Lehigh University
Lehigh Preserve

Theses and Dissertations

1-1-1982

# A feasibility study on using voice recognition as input to a turnkey graphics system for numerical control part programming.

Brian A. Kemmerer

## Recommended Citation

Kemmerer, Brian A., "A feasibility study on using voice recognition as input to a turnkey graphics system for numerical control part programming." (1982). *Theses and Dissertations.* Paper 1975.

# A FEASIBILITY STUDY ON USING VOICE RECOGNITION AS INPUT TO A TURNKEY GRAPHICS SYSTEM FOR NUMERICAL CONTROL PART PROGRAMMING

by

Brian A. Kemmerer

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the degree of
Master of Science
in
Industrial Engineering

Lehigh University
1982

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

5-11-82
(date)

Professor in Charge

Chairman of Department

# ACKNOWLEDGEMENTS

I wish to thank Professor Emory W. Zimmers, Jr., my major thesis advisor, and Professor Louis J. Plebani, my minor advisor for their patience, encouragement and advice.

# TABLE OF CONTENTS

# TABLE OF FIGURES AND TABLES

# A FEASIBILITY STUDY ON USING VOICE RECOGNITION AS INPUT TO A TURNKEY GRAPHICS SYSTEM FOR NUMERICAL CONTROL PART PROGRAMMING

by

Brian A. Kemmerer

## Abstract

The objective of this thesis was to examine the feasibility of applying voice recognition as input to a graphics system. Initially, an analysis of four other input devices was done to familiarize the reader with the benefits and limitations of the traditional input modes. Voice recognition was then evaluated to determine its benefits and limitations as a graphics input device. To facilitate this evaluation a specific graphics package, that of numerical control part programming, was selected as a typical user application which had attributes similar to many graphics software packages.

The general type of commands that are appropriate for voice recognition were discussed. Specific examples were given that are pertinent to the numerical control package. Once command selection was analyzed, the process of word selection was evaluated. Examples of selected words, as they are relevant to the numerical control package, were analyzed.

An experiment to determine the effect of three enviromental factors was done. The three factors tested were: vocabulary size, background noise, and storage time. Using an F-Test at a 95% confidence level, only one of these factors, that of vocabulary size, was found to have a statistically significant effect on the recognition efficiency.

# Chapter 1

# INTRODUCTION

This study was initiated based on the belief that the integration of voice recognition with a turnkey graphics system for the purpose of numerical control part programming may allow productivity improvements to be realized. To understand the implications of this integration, however, it is necessary to have an understanding of the technologies involved. Throughout this study two assumptions have been made. First, it has been assumed that the reader has a basic knowledge of graphics systems. Second, it has been assumed that the reader has a minimal knowledge of voice recognition, and numerical control part programming. Key points of the selected turnkey graphics system, voice recognition units, and portions of numerical control part programming are highlighted in various chapters. This is done to give the reader a better understanding as to the capabilities and limitations of the devices used.

The problems and the benefits associated with the interfacing of two turnkey devices (the voice recognition unit and the interactive graphics system) are addressed in this thesis. Analysis of the various benefits and limitations of using voice recognition as input to interactive graphics was done. Specific vocabularies are listed, but more importantly a methodology for establishing new vocabularies are described and presented. In addition to the process of selecting vocabularies, several environmental factors were tested for their effect on the integration of these two devices. These factors were:

1. Vocabulary size
2. Background noise
3. Storage time

The affect they have on the efficiency of using the voice recognition unit for input is described. It is hoped that the reader will be able to appreciate the significance of these factors. A glossary of terms used is presented in Appendex 1.

# Chapter 2

# PROBLEM DEFINITION

When considering voice recognition for input to the specific turnkey graphics system used in this thesis study there are several problems to be addressed. The first of these problems is integrating the voice unit into the graphics network. This problem can be broken down into a physical linking problem, and a logical linking problem. Physical refers to the actual mechanical connections, and logical refers to the links necessary to have the graphics unit recognize the voice unit as an input device. These problems and their solutions shall be discussed in Chapter 7.

Once the voice recognition unit has been successfully integrated into the graphics system another problem arises. This problem concerns the type of commands that are best suited for input into the graphics system via voice recognition. In Chapter 9 a set of rules to be followed to aid the user in selecting appropriate commands has been prepared. A list of chosen commands (pertaining to the numerical control package used) is presented in Appendix 7.

By the very nature of voice recognition all words (or audio sounds) are not equally suited for input into any system. Because of this there arises a fourth problem when trying to apply the voice to interactive graphics. This problem is that of selecting the words (or audio sound) to associate with the commands that are to be entered using voice recognition. An algorithm for selecting an audio sound for given commands is presented in Chapter 10. A list of selected words is presented in Appendix 8 along with their associated commands.

In summary there are three major obstacles to overcome before voice input to a graphics system become a reality. These obstacles are listed below:

1. Interfacing the hardware
2. Selecting the commands
3. Selecting the words to be associated with the selected commands

All of these problems will be addressed in various chapters of this thesis.

## Chapter 3

## REVIEW OF VOICE RECOGNITION UNITS

In general there are two specific types of voice recognition units that have been developed and tested for application. Both of these types use basically the same scheme for automatic recognition[1] (shown in Figure 3.1). The difference is that the one type of recognition unit is "user general" and the other is "user specific". "User general" refers to the type of voice recognition unit that does not require pre-programming by the specific user that is to use it. This means that once a vocabulary is stored, anybody, irregardless of their own speech patterns, can access the vocabulary. This type of voice recognition analysis is often referred to as "phonetic analysis" because it analyzes the phonetics of speech rather than an audio signal.[2]

"User specific" refers to the types of voice recognition units that must be trained by the specific person that is to access the vocabulary. The user's specific voice pattern is used while establishing the vocabulary set to be accessed later. This is usually done using a spectrum analyzer.

As one might expect the "user specific" recognition units are more compact, less complicated electronically, less expensive, and have a lower percentage of retrieval errors than the "user general" recognition units. The basic reason for this is the vast difference in the voice patterns of different people. With such a great variance in human speech patterns it is very difficult for a machine to compensate and adjust to these differences. This compensating and adjusting is a task which the human ear and brain do very well, but it is also something that has not been duplicated efficiently mechanically or electronically.[3]

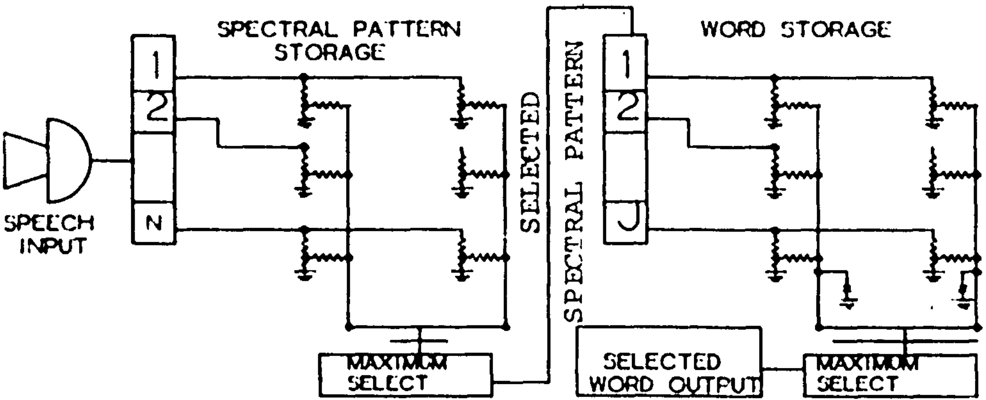


Figure 3.1: Scheme for automatic recognition

The historic development of voice recognition units has been that of a transition from "user general" to "user specific". Bell Laboratories started to develop user general voice recognition units in the early 1960's. They did this in the hope of developing a means of communicating with a computer using voice recognition. After many tests and experiments on both the human voice and the recognition units it would appear they concluded that user general recognition units were impractical because of the high percent of retrieval errors caused by different human voice patterns.[4] Never-the-less the ground work was laid and the basic technological problems were solved in the area of voice recognition. With user general recognition units proving impractical the emphasis in research and development shifted to the area of user specific units.

Today there are many different types of user specific voice recognition units on the market[5]. Although many can trace their heritage back to the basic Bell Laboratory research they differ in respect to fitting algorithms, memory size, usefulness, and of course price. In choosing a specific voice recognition unit one must take into consideration both how and why the recognition unit is to be used, and also the environment in which it is to be used. Another consideration must also be the percent of retrieval errors that can be allowed in a specific application.

Three of the basic distinctions in these user specific voice recognition units are the allocation of memory for the stored vocabulary, where the analysis is done, and the ability to communicate with a host machine. Whether the vocabulary is stored in the recognition unit itself, or in the host computer, has a great impact on both the speed of recognition and the size of the vocabulary. Where the "best of fit" analysis is done (in the host computer or in the recognition unit) also affects the efficiency of the recovery system and the speed of recovery. To be useful most voice recognition units must be able to communicate with other devices, usually computers. How they do this, and the coding patterns used (ASCII, BCD, EBCD)* varies between machines. All these things must be considered while choosing a voice recognition unit for an application.

* See glossary in Appendix 1

# Chapter 4

## TRADITIONAL INPUT DEVICES

One reason graphics systems, especially the chosen graphics system*, lend themselves to voice recognition units is that they are designed in a modular manner. By this it is meant that a graphics system consists of several devices, each having an independent function that fit together to make the system work. The four modula used for input and manipulation of geometric data are listed below:

1. Keyboard
2. Menu
3. Function Keyboard
4. Penstroke recognition

Appendix 2 shows the typical layout of a graphics system.

It is not the intention of this report to be a manual for a turnkey graphics system. The purpose of this chapter, however, is to provide basic information about these four methods of inputting and manipulating data. It is hoped that by knowing about these four traditional methods the reader will be able to relate to how the voice input unit can be used efficiently. A brief summary of each of the four methods, their limitations, and what types of commands each handles best, is described in the following paragraphs.

* The Applicon Graphics system

On the next few pages, and in the rest of this paper the word "command" will be used extensively. In order to clarify what this means with respect to the selected interactive graphics system the following definition is offered. A command is any ASCII string generated by an input device that can be collectively accessed by a single command. In general any command that can be expanded into many commands (or sub-commands) may be referred to as a "Macro." A command or Macro may refer to the process of entering a piece of geometry (such as adding a line), or a control function (such as EDIT-LEVEL-ONE). The only restriction that is made is that a command must be interpretable by the turnkey graphics system.

The keyboard is not only the most traditional way of entering and manipulating information in a computer, but in most graphics systems it is also the most versatile. Any piece of geometry, and any command recognizable to a graphics system can typically be entered using the keyboard alone. Because entering and manipulating geometry usually involves locating and defining coordinate points the keyboard technique is not often used for entering or manipulation of the geometry directly. This is so because locating geometry and defining coordinates with any relative meaning is difficult in many graphics systems using coordinate points as a reference. Entering commands, however, particularly those that do not require selecting geometry is a common use of the keyboard device. In general the keyboard is reserved for either of two types of commands; the first of these is "back-up." "Back-up" in this sense refers to using the keyboard to enter a command after penstroke recognition, menu techniques, or the function keyboard fails. A user soon learns that the keyboard is always available when another device fails entirely or is not performing up to expectation. Examples of these types of commands are listed in Table 4.1.

| Command | Definition |
|---|---|
| GO | Repaint |
| GOSH | Update screen |
| KILL | Ignore last command |
| SE | Select everything |
| DD | Delete selected geometry |
| UU | Unselect |
| UNDR | Undraw selected geometry |
| GRID | Turn grid on or off |

Table 4.1: Examples of High use commands

The second general category of commands that the keyboard is generally used for are commands that are not used very often, or with any regularity. These are the commands that are used every so often, but not often enough to define a penstroke or sacrifice a menu or function keyboard position. For these types of commands even if a penstroke, menu position, or function keyboard position is defined, it is usually faster to type in the command than to remember how to access said command using the other methods. Examples of these types of commands are listed in Table 4.2.

| Command | Definition |
|---|---|
| CDEF PTX | Change definition to Pollytex |
| USE Pl 1 | Use component Pl on level 1 |
| EDIT n | Edit level n |
| GRID n | Change grid size to n |
| SIN | Turn drawing information on or off |
| SH n | Show view n |
| CTRL | Return to control mode |

Table 4.2: Examples of infrequently used commands

Other uses for the keyboard include commands that require a non-standard argument such as "STOR XXX", "ROTA XXX", or commands that are used before the function keyboard menu or pen are enabled. Examples of this type of command are "EDIT", and "LOGI." The keyboard is also used heavily in setting up the function keyboard positions, menu positions, and penstroke recognition. Figure 4.1 shows the area for which keyboard recognition is best suited.

Penstroke recognition is a method of entering and manipulating geometry that is rather unique to the chosen graphics system. What this basically entails is the user training the system to recognize a certain penstroke on the tablet as a command. The pen in the chosen turnkey graphics system has two states which it can be in while it is in use; these states are inking and tracing. Tracing refers to the state where the sensors in the tablet tract the pen while a mechanical switch is closed. The user closes this mechanical switch by pressing down slightly on the pen, he opens the switch by lifting up on the pen. By combining the two states of inking and tracing in various combinations an infinite combination of penstrokes can be obtained. Thus, theoretically any and all commands

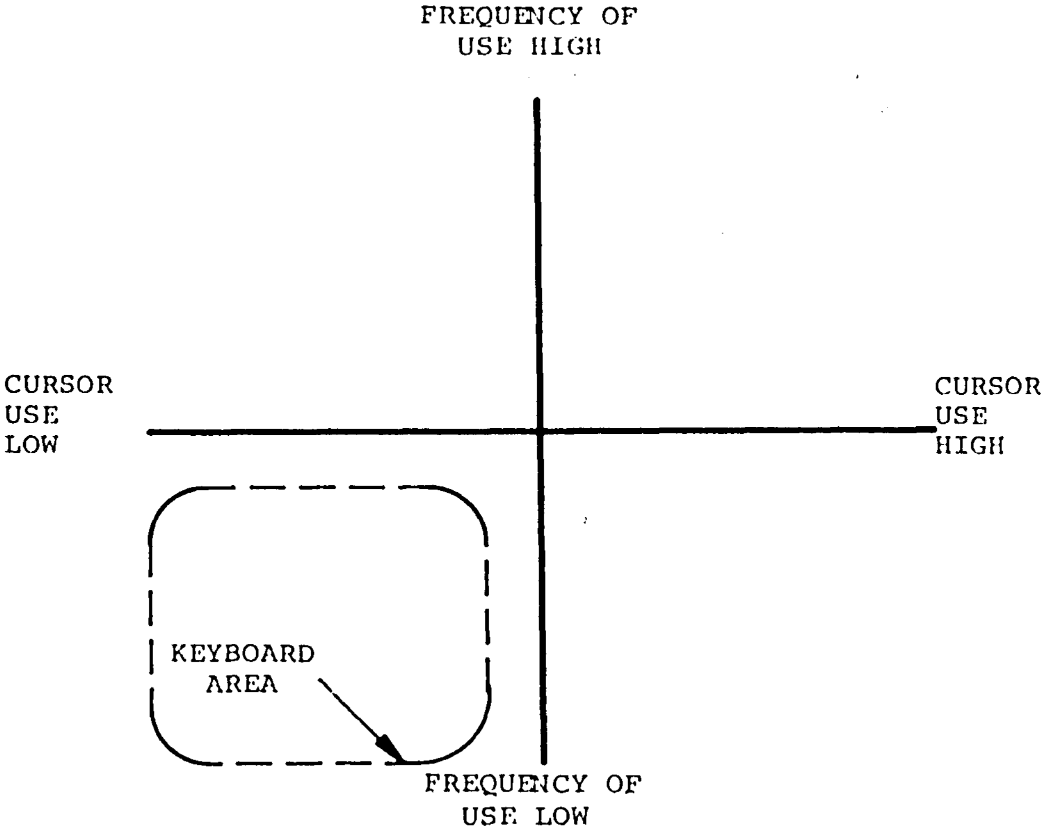


Figure 4.1: Area for which the keyboard is best suited

can be entered using penstroke recognition. Several commands that are usually entered using penstroke recognition are shown in Appendix 3.

There are two very important reasons, however, for limiting the number of commands that are entered by penstroke recognition. The first of these reasons is that although there are an infinite number of possible penstrokes it is not practical to use many that are very complicated. For a penstroke to be useful it must be readily reproducible by the user. For this reason a penstroke that requires a complex combination of inkings and/or tracings is useless. Also a penstroke must be unique to insure the integrity of this input feature. With too many penstrokes defined the user runs a risk of having the graphics processor misinterpret one of the strokes. With many commands defined in penstroke format one also runs the risk of not remembering what command is associated with a particular penstroke. The commands that are commonly entered using penstroke recognition will be remembered, but the commands that are seldom or never used will not be readily associated with their penstrokes. Examples of common penstroke commands can be seen in Table 4.3. This lack of association will render the penstroke inoperable. In short all three criteria for a useful penstroke (reproducibility, uniqueness, and rememberability) cannot be satisfied with an overabundance of recognizable penstrokes.

The second practical and logical reason for limiting the number of commands to be recognized as penstrokes is the time involved in preparing a penstroke for recognition. It makes little sense to establish a penstroke for a command that will only be used on very rare occasions. This is especially the case if the penstroke is likely to be forgotten or because disassociated with the command by the time it is used.

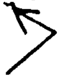
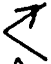
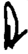
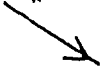
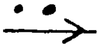
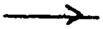
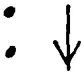
| Stroke | Command | Definition |
|---|---|---|
|  | GO | Repaint screen |
|  | GOSH | Update screen |
|  | DD | Delete selected geometry |
|  | SELECT | Select geometry |
|  | MOVE | Move selected geometry |
|  | ADD | Add a component |
|  | COPY | Copy selected geometry |

Table 4.3: Examples of common penstroke commands

Penstroke recognition when used correctly is perhaps the most powerful input device, and control device in the chosen graphics systems input scheme. Of its many advantages over the other three input devices there are three that stand out as being the most important. The first of these advantages is sheer speed. Once a user becomes familiar with pen use he or she can draw a recognizable stroke very quickly and without much thought. The pen becomes an extension of the users hand and he is able to use it without worrying about positioning on the tablet. In the same vein the graphics processor is just as quick to recognize a penstroke command as a command entered from any of the other input devices. Since a user can almost always make a simple penstroke faster than he can orient himself on the menu, keyboard, or function keyboard the commands entered using penstroke recognition are usually executed faster.

The second advantage of penstroke recognition is related to the users focus of attention. While entering commands via the pen the operator is looking and focusing his attention at the screen where the actual geometry is. The user does not have to look down to orient himself with the keyboard, menu, or function keyboard. All the information concerning inking, tracing, and recognition appears on the screen making the user less distracted.

The one feature of penstroke recognition that makes it very powerful, and in a sense dominant, is that it is very closely related to cursor control. The reason they are so closely related is that the cursor provides the means of telling the graphics system where to insert geometry and text. Many of the penstrokes not only tell the computer to insert a piece of geometry, but they also provide a reference as to where to put it. For example, the penstroke in Figure 4.2 tells the computer to add a line between the end points of the penstrokes.

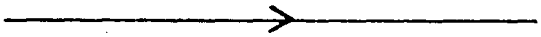


Figure 4.2: Example of an ADD penstroke

Figure 4.3 tells the computer to enter tests starting at the point.

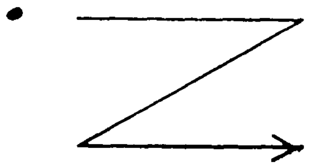


Figure 4.3: Example of an Enter Text penstroke

Not only is cursor control extremely important in entering geometry, but it is also vital in manipulating it. Many commands such as the "MOVE", "COPY", "ROTATE", and "BRIK", use the cursor to provide information as "how far to move", "where to copy to", "what point to rotate about", and "how far to extend the drawing into the third demension." The penstrokes for the "MOVE", "COPY", "ROTATE", and "BRIK", are shown in Figure 4.4.

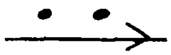
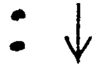
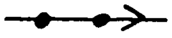
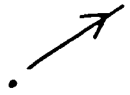
| Stroke | Command | Definition |
|---|---|---|
|  | MOVE | Move selected geometry |
|  | COPY | Copy selected geometry |
|  | BRIK | Extend selected geometry into third dimension |
|  | ADDV | Add vertex (extend a line) |

Table 4.4: Example of penstrokes showing cursor control.

This is not meant to imply that penstroke recognition has a monopoly on the cursor control in most graphics systems. The fact is that any of the other devices, inclusing the keyboard, can be used to control the cursor. In fact for some applications it is more efficient to use one of the other devices (such as using the function keyboard for adding a vertex) than to use penstroke recognition. It is important to point out, however, that by its very nature of inking and tracing penstroke recognition is very closely related to cursor control. Since the cursor always follows the pen, penstroke recognition has a natural advantage over the other three devices for inputting and manipulating geometry where cursor location is relevent. Figure 4.4 shows the type of commands best suited for penstroke recognition.

The use of a function keyboard to enter commands is also very useful in the chosen graphics system. A function keyboard is a system of buttons that the user can define to be interpreted as any command. Typically the function keyboard is a set of from 64 to 128 buttons in a 8 by 8 or larger array. It is important to realize that like penstroke recognition, and the alpha-numeric keyboard, the function keyboard can theoretically be used to execute any command. Like the alpha-numeric keyboard and the penstroke recognition features the function keyboard has its strength and weaknesses. In the opinion of some users one of its dominant weaknesses is its limited space. Un-

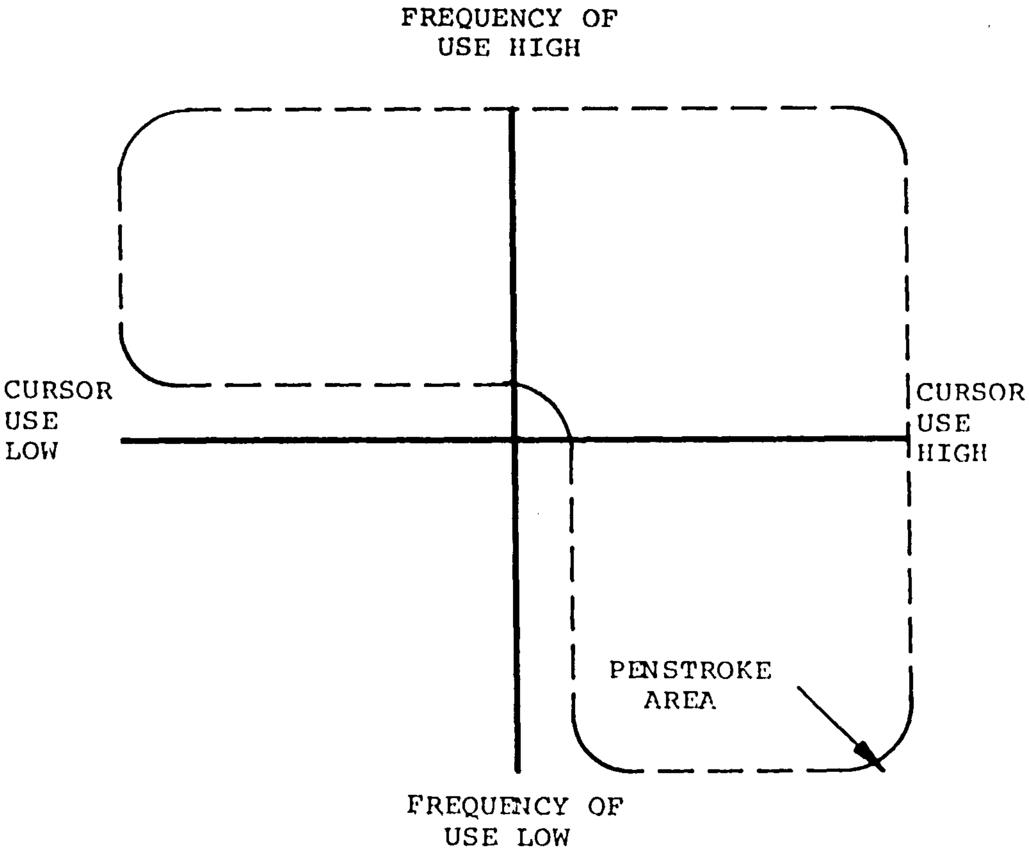


Figure 4.4: Area for which penstroke is best suited

like penstroke recognition, of which there are essentially an unlimited number of unique strokes, and keyboard entry, the function keyboard has a limited capacity of commands directly associated with the number of buttons. For this reason space on the function keyboard may be considered at a premium. Another weakness in using the function keyboard is that of orientation. Unlike the alpha-numeric keyboard which a user develops a feel for orientation or penstroke recognition which does not require the user to look away from the screen; the function keyboard usually requires the user to make eye contact every time it is used. This eye contact is necessary because all of the buttons on the function keyboard feel the same, and except for the corners relative position of the buttons is hard to determine. The need for eye contact both distracts the user and slows him down.

The limited space and the need to orientate on the function keyboard every time it is used have the effect of eliminating two types of commands from practical consideration on the function keyboard. The first of these commands not suited for the function keyboard (dictated by the shortage of space) are commands that are only used once in a long while. With space at such a high premium the user can simply not afford to allocate space on the function keyboard to commands that are used very infrequently. On the other end of the spectrum are commands that are used very frequently. These should not be used on the function keyboard because of the distractions involved in orientating the keyboard. Examples of low use commands are shown in Table 4.2 and examples of very high use commands are shown in Table 4.1.

One very big advantage to the function keyboard is that it can be used in conjunction with the pen. This means that the pen can be used to specify coordinate points by controlling the cursor, while the function keyboard tells

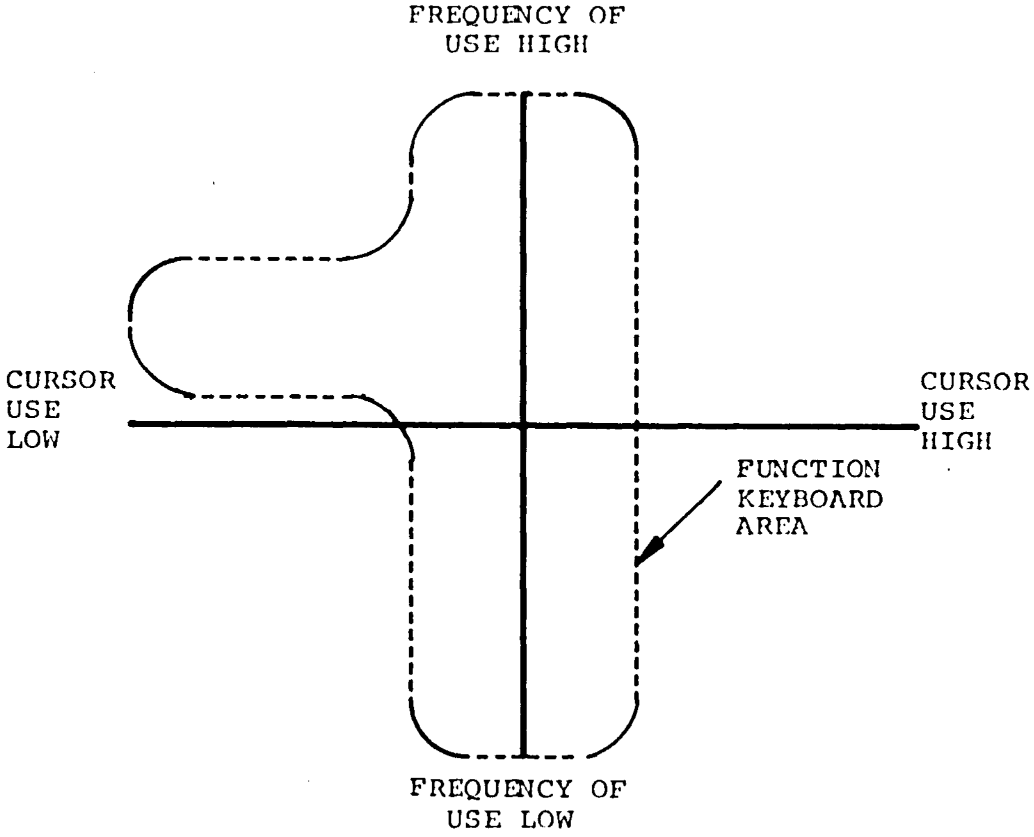


Figure 4.5: Area for which the function keyboard is best suited

the computer what commands to execute. By this method both the position and the command can be entered into the graphics system. A good example of this is that of entering an arc using the pen to point to points on the arc. Simultaneously the user would press the buttons DS (digitize start), to input the start of the arc, DC (digitize corner) to indicate a point on the arc, and DF (digitize finish), to indicate the end of the arc. Figure 4.5 shows how the function keyboard would be used with respect to the commands used and cursor control. Table 4.5 lists several specific commands that fit into this category.

| Command | Definition |
|---|---|
| DS | Digitize start |
| DC | Digitize center |
| DF | Digitize finish |
| GG | Update screen |
| ADDV | Add vertex |

Table 4.5: Commands commonly used on the Function keyboard

The final traditional way of entering and manipulating geometry in a graphics system is by using the menu. The menu is accessed by pressing down on the digitized tablet which is overlayed with a menu, with the pen. On the selected system there can be up to 432 (24 times 18) commands stored on the menu. The basic idea of the menu is very similar to that of the function keyboard, the main difference being that the command is chosen with the pen instead of the user's finger. Since the command is chosen with the pen, the pen cannot be used to control the cursor as easily in this method as it can while using the function keyboard. For this reason menu commands are not usually the type that require high cursor control. Also since there are so many memory locations, commands used very frequently are not usually used on the menu.

Since there are so many menu positions the menu is very useful for commands that are used very seldom. The menu is also used for commands that require a lot of standardized arguments; examples of these are listed in Table 4.6. This combined with the fact that the menu is not the best device for use in commands that require a lot of cursor control, make it best for commands shown in Figure 4.6.

| Command | Definition |
|---|---|
| EDIT 1 | Edit level 1 |
| UEDT 1 | Unedit level 1 |
| TSIZ .5 | Change text size to .5 in. |

Table 4.6: Commonly used menu commands

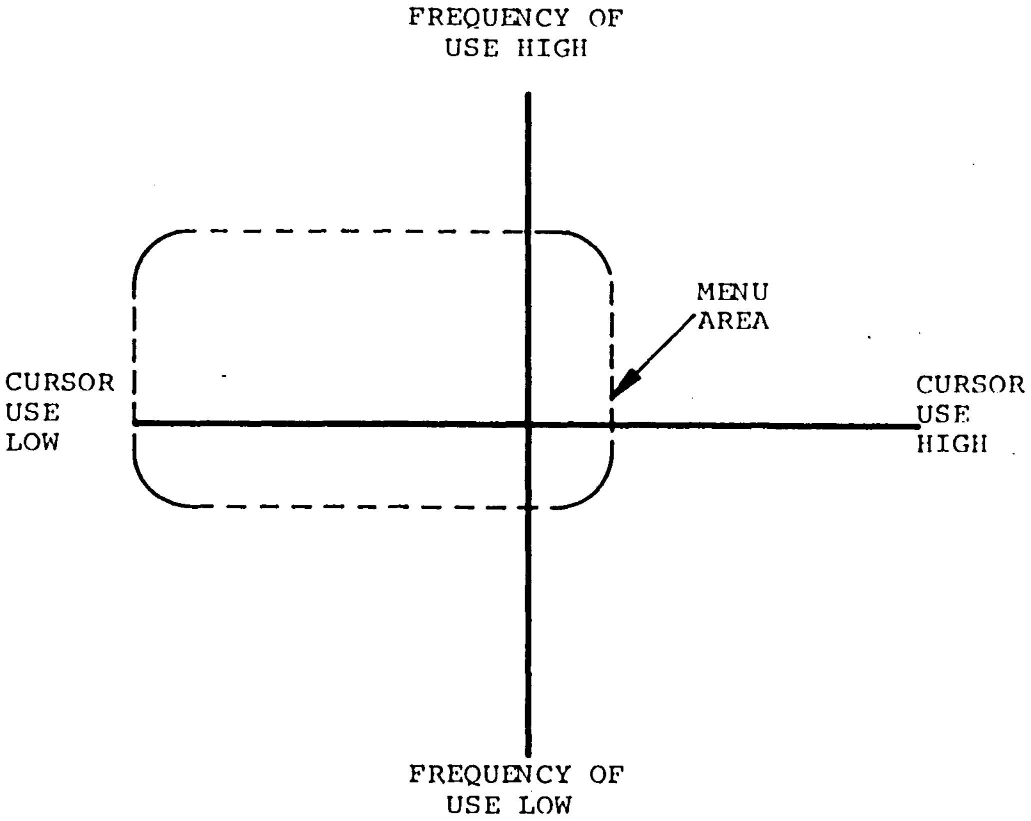


Figure 4.6: Area for which the menu is best suited

Figure 4.7 attempts to clarify that each of the four traditional methods of inputting commands into the selected graphics system have a set of commands that are ideally suited for them. In Chapter 8 there will be a discussion as to which groups of commands are suited best for voice input. It is important to realize at this point, however, that all of the input devices have their place in graphics system. Voice is not intended to replace any of these traditional methods, but merely to compliment them, and make the whole graphics system more efficient.

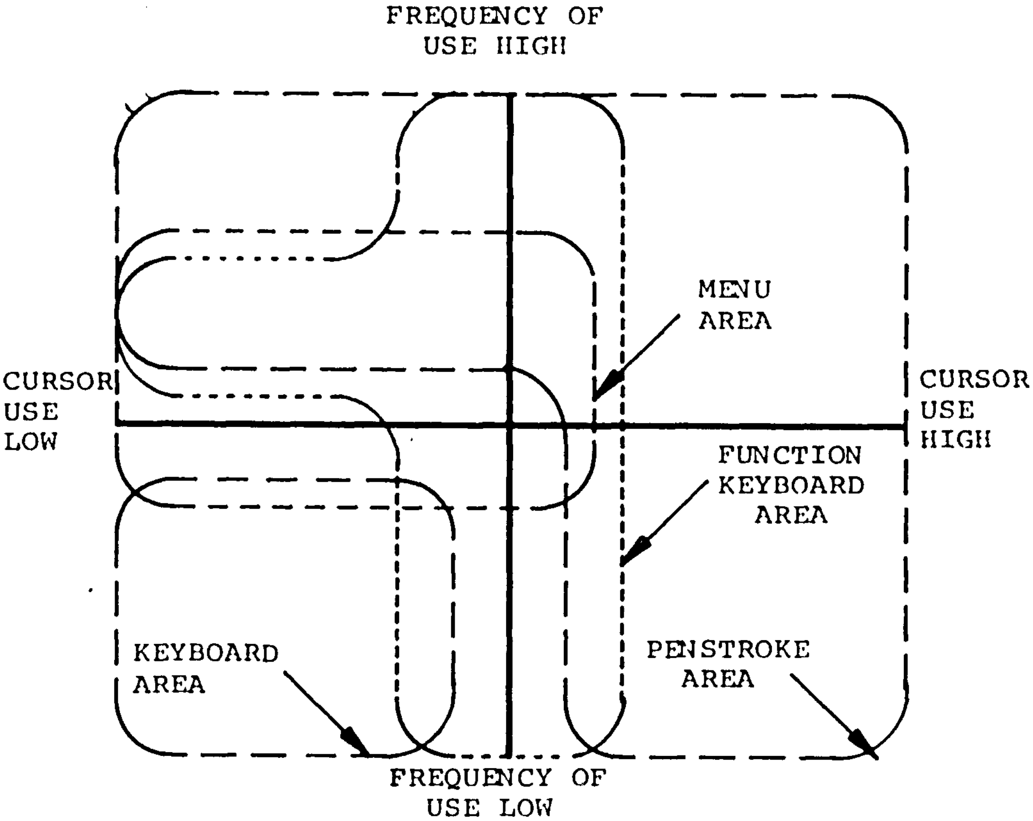


Figure 4.7: Areas for which the different input devices are best suited

# Chapter 5

## VOICE UNIT HARDWARE

The voice recognition unit that was chosen for integration with the interactive graphics system was the Heuristics 7000 Voice Controller. This machine was manufactured by Heuristics of Sunnyville, California and was commercially available for around $3,500. This unit was chosen both for economic and functional reasons. The features that make it particularly suited for input to the graphics system are described in the following paragraphs. Appendix 4 contains details of the Heuristics 7000.

The chosen voice recognition unit has an internal solid state memory of 128 words or phrases. A phrase is a series of words with less than 100 milliseconds of silence between the individual words or syllables.[6] This means it has the ability to recognize up to 128 different phrases; it does not mean it has to have all the allocated spaces filled. More importantly than the memory size, however, is that the voice recognition unit selected has the ability to send and receive this stored vocabulary to and from a host computer. This enables a user to enter a vocabulary once, and have access to it from then on. This also enables the voice recognition unit to be used by more than one person, without having to retrain the vocabulary everytime the users change. It also allows one user to have more than one vocabulary set. Although the programs that enable the transfer of a vocabulary are not provided by the vendor, they are simple enough to be written on sight. The programs that allows this transfer of information to take place are flowcharted in Appendix 5.

Although 128 words or phrases is the maximum vocabulary size at any one time, the user has the option of selectively disenabling and enabling any section of the vocabulary set. These user selected sections or groups of words (up to 255) can be turned on, or turned off at the users discretion. By defining groups and disenabling them when they are not needed the user increases the reliability of the voice controller and the response time. This also makes it possible for two or more users to share the same vocabulary set.

The voice recognition unit used in this thesis study was a self contained unit that was adaptable to the modular configuration of the graphics system. The chosen recognition unit is plug compatible with existing ports in the graphics network. Otherwise stated the interactive graphics system is not corrupted by the addition of the voice controller. Neither the graphics system software, nor its hardware, needs to be modified in order to attach the voice recognition unit. This is not always the case with other voice recognition units. Both the voice controller and the graphics system communicate in ASCII which makes this interface very clean. The modularity of the chosen voice controller mobile. Mobility is also enhanced by the small physical size of the unit (14.5" x 16.75" x 4.5"). The unit itself weighs only 11 pounds, and comes in a self-contained case.

Off-line training of a vocabulary set is also a capability of the chosen voice controller. This enables a user to enter, augument, or adjust a vocabulary set without tying up an entire graphics station. In addition to manipulating the vocabulary set the user has the option of practicing off line. With the use of a standard CRT the voice unit allows new users to practice at saying words consistently. To aid users in both selecting words to teach the unit, and in practicing consistency this voice recognition unit allows the user to display some statistics on a CRT which relate to the audio sound input. The statistics that the unit displays are listed below:[7]

| | |
|---|---|
| LENGTH | How long, in milliseconds, the spoken word was |
| PEAK AMPLITUDE | Maximum amplitude reached when saying the word. Range 0 to 255 |
| AVERAGE AMPLITUDE | A measure of the average amplitude of the spoken word |
| MATCH SCORE | The difference between the spoken word and the matched word. (0 is a perfect match) |
| SECOND BEST FIT | The vocabulary word most similar to the actual matched word. (or *REJECT* if no word was closer than the current reject level.) |

DISTANCE The difference between the matched word and the second best word. (or the reject level if that was the next closest match.)

Figure 5.1 shows an example of these statistics.

| | |
|---|---|
| Word: | "GO" |
| Length of Signal: | 215 |
| Peak Amplitude: | 515 |
| Average Amplitude: | 367 |
| Match Score: | 15 |
| Second Best Fit: | "GOSH" |
| Distance: | 20 |

Figure 5.1: Examples of voice statistics displayed

It was already mentioned that the chosen voice recognition unit allows the user to specify the vocabulary size, and to group selected sections of the vocabulary. This voice unit also gives the user several other options, both in initiating the vocabulary and in accessing it. Most notable of these options are the ability to set the number of samples of a word the unit will accept, and the ability to set the rejection level. By adjusting the number of samples (how many times a user must say a word or phrase for the voice controller to remember or learn it) the user can influence the accuracy of the voice print. The reason for this is that the voice print stored is actually the average voice print of all the times the user said the word while training the machine.[8] Adjusting the rejection level is the means of controlling how close an audio sound must be to a stored audio print before the voice controller recognizes it as a match. Both of these features of the voice recognition unit allow the user to have control over the authenticity of a retrieved word.

The chosen voice controller listens continuously while it is enabled. It can, however, be disenabled very easily to allow for verbal communication between the user and someone else. The gain levels are all adjusted automatically allowing it to adjust to the speaking habits of various users. The chosen unit is also self testing for major problems, in addition it can be adjusted for different parity checks, duplex transmission, and baud rates.

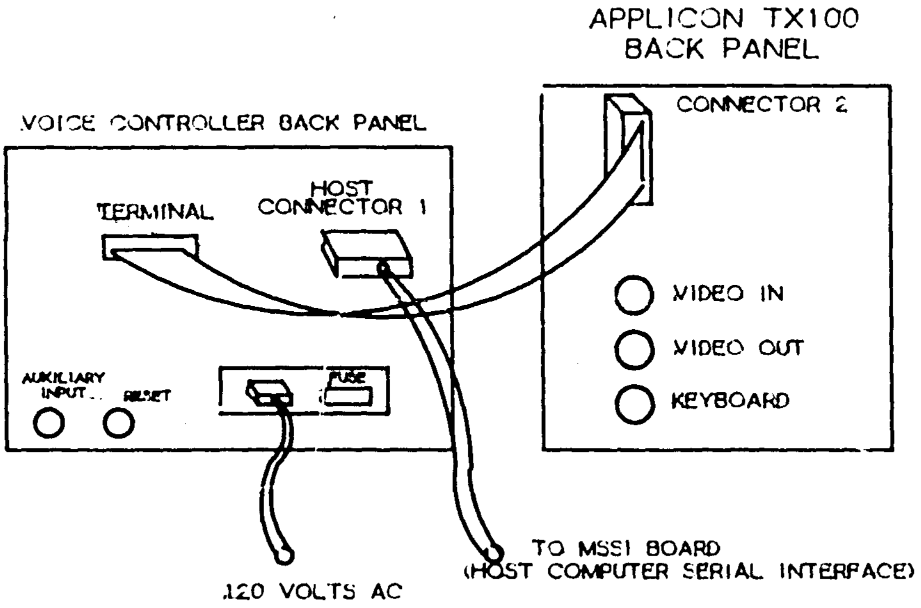
## Chapter 6

## ATTACHING THE VOICE RECOGNITION UNIT

One of the reasons, as mentioned in Chapter 3, the Heuristics 7000 was chosen for integration with the chosen graphics system is its modularity. Because the chosen turnkey graphics system input features are designed on a modular basis, the interface with the voice unit is easily established. The interface between the two devices must be described on two different levels. The first of these levels is the physical link between the two devices. This physical link is accomplished over cables between the voice recognition unit and a port in the graphics system. For the purposes of this study an available port normally used for a teletype was used. In general any available port on the chosen graphics system would probably be acceptable. A cable link between the voice unit and the CRT is also necessary for training and testing purposes. Figure 6.1 shows the physical links necessary for this system.

Once the physical links to the interactive graphics system have been made the user must proceed to logically link the two devices together. This is done by "attaching" the port the voice is linked into the device the graphics screen is defined as. In the case of this study this is done with the command "ATCH TT6 x", where "x" is the screen device number. This "enables" the graphics screen activity to recognize the data coming through the specified port as useful information. This is the same procedure that one follows when linking any of the other input modules (alphanumeric keyboard, function keyboard, menu, or tablet) to the graphics display. For this reason the graphics system sees the voice recognition unit as another input feature. What the graphic system sees is merely a string of ASCII characters, it does not matter to it where this string originated. A source of problems that is often overlooked when interfacing a voice recogrition unit to an already functionable device is that of feedback. For the voice unit to be practical the user must have some form of real-time feedback as to the commands he is entering. This is important both to insure the authenticity of the command entered, and to provide a learning method for the user. When the voice unit is logically linked to the graphics system this feedback is provided automatically. This is the case since most graphics systems automatically echoes back any command entered on the screen.

The simplicity of both the physical and the logical links help to make the efficiency of the voice unit compatible with that of the other devices. Since it is linked in basically the same logical and physical manner as the other input features, voice recognition is also not hindered by being effectively slower than any of the other devices. For the same reason the voice recognition is not prone to any additional mechanical or electronic errors in transmission than any of the other devices.



CONNECTOR 1 --> DB-25 PIN CONNECTOR. MALE ON EACH SIDE

CONNECTOR 2 => DB-25 PIN CONNECTOR FEMALE ON EACH SIDE

Figure 6.1: Diagram of physical links necessary

## Chapter 7

## NUMERICAL CONTROL FOR PART PROGRAMMING

The numerical control package is only one of the software packages available to the user on an interactive graphics system. Other packages available include two-dimensional drafting, three-dimensional drafting, and electrical design. It is likely that voice recognition could be applied to any or more of these other packages. The numerical control package however, provides a very wide range of commands including many that are used in the other packages. For the purposes of this study a specific graphics package was chosen because it allowed concentration on specific commands and provides specific examples of the usefulness of voice input. The objective of this chapter is to provide the reader with an overview of the numerical control package used in this study.

The numerical control package provides a method of producing APT (Automatically Programmed Tool) source code, or a paper tape, for a part to be machined directly from the graphics system. The numerical control package on the chosen interactive graphics system can be divided into the following four divisions:[9]

1. Creating the geometry
2. Defining the tool

3. Preparing the geometry and creating the tool path

4. Creating the Automatically Programmed Tool (APT) source code

Each of these categories will be discussed and examples of common commands in each group will be presented. Appendix 6 contains an example of using the numerical control package using all of these features.

While creating the physical geometry of a part on a turnkey graphics system that is to be applied to the numerical control package the user utilizes many of the commands in the basic drafting packages. These are commands that are used by all users on an interactive graphics system for creating two and three dimensional parts or drawings. Commands of this type are used for adding lines, arcs, circles, conics, and for manipulating them into desired parts. In addition to commands that enter and manipulate the geometry there is also a set of commands used in this division for controlling the geometry. Several of these commands that are used in this section are listed in Table 7.1. Table 7.2 lists several commands from this division of numerical control that were selected for voice input using the rules expressed in Chapter 8. Neither of these two tables is complete, but an effort was made to include examples of all types of commands appropriate to this section. The major proportion of commands in this division of the

| Command | Definition |
|---|---|
| GO | Repaint screen |
| GOSH | Update the screen |
| KILL | Ignore the last command |
| GRID | Turn grid on or off |
| CVUE | Center the screen |
| WW n | Set window width to n |

Table 7.1: Examples of commands used for controlling the geometry

| Command | Word | Definition |
|---|---|---|
| GO | Go | Repaint screen |
| GOSH | Gosh | Update the screen |
| KILL | Kill | Delete the last command |
| SE | Select every-thing | Select all geometry |
| RE | Remove | Remove selected geometry from the screen |
| UU | Unselect | Unselect the selected geometry |

Table 7.2: Examples of commands used for controlling the geometry selected for voice input

numerical control package that are appropriate for voice input are from the control set. As described in Chapter 8 this is the case because these types of commands are commonly used with high frequency and require low cursor control. Also, because the numerical control program requires all geometry to be entered in a special form (called Polyarcs) there are certain extra control commands necessary.

Defining a tool to be used to cut a part on the selected numerical control graphics package is done through a prompting sequence. This sequence is accessed by executing the command "APTP." The system then proceeds to ask the user to input certain parameters and information that pertains to: tool identification, tool geometry, cutting information, and entry and exit information.[10] Specific examples of the parameters and information requested by the numerical control package are listed in Table 7.3.

The nature of the information that is entered in response to the system prompting can be described in two ways. The first way to describe this data is that it is digital in nature. For example, it contains information such as .0005, .025, 2, or 315. The second way to describe this data is that it is infrequent. It is infrequent both in the sense that the same data is usually not entered more than once in the same prompting sequence, and in the sense that the prompting sequence is not activated very often. The one exception is the numerical zero (0), since it is used very often in describing the tool geometry and location.

In the section of the numerical control package that prepares the geometry and creates the tool path there are several subdivisions. As when the geometry is manually prepared for applying APT commands to it, the first step in most graphic numerical control packages is to label the geometry. To do this on the chosen graphics

Tool #
Rough feed
Finish feed
Spindle speed
Coolant (on / off)
Inside tolerance
Outside tolerance
Entry point

Table 7.3: Examples of the parameters requested by the numerical control package

system requires the user to first change certain definitions of the geometric type. This is done by accessing several commands in the numerical control package. Once the definitions are changed and the geometry to be labeled is selected, a single command is used to label the geometry. This command is "Label Lines" and its result is a part labeled in the manner of APT. This is shown in Appendix 6. To create the tool path the user must again change definition. The definition he changes to is that of the tool he would like to use. In this manner the numerical control package is informed of the tool the user desires.

Once the geometry is entered, labeled, and the definitions are changed to permit the graphics system to know what tool to use, a tool path is created. This is done with a single command usually defined as a penstroke since cursor control is required. For the system used in this study this penstroke is a series of three dots and tells the numerical package three things. The three objectives satisfied by this penstroke are as follows:

1. Communicates to the computer that the user wishes to add a tool path
2. Communicates to the computer the point it is to begin cutting at
3. Communicates to the computer the direction the tool is to cut

The applicability of voice recognition for input to this phase of numerical control programming is limited to the following two subsets:

1. Control commands (GO, KILL, UNSELECT, etc.)
2. Changing definition (USE LPT 7, CDEF PTX, USE P1 12, etc.)

After the tool path is created the user has several options on how he would like the APT information displayed. All of these options are accessed by a different command on most graphics systems. The options available to the user are listed in Table 7.4. Because each of these is a single command it is easy to apply them to voice recognition.

APT Motion statements
APT Geometry statements
APT Motion and Geometry Statements
APT Cutter location file

Table 7.4: Options available on display of APT information

## Chapter 8

## COMMAND SELECTION USING VOICE

One of the main points of Chapter 2 was to point out to the reader that each input device has a certain type of command for which it is best suited. The alpha-numeric keyboard is best suited for low use commands requiring many different arguments, and very little cursor control. The menu provides a means of entering infrequently used commands that require a greater amount of cursor control efficiently. The function keyboard is best suited for entering moderately used commands, regardless of the amount of cursor control required. Finally penstroke recognition is most efficient for highly used commands. This is shown in Figure 4.7. Voice recognition likewise is best suited for a certain type of command. As shown in Figure 8.1 these are commands that are used very often, but have low to moderate requirements in cursor control.

Although the voice recognition method of entering a command, like any of the other input methods, could be used for any recognizable commands; it is important to assess its strength and weakness. The main weakness of using voice recognition for input is similar to the main weakness of pen recognition. This weakness is that of remembering what word is associated with what command.

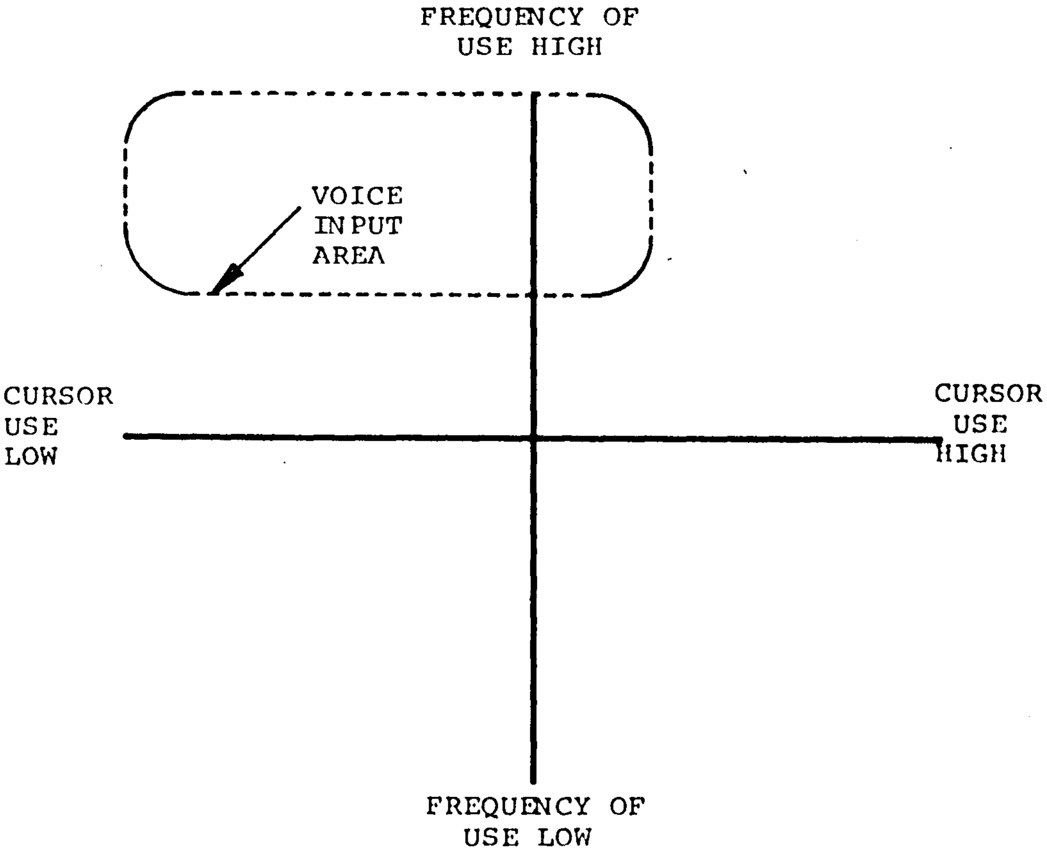


Figure 8.1: Area for which voice recognition is best suited

With the highly used commands this will not be a problem, but with commands that are not used very often this is a major weakness of using voice recognition for input or manipulating of geometry. This problem can be somewhat alleviated by attaching a cue card for user reference (such as on the bottom of the screen). This solution, however, does not totally alleviate the problem, therefore voice is best applied to highly used commands that the user can remember the access word to. Also, like the penstroke recognition method of inputting data the user runs the risk of having two access words sound similar to the voice recognition unit. The problems of the voice recognition unit confusing two words is increased as the number of words programmed into the unit is increased. For these reasons the capacity of the voice recognition unit is limited by more than the storage capability of the specific machine used.

Cursor control using the pen to control the cursor while the voice is used to select the appropriate command is also possible with voice recognition. As mentioned before this is the same manner that the function keyboard uses to control the cursor. The means of doing this is, however, with voice input not easy, and it is further complicated with commands requiring complex cursor control. By complex cursor control it is meant that the cursor must be used to select two or more points of the screen. The voice unit relays the complete command at once to the graphics system. It is difficult to sufficiently delay this process for the interactive graphics system to recognize more than one cursor location. This is the reason that voice recognition is best suited for the moderate to lower levels of cursor control requirements.

Macros as explained in Chapter 4 are a string of commands that are strung together and can be accessed by a single command on any of the input devices. An example of a macro is listed below:

| Macro | Meaning |
|---|---|
| OUT | CTRL; GO; STOR @; |

The voice recognition unit as mentioned before sends the whole string of commands programmed to the graphics system at once. The graphics system, however, has no built-in buffer to store a string of commands. For this reason the voice unit cannot send a string of commands such as the one listed below and expect it all to be executed:

UEDT; EDT; GO

To access a string of commands such as the one above using voice recognition the user must define it as a macro. This macro must be defined in the graphics system itself, and be set up to be assessed by a command on the voice system. The phrase used to access the string above could be "EDIT-LEVEL-ONE." The recognition unit would send this command to the graphics system and it would execute the programmed string of commands.

Despite some of the limitations on using voice recognition for input or manipulation of geometry in a graphics system it has some very strong advantages. Of these advantages, and perhaps the greatest advantage of using voice input is speed. As mentioned before the graphics systems do not distinguish between the input devices when they receive a command. Because of this feature on interactive graphics systems, any advantage of speed must be obtained in the processes that occurs before the command (in ASCII) is sent to the graphics processor. To explain why speed is an advantage of voice recognition one must analyze the process of sending a command to the graphics network. This process is shown in Figure 8.2. The time difference in entering

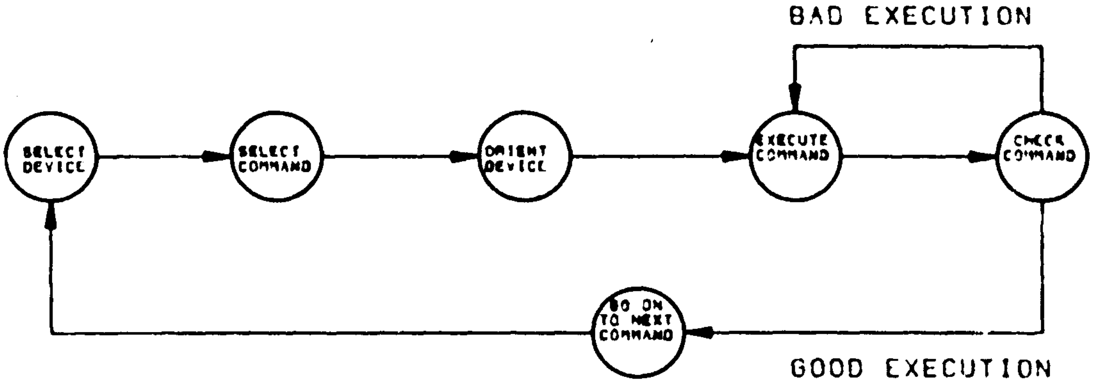


Figure 8.2: Process of sending a command to the graphics system

a command to the graphics system using the different input devices is contained in the execution block of Figure 8.2. For the purpose of this argument it will be assumed that the "hit rate" or percent accuracy of each device is the same.

Using the alpha-numeric keyboard, because of the complex orientation and the physical typing involved, is the slowest of the five methods mentioned. This method requires of the user the most orientation of all the devices, and the most physical dexterity. It also slows user input since he or she must rearrange the pen in his hand, or set it down, to access the keyboard properly. The menu because of its size is the second slowest of the input devices. The process of finding the proper switch on the menu can take quite a while. Often the pressing of the pen on the menu is misinterpreted as a penstroke. Although it is important to realize that voice recognition is faster than the alpha-numeric keyboard and the menu it is not necessarily all that relevant. One must remember that the voice recognition method of entering commands is not competing for the same subset of commands that either the alpha-numeric keyboard method or the menu method is. For this reason it is important that voice recognition have advantages over the function keyboard and penstroke recognition, since these are the methods to which it is to be compared.

For the user to use the function keyboard to enter a command he must first orient it with his hands and eyes. This requires a certain amount of time, but more importantly it forces the user to look away from the graphics screen. This distraction of looking away from the screen requires of the user time that would not be spent using voice recognition. Although using penstroke recognition to enter a command does not require the user to orient the pen (unless cursor control is required) it still requires manual manipulation. While manipulating the pen the user must be certain to tract and ink in the proper order. Due to the static electric build-up of a charge on the tablet this is sometimes difficult. The main reason that voice input may be faster than any of the other input processors is that it saves at least one step in the input process. For any other device you must choose the device, choose the command, orient the device, and execute the command (this is shown in Figure 8.2). Using voice recognition for input requires no orientation. More importantly, if the vocabulary set is chosen correctly, choosing the command and executing it becomes one step. This is because the user typically vocalizes the command to himself before he tries to pair it with a function on one of the devices. One of the criteria for a good vocabulary set, as will be discussed in Chapter 9, is that the user be able to vocalize using it. Since there is no extra requirements for training the graphics system to accept voice data, as compared to the other devices, this is not considered as part of the speed function.

Other than speed, using voice recognition to enter information has the advantage of being very natural. Speaking is a very natural process and does not have to be learned as does typing and using the pen. Although speech recognition does require the user to refine his voice and practice at being consistent this is a skill that is naturally acquired. Also since it is natural for a user to vocalize a command to himself before he tries to execute it this aids in remembering the access words. For commands that are instinctively vocalized (such as "GO", "KILL", "EDIT", "UNSELECT", "DELETE") voice recognition is at its best because of speech being such a natural process. A list of commands (applied to Numerical Control) that are well suited for voice application is listed in Appendix 7.

Chapter 9

## WORD SELECTION USING VOICE

Word selection refers to selecting the access words that will be associated with each command that is to be entered using voice recognition. These access words may be single or multi-syllable words or groups of words called "phrases." The only limitations placed on them by the hardware device (the Heuristics 7000 voice recognition unit) is that they be under three seconds in length, and have no pauses greater than 100 milliseconds.[11] In theory any words, utterances, or phrases that meet these two criteria could be used. In practice, however, there are two practical limitations on the words selected which must be met in order to have an effective input device. The first of these limitations is that any word selected must be identifiable with the command with which it is associated. The second practical limitation is that each word must be unique to the vocabulary set as interpreted by the recognition unit. The reasons for these two limitations and their consequences will be discussed in the following paragraphs.

One of the major justifications for applying voice recognition for inputting commands and geometry to a graphics system is speed. This speed, as explained in the preceeding Chapter, is obtained by eliminating or reducing the time lag between selecting a command and executing it. Part of this reduction in time is made possible because people tend to vocalize in their minds; that is they think in terms of words or word phrases. It is felt the inherent difficulty for people to master thinking in abstract penstrokes or menu positions, which are the languages of the other input devices, causes voice recognition to save a mental step. The mental step that can be saved is that of translating the vocalized word to a penstroke, menu position, function keyboard position, or alpha-numeric command. This mental step is only saved, however, if a user can learn to vocalize using the words in a specific vocabulary set. To aid the user in training himself to vocalize in this manner it is necessary to select words that are logically linked to the commands that are to be executed. Ideally it would be best if the words chosen were the ones that most users already vocalize in. In some cases, as in the commands shown in Table 9.1 below this is possible.

| Command | Word |
|---|---|
| GO | Go |
| KILL | Kill |
| DD | Delete |
| GG | Gosh |
| RE | Redraw |
| SE | Select everything |

Table 9.1: Commands that are already vocalized

Because of the nature of many commands it is impossible to use the common vocalized word as the access word without violating the second practical constraint on word selection. In cases like the ones shown in Table 9.2, words had to be chosen that are logically linked to, but not already vocalized as, the command.

| Command | Definition |
|---|---|
| UEDT n | Unedit level n |
| EDIT n | Edit level n |
| USE P1 n | Use P1 on level n |

Table 9.2: Commands that are not already vocalized in a manner efficient for voice input

The first constraint as described above dealt mainly with the problem of association for both remembering the access word and the speed of its use. The second constraint deals with the authenticity of the retrieved word or command. To insure the integrity of the system the user must be certain that each access word in the vocabulary is unique. "Uniqueness" in this case implies that each word selected to be used is sufficiently different from all other words in the vocabulary set to insure that recovery errors will be minimal. The uniqueness of the vocabulary set has a great impact on the hit rate or recovery rate of the voice unit.

To describe "uniqueness" in more scientific terms the reader must be reintroduced to a concept that was briefly mentioned back in Chapter 3. This concept is that of breaking speech, or any audio signal, down into its fundamental units. These basic units of speech are called "phonemes".[12] Phonemics is the science of describing speech using phonemes. The English language commonly consists of about forty phonemes which can be used to describe any English word. When two words are different by only one phoneme they are called a "minimally distant pair."[13] Examples of minimally distant pairs are shown in Table 9.3.

| | | |
|---|---|---|
| ball | -- | tall |
| boy | -- | toy |
| table | -- | cable |
| cat | -- | hat |

Table 9.3: Examples of "minimal distance pairs"

The electronics used for filtering, and the fitting algorithms used in voice recognition devices have a lot of problems with determining uniqueness of minimally distant pairs. The concept of minimal distant pairs can be extended to phrases that are the same except for a small part of them. Examples of some of these phrases are listed in Table 9.4. It should also be mentioned that the concept of minimal distant pairs makes entering numbers and individual letters using voice recognition difficult.

| Phrase 1 | Phrase 2 |
|---|---|
| Edit level one | Edit level two |
| Unedit level one | Unedit level two |
| Use P1 level one | Use P2 level two |
| Reference level one | Reference level five |

Table 9.4: Examples of the "minimal distance phrases":

A list of words selected, along with the associated commands for numerical control work on the selected graphics system appears in Appendix 8. What is more important, however, is that the reader and the user of voice input understand why some words are better suited than others for a particular application. For this reason a flowchart of the procedures that should be followed while selecting a word for the vocabulary set is also presented in Appendix 8. The user must also have a say in the words selected for his vocabulary set. This will insure that each user feels comfortable with the vocabulary set chosen and takes into account any speech pecularities of a user. Like penstroke recognition, voice recognition can and should be tailored to the particular user. What is suggested in Appendix 8 is merely a guideline to follow.

# Chapter 10

## DESIGN AND METHODOLOGY

This experiment is designed to measure the effect of certain environmental factors on the recognition of a vocabulary by the Heuristics 7000 voice recognition unit. The dependent variable (Y) represents the number of hits in which the goodness of fit statistic given by the recognition unit is below 50. This statistic is an integer between zero and fifty, with zero being a perfect fit and fifty being very poor but still acceptable. If "no fit" is made the statistic is assumed to be greater than fifty. "No fit" is defined as:

1. No word was selected and the audio sound was rejected, or
2. The wrong word was selected as the best fit.

There will be three independent variables (X's) in this experiment, they are listed below:

1. Background or white noise
2. Number of words in the vocabulary set
3. Length of time after storing the vocabulary that it is used

For the variable of background noise there were two levels tested, one with background noise and one without. In both cases the vocabulary was initially stored in the voice recognition unit using no white noise. Because a pure white noise, such as a constant hum, would be completely overlooked by the Heuristics voice controller, a white noise of variable quality was used. The white noise was a tape of a radio that was synchronized with the vocabulary tape. By synchronizing the two tapes it was possible to gain control since the same background noise was used in each case where it was appropriate.

The effect of the number of words in the vocabulary was also tested at two levels. The low level consisted of twenty words that were found to be distinctly different by the recognition unit. The high level consisted of the twenty original words and an additional set of twenty words. The words used in the vocabulary sets were words associated with the Applicon Numerical Control Package.[14]

Once the vocabularies are stored in the computer they can be retrieved at any time, theoretically neither the storage process, nor the passage of time will effect their purity. This was the hypothesis that was tested by the third independent variable. Data was collected on the two vocabulary sets (twenty words and forty words) immediately after storing them in the computer, and again three days after they were stored.

The data for this experiment was collected by selecting two hundred uniformally distributed random numbers.[15] These 200 random numbers were used for two things; the first was to determine the frequency of the words to be tested. The second use for the random numbers was to select the order that the words were tested in. This would mean that each word in the twenty word vocabulary set was tested an average of ten times, in random order. Each word in the forty word vocabulary set was tested an average of five times, in random order. Both the frequency and the order of the words has been held constant throughout all iterations of the experiment.

Control in this experiment was obtained by using prerecorded tapes to both generate the two vocabularies, and to test the vocabularies. These tapes were made in a music studio to eliminate any possible recording or electronic errors. By using these tapes we have eliminated the operator from consideration in this experiment by making him a constant. Control was also obtained by chosing the words that were tested in random order and frequency in accordance with the described distribution.

The analysis of the data in this experiment was that of a 2 by 3 factorial designed experiment. The eight conditions that the experiment generated data for are described in Figures 10.1 and 10.2 on the following page. This method of analysis allows for the determination of the importance  of the three variables, and the importance of the interaction of these factors with one another, on the recognition of a word.

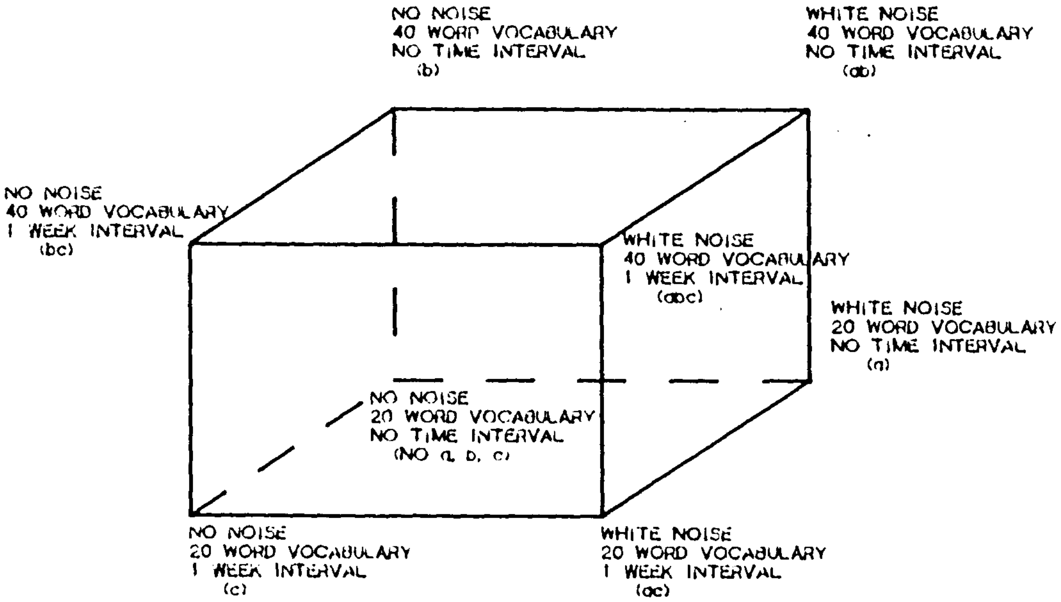


Figure 10.1: Graphical representation of the 2 by 3 factorial experiment

| WHITE NOISE | | | | NO NOISE | | | |
|---|---|---|---|---|---|---|---|
| 20 WORDS | | 40 WORDS | | 20 WORDS | | 40 WORDS | |
| NO TIME | 1 WEEK | NO TIME | 1 WEEK | NO TIME | 1 WEEK | NO TIME | 1 WEEK |

Figure 10.2: Tabular form of the 2 by 3 factorial experiment

## Chapter 11

## RESULTS AND ANALYSIS

The results of the experiment discussed in the previous chapter are presented in Figure 11.1. The numbers in the parentheses are the times that no-fit was made under the given conditions. These figures are also presented in Table 11.1.

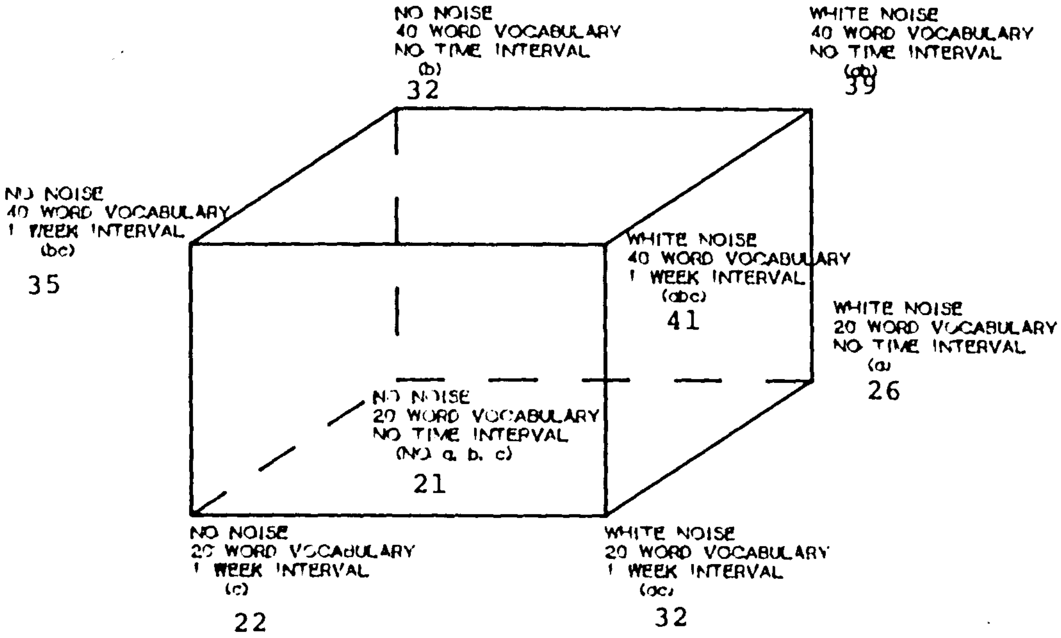


Figure 11.1: Graphical representation of the data collected

| | | | | | |
|---|---|---|---|---|---|
| 1 | = | 21 | c | = | 22 |
| a | = | 26 | ac | = | 32 |
| b | = | 32 | bc | = | 35 |
| ab | = | 39 | abc | = | 41 |

Table 11.1: Data collected

As stated in Chapter 10 the analysis technique that was used to analyze the data was that of a 2 by 3 factoral experiment. This analysis technique provides a method of determining what factors of the experiment are important, or significant. In addition to the individual factors, however, it also provides a method for determining the significance of the interactions of the individual factors. The math of this analysis is presented below in Table 11.2 and in Table 11. on the following page.

$$\begin{aligned}
A &= -1 + a - b + ab - c + ac - bc + abc \\
&= -21 + 26 - 36 + 39 - 22 + 32 - 35 + 41 \\
&= 24
\end{aligned}$$

$$\begin{aligned}
B &= -1 - a + b + ab - c - ac + bc + abc \\
&= -21 - 26 + 36 + 39 - 22 - 32 + 35 + 41 \\
&= 50
\end{aligned}$$

$$\begin{aligned}
AB &= +1 - a - b + ab + c - ac - bc + abc \\
&= +21 - 26 - 36 + 39 + 22 - 32 - 35 + 41 \\
&= -6
\end{aligned}$$

$$\begin{aligned}
C &= -1 - a - b - ab + c + ac + bc + abc \\
&= -21 - 26 - 36 - 39 + 22 + 32 + 35 + 41 \\
&= 8
\end{aligned}$$

$$\begin{aligned}
AC &= +1 - a + b - ab - c + ac - bc + abc \\
&= +21 - 26 + 36 - 39 - 22 + 32 - 35 + 41 \\
&= 8
\end{aligned}$$

$$\begin{aligned}
BC &= +1 + a - b - ab - c - ac + bc + abc \\
&= +21 + 26 - 36 - 39 - 22 - 32 + 35 + 41 \\
&= -6
\end{aligned}$$

$$\begin{aligned}
ABC &= -1 + a + b - ab + c - ac - bc + abc \\
&= -21 + 26 + 36 - 39 + 22 - 32 - 35 + 41 \\
&= -2
\end{aligned}$$

Table 11.2: Calculations of the experiment

| Factor | df | Contrast | Sum of Squares | Mean Squares | F Statistic |
|---|---|---|---|---|---|
| A | 1 | 24 | 72 | 72 | 144 |
| B | 1 | 50 | 312.5 | 312.5 | 625 |
| AB | 1 | -6 | 4.5 | 4.5 | 9 |
| C | 1 | 8 | 8 | 8 | 16 |
| AC | 1 | 8 | 8 | 8 | 16 |
| BC | 1 | -6 | 4.5 | 4.5 | 9 |
| ABC | 1 | -2 | 0.5 | 0.5 | --- |

Table 11.3: Calculations of the experiment continued

Using the statistical tables for the F-distribution[16] (for 95% confidence interval, one degree of freedom in the numerator, and one degree of freedom in the denominator) a level of significance of 161.4 was determined. By comparing this number to the numbers in column 5 of Table 11.3 one can see that only one of the variables tested has a significant effect on the results of the experiment. This factor was the size of the vocabulary set. It should be noted that although the effect of background noise was found to be considerably higher than the effect of storage time, or the effect of any of the interaction terms, it was not found to have a significant effect on the results.

## Chapter 12

## CONCLUSIONS

Throughout this thesis there have been many conclusions reached using both a analytical method and a deduction method. This chapter will attempt to bring together these conclusions for simplification and ease of reference. An attempt will be made to explain how each of these conclusions was obtained and what the implications of each of the conclusions are.

The first conclusion reached in this thesis concerned commands to be chosen for voice application. By analyzing the strengths and weaknesses of voice recognition it was deduced that certain types of commands are best suited for voice recognition. As shown in Figure 8.1 these are commands that are used very often, but have low to moderate requirements in cursor control. As a result of this conclusion there are more commands in the control set (such as "GO", "KILL", "GOSH", etc.) than in the addition and manipulation of geometry set (such as "ADD", "MOVE", "COPY", etc.) that are appropriate for voice input. The arguments and discussions of command voice are presented in detail in Chapter 8.

The next conclusion reached in this thesis was concerning word selection. This was done by analyzing the basis of speech and the process of using the voice unit.

Two specific conclusions were reached; the first being that the access word selected for a given command be logically associated with that command. This conclusion was reached to aid the user in remembering the access words for the given application. It can be concluded that an ideal case would occur when the user vocalized in the vocabulary set given. A second conclusion reached concerning word selection is that elimination of minimally distant pairs, words, or phrases from the vocabulary set is desirable. This conclusion was reached after considering the basic units of speech called "phonemes." In order to insure the uniqueness of a word selected for the vocabulary the word should be examined in terms of these phonemes. This is the case because the electronics of the present-day voice recognition units have a hard time in distinguishing between minimally distant pairs. The result of these two conclusions is a flowchart presented in Appendix 8. This flowchart is a suggested procedure to follow when trying to select words to apply to voice recognition. The details of these conclusions and the arguments for them are discussed in detail in Chapter 9 of this thesis.

The final set of conclusions reached in this thesis were the result of the experiment discussed in Chapter 10. The data and analysis of this experiment is presented in Chapter 11. As shown in Chapter 11 the results indicate that of the three factors tested (background noise, vocabulary size, and storage time) only the vocabulary size was found to have a significant effect on the retrieval accuracy of the voice recognition system. By the statistical analysis used the two other factors were found to have no significant effect on the retrieval accuracy. As a result of this experiment it was determined that excess words in the vocabulary set have a detrimental effect on the entire voice recognition network. It was concluded that the smaller the vocabulary size the better the hit rate would be if everything else were equal. In a similar manner it was concluded that reasonable background noise and storage time have no effect on the efficiency of the voice recognition unit.

# Chapter 13

## SUMMARY

The goal of this study was to determine whether or not the efficiency of a turnkey graphics system can be improved by applying the technology of voice recognition for input. Although both the graphics system and the voice recognition unit used in the experiments were of a specific vendor type, all arguments were presented in a generalized and generic format.

An effort was made in this thesis to familiarize the reader with the general technologies of voice recognition and interactive graphics. Chapters 3 and 4 were dedicated to this cause. Chapters 5 and 6 were presented to introduce the reader to the methodology of attaching the chosen voice recognition unit to the graphics system. In these chapters it was shown that neither complex physical or logical links are needed to make the interface between the two hardware devices. With the basic technologies explained, some problems concerning voice input could be addressed.

Some of the problems concerning voice input discussed in this thesis were:

1. Commands to be applied to voice input
2. Words to be associated with the chosen commands

3. Environmental conditions and their effect on recognition

Arguments and experiments concerning the solution to these problems were presented in Chapters 8 thru 11. In order to make these arguments more relevent, they were presented in a general manner but applied to a specific application. The application they were applied to was the numerical control package on the Applicon Interactive Graphics System. This package was explained in Chapter 7 of this thesis. The conclusions expressed in Chapter 12 are the result of these arguments and experiments.

## Chapter 14

## RECOMMENDATIONS FOR FUTURE WORK

Although this thesis demonstrated the feasibility of using voice recognition both in the application and technical sense, other work needs to be done. One of the things that requires further research is the association of words and commands. Research and experiments should be done to determine what makes a word remembered in terms of its associated command. Along this same line, experiments should be done to find out if an access word is more easily remembered than a penstroke. If these two things were researched, the process of finding and chosing access words would be greatly enhanced. The quality of the chosen access words would also be improved.

Another area where future research could be applied is in the area of off-line training of the voice unit. The feasibility of connecting the voice recognition unit to a small computer such as an Apple, for the purpose of training and practicing should be looked at. This would allow for the user to use the voice recognition unit for training and practicing purposes without tying up a high productivity graphics device.

Other areas of possible future research include controlling the direction of the cutter and voice output. By controlling the direction of the cutter by voice, it is possible that a productivity gain would be realized. The benefit of voice output would possibly be quicker and more recognizible feedback.

Appendix 1

## GLOSSARY OF TERMS[17]

Term

Definition

Add

To sum, by computer, like items or parts augment a design or drafting image with further graphic, dimensional or alpha-numeric information.

Add Text

Type at work station keyboard additional alpha-numeric notation which is electronically placeable at pre-selected locations.

APT

A computer language used to control N/C machine tools.

Assemble

Electronically place related units or parts into predetermined positions as in arrangement and assembly drawings. Units or parts may be electronically copied from other drawings in the data base.

Algorithm

A predefined sequence of steps to be taken to solve problems of a particular type; a procedure attributed to an Uzbek mathematician: Al-Khwarizmi in 9th century.

ASCII

American Standard Code for information interchange. Standard data in specific groupings for computer application.

Baud

A unit of signalling speed equal to the number of discrete conditions or signal events per second. For example, a terminal operating asynchronously requires 10 bits of information per character communicated. If each bit is represented by 1 of 2 possible line states, each bit requires 1 signal event. Thus 9600 baud communication would be equivalent to 960 characters per second.

BCD

Binary Coded Decimal. A standard coding pattern for computer application.

Term Definition

CAD

Interchangeable: Computer-Aided Design or Computer-Aided Drafting; a generic term used in USA, Europe, and Japan. Results of user's input are viewed, tabulated or drafted (plotted) in seconds or minutes.

Cathode Ray Tube, (Refresh)

Specialized type of CRT where screen image is formed by continuously panning electron beam over phosphor-coated tube face. Unless refreshed (repainted) the screen will fade in about 1/30 second.

Cathode Ray Tube, (Storage)

Also a specialized CRT where screen image is maintained indefinitely by a low grade voltage, after once being painted on the screen.

Data Base

A collection of interrelated data items organized by a consistent scheme that allows one or more applications to process the items without regard to physical storage locations.

Delete

Remove electronically from data base; selectively remove portion of symbol or drawing segment, linework or text.

Dimension

Automated placement of dimension lines plus witness lines with types or computer-supplied dimensions to denote size of elemental shapes and areas as well as locations.

Drawing Section

A portion of complete drawing; a repeatable and electronically copied, reduced or enlarged drawing segment for use on other drawings--with or without modification; a combination of symbols and/or text.

EBCD

Extended Binary Coded Decimal. A standard coding pattern for computer application.

Term Defination

Edit
Review or proofread on screen for possible revision; perform revision electronically.

Erase
Same as delete.

Grid
A design/drafting aid available as a placement background on the CRT screen. Grids may be square or stretched in X, Y or Z. Hence 10 x 10, 8 x 8, etc, grids; isometric grids, profile grids.

Human Factors Engineering
Hybrid offspring: engineering plus experimental psychology. Aim: to enhance the effectiveness of the human-machine interface in the use of a CAD system.

Interactive
A technique of user/system communication in which the system immediately acknowledges and acts upon requests entered by the user at a work station.

Isometric
A form of drawing projection, plotted and drafted manually, wherein three faces of an object or feature are shown on three major axes 120 degrees apart and where the angle the front edge makes with the vertical is 35 deg. - 16 min. True isometrics are to scale drawings.

List
A command to request a list of items be printed by the system printer.

Macro
Directions that generate a known set of instructions. Used to eliminate the need to write a set of instructions which are used repeatedly.

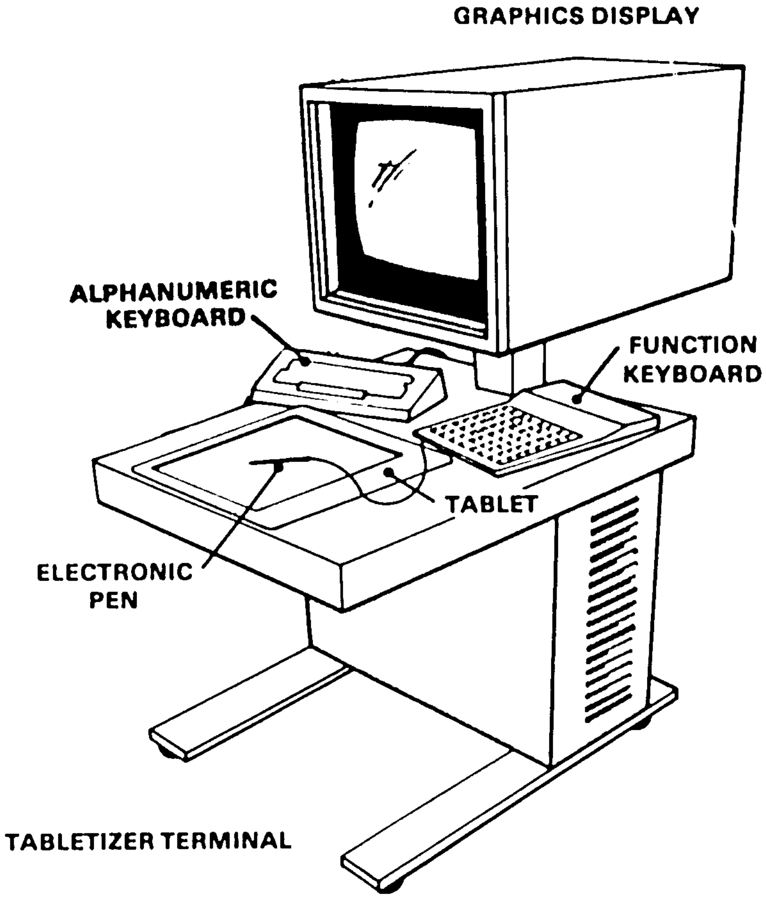
Menu
An area of the digitizing tablets reserved for an array of commands. Allows choosing the commands with the stylus.

Minimal Distant Pairs
Two words, phrases, or audio sounds that are acoustically very similar. Example boy and toy.

| Term | Definition |
|---|---|
| Move | Electronically move a group of items with a single command. |
| Orthographic-Projection | The projection of a point from one plane to another such that the original point and its image lie on a line perpendicular to the viewing plane. |
| Password | A word or code required to gain access to the system. |
| Plot | A command given to the system to get hardcopy output on a variety of plotters. |
| Repaint | Electronic "redraw or re-lettering" of latest status of a design, used right after a revision or edit has been executed, for operator to view correctness of revision or edit. |
| Revise | Alter data base--electronic erase and replace, seldom redraw or retype. |
| Window | Portion of larger design area, in 2-D or 3-D, filling the screen vertically and horizontally; the "distance" from which operator sees drawing or model; relatable to "scale". |
| Zoom | An electronic enlarge-reduce technique for changing scale on screen--infinitely, up or down. |

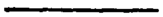
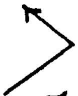
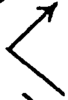
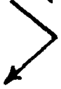
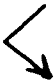
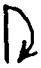
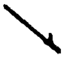
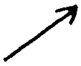
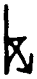
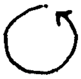
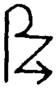
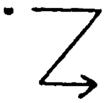
Appendix 2

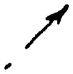
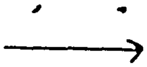
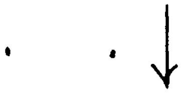
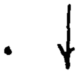
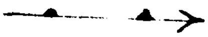
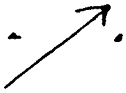
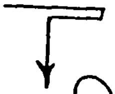
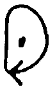
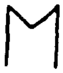
# GRAPHICS STATION LAYOUT[18]



## Appendix 3

## COMMON PENSTROKES[19]

| Penstroke | Command | Definition |
|---|---|---|
|  | ADD | Add a component |
|  | GO | Repaint the screen |
|  | GOSH | Update the screen |
|  | GRID | Turn grid on or off |
|  | CVUE | Center view |
|  | DD | Delete |
| | UU | Unselect |
|  | SELC | Select |
|  | SELN | Select near |
|  | KILL | Disregard last command |
|  | ZOOM | Expand a portion of the screen |
|  | PZ | Show the view before zoom |
|  | ATXT | Add text |

| Penstroke | Command | Definition |
|---|---|---|
|  | ADDV | Add vertex |
|  | MOVE | Move selected geometry |
|  | COPY | Copy selected geometry |
|  | ROTA | Rotate selected geometry |
|  | BRIK | Extend selected geometry into third dimension |
|  | DIST | Distance between two points |
|  | TEAC | Enter teach mode |
|  | UNDR | Remove the selected geometry from the screen |
|  | VMOR | View more of the screen |

## Appendix 4

## DETAILS OF THE VOICE UNIT CHOSSEN

Interface and language compatible: The 7000 Voice Controller interfaces with RS-232-C terminals and is compatible with all programming languages, including FORTRAN, COBOL, PASCAL, and BASIC.

Plug-to-plug compatible; The 7000 is plug compatible with most terminals and to most standard ports on a host computer.

Transparent to host and terminal: When the 7000 is in terminal mode, it has no effect on the information passing between the host computer and the terminal.

64 word/phrase vocabulary: The Voice Controller can be trained to recognize up to 64 words or phrases. A 128 word vocabulary is optionally available.

User trained with isolated phrases: When teaching the unit which words or phrases to recognize, the operator says each word into the microphone at least one time. The operator isolates each word, by leaving at least 100 ms of silence between words. A phrase can be entered as one vocabulary word by leaving less than 100 ms of silence in the phrase.

User-defined vocabulary groups: When only part of the 64-word vocabulary is needed, the vocabulary can be limited temporarily to just the words that are needed by defining those words as a group. Up to 255 groups can be defined and turned on and off independently. When the complete vocabulary is needed, the 7000 returns to the full set of templates. No retaining is necessary.

Programmable rejection level: The 7000 automatically rejects sounds that are significantly different from the trained words in your vocabulary. The amount of difference required to reject a word anywhere from a perfect match to acceptance of any sound, can be adjusted.

Listens continuously: The 7000 is always listening for audio input when recognition is enabled.

Automatic gain control on all audio inputs: All gain, or volume of speech, adjustments on the 7000 occur automatically. The unit adapts to any normal speaking level.

Automatic self test and fault isolation: A self-test is built into the 7000 to check its operating status.

Response time for recognition: The "worst case" resopnse time is 94 ms: 30 ms + 1 ms per word in the vocabulary.

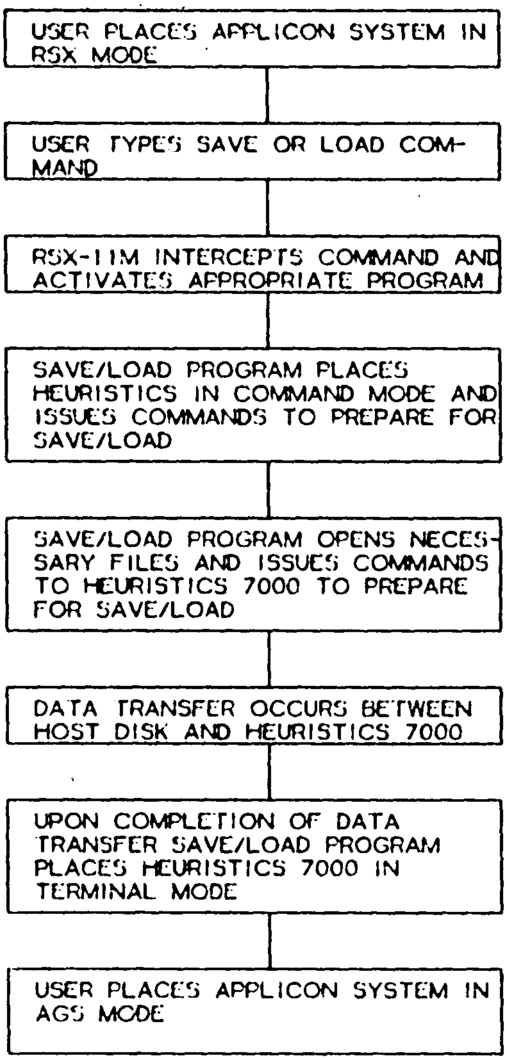
ASCII string sent to host computer: The Voice Controller responds to recognized words by transmitting a stream of up to 255 ASCII characters to the host and/or terminal.

Use host computer to store vocabulary: The Save Vocabulary and Read Vocabulary commands enables the 7000 to store any number of complete vocabularies on the host computer. Each vocabulary can then be retrieved for use whenever needed.

## Appendix 5

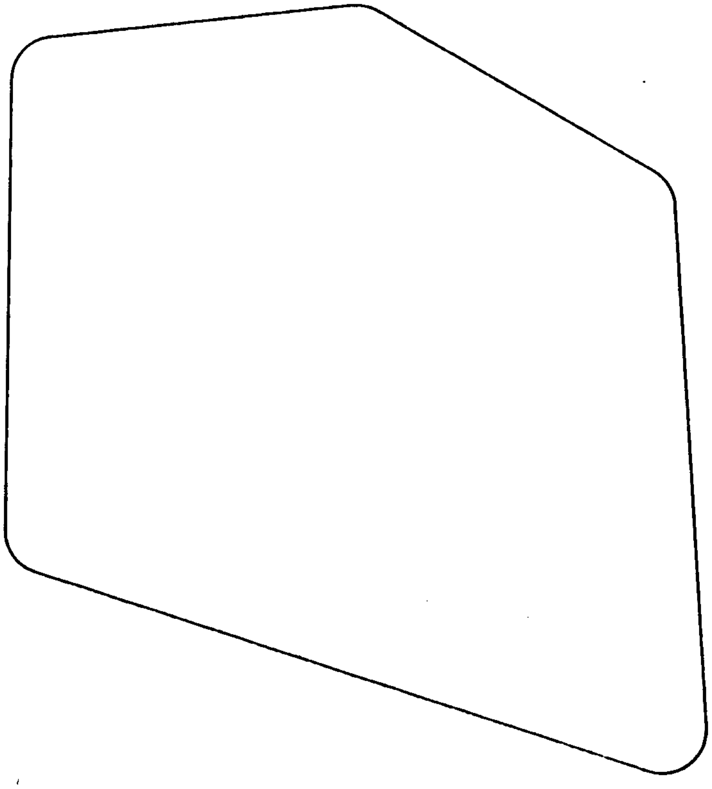
## FLOWCHART OF STORAGE AND RETRIVAL PROGRAMS

FLOWCHART OF SAVE/LOAD PROGRAM PROCESS

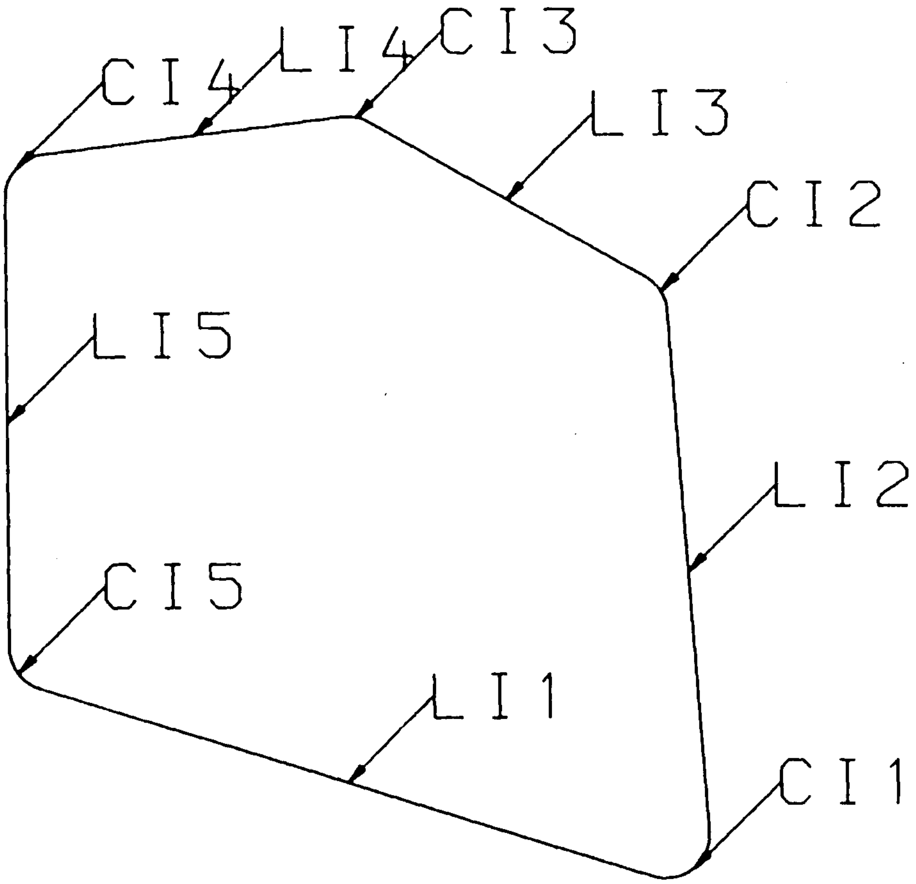

USER PLACES APPLICON SYSTEM IN RSX MODE
USER TYPES SAVE OR LOAD COMMAND
RSX-11M INTERCEPTS COMMAND AND ACTIVATES APPROPRIATE PROGRAM
SAVE/LOAD PROGRAM PLACES HEURISTICS IN COMMAND MODE AND ISSUES COMMANDS TO PREPARE FOR SAVE/LOAD
SAVE/LOAD PROGRAM OPENS NECESSARY FILES AND ISSUES COMMANDS TO HEURISTICS 7000 TO PREPARE FOR SAVE/LOAD
DATA TRANSFER OCCURS BETWEEN HOST DISK AND HEURISTICS 7000
UPON COMPLETION OF DATA TRANSFER SAVE/LOAD PROGRAM PLACES HEURISTICS 7000 IN TERMINAL MODE
USER PLACES APPLICON SYSTEM IN AGS MODE

Appendix 6

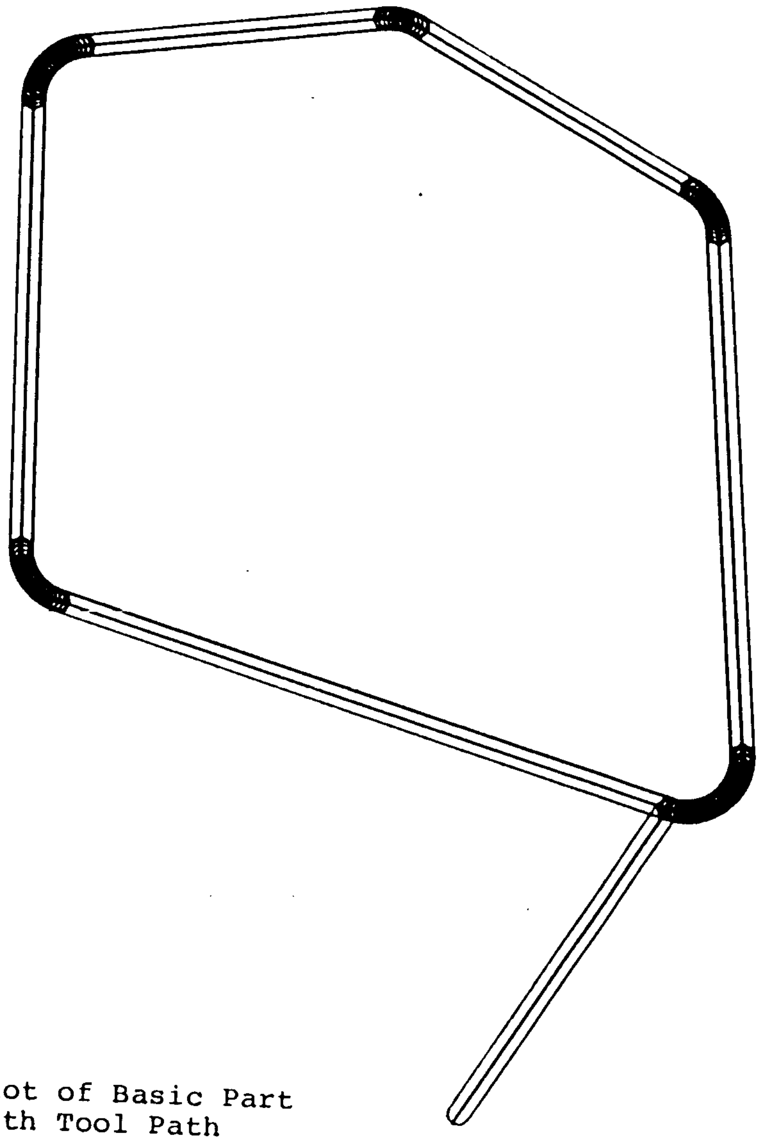
## EXAMPLE OF NUMERICAL CONTROL PART



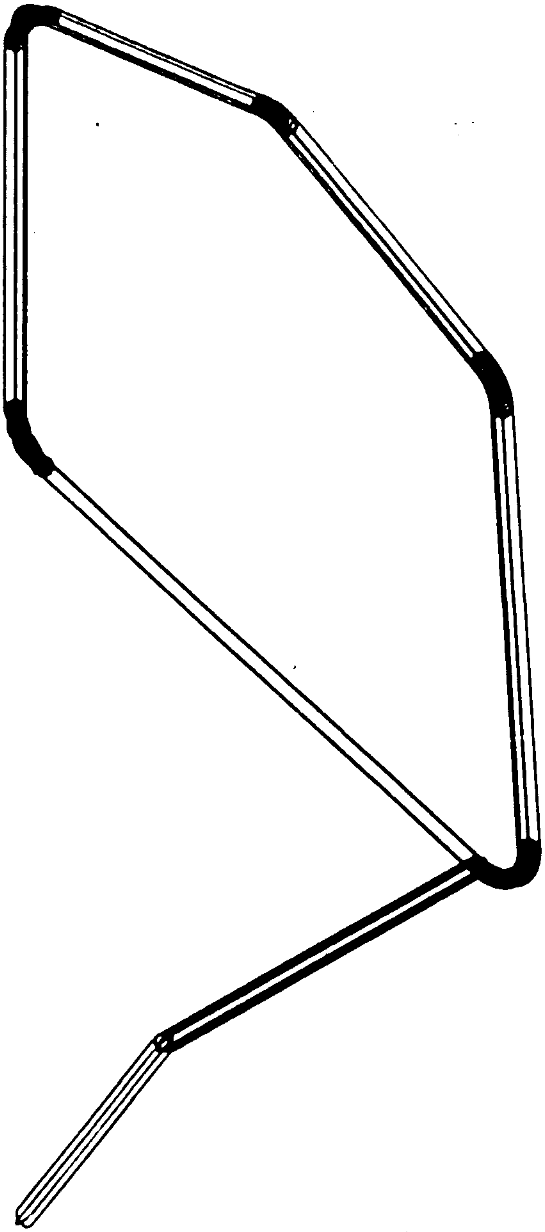
Plot of Basic Part



Plot of Basic Part Labled



Plot of Basic Part
With Tool Path



Isometric View of Basic Part With Tool Path

```
INOUT VER. 003.20-I

CLPRNT
LI1      =LINE/-4.6491,5.8904,0.0,2.2999,3.7188,0.0
CI1      =CIRCLE/2.449,4.196,0.0,0.5
LI2      =LINE/2.9477,4.2317,0.0,2.519,10.2339,0.0
CI2      =CIRCLE/2.0203,10.1983,0.0,0.5
LI3      =LINE/2.2683,10.6324,0.0,-0.8553,12.4173,0.0
CI3      =CIRCLE/-1.1034,11.9832,0.0,0.5
LI4      =LINE/-1.1654,12.4793,0.0,-4.562,12.0547,0.0
CI4      =CIRCLE/-4.5,11.5586,0.0,0.5
LI5      =LINE/-5.0,11.5586,0.0,-5.0,6.3676,0.0
CI5      =CIRCLE/-4.5,6.3676,0.0,0.5
REMARK START OF CUT SEQUENCE 300
         CUTTER/0.25,0.0,0.125,0.0,0.0,0.0,4.5
         COOLNT/ON
         SPINDL/2500
         FEDRAT/10.0
         INTOL/0.0005
         OUTTOL/0.0005
         TLAXIS/0.0,0.0,1.0
           FROM/0.0,0.0,5.0
           RAPID
           GOTO/2.2626,3.5995,5.0
           INDIRV/-0.010152,-0.0161422,0.9998182
         THICK/0.0,0.0
           DNTCUT
           GOTO/-4.6864,5.7711,0.0
           GO/ON,(LINE/(POINT/-4.6864,5.7711,0.0),PERPTO,(LINE/2.2626,$
             3.5995,0.0,-4.6864,5.7711,0.0)),TO,(PLANE/0.0,0.0,1.0,0.0)$
             ,TO,CI5
           CUT
           INDIRV/-0.9544822,0.298268,0.0
           TLLFT, GOFWD/CI5,TANTO,LI5
           GOFWD/LI5,TANTO,CI4
           GOFWD/CI4,TANTO,LI4
           GOFWD/LI4,TANTO,CI3
```

```
GOFWD/CI3,TANTO,LI3
GOFWD/LI3,TANTO,CI2
GOFWD/CI2,TANTO,LI2
GOFWD/LI2,TANTO,CI1
GOFWD/CI1,TANTO,LI1
RAPID
GOTO/2.2626,3.5995,5.0
INI
```

```
NUMERICAL CONTROL INFORMATION FOR DRAWING NCDEMO VERSION 70
MACHINE OPERATION    : PROFILING
ACTIVE TOOL # 5
 CUTTER DIAMETER     : 0.2500"
 CORNER RADIUS       : 0.0000"
                                   TOOL GAGE LENGTH    : 4.5000"
 X OFFSET CORNER RAD: 0.1250"
                                   Y OFFSET CORNER RAD: 0.0000"
 HORIZONTAL ANGLE    : 0.00000
                                   VERTICAL ANGLE      : 0.00000
ACTIVE FEEDRATE TYPE:  FINISH
              ROUGH : 10.0000"
              FINISH: 10.0000"
SPINDLE SPEED        : 2500 RPM
COOLANT              : ON
INSIDE TOLERANCE     : 0.0005"
OUTSIDE TOLERANCE    : 0.0005"
TOOL AXIS            : 0.0000000 0.0000000 1.0000000
TOOL ENTRY           : PLUNGE
TOOL EXIT            : PLUNGE
ENTRY POSITION       : ON CLEARANCE PLANE
EXIT  POSITION       : ON CLEARANCE PLANE
TOOL SET POINT       : 0.0000" 0.0000" 5.0000"
ENTRY POINT          : 0.0000" 0.0000" 5.0000"
EXIT POINT           : 0.0000" 0.0000" 5.0000"
WORKING PLANE        : 0.0000000 0.0000000 1.0000000 0.0000"
RETRACTION PLANE     : 0.0000000 0.0000000 1.0000000 2.0000"
CLEARANCE PLANE      : 0.0000000 0.0000000 1.0000000 5.0000"
DRIVE  SURFACE THICKNESS         : 0.0000"
CHECK  SURFACE THICKNESS         : 0.0000"
PROFILE/POCKET CUTTER STEP-OVER: 0.5000"
PROFILE/POCKET MAX CUT COUNT     : 1
```

```
MULTIPLE CUT TYPE                  : SIDE
NUMBER OF CHECK SURFACES           : 2
MAX SCALLOP HEIGHT                 : 0.0000"
POINT TO POINT DEFINITIONS -
 ADD TOOL PATH VERTEX AND CELL AT: SELECTED VERTEX
 POINT TO POINT PLANE              : ON RETRACTION PLANE
 CELL NAME                         : NONE
 CELL ROTATION                     : 2.00000
ACTIVE TOOLPATH SEQUENCE # : 400
  NEXT TOOLPATH SEQUENCE # : 400

APT LABEL INFORMATION
LABELS WILL BE USED SOMETIMES
NEXT LABEL FOR EACH TYPE:
CIRCLE : CI6
LINE   : LI6
ELLIPSE: EL1
POINT  : PO1
SPLINE : SP1
RULED SURFACE         : RU1
WARPED SURFACE        : WA1
PLANE                 : PLN1
SURFACE OF REVOLUTION: RE1
```

## Appendix 7

## LIST OF COMMANDS APPLIDED TO VOICE [19]

| Command | Definition |
|---|---|
| GO | Repaint the screen |
| SE | Select everything |
| DD | Delete selected geometry |
| GOSH | Update the screen |
| KILL | Disregard the last command |
| UNDR | Take away the selected geometry from the screen |
| RE | Update the screen |
| GRID | Turn the grid on or off |
| ADDV | Add a vertex |
| KI | Abort the sequence |
| EDT | Access all levels |
| UEDT | Do not access any levels |
| EDT 1 | Access level 1 |
| UEDT 1 | Do not access level 1 |
| EDT 7 9 | Access levels 7 and 9 |
| UEDT 7 9 | Do not access levels 7 and 9 |
| EDT 12 | Access level 12 |
| UEDT 12 | Do not access level 12 |
| KILL 0 | Disregard all previous commands since last repaint |
| @C2 | Multifillet |

| Command | Definition |
|---|---|
| GVUE 7 | Show view number 7 |
| SGRID | Display secondary grid |
| UU | Unselect the selected geometry |
| CVUE | Center the view |
| TEAC | Enter teach mode |
| USE P1 1 | Use component P1 on level 1 |
| 0 | Numerical 0 |
| Use LBL 7 | Use component LBL on level 7 |
| CDEF PTX | Change the definition to PTX |
| HELP | Reload the drawing |
| STOR @ | Store the drawing under the previous file name |
| SHOW F | Show the front view |
| SHOW R | Show the right view |
| INFO N | Display N C information |
| APTO G | Output APT geometry statements |
| APTO B | Output APT motion and geometry statements |
| APTO C | Output the cutter location file |
| SIN | Turn the sign on or off |
| SIN N | Turn took information on or off |
| CTRL | Enter control mode |
| LOGO | Logout |
| SH ISO | Show isometric view |

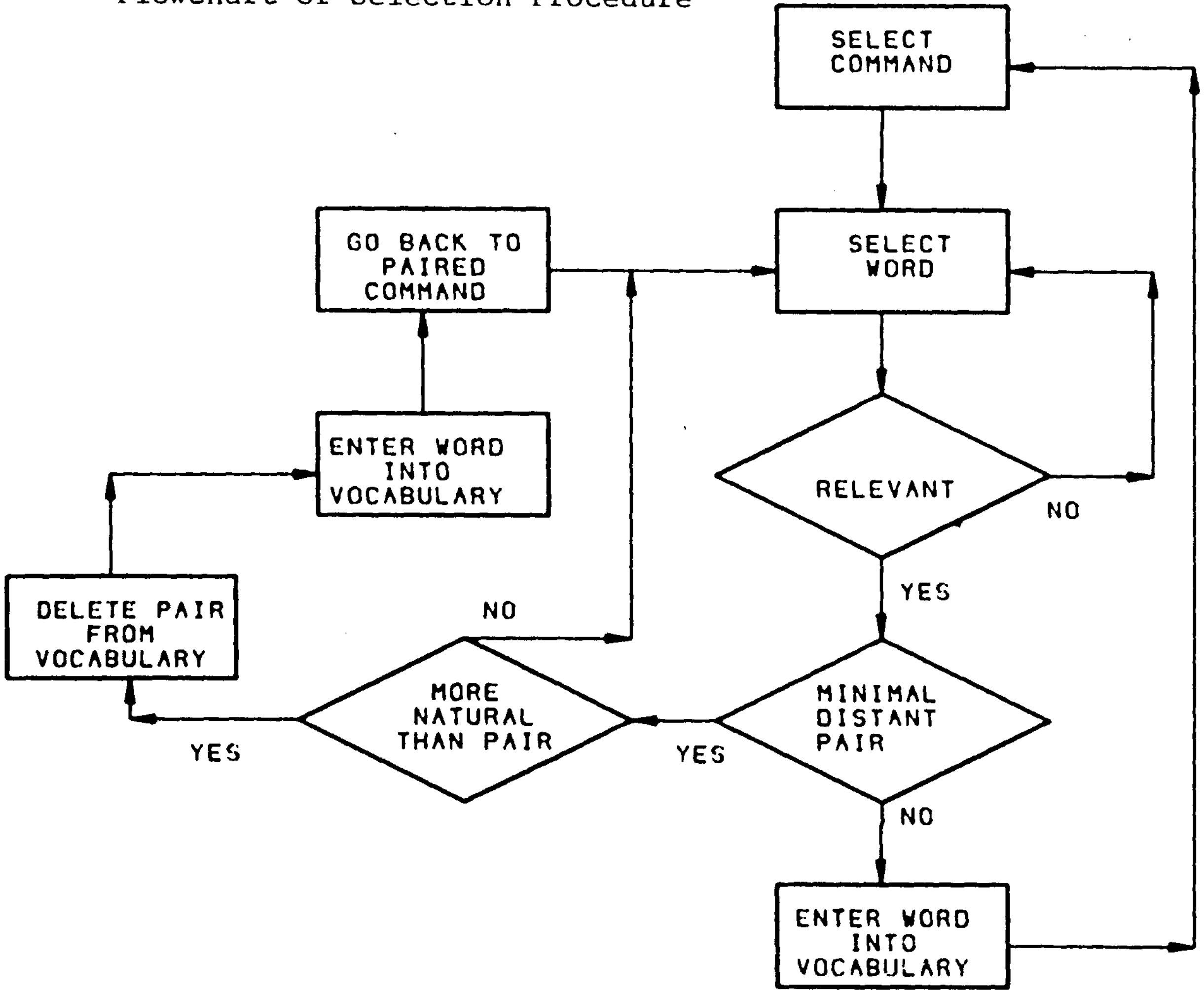
## Appendix 8

## LIST OF PREFERRED WORDS AND FLOWCHART OF SELECTION PROCEDURE

| Command | Word or Phrase |
|---|---|
| GO | Go |
| SE | Select Everything |
| DD | Delete |
| GG | Gosh |
| KILL | Kill |
| UNDR | Take Away |
| RE | Put Back |
| GRID | Grid |
| ADDV | Add Vertex |
| KI | Kill Sequence |
| EDT | Edit All |
| UEDT | Unedit Everything |
| EDT 1 | Access Geometry |
| UEDT 1 | Unedit Geometry |
| EDT 7 9 | On 7 and 9 |
| UEDT 7 9 | Off Lables |
| EDT 12 | On Level 12 |
| UEDT 12 | Off Tool Path |
| KILL 0 | Kill Everything |
| @C2 | Multifillet |

| Command | Word or Phrase |
|---|---|
| GVUE 7 | Get View 7 |
| SGRID | Secondary Grid |
| UU | Unselect |
| CVUE | Center View |
| TEAC | Teach |
| USE P1 1 | Use P1 Level 1 |
| 0 | Zero |
| USE LBL 7 | Use Lables |
| CDEF PTX | PTX |
| HELP | Help |
| STOR @ | Store |
| SH F | Show Front |
| SH R | Right |
| INFO N | N C Information |
| APTO G | APT Geometry |
| APTO B | APT Both |
| APTO C | Cutter Location |
| SIN | Sign |
| SIN N | N C Sign |
| CTRL | Control |
| LOGO | Logout |
| SH ISO | Isometric |

Flowchart of Selection Procedure


SELECT COMMAND
SELECT WORD
GO BACK TO PAIRED COMMAND
ENTER WORD INTO VOCABULARY
RELEVANT
NO
YES
DELETE PAIR FROM VOCABULARY
NO
MORE NATURAL THAN PAIR
YES
YES
MINIMAL DISTANT PAIR
NO
ENTER WORD INTO VOCABULARY

# REFERENCES

1. Flanagan, J. L., Speech Analysis, Synthesis and Perception, second edition., Spring - Verlag, New York, Heidelberg, Berlin, 1972, p 195.

2. Teja, E. R., "Voice Input and Output", EDN, Cahners, November 30, 1979.

3. Moshier, S. L., "Talker Independent Speech Recognition in Commercial Environments", Acoustical Society of America *50, June 1980.

4. Flanagan, J. L., Speech Analysis, Synthesis and Perception, second edition. Spring - Verlag, New York, Heidelberg, Berlin, 1972, pp 198 - 200.

5. Voice Input/Output, The State of the Art, Intergrated Computer Systems, Santa Monica, California, 1980.

6. "Heuristics 7000 Voice Controller", Heuristics, Inc., Sunnyvale, California, 1980.

7. "Preliminary Heuristics 7000 Voice Controller Users Manual", version 2.00, Heuristics Inc., Sunnyvale, California, October 15, 1980, p 4-10.

8. Welch, J. R., "Automatic Speech Recognition -- Putting it to work in Industry", Computer, Institute of Electrical and Electronics Engineers, Inc., New York, New York, May, 1980.

9. "Numerical Control Programming Package Option", AGS/883 Users Manual, version 2.4, Manual No. M-2748, Applicon Inc., Burlington, Massachusetts, July, 1977.

10. "NC - Start Userware Manual" AGS/1807 Document No. A-20616, Applicon Inc., Burlington, Massachusetts, September, 1980.

11. "Preliminary Heuristics 7000 Voice Controllers Users Manual", version 2.00, Heuristics Inc., Sunnyvale, California, October 15, 1980, p 1-2.

12. "The Random House Dictionary of The English Language, The Unabridged Edition, Random House, New York, New York, 1966.

13. Teja, E. R., "Voice Input and Output", EDN, Cahners, November 20, 1979.

14. "AGS/880 User's Guide" volume III, Applicon Inc., Burlington, Massachusetts, August 1979, Revision April 1981.

15. Brownlee, K. A. Statistical Theory and Methodology in Science and Engineering. second edition, John Wiley and Sons, Inc., New York, London, Sydney, 1965.

16. Walpole, R. E. and Myers, R. H. Probability and Statistics for Engineers and Scientists, second edition, MacMillan Publishing Co., Inc., New York, New York, 1972.

17. "Interactive Computer-Aided Design Considerations for Architects, Engineers, and Constructors", Calma Inc., Sunnyvale, California, June 1978.

18. "AGS/880 User's Guide" volume I, Applicon Inc., Burlington, Massachusetts, August 1979, Revised April 1981.

19. "AGS/880 - 1 Command Summary", Applicon Inc., Burlington, Massachusetts, 1980.

# VITA

The author was born in Tehran, Iran, on December 26, 1959, the son of Arthur H. and Betty J. Kemmerer.

He attended secondary school in Center Valley, Pennsylvania, and received his high school diploma from Southern Lehigh High School in June 1977. He continued his studies at Lehigh University, where he received the degree of Bachelor of Science in Industrial Engineering in June 1981. He began working towards the M.S. Degree in Industrial Engineering in the fall of 1981, having received a William C. Gotshall Memorial Fellowship to pursue this further study.