TG Motion

version 4

Interpolator

operation manual

TG Drives, s.r.o.
Olomoucká 1290/79
CZ - 627 00 Brno

Tele: +420 548 141 811
Fax: +420 548 141 890
E-mail: info@tgdrives.cz

Revision History

date	version	revision
31 July 2017	1.0	Initial release

Contents

1. Interpolator

1.1 CNC module

CNC is a software module, which performs a sequence of motion (G-instructions) and input/output commands (M-functions), which is determined by a **G-code**. Interpolator is one of its constituent parts.

1.2 Interpolator

The interpolator module calculates the positions and speeds of individual servo drives (axes) in such a way that the resulting motion is carried out uniformly by all axes. Following interpolation types are available: linear interpolation (movement along a straight line), circular interpolation (movement along a circle performed by two arbitrary axes), or, as the case may be, helix interpolation (two axes perform a circular motion, the other perform a linear motion).

To calculate three independent final trajectories of a multi-axis motion, **TG Motion** offers three independent **Interpolators**, each of which being able to work with as many as ten servo drives. The shared memory of **TGM_Interpolator** acts as an interface between the CNC module and other applications (PLC, Windows applications). Most registers are only intended for reading and are displaying actual values of the CNC module.

As the G-code is mostly written in [mm] units, the interpolators work with [mm] units. To set the conversion from [mm] to [inc], Command structure or Ratio register is used (see below).

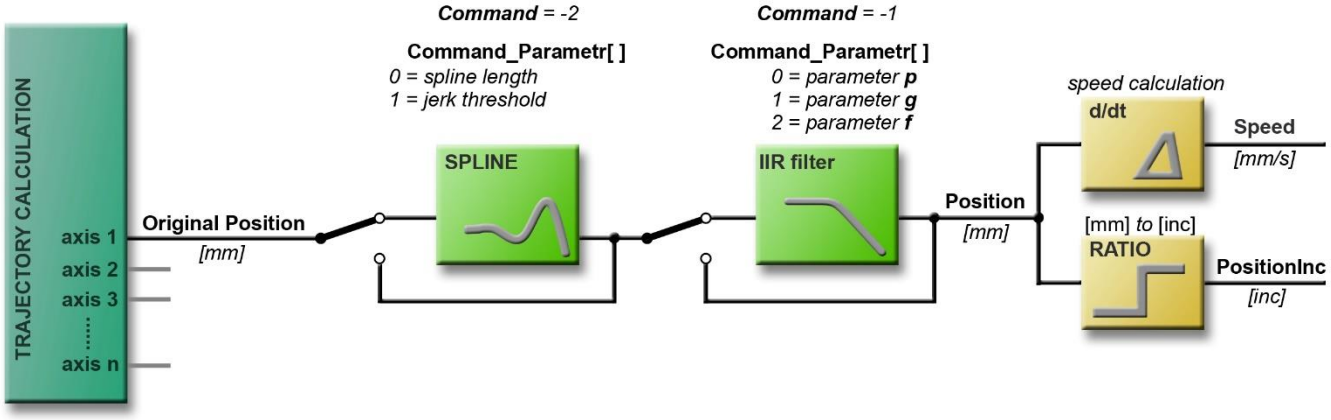

Fig. Interpolator Block Diagram

1.3 PLC and Windows application competences

PLC

Sets the interpolator basic parameters, manages M-functions and deals with the interpolator emergency stops. The interpolator cannot be started from PLC.

Windows applications

Convert the G-code into an intermediate binary code, load it into the interpolator buffer in the shared memory and start and stop the interpolator operation. The interpolator buffer is not accessible.

1.4 G-code

G-code is the name of a programming language, which controls NC and CNC machine tools. This is a text format code, which tells the machine tool what action it is to carry out. The most frequently used commands of the G-code are the **G-instructions** and the **M-functions**. Their designation always starts with a letter (G, M) and if followed by a numerical parameter specifying what the instruction has to perform. The G-instructions are exactly defined, whereas most of the M-functions can be defined by the user.

Example of G-code

```
G00 X60.6051 Y40.7723
G42
M51
G02 X0 Y-2.5202 I3.0336 J-1.2601
G02 X0.922 Y-1.5646 I6.0604 J2.5173
G02 X1.079 Y-1.0269 I4.2143 J3.3479
G02 X2.2624 Y-1.3678 I9.8772 J13.7817
G02 X1.9856 Y-0.8109 I6.1778 J12.2908
G02 X3.015 Y-0.7892 I8.8515 J27.6632
G02 X6.2889 Y0 I3.1444 J33.011
M50
G40
G00 X-28.9085 Y30.4365
G42
M51
G40
M2
```

G-instructions

They are mostly intended for rapid positioning, translation motion along a straight line or an arc, drilling or cutting. It is the role of the interpolator to look after their execution. The G-instruction execution progress can be interrupted by an **M-function** intervention. In this case, the interpolator can be waiting until the **M-function** is finished and subsequently continue executing the **G-code.** *(Refer to Normal and continuous M-functions.)*

Supported G-instructions

basic G-instruction		extended G-instruction	
G0	rapid feed	G4	dwell [s]
G1.	linear interpolation	G25	subprogram call
G2	arc interpolation clockwise (CW)	G26	cycle call
G3	arc interpolation counter clockwise (CCW)	G27	program jump
G40	tool compensation cancel	G29	label or text comment
G41	tool compensation - left of contour	G53	zero point shift cancel
G42	tool compensation - right of contour	G54	zero point shift
G90	absolute programming	G92	coordinate value setting
G91	incremental programming		

M-functions

If an arbitrary M-function is the constituent part of the code, its execution must be ensured in the PLC user code. The Interpolator does not deal with M-functions, it only waits for their execution.

M-functions may be user-defined (subject to exceptions) (see: *register M_Function_Parameter*).

Normal and continuous M-functions

Normal M-functions (**Mx**, where **x** < 1000) – execution of G-code stops, PLC carries out the M-function, when finished, it sets the register M_Func = 0. Subsequently, the interpolator continues performing the G-code of the next instructions.

Continuous M-functions (**Mx**, where **x** > 1000) – PLC launches the execution of the M-function content, but the interpolator does not wait for the M-function execution and continues performing the G-code.

Supported M-function table

basic functions	
M0	program stop
M2	G-code end, may be redefined
M3	spindle rotation CW, may be redefined
M4	spindle rotation CCW, may be redefined
M5	spindle stop, may be redefined
M6	tool change, may be redefined
M7	cooling on, may be redefined
M8	lubrication on, may be redefined
M9	cooling and lubrication off, may be redefined
M17	return from subprogram (RETURN), may be redefined
M29	text message output (PRINT), may be redefined
M30	G-code end, may be redefined
M80	mirroring off, may be redefined
M81	mirroring in x axis, may be redefined
M82	mirroring in y axis, may be redefined
M83	mirroring in z axis, may be redefined
M84	mirroring in x and y axes, may be redefined
M85	mirroring in x and z axes, may be redefined
M86	mirroring in y and z axes, may be redefined
M87	mirroring in x, y and z axes, may be redefined
M99	feed default value definition, may be redefined

2. Trajectory smoothing

The **G-code** often works with polynomials consisting of short straight-line segments. Sometimes, abrupt speed changes may occur in consequence of an incorrectly written code. All of the instantaneous speed changes may result in an undesirable acceleration in some axes, and, therefore, a mechanical shock in some servo drives. Two tools have been designed to smooth up the calculated trajectory.

2.1 Spline

Spline – smoothing function

It is recommended to use the **Spline** function in the case of incorrectly created G-codes, in which the individual segments do not link up smoothly and continuously, or in the case of G-codes, where the resulting trajectory has been assembled from linear segments which have arisen from a coarse division. The **Spline** is also suited to smooth up abrupt speed changes resulting from the mechanics mathematical model (for example, tilting heads).

The **Spline** function is activated in all axes simultaneously. It cannot be applied to some of the axes only. When the **Spline** function having been activated, the change in acceleration (jerk) is checked in each axis. In the case where a change exceeding the permitted one is found, the segment in question will be interpolated by a curve.

*The longer the Spline (the larger the Spline buffer) is, the better is the smoothing. The smoothing by means of **Spline** function goes to the detriment of the positioning accuracy.*

Spline activation and parameterization

To activate and parameterize the **Spline** function, structure **Command** of the interpolator in question is used. If the Command register value = −2, the Command parameters have the following meanings:

Command_Parametr[0] – specifies the number of points, which are interpolated by the spline. In this way, the size of the spline buffer is determined, too. The range of the parameter setting is 50–500 points (steps). For Cycle_Time = 500 µs, the calculation step is 100 µs, for Cycle_Time = 250 µs, the calculation step is 50 µs.
To turn off the **Spline** function, set the spline length to zero (Command_Parametr[0]=0).
Command_Parametr[1] – specifies the limiting value of the acceleration change (jerk), i.e., from what acceleration value up the Spline function will be activated. Its unit is [mm/s³]. A suitable value to set is 1 000 000 mm/s³.

After both mentioned parameters are set, the value to which **Command** register must be set to −2. After the Command is carried out, TG Motion will set this **Command** to zero.

Movement delay

With respect to the positioning as calculated by the interpolator, the motion will be delayed by the length of the Spline buffer after the passage through the Spline function takes place. All axes are positioning synchronously all the time, because the buffer size is the same for all axes. Also the M-functions are called synchronously with the resulting motion, because a buffer of the same size is used for them.

Example: Spline setting

A **Spline** function with a spline length of 20 ms (for Cycle_Time = 500 µs) is turned on. First of all, the values of Command_Parametr should be set and, as the last, the Command register value set to −2.

```
Interpolator.Command_Parametr[0] = 200;
Interpolator.Command_Parametr[1] = 1000000;
Interpolator.Command = -2;
```

2.2 IIR Filter

IIR Filter – Smoothing function

The **IIR Filter** (Infinite Impulse Response) has been designed to smooth up the resulting speed and remove undesired rapid changes. It is a mathematical model of a low pass filter with a slope of 12 dB/octave (2-Pole) calculated by means of ***H(s) = q / (s^2 + p*s + q)***, allowing to set three parameters, namely,

p, **q** – parameters to set the filter response (see the table)
f (cut-off) – time during which the fitted trajectory returns to the original, calculated trajectory

Parameter values for selected filters

p	q	filter type	description
√2	2	Butterworth	an overshoot takes place before **f**, low delay
3	3	Bessel	moderate overshoot, smooth shape, higher delay
2	1	critically damped	smooth and gentle shape, no overshoot, highest delay

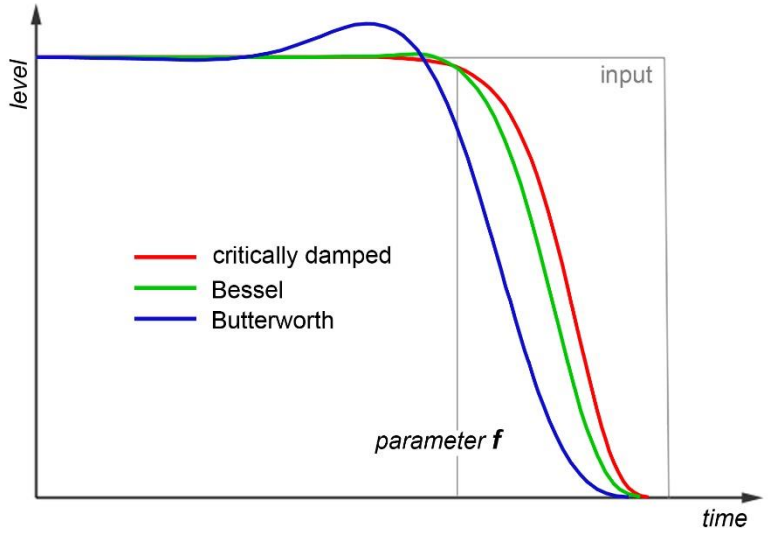

Fig. Graphical representation of the response of some filter types

IFR Filter activation and parameterization

The **IIR Filter** parameter values can be set by means of **Command** structure**.** Given the Command register value = −1, the Command parameters have the following meanings:

Command_Parametr[0] – sets the filter parameter **p**.
Command_Parametr[1] – sets the filter parameter **q**.
Command_Parametr[2] – determines **f** – the time needed for returning to the original trajectory.

The Command values are of integer type. The actual physical values of the parameter are obtained by dividing the Command value by 10^6 (Command_Parametr[1] = 1 000 000, filter parameter = 1.0).

Command_Parametr[3] – sets the filter activity bit mask for the different axes. Only the first 10 bits are used for the 10 axes, the other bits being ignored (see the *registers*).

Movement delay

IIR Filters are delaying the calculated movement. The delay depends on input data. It changes dynamically and its amount cannot be determined in advance. M-functions are signalled from a continuous G-code for the Interpolator being "in standstill". The physical movement of the servo drives may not be completed at this time.

*When an **IIR Filter** is used, the servo drive will "float" and be closing on the Interpolator. One has always wait until the physical movement of all servo dives is really completed.*

The parameter setting of **IIR Filters** is common for all interpolator axes. The filter can be activated independently for each of the axes using the bit mask of the register of **Command_Parametr[3]**. Therefore, one has to keep in mind that the resulting movement of the axes will not probably be mutually as synchronous as it was originally calculated.

3. Structure Command

3.1 Command_Parametr registers

In this chapter the meaning of the registers of Command_Parametr[ ] is described for selected register values.

Setting the mm to inc conversion ratio – Command = 1024

In the G-code, one mostly uses the [mm] units, so that the Interpolator uses also the [mm] units. As the servo drives are operating with increments [inc], it is essential to specify the inc/mm conversion ratio on the basis of the inc/rev ratio of the servo drive Servo[x].Resolution). The conversion ratio is set by means of the structure **Command** of the respective Interpolator. The calculated position in the respective axis will be multiplied by the conversion ratio and the converted value will be sent to the servo drive.

Command_Parametr	description
0	specifies the number of the axis for which the ratio is set [0–9]
1	numerator in the conversion ratio [inc]
2	denominator in the conversion ratio [mm]

Setting the actual position – Command = 2048 and Command = 2049

The position in the trajectory is set by means of Command = 2048, whereas the position beyond the trajectory, by means of Command = 2049. For both variants, all Command_Parametr[0–9] acquire the same meaning.

Command_Parametr	description
0	actual position in axis 0 [inc]
…	…
9	actual position in axis 9 [inc]

Setting the IIR Filter parameter – Command = −1

Command_Parametr	description
0	**p** parameter of filter **–** setting the filter response (combined with **q** parameter)
1	**q** parameter of filter **–** setting the filter response (combined with **p** parameter)
2	**f** parameter of filter **–** time during which the fitted trajectory returns to the original trajectory
3	bit mask of IIR Filter activity for each axis of the Interpolator *Example: xxxx xx00 0000 0101 – only IIR Filters for axes 0 and 2 are active*

Spline parameter setting – Command = −2

Command_Parametr	description
0	specifies the number of points to be interpolated by the spline; the setting range is 50–500 points
1	upper limit of acceleration change evaluation [mm/s³]

4. Structure LookAheadBuffer

4.1 Description of the structure

The structure **LookAheadBuffer** is a table of important parameters of eight parts – consecutive G-code items. The first item in the table is just the part being actually executed, the next items are seven immediately following parts.

LookAheadBuffer functions as a shift buffer. After the actual part is executed, the table data will shift. The first is again the part being actually executed. To the last place, the 8th following part from the actually executed one is inserted. The structure **LookAheadBuffer** is filled by the Interpolator. From the viewpoint of PLC, its registers are for reading only.

G-code part

As a part, one G-code item is defined, namely either a G-instruction (G0–G3) or a normal M-function which is meaningful from the motion viewpoint (M3–M999).

4.2 Use of the structure

The structure **LookAheadBuffer** is used to modify the technologies according to the instructions and function of the following parts or, as the case may be, their values. As the Interpolator can neither be switched on or off from the PLC (except for emergency stop), all necessary measures must be taken during the existing M-function operation. This must be ensured by the PLC code developer.

For example, PLC may, during the M-function execution, change the angle of the machining head in the square corner according to the following function tangent, or it may slow down the motion by means of Rel_Speed register, provided that it foresees a heavy change of the tangent. Actual tangent is available in the case of a circular motion, which may be used to rotate the head. In the case of short straight-line segments, a broken line may be smoothed up by gradual rotations of the head. To perform all of these actions, one has to know what will follow the part being actually executed. It is just the purpose for which the structure **LookAheadBuffer** may be used.

Remark: It may sometimes happen that a negative M-function number appears on the Movement_Code register for Movement_Type = 2. This is an internal function of TG Motion, which is not necessary to be dealt with.

The registers of the structure **LookAheadBuffer** are described in a table in the Appendix chapter .

5. Appendix

A. List and description of registers in the structure Interpolator

Interpolator

name	access	offset	description
Number	R	0	interpolator number, may equal to 0, 1, 2
Number_Axes	RW	4	number of axes, with which the interpolator operates; permitted range: 1–10
Buffer_Size	R	8	maximum number of G-code parts, permitted values are 1000–100 000
Command	RW	12	command number: 4 = emergency stop on the trajectory 5 = emergency stop (after the stop it is reported as out of the trajectory) 8 = normal stop on the trajectory 9 = normal stop (out of the trajectory) 1024 = setting the conversion ratios (mm to inc) – see chapter *Command* 2048 = setting the actual position (on the trajectory) – see chapter *Command* 2049 = setting the actual position (out of the trajectory) – see chapter *Command* −1 = setting the IIR Filter parameters – see chapter *Command* −2 = setting the Spline parameters – see chapter *Command*
Command_Parametr [0-11] *(12 registers)*	RW	16	12 parameters of integer type, whose meaning and values depend on the type of command number
Command_Status	R	64	actual status of the command: 0 = previous command was executed successfully and next command can be activated 1 = actual command is being performed −1 = error occurred during the command execution
Status	R	68	actual status of interpolator: 1 = movement on trajectory is under way 3 = stop at the trajectory end 4 = at least one part of the trajectory is in the buffer, start may be called 6 = stopping on emergency ramp 7 = interpolator stopped after emergency stop 8 = interpolator was stopped during the buffer filling
Act_Part	R	72	number of the part actually executed
Address_External_Position	RW	76	offset of TGM_Data address, where external sensor position is saved; value is of integer type
M_Func	RW	80	value of M-function; if M < 1000 the Interpolator stops and waits until the value reaches zero; it then continues with a next segment; if M > 1000, execution of the G-code continues
Act_G_Func	R	84	value of the actually executed G-instruction
Act_M_Func	R	88	value of the actually executed M-function
Last_Cont_M_Func	R	92	saved value of the last continuous M-function (M > 1000)
Run_Flag	R	96	- lower 16 bits indicate the status of the actual part: 0 = STOP (interpolator is inactive) 1 = RUN (normal G-instruction is being executed) 2 = WAIT_WINDOW (for turning) 3 = WAIT_PULSE (for turning) 4 = WAIT_MFUNC (M-function execution started 5 = WAIT_MFUNC_WAIT_FOR_END (waiting for M-function termination) - upper 16 bits for the speed progress status: 1 = no movement 2 = acceleration 3 = required movement speed has been reached 4 = deceleration 5 = next speed segment 6 = deceleration on the last segment 7 = deceleration on emergency ramp *note: from TGM420 version on*
Tool_Number	R	100	actual tool number (drilling, milling cutter, …) *note: from TGM420 version on*
Orig_Position *(10 registers)*	R	376	calculated coordinates of all axes [mm]
Position *(10 registers)*	R	456	coordinates modified by Spline function or IR Filter [mm]
PositionInc *(10 registers)*	R	536	register Position multiplied by Ratio (conversion ratio) [inc]
Backlash *(10 registers)*	R	616	actual backlash value for each axis [inc]
Offset *(10 registers)*	RW	696	the offset values are added to PositionInc position, these values are adjusted by the user [inc]
Speed *(10 registers)*	R	776	actual speed of each axis after Spline and IIR [mm/s]
Ratio	RW	856	G-code unit conversion ratio (multiplier) (usually mm) for position increments (servo drive positions)

name	access	offset	description
M_Function_Parameter *(32 registers)*	R	936	- parameters of G-code M function, in total, 26 values alphabetically sorted; some letters cannot be used as they are reserved by the system - reserved parameters (the indices are counted up from 0): G = index 6 M = index 12 N = index 13 P = index 15
Rel_Speed	RW	1192	relative speed of interpolated movement, the coefficient being within 0.01–2 (1–200 %)
Set_Speed	R	1200	required speed from G-code (resulting from G-code F-instruction or from speed table) [mm/min]
Act_Speed	R	1208	actual speed [mm/min]
Move_Distance	R	1216	actual total covered distance [mm]
LookAheadBuffer	R	2048	information table on the segments to follow; in sum, 8 items of the structure described in following table; the first item describes the segment, which is being carried out; one entry in LookAheadStructure is 1792 bytes long *note: from TGM420 version on*

B. List and description of registers in the structure LookAheadBuffer

Structure LookAheadBuffer

note: from TGM420 version on

name	access	offset	description
AllParams *(26 registers)*	R	0	all entered addresses of M-functions of a particular G-code for a given segment (26 letters of English alphabet)
Tangent	R	208	actual movement tangent in XY plane; if the segment, which is being executed, is an arc (G2 or G3), the tangent changes continuously; it determines the movement tangent for future segments
MovementType	R	216	type of record: 0 = invalid record (the number of segments left for execution is less than the actual index of the LookAheadBuffer table), less than 8 segments remain to the movement execution end 1 = normal function (G0, G1, G2, G3) 2 = M function
MovementCode	R	220	G-instruction or normal M-function number (the number of the continuous M-function will not appear in the register, but it will appear in AllParams, in letter M)
Plane	R	224	- circular interpolation plane: 17 = XY 18 = XZ 19 = YZ - no other planes are defined
Tool	R	228	tool number
EndPos *(10 registers)*	R	232	end position of each axis segment [mm] (absolute coordinates)
StartAngle	R	312	arc starting angle [°] – angle of the arc center connection line with the starting point; the register value is cyclical (0–360); for linear movements or M-functions, the register value is greater than 10^{38}
EndAngle	R	320	arc end angle [°] – angle of the arc center connection line with the initial point; the register value is cyclical (0–360); for linear movements or M-function the register value is greater than 10^{38}
Radius	R	328	- arc radius [mm] - for linear movements or M-functions the register value is 0