45	25	10	0
6. Do you hve a problem in defining the miscellaneous function word (M-word)?	20	45	25	10	0
7. Do you have a problem in commenting your NC code?	20	55	25	0	0
8. Do you have a problem in defining the machining sequence (e.g. facing, grooving and cutting)?	0	30	50	15	5
9. Do you have a problem in understanding the machine specifications (e.g. tool offset)?	15	30	25	25	5
10. Do you have a problem in understanding the part drawing?	30	40	30	0	0
11. Do you have a problem in the NC programming technique?	10	45	40	5	0

LATHE not only meets the design requirements but also provides a friendly and efficient study environment. The NC-LATHE is currently an operational program and it promises to be an invaluable tool for students to learn about NC lathes and lathe programming.

CONCLUSIONS

NC-LATHE was developed and validated to assist users in teaching and/or learning NC lathe programming. NC-LATHE integrates three knowledge-based systems with an NC lathe simulation model for instructional purposes. NC-LATHE possesses six different functions: NC program development, operations and programming advisory, NC lathe set-up, NC program execution, verification and error reporting.

Two checklists were designed and implemented to ensure the quality of NC-LATHE. Ten cases with various machining operations on the lathe were simulated in practical circumstances to verify NC-LATHE performance. The results show that NC-LATHE provides accurate and effective numerical control of most lathe operations. The user-friendly interface and animation capabilities enhance the effectiveness of NC-LATHE as a training tool and programming aid. The major advantages of NC-LATHE in teaching and/or learning in NC lathe operations and programming are the following.

1. Allowing users to get the 'real-life' experience away from NC lathes.
2. Removing the fear of working with complex NC lathes.
3. Studying NC machining processes away from the machine tool.
4. Avoiding costly mistakes that usually occur in actual environments.
5. Reducing training costs.
6. Making it easier and cheaper to correct NC programming errors.
7. Saving valuable production time on machine tools.

In the future development of NC-LATHE, several extensions are suggested as follows.

1. To integrate NC-LATHE with operations of other NC machine tools (e.g. milling machine, drilling machine) to simulate an integrated machining center.
2. To extend the depth and width of the existing knowledge base in diagnosing various machine cutting problems.
3. To enhance the advisers system by including a tool adviser to assist the user in tool selection.
4. To incorporate a multimedia system to display various machining operations on the screen.
5. To develop a program which can generate an NC program automatically from a finished drawing.
6. To integrate with a high-level language module such as APT (automatic programming tool) to support both manually and computer-assisted programming.

REFERENCES

1. United States Department of Commerce, *1991 U.S. Industrial Outlook*, Washington, DC (1991), 21–28.
2. United States Department of Labor, *Occupational Outlook Handbook*, Washington, DC (1989), 438–440.
3. United States Department of Labor, *1990 Occupational Projections and Training Data*, Washington, DC (1990).
4. L. J. Adrignola, Simulation for machine tool training, *Autofact '88 Conference Proceedings*, 13.1–13.8 (1988).
5. E. P. DeGamo, J. T. Black and R. A. Kohser, *Materials and Processes in Manufacturing*, Macmillan, New York (1988) 584–655.
6. S. Kalpakjian, *Manufacturing Engineering and Technology*, Addison Wesley, New York (1992).
7. B. W. Niebel, A. B. Draper and R. A. Wysk, *Modern Manufacturing Process Engineering*, McGraw-Hill, New York (1989).
8. J. A. Schey, *Introduction to Manufacturing Processes*, McGraw-Hill (1977), 225–256.
9. Society of Manufacturing Engineers, *Tool and Manufacturing Handbook*, 4th edn, vol 1, *Machining*, Dearborn, Michigan (1983).
10. M. Weck, *Handbook of Machine Tools*, Wiley, New York (1984).
11. G. C. Stanton, *Numerical Control Programming: Manual CNC and APT/Compact 2*, John Wiley, New York (1988).
12. J. S. Carson, Convincing users of model's validity is challenging aspect of modeler's job, *Ind. Engng*, **18(6)**, 74–85 (1986).
13. R. E. Shannon, *Systems Simulation: the Art and Science*, McGraw-Hill, New York (1975), 1–31.
14. R. Biekert, An Experimental Comparison of Two Methods of Teaching Numerical Control Manual Programming Concepts: Visual Media Versus Hands-On Equipment, Doctoral Dissertation, Arizona State University (1971).
15. D. P. Taylor, The Effects of Teaching Numerical Control Concepts Via Simulator Versus Non-Simulator Activities on the Achievements Programming Proficiency and Attitude of High School Students, Doctoral Dissertation, The Ohio State University (1973).
16. N. L. Flavell and E. Kincaid, 'Oops', said the NC programmer, *Am. Mach.*, **127**, 134–136 (1983).

Dr Daoh-Tiarn Zhang is an associate professor in the Institute of Business Administration, National Chung Cheng University in Taiwan. He received his BS degree in Industrial Engineering from the Chung-Cheng Institute of Technology in 1973, his MBA degree from National Cheng-Chi University in 1978 and his PhD in Industrial Engineering from the University of Houston in 1992. He had 15 years working experience in manufacturing systems before he joined the University of Houston graduate program for his doctoral degree. His main research interests are computer-integrated manufacturing and operation management.

Dr Jacob Jen-Gwo Chen is an associate professor and chairman in the Department of Industrial Engineering, the University of Houston, Texas, USA. He received his BS degree in Biology from the National Taiwan Normal University, his MS in Industrial Management from the Central Missouri State University and his PhD in Industrial Engineering from the University of Oklahoma. He is the director of the ergonomics laboratory and in charge of the graduate programs in the Department of Industrial Engineering at the University of Houston. His main research interests lie in the areas of expert systems, automation and ergonomics. During the past few years he has conducted several sucessful seminars in Taiwan, China and the USA covering expert systems, automation, plant layout, total quality management and computerized project management.