TM1111

Integrated Motion Control – Path-controlled movements

PERFECTION IN AUTOMATION
www.br-automation.com

Prerequisites and requirements

Training modules	TM1110 – Integrated Motion Control (Axis Groups)
Software	Automation Studio 4.2 Automation Runtime 4.08 mapp Technology 1.00.0
Hardware	-

Table of contents

1 Introduction

Today, nearly every machine tool uses CNC technology. This technology makes path-controlled movements possible, and is **completely integrated in B&R's control solutions**.

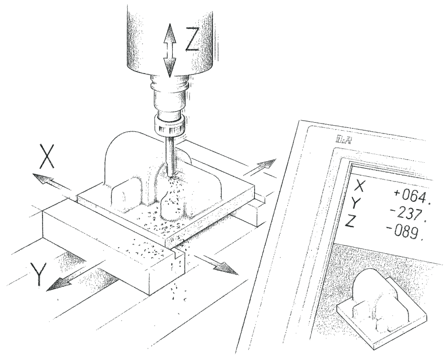


Figure 1: CNC machines with 3 axes

B&R makes it easy to combine CNC machines and machines with conventional drive technology. This allows CNC machines to draw from the entire palette of B&R automation technology.

This training module describes the functionality of B&R's CNC technology and its numerous application options.

The fundamental concepts and procedures involved are described clearly and simply.

1.1 Learning objectives

With the aid of selected examples, you will learn how to perform path-controlled movements using an axis group.

- You will become familiar with the CNC technology at B&R.
- You will learn which machines can be used for path-controlled movements.
- You will find out how path calculation functions.
- You will learn how to easily program path-controlled movements.
- You will learn how CNC programs can be structured using main programs and subprograms.
- You learn the effects of dynamic parameters on a path-controlled movement.
- You will learn how to communicate between CNC programs and the B&R controller.
- You will learn how to work with path-synchronous and interpreter-synchronous events.

2 CNC technology

The term CNC refers to a large group of machines that use computer logic to control the movements of multiple axes simultaneously. CNC stands for "Computer Numerical Control".

Development

Manually operated machines only permit movements parallel to an axis, as the axes have to be operated via handwheels. This places severe restrictions on the ways that a workpiece can be machined.

While the introduction of NC (Numeric Control) did automate this process, it did nothing to alleviate the problem of being limited to parallel movements along each axis.

Not until the first CNC machines were developed in the 1970s could machines perform movements with freely programmable paths. These paths are created as CNC programs using G-codes defined in DIN 66025 and ISO 6983.

Figure 2: Metalworking

Today, very few CNC programs are written by hand, with many machines offering a simple graphical programming interface supported by a CAD system.

Shop Floor Programming (SFP)

A shop floor programming interface supports operators with input masks custom-tailored to the particular machine or process they are working with. The range of functions relies heavily on the respective technology and the functions offered by the machine itself. The CNC program works in the background, concealed from the operator, functioning as a universal interface to the CNC machine.

Distributed Numerical Control (DNC)

DNC takes things another step further and isolates the CNC programming process from the machine entirely. The CNC machine is integrated in a network and the CNC program is created on another computer, which then transfers it to the CNC machine.

3 Path-controlled movements

Path-controlled movements are the foundation for complex, high-precision manufacturing. Generic Motion Control (GMC) allows them to be seamlessly integrated into the machine application. Standardized PLCopen function blocks make it easy to program path-controlled movements. The range of potential applications is extremely diverse, reaching from CNC machines to robotic applications with high-speed movement sequences such as pick-and-place tasks.

2D path control

2D path control allows two axes to be controlled on a fixed plane. Laser cutting machines are a practical example of this.

$2^1/_2$D path control

$2^1/_2$D path control allows up to two axes to be controlled on different planes. These variants can be used in simple milling machines where the third axis is always stationary during movement.

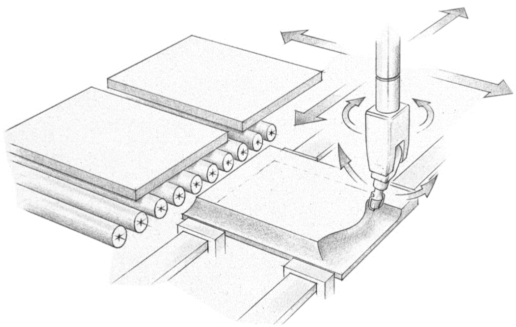


Figure 3: Five-axis milling machines

3D path control

3D path control allows three or more axes to be controlled simultaneously in three dimensional space. However, the direction in three-dimensional space (orientation) can also be controlled in addition to the position in three-dimensional space.

The programmed path is generally composed of straight lines and arcs. Straight segments are defined by setting a new end position. Arcs require additional information about the position of the center of the circle and its radius.

Corrections

In addition to the task of interpolation, CNC machines can also calculate corrective and compensatory movements. These can be used to handle various tasks automatically.

- **Constant cutting speed**
  Many processes require the tool to move at a constant speed, regardless of the direction in which it is moving. A practical example of this is the application of glue to an object.
- **Coordinate transformations**
  Using transformations, the same CNC program can be run at different positions, for example to apply the same labeling at different positions on a tool.
- **Cutter diameter compensation**
  If different tools are used, it is much easier to make adjustments using compensation functions than by having to laboriously modify the CNC program. For example, these functions can automatically compensate for cutter diameter or the rounded tip of a parting tool.

4 Application of CNC programs

A CNC program controls the processing performed by a CNC machine. It contains data in ASCII format that describes the path to be traversed.

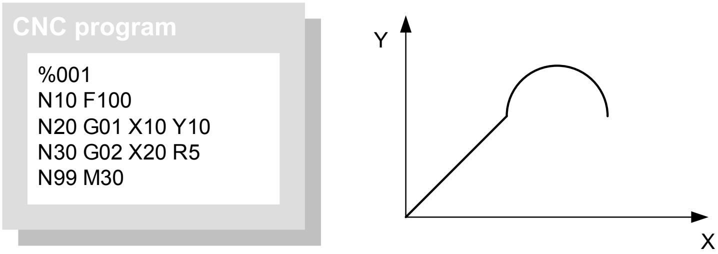


Figure 4: CNC program

4.1 Storage of CNC programs

The CNC program can be created in Automation Studio and transferred to the controller as a **data object**. Alternatively, the program can be stored as a **file**. In this case, the CNC program is not transferred with the project via Automation Studio but is instead loaded from the file system at runtime. Any storage location can be defined for the file. It is possible store the program in the controller's memory (CompactFlash), on a USB data carrier or on the network.

4.2 Programming

A CNC program contains instructions describing the geometry and dynamics of the path to be traversed. The instructions are standardized in the standards DIN66025 and ISO 6983.

> Motion \ Reference manual \ ARNC0 \ CNC programming instructions

Commands

The path is specified with a G-code. These enable movements such as linear and circular interpolation, coordinate system shifting, tool radian correction and much more.

> Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ G-codes

Technical parameters

In addition to the G functions, a CNC program also contains some important technical parameters. One of the most important parameters is the feed, which is specified as an F-word. This parameter describes the velocity of the path movements, and is the vector sum of the velocity of the individual axes in a path-controlled movement (see "Feed rate" on page 17).

> Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ technology parameters

Program design

Programming elements such as comments and block numbers can help you design a CNC program that is clearly organized and easy to read.

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ Program design

Operators

Use of operators and functions enables complex calculations to be performed in a CNC program.

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ Operators

Control blocks

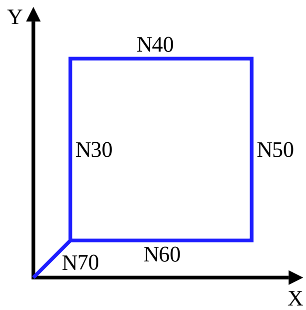
To enable efficient and flexible programming, control blocks such as loops and branching can be applied in CNC programs.

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ control blocks

Exercise:

1 Switch on and home the axis group with MpCnc2Axis
2 Create a simple CNC program and transfer it to the controller

```
%001 (Square)
N10 F1000 (Set the Feedrate)
N20 G01 X10 Y10 (Move to first point)
N30 G01 Y50 (Draw the first line ..)
N40 G01 X50
N50 G01 Y10
N60 G01 X10
N70 G01 X0 Y0 (Return to position zero)
N99 M30(End of program)
```



3 Run the CNC program using the MpCnc2Axis mapp component
4 Control the position progression of the axis during movement

5 Axis types and axis properties

An axis group can be comprised of a variety of different types of axes. In addition to path-controlled axes, i.e. path axes and auxiliary axes, there are also non-path-controlled axes.

With non-path-controlled axes, it is not possible to perform a path-controlled movement, so they will not be discussed here in any further detail. This topic was handled in more depth in TM1110 – Integrated Motion Control (Axis Groups) .

The difference between path axes and auxiliary axes is that auxiliary axes usually lack a geometric link to the other axes.

Path axes, on the other hand, generally form a Cartesian coordinate system or are linked to one another via other geometric relationships.

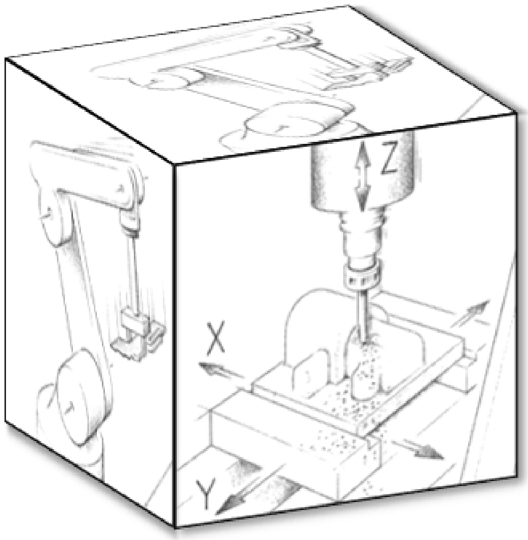


Figure 5: Generic Motion Control

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis configuration

6 Path calculation

Path calculation is the core functionality for a path-controlled movement. It generates cyclic position setpoints for the axis group's path-controlled axes from the movement command. Path calculation is performed in steps and can be broken down into several function units.

6.1 Function units in path calculation

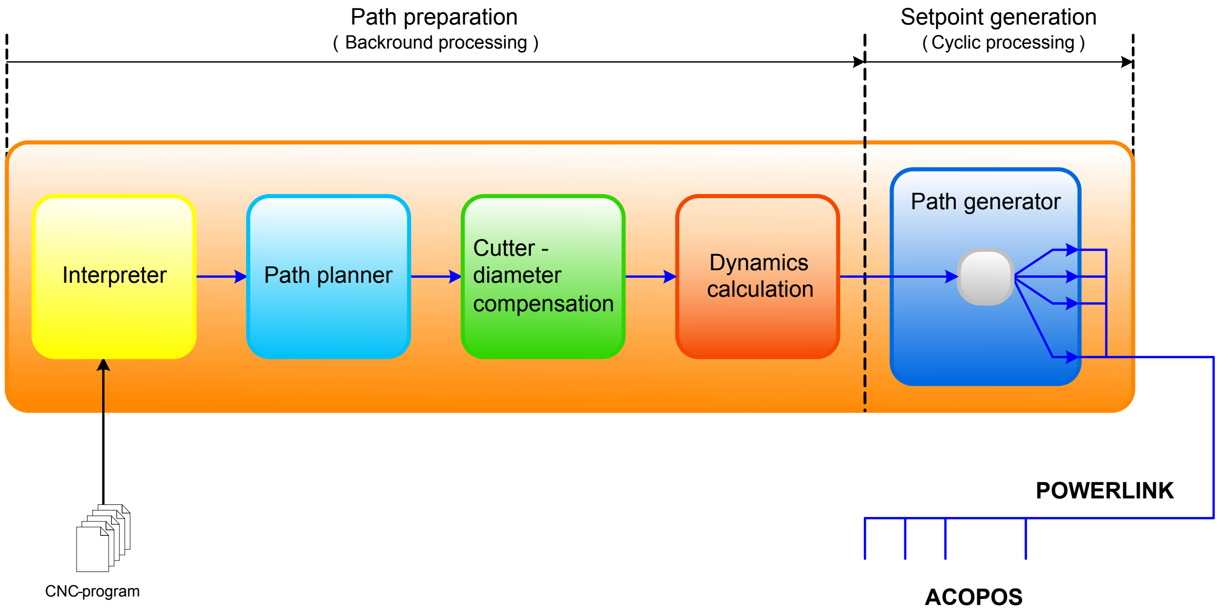


Figure 6: Path calculation in an axis group

function unit	Task
Interpreter	Interprets the CNC program
Path planner	The geometric path specified by the CNC program is generated
Cutter diameter compensation	Adjustment of the geometric path generated by the path planner around the radius of the tool
Dynamics calculation	Calculation of the path speed profile
Path generator	Generation of cyclic setpoints for individual axes

All function units that process data leading up to the path generator are referred to as "background processing" for the path calculation. Their task is to prepare the path. They only run during the CPU idle time (no cyclic tasks are active).

The speed of path preparation therefore depends on how heavily the CPU resources are being utilized by cyclic tasks.

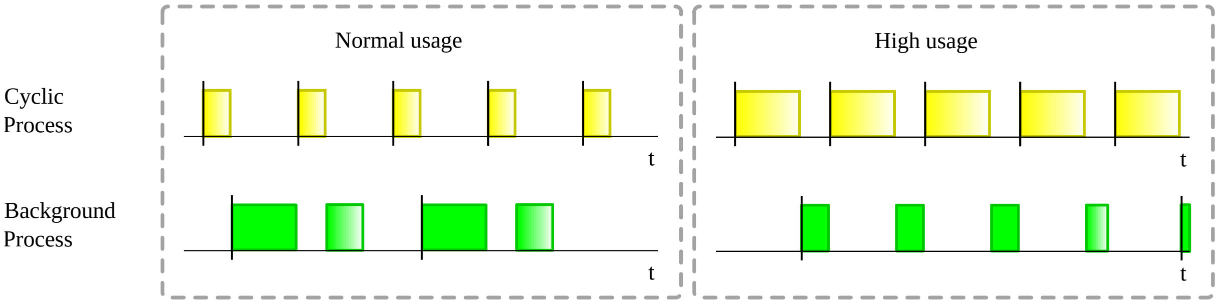


Figure 7: Effect of CPU usage on background processing

If the path generator doesn't receive any path sections (CPU load too high), path movement will come to a stop.

Movement will continue as soon as new path sections become available to the path generator.

Interpreter

The interpreter reads CNC programs and compiles the G-code they contain into a symbolic representation ("Motion Packets") for further processing in the system.

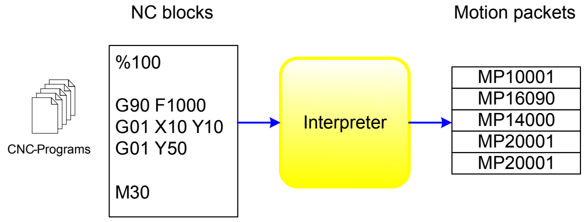


Figure 8: Interpreter

Path planner

The path planner generates a geometric representation of the programmed path movements.

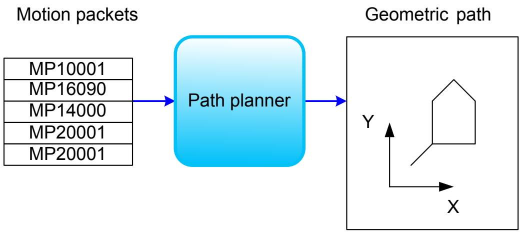


Figure 9: Path planner

Cutter diameter compensation

The cutter diameter compensation function interprets the geometric representation as the outer contour of a workpiece and uses the tool dimension data provided by the user to create a new geometric representation of the programmed path movements.

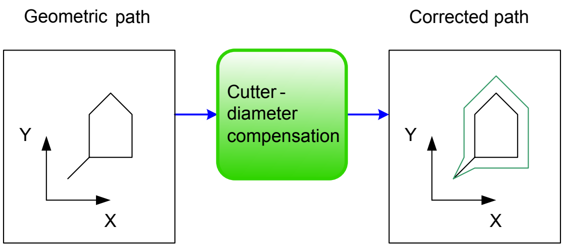


Figure 10: Cutter diameter compensation

Dynamics calculation

The dynamics calculation creates a velocity profile to match the geometric description of the path movement.

This takes into consideration the limit values defined by the user (velocity, acceleration, jolt) for the axes and the path. For more information, see section Dynamics calculation.

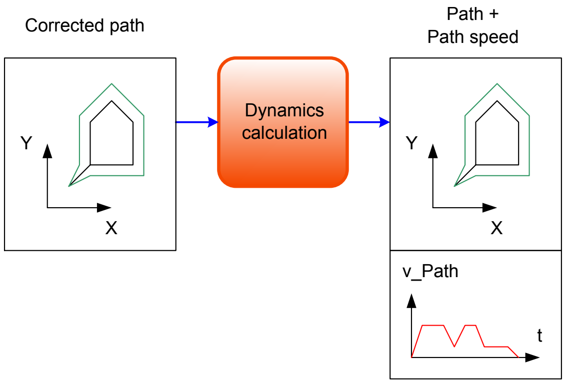


Figure 11: Dynamics calculation

Path generator

The path generator represents the "cyclic processing" (real-time processing) functional unit of path generation. Its task is to generate and output the setpoints for the axes involved.

To do this, it uses the data it receives from the "background processing" functional units of path generation.

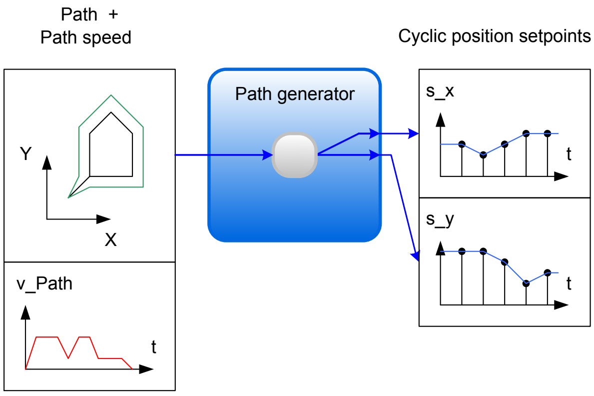


Figure 12: Path generator

6.2 Setpoint generation

Path-controlled movement

For a path-controlled movement, however, the position setpoints are generated by the controller's path generator and sent to the drive as cyclic position setpoints.

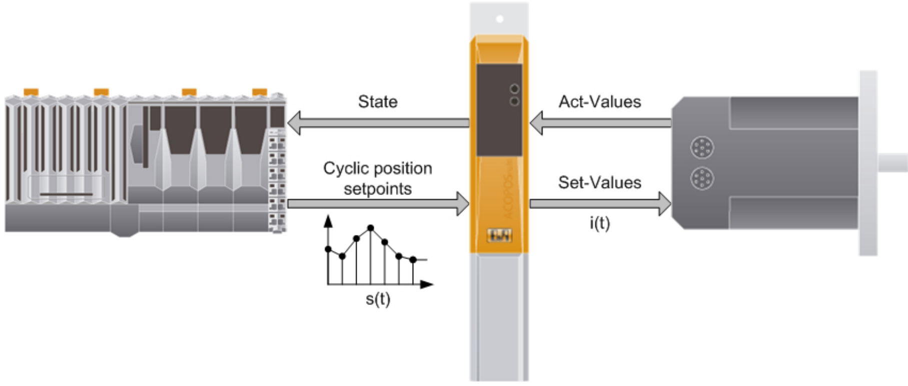


Figure 13: Path-controlled movement – Cyclic setpoint transfer

> The shorter the interval selected between position setpoints, the greater the precision with which the drive can execute the path-controlled movement. However, this places high demands on the controller's necessary computing power and the network speed.

Single-axis movement

Unlike the cyclic preset positions, for solely single-axis movements, parameters describing the movement profile (position, speed, acceleration) are transferred from the controller to the drive.

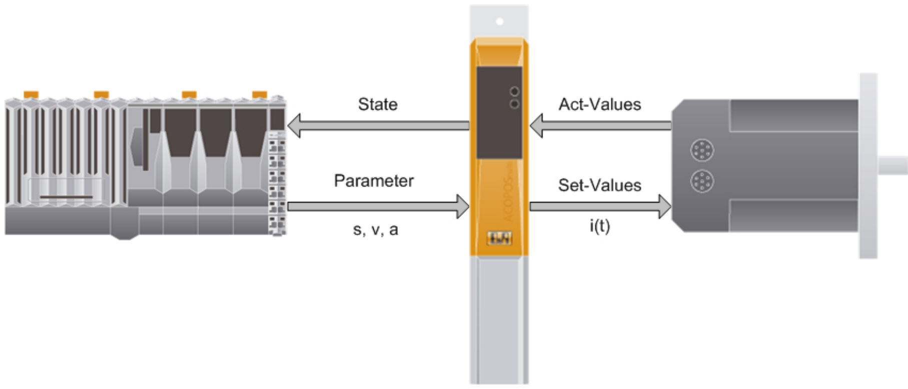


Figure 14: Single-axis movement – Parameter transfer

The profile generator on the drive then generates the setpoints for the position controller.

7 Application programs

The application program for path-controlled movements is an automatic sequence for controlling the CNC machine.

It is important to call the function blocks in a consistent and clearly organized manner. Errors must also be taken into consideration and responded to appropriately.

In high-level programming languages, a state machine (step sequencer) should be used to establish a clearly structured sequential process.

Figure 15: A more in-depth look at the application program

For an overview of available function blocks, see:

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - Controlling a CNC system \ Function blocks

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Function blocks

Figure 16: PLCopen @ B&R

Requirements for a path-controlled movement

Before a path-controlled movement can be started, the axis group must first be prepared.

- The controllers of the axes in the axis group must be switched on
- The axes of the axis group must be homed

The programming tips in TM1110 – Integrated Motion Control (Axis Groups) can help you get the axis group prepared like this.

Exercise: Interrupting and continuing the program

1 Run a CNC program and monitor the course of movement.
2 Interrupt the CNC program with "Interrupt".
3 Check the outputs of the "info" structure
4 Continue the CNC program with "Continue"

8 CNC subprograms

Using subprograms helps create more clearly structured code and increases the re-usability of the CNC program.

Local subprograms

Local subprograms can only be called by the main program and other local subprograms.
They are located in the same file as the main program.

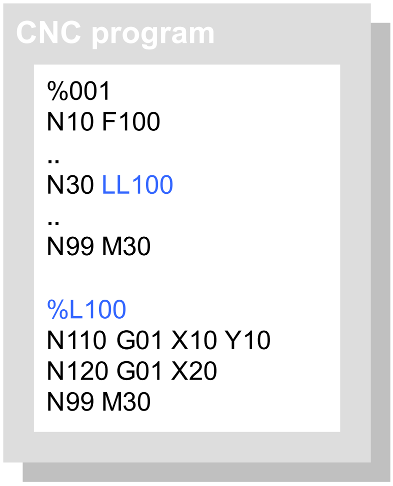
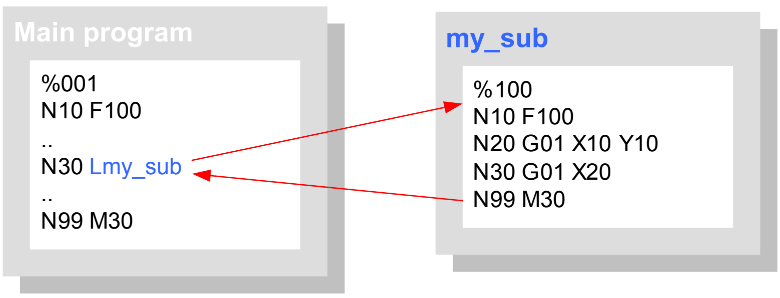


Figure 17: Local subprogram

Global subprograms



A global subprogram is a main program that is called from within another main program.

Figure 18: Global subprogram

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ Main program / subprogram

Exercise:

A repeated contour should be performed several times using a subprogram at various positions on the tool.

1. Create a local subprogram in a main program, which performs a rectangle relative to the current position.
2. Move to various absolute points in the main program which should form the rectangle by opening the subprogram.
3. Run the main program and check the movement.

9 Dynamics calculation

Extreme demands with regard to dynamic properties are placed on CNC machines. They must provide features such as constant feed rate, maximum repeat precision and high availability while withstanding great mechanical stress. Meeting these demands requires a number of mechanical engineering challenges to be mastered.

If the machine is structurally able to handle highly dynamic movements, it is up to the controller to make the most of this potential in order to produce as efficiently as possible.

This can only be achieved when a CNC machine is constantly operated at its limits. Successfully configuring and maintaining these limits is fundamental requirement for a successful path-controlled movement.

By automatically limiting maximum speeds, the machine can constantly be operated at its limits. If the requirements call for a constant feed forward speed, however, then the generated speed profile needs to be adjusted accordingly to this parameter.

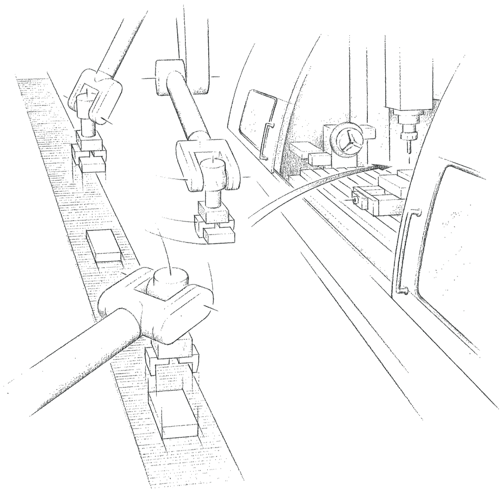


Figure 19: Dynamics calculation

9.1 Feed rate

The feed rate is is one of the most important process parameters involved in a path-controlled movement. It defines the speed of the path in units per minute.

By default, the path axes are involved in the feed rate calculation.

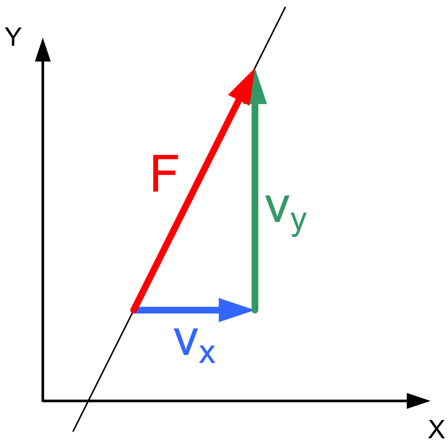


Figure 20: Feed vectors of an XY movement

The feed rate of a path-controlled movement is the vector sum of the speeds of the individual axes:

$$F = \sqrt{v_x^2 + v_y^2}$$

The feed forward is a modal setting and takes effect starting with the line in the CNC program in which it is programmed and ending with the line in which it is changed.

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ Technology parameters \ Feed

9.2 Limit values

A path-controlled movement is always restricted to the limit values of the axis group. These limits can be configured using the axis group parameters or may result from the limits of the individual path axes.

A path-controlled movement is limited by the following values:

- Maximum speed and acceleration values of the axes
- Maximum path speed
- Maximum path acceleration

Motion \ Reference manual \ ARNC0 \ Reference for the axis controller software \ NC objects \ CNC system \ Limit values

Limiting factors for the feed forward

The following example demonstrates that the feed forward speed to be achieved can be influenced by various limits.

CNC program

```
%001
N1 G01 X200 F1000
N2 G01 Y100 F1000
N3 G01 X0 F10000
M30
```

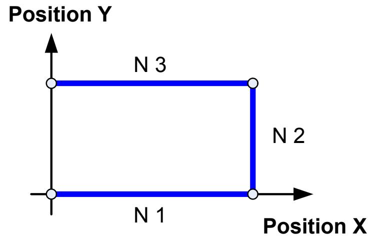


Figure 21: Example contour

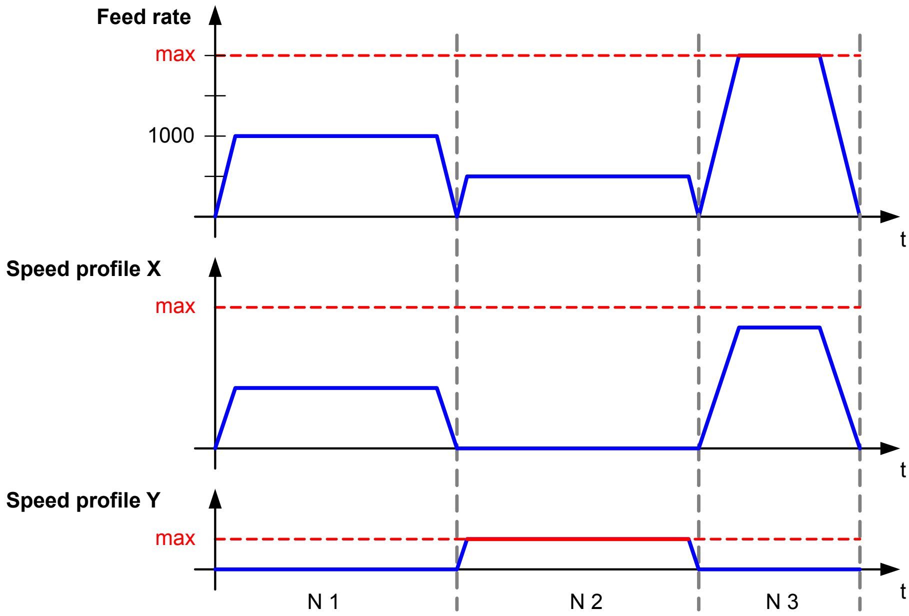


Figure 22: Speed profile

Movement N1 is performed with the feed forward F1000 defined in the CNC program, as neither the maximum path speed nor the axis speed is exceeded.

Movement N2 is performed with axis Y. The maximum speed here is lower than the speed on the X axis. The feed forward is restricted so that this maximum speed is not lost. The programmed feed forward F1000 is not achieved.

Movement N3 is performed with programmed feed forward F10000. In this case, the feed forward is restricted by the path's maximum configured speed, restricted to F2000.

Exercise:

Run a CNC program and check the feed forward calculation formula. Check the impact of the speed limit values on the feed forward.

1. Execute the following CNC program and record the path speed and the individual axis speeds in NC Test

```
%001
N1 G00 X0 Y0
N2 G01 X100 Y100 F600
M30
```

2. Check the recorded path speed using the feed forward formula
3. Modify the maximum path speed (limit.v) to that it acts as a limit
4. Repeat steps 1 and 2 with the new limit value

9.3 Path section transitions

A CNC program consists of individual blocks. If these blocks are positioning commands (G00, G01, G02, G03), then each of these positioning commands corresponds to a separate path section. Where two path sections meet there is a transition.

There are two main kinds of path section transition:

- Tangential transitions
- Non-tangential transitions

Motion \ Reference manual \ ARNC0 \ Reference for the axis controller software \ NC objects \ CNC system \ Limit values \ Path section transitions

Depending on the shape of the adjoining path sections, jumps in acceleration and speed may occur.

Velocity jumps - non-tangential path section transition

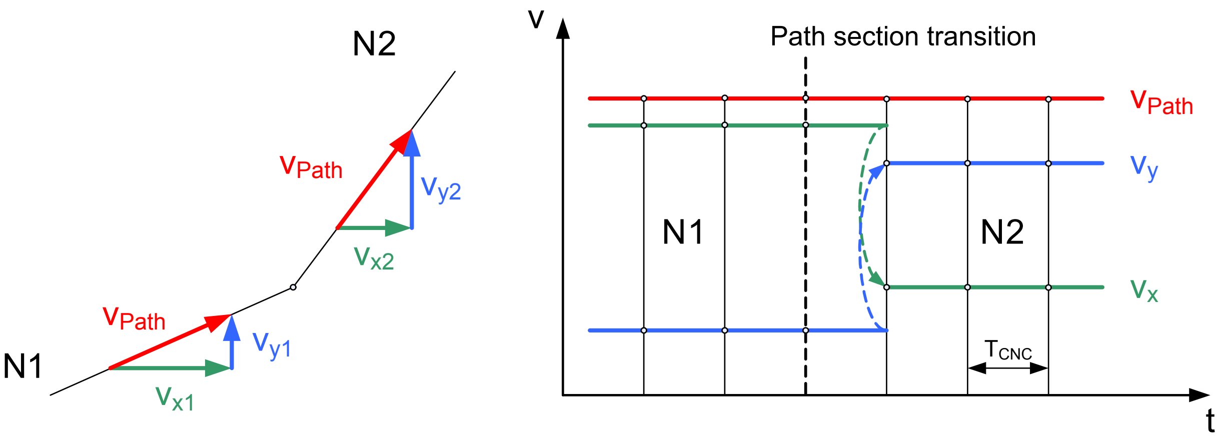


Figure 23: Axis velocity jumps during non-tangential path section transition

Velocity reduction - non-tangential path section transition

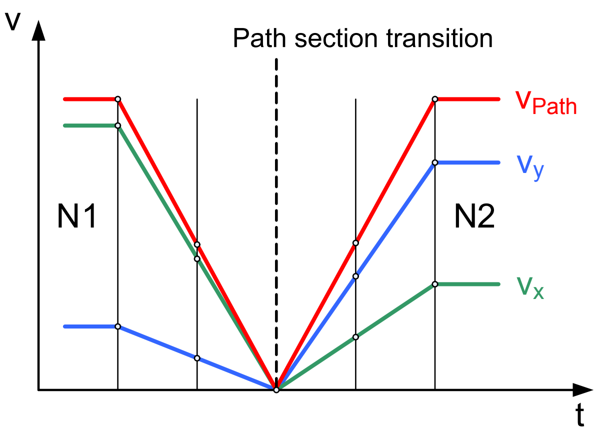


Figure 24: Path velocity reduction during non-tangential path section transition

A jump in velocity or acceleration cannot be handled physically.

The path velocity is therefore reduced to zero by default.

This is important in order to maintain **path precision** at all times.

If the main priority is executing the CNC program as quickly as possible and path precision at the transitions is less important, then jumps in speed and acceleration may be permitted.

These jumps are restricted to the limits of the individual axes and results in a rounding of the path section transitions due to the cyclic transmission of position setpoints.

Keep in mind, however, that extreme changes in velocity and acceleration cause substantial mechanical stress and may cause undesirable oscillations.

This transition rounding within the axis limits is utilized to considerably reduce program execution time. Corners can be rounded at a constant velocity without exceeding production tolerances.

The path velocity and the individual axis positions can be filtered to perform time-optimized movements without creating excessive mechanical stress.

Motion \ Reference manual \ ARNC0 \ Reference for the axis controller software \ NC objects \ CNC system \ Limit values \ Path section transitions \ Parameters

If certain path section transitions should not be rounded, it is possible to selectively apply an accuracy hold (G60) to the respective blocks in the CNC program.

Commands can also be used to influence the programmed path geometrically to meet defined precision criteria.

For more information, see:

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ G-codes

Selection of useful G-codes

- G25: Tangential transition arcs
- G60: Accuracy hold
- G108/G109/G110: Define path acceleration / deceleration
- G126: Rounding edges

10 CNC-PLC communication

There is a clear trend towards modularization in mechanical engineering and software programming. This considerably increases efficiency and cost effectiveness. For programming, the individual software modules must work together smoothly.

Communication between the CNC program and the machine is a key topic these days. Previously only the coolant or lighting were controlled from the CNC program, but today's solutions require signals to be sent to automated pick-and-place machines, tool changers and conveyor belts.

Interaction between path-controlled movements and PLC systems can take on various forms. It is essential to choose the right type of communication for the job at hand.

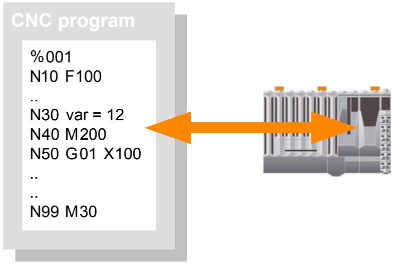


Figure 25: CNC program - PLC interaction

Communication may take place using M-codes or by directly accessing variables in the application program.

10.1 M-codes

M-codes are path-synchronous events which are sent from the CNC program to the application program. They are programmed in the CNC program and their current value can be read in the controller using PLCopen function blocks.

In addition to predefined M-codes, M-codes are generally tailored to a specific machine and can be used for a wide variety of tasks. They are addressed in the CNC program using a unique code.

Example:

```
%001 (M-Function)
N10 F1000 (Set the Feedrate)
N20 G01 X10 Y10 (Move to point)
N30 M200 (Open gripper)
N40 G01 X0 Y0 (Return to zero)
N99 M30(End of program)
```

M200 is set in the application program as soon as the X and Y axes reach the programmed position 10.

M-codes must be reset by the application program on the PLC. There are two ways that an M-code can be configured:

- Blocking (Synchronized M-codes)
- Non-blocking (non-synchronized M-codes)

> ? Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ M-codes

Blocking M-codes

As the name suggests, blocking M-codes have a blocking effect. The CNC program is halted until the M-code is reset by the PLCopen function block.

They are used in cases where the CNC program must wait until a task has been successfully completed before continuing on.

Areas of application

- Changing tools
- Controlling grippers
- Operating clamps

Non-blocking M-codes

Non-blocking M-codes have no impact on the processing of the CNC program. They are used in cases where a path-synchronous action should be triggered in the application program, assuming the action is not time-critical or must not pause the CNC program.

Areas of application

- Operating valves (coolant)
- Activating exhaust/extraction/vacuum units
- Switching on laser/plasma/water jet

Application program for M-codes

For every M-code used, a separate MC_BR_MFunction() function block should be added to the application program, which can be used to read and reset the M-code.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Function blocks \ MC_BR_MFunction

Exercise:

Creating a CNC program for milling machine. M-codes should be used to activate the milling spindle and control the coolant valve.

1. Add two MC_BR_MFunction() function blocks to the program.
2. Enter the following CNC program with the blocking M-code M50 and the non-blocking M-code M40 into the CNC program.

```
%001
N10 F1000 (Set the Feedrate)
N20 G00 X0 Y0 (Move to start)
N30 M50 (Switch on spindle)
N40 G01 X10 Y10 (Move to start of milling)
N50 M40 (Activate cooling)
N60 G01 X100 (Start milling)
N99 M30(End of program)
```

3. Run the CNC program and check the M-codes with the function blocks.
4. Reset the blocking M-code to resume the CNC program.

10.2 Interpreter-synchronous versus path-synchronous

Optimum timing of communication between the CNC program and the PLC is an important topic.

Due to the way that information is stored and processed in a CNC program, it is not sufficient to look at the blocks of a CNC program individually.

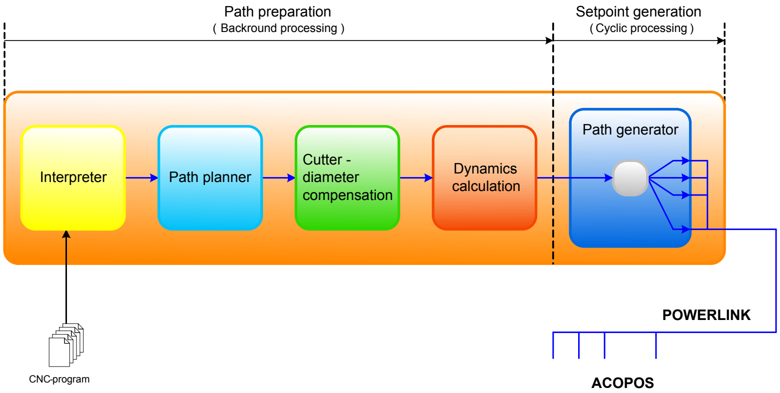


Figure 26: Path calculation in an axis group

You need to know the subsequent blocks in a CNC program in order to correctly implement functions such as cutter diameter compensation or edge rounding, or even simply to ensure that the program is executed properly without violating any limit values.

Read access

As a general rule, values should always be **read interpreter-synchronously**.

Write access

When a CNC program is being executed, values can be written in two ways:

- **Interpreter synchronous**
  At the same time that the interpreter interprets the respective block in the CNC program.
- **Path-synchronous**
  The value is written when the path-controlled movement reaches the respective block in the CNC program.

If the type of information being exchanged between CNC and PLC requires both read and write access to a value, the interpreter in the CNC program can be paused.

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ G-codes \ G170 and G172

However, pausing the interpreter with G170 or G172 always results in the movement being stopped at this block in the CNC program.

Example of interpreter-synchronous write access

```
def ip_synch  USINT  var (create variable inside cnc program)
```

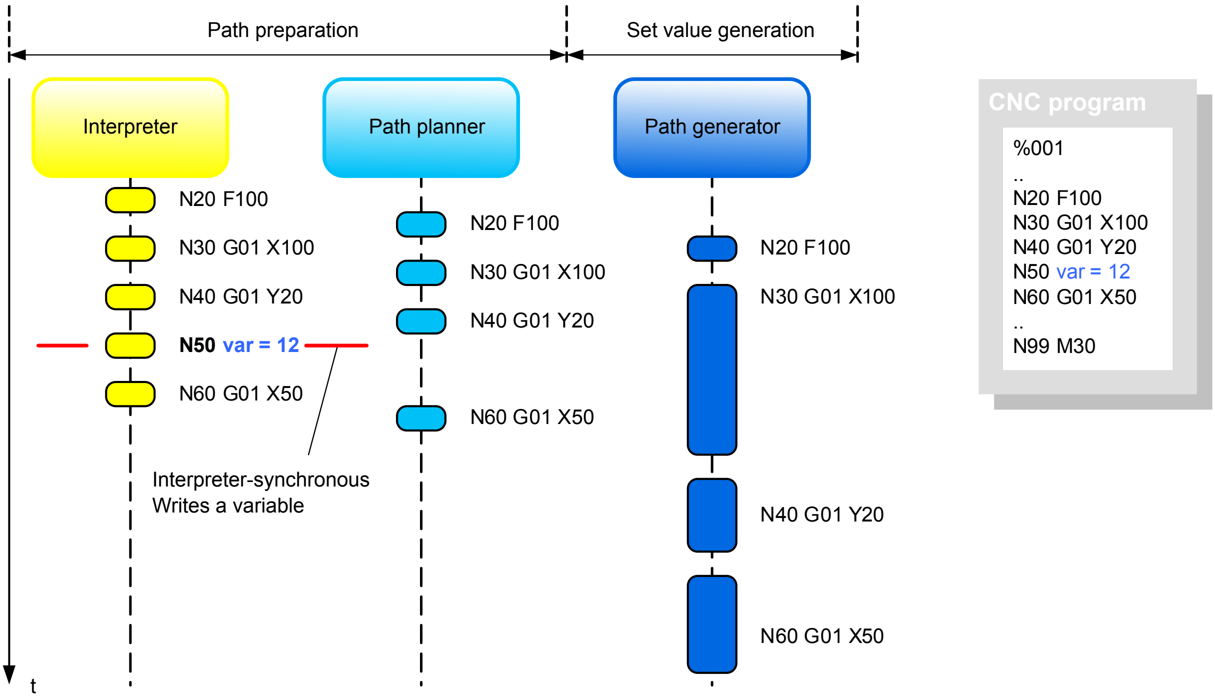


Figure 27: Interpreter-synchronous write access to a variable

Example of path-synchronous write access

```
def path_synch  USINT  var (create variable inside cnc program)
```

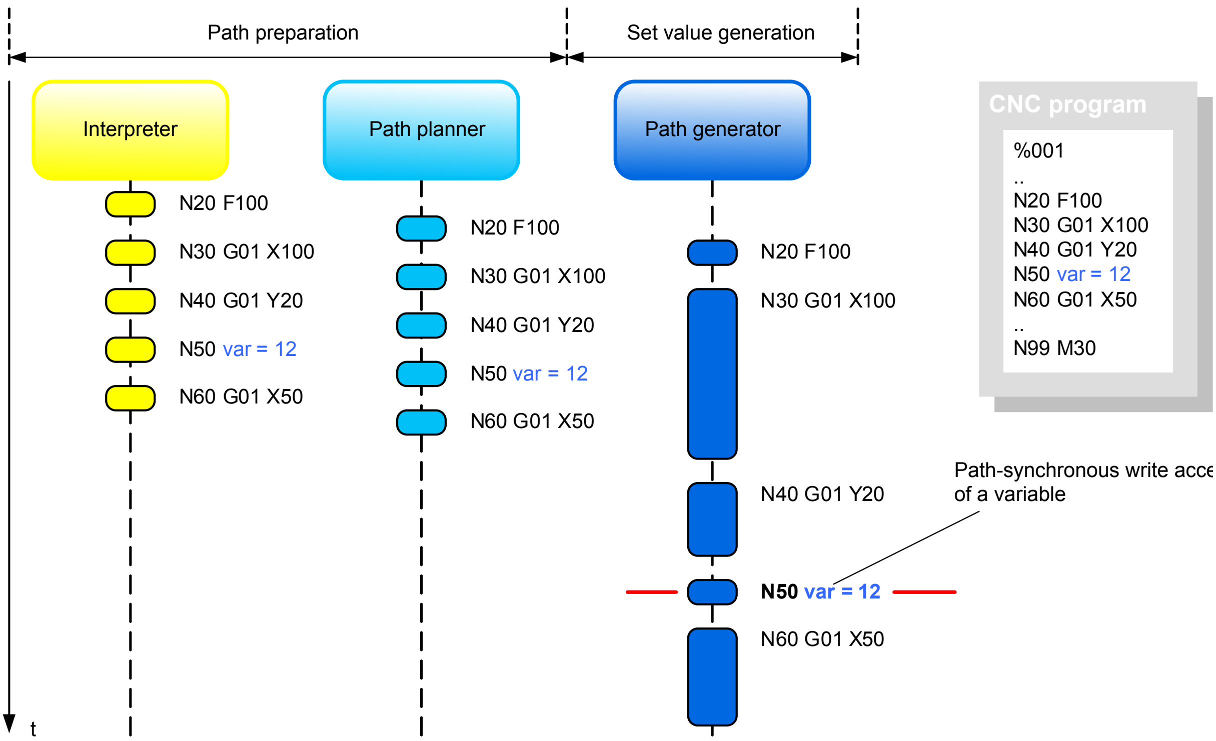


Figure 28: Path-synchronous write access to a variable

10.3 Variables

Contrary to M-codes, variables are a way to exchange more detailed information with the application program. They offer the ability to use modifiable values within the CNC program.

As a general rule, it is best to keep a CNC program as simple as possible in order to avoid unintended movements.

Variable scope

Variables can be defined with different scopes depending on where they should be applied.

- plc_global
  Variables used to address existing local or global control variables.
  These variables can be used to exchange information between the application program and the CNC program.

- nc_global
  Variables applied across all axis groups.
  These can be used to exchange information between the CNC programs for various axis groups.

- ip_global
  Variables applied across within a single axis group.
  These variables allow you to exchange information from one CNC program to another CNC program in the same axis group.

- local
  These variables are only applied within a single CNC program.
  They can be used to store the results of calculations or as loop counters.

Additional information and examples on the topic of variables can be found here:

Motion \ Reference manual \ ARNC0 \ CNC programming instructions \ Variables

Exercise:

The feed forward should be programmed in a CNC program using a local variable. A PLC variable should be used to specify the target position of a movement. A CNC program can therefore be used that can move to various positions without any changes.

1 Using a local variable as the feed forward definition in a CNC program.

```
def DINT feedrate = 1000
F=feedrate
```

2 Create a PLC variable and use it as the target position in a CNC program.

```
def plc_global  REAL task:positionX as posX
G01 X=posX
```

Variables can be **written** in two ways within a CNC program:

- Interpreter Synchronous
- Path-synchronous

Due to the predictive planning of path-controlled movements, variables are always **read interpreter-synchronously.**

11 Summary

With B&R motion control technology, path-controlled movements are easy to integrate in a machine. Regardless of the drive hardware used, path-controlled movements can easily be implemented in a CNC program. The application program is clearly structured and easy to implement thanks to the use of PLCopen function blocks. The user is further supported by the sample programs offered for Automation Studio.

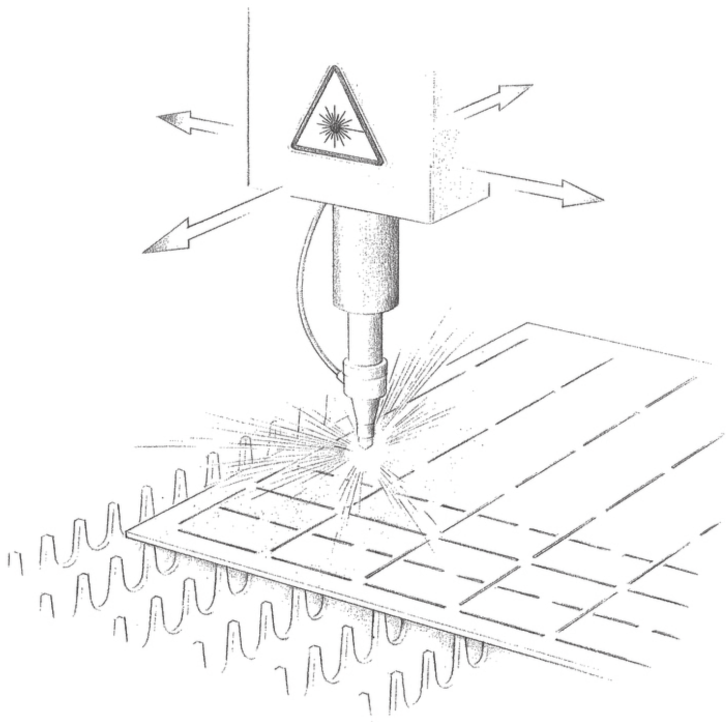


Figure 29: Laser cutting

Integrating path calculation on the controller makes it possible not only to control multiple axes simultaneously with high precision, but it also allows for optimal communication between the CNC program and the application program. This helps achieve the best possible combination of technology functions and machine functions.

Seminars and training modules

The Automation Academy provides targeted training courses for our customers as well as our own employees.
At the Automation Academy, you'll develop the skills you need in no time!
Our seminars make it possible for you to improve your knowledge in the field of automation engineering.
Once completed, you will be in a position to implement efficient automation solutions using B&R technology. This will make it possible for you to secure a decisive competitive edge by allowing you and your company to react faster to constantly changing market demands.

Automation Studio seminars and training modules

Programming and configuration	Diagnostics and service
SEM210 – Basics SEM246 – IEC 61131-3 programming language ST* SEM250 – Memory management and data storage SEM410 – Integrated motion control* SEM441 – Motion control (multi-axis systems) ** SEM480 – Hydraulics** SEM1110 – Axis groups and path-controlled movements** SEM510 – Integrated safety technology* SEM540 – Safe motion control*** SEM610 – Integrated visualization*	SEM920 – Diagnostics and service for end users SEM920 – Diagnostics and service with Automation Studio SEM950 – POWERLINK configuration and diagnostics* If you don't happen to find a seminar on our website that suits your needs, keep in mind that we also offer customized seminars that we can set up in coordination with your sales representatives: SEM099 – Individual training day Please visit our website for more information****: **www.br-automation.com/academy**

Overview of training modules	
TM210 – Working with Automation Studio TM213 – Automation Runtime TM223 – Automation Studio Diagnostics TM230 – Structured Software Development TM240 – Ladder Diagram (LD) TM241 – Function Block Diagram (FBD) TM242 – Sequential Function Chart (SFC) TM246 – Structured Text (ST) TM250 – Memory Management and Data Storage TM400 – Introduction to Motion Control TM410 – Working with Integrated Motion Control TM440 – Motion Control: Basic Functions TM441 – Motion Control: Multi-axis Functions TM1110 – Integrated Motion Control (Axis Groups) TM1111 – Integrated Motion Control (Path Controlled Movements) TM450 – Motion Control Concept and Configuration TM460 – Initial Commissioning of Motors TM500 – Introduction to Integrated Safety TM510 – Working with SafeDESIGNER TM540 – Integrated Safe Motion Control	TM600 – Introduction to Visualization TM610 – Working with Integrated Visualization TM630 – Visualization Programming Guide TM640 – Alarm System, Trends and Diagnostics TM670 – Advanced Visual Components TM920 – Diagnostics and service TM923 – Diagnostics and Service with Automation Studio TM950 – POWERLINK Configuration and Diagnostics TM261 – Closed-loop Control with LOOPCONR TM280 – Condition Monitoring for Vibration Measurement TM480 – The Basics of Hydraulics TM481 – Valve-based Hydraulic Drives TM482 – Hydraulic Servo Pump Drives TM490 – Printing machine technology In addition to a printed version, our training modules are also available on our website for download as electronic documents (login required): Visit our website for more information: **www.br-automation.com/academy**

Process control seminars and training modules

Process control standard seminars	Process control training modules
SEM841 – Process Control Training: Basic 1 SEM842 – Process Control Training: Basic 2 SEM890 – Advanced Process Control Solutions	TM800 – APROL System Concept TM811 – APROL Runtime System TM812 – APROL Operator Management TM813 – APROL XML Queries and Audit Trail TM830 – APROL Project Engineering TM890 – The Basics of LINUX Visit our website for more information: **www.br-automation.com/academy**

* SEM210 - Basics is a prerequisite for this seminar.
** SEM410 - Integrated motion control is a prerequisite for this seminar.
*** SEM410 - Integrated motion control and SEM510 - Integrated safety technology are prerequisites for this seminar.
**** Our seminars are listed in the Academy\Seminars area of the website. Seminar titles may vary by country. Not all seminars are available in every country.

TM1111TRE.40-ENG