9.2 PART PROGRAMMING FOR NC SYSTEMS

NC program

A program fo NC consists of a sequence of directions that causes an NC machine to accomplish a certain operation. The *NC program* describes the sequence of actions of the controlled NC machine. These actions include but are not limited to

- component movements, incl. direction, velocity and positioning;
- tool selection, tool change, tool offsets, and tool corner wear compensation;
- spindle rotation and spindle rotation speed, incl. possibility to change it to keep constant
- cutting speed for different diameters in turning;
- application of cutting fluids.

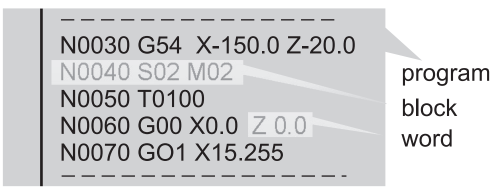
A *part program* is simply an NC program used to manufacture a part. *Part programming* for NC may be performed manually (*manual part programming*) or by the aid of a computer (*Computer-aided part programming*).

Many *programming languages* have been developed for part programming. The first that used English-like statements and one of the most popular languages is called APT (for *Automatically Programmed Tools*). Many variations of APT have been developed, including ADAPT (ADaptation of APT), EXAPT (a European flavor of APT), UNIAPT (APT controller for smaller computer systems), etc.

NC programming for complex parts are generated using advanced computer programs (CAD/CAM programs), which create automatically the machine code (so called *G-code*) in a graphic environment. Machine code is also largely used for manual part programming of simple shapes and is covered in the present section.

Machine code

The structure of a NC program written in machine code is standardized and for a two-axis NC system has the following format:



Structure of a NC program.

NC program block consists of a number of program words. The NC program is executed block by block: each next block is entered in the system and executed only after entirely completing the current block.

Each *program word* is an ordered set of characteristics, letters and numbers, to specify a single action of the machine tool. Program words fall into two categories,

1. *modal*, which are active in the block in which they are specified and remain active in the subsequent blocks until another program word overrides them;
2. *non-modal*, which are only active in the block in which they are specified.

Some of the most important program words are as follows

- ***sequence numbers*** (N****)
  Sequence numbers are a means of identifying program blocks. In some systems they are not required although sequence numbers are needed in most *canned cycles* (covered later in this section);
- ***preparatory functions (***also ***G-codes)*** (G**)
  Preparatory functions are used to set up the mode in which the rest of the operation is to be executed.
  Some of examples of G-codes are given in the table:

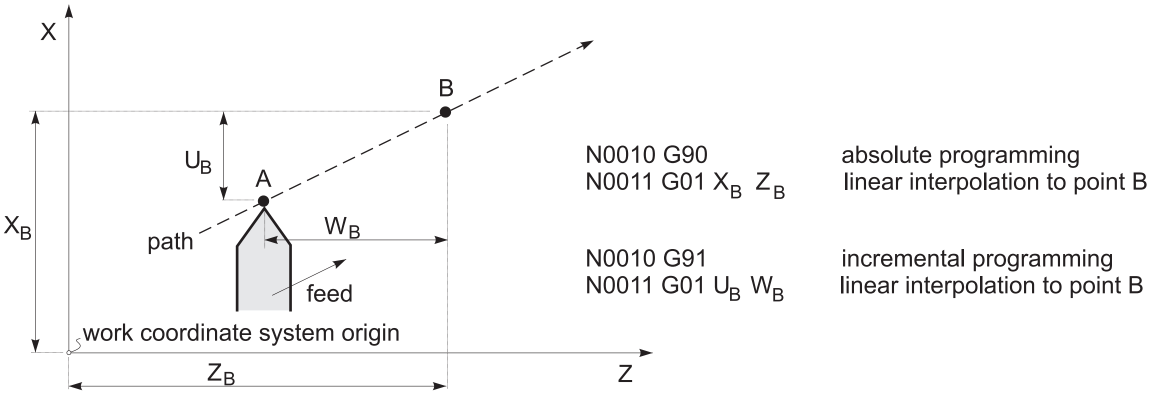
G00	Positioning (not cutting)
G01	Linear interpolation
G02	Clockwise circular interpolation
G03	Counterclockwise circular interpolation
G20	Inch data input
G21	Metric data input
G54	Workpart coordinate preset
G80	Canned cycle cancel
G81-89	Canned cycles
G90	Absolute programming
G91	Incremental programming

- ***dimension words*** (D****.***), where D stands for X, Z, U, or W
  Dimension words specify the coordinate positions of the programmed path. X and Z specify the *absolute coordinates*, and U and W specify the *incremental coordinates* (absolute and incremental programming are explained later in this section);
- ***arc center coordinates*** (D****.***), where D stands for I, or K
  Arc center coordinates specify the incremental coordinate position of the arc center (I in the direction of X-axis, and K in the direction of Z-axis), measured from the arc starting point;
- ***feed function*** (F**.**)
  Specifies the velocity of feed motion;
- ***spindle control function*** (S****)
  Specifies spindle rotational speed in revolutions per minute, or cutting velocity in meter per minute depending on the type of NC system and machine tool;
- ***tool calls*** (F**.**)
  The tool call word is used to access the required tool. It also gives the information for the radial compensation of tool corner wear for each new run of the program (and each new part);
- ***miscellaneous functions*** (M**)
  The M-function perform miscellaneous machine actions such as these listed in the table:

M00	Program stop
M02	Program end
M03	Start spindle CW
M04	Start spindle CCW
M05	Stop spindle
M06	Execute tool change
M07	Turn coolant on
M25	Open chuck
M26	Close chuck

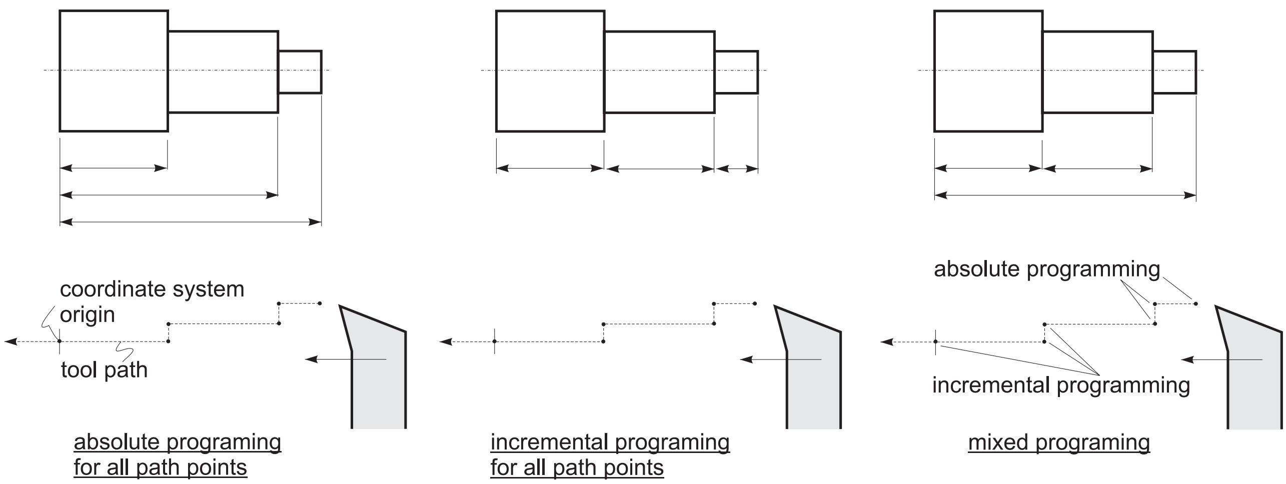
Absolute vs. incremental programming

Absolute and incremental programming specify the coordinates of points with respect to the work coordinate system (*absolute coordinates*), or from the point where the component is located (*incremental coordinates*):



Absolute (X and Z) and incremental (U and W) coordinates of point B, and sections of NC programs showing both types of programming.

Incremental positioning is also called a *point-to-point positioning* (do not mix with point-to-point NC systems). Both types of programming can be used for the whole program or just for certain sections of the program. Which kind of programming to apply generally depends on the type of dimensioning used in the part drawing. The next figure illustrates some examples of different dimensioning styles applied to one and the same part configuration, which suggest either absolute, or incremental, or mixed programming:



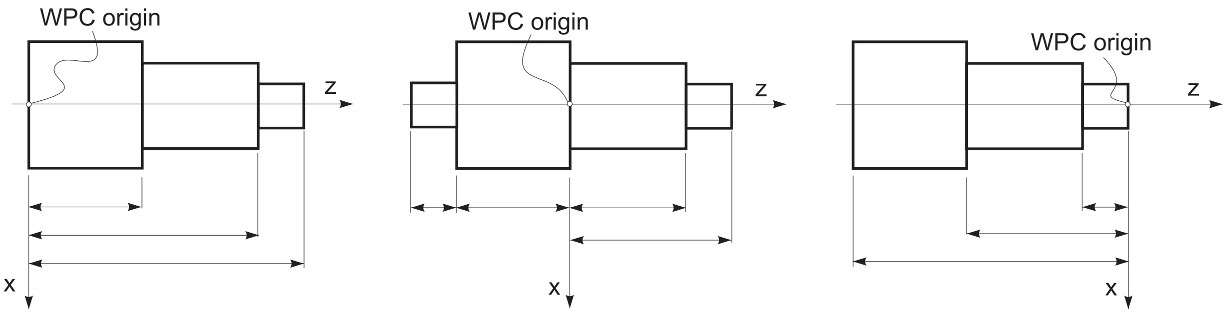
The type of part dimensioning defines the type of programming used.

Program points

The NC system must know where the part is positioned in the work space. The procedure for defining the *work coordinate system* (WPC) is called *workpiece coordinate setting*. Two important factors deal with workpiece coordinate setting,

❶ where the *part datum* (the origin of the WPC) is situated with respect to the workpiece;
❷ where the part datum is situated with respect to the machine tool.

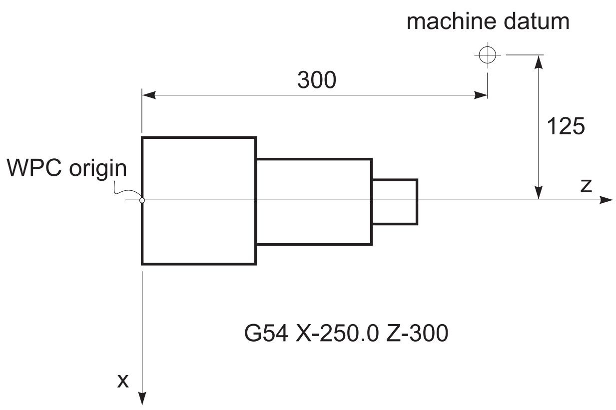
The WPC origin may be located at any part of the workpiece, but to avoid dimensional recalculations and respectively errors, the good programmers will chose the WPC origin at the point, from where the part features are dimensioned:



Selection of WPC origin.

The methods for locating the positions of the WPC origin with respect to the machine tool varies for each machine tool. Some systems use a *zero-set button* to set the WPC origin. On other types of NC systems, the WPC is set with a G54 or a G92 code followed by X, and Z dimensions.

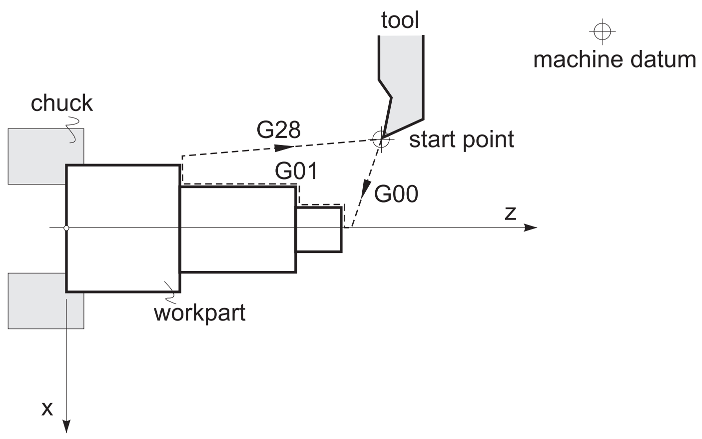
The G54 code tells the machine where the position of the WPC measured from the machine zero point is. *Machine zero point* (*machine datum*) is a fixed point on the machine tool and cannot be programmed or altered.



Example of how G54 is used to set the WPC. Note that in turning X is given as a dimeter, not radius.

Another important point is the *program start point* (also *tool home position*). This point is selected by the programmer at some distance from the workpiece, not too far to save some time when the tool returns home, and not too close to allow for safe indexing of the tool turret when the cutting tool is changed. The program, therefore the new part machining, starts and ends with the tool at home position, but the tool needs also to be returned to home whenever a tool change take place during the program execution.

Some NC system use a G28 command to return to home position; other systems return to home authomatically when a tool change (M06) is commanded.

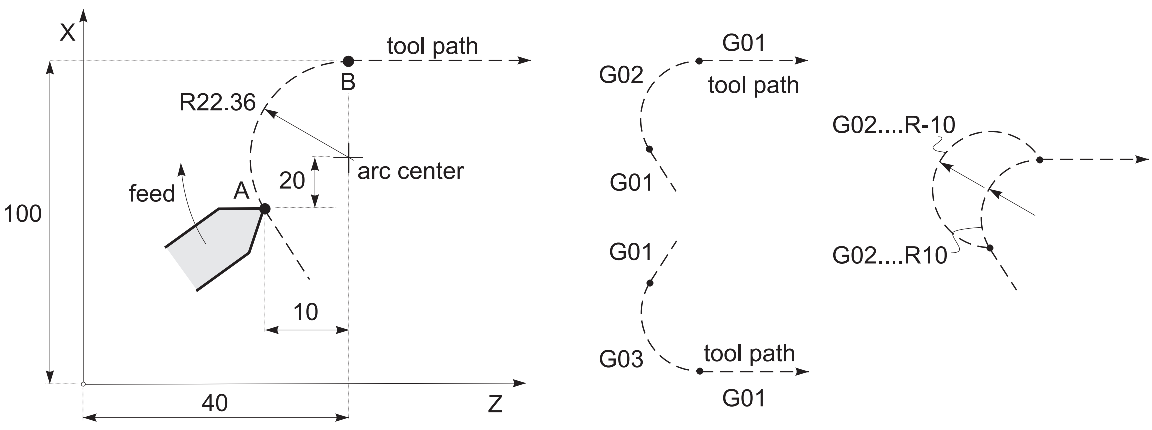


The use of G-codes for rapid positioning of the tool (G00), linear feed motion (G01) and rapid home return (G28).

Linear and circular interpolation

A G01 *linear interpolation* code moves the tool to a position with coordinates defined with program words in a straight, including angular line at the specified with F-code feed rate. The command is modal and is active until either a G00, or G02, or G03 overrides it.

NC system are capable of commanding a circular motion. Arc movement is known as *circular interpolation* and is carried out with a G02 (*clockwise circular interpolation*) or G03 (*counter clockwise circular interpolation*) codes. The arc radius is specified either by the incremental dimensional words I and K, which defines the position of arc centerpoint with respect to the arc start point, or directly by the radius R-code. In both methods, the program block, which starts with a G02 or G03 codes must also include the coordinates of the arc end point. If R-code is used, arcs less than 180° are given a positive radius and arcs more than 180° are given a negative radius value:



N0010 G02 X100.0 Z40.0 I20.0 W10

or

N0010 G02 X100.0 Z40.0 R22.36

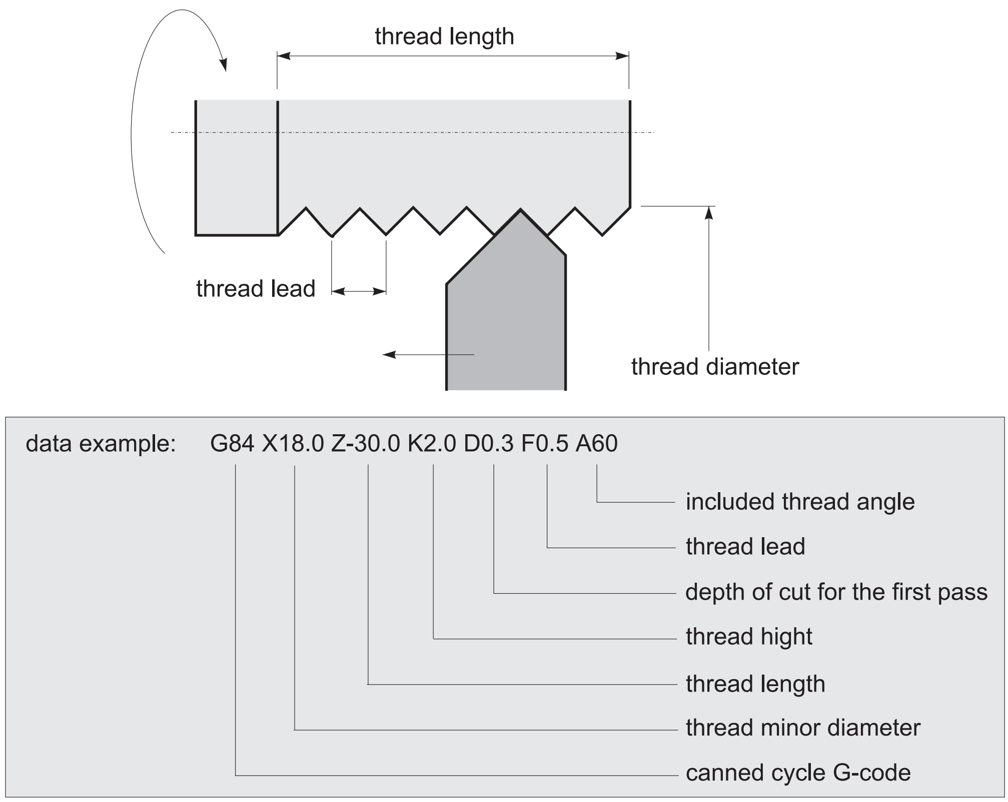
Linear and circular interpolation.

Cycles

The repetitive program (and machining) sequence is called a *cycle*. Cycles are classified into two principle groups,

❶ *canned cycles* (also *fixed cycles*), and
❷ *user-defined cycles* (*sub-routines*).

Canned cycles are an inbuilt feature of the NC system. The usage of canned cycles makes easier programming for threading, drilling holes and other repetitive machining tasks. The next figure illustrates a thread cutting canned cycle:



Example of threaded canned cycle.

User sub-routines are useful, when the necessary canned cycle is not available. The user sub-routine is a NC program, which describes a sequence of operations, which is often repeated when machining particular part. The sub-routine is called from the main NC program with a M98 command.

A special type of user-defined cycles are so-called *macros*, which are generic cycles with parametric variables. The macro is called from the main program with a set of numerical values for these variables. This allow to use one and the same macro to machine different in size, but similar in shape components. Programming with macros is often referred to as a parametric programming.