DOCUMENTATION ISG-kernel

Functional description
Limiting decoder block ahead

Short Description:
FCT-C24

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

or (EN)

https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

Copyright

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

➢ Indicates an action.

⇨ Indicates an action statement.

⚠ **DANGER**

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ **CAUTION**

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Release Note

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

List of figures

1   Overview

Task

Due to cascaded processing in the NC channel there is a time delay between decoding an NC program and executing the motion blocks in the interpolator. This means that decoding always precedes execution.

With HSC programs this look-ahead function may involve several hundred motion blocks. In addition, changes initiated by the user in the NC program may only become effective with a time delay.

By limiting the decoder block ahead function the reaction time to program influences can be shortened.

Properties

With certain technologies, such as flame or water-jet cutting, the user can adapt specific process parameters by using variables, e.g. feed rate in the running NC program.

When the block-ahead limit is active, it means that these changes become effective rapidly and are included timely in the cutting process.

Parametrisation and programming

Two types of block ahead limiting can be activated:

1. By configuration in the channel list, whereby block-ahead limiting becomes immediately effective at program start.
2. By programming in the NC program, whereby block-ahead limiting can also be activated only for specific program parts.

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Description

Motivation and function

After an NC program is started, the individual NC blocks are continuously read

1. by an interpreter module, called the decoder,
2. converted into an internal representation and
3. output to the NC channel for further processing.

In the NC channel, the data is then read in by further modules (tool radius compensation, path preparation, etc.), edited and passed through up to the interpolator.

Current processing of commands in the interpolator corresponds to the **actual state** of the machine, whereas the decoded commands correspond to the **future state** of the machine.

During program decoding, parameters (e.g. external variables) are read and depending on this, they run through different program sequences.

Decoder block ahead is not limited. As a result, the time may be very long between program influencing during decoding and program execution. For the user a parameter change that influences program decoding is only noticeable after a considerable delay in execution.

To keep this program influencing as close as possible in time to program execution, the decoder block ahead must be kept as short as possible, i.e. the user notices program influences within a relatively short period of time.

```
%Main
N10 X.. Y.. Z.. G01 F2000
N20 X.. Y..
N30 $IF V.E.Param1
N40 Z..
N...
N200 $IF V.E.Ext_influence
N210 X..
N220 $ENDIF
N1000 M30
```

Time
sequence

NC channel

DECODER

```
%Main
N10 X.. Y.. Z.. G01 F2000
N20 X.. Y..
N30 $IF V.E.Param1
N40 Z..
N...
N200 $IF V.E.Ext_influence
N210 X..
N220 $ENDIF
N1000 M30
```

Decoding at N200

N190..

TOOL RADIUS COMPENSATION

Internal
dynamic buffer

```
%Main
N10 X.. Y.. Z.. G01 F2000
N20 X.. Y..
N30 $IF V.E.Param1
N40 Z..
N...
N200 $IF V.E.Ext_influence
N210 X..
N220 $ENDIF
N1000 M30
```

N130..

PATH PREPARATION

Internal
dynamic buffer

```
%Main
N10 X.. Y.. Z.. G01 F2000
N20 X.. Y..
N30 $IF V.E.Param1
N40 Z..
N...
N200 $IF V.E.Ext_influence
N210 X..
N220 $ENDIF
N1000 M30
```

**Decoder
block
ahead**

**N30..
N20..**

INTERPOLATION

Internal
dynamic buffer
N10 X.. Y.. Z..

Interpolation
at N10

```
%Main
N10 X.. Y.. Z.. G01 F2000
N20 X.. Y..
N30 $IF V.E.Param1
N40 Z..
N...
N200 $IF V.E.Ext_influence
N210 X..
N220 $ENDIF
N1000 M30
```

Fig. 1: Decoder block ahead and NC channel memory effect

Limiting block ahead decoding

Therefore, due to the cascaded processing of NC blocks, program decoding is always ahead of actual execution/interpolation.

The number of NC blocks by which the decoder leads is not constant and is only limited indirectly by the number of available memory locations. As a result, the decoder block ahead is dependent on:

- the type of processed NC blocks themselves (short/long motion blocks, parameter calculation, technology functions, switching commands, etc.)
- the functions active in the NC channel (TRC, HSC, spline interpolation, etc.) and their buffer sizes.
- system settings (cycle time, task distribution).

In the case of certain program sequences, the user would like to specify the maximum number of blocks that decoding may be ahead of the interpolator level, for example.

To achieve this, the decoder block ahead can be defined by:

- specifying a number of NC blocks or motion blocks or
- a time-based specification (motion time).

GEOMETRY
N10 Z40
...
N20 F= V.E.Feedrate
N30 G91 Z60
N40 F= V.E.Feedrate
...
M30

CAD/CAM system

%program
N10 G01 F1 Z40
...
N20 F= V.E.Feedrate
N30 G91 Z20
N30 G91 Z20
N30 G91 Z20
N40 F= V.E.Feedrate
...

~~Online Parameter Changing~~

PLC

CNC channel

Feedrate for calculation of rapid movement times:
$F_{Calc} = MIN [F_a , F_{Prog}]$

Check block or time dependent decoder ahead limiting **STOP**

DEC

Chamfer/radius

Δt_{Dec}

TRC

Equidistant

Contour masking

Δt_{TRC}

BAVO

Micro joints

Contour

Spline

Dynamic

$\Delta t_{Path\ prep.}$

report →

- Block count
- Block request in the event of undersupply

Average feed rate F_a of the next block (after dynamic planning and override)

IPO
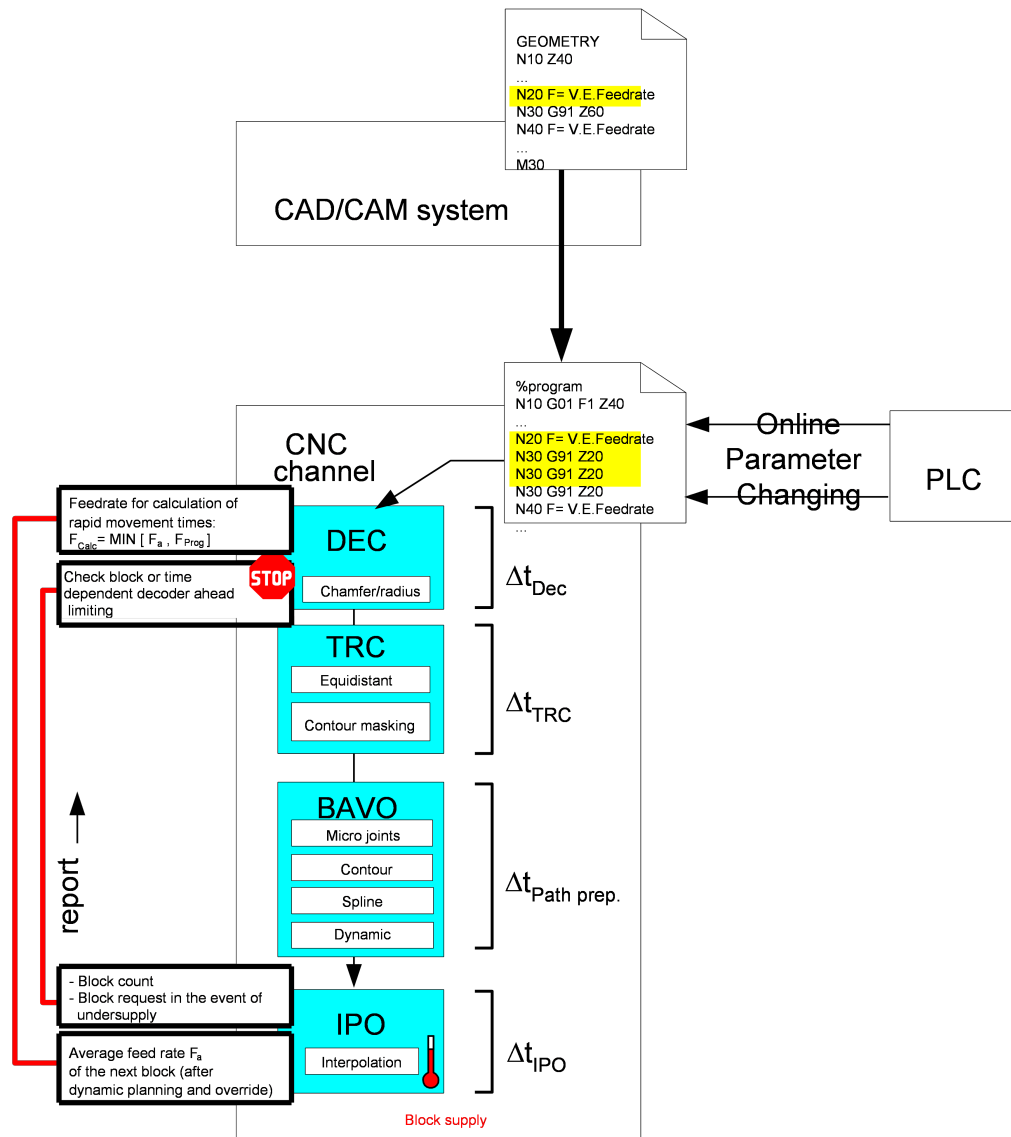
Interpolation

Δt_{IPO}

Block supply

Fig. 2: Data to determine decoder block ahead

3   Block ahead limiting on the basis of the number of blocks

Definition of limiting

Limiting is achieved by defining the number of NC blocks by which the decoder may be ahead of interpolation.

This user can make this definition:

- in the NC program or
- in the channel parameters before the control unit starts.

As a result the CNC provides each decoded CNC row with a unique number (block count).

Status display

When decoding reaches the block ahead limit, the waiting state of the decoder is indicated by the following status bit.

The status bit is cancelled when decoding is continued. The display datum can be indicated in the diagnostic data.

0x00100000   BLOCK_AHEAD_LOCK_ACTIVE

Internal sequence

For every NC block output, the decoder saves the associated block count and an associated output count in an internal table.

During the program run, the decoder determines the related delta of the output count (block ahead) using the block count signalled by the interpolator.

If this delta is greater than the permitted defined limit, the decoder interrupts processing. Before continuing the decoding function, the decoder waits until the interpolator again undershoots the defined limit, i.e. it catches up.
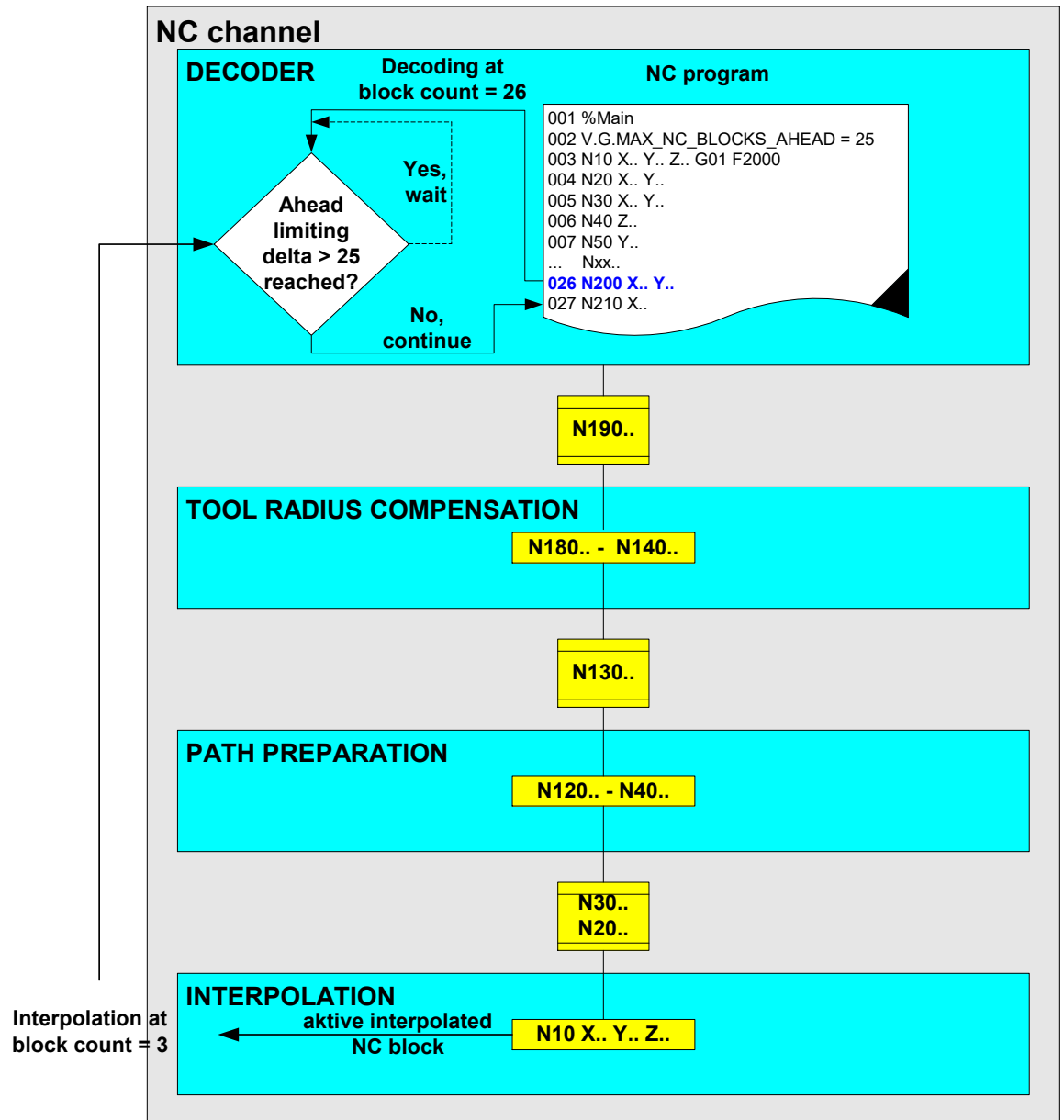
NC channel

DECODER

Decoding at block count = 26

NC program

```
001 %Main
002 V.G.MAX_NC_BLOCKS_AHEAD = 25
003 N10 X.. Y.. Z.. G01 F2000
004 N20 X.. Y..
005 N30 X.. Y..
006 N40 Z..
007 N50 Y..
...   Nxx..
026 N200 X.. Y..
027 N210 X..
```

Ahead limiting delta > 25 reached?

Yes, wait

No, continue

N190..

TOOL RADIUS COMPENSATION

N180.. - N140..

N130..

PATH PREPARATION

N120.. - N40..

N30..
N20..

INTERPOLATION

aktive interpolated NC block

N10 X.. Y.. Z..

Interpolation at block count = 3

Fig. 3: Checking the decoder/interpolator block ahead limit function

In the figure the delta of the block ahead limit is 26-3 = 23, i.e. decoding is continued since the block ahead limit of 25 is not yet reached.

3.1 Application

Selecting block ahead limit

The block ahead limit can be defined

- directly in the NC program or
- in the channel parameter list.

3.1.1 Block number limiting via NC command

Activating via NC command

Global variables are provided to define and select decoder block ahead limiting in the NC program.

2 (exclusive) block number limiting options are possible via variables:

V.G.MAX_NC_BLOCKS_AHEAD = *<value>*

This variable defines the maximum number of channel-related NC blocks (rows) by which the decoder may be ahead of the interpolator. Channel-related NC blocks are program rows that lead to further actions in the next NC channel (motion blocks, technology functions, switching commands).

Pure parameter calculations, empty rows and comment rows without block numbers are only processed at interpolator level and are ignored when the block ahead value is defined.

V.G.MAX_MOTION_BLOCKS_AHEAD = *<value>*

This variable defines the maximum number of NC motion blocks (G00, G01, G02/03) by which the decoder may be ahead of the interpolator. All other NC program rows are ignored when the block ahead value is defined.

Activate block ahead limiting by setting the V.G. variables to a value unequal to 0. Before you start block ahead limiting, set the decoder and interpolator to the same processing state in relation to the NC program, i.e. they should be synchronous. This can be achieved by:

- selecting block ahead limiting immediately at the start of the NC program or
- executing a #FLUSH WAIT immediately before selecting block ahead limiting

Block ahead limiting then remains active until main program end M30.

> **!**
>
> ### Attention
>
> Only one block ahead limiting option may be active at any one time. The error message P-ERR-21575 is output if variables are used at the same time.

Block number limiting via NC command

If the value of these V.G variables is set to 0, limiting is deactivated again.

Programing Example

Block number limiting via NC command

```
Block ahead limiting by definition in the NC program

%MAIN
N005 V.G.MAX_NC_BLOCKS_AHEAD = 10 ;Select block ahead limiting 10 NC
blocks
N190 #HSC[ MODE 1 CONTERROR 0.01]
N270 G54 G90

…
N320 #HSC ON
N47200: G1 F4000 Y597.771 Z-5.596
N47210 Y597.343 Z-5.205
N47220 Y596.911 Z-4.861
N47230 Y596.475 Z-4.561
N47240 Y596.067 Z-4.321
N47250 Y595.643 Z-4.112
N47260 Y595.278 Z-3.965
N47270 Y594.905 Z-3.841
N47280 Y594.524 Z-3.747
N47290 Y594.136 Z-3.683

…
N47350 V.G.MAX_NC_BLOCKS_AHEAD = 0 ;Deselect block ahead limiting

…
N47449 #FLUSH WAIT ;Synchronise decoder and interpolator
N47450 V.G.MAX_MOTION_BLOCKS_AHEAD = 20 ;Select block ahead limiting 20
motion blocks
N47440 Y589.152 Z-5.735
N47450 Y588.76 Z-6.156
N47460 Y588.235 Z-6.767
N47470 Y587.299 Z-7.889
N47480 Y587.034 Z-8.198
N47490 Y586.768 Z-8.463
N47500 Y586.492 Z-8.686
N47510 Y586.235 Z-8.853
N47520 Y585.994 Z-8.971
N47530 Y585.791 Z-9.039
N475403 V.G.MAX_MOTION_BLOCKS_AHEAD = 0 ;Deselect block ahead limiting
…
N999999 M30
```

3.1.2 Limiting the number of blocks via channel parameters

Deactivating via channel parameters

As an alternative, corresponding parameters are available in the channel parameter list to configure block ahead:

max_nc_blocks_ahead   *<Wert>* (P-CHAN-00216)

This parameter defines the maximum number of channel-related NC blocks (rows) by which the decoder may be ahead of the interpolator.

max_motion_blocks_ahead   *<Wert>* (P-CHAN-00246)

This parameter defines the maximum number of NC motion blocks (G00, G01, G02/03) by which the decoder may be ahead of the interpolator.

After program start, the block ahead limit is immediately active and remains selected until the main program ends M30.

The block ahead limit can be modified or deactivated at any time in the NC program by writing the corresponding variable V.G.MAX_NC_BLOCKS_AHEAD or V.G.MAX_MO-TION_BLOCKS_AHEAD.

The value set in the channel parameter list then applies again the next time the program is started.

> ### ⚠ Attention
>
> Only one block ahead limit option may be activated in the configuration at any one time. When parameters are used at the same time, the error message P-ERR-21574 is output.

Deactivating via channel parameters

No limit is active after program start if these parameters are set equal to 0 or if they are not specified in the list.

3.2   Special features and restrictions

Restrictions / special features

If multiple "empty" rows are programmed in the NC program that are only evaluated internally in the decoder, no commands are output to the NC channel.

In particular in the case of block ahead limiting via "MAX_NC_BLOCKS_AHEAD", this can lead to a situation in which decoding is interrupted because the channel no longer supplies any new blocks (jamming).

To avoid this, no large blocks of comment lines, empty rows, parameter calculations, etc. should occur if possible after block ahead limiting is selected.

It may also be necessary to increase the block ahead limit value.

---

i	**Notice**
	To ensure that no jamming of the NC channel occurs when decoder block ahead is limited, you are advised to define a limit of at least **10 rows**.

Since path preparation saves a different number of blocks depending on the active functions in the NC channel, a considerably higher block ahead limit may be required.

The overall storage effect of the channel depends on the following active functions, for example:

- Tool radius compensation
- Blending G61/G262
- #HSC (spline)
- Microwebs
- Planning the dynamics

Depending on the path dynamics, feed rate fluctuations may occur when block ahead limiting is active if motion blocks are relatively short.

---

i	**Notice**
	For stable execution, monitored block ahead mode can be optionally selected by the channel parameter P-CHAN-00270, see Section "Time-related block ahead limiting [▶ 17]"):
	dec_max_ahead_protected ACTIVE

---

4   Time-related block ahead limiting

Definition of limiting

In the case of time-related block ahead limiting, the maximum permissible decoding lead is defined by specifying a time in seconds, i.e. The limit defines by how many seconds the decoding of program execution (interpolation) may be ahead.

In this case, only the motion time of motion blocks (such as G00, G01, G02, G03) is considered as well as the dwell time (G04, #TIME). Other time-consuming NC functions such as M functions, etc. are not considered.

The decoder block ahead limiting function is based on the ideal programmed command velocity and the current block length. This results in particular in deviations in functions which influence contours and dynamics, such as:

- Real-time influences of velocity via FEEDHOLD, Override
- Velocity weightings, e.g. via #VECTOR LIMIT
- Deviation if the programmed velocity cannot be reached, for example due to the geometry path
- Contour influences by polynomial smoothing or spline interpolation

In order to estimate the motion time for block ahead limiting realistically, the interpolation supplies a mean feed velocity (average velocity) of the next motion blocks to be executed.

The average velocity includes curvature-dependent dynamic limits and real-time influences such as override and feedhold. The calculated motion time is then based on the minimum average velocity and the feed valid in the block at the time of decoding (F word, rapid traverse)-

The following conditions apply to calculation of the motion time:

- Linear and circular blocks are considered.
- When motion time on the path is determined, only feed axes are considered (#FGROUP)
- If only slave axes are programmed in the block, the slowest axis is used to calculate motion time.

Restrictions:

- Independent axis motions (INDP_SYN. INDP_ASYN) and the "Z" amount of helical motions are not considered.
- No consideration given to M/H functions, tool changes, etc.

Selecting and deselecting time-related block ahead limiting is executed analogous to block-related block ahead limiting either

- in the NC program by the special V.G variable **V.G.MAX_TIME_AHEAD** = <value in [sec]> or
- generally by the channel parameter P-CHAN-00269: **max_time_ahead** *<value in [10-6 sec]>*

Status display

When decoding reaches the block ahead limit, the decoder waiting state is indicated by the following status bit. The status bit is cancelled when decoding is continued.

0x00200000   TIME_AHEAD_LOCK_ACTIVE

The display datum can be indicated in the diagnostic data.

Internal sequence

Calculating the relevant motion times takes place completely in the decoder.

As a result, the decoder estimates a minimum execution time = path length/velocity for each (motion) block.

The decoder stores the block count and the time required in a table for each motion block that is output. Together with the block count indicated by the interpolator, the time lead relative to the motion block last processed is then determined via the table.

If this delta is greater than the defined permissible limit, the decoder interrupts its execution and waits before continuing decoding until the interpolator again falls below the defined limit, i.e. has in fact "caught up" again.

Effectiveness

Block ahead limiting is only active when the interpolator block supply (Monitored operating mode, section " [▷ 17]Time-related block ahead limiting") is ensured.

Programing Example

Time-related block ahead limiting

Time-related block ahead limiting by definition in the NC program

```
%MAIN

N190 #HSC[ MODE 1 CONTERROR 0.01]
N270 G54 G90
N280 V.G.MAX_TIME_AHEAD = 2 ;Select block ahead limiting 2 seconds
…
N320 #HSC ON
N47200: G1 F4000 Y597.771 Z-5.596
N47210 Y597.343 Z-5.205
N47220 Y596.911 Z-4.861
N47230 Y596.475 Z-4.561
N47240 Y596.067 Z-4.321
N47250 Y595.643 Z-4.112
N47260 Y595.278 Z-3.965
N47270 Y594.905 Z-3.841
N47280 Y594.524 Z-3.747
N47290 Y594.136 Z-3.683
…
N47440 Y589.152 Z-5.735
N47450 Y588.76 Z-6.156
N47460 Y588.235 Z-6.767
N47470 Y587.299 Z-7.889
N47480 Y587.034 Z-8.198
N47490 Y586.768 Z-8.463
N47500 Y586.492 Z-8.686
N47510 Y586.235 Z-8.853
N47520 Y585.994 Z-8.971
N47530 Y585.791 Z-9.039
N475403 V.G.MAX_TIME_AHEAD = 0 ;Deselect block ahead limiting
…
N999999 M30
```

4.1   Programming examples

Example 1

In the test program below, a square with an edge length of 100 mm is moved. Each side is segmented into 100 single blocks. Based on a time ahead of 2 seconds, as values become increasingly smaller it becomes apparent that the contouring speed can no longer be kept constant due to the pulsating supply of blocks.

```
% Quadrat.nc

#SLOPE [TYPE=HSC]
G133 100
V.G.MAX_TIME_AHEAD = 2 ;seconds

G00 G90 X0 Y0
P40 = 5000
P30=100.0 (* Side length of square *)
P20 = 100 (* Number of blocks on side length*)
P10 = P30/P20

$FOR P1=1,P20, 1
N[P1]G01 G91 XP10 FP40
$ENDFOR
G90
$FOR P1=1,P20, 1
N[P1+1000]G01 G91 YP10
$ENDFOR
G90
$FOR P1=1,P20, 1
N[P1+2000]G01 G91 X-P10
$ENDFOR
G90
$FOR P1=1,P20, 1
N[P1+3000]G01 G91 Y-P10
$ENDFOR
G90
V.G.MAX_TIME_AHEAD = 0 ;seconds
M30
```

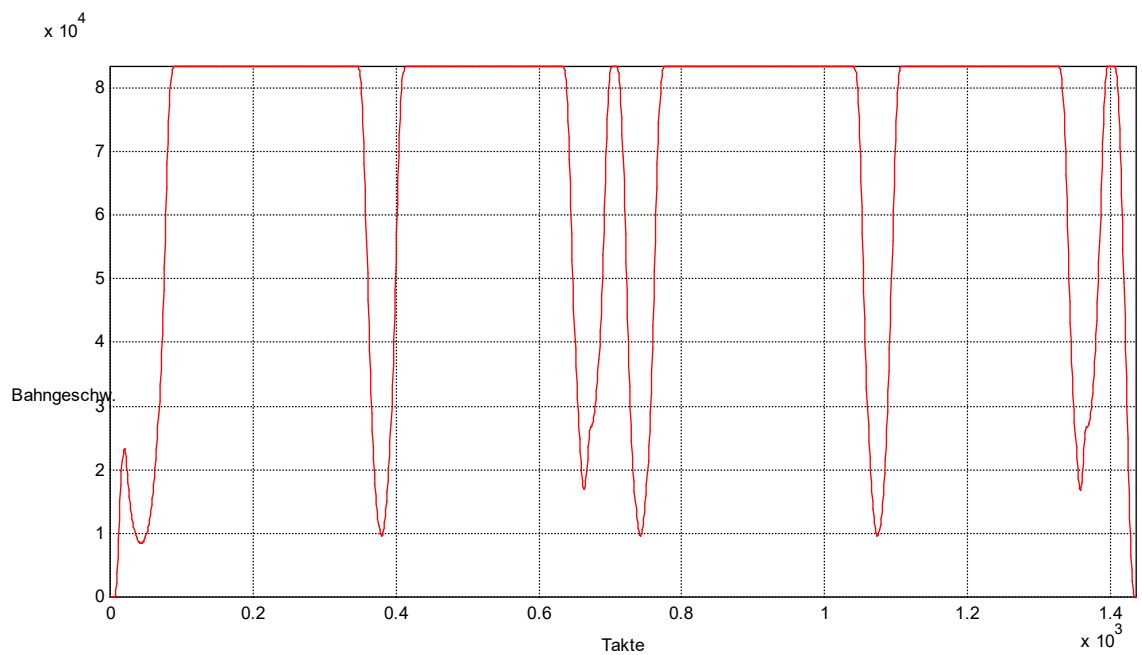**Fig. 4: Test path: Path velocity curve at different times ahead:**

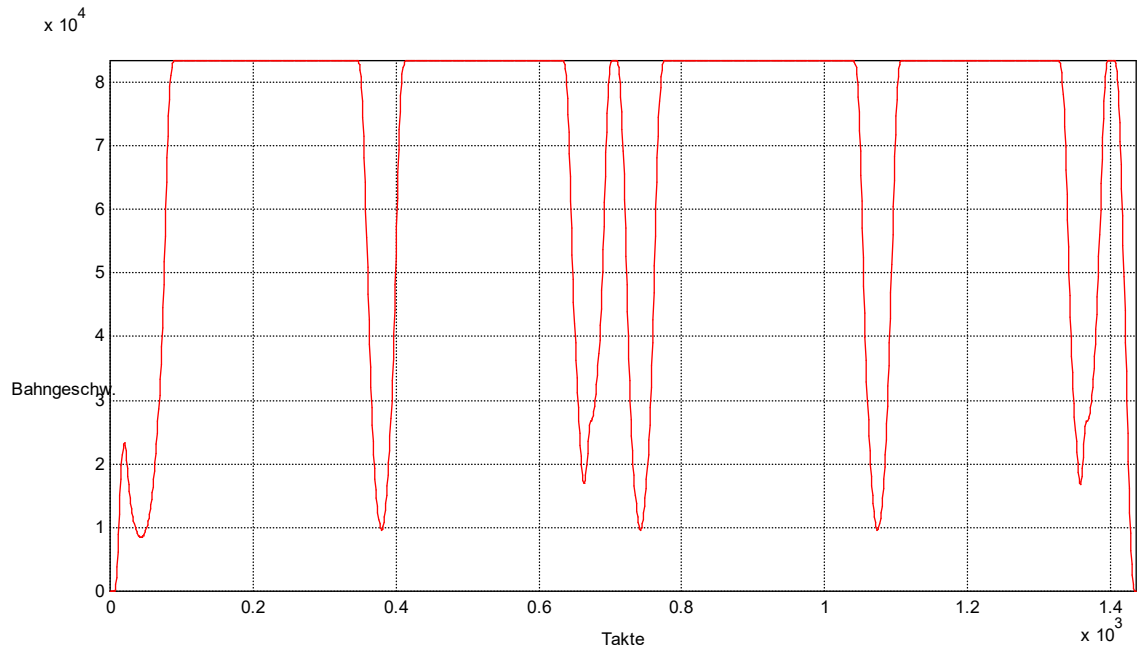**Fig. 5: Time ahead: 2 seconds, block supply is sufficient**

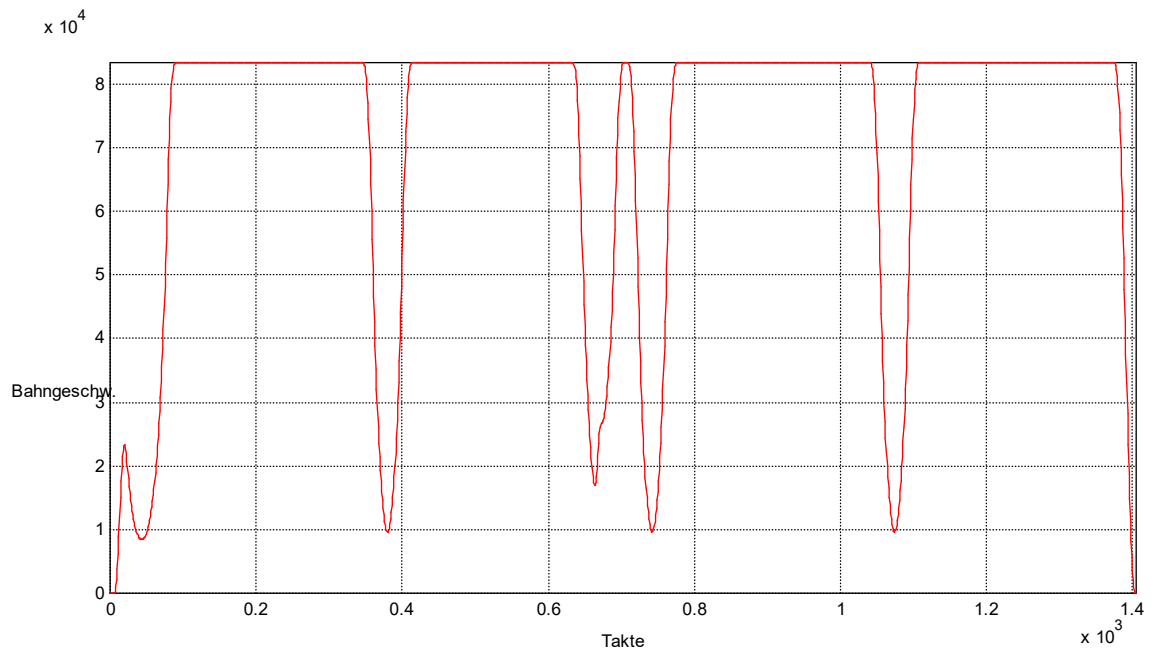**Fig. 6: Time ahead: 0.5 seconds, block supply is sufficient**

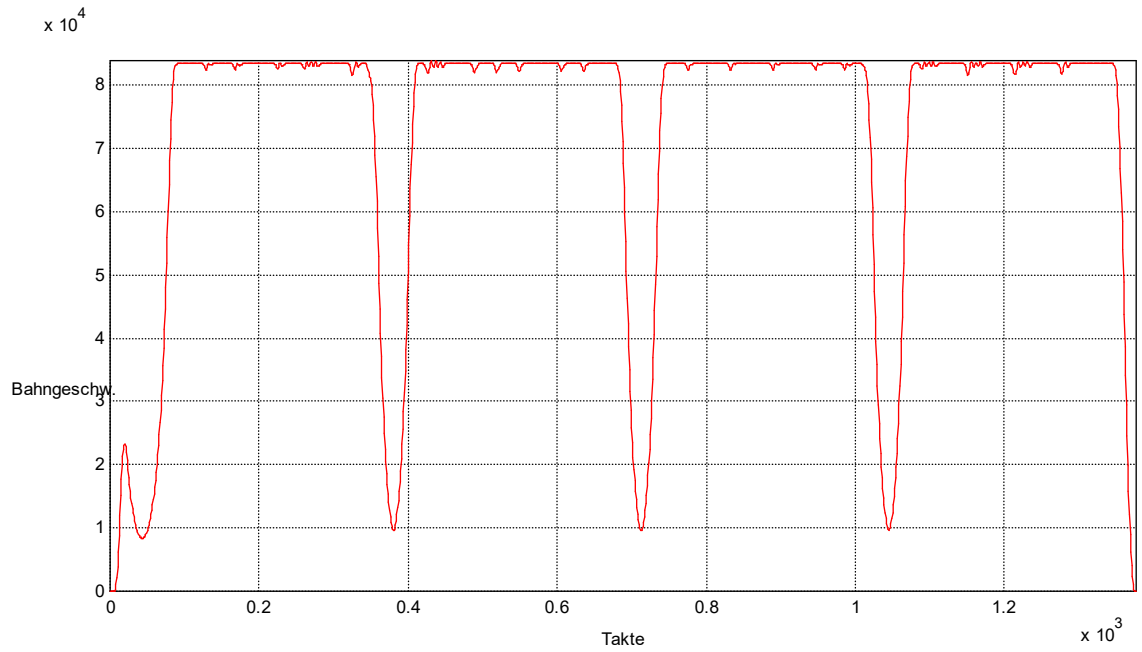**Fig. 7: Time ahead: 0.25 seconds, block supply is sufficient**

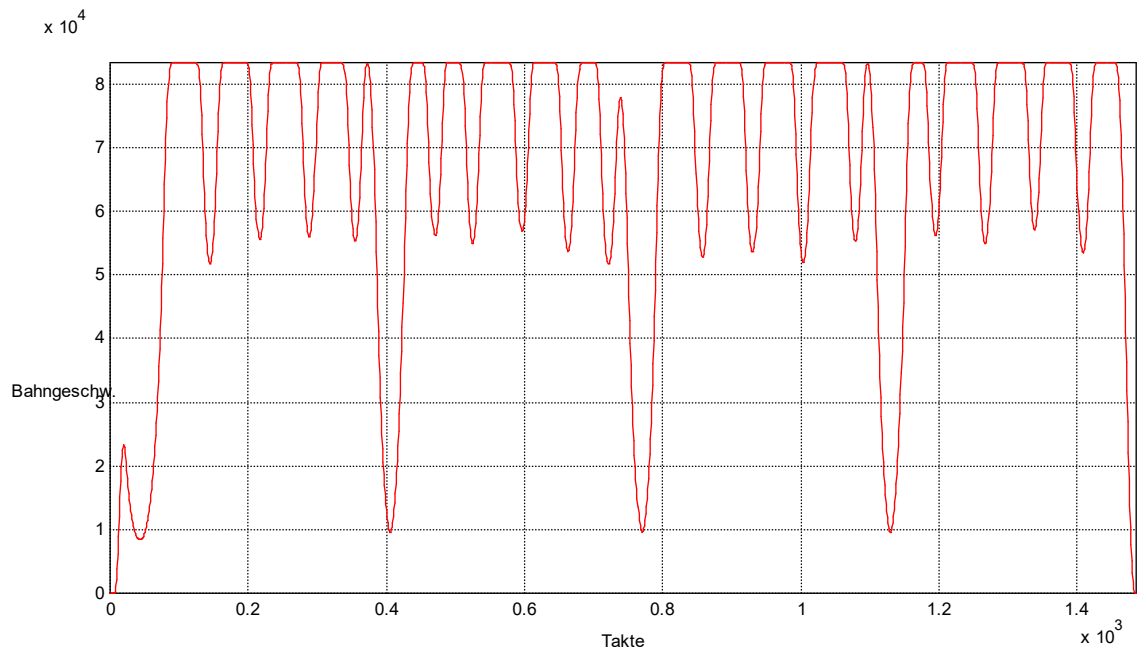**Fig. 8: Time ahead: 0.225 seconds, block supply starts to pulse, path velocity varies**

Fig. 9: Time ahead: 0.2 seconds, block supply pulses more strongly, path velocity varies
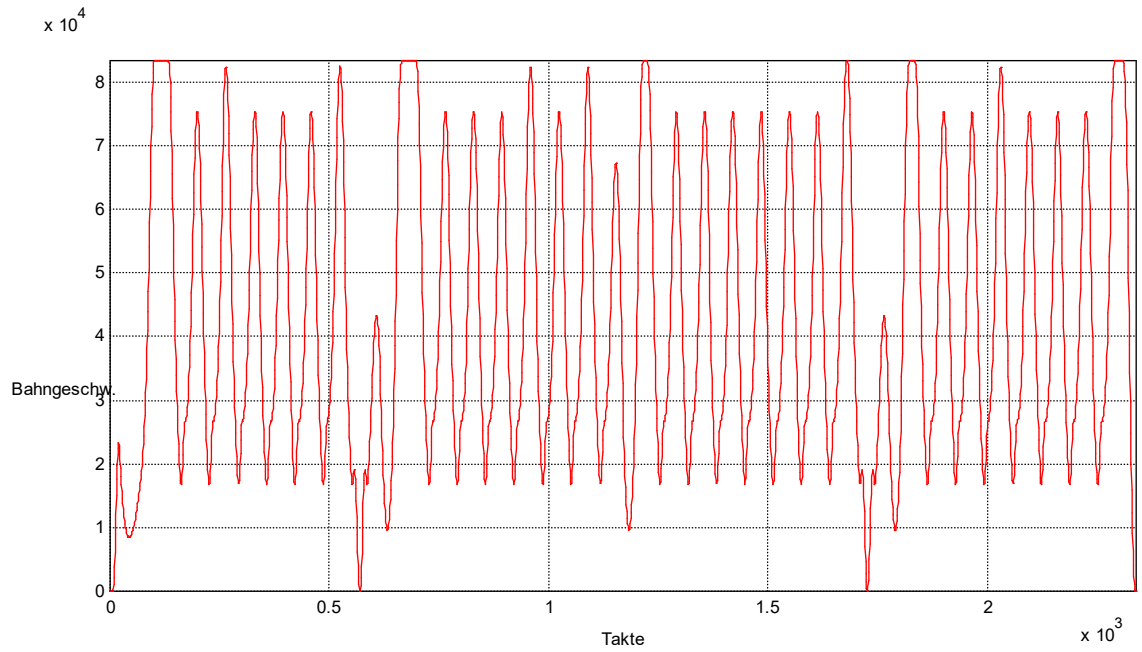
Fig. 10: Time ahead: 0.1 seconds, block supply pulses strongly, path velocity varies

Example 2

In default mode the mean feed velocity is always included in the calculation of time ahead. This function can be deactivated by the channel parameter P-CHAN-00428 (calc_average_feed_ahead) [▸ 35] to run in programs and for diagnostic purposes.

The idealised test program listed below examines and illustrates the influence of the mean feed velocity on the accuracy of the actual time ahead.

Test 1:

Program with 40 mm linear blocks and feed limits, time ahead 2s

```
%average_feed_ahead_1
F60000 G01
V.G.MAX_TIME_AHEAD = 2
#VECTOR LIMIT ON [VEL=20000]
$FOR P1=0,100,1
  G91 X40
$ENDFOR
G91 Y10
#VECTOR LIMIT ON [VEL=40000]
$FOR P1=0,100,1
  G91 X-40
$ENDFOR
G91 Y-10
M30
```
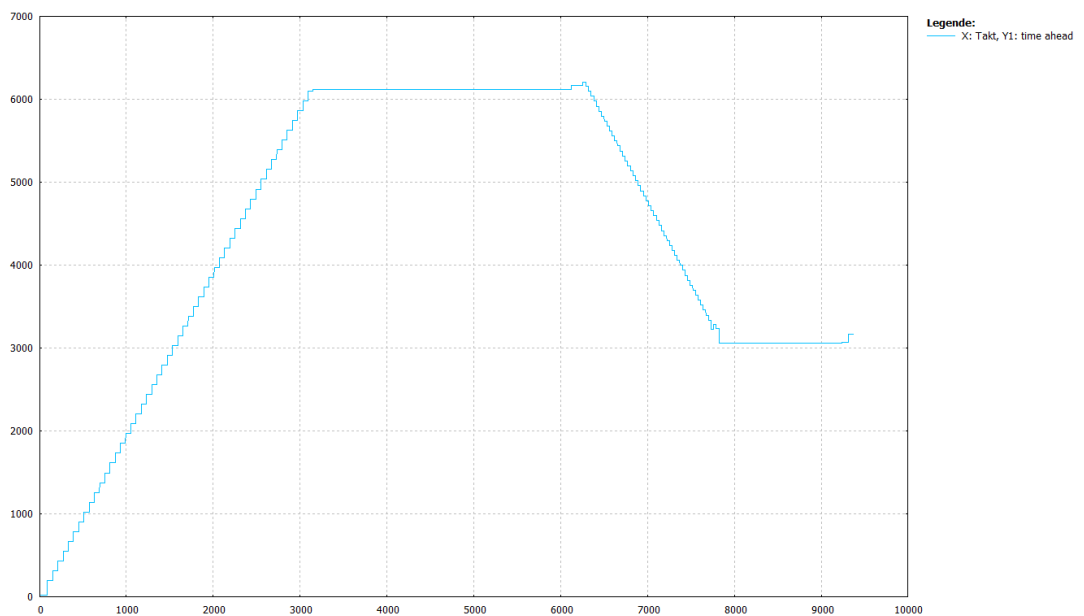

Legende:
X: Takt, Y1: time ahead

Fig. 11: Block ahead limiting with no consideration given to average velocity (P-CHAN-00428 = 0, block lengths 40mm and active block ahead limits)

Block supply is ensured by the long geometry blocks (40mm). The velocity influence by #VECTOR LIMIT is not considered in the time ahead calculation due to P-CHAN-00428 = 0. The required time ahead of 2 s is exceeded by a considerable amount
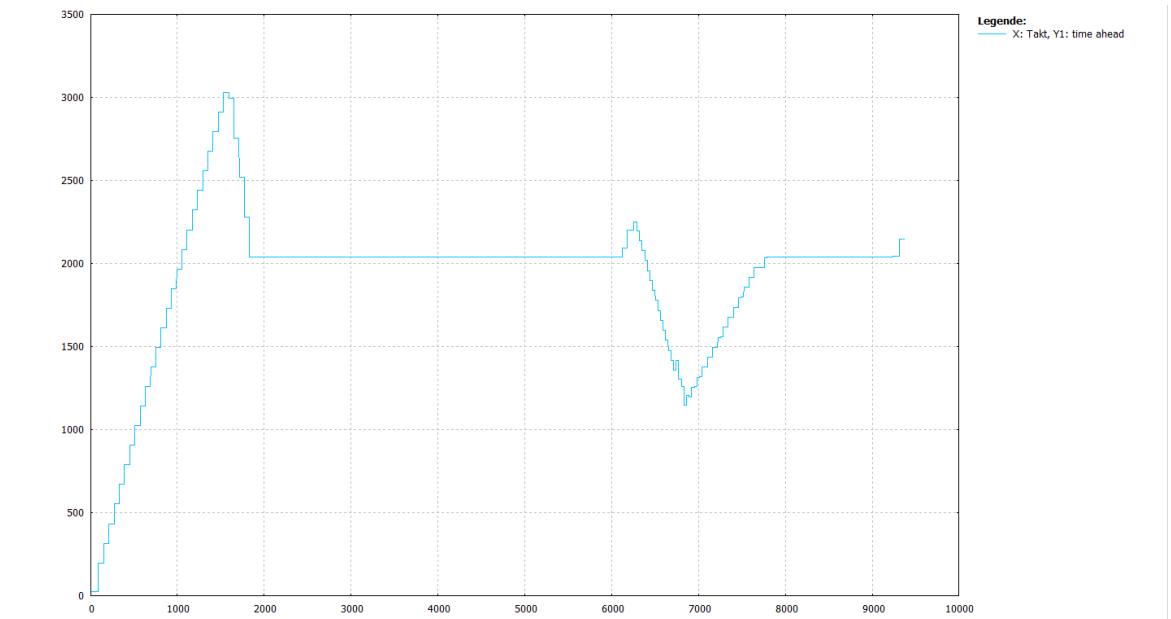
Fig. 12: Block ahead limiting with no consideration given to average velocity (default or P-CHAN-00428 = 1, block lengths 40mm and active feed limits)

At start a slight excess occurs but this is re-adjusted. The required time ahead of 2 s is basically maintained

Test 2:

Program with 10 mm (short) linear blocks and feed limits, time ahead 2s

```
%average_feed_ahead_2
F60000 G01
V.G.MAX_TIME_AHEAD = 2
#VECTOR LIMIT ON [VEL=20000]
$FOR P1=0,400,1
  G91 X10
$ENDFOR
G91 Y10
#VECTOR LIMIT ON [VEL=40000]
$FOR P1=0,400,1
  G91 X-10
$ENDFOR
G91 Y-10
M30
```
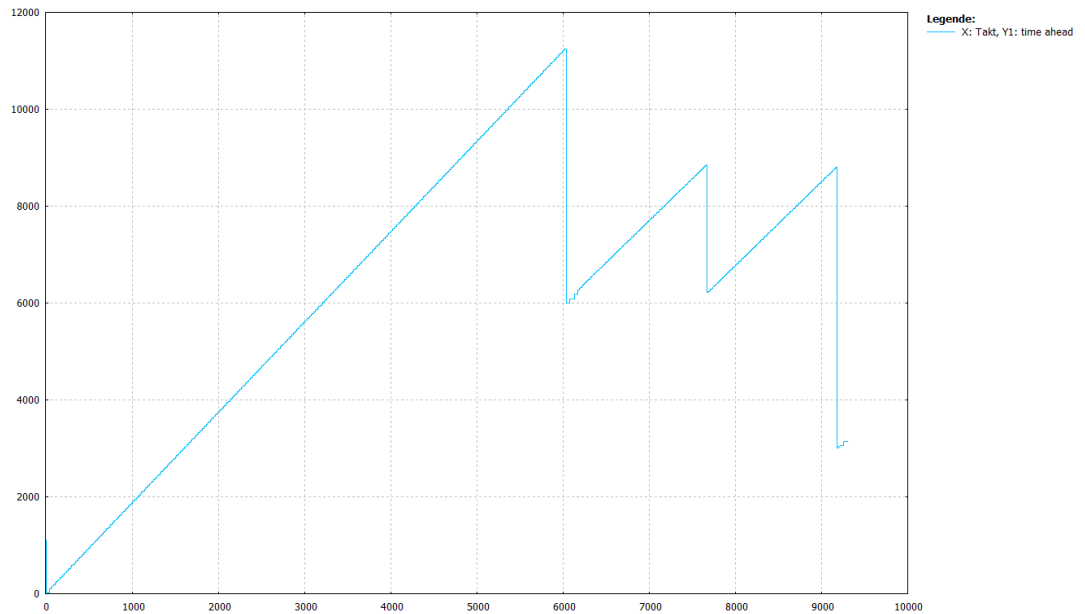
Fig. 13: Block ahead limiting with no consideration given to average velocity (P-CHAN-00428 = 0, block lengths 10mm and active block ahead limits)

Due to the short geometry blocks (10mm) the priority for interpolation is to ensure block supply. Stable block ahead limiting is not achieved. The required time ahead of 2 s is exceeded by a considerable amount.
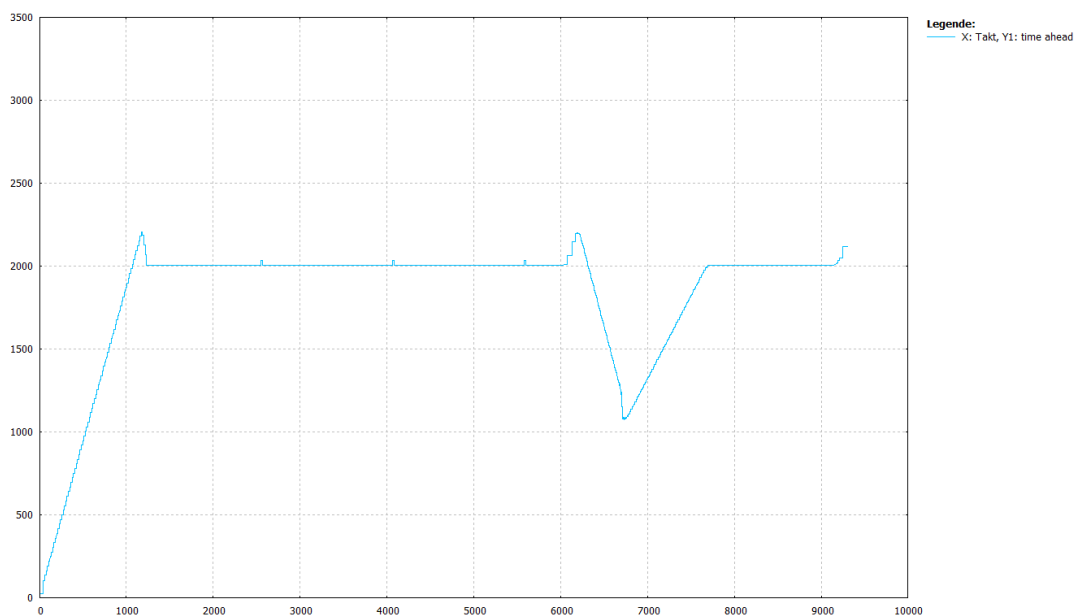

Fig. 14: Block ahead limiting with no consideration given to average velocity (default or P-CHAN-00428 = 1, block lengths 10mm and active feed limits)

At start a slight excess occurs but this is re-adjusted. The required lime ahead of 2 s is basically maintained.

Test 3:

Program with 40mm linear blocks without feed limits, time ahead 2s

```
%average_feed_ahead_3
F60000 G01
V.G.MAX_TIME_AHEAD = 2
$FOR P1=0,100,1
  G91 X40
$ENDFOR
G91 Y10
$FOR P1=0,100,1
  G91 X-40
$ENDFOR
G91 Y-10
M30
```


Fig. 15: Block ahead limiting with consideration given to average velocity (default or P-CHAN-00428 = 1, block lengths 40mm without feed limits)

At start a slight excess occurs but this is re-adjusted. The required time ahead of 2 s is basically maintained.

Test 4:

Program with 10mm (short) linear blocks without feed limits, time ahead 2s

```
%average_feed_ahead_4
F60000 G01
V.G.MAX_TIME_AHEAD = 2
$FOR P1=0,400,1
  G91 X10
$ENDFOR
G91 Y10
$FOR P1=0,400,1
  G91 X-10
$ENDFOR
G91 Y-10
M30
```
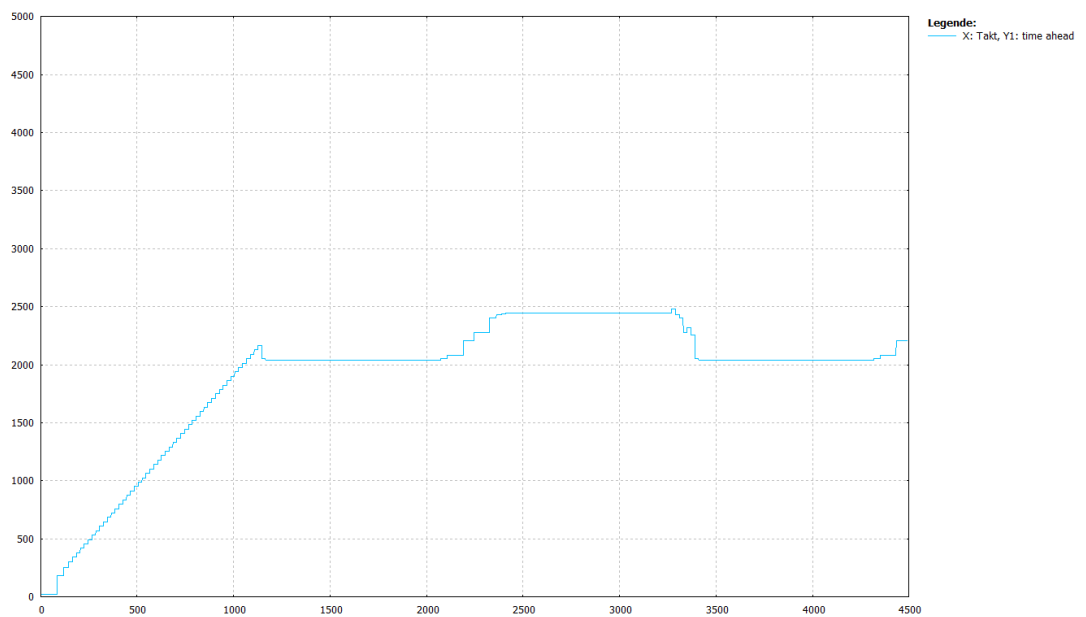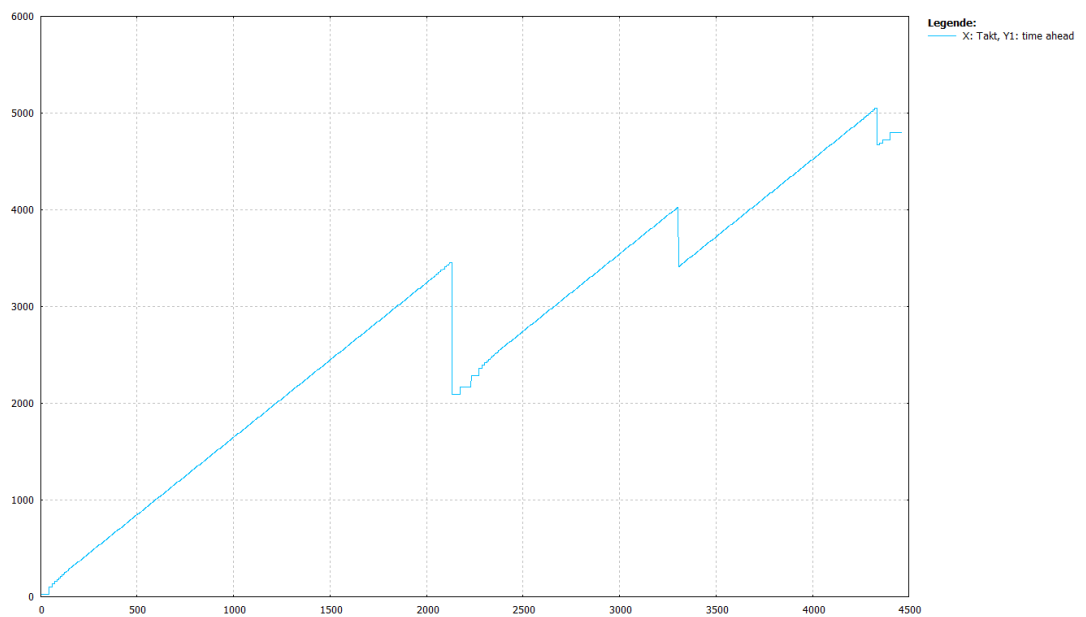

Fig. 16: Block ahead limiting with consideration given to average velocity (default or P-CHAN-00428 = 1, block lengths 10mm without feed limits)

Due to the short geometry blocks (10mm) the priority for interpolation is to ensure block supply. Stable block ahead limiting is not achieved. The required time ahead of 2 s is exceeded by a considerable amount.

Example 3

The graphics below show the curve of time ahead in an actual program. It features strongly fluctuating feeds (rapid traverse and normal feed). The resulting deceleration and acceleration phases cannot be exactly considered in advance when estimating block ahead limiting. This then results in a more or less strong deviation in the programmed time ahead.

Test 1:
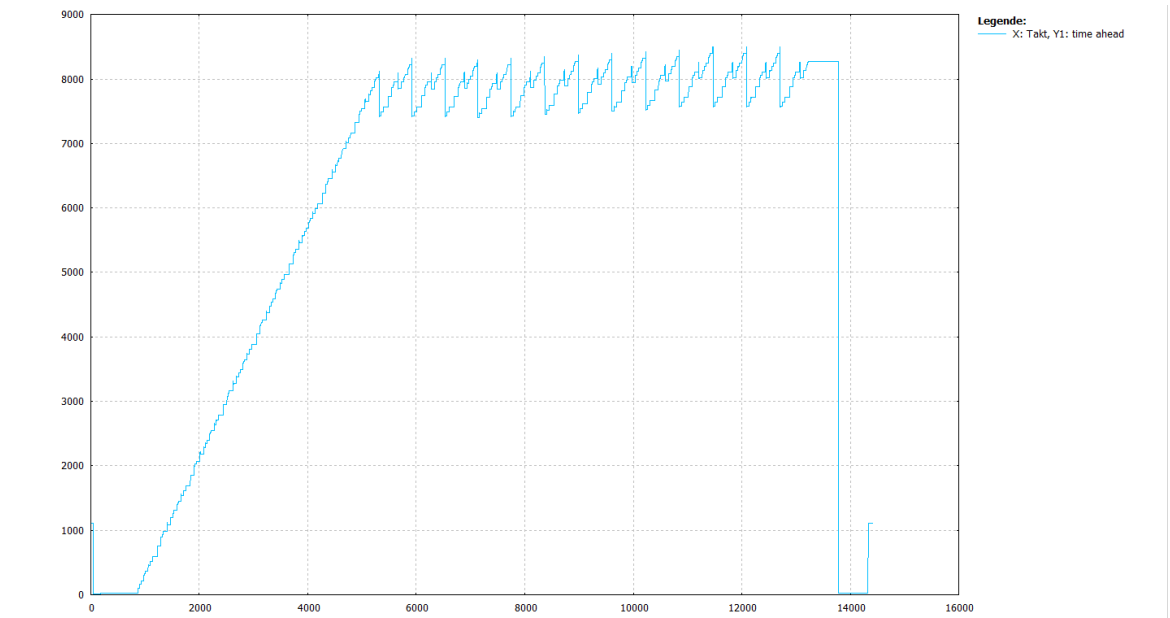

Fig. 17: Block ahead limiting with no consideration given to average velocity (P-CHAN-00428 = 0), block supply is insufficient due to the very high feed, time ahead at 7s to 8s
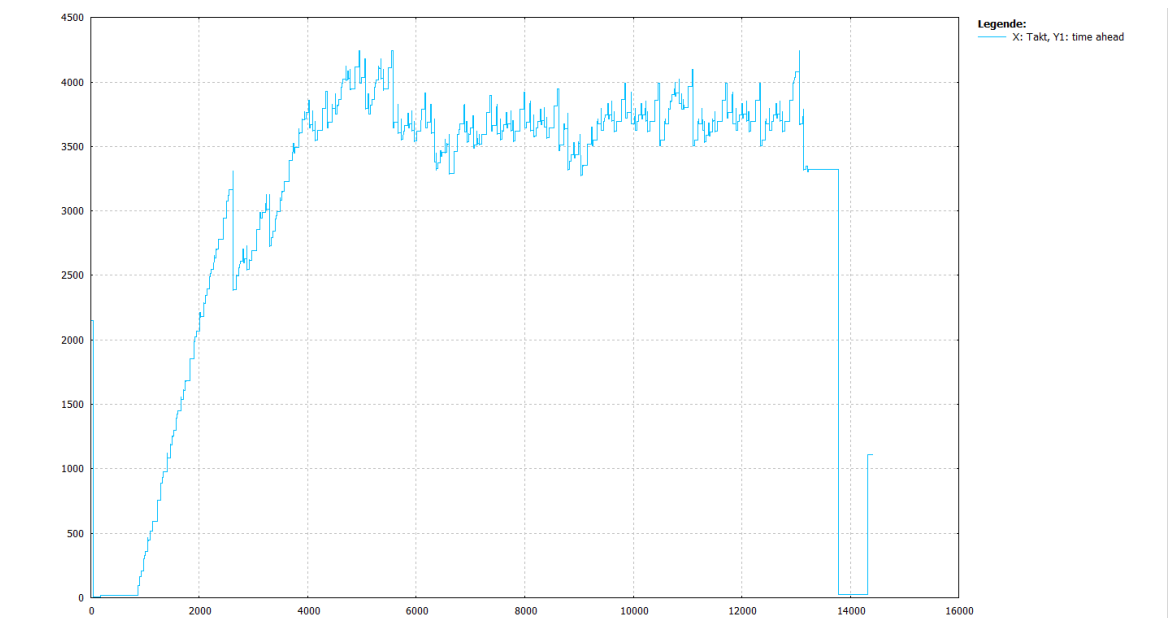
Test 2:

Fig. 18: Block ahead limiting with consideration given to average velocity (default or P-CHAN-00428 = 1), block supply sufficient, system can settle. Much improved time ahead between 3s and 4s

5 Monitored block ahead operation

Effectiveness of block ahead limiting

As far as possible, use of block ahead limiting must not influence the programmed contouring motion in its course and its dynamics. The following priorities apply here:

Priority 1: Block supply and path dynamics

The top priority is to maintain interpolator block supply since this is the basic condition to achieve a stable sequence of the path motion. The programmed path velocity or maintaining path dynamics should not deviate too far from the possible optimal feed rate caused by block ahead limiting

Priority 2: block ahead limiting

Only when the above criteria are met can block ahead limiting attempt to keep to the programmed feed rate value.

Monitored block ahead operation

To ensure this staggered consideration, block ahead operation can take place in a monitored mode.

If the interpolator determines that

 • block supply is no longer sufficient due an active block ahead limit and threatens motion stop,
 • or the programmed path velocity can no longer be maintained within the dynamic limits,

this is also signalled to the decoder in addition to the current block count.

Then, the decoder immediately cancels any active block ahead limit to be able to supply the channel with blocks again, even if the block ahead limit is still exceeded.

Thus, the block ahead limit is considered or kept to only if the interpolator's block supply is sufficient and the programmed path dynamics can be kept to.

Due to these priorities, block supply can never be interrupted and consequently the NC channel path motion can never be stopped.

Mode selection

Monitoring mode is always active by default with **time-related** block ahead limiting.

With **block-related** block ahead limiting, the monitoring mode can be additionally activated by the channel parameter P-CHAN-00270 to ensure downward compatibility:

dec_max_ahead_protected   **NONE | ACTIVE**

Monitoring limits

Although monitoring mode ensures stable execution, the user should nevertheless make sure not to set block ahead limit values which are too low.

It values are too low, there is otherwise a risk that the block supply by the decoder is interrupted too often and is thus requested again by the interpolator. Due to the buffering effect of the NC channel, this may lead to "pulsing" of the block supply and to a high fluctuation in path velocity (see program example in section "Block number limiting via NC command [▶ 13]").

6   Parameter

6.1   Overview

Constant	Description
0x00100000	TIME_AHEAD_LOCK_ACTIVE status display: Decoding in block ahead limiting. Waiting for interpolator.
0x00200000	TIME_AHEAD_LOCK_ACTIVE status display: Decoding in block ahead limiting. Waiting for interpolator.

ID	Parameter	Description
P-CHAN-00216	max_nc_blocks_ahead	Block ahead limiting via (relevant) number of blocks
P-CHAN-00246	max_mo-tion_blocks_ahead	Block ahead limiting via number of motion blocks
P-CHAN-00269	max_time_ahead	Block ahead limiting via motion time
P-CHAN-00270	dec_max_ahead_pro-tected	Block ahead limiting in "monitored mode"
P-CHAN-00428	calc_aver-age_feed_ahead	Block ahead limiting with consideration given to the current mean average velocity

6.2   Description

P-CHAN-00216	**Select decoder block ahead limiting via number of blocks**
Description	Parameter for activating and configuring the decoder block ahead limiting function based on NC blocks by setting a value unequal to zero. This value defines the maximum number of (relevant) NC blocks (rows) by which the decoder block ahead function may be ahead of the interpolator.
Parameter	max_nc_blocks_ahead
Data type	UNS32
Data range	0: No decoder block ahead limiting (default). >0: After program start, decoder block ahead limiting function is immediately active with the number of NC blocks and this function remains selected until main program end M02/M30.
Dimension	----
Default value	0
Remarks	Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30.

P-CHAN-00246	Selecting decoder ahead limiting via number of NC motion blocks
Description	Parameter for activating and configuring the decoder block ahead limiting function based on NC blocks by setting a value unequal to zero. The value defines the maximum number of pure NC motion blocks by which the decoder may be ahead of the interpolator.
Parameter	max_motion_blocks_ahead
Data type	UNS32
Data range	0: No decoder block ahead limiting (default). >0: After program start the decoder block ahead limiting function is immediately active with the defined number of NC motion blocks and remains selected until main program end M02/M30.
Dimension	----
Default value	0
Remarks	Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30.

P-CHAN-00269	Selecting time-based decoder block ahead limiting
Description	This parameter activates and configures the time-based limiting of the decoder ahead function by setting a value unequal to zero. The value defines the maximum time in microseconds by which the decoder may be ahead of the interpolator.
Parameter	max_time_ahead
Data type	REAL64
Data range	0: No decoder block ahead limiting (default). >0: The time-based decoder block ahead limiting function is active after program start at once and remains selected until main program end M30.
Dimension	µs
Default value	0
Remarks	Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30.

P-CHAN-00270	Block ahead limiting in protected mode
Description	To ensure a stable process, the block ahead limiting function can be executed in a special mode. In this mode the function is only effective if the block supply to the interpolator and path dynamics are not jeopardised.
Parameter	dec_max_ahead_protected
Data type	STRING
Data range	NONE: No monitoring of block ahead limiting (default). ACTIVE: Block ahead limiting is monitored. This function is optional for block-based ahead limiting. This function is always ACTIVE when the time-based ahead limiting function is selected.
Dimension	----
Default value	NONE
Remarks	

P-CHAN-00428	Deactivate time ahead calculation based on mean feed velocity
Description	In order to estimate the motion time for block ahead limiting realistically, the interpolation supplies a mean feed velocity (average velocity) of the next motion blocks to be executed. In default mode the mean feed velocity is always included in the calculation of time ahead. This function can be deactivated by this channel parameter to run in programs and for diagnostic purposes.
Parameter	calc_average_feed_ahead
Data type	BOOLEAN
Data range	0: Time ahead calculation without mean feed velocity. 1: Time ahead calculation with mean feed velocity (default).
Dimension	----
Default value	TRUE
Remarks	This parameter is available as of CNC Build V2.11.2808.05 and higher. Only use for tests or diagnostic purposes!

7 Appendix

7.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

---

ℹ	**Notice**
Change options for favourite links in your browser;
Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.
We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

E-Mail: documentation@isg-stuttgart.de

Keyword index

P