7 Parametric Programming

Parametric programming uses functions that can be assigned to all the NC addresses in place of numerical values and that can be used as particular functions.

Functions used in parametric programming:
- Program L variables,
- External E parameters.

7.1 "L" Local variables

Definition

Variables are elements that can be substituted for numerical values to provide a programming aid. Program variables are defined by letter address L followed by a number from one to three digits.

L variables represent decimal numbers are should be used in order to keep the maximum precision on floating point operation (divide, square root, trigonometric etc.)

List of L Variables

- Variables L0 to L19,
- Variables L100 to L199,
- Variables L900 to L959.

Variables L0 to L19, L100 to L199, L900 to L959 have the same format and use but cause differences in programming (see chapter 7.2).

Variable Assignments

L Variables can be assigned to all the programmable NC addresses.

The assignment of an L variable to an NC address automatically results in the correct use of the unit for the programmed address.

Initialisation

The variables are initialised to zero:
- at power on,
- at the end of the program (M02),
- after a reset.

Application

The values assigned to L variables can be:
- integer or decimal numbers (maximum 8 digits plus sign),
- fixed values or values resulting from operations.

Use

L Variables can be used:
- to perform operations,
- for increments and decrements,
- for conditional branches with function G79 after comparison with an expression, use caution with equivalence tests.
- jointly with programming of external E parameters to make transfers.

7.1.1 Arithmetic Operations

Arithmetic operation	Symbol	Arithmetic operation	Symbol
Addition	+	Multiplication	*
Subtraction	-	Division	/

Division by zero is impossible and results in error 94.

7.1.2 Arithmetic Functions

Arithmetic function	Symbol	Arithmetic function	Symbol
Sine	S	Arc tangent	A
Cosine	C	Square root	R
Truncation	T		

Sine (S) and cosine (C)	The term following these functions is in degrees.
Truncation	Extraction of the integer value of the number following the symbol.
Arc tangent	The result of the operation is in thousandths of a degree.

Extraction of the square root of a negative number is impossible and results in error 94.

7.1.3 Logic Operations

Logic operation	Symbol	Logic operation	Symbol
AND	&	OR	!

AND and OR The operations are performed on values from which the decimal part is truncated (truncating performed automatically by the system) and which are expressed in binary.

7.1.4 Comparison Symbols Used with L Variables

Comparison symbol		Comparison symbol	
Equal to	=	Greater than or equal to	> =
Greater than	>	Less than or equal to	< =
Less than	<	Different from	< >

7.1.5 Conversion of the Internal Unit

In parametric expressions, functions U and M are used to convert values expressed in the internal system unit (see Chapters 2 and 3) to the programming unit:

- Function U is specific to linear axes
- Function M is specific to rotary axes.

Use of function U

Case of a system with micrometres (µm) as internal unit for linear axes.

Reminder: Programming in inches by G70 or metric system by G71.

- U254000 returns the value 254 for G71 (254 mm)
- U254000 returns the value 10 for G70 (10 inches).

Use of function M

Case of a system with 0.0001 degree as internal unit for rotary axes.

External parameter E74000 defines the reference position of axis 4 (B axis).

Conversion from 0.0001 degree (internal unit) to the programming unit in degrees. For any position on the B axis, the programming is:

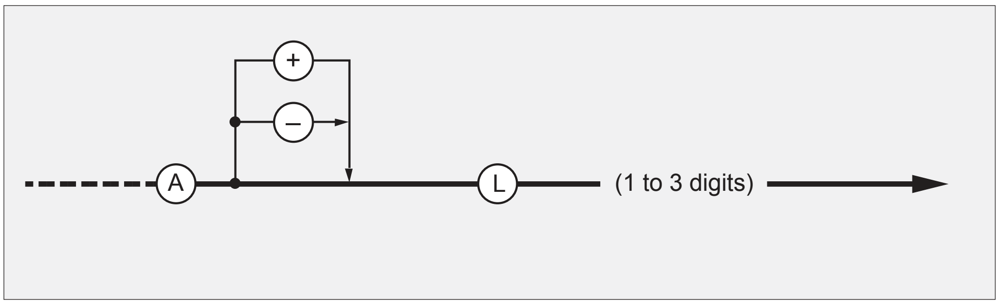
- L0=ME74000 or L0=[.IRX(8)]
- Variable L0 will hold the position on the B axis in degrees.

7.2 Programming Syntax for L Variables

Programming syntax for L variables is given as Conway diagrams and followed by programming examples.

7.2.1 Assignment of a Variable to an NC function

Syntax



A	NC function.
+	**/ -**Sign.
L	Variable used as numerical value.

Example

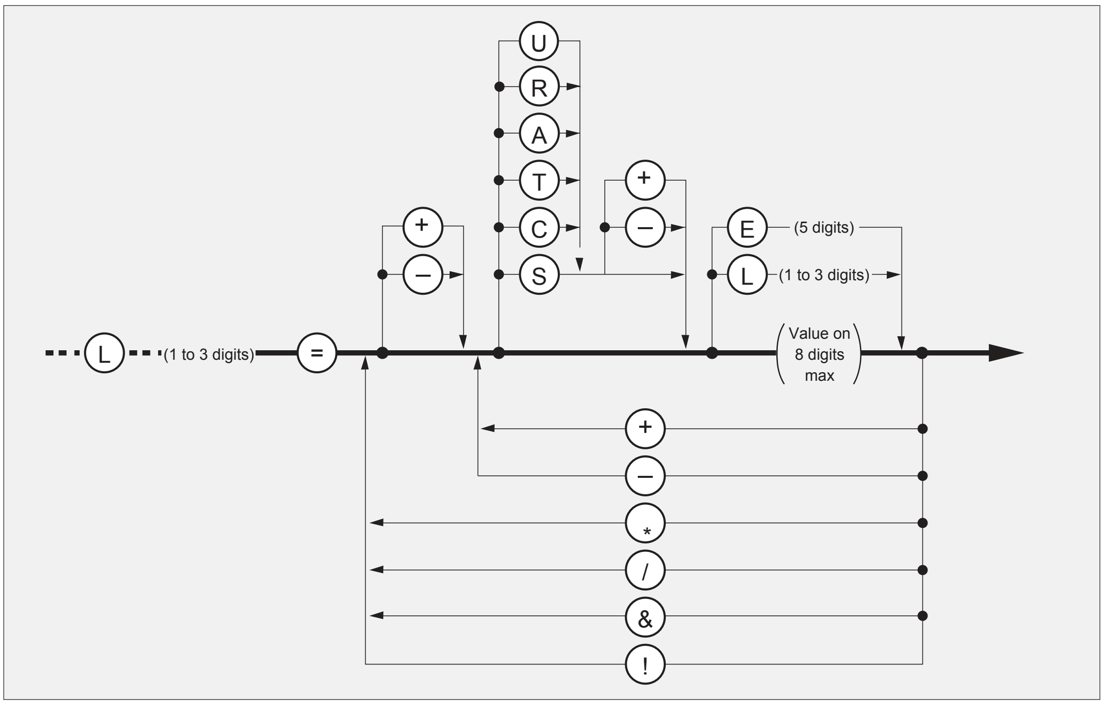
Use of a variable with NC addresses that have different units.

Assignment of variable L5 to addresses X and F.

```
N..
L5 = 18                         Declaration of the value of variable L5
N..
N.. G00 XL5                     L5 equals 18 mm with the X address
N..
N80 G94 G01 Z70 FL5             L5 equals a feed rate of 18 mm/min with the F address
N..
```

7.2.2 Declaration of a Variable in the Program

Syntax



The operations given in this diagram are detailed in the previous pages.

Notes

When the result of an operation giving a fractional number is assigned to an L variable, the system only keeps the first eight digits (integer and decimal parts) and ignores the rest. If the integer part of the result exceeds eight digits, the system reports an error.

Note on > = < Tests on L parameters

Be careful when making test on L variable. As an L variable is coded in Floating point the actual value can be slightly different from the displayed value. This is due to the principle of floating point calculation and cannot be considered as a default. The difference can be on the 10th or further digit after the decimal point. Therefore when making test on L values it is always better to check boundaries rather than exact value.

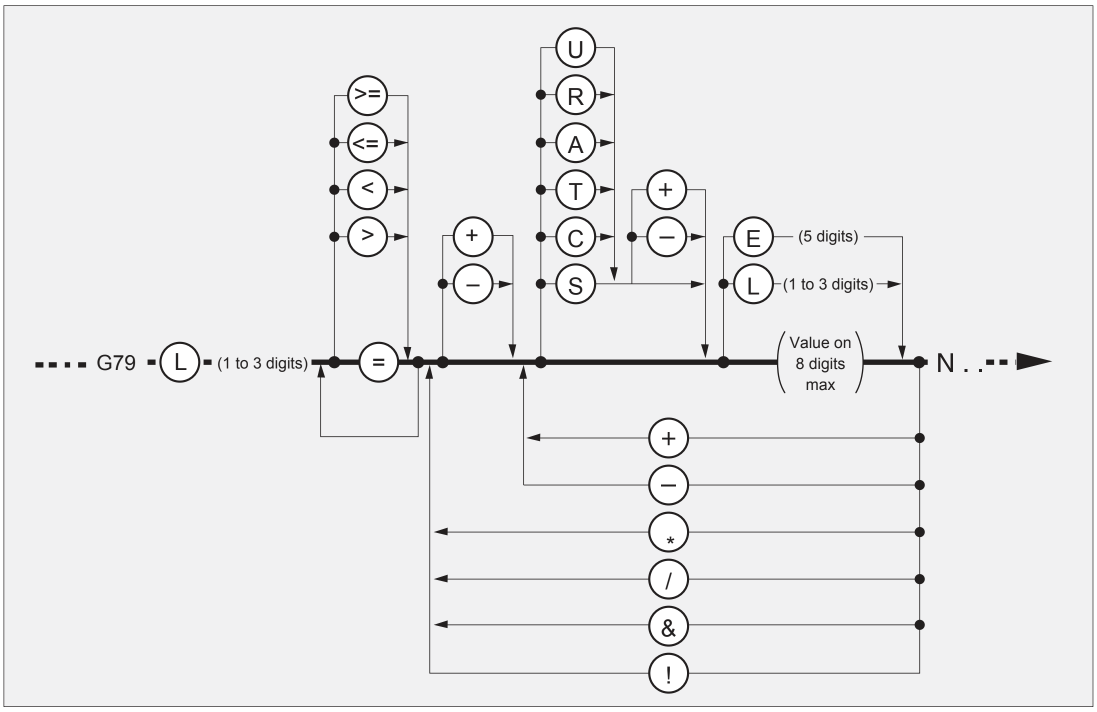
Example

Use of the variables with arithmetic operations.

`N..`	
`L1 = 5`	Declaration of the value of L1
`L2 = L1 + 5.3 * 3 * S30`	After the operation, L2 takes on the value 15.45 (sine 30° = 0.5)
`L3 = 100 / 3`	After the operation, L3 takes on the value 33.333333 (limitation to eight digits)
`N..`	
`N90 G00 XL2 Z30`	The value of L2 (15.45) is assigned to the X axis
`N100 XL3`	The value of L3 (truncated to 33.333 if the resolution is .001) is assigned to the X axis
`N..`	

7.2.3 Test of a Variable for a Conditional Branch

Syntax



Example

Use of a variable with a conditional test on the variable contents.

```
N..
N50 L1 = 0                          Initialisation of the variable with zero
N60 L1 = L1 + 1                     Increment of the variable
N..
N..
N200 G79 L1 < 10 N60                Condition: if L1 < 10, jump to N60; else continue
N210
N..
```

7.2.4 Notes on Programming Variables L100 to L199 and L900 to L959

Variables L100 to L199

Variables L100 to L199 have the same format and use as variables L0 to L19 but cause a different in programming.

Writing into a variable L100 to L199 suspends preparation of the block to which it belongs until execution of the previous block is completed. This is to prevent unexpected interaction if an external event (PLC generated subroutine or operator input in MDI for example) occurs that should generate a second block preparation sequence.

A block including a variable L100 to L199 cannot therefore be preceded by a block whose execution requires knowledge of the following block(s), e.g.:

- Profile Geometry Programming (PGP), or
- tool radius offset (G41, G42).

Programming of variables L100 to L199 with function M999

Writing of variables and transfer of current values into the part program are normally suspended until execution of the previous blocks is completed.

Programming M999 prohibits the operator or PLC from intervening in execution of a sequence of blocks, therefore allowing writing into these variables without stopping block preparation.

Variables L900 to L959

It is strongly not recommended to use variables L900 to L959 in a program with canned cycles (G81, G82, etc.), since they risk being overwritten when a cycle is called.

7.2.5 Equivalence of Variables L900 to L925

Variables L900 to L925 are equivalent to addresses A to Z respectively.

Example:

A = 250 is equivalent to L900 = 250

B = 1234 is equivalent to L901 = 1234 (and so forth to L925 for Z).

7.2.6 Symbolic Addressing of Variables L900 to L925 and L926 to L951

Variables L900 to L925 and L926 to L951 can be addressed by alphabetic symbols preceded by «'« (apostrophe). The variables can be used on the left or right side of an equation.

Variables L900 to L925

Variables L900 to L925 can be addressed by symbols 'A to 'Z respectively.

Example:

'C = 'A + 'B is equivalent to L902 = L900 + L901

Variables L926 to L951

Variables L926 to L951 can be addressed by symbols 'EA to 'EZ respectively
('EA = L926, 'EB = L927, etc. down to 'EZ = L951).

Example:

'A = 'B - 'EA/'EZ is equivalent to L900 = L901-L926/L951