UNIDEX III
(VERSION 3A)

ADDENDUM TO
UNIDEX III MANUAL
REV. 0-1A

APRIL, 1985

AEROTECH, INC., 101 Zeta Drive, Pittsburgh, PA 15238
(412) 963-7470 TWX 710-795-3125

CONTENTS

INTRODUCTION

This addendum describes the additional programming features of Unidex-IIIA with Version-3A software. Wherever appropriate, reference is made to the Unidex-III Manual Rev.0-1A.

SECTION - 1

FEATURES OF UNIDEX-IIIA. (Refer Section 1-2 of Manual)

The additional programmable functions of Unidex-IIIA are as follows:

- Vector Feed Rate Mode (Linear Interpolation between two points.)
- Register operations and Register based moves
- Programs executable as subroutines
- Reset System on Interrupt
- Reset System by program statement
- Feed Hold on Interrupt
- Multiple inputs concurrently programmable as interrupt inputs
- Programmable message display
- Programmable message buffer
- Internal register & I/O status monitoring and display
- Interactive Data Input from keyboard
- Acceleration/Deceleration programming

SECTION - 2

GETTING ACQUAINTED WITH UNIDEX-IIIA (Chapter 2 of Manual)

2.1 POWER UP.

The system initialization at power up also checks the User Memory for any alterations since the previous EDIT session. A validity test is first carried out by verifying a prestored word (16 bytes long). Next, a checksum verification is done on the entire available User Memory. The check sum is stored in the first two bytes of the memory and the prestored word in addresses 3 through 18.

The self testing during power-up is done in the following order:

1. EPROMs: Checksum verification. A failure will cause the message " ROM ERR" to be displayed by the alphanumeric display. The operating system does not proceed further.

2. SYSTEM RAMs: Write and Read testing. The information in the RAMs is not destroyed during this test. Failure causes message "SRAM ERR" to be displayed.

3. USER MEMORY:
a) Write and Read testing without changing existing information. Failure causes message "URAM ERR".
b) Verification of prestored word. Failure causes message "URAM ALT". (Means User RAM altered). This message is expected when User Memory RAM chips have been powered down for any reason such as installation of new RAM chips or a battery failure.
c) Checksum verification. Failure may be due to errors written into the User Memory due to electrical noise.

If the User Memory was cleared prior to powering down Unidex-IIIA, the message "URAM CLR" is displayed at power-up if no failure occurs in the above three tests.

If however, one of the programs in the User Memory has been selected as a Boot-strap program to run at power up, Unidex-IIIA will execute the appropriate program if the self diagnostic tests are completed without failure.

If alternately, a program (say 47) has been selected as a set-up program, the display will show "RUN - 47" and pressing the [EXEC] key will start the execution of the program.

If there are no failures in self testing, the User Memory is not clear and if there are no Boot-strap or Set-up programs selected, the message " SYS RDY" is displayed.

2.2 FRONT PANEL RESET.

Performs Self diagnostic tests as above. However, front panel reset does not cause the Boot-strap program to run.

Resetting of Unidex-IIIA is disabled when editing a program in the User Memory. If the [RES] key is pressed when editing, the message "EDITING!" is displayed. Pressing the [EXEC] key will bring back the display prior to pressing the [RES] key.

2.3 DEFAULT STATES AT POWER UP.

The default states that Unidex-IIIA powers up in are as listed below:

- Local with remote enabled
- Incremental mode (G91) [alternates with G90]
- Non Corner-rounding mode (G24)[alternates with G23]
- Independent Feedrate Mode (G00)[alternates with G01]
- Normal Move Mode (G49) [alternates with G47 & G48]
- No Accel/Decel (G36) [alternates with G37,G38 & G39]

2.4 SELECTING A BOOT-STRAP PROGRAM

A program to be run as a Boot-strap program at power up is selected using the following Key sequence. Any Set-up program selected is cancelled.

Power up		
[EDIT]	"PGM -<1>"	
[SHIFT] [F]	"BOOT 37"	Program #37 is currently the Boot-strap program.
[2]	"BOOT 72"	
[5]	"BOOT 25"	
[EXEC]	"BOOT 25"	EDIT led is turned off. Program #25 is now the Boot-strap program.
Power down		
Power up	Executes Program #25.	

2.5 SELECTING A SET-UP PROGRAM

The key sequence is described below:

End of Program Execution	" READY "	
[EDIT]	"EDIT- 41"	Program #41 just executed.
[SHIFT] [G]	"SETUP 79"	Current setup program
[1] [9]	"SETUP 19"	
[EXEC]	"SETUP 19"	EDIT led turns off. Program #19 is now the Setup program. Boot_ Strap pgm. cancelled.
[RES]	"RUN - 19"	
[EXEC]	Executes Program #19.	

Boot-Strap and Set-up programs are mutually exclusive. Unidex-IIIA mav be programmed to either run a boot-strap program or have a set-up program but not both simultaneously.

2.6 DESELECTING BOOT_STRAP & SET-UP PROGRAMS

The Boot_strap and Set-up programs mav be deselected with the following key sequence.

[EDIT]	"EDIT- 76"	
[SHIFT] [M]	" NOBOOT "	
[EXEC]	" NOBOOT "	EDIT led turns off. Programs #25 & #19 deselected.
[RES]	" SYS RDY"	
Power down		
Power up	" SYS RDY"	

SECTION - 3

SYSTEM PROGRAMMING OF UNIDEX-IIIA (Section 4-3 of Manual)

3.1 REMOTE SELF DIAGNOSTIC TEST.

When operating Unidex-IIIA is in the remote mode of operation, either via the RS232/422 interface or the IEEE-488 interface, the self diagnostic tests described in Section-2 above may be performed and any failure information retrieved by the remote controller.

The character string shown below sent to Unidex-IIIA will initiate the self-test.

"V cr lf" Note: Unidex-IIIA is required to be addressed prior to sending this character string.

As described in Section-2, the test may be prematurely terminated due to a failure. At the end of the testing sequence, Unidex-IIIA will issue a SERVICE REQUEST to the controller. The controller is required to do SERIAL POLL of Unidex-IIIA to retrieve any failure information and enable further communication with Unidex-IIIA.

The status byte obtained as a result of serial polling after self-testing is described below:

	SET	CLEAR
Bit-7	Indicates failure	No failures
Bit-6	Always set. Setting Bit-6 initiates the SRQ	
Bit-5	Not Used	
Bit-4	Not Used	
Bit-3	User Memory Altered	User Memory unchanged
Bit-2	User Memory Write/Read Failure	User Memory Write/Read: No failure
Bit-1	System RAM Write/Read Failure	System RAM Write/Read: No failure
Bit-0	EPROM Checksum Failure	EPROM Checksum verified

Note: Unidex-IIIA will issue a SERVICE REQUEST when the Self-test terminates either due to a failure or upon completion without any failure. Controller must SERIAL POLL Unidex-IIIA to enable further communication with Unidex-IIIA.

With a User Memory of 32K bytes the self-test may take upto 4 seconds to complete making the SRQ necessary to indicate the terminaton of the test.

It is also important to note that a EPROM error may cause a bus hang up if the error in the EPROM is in the remote interface software. A bus hang up situation is detectable by the remote controller.

3.2 PRINTING REGISTERS X1-X4 & Y1-Y4

The content of a register such as X1 may be accessed by a computer controlling Unidex-IIIA in remote or may be printed on a printer connected to the RS-232 or RS-422 connector just as it is possible to do the same with the axis position registers.

When Unidex-IIIA is in the local mode, with a printer connected to either the RS-232 or the RS-422 connector, the register values may be printed as shown below.

KEY SEQUENCE	CHARACTERS PRINTED
[PRT] [X] [1] [EXC]	<9 spaces> X1:REG = 0000354000 <crlf>
[PRT] [Y] [4] [EXC]	<9 spaces> Y4:REG = -0000000250 <crlf>
[PRT] [X] [EXC]	<9 spaces> X:AXIS = 0456000789 <crlf>

In the Remote mode of operation, the controller must send the print command and be ready to read the data sent by Unidex-IIIA. When using the IEEE-488 bus it is necessary that Unidex-IIIA be addressed to talk after sending the print command.

FROM CONTROLLER	TO CONTROLLER
P X 1 <cr> <lf>	<sp> 0 0 0 0 3 5 4 0 0 0 <cr> <lf>
P Y 4 <cr> <lf>	- 0 0 0 0 0 0 0 2 5 0 <cr> <lf>
P X <cr> <lf>	<sp> 0 4 5 6 0 0 0 7 8 9 <cr> <lf>

3.3 PRINTING THE MESSAGE BUFFER

The characters in the Message Buffer may also be printed or accessed by a controller. Unidex-IIIA puts out the 32 characters in the message buffer followed by <cr> and <lf>. The characters in the message buffer may be any of the ASCII coded character set.

In the Local Mode, the key sequence [PRT] [M] causes the alphanumeric display 'PRNT MES'. Pressing the [EXC] key now will print out the contents of the message buffer.

KEY SEQUENCE	MESSAGE PRINTED (Examples)
[PRT] [M] [EXC]	<9 spaces><32 chars.of mess.buf.><crlf>

FROM CONTROLLER	TO CONTROLLER
P M <cr> <lf>	<32 characters of message buffer><cr><lf>

SECTION -4

MOTION PROGRAMMING OF UNIDEX-IIIA (Chapter 5 of Manual)

4.1 INDEPENDENT FEEDRATE/VECTOR FEEDRATE [G00/G01]

Unidex-IIIA powers up in the Independent Feedrate Mode. In this mode, the feed rates for the X and Y axes are entered separately following the axis move commands. An "F" command has to be preceeded by an axis move command.

```
EX:  G00 *  X3000 F600 Y4000 F800 *
```

In the Vector Feedrate mode, the Feedrate Command is an individual command and stands alone.

```
EX:  G01 * F1000  X3000 Y4000 *
```

The feedrate of 1000 steps/sec. will be vectored into an X-axis feedrate of 600 steps/sec. and a Y_axis feedrate of 800 steps/sec. in order that the move between the start and end points is in a straight line.

When changing from "G00" mode to "G01" mode, Unidex-IIIA expects a vector feedrate to be available either from an earlier vector move or from a new feed command. But when changing from "G01" to G00", the X and Y components of the feedrate from the most recent vector move will be effective unless new independent feedrates are entered.

```
EX: G00 * X100 F100 Y100 F100 * G01 * X200 Y300 *
```

Unidex-IIIA will display "-F-ERROR" after making the first move of X100 Y100.

```
EX: G01 * F1000 X3000 Y4000 * G00 * X100 Y100 *
```

The first vector move of X3000 Y4000 will be made with an X-feedrate of 600 and a Y-feedrate of 800. These feedrates will be effective during the second move of X100 Y100.

The accuracy of the component feedrates in a vector move depends on the length of the vector. A longer move makes for better accuracy. The best resolution of the clock output is 1 micro-second. The best results are achieved for feedrates of 250 steps/sec. and higher and when the axis move values are of the same order of magnitude and constitute a vector move of greater than 1000 steps.

4.2 UNIDEX-IIIA RESET [G99]

The command G99 resets Unidex-IIIA. All the initialization procedures are performed except the self diagnostic tests and execution of the Boot-strap program.

```
EX: G91 G00 * X100 F100 Y100 F100 * G99 *
```

Unidex-IIIA will make the move of X100 Y100 and when the move is complete, reset itself.

4.3 EXECUTION OF A PROGRAM AS SUBROUTINE [N < dd]

A program may be called to run as a subroutine of the currently running program. When the subroutine program ends, control is automatically returned to the main program. The subroutine programs may also be executed independently like any program. Subroutine programs are executed as independent subroutines: the following statuses of the main program are preserved:

1. Incremental mode / Absolute mode
2. Non-corner rounding mode / Corner rounding mode
3. Independent feedrate mode / Vector feedrate mode
4. Normal move mode / Register operations mode
5. The values of axis feedrates.

The command that initiates a subroutine program call is,

```
N < dd
```

where "dd" is the two digit program number to be executed as a subroutine.

Subroutines may be nested 16 levels deep. Unidex-IIIA maintains a stack space that can store upto 16 return addresses. All types of subroutine calls are included:

Program subroutines	-	N < dd
Independent Subroutines	-	N = ddddd
Ordinary Subroutines	-	N - ddddd

If a subroutine call is made when the stack is full, or if a return from a subroutine is executed when the stack is empty, Unidex-IIIA flags a Stack Overflow Error and displays the message "STK OVFL".

A program that is called as a subroutine must be entered into the User Memory before the main program that is the calling program. If down loading programs from a remote controller, the subroutine program must be downloaded before the calling program. Unidex-IIIA performs a precompilation when quitting EDIT mode whereby the jump addresses for the "N > dddd" command and the subroutine addresses for all the three types of subroutines are computed and stored within each program. If a subroutine is not available corresponding to a call, a Compile Error is generated. Unidex-IIIA then displays the message "C-ERR dd" where "dd" is the number of the program which generated the error.

4.4 PUSHING A LINE ADDRESS ONTO THE STACK [N + dddd]

The address of a line (number "dddd") may be put on the same stack used for storing subroutine return addresses.

```
   EX:    N + 2500 *     Stack the address of line number 2500
```

If the above command was the most recent stacking operation, then executing a "M99" command will unstack the address of line number 2500 and program will execute that line.

4.5 POP STACK AND CONTINUE [M89]

This command is used to decrement the stack pointer so that the next unstacking will result NOT in the most recently stacked address being unstacked but the one prior to that.

Use of this command may be convenient if say, we want to jump out of a subroutine without going through the normal procedure of a return (M99). Jumping out of a program subroutine is not possible.

Pushing and Popping the stack are very sensitive and tricky operations and great caution has to be exercised when incorporating these commands.

The sample programs in page 12 illustrates the use of the above commands.

Example:

```
                Program # 25

 1   G90 G01 * F1000 X100 Y200 *  ; Do a Move.
 2   G273 N+3000 *               ; If C-3 high stack line
                                 ; address 3000.
 3   N + 2000 *                  ; Else stack 2000
 4   N < 37 *                    ; Pgm # 37 as subroutine
 5   M99                         ; Unstack and jump to line
 6   N3000 *                     ; address 2000 or 3000
 7   M=167 X2000 Y5674 *         ; Do a move
 8   N - 300  *                  ; Subroutine call
 9   More Moves                  ; Sub. 300 returns to these
                                 ; Moves
10   N2000 X4563 Y1342 M=34 *    ; More moves
11   More Moves
12   M2 *                        ; End of main program #25

13   N300                        ; Subroutine 300
14   M=1 D200 M=0 D200           ; M functions
15   G271 M89 N>2000 *           ; Test input  C1 and if low
                                 ; skip to End-of-block. If
                                 ; high, pop stack and jump to
                                 ; line address 2000.
16   G272 N>300 *                ; If input C2 is high go back
                                 ; to line 300 (start of sub)
17   M99 *                       ; If low return from sub.


           Program # 37

  G91 G00 *                       ; Initialize modes
  G661 = 10 * N10                 ; Repeat loop counter
  X10 F10 M=8 Y10 F10 *           ; These moves trace a square
  X-10 * Y-10 M=0                 ; of side equal to ten steps
  G671 N>10 *                     ; End of repeat loop.
  M2 *                            ; End of Program # 37
```

In LINE 2 of the main program, either line address 2000 or line address 3000 is stacked based on the state of the input C-3. In LINE 5, after executing program #37, "M99" causes a jump to either line address 3000 or 2000.

So why not test C-3 input at LINE 5 and jump to wherever required. We do not care whether the state of input C-3 has been changed by program #37 or not. The destination of the jump has been decided prior to executing program #37.

4.6 REGISTER OPERATIONS [G45,G46,G47,G48,G49]

Unidex-IIIA has eight registers X1,X2,X3,X4, Y1,Y2,Y3 and Y4 that may be used by a motion program in addition to the X and Y position registers. These registers as well as the X and Y registers are used in register operations described in the following paragraphs.

Unidex-IIIA is in the Register Mode when handling registers. Normal mode is the default at power up and return to normal mode from register mode is effected by the modal command - "G49"

Simple register arithmetic may be performed using the plus, minus and equal-to signs. A register may be added to or subtracted from a number or a register to arrive at a value required for a specific operation. There are certain restrictions on the syntax when using these signs as described below.

4.7 ASSIGNING A VALUE TO A REGISTER [G47]

Any of the registers may be assigned a value not greater than +/- 2,000,000,000. This value may be a number or the value of an arithmetic expression including registers numbers and the "+" and "-" signs.

```
EX:  G47 * X1 = 2500 * X = 5000 Y = 3000 * G49 *
```

The above line assigns the value 2500 to register X1 and value 5000 to X position register and 3000 to Y position register.

```
     G47 * Y3 = X1 + X - Y -350 *
```

This line assigns the value of the expression to the right of the "=" sign (4150) to register Y3.

When using "G47" it is illegal to use a "+" or "-" sign on the left side of the "=" sign. Unidex-IIIA will flag an error and if in local mode, will display "ILL.CHAR"

4.8 REGISTER BASED MOVE [G48]

An X-axis move or a Y-axis move or a combined two axis move may be performed based on register values. In the incremental mode, the number of steps for the move is realized by evaluating an expression containing registers and numbers. In the absolute mode, the value of the expression is used as the destination position for the axis.

```
   EX: G91 G01 * F10000 * G48 * X = 5000 Y = X + 1000 *
```

The above line makes for a two axis move at a vector feedrate of 10000 steps/sec. X axis moves 5000 steps and Y axis moves 6000 steps.

```
   EX: G90 G00* G47 X3 =100 * G48 X=X3+10 F100 Y=200 F50 *
```

Register X3 is assigned a value of 100. X axis moves to position 110 at 100 steps/sec. and Y axis moves to position 200 at 50 steps/sec., in a two axis move.

4.9 REGISTER COMPARISON & CONDITIONAL SKIP [G45,G46]

Just as conditional skip to end-of-block is made based on the level of a C-input, or a repeat counter decrementing to zero, skip to end-of-block can be performed based on comparisons between registers and/or numbers.

Command "G45" allows a skip to end-of-block if the condition specified is TRUE.

Command "G46" allows a skip to end-of-block if the condition specified is FALSE.

The end-of-block automatically cancels the register comparison mode.

```
   EX:   G90 G00 * G45 X >= 3000 X3000 F1000 *
```

If the value of X position register is greater than or equal to 3000, skip to end-of-block,or else move X axis to position 3000 at 1000 steps/sec.

```
   EX:   G90 G00 * G46 X < 3000 G48 X=3000 F1000 *
```

This line performs exactly the same function.

```
   EX:   G45 X1 + 200 <> Y - X3 +500 M=1 N>20 * M=2 *
```

Here if (X1+200) is equal to (Y-X3+500), M output no.1 is turned on and program execution moves to line number 20. If the two expressions are not equal, output no.2 is turned on.

As can be seen in the examples above, multiple symbols may be concatenated to test for the following conditions:

1. Less Than	<
2. Greater Than	>
3. Equal to	=
4. Less Than or Equal to	<=
5. Greater Than or Equal to	>=
6. Not Equal to	<> or ><
7. Unconditional	<=> or <>= or =<>

Example:

The following is a program that calls a subroutine to do a rectangle 10 times. The size of the rectangle is passed to the subroutine in X1 and Y1 registers and the number of repeat loops to be performed is passed in register X2. This enables different rectangles to be described and the repeat loop to be executed different number of times using the same subroutine.

```
G90 G01 G24 *                     ;Initialize
G47 X1=100 Y1=50 *                ;Rectangle size
X2 = 10                           ;Loop count
N=200 *                           ;Subroutine call
...
...
M2 *                              ;End of program


N200 * G00 *                      ; Subroutine begins
N40 G48 *                         ; Reg.based moves. Loop begins
X=X1 F100 * Y=Y1 F50 *            ; Two sides of rectangle
X=-X1 * Y=-Y1 *                   ; Two more sides
G47 X2 = X2-1 *                   ; Decrement counter
G45 X2 = 0 N>40 *                 ; If counter zero quit loop
M99 *                             ; Return
```

It can be seen that the repeat loop in the subroutine was realized using a register operation rather than the usual G661 = 10 and G671 combination. The count of 10 may be passed to the subroutine via repeat counter by statement G661 = 10 prior to calling the subroutine. The difference is that G671 always decrements by ONE and quits the loop only when ZERO. The Register operation provides for more flexibility.

4.10 PROGRAMMBLE FEED-HOLD INTERRUPT [G201 - G204] [G211 - G214]

The C-inputs may be programmed to interrupt and halt program execution and stop clock output from the axis indexers on a high to low or a low to high transition on the input programmed. A transition in the opposite direction releases the hold and allows clock output and resumes program execution.

G201 arms input C-1 to interrupt on a -VE (high to low) transition. Similarly G202, G203 and G204 arm C-2,C-3 and C-4.

G211, G212, G213 and G214 arm the respective inputs to interrupt on the +VE (low to high) transition.

EX: G201 * X1000 F1000 M-27 D1000 Y2000 F2000 *

Input C-1 is armed to interrupt on the high to low (-ve) transition which may occur at any time. If the interrupt occurs during the dwell, the duration of the dwell will be extended by the feed hold. The amount of extention will be time C-1 input remains low.

A command arming an input to interrupt invalidates all previous interrupt arming commands relating to that input.

More than one input may be concurrently programmed to interrupt. In such a case, the inputs behave in a manner similar to switches connected in series. Any one input may halt the program. All inputs must release hold before program can resume.

EX: G211 G212 * X100000 F10000 Y200000 F10000 *

The following sequence will hold true:

C-1 goes high	-	Indexer clock outputs stopped
C-2 goes high	-	No change
C-1 goes low	-	No change
C-2 goes low	-	Clock outputs resume.

4.11 PROGRAMMABLE RESET INTERRUPT [G401 - G404] [G411 - G414]

The same C-1,C-2,C-3 and C-4 inputs may be programmed to interrupt and reset Unidex-IIIA. The reset effected is the same as the 'Clear Device' operaion from a remote controller or the reset executed by the program statement "G99". System initialization is performed without the self test function. Boot-strap program does not execute.

G401 through G404 arm inputs C-1 through C-4 to interrupt on the -VE transition. G411 - G414 causes interrupts on the +VE transition.

4.12 PROGRAMMABLE ABORT INTERRUPT [G301 - G304] [G311 - G314]

The interrupts can be programmed such that the following actions are taken:

1. Halt program execution
2. Stop clock outputs from indexer
3. Update position registers based on clock pulses actually put out by the indexer
4. Set a User Flag to indicate the occurrence of the interrupt
5. Skip to end-of-block and continue program execution

Example:

```
 1   G90 G01 G24 G49 * F1000 * ;Initilize & Feedrate
 2   G7 *                      ;Go Home
 3   G313                      ;Arm Input C-3 to interrupt
                               ;on +VE transition
 4   N200 X15000 Y25000 *      ;Move command
 5   G533 N>300 *              ;If Flag-3 is CLEAR,
                               ;end program.
 6   N=100 *                   ;If SET ,do subroutine
 7   N>200                     ;Go back and complete
                               ;interrupted move
 8   N300 M2 *                 ;End


 9   N100 * G91 G00 *          ;Subroutine begins
10   G503 *                    ;Clear Flag
11   G47 X1=X Y1=Y *           ;Store position of interrupt
12   G49 * X10 F10 Y10 F10     ;Moves
13   Moves
14   G90 G48 X=X1 Y=Y1         ;Get back to position of
                               ;interrupt
15   G49 * M99 *               ;Return
```

It is now possible to remember the X and Y positions when the interrupt occurs (LINE 11) and after executing a subroutine go back to the position of interrupt (LINE 14) and complete the move (LINE 7)

4.13 DISABLING INTERRUPTS [G321 - G325]

All the interrupt arming commands described in the preceding paragraphs give the programmed input the capability to interrupt at any time during program execution. To disable an input from interrupting, the following commands may be used:

```
G321                  - Disable C-1 input from interrupting
G322,G323,G324    - Disable C-2,C-3,C-4 inputs
G325          - Disable all four inputs from interrupting
```

4.14 PROGRAMMABLE HALT & SERVICE REQUEST [G25] [G25 = ddd]

The command "G25" causes the motion program to halt and display a message "G25 =   " if in local mode. To resume the motion program, [EXEC] key may be pressed once.

In the remote mode, Unidex-IIIA issues a SERVICE REQUEST to the remote controller. The controller is required to do a SERIAL POLL. The status byte returned indicates that the SRQ was generated by a "G25" by virtue of Bit-1 being set. The remote controller may read the position registers and status bytes and resume the motion program by sending to Unidex-IIIA the character string: "B cr lf".

The command "G25 = ddd" assigns a number (up to 255) to the programmed HALT. In local mode the message display is "G25 =ddd". The number is to identify the HALT within a motion program. The program may be halted at different times in order to perform different operations external to the program.

When in the remote mode, the remote controller may access the G25 number by sending to Unidex-IIIA the string, "P N cr lf". Unidex-IIIA returns the number in binary form as a single byte followed by <crlf>.

If a plain "G25" comes after a "G25=ddd" in a motion program, the number "ddd" remains assigned to G25. The number changes only when a new number is assigned.

When in the G25 state, X and Y position and the status bytes may also be accessed by the controller.

Both in Local as well as Remote mode, a command to print the directory ("PD") or to print a program ("Pdd") will cancel the G25 state and inhibit continuation of the progam. Pressing the Mode Keys [AUT], [SGL], [EDT] or [IMD] will also cancel the G25 state.

A Manual Slew operation may be performed while in the G25 state. Quitting the SLEW mode by the key sequence [CE] [EXEC] will continue program execution with the position registers updated during the slew. However if any of the axes encounters a limit, the G25 state is cancelled.

In the Remote Mode, after doing a SERIAL POLL to exit the SRQ state a GO-TO-LOCAL command may be executed by the controller to put Unidex-IIIA in the Local with Remote Enabled state and give control to the keyboard. After a Manual Slew operation, the [RMT] key may be pressed to put Unidex-IIIA back into the Remote Enabled state. The controller may now send a second GO-TO-LOCAL command and then address Unidex-IIIA to resume the Remote mode of operation.

4.15 PROGRAMMABLE HALT & ENTRY INTO SLEW MODE [G78*,G79] [G78 = ddd*, G79 = ddd]

The command "G78*" halts the motion program and puts Unidex-IIIA in the SLEW mode. The keyboard or the joysick mav be used to slew the X and Y axes. When using the keyboard to slew, the key sequence

[CE] [EXEC]

may be pressed to exit the slew mode and resume the motion program. If instead the joystick is used to slew, and the front panel is inaccessible, the Joystick Button may be pressed to exit slew mode and resume program execution. Pressing the Joystick Button will not clear the position registers in this case (as in the normal Manual Slew which is initiated from the front panel). If a position register is to be cleared, that may be done by a "G92" command in the program immediately following the "G78*" or "G79" command.

Assigning a number to G78 makes for a message display "SLEW=ddd". The number "ddd" identifies the SLEW operation and may be any value up to "255".

The axis slew feedrates or the joystick slew rate division factor may be programmed along with the "G78" or the "G78=ddd" command. If these vaues are not programmed Unidex-IIIA assumes the previously programmed values (the default values if they were never programmed).

EX: G78 = 57 X F1000 Y F2000 *

Unidex-IIIA halts and displavs "SLEW= 57". Keyboard slew will move X axis at 1000 steps/sec. and Y axis at 2000 steps/sec. Joystick division factor is the previous value.

EX: G78 F - 10 *

Unidex-IIIA halts and displays "SLEW= 57". Joystick clock rates will now be divided by a factor of 10. Keyboard slew rates will remain at 1000 and 2000 steps/sec.

The "*" is essential after a "G78" to indicate to the program interpreter the extent of the command. Without it the interpreter looks for feedrates or jovstick division factor following the "G78". In cases where the "*" is inconvenient, "G79" or "G79=ddd" may be used. No slew feedrates or joysick division factor may be entered in the case of "G79" or "G79=ddd".

The X and Y position registers will be updated as a result of the slewing operation. When the motion program resumes, these new values will be effective.

4.16 PROGRAMMABLE MESSAGE DISPLAY ["mmmm"]

Arbitrary messages of up to eight characters in length or the contents of a register such as an axis position register or one of the eight general purpose registers X1--Y4 may be displayed on the alphanumeric display as a part of a motion program. Also displayable are the state of the eight M-OUTPUTS, the eight bits of the FLAG REGISTER and the four C-INPUTS. These binary values are represented by a string of "1"s and "0"s.

The legends on the keyboard have been modified to accommodate this new feature in the local mode of operation. All the possible ASCII characters are not available due to the small number of keys. However, all the numerals, the entire alphabet and a useful set of symbols are provided.

When editing a program in the local mode, pressing the keys [SHIFT] ["] does the following:

1. Enters the quotation mark (") on the display.
2. Starts the EDIT LED flashing on and off to indicate that we are now in the Message Entry Mode.
3. The keys on the front panel have new character assignment. These characters are printed on the right hand bottom corner of the upper and lower sections of the keycap. Where there are no such characters, the key retains the normal character assignment.
4. The [CE] and [SHIFT] keys do not change their function.

The key sequence [SHIFT] ["] will exit the Message Entry Mode and turn the EDIT LED continuously ON.

When editing a program from the keyboard, "STP" and "BST" will align the closing quote with the right most position of the alphanumeric display. The entire message including the quotes is treated as a single command for the edit functions: STEP, BACKSTEP and CLEAR-COMMAND. In the SEARCH mode, if the character string searched for is part of a message, the cursor (right most position of display) is aligned with the closing quote.

The [CE] key will put Unidex-IIIA in the Message Entry Mode if the character cleared is the closing quote. The EDIT LED will start to blink on and off. If the entire message is cleared including the opening quote, Unidex-IIIA exits the Message Entry Mode.

If for some reason, Unidex-IIIA is powered down while re-editing a message (in the Message Entry Mode), the program being edited will neither compile nor execute correctly because the closing quote is missing from the program. The program should be cleared from the user memory and re-entered.

4.16.1 DISPLAYING ALPHANUMERIC CHARACTERS

Any of the valid ASCII codes is permissible as a character. Some of the codes may not be displayable in the alphanumeric display provided in Unidex-IIIA.

The message to be displayed is enclosed within quotation marks. Unidex-IIIA completes execution of all moves up to the message and then displays the message. If there are more than eight characters within the quotes, only the first eight are displayed. If there are less than eight, they are displayed starting from the left end of the display. A maximum of 32 characters may be entered within the quotes. All characters after the 33rd merely replace the previous one and will eventually be replaced by the closing quote.

```
EX: G91 G00 "X-AXIS"X100F100 "Y-AXIS MOVE" Y100F100 *
```

Unidex-IIIA will display "X-AXIS " and make the X move of 100 steps. Then Unidex-IIIA will display "Y-AXIS M" and make the Y move.

Display updates by Unidex-IIIA such as " READY " or "G25 =126" will override any programmed messages. In order that a message may stay on display for a length of time it may be required to program a dwell after the message.

EX: "KARUMBA" D10000 *

The message stays on for 10 seconds before Unidex-IIIA displays " READY "

4.16.2 DISPLAYING REGISTER VALUES

The value contained in the axis position registers or any of the general purpose registers may also be displayed on the alphanumeric display. Since the display is limited to only 8 characters, the value to be displayed may be formatted to any number of digits from 1 to 10. These formatted numbers may form whole or part of a message to be displayed along with other register values and alphanumeric characters.

The "<" and ">" signs are used to enclose the name of the register the content of which is to form part of the message display. A "/" and a numeral following the register name formats the value to a length equal to the value of the numeral. A numeral "0" or the absence of a numeral after the "/" causes the default message length of 10 digits. For a negative register value, the minus sign takes up the first digit space.

If the X1 register has a value of 4567 and Y axis position register has a value -5500, the following examples demonstrate the effect of different message display commands.

MESSAGE COMMAND	DISPLAY
"<X1>"	00000045
"<X1/4>"	4567
"<X1/8>"	00004567
"X1=<X1/5>"	X1=04567
"<Y>"	-0000055
"Y = <Y/4>"	Y = -500
"Y =<Y/5>"	Y =-5500
"<Y/5><X1/3>"	-5500567

4.16.3 DISPLAYING M-OUTPUT,C-INPUT & FLAG REGISTER STATUS

The status of the conditional inputs, the M-function outputs or the flag register may be displayed as a message. These statuses are represented by a string of "1"s and "0"s.

The C-input status is 4 digits long and the M-output as well as the flag register status has 8 digits.

The M_function output displayed here is the status of the peripheral data register as read by the microprocessor. This register will contain the most recently programmed M output function in BINARY or BCD format. The actual voltage levels at the connector at the rear interface board depends on the loading of the M-output buffer/drivers.

MESSAGE COMMAND	DISPLAY
"<C>"	1010
"INP=<C>"	INP=1010
"<M>"	10011001
"<F>"	00110001

4.17 MESSAGE BUFFER

Unidex-IIIA features a 32 character buffer that may be written into, printed out or read by a controller. All ASCII coded characters are permitted to be written into the message buffer.

4.17.1 PROGRAMMED INPUT TO MESSAGE BUFFER [G63 "mmmm"]

The message buffer may be written into from a motion program. When a message command (characters enclosed in quotes as in the previous subsection) is preceeded by G63, the message is written into the message buffer instead of being displayed. The first 32 characters of the message generated by the message command are written into the message buffer. A message of less than 32 characters is written leaving the last few bytes of the message buffer as spaces.

The "G63" remains active only for the first message following it. "G63" turns on the Message Buffer Mode and only a message can cancel it. Other motion commands are allowed between the "G63" and the opening quotation.

COMMAND	MESSAGE BUFFER
G63 "X AXIS NOW AT <X>"	X AXIS NOW AT 0000004567
G63 "<X1>,<Y>,<X1/8>,1"	0000004567,-00005500,00004567,1
G63 "M-OUTPUTS = <M>"	M-OUTPUTS = 10011001

```
G63 X10F10 "" "MESSAGE"     Move X axis  10 steps.
                            Write all spaces into Message buffer.
                            Show  "MESSAGE " on display.
```

In order that the data entered into the message buffer be current, program execution upto the message command is completed before computing the register values and statuses to be written into the message buffer.

4.17.2 PROGRAMMED PRINTING OF THE MESSAGE BUFFER [G65] [G65=ddd]

The data in the Message Buffer may be printed via one of the serial communication ports (RS-232 or RS-422) when Unidex-IIIA is operating in the Local mode.

The command "G65" or "G65=ddd" puts out the 32 characters in the message buffer in the format shown below.

```
<9 spaces><32 chars. of mess. buff.><cr><lf>
```

The string of 9 spaces at the beginning is provided as a left margin for printing.

When operating in the Remote Mode, "G65" is identical to "G25". Unidex-IIIA issues a SERVICE REQUEST to the controller. The controller is required to perform a SERIAL POLL. The controller may then read an identification number (the three digits following "G65=") assigned to the Service Request, by sending to Unidex-IIIA the command "P N crlf". Refer to Section 4.14 for more details on "G25".

After decoding the identification number, the controller may decide to read the message buffer.

Reading the Message Buffer is accomplished by sending the command "P M cr lf" to Unidex-IIIA. Unidex-IIIA responds by sending to the controller, the 32 characters in the message buffer followed by <crlf>.

FROM CONTROLLER	TO CONTROLLER
P M <cr> <lf>	ABCDEFGHIJKLMNOPQRSTUVWXYZ123456<cr><lf>
P M <cr> <lf>	0004567 -005500 10011001 1101 <cr><lf>

4.18 DATA INPUT TO UNIDEX-IIIA [G62...]

It is possible to execute a programmed halt and enter data either into the message buffer or any one of the registers before resuming program execution. The keyboard is used to enter the data and restart the program.

The data may be alphanumeric characters such as a part number or a value resulting from a measurement in an automated inspection set up. These characters can be entered into the message buffer for later access by a controller or for printing in the local mode.

Alternately, the data entered may be a numeric value for one of the registers in Unidex-IIIA. This value may be used later in the program as a position coordinate or a loop counter or for a register comparison operation.

Data Input is initiated by the "G62..." command. The characters following "G62" specify the destination of the data entered. They also determine the range of characters that may be entered and also whether Unidex-IIIA displays a prompt to the operator. The following paragraphs describe the command syntax in detail.

4.18.1 INPUT FROM KEYBOARD TO MESSAGE BUFFER [G62D] [G62D<]

To enter alphanumeric data into the message buffer, the motion command "G62D" is used. Unidex-IIIA will halt the program and the display will prompt "ENTR MSG". The EDIT LED will flash on and off to indicate message entry mode. The keyboard character assignment changes as described in Setion 4.16. As characters are entered, the display will scroll from right to left. The [CE] key may be used to edit the message. A maximum of 32 characters may be entered from the keyboard. From the 33rd character only the last character will be replaced and the display will not scroll.

Pressing [SHIFT] ["] keys will terminate the message entry mode. To resume program execution, the [EXC] key may now be pressed.

The message may be entered again before resuming the program by pressing [SHIFT]["]. Unidex-IIIA is now back in the message entry mode.

If it is desirable to display a customized prompt instead of the default prompt "ENTR MSG", the program may execute a message display command and then "G62D<". The "<" character prevents Unidex-IIIA from updating the display and the display retains the programmed message display.

The following example illustrates the above. Unidex-IIIA is in the Local Mode.

```
EX:  G91G00 X10F10 G62D G65 Y10F10 "VOLTAGE" G62D<*
     ^^^^^^ ^^^^^^ ^^^^ ^^^ ^^^^^^ ^^^^^^^^^ ^^^^^
       a      b      c   d    e        f       g
```

a. Initial set up
b. X axis move
c. Unidex-IIIA prompts "ENTR MSG" and waits for keyboard entry. The EDIT LED flashes ON & OFF

KEYBOARD ENTRY	DISPLAY	Comments
	ENTR MSG	EDLT LED flashing
[A]	A	
[SHIFT][B]	AB	
[CE]	A	
[CE]		Displav blank
[SHIFT][C]	C	
[R]	CR	
[SHIFT]["]	CR	EDIT LED turned off
[SHIFT]["]		Display blank & EDIT LED flashing
[L]	L	
[1]	L1	
[SHIFT]["]	L1	EDIT LED turned off
[EXC]	RUNNING	Program resumes

d. Unidex-IIIA prints out the message buffer
e. Y axis move
f. Message display. Unidex-IIIA shows "VOLTAGE ".
g. The display does not change but Unidex-IIIA goes into Message Entry Mode. The message may be entered as shown in Item c. above.

4.18.2 INPUT FROM KEYBOARD TO A REGISTER [G62...]

Numeric data input to a Unidex-IIIA register is similar to data entry into the message buffer. However the range of characters accepted by Unidex-IIIA is limited to the numerals and the "-" sign. The [CE] key deletes the most recent entry. The [EXC] key resumes program execution. The EDIT LED does NOT flash on and off.

The different commands and corresponding Unidex-IIIA prompts are listed below. As in the case of data input to the message buffer, the prompt can be suppressed by a "<" sign after the register name.

COMMAND	PROMPT	FUNCTION
G62X	ENTER X	X position register gets the numeric value entered.
G62Y	ENTER Y	Y position reg. gets value
G62X1	ENTER X1	X1 register gets value
..	..	..
G62Y4	ENTER Y4	Y4 register gets value
G62M	ENTER M	M-output register gets value
G62F	ENTER F	Flag Register gets value

The last 10 digits entered will be used to compute the value to be entered into the specified register if the register is X,Y or X1 thru Y4. In the case of M-output and the Flag register, the 8 least significant bits of the binary value of the entered number will be loaded into the register. The real maximum value that can be entered into these registers is 255. A negative number will be converted to its 2's complement binary equivalent. When entering a negative number the "-" sign should be the first character entered. Unidex-IIIA will not accept the minus sign after a numeral unless the numeral is cleared using the [CE] key.

SECTION - 5

HANDLING LIMITS

It is conceivable that Unidex-IIIA may get into a LIMIT situation by running into one of the axis limits. In such a case there are a number of protection features built into the system.

The hardware that drives the axes motors (the stepper translators, the servo amplifiers) prevents the motors from running further into a limit by inhibiting the clock pulses put out by Unidex-IIIA. Clock pulses in the opposite direction are allowed to pass through permitting a retreat from the limit.

Unidex-IIIA itself inhibits the indexers from putting out any more clock pulses when a limit is detected in either axis. Unidex-IIIA also updates the position registers with the number of clock pulses actually put out by the indexers. The alphanumeric display indicates which axis is in limit with either "-X-LIMIT" or "-Y-LIMIT".

5.1 LIMIT AT POWER UP

If one of the limits is active at power up Unidex-IIIA detects it after initialization and self diagnostic tests are completed. Unidex-IIIA will then default to the LOCAL mode and indicate the axis in limit. The procedure to exit from a limit is described in the following paragraphs.

5.2 LIMIT IN LOCAL MODE

When operating in the LOCAL Mode, the front panel keys may be used to back out of a limit. The [SLEW] key may be pressed to enter the manual slew mode and the appropriate "arrow" key used to slew out of a limit. Alternately one can execute an IMMEDIATE command or execute a program from the User Memory in the AUTO or SNGL Mode to come out of the limit.

It has to be noted that any move commanded by Unidex-IIIA to go further into the limit will be executed by Unidex-IIIA but will be inhibited by the subsequent hardware as described above. This will result in the axis position registers having incorrect info ion and therefore not matching the tracking displays.

It is also important to exp ct the limit switches to exhibit contact bounce and create a second limit situation when backing out of limit. The limit switches are not debounced because ignoring a noisy or a faulty limit switch compromises the protection features built into the system.

5.3 LIMIT IN REMOTE MODE OF OPERATION

When operating in the REMOTE mode, Unidex-IIIA may be set up to do one of the following when a limit is encountered.

1. Jumper 37 removed:

REMAIN IN REMOTE STATE and issue a SERVICE REQUEST to the controller. The controller has to SERIAL POLL Unidex-IIIA to continue communication with it.

The controller may then proceed to send appropriate commands to back out of the limit.

2. Jumper 37 installed.

GO TO "LOCAL with REMOTE ENABLED" state and issue a SERVICE REQUEST. The controller has to SERIAL POLL Unidex-IIIA. The status byte returned will reflect the Remote Enabled state. Unidex-IIIA may now be accessed from the keyboard or via one of the remote interfaces.

If a limit is encountered while in the Remote Enabled state, Unidex-IIIA goes to the LOCAL state. Thus if the limit switch has contact bounce and generates a -VE transition more than once, Unidex-IIIA will be in the LOCAL state and will not be accessible via the remote interfaces.

5.4 LIMIT DURING MANUAL SLEW OPERAION

If Unidex-IIIA detects a limit during a manual slew from the keyboard, the indexers are inhibited and a message indicating the axis in limit is displayed. The [SLEW] key has to be pressed to re-enter the manual slew mode and back out of limit.

If the Joystick is used for slewing, backing out of a limit may be accomplished without going to the front panel. Detecting a limit disables the joystick, but a change in the direction of any one of the axes (from the joystick) reenables it and allows the use of the joystick to back out from a limit. That is, if a limit is encountered when slewing X-axis in a CW direction, moving the joystick in the opposite direction automatically re-enables it and slews the X-axis out of the limit in the CCW direction.

5.5 LIMIT DURING PROGRAMMED SLEW (G78,G79)

If the slew operation is initiated by a "G78" or "G79" in a motion program, and a limit is encountered while slewing, it is possible to back out of the limit and resume program execution.

The message "-X-LIMIT" or "-Y-LIMIT" will be displayed when the limit is detected. The [SLEW] key must be pressed to re-enter the slew mode and the appropriate arrow key pressed to back out of limit and slew the axis to the desired position. The key sequence [CE] [EXEC] will then exit the slew mode and resume execution of the program.

If the joystick is being used, changing the direction re-enables it and after backing out of the limit, program execution may be resumed by pressing the joystick button.

If Unidex-IIIA is being operated in the REMOTE mode, a SERVICE REQUEST is issued when the limit is detected. The controller MUST do a SERIAL POLL to release control to to the keyboard and joystick so that one of the arrow keys or the joystick may be used to back out of limit. The program execution may be resumed as described above after the serial poll.

SECTION - 6

OEM PROGRAMMING CAPABILITIES OF UNIDEX-IIIA

For OEM customers, the RS232 Serial Comm. board (690D1268) is modified to include a jumper (the OEM jumper). The function of the jumper is described in the following paragraphs.

For Standard customers, it is required that the Switch No. 5 of SW1 on the Interface Board (rear panel) be permanently "ON".

The character length for serial communication for both the OEM customer as well as the standard customer is now,

```
7 bits  -  SW1 : Switch 6  -ON        or
8 bits  -  SW1 : Switch 6  -OFF
```

The options of 5 bits and 6 bits are not available.

6.1 FUNCTIONS OF THE OEM JUMPER

The status of the OEM jumper (IN or OUT) determines whether programs 50 thru 99 in the user memory may be accessed for editing and printing. The M-function outputs M:5 thru M:8 are not accessible to programs 1 thru 49 when the jumper is out.

When the OEM jumper is IN, Unidex-IIIA allows normal operation. All programs are accessible for editing or printing. Any M-fuction output may be controlled by any program.

When the OEM jumper is OUT, (or SW1 : switch 5 is in the "OFF" position), Unidex-IIIA will do the following.

1. Programs 50 thru 99 will not be accessible for EDITing or PRINTing. They may be run in the AUT or SGL modes or as a subroutine of another program. Unidex-IIIA will display the message " WHAT ? " and turn off all modes if attempt is made to access these programs. In the Remote mode, Unidex-IIIA will issue a SERVICE REQUEST to the controller instead of the message.

```
EX:    [EDT] [6] [5] [EXC]          " WHAT ? "
       [PRT] [0] [0] [EXC]          " WHAT ? "

Remote:    "P 87 cr lf"    Unidex-IIIA requests service.
```

The Directory may be printed out listing all programs.

2. Clear Memory operation will not be carried out.

   EX: [EDT] [CC] [0] [0] [EXC] " WHAT ? "

   Remote: "E $ 0 0 cr lf" Unidex-IIIA requests service.

3. Programs 1 thru 49 or the Immediate mode may only modify M-fuction-outputs M:1 thru M:4. The outputs M:5 thru M:8 are accessible only to programs 50 thru 99. These outputs (M:5 thru M:8) will remain unchanged unless a program numbered 50 thru 99 changes them.

   For an M-function command in a program numbered 1 thru 49, or in an Immediate mode operation, the four outputs M:5 thru M:8 will not be affected.

4. The program selected as the Boot-strap program with the OEM jumper IN (OEM Boot-Strap) will be executed before the Boot-strap program selected with the jumper OUT (USER Boot-Strap)

5. The OEM Boot-Strap program or the OEM Set-up program can be deselected only with the OEM jumper IN.

6. The USER Boot-Strap or Set-up program is deselected when the OEM Boot-Strap or Set-up program is deselected. They have to be re-selected after removing the jumper.

SECTION - 7

ACCELERATION/DECELERATION PROGRAMMING

Unidex-IIIA implements a form of programmable acceleration and deceleration. With this feature it is possible to ramp up the axis feedrates at the start of a move and ramp down at the end of the move, increasing the performance limit of the amplifiers/translators, motors and stages.

Acceleration and Deceleration are always simultaneously programmed. The feedrate ramp may be programmed to be linear or exponential. Also programmable is the TIME it takes to attain 100% programmed feedrate and the start-stop feedrate. In the Vector Feedrate mode, the acceleration rate may be programmed in STEPS/sec./sec.

After programming the ACCEL/DECEL paramaters, the program may switch between linear, exponential or no accel/decel as desired. The acceleration time and the start-stop feedrate default to specific values at power up.

It should be noted that ACCEL/DECEL is NOT IN EFFECT during HOME cycle or during SLEW operation.

7.1 ACCELERATION/DECELERATION IN OPERATION

Unidex-IIIA achieves acceleration and deceleration profiles by discreetly incrementing at the start and decrementing at the end of the move, the frequency of the indexer clock outputs.

The feedrate ramp (whether linear or exponential) is effected in a maximum of 24 discreet steps. The intermediate values of feedrates for a programmed feedrate value are computed using prestored set of coefficients (one set for linear and a second set for exponential accel/decel). From the programmed Acceleration Time or the Acceleration Rate, the time period between updating the feedrate from one value to the next (according to the required ramp) is derived. We will refer to this time as the sampling time.

Sampling time (Tsam) = Acceleration time / 23. (G00)
or = Vector feedrate/Accn. rate / 23 (G01 G38)

The divisor is 23 because 24 samples have only 23 time periods between them. The default value for sampling time is 8 milliseconds making the acceleration time 184 mS.

If the duration of the move as computed assuming no accel /decel, is longer than the acceleration time, a nice trapezoidal feedrate profile is obtained. If the duration is shorter, we do not attain 100% feedrate and the feedrate profile is triangular. In the latter case, to keep computations to a minimum, Unidex-IIIA defaults to linear accel/decel.

The detailed computations are as shown below. The definitions of different values are first given.

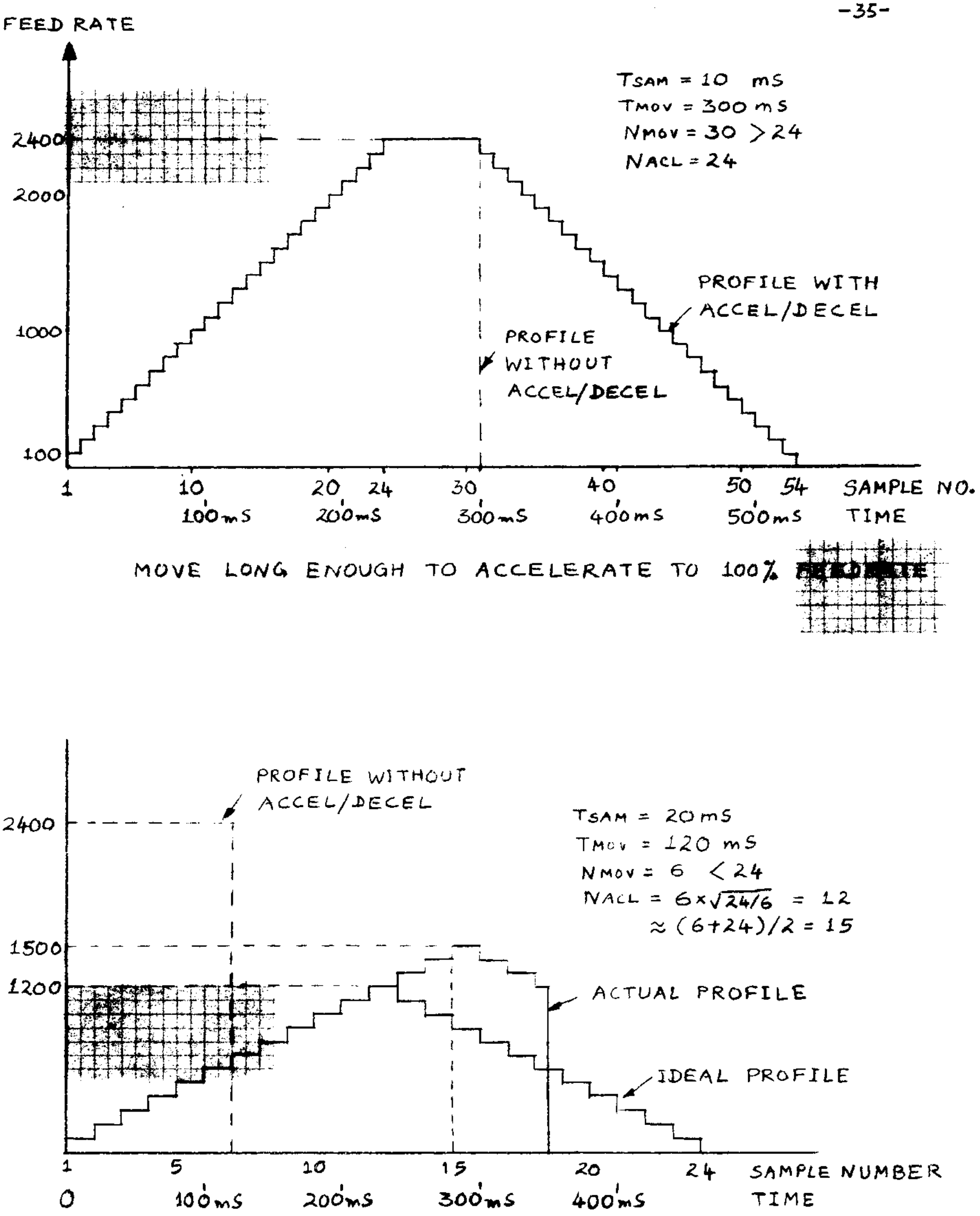
Tmov : Duration of move without accel/decel
Nmov : Number of samples in move computing from Tmov
Nacl : Number of samples in acceleration or deceleration phase of the move (max. 24)
Lmov : Length of move in steps as programmed
Tclk : Time period between clock pulses computed for the programmed feedrate (= 1 / Feedrate)
Fmax : Maximum feedrate achieved
Fdrt : Programmed feedrate

```
Tmov  =  (Lmov - 1) * Tclk
Nmov  =  Tmov / Tsam

Nacl  = 24   if   Nmov > 24
      = Nmov * SQRT(24/Nmov)   if   Nmov < 24
      Approximated in Unidex-IIIA by  Integer[(Nmov+24)/2]
      (This approximation results in a positive error that
       causes the deceleration phase to be delayed)

Fmax  = Fdrt * Nacl / 24  (Unidex-IIIA defaults to linear
                            accel/decel when Nacl < 24 )
```

The Figure 7.1 on Page-35 illustrate the above.

7.2 PARAMETERS OF ACCEL/DECEL

The relevant parameters that may be programmed in the Unidex-IIIA to obtain a desired acceleration/deceleration profile are described below. Once programmed these values remain effective until reprogrammed or Unidex-IIIA is reset or powered down. The programmed values remain unchanged at the end of program execution and carry over to the next



MOVE LONG ENOUGH TO ACCELERATE TO 100% FEEDRATE

MOVE NOT LONG ENOUGH TO ATTAIN 100% FEEDRATE

FIGURE 7.1

program. Therefore, it is a good practice to initialize all parameters to required values at the start of a program.

7.2.1 ACCELERATION TIME

This is the time taken by the axis to accelerate to 100% of the programmed feedrate and also the time taken to decelerate at the end of the move. The acceleration time is specified in milliseconds. The range of the value is between 23 and 4,000,000. Unidex-IIIA defaults to a value of 184 milliseconds (corresponding to a sampling time of 8 milliseconds). The minimum sampling time is 1 millisec. If the value computed from the programmed parameters is less, Unidex-IIIA substitutes the value of 1 millisecond.

The acceleration time may be separately specified for linear and exponential accel/decel.

7.2.2 ACCELERATION RATE

In the Vector feedrate mode of operation, the rate of acceleration in steps/sec./sec. may be specified. This rate applies to the Vector feedrate and therefore the individual components of the acceleration rate applied to each axis will be a lesser value.

The acceleration profile in this case is necessarily linear. After computing the component axis feedrates and the sampling time, the implementation of accel/decel is identical to the description above. The criterion for the lower limit of sampling time must be met when specifying an acceleration rate and programming a vector feedrate. A computed sampling time of less than 1 millisecond will be replaced by 1 millisecond.

The default value at power up is 250 steps/sec./sec. The range of the value is from 50 to 5,000,000.

7.2.3 START-STOP FREQUENCY [G76=ddd]

The start-stop frequency is a value of axis feedrate below which the indexer clock pulses may directly be applied to the servo system without exceeding the limits of performance of the system. This value may be programmed within a range of 31 to 166,500 steps/sec. Unidex-IIIA powers up with a default value of 30.52 steps/sec.

When accel/decel is active, if the feedrate programmed in the independent feedrate mode is less than the start-stop frequency, the related axis move is made without accel/decel. In the vector feedrate mode, the default value of 30.52 is always effective. This is done so that both the axes will move in a synchronized fashion regardless of their component feedrate values as long as the component values are greater than the default value.

7.3 LINEAR ACCEL/DECEL [G37] [G37=dddd]

Linear accel/decel mode is effected with the modal command "G37". All subsequent moves will be made with the accel/decel parameters in effect.

The acceleration time is programmed with the command "G37=dddd" where "dddd" specifies the time to attain 100% feedrate in milliseconds. The range of values and other constraints are described in paragraph 7.2.1.

A command such as "G37= 500" specifying the acceleration time does not activate the linear accel/decel mode. To do that the command "G37" is required.

Linear accel/decel may be activated either in the "G00" or the "G01" mode.

7.4 EXPONENTIAL ACCEL/DECEL [G39] [G39=dddd]

The syntax is identical to the Linear accel/decel commands. The range of values are also the same. The only difference is that the intermediate feedrate values for acceleration and deceleration phase of the move are computed using a set of coefficients different from that for the case of Linear accel/decel.

7.5 VECTOR ACCEL/DECEL [G38] [G38=dddd]

Vector Accel/Decel is a name given to the Linear accel/decel mode where the specified parameter is the acceleration rate instead of the acceleration time. The rate is programmed with a "G38=ddddd" command. A "G38" causes subsequent moves to be made with accel/decel. Both of these commands are effective only in the Vector feedrate mode ("G01").

A vector feedrate should be effective before specifying the acceleration rate so that Unidex-IIIA may compute the sampling time. If a "G38" command is executed in the "G00" mode, Unidex-IIIA defaults to Linear accel/decel with the currently effective value of linear acceleration time. If a "G38=dddd" command is executed in "G00" mode, Unidex-IIIA retains the value until required in a vector move or re-programed.

The programmed start-stop frequency has no effect on Vector accel/decel. In order that the two axes maintain their relative trajectories during the move, the computed feedrate components are applied to the axes. The minimum component feedrate in this case is the default start-stop frequency value.

7.6 NO ACCEL/DECEL [G36]

This is the power up default mode. A "G36" in the motion program will cancel any active Accel/Decel Modes. One of the commands "G37", "G38", or "G39" is required to initiate some form of Accel/Decel.

7.7 ACCEL/DECEL & CORNER ROUNDING

Unidex-IIIA treats the Corner Rounding feature independently from the Accel/Decel feature. When moves are programmed without accel/decel, corner rounding may be used in a D.C. drive system to "smooth out" a trajectory consisting of many point-to-point moves. The smoothing is achieved because a D.C. system has a positional lag during indexing that is made up after the indexer completes the clock pulses. Therefore if the next move is begun before the lag is made up, the two moves would appear to be a single continuous move. If the second move is at an angle to the first one, the resulting trajectory would have a rounded corner.

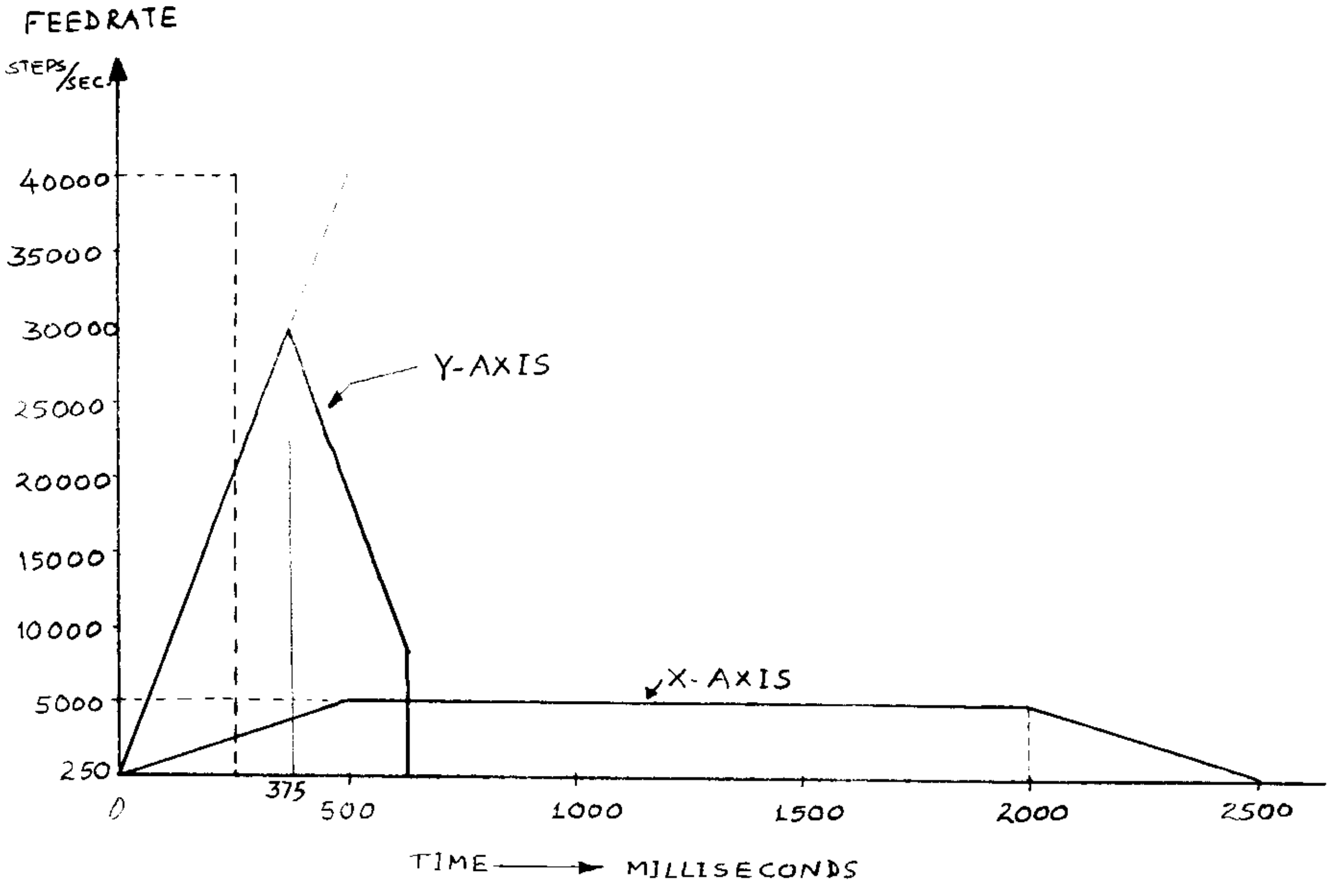
When Accel/Decel is active, the lag described above will be small due to the deceleration at the end of the move. Hence the corner rounding effect will be minimal.

7.8 EXAMPLES

```
1.  G91 G00 G76=250 G37=500 G37 X10000 F5000 Y10000 F40000 *
    ^^^ ^^^ ^^^^^^^ ^^^^^^^ ^^^ ^^^^^^^^^^^^ ^^^^^^^^^^^^^
     a   b     c       d     e        f            g
```

a. Incremental mode
b. Independent feedrate mode
c. Start-stop feedrate of 250 steps/sec.
d. Linear acceleration time of 500 milliseconds
e. Initiate Linear acceleration
f. X axis move of 10000 steps at 5000 steps/sec.
g. Y axis move of 10000 steps at 40000 steps/sec.

The feedrate profiles for the two axes are as shown in the diagram below.



```
2.  G91 G01 F10000 G38=20000 G38 X5000 Y10000 G36 X1000 *
        ^^^ ^^^^^^ ^^^^^^^^^ ^^^ ^^^^^^^^^^^^ ^^^ ^^^^^
         a    b         c      d       e        f    g
```

a. Vector Feedrate mode
b. Feedrate of 10000 steps/sec.
c. Acceleration rate of 20000 steps/sec./sec.
d. Initiate accel/decel (linear)
e. X and Y axis vector move with accel/decel
f. Cancel accel/decel
g. X axis only move.

```
3.  G91 G00 G76=500 G39 X10000 F10000 G37 Y5000 F8000
                    ^^^^^^^^^^^^^^^^^ ^^^^^^^^^^^^^^^
                            a               b
```

a. X axis move with expon. accel/decel. Acceleration time is default value or has been previously programmed.

b. Y axis move with linear accel/decel. Acceleration time as above.