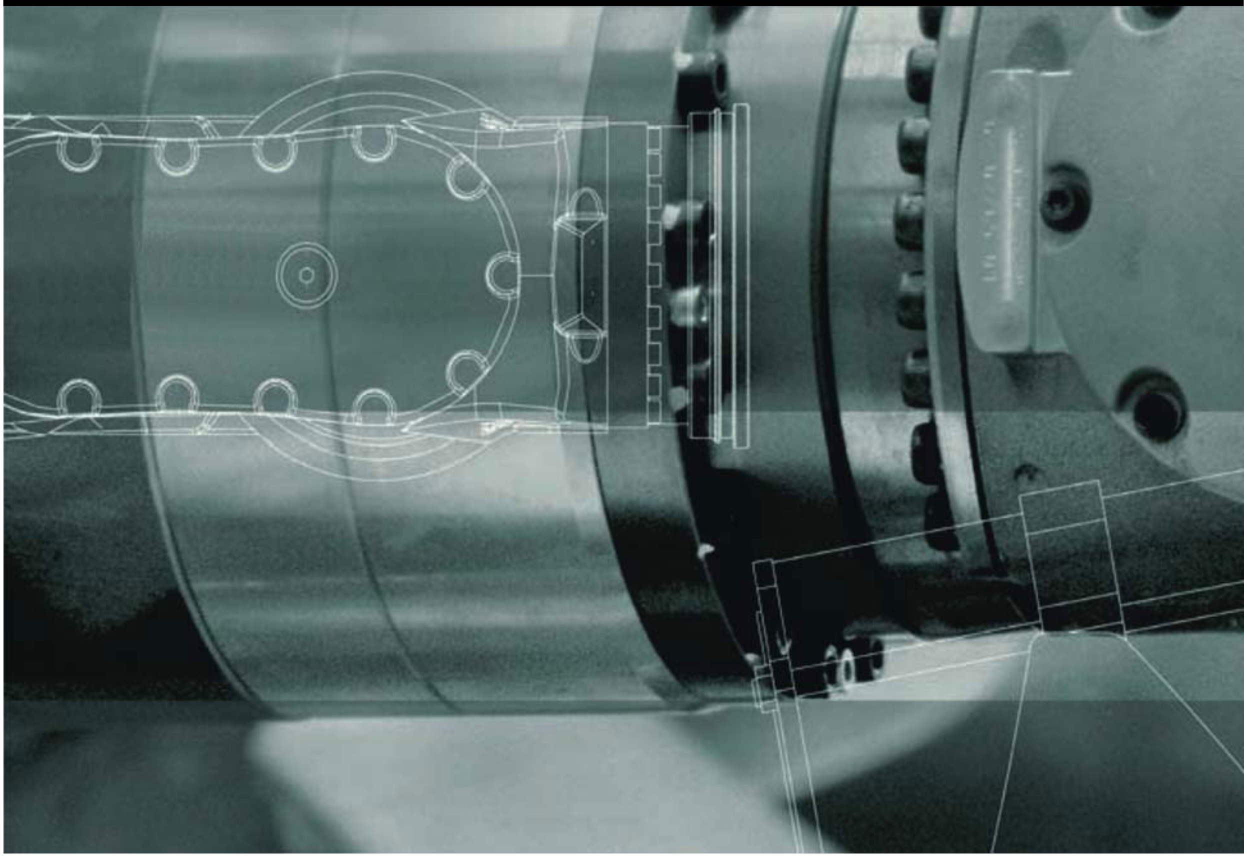
KUKA System Software

KUKA Deutschland GmbH

KUKA System Software 8.3

Operating and Programming Instructions for End Users



Issued: 18.04.2018

Version: KSS 8.3 END V6

KUKA Deutschland GmbH
Zugspitzstraße 140
D-86165 Augsburg
Germany

Other functions not described in this documentation may be operable in the controller. The user has no claims to these functions, however, in the case of a replacement or service work.

We have checked the content of this documentation for conformity with the hardware and software described. Nevertheless, discrepancies cannot be precluded, for which reason we are not able to guarantee total conformity. The information in this documentation is checked on a regular basis, however, and necessary corrections will be incorporated in the subsequent edition.

Subject to technical alterations without an effect on the function.

KIM-PS5-DOC

Translation of the original documentation

Publication:	Pub KSS 8.3 END (PDF) en
Book structure:	KSS 8.3 END V6.3
Version:	KSS 8.3 END V6

Contents

1 Introduction

1.1 Target group

This documentation is aimed at users with the following knowledge and skills:

- Basic knowledge of the industrial robot

For optimal use of our products, we recommend that our customers take part in a course of training at KUKA College. Information about the training program can be found at www.kuka.com or can be obtained directly from our subsidiaries.

1.2 Industrial robot documentation

The industrial robot documentation consists of the following parts:

- Documentation for the manipulator
- Documentation for the robot controller
- Operating and programming instructions for the System Software
- Instructions for options and accessories
- Parts catalog on storage medium

Each of these sets of instructions is a separate document.

1.3 Representation of warnings and notes

Safety

These warnings are relevant to safety and **must** be observed.

DANGER	These warnings mean that it is certain or highly probable that death or severe injuries **will** occur, if no precautions are taken.

WARNING	These warnings mean that death or severe injuries **may** occur, if no precautions are taken.

CAUTION	These warnings mean that minor injuries **may** occur, if no precautions are taken.

NOTICE	These warnings mean that damage to property **may** occur, if no precautions are taken.

These warnings contain references to safety-relevant information or general safety measures. These warnings do not refer to individual hazards or individual precautionary measures.

This warning draws attention to procedures which serve to prevent or remedy emergencies or malfunctions:

SAFETY INSTRUCTIONS	The following procedure must be followed exactly!

Procedures marked with this warning **must** be followed exactly.

Notices

These notices serve to make your work easier or contain references to further information.

Tip to make your work easier or reference to further information.

1.4 Trademarks

Windows is a trademark of Microsoft Corporation.

WordPad is a trademark of Microsoft Corporation.

2 Product description

2.1 Overview of the industrial robot

The industrial robot consists of the following components:

- Manipulator
- Robot controller
- Teach pendant
- Connecting cables
- Software
- Options, accessories

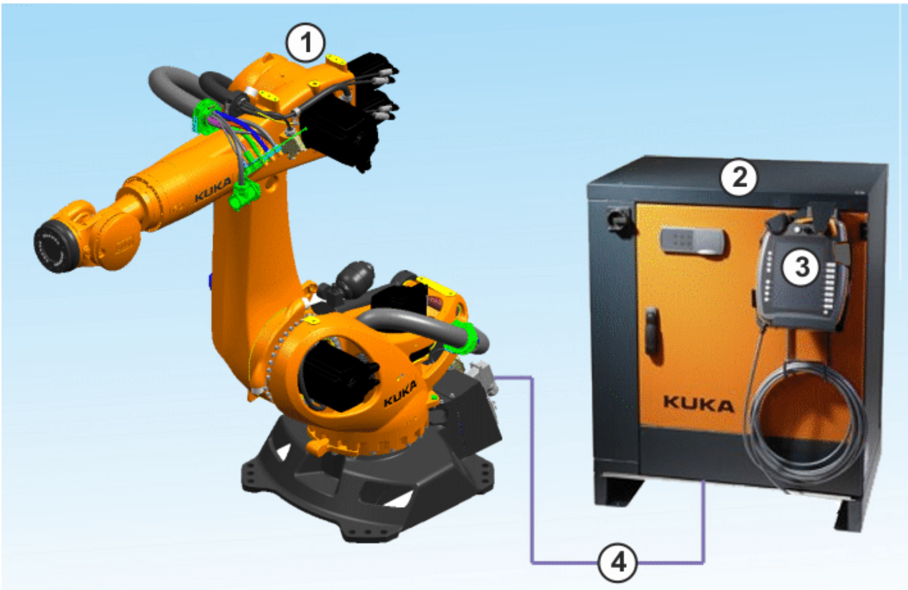


Fig. 2-1: Example of an industrial robot

1	Manipulator	3	KUKA smartPAD
2	Robot controller	4	Connecting cables

2.2 Overview of KUKA System Software (KSS)

Description

The KUKA System Software (KSS) is responsible for all the basic operator control functions of the industrial robot.

- Path planning
- I/O management
- Data and file management
- etc.

Additional technology packages, containing application-specific instructions and configurations, can be installed.

smartHMI

The user interface of the KUKA System Software is called KUKA smartHMI (smart Human-Machine Interface).

Features:

- User administration
- Program editor
- KRL (KUKA Robot Language)

- Inline forms for programming
- Message display
- Configuration window
- etc.

(>>> 4.2 "KUKA smartHMI user interface" Page 45)

Depending on customer-specific settings, the user interface may vary from the standard interface.

2.3 System requirements

The System Software 8.3 can be run on the following robot controller:

- KR C4
- with at least 2 GB RAM
- with Windows Embedded Standard 7 V4.x

2.4 Intended use of the KUKA System Software

Use

The KUKA System Software is intended exclusively for the operation of a KUKA industrial robot or customer-specific kinematic system.

Each version of the KUKA System Software may be operated exclusively in accordance with the specified system requirements.

Misuse

Any use or application deviating from the intended use is deemed to be misuse and is not allowed. KUKA Deutschland GmbH is not liable for any damage resulting from such misuse. The risk lies entirely with the user.

Examples of such misuse include:

- Operation of a kinematic system that is neither a KUKA industrial robot nor a customer-specific kinematic system
- Operation of the KSS not in accordance with the specified system requirements

2.5 KUKA USB sticks

Special USB sticks from KUKA are available for the KR C4 and VKR C4 robot controllers. The sticks are available in the following variants:

- Bootable variant, exclusively in conjunction with the software KUKA.RecoveryUSB

  Color: Orange
- Non-bootable variant for data backup

  Color: Gray or black

> For further information about the USB sticks, please contact KUKA Deutschland GmbH.

> **NOTICE** For activities on the robot controller requiring a USB stick, use of a KUKA stick is generally recommended.
>
> - The KUKA sticks are validated for use with the KR C4/VKR C4.
> - Data may be lost if sticks from other manufacturers are used.

3 Safety

3.1 General

3.1.1 Liability

The device described in this document is either an industrial robot or a component thereof.

Components of the industrial robot:

- Manipulator
- Robot controller
- Teach pendant
- Connecting cables
- External axes (optional)
  e.g. linear unit, turn-tilt table, positioner
- Software
- Options, accessories

The industrial robot is built using state-of-the-art technology and in accordance with the recognized safety rules. Nevertheless, misuse of the industrial robot may constitute a risk to life and limb or cause damage to the industrial robot and to other material property.

The industrial robot may only be used in perfect technical condition in accordance with its designated use and only by safety-conscious persons who are fully aware of the risks involved in its operation. Use of the industrial robot is subject to compliance with this document and with the declaration of incorporation supplied together with the industrial robot. Any functional disorders affecting safety must be rectified immediately.

Safety information

Information about safety may not be construed against KUKA Deutschland GmbH. Even if all safety instructions are followed, this is not a guarantee that the industrial robot will not cause personal injuries or material damage.

No modifications may be carried out to the industrial robot without the authorization of KUKA Deutschland GmbH. Additional components (tools, software, etc.), not supplied by KUKA Deutschland GmbH, may be integrated into the industrial robot. The user is liable for any damage these components may cause to the industrial robot or to other material property.

In addition to the Safety chapter, this document contains further safety instructions. These must also be observed.

3.1.2 Intended use of the industrial robot

The industrial robot is intended exclusively for the use designated in the “Purpose” chapter of the operating instructions or assembly instructions.

Any use or application deviating from the intended use is deemed to be misuse and is not allowed. The manufacturer is not liable for any damage resulting from such misuse. The risk lies entirely with the user.

Operation of the industrial robot in accordance with its intended use also requires compliance with the operating and assembly instructions for the individual components, with particular reference to the maintenance specifications.

Misuse

Any use or application deviating from the intended use is deemed to be misuse and is not allowed. This includes e.g.:

- Use as a climbing aid
- Operation outside the specified operating parameters
- Operation without the required safety equipment

3.1.3 EC declaration of conformity and declaration of incorporation

The industrial robot constitutes partly completed machinery as defined by the EC Machinery Directive. The industrial robot may only be put into operation if the following preconditions are met:

- The industrial robot is integrated into a complete system.

  or: The industrial robot, together with other machinery, constitutes a complete system.

  or: All safety functions and safeguards required for operation in the complete machine as defined by the EC Machinery Directive have been added to the industrial robot.
- The complete system complies with the EC Machinery Directive. This has been confirmed by means of a conformity assessment procedure.

EC declaration of conformity

The system integrator must issue an EC declaration of conformity for the complete system in accordance with the Machinery Directive. The EC declaration of conformity forms the basis for the CE mark for the system. The industrial robot must always be operated in accordance with the applicable national laws, regulations and standards.

The robot controller has a CE mark in accordance with the EMC Directive and the Low Voltage Directive.

Declaration of incorporation

The partly completed machinery is supplied with a declaration of incorporation in accordance with Annex II B of the EC Machinery Directive 2006/42/EC. The assembly instructions and a list of essential requirements complied with in accordance with Annex I are integral parts of this declaration of incorporation.

The declaration of incorporation declares that the start-up of the partly completed machinery is not allowed until the partly completed machinery has been incorporated into machinery, or has been assembled with other parts to form machinery, and this machinery complies with the terms of the EC Machinery Directive, and the EC declaration of conformity is present in accordance with Annex II A.

3.1.4 Terms used

STOP 0, STOP 1 and STOP 2 are the stop definitions according to EN 60204-1:2006.

Term	Description
Axis range	Range of each axis, in degrees or millimeters, within which it may move. The axis range must be defined for each axis.
Stopping distance	Stopping distance = reaction distance + braking distance The stopping distance is part of the danger zone.
Workspace	Area within which the robot may move. The workspace is derived from the individual axis ranges.
User	The user of the industrial robot can be the management, employer or delegated person responsible for use of the industrial robot.
Danger zone	The danger zone consists of the workspace and the stopping distances of the manipulator and external axes (optional).

Term	Description
Service life	The service life of a safety-relevant component begins at the time of delivery of the component to the customer. The service life is not affected by whether the component is used or not, as safety-relevant components are also subject to aging during storage.
KUKA smartPAD	see “smartPAD”
Manipulator	The robot arm and the associated electrical installations
Safety zone	The safety zone is situated outside the danger zone.
Safe operational stop	The safe operational stop is a standstill monitoring function. It does not stop the robot motion, but monitors whether the robot axes are stationary. If these are moved during the safe operational stop, a safety stop STOP 0 is triggered. The safe operational stop can also be triggered externally. When a safe operational stop is triggered, the robot controller sets an output to the field bus. The output is set even if not all the axes were stationary at the time of triggering, thereby causing a safety stop STOP 0 to be triggered.
Safety STOP 0	A stop that is triggered and executed by the safety controller. The safety controller immediately switches off the drives and the power supply to the brakes. **Note:** This stop is called safety STOP 0 in this document.
Safety STOP 1	A stop that is triggered and monitored by the safety controller. The braking operation is carried out by the non-safety-oriented section of the robot controller and monitored by the safety controller. As soon as the manipulator has stopped, the safety controller deactivates the drives and the power supply of the brakes. When a safety STOP 1 is triggered, the robot controller sets an output to the field bus. The safety STOP 1 can also be triggered externally. **Note:** This stop is called safety STOP 1 in this document.
Safety STOP 2	A stop that is triggered and monitored by the safety controller. The braking operation is carried out by the non-safety-oriented section of the robot controller and monitored by the safety controller. The drives remain activated and the brakes released. As soon as the manipulator is at a standstill, a safe operational stop is triggered. When a safety STOP 2 is triggered, the robot controller sets an output to the field bus. The safety STOP 2 can also be triggered externally. **Note:** This stop is called safety STOP 2 in this document.
Safety options	Generic term for options which make it possible to configure additional safe monitoring functions in addition to the standard safety functions. Example: SafeOperation
smartPAD	Programming device for the robot controller The smartPAD has all the operator control and display functions required for operating and programming the industrial robot.
Stop category 0	The drives are deactivated immediately and the brakes are applied. The manipulator and any external axes (optional) perform path-oriented braking. **Note:** This stop category is called STOP 0 in this document.

Term	Description
Stop category 1	The manipulator and any external axes (optional) perform path-maintaining braking. ■ Operating mode T1: The drives are deactivated as soon as the robot has stopped, but no later than after 680 ms. ■ Operating modes T2, AUT (KR C4), AUT EXT (KR C4), EXT (VKR C4): The drives are switched off after 1.5 s. **Note:** This stop category is called STOP 1 in this document.
Stop category 1 - Drive Ramp Stop	The manipulator and any external axes (optional) perform path-oriented braking. ■ Operating mode T1: The drives are deactivated as soon as the robot has stopped, but no later than after 680 ms. ■ Operating modes T2, AUT (KR C4), AUT EXT (KR C4), EXT (VKR C4): The drives are switched off after 1.5 s. **Note:** This stop category is called STOP 1 - DRS in this document.
Stop category 2	The drives are not deactivated and the brakes are not applied. The manipulator and any external axes (optional) are braked with a path-maintaining braking ramp. **Note:** This stop category is called STOP 2 in this document.
System integrator (plant integrator)	The system integrator is responsible for safely integrating the industrial robot into a complete system and commissioning it.
T1	Test mode, Manual Reduced Velocity (<= 250 mm/s)
T2	Test mode, Manual High Velocity (> 250 mm/s permissible)
External axis	Axis of motion that does not belong to the manipulator, yet is controlled with the robot controller. e.g. KUKA linear unit, turn-tilt table, Posiflex

3.2 Personnel

The following persons or groups of persons are defined for the industrial robot:

- User
- Personnel

> All persons working with the industrial robot must have read and understood the industrial robot documentation, including the safety chapter.

User

The user must observe the labor laws and regulations. This includes e.g.:

- The user must comply with his monitoring obligations.
- The user must carry out briefing at defined intervals.

Personnel

Personnel must be instructed, before any work is commenced, in the type of work involved and what exactly it entails as well as any hazards which may exist. Instruction must be carried out regularly. Instruction is also required after particular incidents or technical modifications.

Personnel includes:

- System integrator
- Operators, subdivided into:
  - Start-up, maintenance and service personnel
  - Operating personnel

- Cleaning personnel

> ⚠ Installation, exchange, adjustment, operation, maintenance and repair must be performed only as specified in the operating or assembly instructions for the relevant component of the industrial robot and only by personnel specially trained for this purpose.

System integrator

The industrial robot is safely integrated into a complete system by the system integrator.

The system integrator is responsible for the following tasks:

- Installing the industrial robot
- Connecting the industrial robot
- Performing risk assessment
- Implementing the required safety functions and safeguards
- Issuing the EC declaration of conformity
- Attaching the CE mark
- Creating the operating instructions for the system

Operators

The operator must meet the following preconditions:

- The operator must be trained for the work to be carried out.
- Work on the system must only be carried out by qualified personnel. These are people who, due to their specialist training, knowledge and experience, and their familiarization with the relevant standards, are able to assess the work to be carried out and detect any potential hazards.

> ⚠ Work on the electrical and mechanical equipment of the industrial robot may only be carried out by specially trained personnel.

3.3 Workspace, safety zone and danger zone

Workspaces are to be restricted to the necessary minimum size. A workspace must be safeguarded using appropriate safeguards.

The safeguards (e.g. safety gate) must be situated inside the safety zone. In the case of a stop, the manipulator and external axes (optional) are braked and come to a stop within the danger zone.

The danger zone consists of the workspace and the stopping distances of the manipulator and external axes (optional). It must be safeguarded by means of physical safeguards to prevent danger to persons or the risk of material damage.

3.3.1 Determining stopping distances

The system integrator's risk assessment may indicate that the stopping distances must be determined for an application. In order to determine the stopping distances, the system integrator must identify the safety-relevant points on the programmed path.

When determining the stopping distances, the robot must be moved with the tool and loads which are also used in the application. The robot must be at operating temperature. This is the case after approx. 1 h in normal operation.

During execution of the application, the robot must be stopped at the point from which the stopping distance is to be calculated. This process must be repeated several times with a safety stop 0 and a safety stop 1. The least favorable stopping distance is decisive.

A safety stop 0 can be triggered by a safe operational stop via the safety interface, for example. If a safety option is installed, it can be triggered, for instance, by a space violation (e.g. the robot exceeds the limit of an activated workspace in Automatic mode).

A safety stop 1 can be triggered by pressing the EMERGENCY STOP device on the smartPAD, for example.

3.4 Triggers for stop reactions

Stop reactions of the industrial robot are triggered in response to operator actions or as a reaction to monitoring functions and error messages. The following table shows the different stop reactions according to the operating mode that has been set.

Trigger	T1, T2	AUT, AUT EXT
Start key released	STOP 2	-
STOP key pressed	STOP 2	
Drives OFF	STOP 1	
$MOVE_ENABLE input drops out	STOP 2	
Power switched off via main switch or power failure	STOP 0	
Internal error in non-safety-oriented part of the robot controller	STOP 0 or STOP 1 (dependent on the cause of the error)	
Operating mode changed during operation	Safety stop 2	
Safety gate opened (operator safety)	-	Safety stop 1
Enabling switch released	Safety stop 2	-
Enabling switch pressed fully down or error	Safety stop 1	-
E-STOP pressed	Safety stop 1	
Error in safety controller or periphery of the safety controller	Safety stop 0	

3.5 Safety functions

3.5.1 Overview of the safety functions

The following safety functions are present in the industrial robot:

- Selecting the operating mode
- Operator safety (= connection for the monitoring of physical safeguards)
- EMERGENCY STOP device
- Enabling device
- External safe operational stop
- External safety stop 1
- External safety stop 2
- Velocity monitoring in T1

The safety functions of the industrial robot meet the following requirements:

- **Category 3** and **Performance Level d** in accordance with EN ISO 13849-1

The requirements are only met on the following condition, however:

- The EMERGENCY STOP device is pressed at least once every 12 months.

The following components are involved in the safety functions:

- Safety controller in the control PC
- KUKA smartPAD
- Cabinet Control Unit (CCU)
- Resolver Digital Converter (RDC)
- KUKA Power Pack (KPP)
- KUKA Servo Pack (KSP)
- Safety Interface Board (SIB) (if used)

There are also interfaces to components outside the industrial robot and to other robot controllers.

DANGER In the absence of operational safety functions and safeguards, the industrial robot can cause personal injury or material damage. If safety functions or safeguards are dismantled or deactivated, the industrial robot may not be operated.

During system planning, the safety functions of the overall system must also be planned and designed. The industrial robot must be integrated into this safety system of the overall system.

3.5.2 Safety controller

The safety controller is a unit inside the control PC. It links safety-relevant signals and safety-relevant monitoring functions.

Safety controller tasks:

- Switching off the drives; applying the brakes
- Monitoring the braking ramp
- Standstill monitoring (after the stop)
- Velocity monitoring in T1
- Evaluation of safety-relevant signals
- Setting of safety-oriented outputs

3.5.3 Selecting the operating mode

Operating modes

The industrial robot can be operated in the following modes:

- Manual Reduced Velocity (T1)
- Manual High Velocity (T2)
- Automatic (AUT)
- Automatic External (AUT EXT)

Do not change the operating mode while a program is running. If the operating mode is changed during program execution, the industrial robot is stopped with a safety stop 2.

Operating mode	Use	Velocities
T1	For test operation, programming and teaching	■ Program verification: Programmed velocity, maximum 250 mm/s ■ Jog mode: Jog velocity, maximum 250 mm/s
T2	For test operation	■ Program verification: Programmed velocity ■ Jog mode: Not possible
AUT	For industrial robots without higher-level controllers	■ Program operation: Programmed velocity ■ Jog mode: Not possible
AUT EXT	For industrial robots with higher-level controllers, e.g. PLC	■ Program operation: Programmed velocity ■ Jog mode: Not possible

Mode selector switch

The user can change the operating mode via the connection manager. The connection manager is a view that is called by means of the mode selector switch on the smartPAD.

The mode selector switch may be one of the following variants:

- With key

  It is only possible to change operating mode if the key is inserted.
- Without key

> **⚠ WARNING** If the smartPAD is fitted with a switch without a key: An additional device must be present to ensure that the relevant functions cannot be executed by all users, but only by a restricted group of people.
> The device itself must not trigger motions of the industrial robot or other hazards. If this device is missing, death or severe injuries may result.

The system integrator is responsible for ensuring that such a device is implemented.

3.5.4 “Operator safety” signal

The “operator safety” signal is used for monitoring physical safeguards, e.g. safety gates. Automatic operation is not possible without this signal. In the event of a loss of signal during automatic operation (e.g. safety gate is opened), the manipulator stops with a safety stop 1.

Operator safety is not active in modes T1 (Manual Reduced Velocity) and T2 (Manual High Velocity).

> ⚠ **WARNING** Following a loss of signal, automatic operation may only be resumed when the safeguard has been closed and when the closing has been acknowledged. This acknowledgement is to prevent automatic operation from being resumed inadvertently while there are still persons in the danger zone, e.g. due to the safety gate closing accidentally.
> The acknowledgement must be designed in such a way that an actual check of the danger zone can be carried out first. Other acknowledgement functions (e.g. an acknowlegement which is automatically triggered by closure of the safeguard) are not permitted.
> The system integrator is responsible for ensuring that these criteria are met. Failure to met them may result in death, severe injuries or considerable damage to property.

3.5.5 EMERGENCY STOP device

The EMERGENCY STOP device for the industrial robot is the EMERGENCY STOP device on the smartPAD. The device must be pressed in the event of a hazardous situation or emergency.

Reactions of the industrial robot if the EMERGENCY STOP device is pressed:

- The manipulator and any external axes (optional) are stopped with a safety stop 1.

Before operation can be resumed, the EMERGENCY STOP device must be turned to release it.

> ⚠ **WARNING** Tools and other equipment connected to the robot must be integrated into the EMERGENCY STOP circuit on the system side if they could constitute a potential hazard.
> Failure to observe this precaution may result in death, severe injuries or considerable damage to property.

There must always be at least one external EMERGENCY STOP device installed. This ensures that an EMERGENCY STOP device is available even when the smartPAD is disconnected.

(>>> 3.5.7 "External EMERGENCY STOP device" Page 24)

3.5.6 Logging off from the higher-level safety controller

If the robot controller is connected to a higher-level safety controller, this connection will inevitably be terminated in the following cases:

- Switching off the voltage via the main switch of the robot
  Or power failure
- Shutdown of the robot controller via the smartHMI
- Activation of a WorkVisual project in WorkVisual or directly on the robot controller
- Changes to **Start-up** > **Network configuration**
- Changes to **Configuration** > **Safety configuration**
- **I/O drivers** > **Reconfigure**
- Restoration of an archive

Effect of the interruption:

- If a discrete safety interface is used, this triggers an EMERGENCY STOP for the overall system.

- If the Ethernet interface is used, the KUKA safety controller generates a signal that prevents the higher-level controller from triggering an EMERGENCY STOP for the overall system.

> If the Ethernet safety interface is used: In his risk assessment, the system integrator must take into consideration whether the fact that switching off the robot controller does not trigger an EMERGENCY STOP of the overall system could constitute a hazard and, if so, how this hazard can be countered.
> Failure to take this into consideration may result in death, injuries or damage to property.

> **WARNING** If a robot controller is switched off, the E-STOP device on the smartPAD is no longer functional. The user is responsible for ensuring that the smartPAD is either covered or removed from the system. This serves to prevent operational and non-operational EMERGENCY STOP devices from becoming interchanged.
> Failure to observe this precaution may result in death, injuries or damage to property.

3.5.7 External EMERGENCY STOP device

Every operator station that can initiate a robot motion or other potentially hazardous situation must be equipped with an EMERGENCY STOP device. The system integrator is responsible for ensuring this.

There must always be at least one external EMERGENCY STOP device installed. This ensures that an EMERGENCY STOP device is available even when the smartPAD is disconnected.

External EMERGENCY STOP devices are connected via the customer interface. External EMERGENCY STOP devices are not included in the scope of supply of the industrial robot.

3.5.8 Enabling device

The enabling devices of the industrial robot are the enabling switches on the smartPAD.

There are 3 enabling switches installed on the smartPAD. The enabling switches have 3 positions:

- Not pressed
- Center position
- Panic position

In the test modes, the manipulator can only be moved if one of the enabling switches is held in the central position.

- Releasing the enabling switch triggers a safety stop 2.
- Pressing the enabling switch down fully (panic position) triggers a safety stop 1.
- It is possible to hold 2 enabling switches in the center position simultaneously for up to 15 seconds. This makes it possible to adjust grip from one enabling switch to another one. If 2 enabling switches are held simultaneously in the center position for longer than 15 seconds, this triggers a safety stop 1.

If an enabling switch malfunctions (e.g. jams in the central position), the industrial robot can be stopped using the following methods:

- Press the enabling switch down fully.

- Actuate the EMERGENCY STOP device.
- Release the Start key.

> **⚠ WARNING** The enabling switches must not be held down by adhesive tape or other means or tampered with in any other way.
> Death, injuries or damage to property may result.

3.5.9 External enabling device

External enabling devices are required if it is necessary for more than one person to be in the danger zone of the industrial robot.

External enabling devices are not included in the scope of supply of the industrial robot.

> Which interface can be used for connecting external enabling devices is described in the “Planning” chapter of the robot controller operating instructions and assembly instructions.

3.5.10 External safe operational stop

The safe operational stop can be triggered via an input on the customer interface. The state is maintained as long as the external signal is FALSE. If the external signal is TRUE, the manipulator can be moved again. No acknowledgement is required.

3.5.11 External safety stop 1 and external safety stop 2

Safety stop 1 and safety stop 2 can be triggered via an input on the customer interface. The state is maintained as long as the external signal is FALSE. If the external signal is TRUE, the manipulator can be moved again. No acknowledgement is required.

If interface X11 is selected as the customer interface, only the signal **Safety stop 2** is available.

3.5.12 Velocity monitoring in T1

The velocity at the mounting flange is monitored in T1 mode. If the velocity exceeds 250 mm/s, a safety stop 0 is triggered.

3.6 Additional protective equipment

3.6.1 Jog mode

In the operating modes T1 (Manual Reduced Velocity) and T2 (Manual High Velocity), the robot controller can only execute programs in jog mode. This means that it is necessary to hold down an enabling switch and the Start key in order to execute a program.

- Releasing the enabling switch triggers a safety stop 2.
- Pressing the enabling switch down fully (panic position) triggers a safety stop 1.
- Releasing the Start key triggers a STOP 2.

3.6.2 Software limit switches

The axis ranges of all manipulator and positioner axes are limited by means of adjustable software limit switches. These software limit switches only serve as machine protection and must be adjusted in such a way that the manipulator/positioner cannot hit the mechanical end stops.

The software limit switches are set during commissioning of an industrial robot.

> Further information is contained in the operating and programming instructions.

3.6.3 Mechanical end stops

Depending on the robot variant, the axis ranges of the main and wrist axes of the manipulator are partially limited by mechanical end stops.

Additional mechanical end stops can be installed on the external axes.

> **WARNING** If the manipulator or an external axis hits an obstruction or a mechanical end stop or mechanical axis limitation, the manipulator can no longer be operated safely. The manipulator must be taken out of operation and KUKA Deutschland GmbH must be consulted before it is put back into operation.

3.6.4 Mechanical axis limitation (optional)

Some manipulators can be fitted with mechanical axis limitation systems in axes A1 to A3. The axis limitation systems restrict the working range to the required minimum. This increases personal safety and protection of the system.

In the case of manipulators that are not designed to be fitted with mechanical axis limitation, the workspace must be laid out in such a way that there is no danger to persons or material property, even in the absence of mechanical axis limitation.

If this is not possible, the workspace must be limited by means of photoelectric barriers, photoelectric curtains or obstacles on the system side. There must be no shearing or crushing hazards at the loading and transfer areas.

> This option is not available for all robot models. Information on specific robot models can be obtained from KUKA Deutschland GmbH.

3.6.5 Options for moving the manipulator without drive energy

> The system user is responsible for ensuring that the training of personnel with regard to the response to emergencies or exceptional situations also includes how the manipulator can be moved without drive energy.

Description

The following options are available for moving the manipulator without drive energy after an accident or malfunction:

- Release device (optional)

  The release device can be used for the main axis drive motors and, depending on the robot variant, also for the wrist axis drive motors.

- Brake release device (option)

  The brake release device is designed for robot variants whose motors are not freely accessible.
- Moving the wrist axes directly by hand

  There is no release device available for the wrist axes of variants in the low payload category. This is not necessary because the wrist axes can be moved directly by hand.

Information about the options available for the various robot models and about how to use them can be found in the assembly and operating instructions for the robot or requested from KUKA Deutschland GmbH.

NOTICE Moving the manipulator without drive energy can damage the motor brakes of the axes concerned. The motor must be replaced if the brake has been damaged. The manipulator may therefore be moved without drive energy only in emergencies, e.g. for rescuing persons.

3.6.6 Labeling on the industrial robot

All plates, labels, symbols and marks constitute safety-relevant parts of the industrial robot. They must not be modified or removed.

Labeling on the industrial robot consists of:

- Identification plates
- Warning signs
- Safety symbols
- Designation labels
- Cable markings
- Rating plates

Further information is contained in the technical data of the operating instructions or assembly instructions of the components of the industrial robot.

3.6.7 External safeguards

The access of persons to the danger zone of the industrial robot must be prevented by means of safeguards. It is the responsibility of the system integrator to ensure this.

Physical safeguards must meet the following requirements:

- They meet the requirements of EN ISO 14120.
- They prevent access of persons to the danger zone and cannot be easily circumvented.
- They are sufficiently fastened and can withstand all forces that are likely to occur in the course of operation, whether from inside or outside the enclosure.
- They do not, themselves, represent a hazard or potential hazard.
- Prescribed clearances, e.g. to danger zones, are adhered to.

Safety gates (maintenance gates) must meet the following requirements:

- They are reduced to an absolute minimum.

- The interlocks (e.g. safety gate switches) are linked to the operator safety input of the robot controller via safety gate switching devices or safety PLC.
- Switching devices, switches and the type of switching conform to the requirements of Performance Level d and category 3 according to EN ISO 13849-1.
- Depending on the risk situation: the safety gate is additionally safeguarded by means of a locking mechanism that only allows the gate to be opened if the manipulator is safely at a standstill.
- The button for acknowledging the safety gate is located outside the space limited by the safeguards.

Further information is contained in the corresponding standards and regulations. These also include EN ISO 14120.

Other safety equipment

Other safety equipment must be integrated into the system in accordance with the corresponding standards and regulations.

3.7 Overview of operating modes and safety functions

The following table indicates the operating modes in which the safety functions are active.

Safety functions	T1	T2	AUT	AUT EXT
Operator safety	-	-	Active	Active
EMERGENCY STOP device	Active	Active	Active	Active
Enabling device	Active	Active	-	-
Reduced velocity during program verification	Active	-	-	-
Jog mode	Active	Active	-	-
Software limit switches	Active	Active	Active	Active

3.8 Safety measures

3.8.1 General safety measures

The industrial robot may only be used in perfect technical condition in accordance with its intended use and only by safety-conscious persons. Operator errors can result in personal injury and damage to property.

It is important to be prepared for possible movements of the industrial robot even after the robot controller has been switched off and locked out. Incorrect installation (e.g. overload) or mechanical defects (e.g. brake defect) can cause the manipulator or external axes to sag. If work is to be carried out on a switched-off industrial robot, the manipulator and external axes must first be moved into a position in which they are unable to move on their own, whether the payload is mounted or not. If this is not possible, the manipulator and external axes must be secured by appropriate means.

DANGER In the absence of operational safety functions and safeguards, the industrial robot can cause personal injury or material damage. If safety functions or safeguards are dismantled or deactivated, the industrial robot may not be operated.

⚠ DANGER Standing underneath the robot arm can cause death or injuries. For this reason, standing underneath the robot arm is prohibited!

⚠ CAUTION The motors reach temperatures during operation which can cause burns to the skin. Contact must be avoided. Appropriate safety precautions must be taken, e.g. protective gloves must be worn.

smartPAD

The user must ensure that the industrial robot is only operated with the smartPAD by authorized persons.

If more than one smartPAD is used in the overall system, it must be ensured that it is clearly recognizable which smartPAD is connected to which industrial robot. They must not be interchanged.

⚠ WARNING The operator must ensure that decoupled smartPADs are immediately removed from the system and stored out of sight and reach of personnel working on the industrial robot. This serves to prevent operational and non-operational EMERGENCY STOP devices from becoming interchanged.
Failure to observe this precaution may result in death, severe injuries or considerable damage to property.

Modifications

After modifications to the industrial robot, checks must be carried out to ensure the required safety level. The valid national or regional work safety regulations must be observed for this check. The correct functioning of all safety functions must also be tested.

New or modified programs must always be tested first in Manual Reduced Velocity mode (T1).

After modifications to the industrial robot, existing programs must always be tested first in Manual Reduced Velocity mode (T1). This applies to all components of the industrial robot and includes e.g. modifications of the external axes or to the software and configuration settings.

Faults

The following tasks must be carried out in the case of faults in the industrial robot:

- Switch off the robot controller and secure it (e.g. with a padlock) to prevent unauthorized persons from switching it on again.
- Indicate the fault by means of a label with a corresponding warning (tag-out).
- Keep a record of the faults.
- Eliminate the fault and carry out a function test.

3.8.2 Transportation

Manipulator

The prescribed transport position of the manipulator must be observed. Transportation must be carried out in accordance with the operating instructions or assembly instructions of the robot.

Avoid vibrations and impacts during transportation in order to prevent damage to the manipulator.

Robot controller

The prescribed transport position of the robot controller must be observed. Transportation must be carried out in accordance with the operating instructions or assembly instructions of the robot controller.

Avoid vibrations and impacts during transportation in order to prevent damage to the robot controller.

External axis (optional)

The prescribed transport position of the external axis (e.g. KUKA linear unit, turn-tilt table, positioner) must be observed. Transportation must be carried out in accordance with the operating instructions or assembly instructions of the external axis.

3.8.3 Start-up and recommissioning

Before starting up systems and devices for the first time, a check must be carried out to ensure that the systems and devices are complete and operational, that they can be operated safely and that any damage is detected.

The valid national or regional work safety regulations must be observed for this check. The correct functioning of all safety functions must also be tested.

> The passwords for the user groups must be changed in the KUKA System Software before start-up. The passwords must only be communicated to authorized personnel.

> **WARNING** The robot controller is preconfigured for the specific industrial robot. If cables are interchanged, the manipulator and the external axes (optional) may receive incorrect data and can thus cause personal injury or material damage. If a system consists of more than one manipulator, always connect the connecting cables to the manipulators and their corresponding robot controllers.

> If additional components (e.g. cables), which are not part of the scope of supply of KUKA Deutschland GmbH, are integrated into the industrial robot, the user is responsible for ensuring that these components do not adversely affect or disable safety functions.

> **NOTICE** If the internal cabinet temperature of the robot controller differs greatly from the ambient temperature, condensation can form, which may cause damage to the electrical components. Do not put the robot controller into operation until the internal temperature of the cabinet has adjusted to the ambient temperature.

Function test

The following tests must be carried out before start-up and recommissioning:

General test:

It must be ensured that:

- The industrial robot is correctly installed and fastened in accordance with the specifications in the documentation.
- There is no damage to the robot that could be attributed to external forces. Examples: Dents or abrasion that could be caused by an impact or collision.

> **WARNING** In the case of such damage, the affected components must be exchanged. In particular, the motor and counterbalancing system must be checked carefully.
> External forces can cause non-visible damage. For example, it can lead to a gradual loss of drive power from the motor, resulting in unintended movements of the manipulator. Death, injuries or considerable damage to property may otherwise result.

- There are no foreign bodies or loose parts on the industrial robot.
- All required safety equipment is correctly installed and operational.

- The power supply ratings of the industrial robot correspond to the local supply voltage and mains type.
- The ground conductor and the equipotential bonding cable are sufficiently rated and correctly connected.
- The connecting cables are correctly connected and the connectors are locked.

Test of the safety functions:

A function test must be carried out for the following safety functions to ensure that they are functioning correctly:

- Local EMERGENCY STOP device
- External EMERGENCY STOP device (input and output)
- Enabling device (in the test modes)
- Operator safety
- All other safety-relevant inputs and outputs used
- Other external safety functions

3.8.3.1 Checking machine data and safety configuration

⚠ WARNING The industrial robot must not be moved if incorrect machine data or an incorrect controller configuration are loaded. Death, severe injuries or considerable damage to property may otherwise result. The correct data must be loaded.

- Following the start-up procedure, the practical tests for the machine data must be carried out. The tool must be calibrated (either via an actual calibration or through numerical entry of the data).
- Following modifications to the machine data, the safety configuration must be checked.
- After activation of a WorkVisual project on the robot controller, the safety configuration must be checked.
- If machine data are adopted when checking the safety configuration (regardless of the reason for the safety configuration check), the practical tests for the machine data must be carried out.
- System Software 8.3 or higher: If the checksum of the safety configuration has changed, the safe axis monitoring functions must be checked.

Information about checking the safety configuration and the safe axis monitoring functions is contained in the Operating and Programming Instructions for System Integrators.

If the practical tests are not successfully completed in the initial start-up, KUKA Deutschland GmbH must be contacted.

If the practical tests are not successfully completed during a different procedure, the machine data and the safety-relevant controller configuration must be checked and corrected.

General practical test

If practical tests are required for the machine data, this test must always be carried out.

For 6-axis robots:

The following methods are available for performing the practical test:

- TCP calibration with the XYZ 4-point method

  The practical test is passed if the TCP has been successfully calibrated.

Or:

1. Align the TCP with a freely selected point. The point serves as a reference point.
   - The point must be located so that reorientation is possible.
   - The point must not be located on the Z axis of the FLANGE coordinate system.
2. Move the TCP manually at least 45° once in each of the A, B and C directions.

   The movements do not have to be accumulative, i.e. after motion in one direction it is possible to return to the original position before moving in the next direction.

   The practical test is passed if the TCP does not deviate from the reference point by more than 2 cm in total.

For palletizing robots:

Palletizing robots, in this case, are either robots that can be used only as palletizers from the start or robots operated in palletizing mode. The latter must also be in palletizing mode during the practical test.

First part:

1. Mark the starting position of the TCP.

   Also read and note the starting position from the **Actual position** – **Cartesian** display on the smartHMI.
2. Jog the TCP in the X direction. The distance must be at least 20% of the robot's maximum reach. Determine the exact length via the **Actual position** display.
3. Measure the distance covered and compare it with the distance value displayed on the smartHMI. The deviation must be < 5%.
4. Repeat steps 1 and 2 for the Y direction and Z direction.

The first part of the practical test is passed if the deviation is < 5% in every direction.

Second part:

- Rotate the tool manually about A by 45°: once in the plus direction, once in the minus direction. At the same time, observe the TCP.

The second part of the practical test is passed if the position of the TCP in space is not altered during the rotations.

Practical test for axes that are not mathematically coupled

If practical tests are required for the machine data, this test must be carried out when axes are present that are not mathematically coupled.

1. Mark the starting position of the axis that is not mathematically coupled.

   Also read and note the start position from the **Actual position** display on the smartHMI.
2. Move the axis manually by a freely selected path length. Determine the path length from the **Actual position** display.
   - Move linear axes a specific distance.
   - Move rotational axes through a specific angle.
3. Measure the length of the path covered and compare it with the value displayed on the smartHMI.

   The practical test is passed if the values differ by no more than 5%.
4. Repeat the test for each axis that is not mathematically coupled.

Practical test for robot on KUKA linear unit

If practical tests are required for the machine data, this test must be carried out if the robot and KL are mathematically coupled.

- Move the KL manually in Cartesian mode.

  The practical test is passed if the TCP does not move at the same time.

Practical test for couplable axes

If practical tests are required for the machine data, this test must be carried out when axes are present that can be physically coupled and uncoupled, e.g. a servo gun.

1. Physically uncouple the couplable axis.
2. Move all the remaining axes individually.

   The practical test is passed if it has been possible to move all the remaining axes.

3.8.3.2 Start-up mode

Description

The industrial robot can be set to Start-up mode via the smartHMI user interface. In this mode, the manipulator can be moved in T1 without the external safeguards being put into operation.

The safety interface used affects “Start-up” mode:

Discrete safety interface

- System Software 8.2 or earlier:

  Start-up mode is always possible if all input signals at the discrete safety interface have the state “logic zero”. If this is not the case, the robot controller prevents or terminates Start-up mode.

  If an additional discrete safety interface for safety options is used, the inputs there must also have the state “logic zero”.
- System Software 8.3 or higher:

  Start-up mode is always possible. This also means that it is independent of the state of the inputs at the discrete safety interface.

  If an additional discrete safety interface is used for safety options: The states of these inputs are also irrelevant.

Ethernet safety interface

The robot controller prevents or terminates Start-up mode if a connection to a higher-level safety system exists or is established.

Effect

When the Start-up mode is activated, all outputs are automatically set to the state “logic zero”.

If the robot controller has a peripheral contactor (US2), and if the safety configuration specifies for this to switch in accordance with the motion enable, then the same also applies in Start-up mode. This means that if motion enable is present, the US2 voltage is switched on – even in Start-up mode.

NOTICE	The maximum number of switching cycles of the peripheral contactors is 175 per day.

Hazards

Possible hazards and risks involved in using Start-up mode:

- A person walks into the manipulator’s danger zone.
- In a hazardous situation, a disabled external EMERGENCY STOP device is actuated and the manipulator is not shut down.

Additional measures for avoiding risks in Start-up mode:

- Cover disabled EMERGENCY STOP devices or attach a warning sign indicating that the EMERGENCY STOP device is out of operation.
- If there is no safety fence, other measures must be taken to prevent persons from entering the manipulator’s danger zone, e.g. use of warning tape.

Use

Intended use of Start-up mode:

- Start-up in T1 mode when the external safeguards have not yet been installed or put into operation. The danger zone must be delimited at least by means of warning tape.
- Fault localization (periphery fault).
- Use of Start-up mode must be minimized as much as possible.

> **⚠ WARNING** Use of Start-up mode disables all external safeguards. The service personnel are responsible for ensuring that there is no-one in or near the danger zone of the manipulator as long as the safeguards are disabled.
> Failure to observe this precaution may result in death, injuries or damage to property.

Misuse

Any use or application deviating from the intended use is deemed to be misuse and is not allowed. KUKA Deutschland GmbH is not liable for any damage resulting from such misuse. The risk lies entirely with the user.

3.8.4 Manual mode

General

Manual mode is the mode for setup work. Setup work is all the tasks that have to be carried out on the industrial robot to enable automatic operation. Setup work includes:

- Jog mode
- Teaching
- Programming
- Program verification

The following must be taken into consideration in manual mode:

- New or modified programs must always be tested first in Manual Reduced Velocity mode (T1).
- The manipulator, tooling or external axes (optional) must never touch or project beyond the safety fence.
- Workpieces, tooling and other objects must not become jammed as a result of the industrial robot motion, nor must they lead to short-circuits or be liable to fall off.
- All setup work must be carried out, where possible, from outside the safeguarded area.

Setup work in T1

If it is necessary to carry out setup work from inside the safeguarded area, the following must be taken into consideration in the operating mode **Manual Reduced Velocity (T1)**:

- If it can be avoided, there must be no other persons inside the safeguarded area.
- If it is necessary for there to be several persons inside the safeguarded area, the following must be observed:
  - Each person must have an enabling device.
  - All persons must have an unimpeded view of the industrial robot.
  - Eye-contact between all persons must be possible at all times.
- The operator must be so positioned that he can see into the danger area and get out of harm’s way.
- Unexpected motions of the manipulator cannot be ruled out, e.g. in the event of a fault. For this reason, an appropriate clearance must be main-

tained between persons and the manipulator (including tool). Guide value: 50 cm.

The minimum clearance may vary depending on local circumstances, the motion program and other factors. The minimum clearance that is to apply for the specific application must be decided by the user on the basis of a risk assessment.

Setup work in T2

If it is necessary to carry out setup work from inside the safeguarded area, the following must be taken into consideration in the operating mode **Manual High Velocity (T2)**:

- This mode may only be used if the application requires a test at a velocity higher than that possible in T1 mode.
- Teaching and programming are not permissible in this operating mode.
- Before commencing the test, the operator must ensure that the enabling devices are operational.
- The operator must be positioned outside the danger zone.
- There must be no other persons inside the safeguarded area. It is the responsibility of the operator to ensure this.

3.8.5 Simulation

Simulation programs do not correspond exactly to reality. Robot programs created in simulation programs must be tested in the system in **Manual Reduced Velocity mode (T1)**. It may be necessary to modify the program.

3.8.6 Automatic mode

Automatic mode is only permissible in compliance with the following safety measures:

- All safety equipment and safeguards are present and operational.
- There are no persons in the system.
- The defined working procedures are adhered to.

If the manipulator or an external axis (optional) comes to a standstill for no apparent reason, the danger zone must not be entered until an EMERGENCY STOP has been triggered.

3.8.7 Maintenance and repair

After maintenance and repair work, checks must be carried out to ensure the required safety level. The valid national or regional work safety regulations must be observed for this check. The correct functioning of all safety functions must also be tested.

The purpose of maintenance and repair work is to ensure that the system is kept operational or, in the event of a fault, to return the system to an operational state. Repair work includes troubleshooting in addition to the actual repair itself.

The following safety measures must be carried out when working on the industrial robot:

- Carry out work outside the danger zone. If work inside the danger zone is necessary, the user must define additional safety measures to ensure the safe protection of personnel.
- Switch off the industrial robot and secure it (e.g. with a padlock) to prevent it from being switched on again. If it is necessary to carry out work with the

robot controller switched on, the user must define additional safety measures to ensure the safe protection of personnel.

- If it is necessary to carry out work with the robot controller switched on, this may only be done in operating mode T1.
- Label the system with a sign indicating that work is in progress. This sign must remain in place, even during temporary interruptions to the work.
- The EMERGENCY STOP devices must remain active. If safety functions or safeguards are deactivated during maintenance or repair work, they must be reactivated immediately after the work is completed.

DANGER Before work is commenced on live parts of the robot system, the main switch must be turned off and secured against being switched on again. The system must then be checked to ensure that it is deenergized.
It is not sufficient, before commencing work on live parts, to execute an EMERGENCY STOP or a safety stop, or to switch off the drives, as this does not disconnect the robot system from the mains power supply. Parts remain energized. Death or severe injuries may result.

Faulty components must be replaced using new components with the same article numbers or equivalent components approved by KUKA Deutschland GmbH for this purpose.

Cleaning and preventive maintenance work is to be carried out in accordance with the operating instructions.

Robot controller

Even when the robot controller is switched off, parts connected to peripheral devices may still carry voltage. The external power sources must therefore be switched off if work is to be carried out on the robot controller.

The ESD regulations must be adhered to when working on components in the robot controller.

Voltages in excess of 50 V (up to 780 V) can be present in various components for several minutes after the robot controller has been switched off! To prevent life-threatening injuries, no work may be carried out on the industrial robot in this time.

Water and dust must be prevented from entering the robot controller.

Counterbalancing system

Some robot variants are equipped with a hydropneumatic, spring or gas cylinder counterbalancing system.

The hydropneumatic and gas cylinder counterbalancing systems are pressure equipment and, as such, are subject to obligatory equipment monitoring and the provisions of the Pressure Equipment Directive.

The user must comply with the applicable national laws, regulations and standards pertaining to pressure equipment.

Inspection intervals in Germany in accordance with Industrial Safety Order, Sections 14 and 15. Inspection by the user before commissioning at the installation site.

The following safety measures must be carried out when working on the counterbalancing system:

- The manipulator assemblies supported by the counterbalancing systems must be secured.
- Work on the counterbalancing systems must only be carried out by qualified personnel.

Hazardous substances

The following safety measures must be carried out when handling hazardous substances:

- Avoid prolonged and repeated intensive contact with the skin.

- Avoid breathing in oil spray or vapors.
- Clean skin and apply skin cream.

> To ensure safe use of our products, we recommend regularly requesting up-to-date safety data sheets for hazardous substances.

3.8.8 Decommissioning, storage and disposal

The industrial robot must be decommissioned, stored and disposed of in accordance with the applicable national laws, regulations and standards.

3.8.9 Safety measures for “single point of control”

Overview

If certain components in the industrial robot are operated, safety measures must be taken to ensure complete implementation of the principle of “single point of control” (SPOC).

The relevant components are:

- Submit interpreter
- PLC
- OPC server
- Remote control tools
- Tools for configuration of bus systems with online functionality
- KUKA.RobotSensorInterface

> The implementation of additional safety measures may be required. This must be clarified for each specific application; this is the responsibility of the system integrator, programmer or user of the system.

Since only the system integrator knows the safe states of actuators in the periphery of the robot controller, it is his task to set these actuators to a safe state, e.g. in the event of an EMERGENCY STOP.

T1, T2

In modes T1 and T2, the components referred to above may only access the industrial robot if the following signals have the following states:

Signal	State required for SPOC
$USER_SAF	TRUE
$SPOC_MOTION_ENABLE	TRUE

Submit interpreter, PLC

If motions, (e.g. drives or grippers) are controlled with the submit interpreter or the PLC via the I/O system, and if they are not safeguarded by other means, then this control will take effect even in T1 and T2 modes or while an EMERGENCY STOP is active.

If variables that affect the robot motion (e.g. override) are modified with the submit interpreter or the PLC, this takes effect even in T1 and T2 modes or while an EMERGENCY STOP is active.

Safety measures:

- In T1 and T2, the system variable $OV_PRO must not be written to by the submit interpreter or the PLC.
- Do not modify safety-relevant signals and variables (e.g. operating mode, EMERGENCY STOP, safety gate contact) via the submit interpreter or PLC.

  If modifications are nonetheless required, all safety-relevant signals and variables must be linked in such a way that they cannot be set to a dangerous state by the submit interpreter or PLC. This is the responsibility of the system integrator.

OPC server, remote control tools

These components can be used with write access to modify programs, outputs or other parameters of the robot controller, without this being noticed by any persons located inside the system.

Safety measure:

If these components are used, outputs that could cause a hazard must be determined in a risk assessment. These outputs must be designed in such a way that they cannot be set without being enabled. This can be done using an external enabling device, for example.

Tools for configuration of bus systems

If these components have an online functionality, they can be used with write access to modify programs, outputs or other parameters of the robot controller, without this being noticed by any persons located inside the system.

- WorkVisual from KUKA
- Tools from other manufacturers

Safety measure:

In the test modes, programs, outputs or other parameters of the robot controller must not be modified using these components.

3.9 Applied norms and regulations

Name/Edition	Definition
2006/42/EU:2006	**Machinery Directive:** Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast)
2014/30/EU:2014	**EMC Directive:** Directive 2014/30/EC of the European Parliament and of the Council dated 26 February 2014 on the approximation of the laws of the Member States concerning electromagnetic compatibility
2014/68/EU:2014	**Pressure Equipment Directive**: Directive 2014/68/EU of the European Parliament and of the Council dated 15 May 2014 on the approximation of the laws of the Member States concerning pressure equipment (Only applicable for robots with hydropneumatic counterbalancing system.)
EN ISO 13850:2015	**Safety of machinery:** Emergency stop - Principles for design
EN ISO 13849-1:2015	**Safety of machinery:** Safety-related parts of control systems - Part 1: General principles of design
EN ISO 13849-2:2012	**Safety of machinery:** Safety-related parts of control systems - Part 2: Validation

EN ISO 12100:2010	**Safety of machinery:** General principles of design, risk assessment and risk reduction
EN ISO 10218-1:2011	**Industrial robots – Safety requirements:** Part 1: Robots **Note:** Content equivalent to **ANSI/RIA R.15.06-2012, Part 1**
EN 614-1:2006+A1:2009	**Safety of machinery:** Ergonomic design principles - Part 1: Terms and general principles
EN 61000-6-2:2005	**Electromagnetic compatibility (EMC):** Part 6-2: Generic standards; Immunity for industrial environments
EN 61000-6-4:2007 + A1:2011	**Electromagnetic compatibility (EMC):** Part 6-4: Generic standards; Emission standard for industrial environments
EN 60204-1:2006/A1:2009	**Safety of machinery:** Electrical equipment of machines - Part 1: General requirements

4 Operation

4.1 KUKA smartPAD teach pendant

4.1.1 Front view

Function

The smartPAD is the teach pendant for the industrial robot. The smartPAD has all the operator control and display functions required for operating and programming the industrial robot.

The smartPAD has a touch screen: the smartHMI can be operated with a finger or stylus. An external mouse or external keyboard is not necessary.

Overview

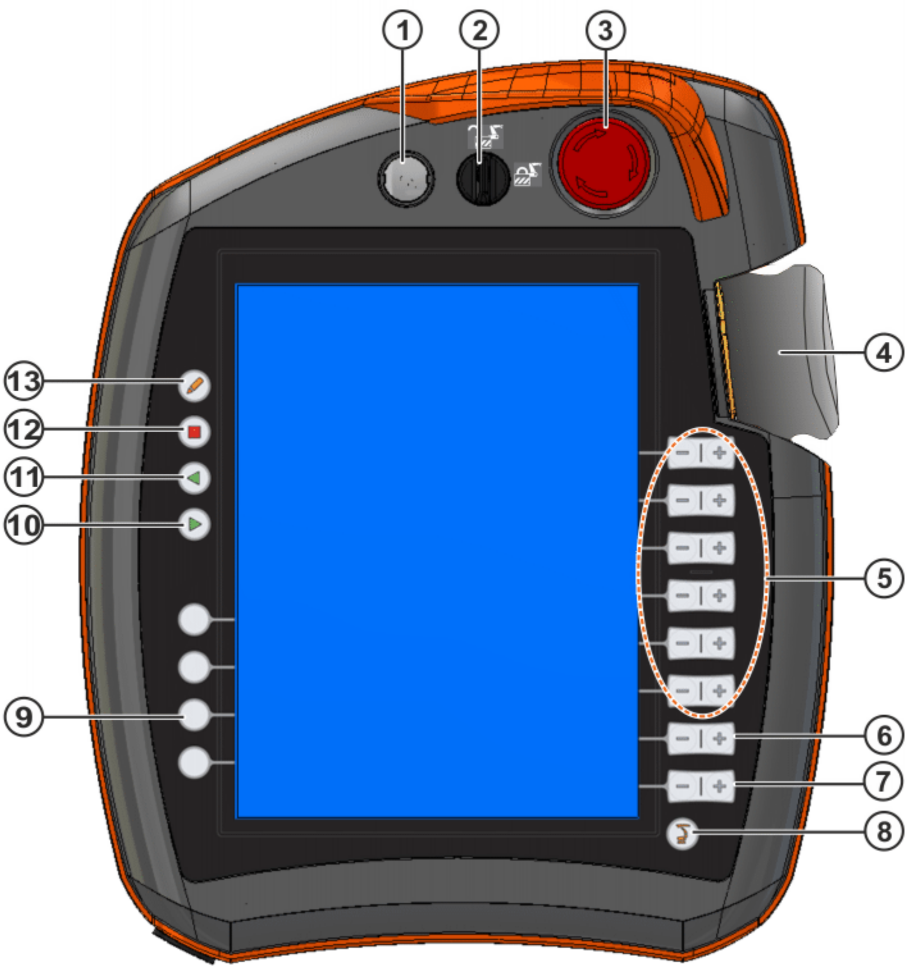


Fig. 4-1: KUKA smartPAD, front view

Item	Description
1	Button for disconnecting the smartPAD (>>> 4.1.3 "Disconnecting and connecting the smartPAD" Page 44)
2	Mode selector switch. The switch may be one of the following variants: ■ With key ■ Without key The mode selector switch is used to call the connection manager. The operating mode can be changed by using the connection manager. (>>> 4.12 "Changing operating mode" Page 59)
3	EMERGENCY STOP device Stops the robot in hazardous situations. The EMERGENCY STOP button locks itself in place when it is pressed.
4	Space Mouse: For moving the robot manually
5	Jog keys: For moving the robot manually
6	Key for setting the program override
7	Key for setting the jog override
8	Main menu key: Shows the menu items on the smartHMI. It can also be used for creating screenshots.
9	Status keys. The status keys are used primarily for setting parameters in technology packages. Their exact function depends on the technology packages installed.
10	Start key: The Start key is used to start a program.
11	Start backwards key: The Start backwards key is used to start a program backwards. The program is executed step by step.
12	STOP key: The STOP key is used to stop a program that is running.
13	Keyboard key Displays the keyboard. It is generally not necessary to press this key to display the keyboard, as the smartHMI detects when keyboard input is required and displays the keyboard automatically. (>>> 4.2.1 "Keypad" Page 46)

4.1.2 Rear view

Overview

Fig. 4-2: KUKA smartPAD, rear view

1	Enabling switch	4	USB connection
2	Start key (green)	5	Enabling switch
3	Enabling switch	6	Identification plate

Description

Element	Description
Rating plate	Rating plate
Start key	The Start key is used to start a program.
Enabling switch	The enabling switch has 3 positions: ■ Not pressed ■ Center position ■ Fully pressed (panic position) The enabling switch must be held in the center position in operating modes T1 and T2 in order to be able to jog the manipulator. In the operating modes Automatic and Automatic External, the enabling switch has no function.
USB connection	The USB connection is used, for example, for archiving and restoring data. Only for FAT32-formatted USB sticks.

4.1.3 Disconnecting and connecting the smartPAD

Description

The smartPAD can be disconnected while the robot controller is running.

> ⚠ **WARNING** If the smartPAD is disconnected, the system can no longer be switched off by means of the EMERGENCY STOP device on the smartPAD. For this reason, an external EMERGENCY STOP must be connected to the robot controller.
> The user is responsible for ensuring that the smartPAD is immediately removed from the system when it has been disconnected. The smartPAD must be stored out of sight and reach of personnel working on the industrial robot. This prevents operational and non-operational EMERGENCY STOP devices from becoming interchanged.
> Failure to observe these precautions may result in death, injuries or damage to property.

Procedure

Disconnection:

1. Press the disconnect button on the smartPAD.

   A message and a counter are displayed on the smartHMI. The counter runs for 30 s. During this time, the smartPAD can be disconnected from the robot controller.

> If the smartPAD is disconnected without the counter running, this triggers an EMERGENCY STOP. The EMERGENCY STOP can only be canceled by plugging the smartPAD back in.

2. Disconnect the smartPAD from the robot controller.

   If the counter expires without the smartPAD having been disconnected, this has no effect. The disconnect button can be pressed again at any time to display the counter again.

Connection:

- Connect the smartPAD to the robot controller.

A smartPAD can be connected at any time. Precondition: Same smartPAD variant as the disconnected device. The EMERGENCY STOP and enabling switches are operational again 30 s after connection. The smartHMI is automatically displayed again. (This may take longer than 30 s.)

The connected smartPAD assumes the current operating mode of the robot controller.

> The current operating mode is not, in all cases, the same as that before the smartPAD was disconnected: if the robot controller is part of a RoboTeam, the operating mode may have been changed after disconnection, e.g. by the master.

> ⚠ **WARNING** The user connecting a smartPAD to the robot controller must subsequently stay with the smartPAD for at least 30 s, i.e. until the EMERGENCY STOP and enabling switches are operational once again. This prevents another user from trying to activate a non-operational EMERGENCY STOP in an emergency situation, for example.
> Failure to observe this can lead to death, injury or property damage.

4.2 KUKA smartHMI user interface

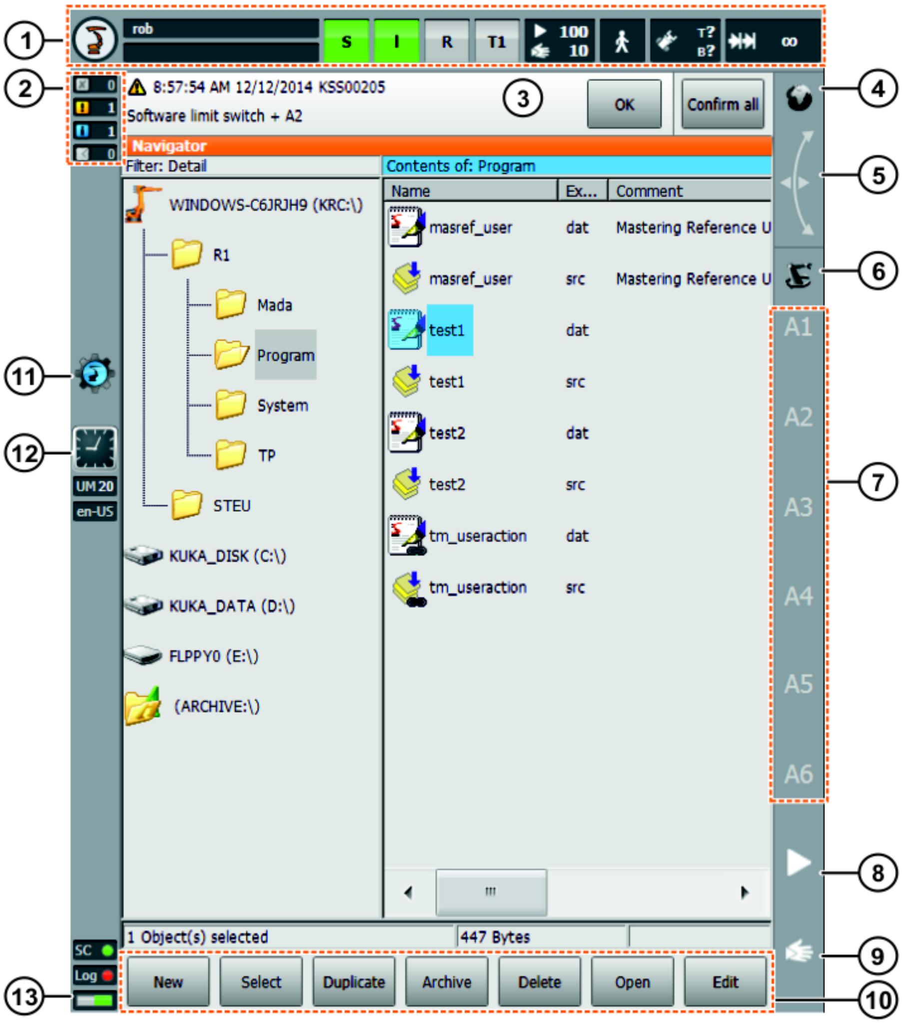


Fig. 4-3: KUKA smartHMI user interface

Item	Description
1	Status bar (>>> 4.2.2 "Status bar" Page 47)
2	Message counter The message counter indicates how many messages of each message type are active. Touching the message counter enlarges the display.
3	Message window By default, only the last message is displayed. Touching the message window expands it so that all active messages are displayed. An acknowledgeable message can be acknowledged with **OK**. All acknowledgeable messages can be acknowledged at once with **All OK**.
4	**Space Mouse** status indicator This indicator shows the current coordinate system for jogging with the Space Mouse. Touching the indicator displays all coordinate systems, allowing a different one to be selected.

Item	Description
5	**Space Mouse alignment** indicator Touching this indicator opens a window in which the current alignment of the Space Mouse is indicated and can be changed. (>>> 4.14.8 "Defining the alignment of the Space Mouse" Page 70)
6	**Jog keys** status indicator This indicator shows the current coordinate system for jogging with the jog keys. Touching the indicator displays all coordinate systems, allowing a different one to be selected.
7	Jog key labels If axis-specific jogging is selected, the axis numbers are displayed here (A1, A2, etc.). If Cartesian jogging is selected, the coordinate system axes are displayed here (X, Y, Z, A, B, C). Touching the label causes the selected kinematics group to be displayed.
8	Program override (>>> 7.5 "Setting the program override (POV)" Page 166)
9	Jog override (>>> 4.14.3 "Setting the jog override (HOV)" Page 66)
10	Button bar. The buttons change dynamically and always refer to the window that is currently active in the smartHMI. At the right-hand end is the **Edit** button. This can be used to call numerous commands relating to the Navigator.
11	WorkVisual icon Touching the icon takes you to the **Project management** window. (>>> 6.11.1 "Project management window" Page 158)
12	Clock The clock displays the system time. Touching the clock displays the system time in digital format, together with the current date.
13	Life sign display If the display flashes in the following manner, this indicates that the smartHMI is active: The left-hand and right-hand lamps alternately light up green. The change is slow (approx. 3 s) and uniform.

4.2.1 Keypad

The smartPAD has a touch screen: the smartHMI can be operated with a finger or stylus.

There is a keypad on the smartHMI for entering letters and numbers. The smartHMI detects when the entry of letters or numbers is required and automatically displays the keypad.

The keypad only ever displays the characters that are required. If, for example, a box is edited in which only numbers can be entered, then only numbers are displayed and not letters.

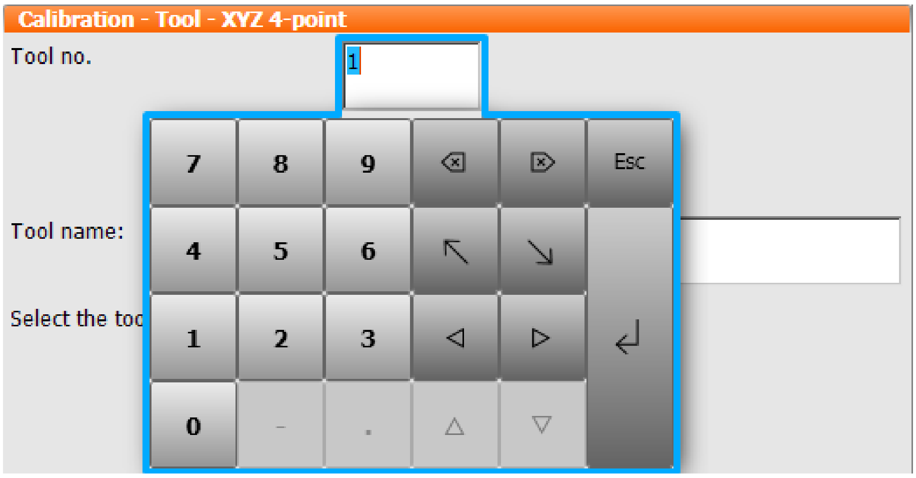


Fig. 4-4: Example keypad

4.2.2 Status bar

The status bar indicates the status of certain central settings of the industrial robot. In most cases, touching the display opens a window in which the settings can be modified.

Overview

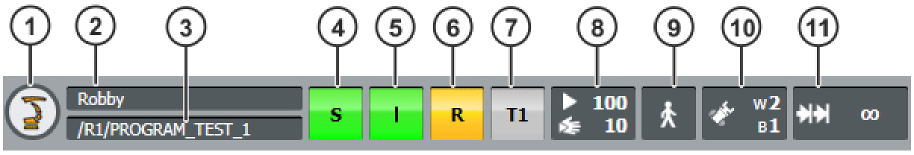
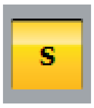
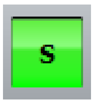
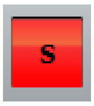
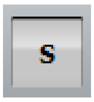


Fig. 4-5: KUKA smartHMI status bar

Item	Description
1	Main menu key. Shows the menu items on the smartHMI.
2	Robot name. The robot name can be changed.
3	If a program has been selected, the name is displayed here.
4	**Submit interpreter** status indicator
5	**Drives** status indicator. Touching the display opens a window in which the drives can be switched on or off. (>>> 4.2.4 "Drives status indicator and Motion conditions window" Page 48)
6	**Robot interpreter** status indicator. Programs can be reset or canceled here. (>>> 7.6 "Robot interpreter status indicator" Page 167)
7	Current operating mode
8	**POV/HOV** status indicator. Indicates the current program override and the current jog override.
9	**Program run mode** status indicator. Indicates the current program run mode.
10	**Tool/base** status indicator. Indicates the current tool and base.
11	**Incremental jogging** status indicator.

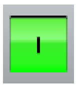
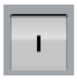
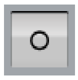
4.2.3 Submit interpreter status indicator

Icon	Color	Description
	Yellow	The submit interpreter is selected. The block pointer is situated on the first line of the selected SUB program.
	Green	A SUB program is selected and running.
	Red	The submit interpreter has been stopped.
	Gray	The submit interpreter is deselected.

4.2.4 Drives status indicator and Motion conditions window

Drives status indicator

The **Drives** status indicator can display the following statuses:

Statuses			

Meaning of the symbols and colors:

Symbol: **I**	The drives are ON. ■ The intermediate circuit is fully charged.
Symbol: **O**	The drives are OFF. ■ The intermediate circuit is not charged or incompletely charged.
Color: Green	■ The enabling switch has been pressed (center position) or is not required. ■ And: There are no active messages preventing robot motion.
Color: Gray	■ The enabling switch has not been pressed or fully pressed. ■ And/or: There are active messages preventing robot motion.

> ■ Drives ON does not automatically mean that the KSPs switch to servo-control and supply the motors with current.
>
> ■ Drives OFF does not automatically mean that the KSPs terminate the power supply to the motors.
>
> Whether or not the KSPs supply the motors with current depends on whether the drives enable signal has been received from the safety controller.

"Motion conditions" window

Touching the **Drives** status indicator opens the **Motion conditions** window. The drives can be switched on or off here.

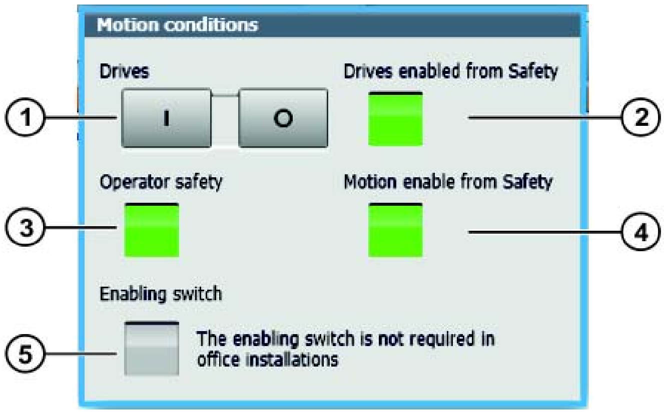


Fig. 4-6: "Motion conditions" window

Item	Description
1	**I**: Touch to switch on the drives. **O**: Touch to switch off the drives.
2	**Green**: The drives enable signal has been received from the safety controller. **Gray**: The safety controller has triggered a safety stop 0 or terminated a safety stop 1. No drives enable signal present, i.e. the KSPs are not under servo-control and are not supplying the motors with current.
3	**Green**: Signal **Operator safety** == TRUE **Gray**: Signal **Operator safety** == FALSE (>>> "Signal Operator safety == TRUE" Page 49)
4	**Green**: The motion enable signal has been received from the safety controller. **Gray**: The safety controller has triggered a safety stop 1 or a safety stop 2. No motion enable present.
5	**Green**: The enabling switch is pressed (center position). **Gray**: The enabling switch has not been pressed or fully pressed, or is not required.

Signal Operator safety == TRUE

The conditions under which the signal **Operator safety** is TRUE depend on the controller variant and the operating mode:

Controller	Operating mode	Condition
KR C4	T1, T2	■ The enabling switch is pressed.
	AUT, AUT EXT	■ The physical safeguard is closed.
VKR C4	T1	■ Enabling switch is pressed ■ E2 is closed
	T2	■ The enabling switch is pressed. ■ E2 and E7 are closed.
	AUT EXT	■ The physical safeguard is closed. ■ E2 and E7 are open.

4.3 Switching on the robot controller and starting the KSS

Procedure

- Turn the main switch on the robot controller to ON.

  The operating system and the KSS start automatically.

If the KSS does not start automatically, e.g. because the Startup function has been disabled, execute the file StartKRC.exe in the directory C:\KRC.

If the robot controller is logged onto the network, the start may take longer.

4.4 Calling the main menu

Procedure

- Press the main menu key on the smartPAD. The **Main menu** window opens.

  The display is always the same as that which was in the window before it was last closed.

Description

Properties of the **Main Menu** window:

- The main menu is displayed in the left-hand column.
- Touching a menu item that contains an arrow opens the corresponding submenu (e.g. **Configure**).

  Depending on how many nested submenus are open, the **Main Menu** column may no longer be visible, with only the submenus remaining visible.
- The arrow key in the top right-hand corner closes the most recently opened submenu.
- The Home key in the top right-hand corner closes all open submenus.
- The most recently selected menu items are displayed in the bottom section (maximum 6).

  This makes it possible to select these menu items again directly without first having to close other submenus that might be open.
- The white cross on the left-hand side closes the window.

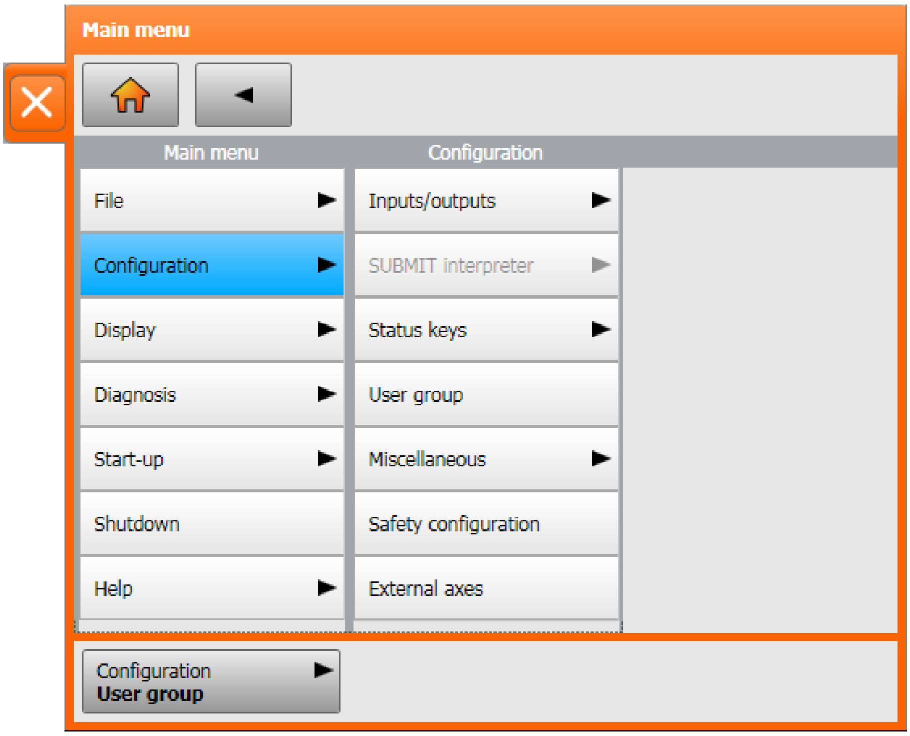


Fig. 4-7: Example: Configuration submenu is open.

4.5 Exiting or restarting KSS

Description

The KSS starts in whatever operating mode was most recently selected. Exceptions:

- If the most recent operating mode was T2, the mode after starting is T1.
- After an initial cold start, the operating mode is T1.

Precondition

- For certain options: “Expert” user group

> **NOTICE** If, on shutting down, the option **Reboot control PC** is selected, the main switch on the robot controller must not be pressed until the reboot has been completed. System files may otherwise be destroyed.
> If this option was not selected on shutting down, the main switch can be pressed once the controller has shut down.

Procedure

1. Select the menu item **Shutdown** in the main menu.
2. Select the desired options.
3. Press **Shut down control PC** or **Reboot control PC**.
4. Confirm the request for confirmation with **Yes**. The System Software is terminated and restarted in accordance with the selected option.

After the restart, the following message is displayed:

- *Cold start of controller*
- Or, if **Reload files** has been selected: *Initial cold start of controller*

“Shutdown” window

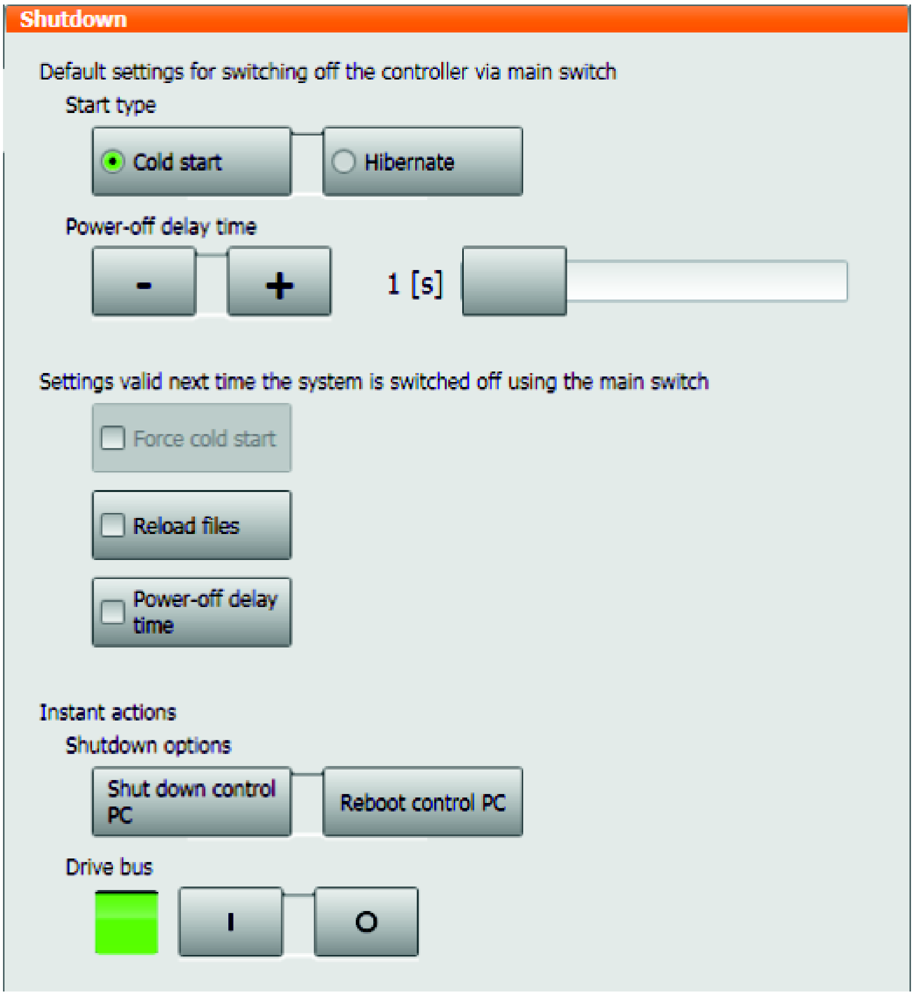


Fig. 4-8: “Shutdown” window

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td colspan="2">Default settings for switching off the system

These settings can only be modified in the user group “Expert”.</td></tr>
<tr><td>Cold start</td><td>Cold start is the standard start type. (>>> "Start types" Page 53)</td></tr>
<tr><td>Hibernate</td><td>Hibernate is the standard start type.</td></tr>
<tr><td>Power-off delay time</td><td>If the robot controller is switched off at the main switch, it is only shut down after the delay time defined here. During the delay time, the robot controller is powered by its battery.

The delay time can only be modified in the user group “Expert”.

Note:
■ The power-off delay time only applies if the voltage is switched off via the main switch. The power failure delay time applies for genuine power failures.
■ Exception for “KR C4 compact”: The power-off delay time has no function for this controller variant! The power failure delay time applies here, even when switching off via the main switch.

(>>> 4.5.1 "Shutting down after power failure" Page 53)</td></tr>
<tr><td colspan="2">Settings that are only valid next time the system is switched off</td></tr>
<tr><td>Force cold start</td><td>Activated: The next start is a cold start.

Only available if Hibernate has been selected.</td></tr>
<tr><td>Reload files</td><td>Activated: The next start is an initial cold start.

This option must be selected in the following cases:
■ If XML files have been changed directly, i.e. the user has opened the file and modified it.
(Any other changes to XML files, e.g. if the robot controller modifies them in the background, are irrelevant.)
■ If hardware components are to be exchanged after shutdown.

Can only be selected in the “Expert” user group. Only available if Cold start or Force cold start has been selected.

Depending on the hardware, the initial cold start takes approx. 30 to 150 seconds longer than a normal cold start.</td></tr>
<tr><td>Power-off delay time</td><td>Activated: The delay time is adhered to the next time the system is shut down.

Deactivated: The delay time is ignored the next time the system is shut down.</td></tr>
<tr><td colspan="2">Instant actions

Only available in operating modes T1 and T2.</td></tr>
<tr><td>Shut down control PC</td><td>The robot controller shuts down.</td></tr>
<tr><td>Reboot control PC</td><td>The robot controller shuts down and then reboots with a cold start.</td></tr>
<tr><td>Drive bus

OFF / ON</td><td>The drive bus can be switched off or on.

Drive bus status indicator:
■ Green: Drive bus is on.
■ Red: Drive bus is off.
■ Gray: Status of the drive bus is unknown.</td></tr>
</table>

Start types

Start type	Description
Cold start	After a cold start the robot controller displays the Navigator. No program is selected. The robot controller is reinitialized, e.g. all user outputs are set to FALSE. **Note:** If XML files have been changed directly, i.e. the user has opened the file and modified it, these changes are taken into consideration in the case of a cold start with **Reload files**. This cold start is called an “initial cold start”. In the case of a cold start without **Reload files**, these changes are not taken into consideration.
Hibernate	After a start with Hibernate, the previously selected robot program can be resumed. The state of the kernel system: programs, block pointer, variable contents and outputs, is completely restored. Additionally, all programs that were open parallel to the robot controller are reopened and have the same state that they had before the system was shut down. The last state of Windows is also restored.

4.5.1 Shutting down after power failure

In the case of a power failure, the robot comes to a standstill. However, the robot controller does not shut down immediately but rather only after the power failure delay time. In other words, brief power failures are overridden through this delay time. The error messages must then only be acknowledged and the program can be resumed.

During the delay time, the robot controller is powered by its battery.

Robot controller	Power failure delay time
“KR C4 compact” variant	1 s
All other variants of KR C4	3 s

If the power failure lasts longer than the power failure delay time and the robot controller shuts down, then the standard start type defined in the **Shutdown** window applies for the restart.

- The power failure delay time does not apply if the voltage is switched off via the main switch. The power-off delay time applies for this.

  Exception for “KR C4 compact”: For this controller variant, the power failure delay time also applies when switching off via the main switch.
- The power failure delay time is particularly important for systems without a reliable mains supply. Delay times of up to 240 s are possible. If the existing times are to be modified, please contact KUKA Deutschland GmbH.

4.6 Switching drives on/off

Procedure

1. In the status bar, touch the **Drives** status indicator. The **Motion conditions** window opens.

   (>>> 4.2.4 "Drives status indicator and Motion conditions window" Page 48)

2. Switch the drives on or off.

4.7 Switching the robot controller off

Description

When the system is switched off, the robot stops and the robot controller shuts down. The robot controller automatically backs up data.

If a power-off delay time is configured, the robot controller shuts down only after this time has passed. In other words, brief power-downs are overriden through this delay time. The error messages must then only be acknowledged and the program can then be resumed.

During the delay time, the robot controller is powered by its battery.

Procedure

- Turn the main switch on the robot controller to OFF.

NOTICE The main switch must not be operated if the robot controller has been exited with the option **Reboot control PC** and the reboot has not yet been completed. System files may otherwise be destroyed.

4.8 Setting the user interface language

Procedure

1. In the main menu, select **Configuration** > **Miscellaneous** > **Language**.
2. Select the desired language. Confirm with **OK**.

Description

The following languages are available:

Chinese (simplified)	Polish
Danish	Portuguese
German	Romanian
English	Russian
Finnish	Swedish
French	Slovak
Greek	Slovenian
Italian	Spanish
Japanese	Czech
Korean	Turkish
Dutch	Hungarian

4.9 Creating a screenshot on the smartPAD

Procedure

- Press the main menu key in the bottom right-hand corner of the smartPAD twice.

The screenshot is saved in the directory C:/KUKA/Screenshot.

4.10 Online documentation and help for messages

4.10.1 Calling online documentation

Description

The documentation of the System Software can be displayed on the robot controller. Certain technology packages also have documentation that can be displayed on the robot controller.

Procedure

1. In the main menu, select **Help** > **Documentation**. Then select either **System Software** or the menu item for the technology package.

   The **KUKA Embedded Information Service** window is opened. The table of contents of the documentation is displayed.
2. Touch a chapter. The topics it contains are displayed.
3. Touch a topic. The description is displayed.

Example

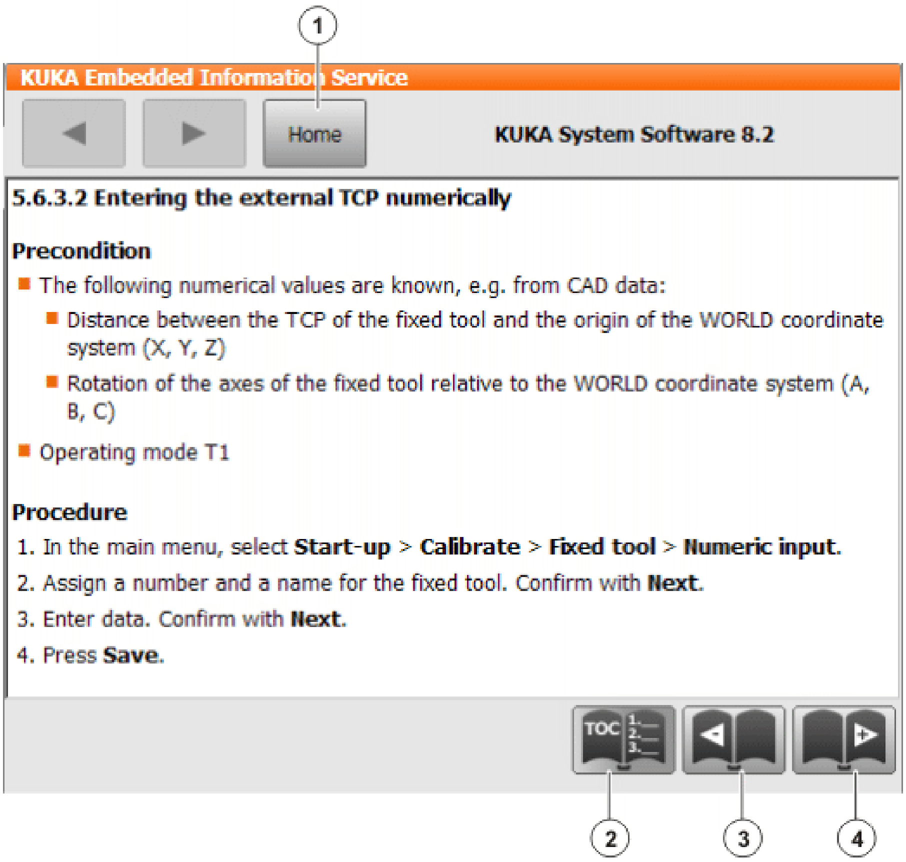


Fig. 4-9: Online documentation – Example from the KUKA System Software

Item	Description
1, 2	Displays the table of contents.
3	Displays the previous topic in the table of contents.
4	Displays the next topic.

4.10.2 Calling help for the messages

Description

The help for a message can be called in the following ways:

- Call the help for a specific message that is currently displayed in the message window.
- Display an overview of the possible messages and call the help for a message there.

Procedure

Calling the help for a message in the message window

Most messages contain a button with a question mark. Help is available for these messages.

1. Touch the question mark. The **KUKA Embedded Information Service – Message page** window is opened.

The window contains a variety of information about the message. (>>> Fig. 4-10)

2. The window often also contains information about the causes of the message and the corresponding solutions. Details can be displayed:
   a. Touch the magnifying glass icon next to the cause. The detail page is opened. (>>> Fig. 4-11)
   b. Open the descriptions of the cause and solution.
   c. If the message has several possible causes: the magnifying glass icons with arrows can be used to jump to the previous or next detail page.

Procedure

Display an overview of the messages and call the help for a message.

1. In the main menu, select **Help** > **Messages**. Then select either **System Software** or the menu item for the technology package.

   The **KUKA Embedded Information Service** – **Index page** window is opened. The messages are sorted by module ("module" refers here to a subsection of the software).
2. Touch an entry. The messages of this module are displayed.
3. Touch a message. The message page is displayed.

   The window contains a variety of information about the message. (>>> Fig. 4-10)
4. The window often also contains information about the causes of the message and the corresponding solutions. Details can be displayed:
   a. Touch the magnifying glass icon next to the cause. The detail page is opened. (>>> Fig. 4-11)
   b. Open the descriptions of the cause and solution.
   c. If the message has several possible causes: the magnifying glass icons with arrows can be used to jump to the previous or next detail page.

Message page

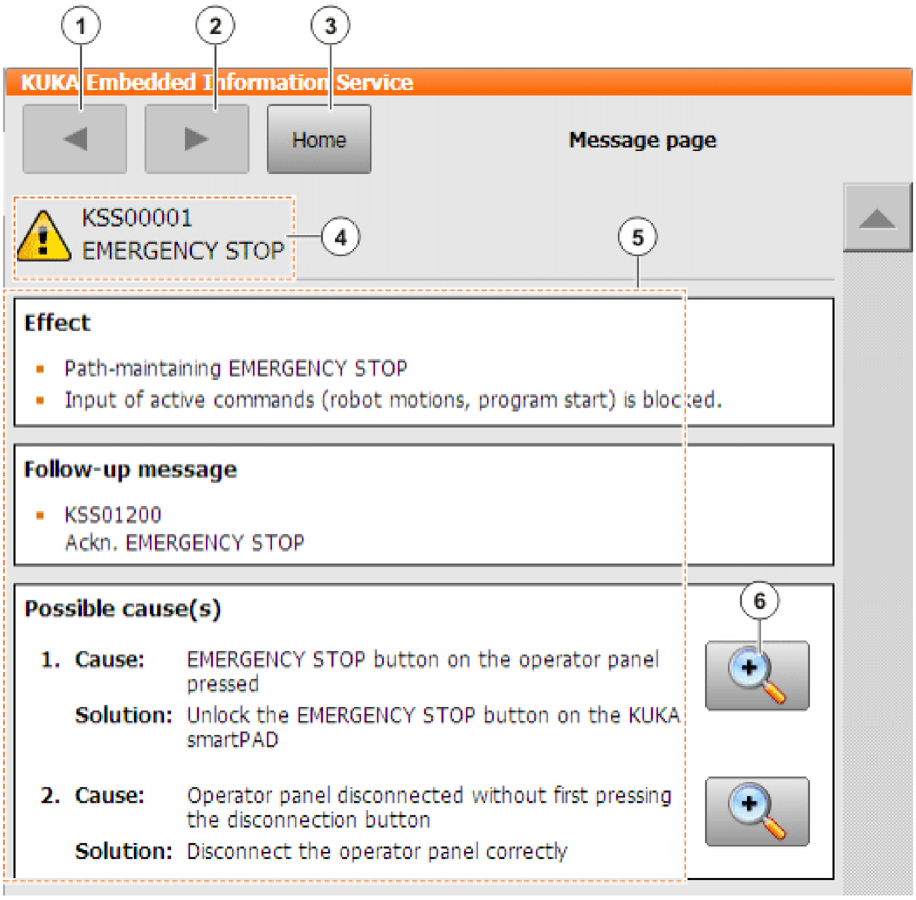


Fig. 4-10: Message page – Example from the KUKA System Software

Item	Description
1	Displays the previous page.
2	This button is only active if the other arrow button has been used to jump to the previous page. This button can then be used to return to the original page.
3	Displays the list with the software modules.
4	Message number and text
5	Information about the message There may be less information available than in the example.
6	Displays details about this cause/solution. (>>> Fig. 4-11)

Detail page

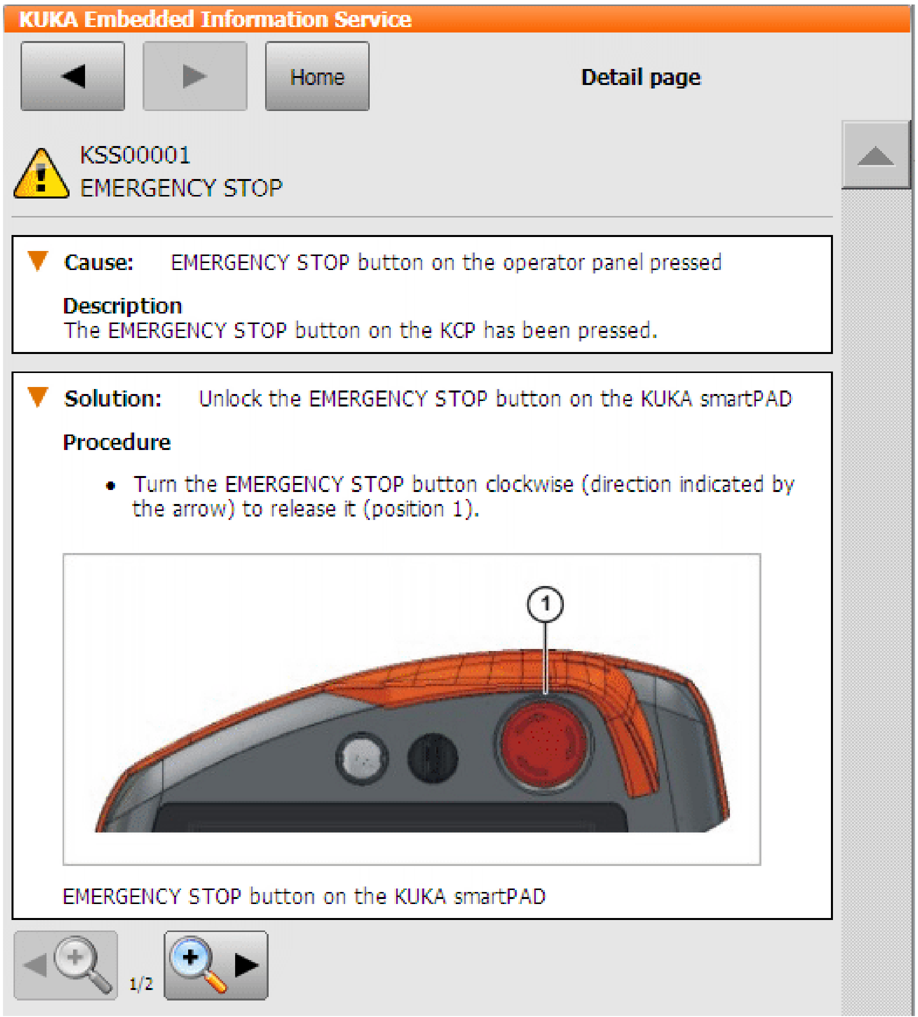


Fig. 4-11: Detail page – Example from the KUKA System Software

4.11 Changing user group

Procedure

1. Select **Configuration** > **User group** in the main menu. The current user group is displayed.
2. Press **Default** to switch to the default user group. (**Default** is not available if the default user group is already selected.)

   Press **Login...** to switch to a different user group. Select the desired user group.
3. If prompted: Enter password and confirm with **Log-on**.

Description

Different functions are available in the KSS, depending on the user group. The following user groups are available:

- **Operator**

  User group for the operator. This is the default user group.
- **User**

  User group for the operator. (By default, the user groups “Operator” and “User” are defined for the same target group.)
- **Expert**

  User group for the programmer. This user group is protected by means of a password.

- **Safety recovery**

  User group for the start-up technician. This user can activate and configure the safety configuration of the robot.

  This user group is protected by means of a password.
- **Safety maintenance**

  This user group is only relevant if KUKA.SafeOperation or KUKA.SafeRangeMonitoring is used. The user group is protected by means of a password.
- **Administrator**

  The range of functions is the same as that for the user group “Expert”. It is additionally possible, in this user group, to integrate plug-ins into the robot controller.

  This user group is protected by means of a password.

The default password is “kuka”.

When the system is booted, the default user group is selected.

If the mode is switched to AUT or AUT EXT, the robot controller switches to the default user group for safety reasons. If a different user group is desired, this must be selected subsequently.

If no actions are carried out in the user interface within a certain period of time, the robot controller switches to the default user group for safety reasons. The default setting is 300 s.

4.12 Changing operating mode

> Do not change the operating mode while a program is running. If the operating mode is changed during program execution, the industrial robot is stopped with a safety stop 2.

Precondition

- The robot controller is not executing a program.
- If the mode selector switch is the variant with a key: the key is inserted in the switch.

Procedure

1. Turn the mode selector switch on the smartPAD. The connection manager is displayed.
2. Select the operating mode. (>>> 3.5.3 "Selecting the operating mode" Page 21)
3. Return the mode selector switch to its original position.

   The selected operating mode is displayed in the status bar of the smartPAD.

Operating mode	Use	Velocities
T1	For test operation, programming and teaching	■ Program verification: Programmed velocity, maximum 250 mm/s ■ Jog mode: Jog velocity, maximum 250 mm/s
T2	For test operation	■ Program verification: Programmed velocity ■ Jog mode: Not possible

Operating mode	Use	Velocities
AUT	For industrial robots without higher-level controllers	■ Program operation: Programmed velocity ■ Jog mode: Not possible
AUT EXT	For industrial robots with higher-level controllers, e.g. PLC	■ Program operation: Programmed velocity ■ Jog mode: Not possible

4.13 Coordinate systems

Overview

The following Cartesian coordinate systems are defined in the robot controller:

- WORLD
- ROBROOT
- BASE
- TOOL

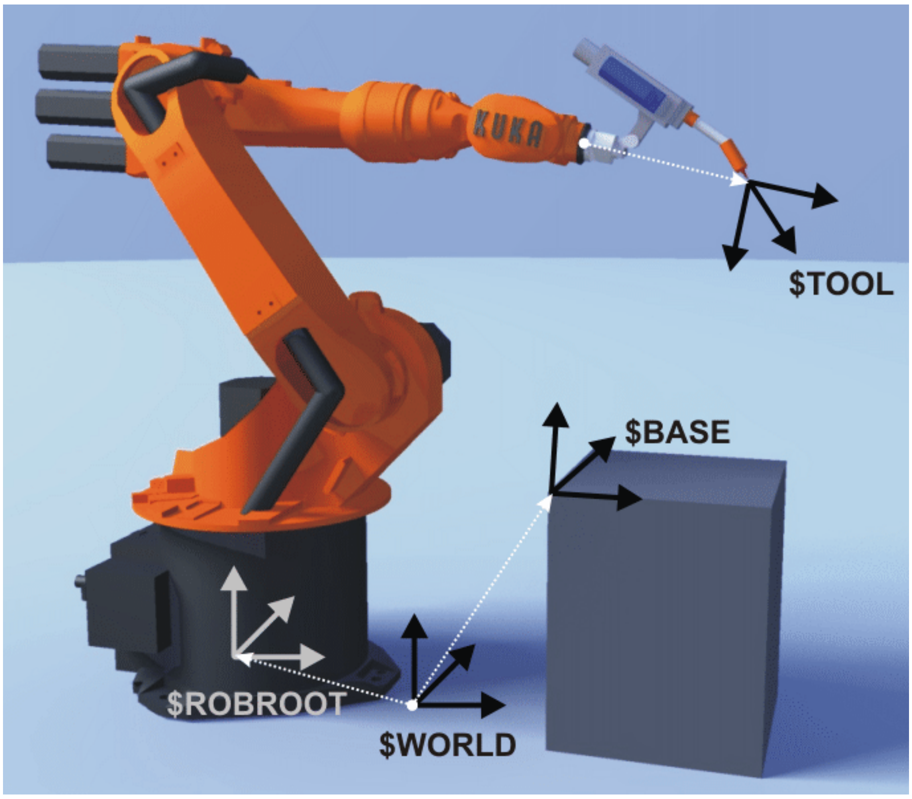


Fig. 4-12: Overview of coordinate systems

Description

WORLD

The WORLD coordinate system is a permanently defined Cartesian coordinate system. It is the root coordinate system for the ROBROOT and BASE coordinate systems.

By default, the WORLD coordinate system is located at the robot base.

ROBROOT

The ROBROOT coordinate system is a Cartesian coordinate system, which is always located at the robot base. It defines the position of the robot relative to the WORLD coordinate system.

By default, the ROBROOT coordinate system is identical to the WORLD coordinate system. $ROBROOT allows the definition of an offset of the robot relative to the WORLD coordinate system.

BASE

The BASE coordinate system is a Cartesian coordinate system that defines the position of the workpiece. It is relative to the WORLD coordinate system.

By default, the BASE coordinate system is identical to the WORLD coordinate system. It is offset to the workpiece by the user.

(>>> 5.6.2 "Base calibration" Page 118)

TOOL

The TOOL coordinate system is a Cartesian coordinate system which is located at the tool center point.

As standard, the origin of the TOOL coordinate system is located at the flange center point. (In this case it is called the FLANGE coordinate system.) The TOOL coordinate system is offset to the tool center point by the user.

(>>> 5.6.1 "Tool calibration" Page 112)

Angles of rotation of the robot coordinate systems

Angle	Rotation about axis
Angle A	Rotation about the Z axis
Angle B	Rotation about the Y axis
Angle C	Rotation about the X axis

4.14 Jogging the robot

Description

There are 2 ways of jogging the robot:

- Cartesian jogging

  The TCP is jogged in the positive or negative direction along the axes of a coordinate system.
- Axis-specific jogging

  Each axis can be moved individually in the positive or negative direction.

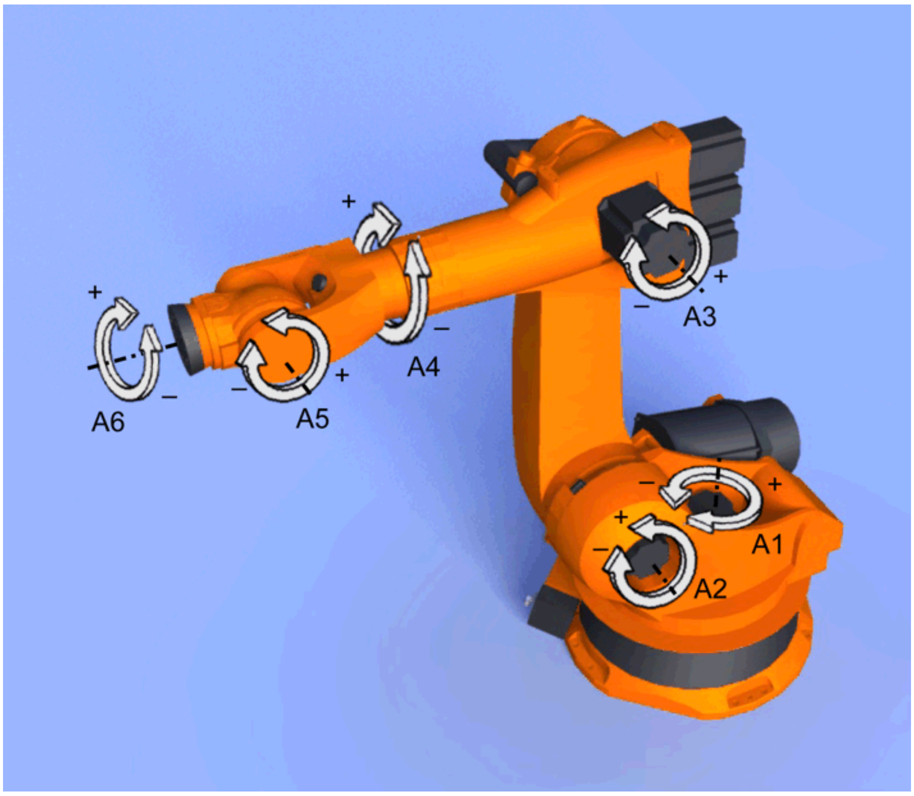


Fig. 4-13: Axis-specific jogging

There are 2 operator control elements that can be used for jogging the robot:

- Jog keys
- Space Mouse

While the robot is being jogged using the keys, the Space Mouse is disabled until the robot comes to a standstill. While the Space Mouse is actuated, the keys are disabled.

Overview

	Cartesian jogging	Axis-specific jogging
Jog keys	(>>> 4.14.6 "Cartesian jogging with the jog keys" Page 67)	(>>> 4.14.5 "Axis-specific jogging with the jog keys" Page 67)
Space Mouse	(>>> 4.14.9 "Cartesian jogging with the Space Mouse" Page 71)	Axis-specific jogging with the Space Mouse is possible, but is not described here.

4.14.1 “Jog options” window

Description

All parameters for jogging the robot can be set in the **Jogging Options** window.

Procedure

To open the **Jogging options** window:

1. Open a status indicator on the smartHMI, e.g. the **Cur. tool/base** status indicator.

   (Not possible for the **Submit interpreter**, **Drives** and **Robot interpreter** status indicators.)

   A window opens.
2. Press **Options**. The **Jogging Options** window is opened.

For most parameters, it is not necessary to open the **Jogging Options** window. They can be set directly via the smartHMI status indicators.

4.14.1.1 “General” tab

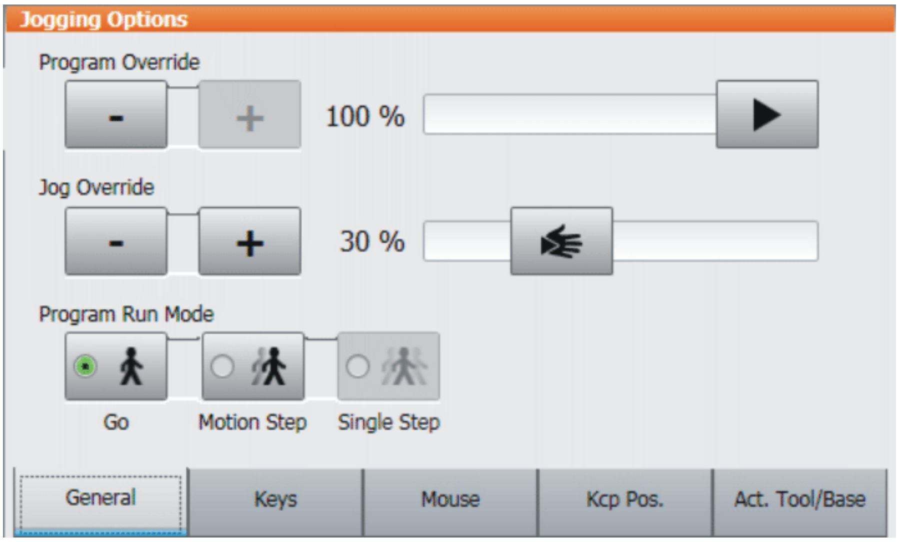


Fig. 4-14: General tab

Description

Item	Description
1	Set the program override. (>>> 7.5 "Setting the program override (POV)" Page 166)
2	Set the jog override. (>>> 4.14.3 "Setting the jog override (HOV)" Page 66)
3	Select the program run mode. (>>> 7.2 "Program run modes" Page 163)

4.14.1.2 “Keys” tab

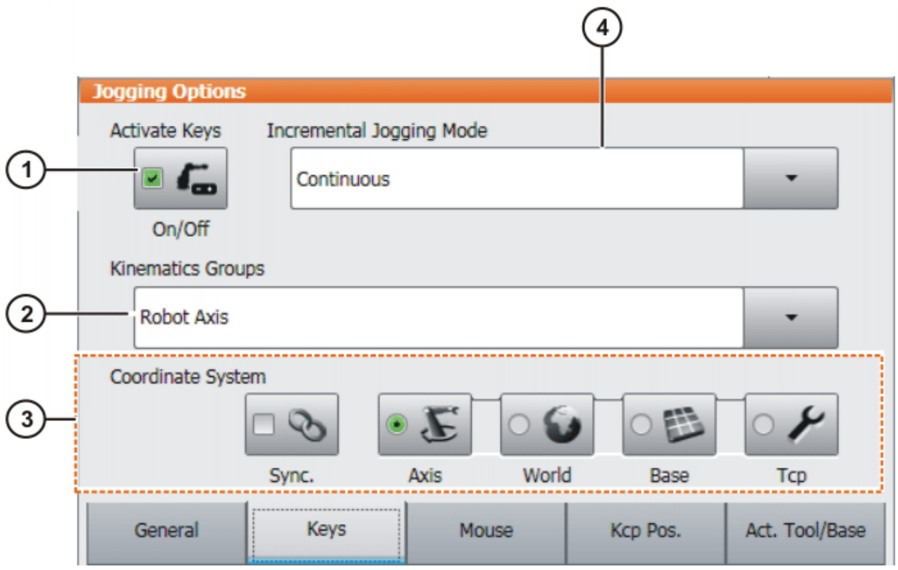


Fig. 4-15: Keys tab

Description

Item	Description
1	Activate jog mode “Jog keys” (>>> 4.14.2 "Activating the jog mode" Page 66)
2	Select a kinematics group. The kinematics group defines the axes to which the jog keys refer. Default: **Robot axes** (= A1 to A6) Depending on the system configuration, other kinematics groups may be available. (>>> 4.15 "Jogging external axes" Page 72)
3	Select the coordinate system for jogging with the jog keys: ■ **Axes**, **World**, **Base** or **Tool** Check box **Sync.**: ■ Check box not active (default): On the **Keys** and **Mouse** tabs, different coordinate systems can be selected. ■ Check box active: On the **Keys** and **Mouse** tabs, only one coordinate system can be selected, which is the same in each case. If the coordinate system is changed on one tab, the setting on the other is adapted automatically.
4	Incremental jogging (>>> 4.14.10 "Incremental jogging" Page 71)

4.14.1.3 “Mouse” tab

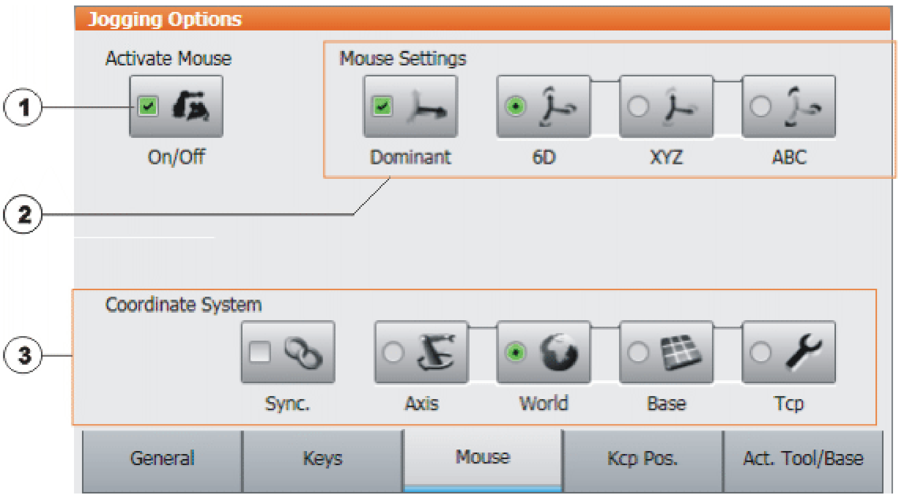


Fig. 4-16: Mouse tab

Description

Item	Description
1	Activate jog mode “Space Mouse” (>>> 4.14.2 "Activating the jog mode" Page 66)

Item	Description
2	Configure the Space Mouse (>>> 4.14.7 "Configuring the Space Mouse" Page 68)
3	Select the coordinate system for jogging with the Space Mouse: ■ **Axes**, **World**, **Base** or **Tool** Check box **Sync.**: ■ Check box not active (default): On the **Keys** and **Mouse** tabs, different coordinate systems can be selected. ■ Check box active: On the **Keys** and **Mouse** tabs, only one coordinate system can be selected, which is the same in each case. If the coordinate system is changed on one tab, the setting on the other is adapted automatically.

4.14.1.4 “KCP pos.” tab

Fig. 4-17: Kcp Pos. tab

Description

Item	Description
1	(>>> 4.14.8 "Defining the alignment of the Space Mouse" Page 70)

4.14.1.5 “Cur. tool/base” tab

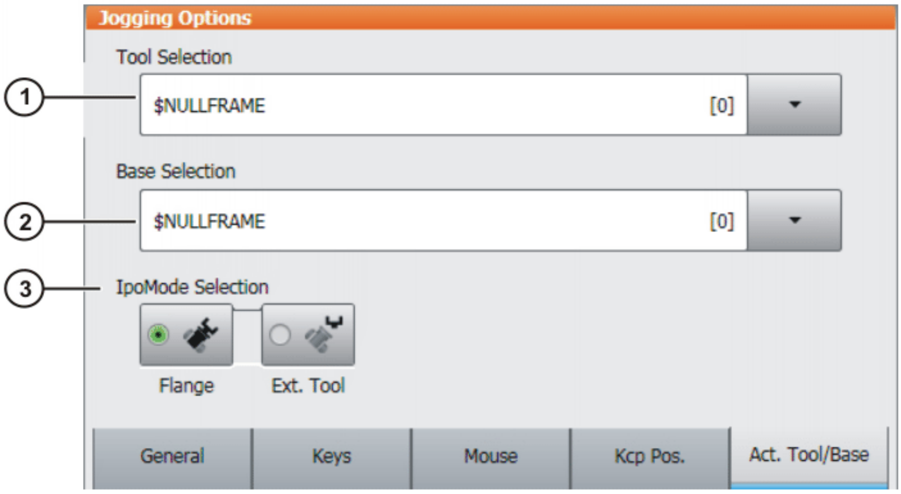


Fig. 4-18: Act. Tool/Base tab

Description

Item	Description
1	The current tool is displayed here. A different tool can be selected. (>>> 4.14.4 "Selecting the tool and base" Page 67) The display **Unknown [?]** means that no tool has yet been calibrated.
2	The current base is displayed here. A different base can be selected. (>>> 4.14.4 "Selecting the tool and base" Page 67) The display **Unknown [?]** means that no base has yet been calibrated.
3	Select the interpolation mode: ■ **Flange**: The tool is mounted on the mounting flange. ■ **ext. Tool**: The tool is a fixed tool.

4.14.2 Activating the jog mode

Procedure

1. Open the **Jogging Options** window.
   (>>> 4.14.1 "“Jog options” window" Page 62)
2. To activate the jog mode “Jog keys”:
   On the **Keys** tab, activate the **Activate Keys** check box.
   To activate the jog mode “Space Mouse”:
   On the **Mouse** tab, activate the **Activate Mouse** check box.

Description

Both jog modes “Jog keys” and “Space Mouse” can be activated simultaneously. If the robot is jogged using the keys, the Space Mouse is disabled until the robot comes to a standstill. If the Space Mouse is actuated, the keys are disabled.

4.14.3 Setting the jog override (HOV)

Description

Jog override determines the velocity of the robot during jogging. The velocity actually achieved by the robot with a jog override setting of 100% depends on various factors, including the robot type. The velocity cannot exceed 250 mm/s however.

Procedure

1. Touch the **POV/HOV** status indicator. The **Overrides** window is opened.
2. Set the desired jog override. It can be set using either the plus/minus keys or by means of the slider.
   - Plus/minus keys: The value can be set to 100%, 75%, 50%, 30%, 10%, 3%, 1%
   - Slider: The override can be adjusted in 1% steps.
3. Touch the **POV/HOV** status indicator again. (Or touch the area outside the window.)

   The window closes and the selected override value is applied.

> The **Jog options** window can be opened via **Options** in the **Overrides** window.

Alternative procedure

Alternatively, the override can be set using the plus/minus key on the lower right-hand side of the smartPAD.

4.14.4 Selecting the tool and base

Description

One tool (TOOL coordinate system) and one base (BASE coordinate system) must be selected for Cartesian jogging.

Procedure

1. Touch the status indicator **Cur. tool/base**. The **Cur. tool/base** window opens.
2. Select the desired tool and base.
3. The window closes and the selection is applied.

4.14.5 Axis-specific jogging with the jog keys

Precondition

- The jog mode “Jog keys” is active.
- Operating mode T1

Procedure

1. Select **Axes** as the coordinate system for the jog keys.
2. Set jog override.
3. Hold down the enabling switch.

   Axes A1 to A6 are displayed next to the jog keys.
4. Press the Plus or Minus jog key to move an axis in the positive or negative direction.

> The position of the robot during jogging can be displayed: select **Display** > **Actual position** in the main menu.

4.14.6 Cartesian jogging with the jog keys

Precondition

- The jog mode “Jog keys” is active.
- Operating mode T1
- Tool and base have been selected.

  (>>> 4.14.4 "Selecting the tool and base" Page 67)

Procedure

1. Select **World**, **Base** or **Tool** as the coordinate system for the jog keys.
2. Set jog override.
3. Hold down the enabling switch.

The following designations are displayed next to the jog keys:

- **X**, **Y**, **Z**: for the linear motions along the axes of the selected coordinate system
- **A**, **B**, **C**: for the rotational motions about the axes of the selected coordinate system

4. Press the Plus or Minus jog key to move the robot in the positive or negative direction.

> The position of the robot during jogging can be displayed: select **Display** > **Actual position** in the main menu.

4.14.7 Configuring the Space Mouse

Procedure

1. Open the **Jog options** window and select the **Mouse** tab.
   (>>> 4.14.1 "“Jog options” window" Page 62)
2. Group **Mouse settings**:
   - **Dominant** check box:
     Activate or deactivate dominant mode as desired.
   - **6D**/**XYZ**/**ABC** option box:
     Select whether the TCP is to be moved using translational motions, rotational motions, or both.
3. Close the **Jog options** window.

Description

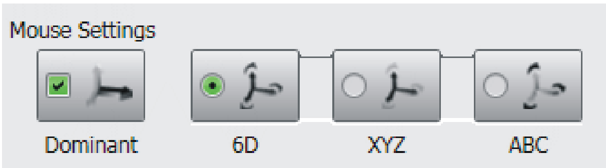


Fig. 4-19: Mouse settings

Dominant check box:

Depending on the dominant mode, the Space Mouse can be used to move just one axis or several axes simultaneously.

Check box	Description
Active	Dominant mode is activated. Only the coordinate axis with the greatest deflection of the Space Mouse is moved.
Inactive	Dominant mode is deactivated. Depending on the axis selection, either 3 or 6 axes can be moved simultaneously.

Option	Description
6D	The robot can be moved by pulling, pushing, rotating or tilting the Space Mouse. The following motions are possible with Cartesian jogging: ■ Translational motions in the X, Y and Z directions ■ Rotational motions about the X, Y and Z axes
XYZ	The robot can only be moved by pulling or pushing the Space Mouse. The following motions are possible with Cartesian jogging: ■ Translational motions in the X, Y and Z directions
ABC	The robot can only be moved by rotating or tilting the Space Mouse. The following motions are possible with Cartesian jogging: ■ Rotational motions about the X, Y and Z axes

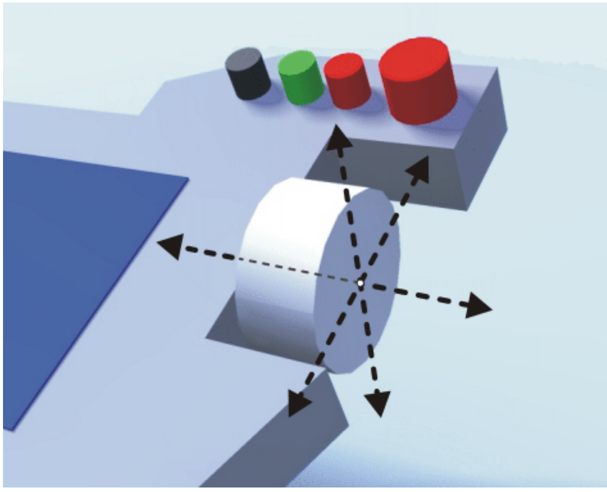


Fig. 4-20: Pushing and pulling the Space Mouse

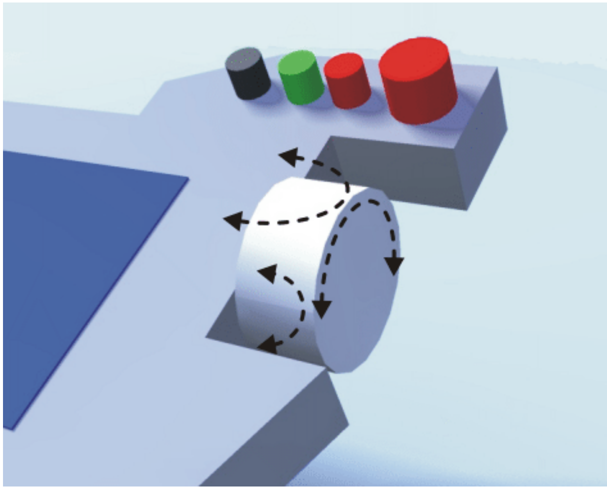


Fig. 4-21: Rotating and tilting the Space Mouse

4.14.8 Defining the alignment of the Space Mouse

Description

The functioning of the Space Mouse can be adapted to the location of the user so that the motion direction of the TCP corresponds to the deflection of the Space Mouse.

The location of the user is specified in degrees. The reference point for the specification in degrees is the junction box on the base frame. The position of the robot arm or axes is irrelevant.

Default setting: 0°. This corresponds to a user standing opposite the junction box.

Switching to Automatic External mode automatically resets the alignment of the Space Mouse to 0°.

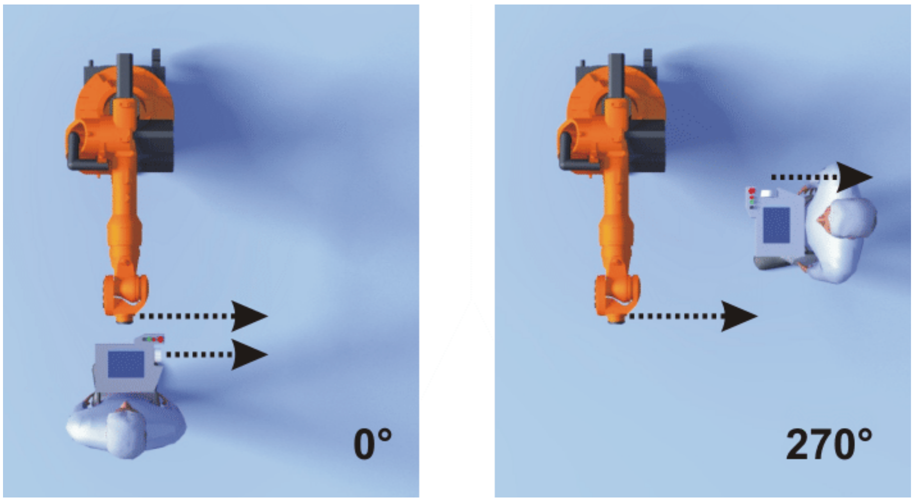


Fig. 4-22: 6D mouse: 0° and 270°

Precondition

- Operating mode T1

Procedure

1. Open the **Jog options** window and select the **KCP pos.** tab.

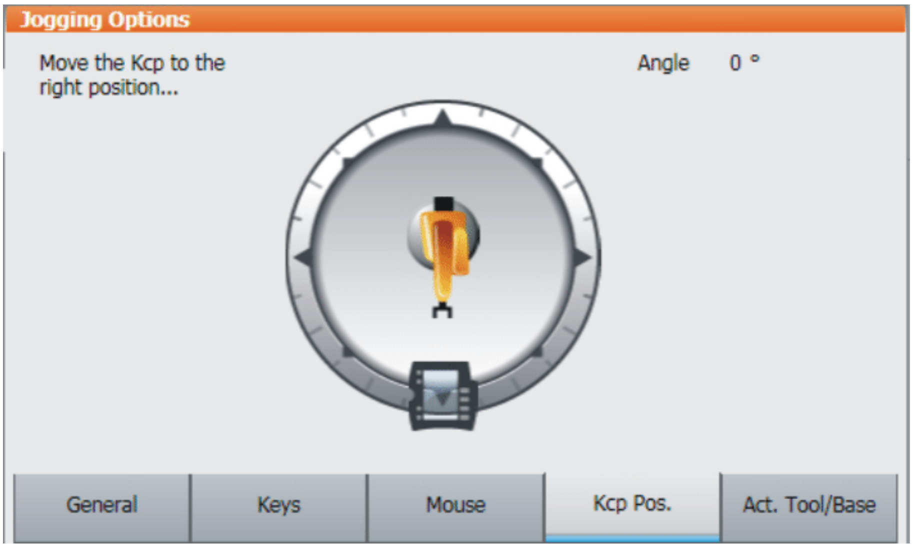


Fig. 4-23: Defining the alignment of the Space Mouse

2. Drag the smartPAD to the position corresponding to the location of the user (in 45° steps).
3. Close the **Jog options** window.

4.14.9 Cartesian jogging with the Space Mouse

Precondition

- The jog mode “Space Mouse” is active.
- T1 mode
- Tool and base have been selected.
  (>>> 4.14.4 "Selecting the tool and base" Page 67)
- The Space Mouse is configured.
  (>>> 4.14.7 "Configuring the Space Mouse" Page 68)
- The alignment of the Space Mouse has been defined.
  (>>> 4.14.8 "Defining the alignment of the Space Mouse" Page 70)

Procedure

1. Select **World**, **Base** or **Tool** as the coordinate system for the Space Mouse.
2. Set the jog override.
3. Press and hold down the enabling switch.
4. Move the robot in the desired direction using the Space Mouse.

> i The position of the robot during jogging can be displayed: select **Display** > **Actual position** in the main menu.

4.14.10 Incremental jogging

Description

Incremental jogging makes it possible to move the robot a defined distance, e.g. 10 mm or 3°. The robot then stops by itself.

Incremental jogging can be activated for jogging with the jog keys. Incremental jogging is not possible in the case of jogging with the Space Mouse.

Areas of application:

- Positioning of equidistant points
- Moving a defined distance away from a position, e.g. in the event of a fault
- Mastering with the dial gauge

The following options are available:

Setting	Description
Continuous	Incremental jogging is deactivated.
100 mm / 10°	1 increment = 100 mm or 10°
10mm / 3°	1 increment = 10 mm or 3°
1mm / 1°	1 increment = 1 mm or 1°
0.1mm / 0.005°	1 increment = 0.1 mm or 0.005°

Increments in mm:

- Valid for Cartesian jogging in the X, Y or Z direction.

Increments in degrees:

- Valid for Cartesian jogging in the A, B or C direction.
- Valid for axis-specific jogging.

Precondition

- The jog mode “Jog keys” is active.
- Operating mode T1

Procedure

1. Select the size of the increment in the status bar.
2. Jog the robot using the jog keys. Jogging can be Cartesian or axis-specific.

Once the set increment has been reached, the robot stops.

If the robot motion is interrupted, e.g. by releasing the enabling switch, the interrupted increment is not resumed with the next motion; a new increment is started instead.

4.15 Jogging external axes

Description

External axes cannot be moved using the Space Mouse. If “Space Mouse” mode is selected, only the robot can be jogged with the Space Mouse. The external axes, on the other hand, must be jogged using the jog keys.

Precondition

- The jog mode “Jog keys” is active.
- Operating mode T1

Procedure

1. Select the desired kinematics group, e.g. **External axes**, on the **Keys** tab in the **Jog options** window.

   The type and number of kinematics groups available depend on the system configuration.
2. Set jog override.
3. Hold down the enabling switch.

   The axes of the selected kinematics group are displayed next to the jog keys.
4. Press the Plus or Minus jog key to move an axis in the positive or negative direction.

Kinematic groups

Depending on the system configuration, the following kinematics groups may be available.

Kinematics group	Description
Robot axes	The robot axes can be moved using the jog keys. The external axes cannot be jogged.
External axes	All configured external axes (e.g. external axes E1 to E5) can be moved using the jog keys.
NAME / **External Kinematics Group** *n*	The axes of an external kinematics group can be moved using the jog keys. The name is taken from the system variable $ET*n*_*NAME* (*n* = number of the external kinematic system). If $ET*n*_*NAME* is empty, the default name **External Kinematics Group** *n* is displayed.
[User-defined kinematics group]	The axes of a user-defined kinematics group can be moved using the jog keys. The name corresponds to the name of the user-defined kinematics group.

4.16 Bypassing workspace monitoring

Description

Workspaces can be configured for a robot. These serve to protect the system. A workspace is always of one of the two following types:

- Space that the robot must not leave

  A monitoring function is triggered if the robot leaves the space.
- Space that the robot must not enter

  A monitoring function is triggered if the robot enters the space.

Exactly what reactions occur when the monitoring function is triggered depends on the configuration.

One possible reaction, for example, is that the robot stops. In this case, the workspace monitoring function must be bypassed or deactivated in order to be able to move the robot back out of the violated space.

Precondition

- User group “Expert”
- Operating mode T1

Procedure

1. In the main menu, select **Configuration** > **Miscellaneous** > **Workspace monitoring** > **Override**.
2. Move the robot manually out of the violated space.

   Once the robot has left the violated space, the workspace monitoring is automatically active again.

4.17 Display functions

4.17.1 Measuring and displaying energy consumption

Description

The overall energy consumption of the robot and robot controller can be displayed on the smartHMI. A prerequisite for this is that measurement of energy consumption is possible for the robot type used.

The energy consumption of optional robot controller components (e.g. US1, US2, etc.) and of other controllers is not taken into consideration. It is always the consumption for the last 60 minutes since the most recent cold start that is displayed. Furthermore, the user has the option of starting and stopping measurements manually.

Traces can be made for the consumption values. The predefined configuration Tracedef_KRC_EnergyCalc is available for this.

The data can also be transferred to a higher-level controller by means of PROFIenergy. PROFIenergy is a component of KR C4 PROFINET.

There are two ways of starting and stopping measurements:

- In the **Energy consumption** window (>>> Fig. 4-24)
- Via KRL

Precondition

- Measurement of energy consumption is possible for the robot type used.

  If not, the boxes in the **Energy consumption** window are grayed out.

Procedure

Starting and stopping a measurement in the **Energy consumption** window:

1. In the main menu, select **Display** > **Energy consumption**. The **Energy consumption** window opens.
2. If required, activate the check box next to **Refresh**.
3. Press **Start measuring**. A red dot to the right of the top line now indicates that a measurement is in progress.
4. To stop the measurement, press **Stop measuring**. The result is displayed.

Starting and stopping a measurement via KRL:

1. Start the measurement via $ENERGY_MEASURING.ACTIVE = TRUE (possible via the KRL program or variable correction function). The measurement starts.
2. In the main menu, select **Display** > **Energy consumption**. The **Energy consumption** window opens. A red dot to the right of the top line indicates the measurement that is in progress.
3. If required, activate the check box next to **Refresh**.

4. Stop the measurement by means of $ENERGY_MEASURING.ACTIVE = FALSE.

The **Energy consumption** window can also be opened independently of the measurement. The top line always indicates the result of the active or most recent measurement.

Measurement properties

- A measurement that has been started runs until it is stopped. This is not dependent on whether the **Energy consumption** window is open or closed.
- A measurement that has been started via KRL can be stopped via KRL or via the **Stop measuring** button.
- A measurement that has been started by means of **Start measuring** can only be stopped by means of **Stop measuring** as long as the **Energy consumption** window remains open. If an attempt is made to stop the measurement via KRL, the robot controller displays the following message: *Energy measurement cannot currently be stopped*.

  Once the **Energy consumption** window has been closed again, the measurement can also be stopped via KRL. This prevents a measurement started in the **Energy consumption** window from permanently blocking measurements via KRL.
- It is not possible to start a measurement while a measurement is already active. In this case, the robot controller displays the following message: *An energy measurement is already active.*. The active measurement must be stopped first.

"Energy consumption" window

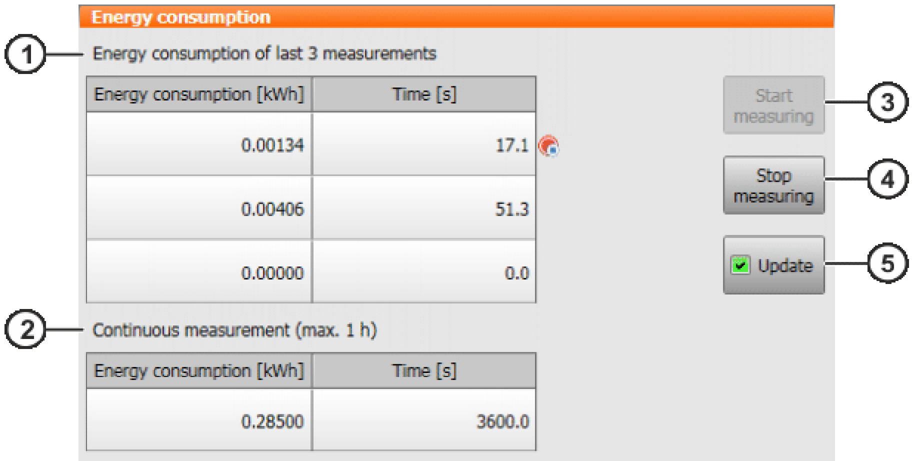


Fig. 4-24: "Energy consumption" window

Item	Description
1	Results of the measurements started by the user The last 3 results are displayed. The most recent result is displayed in the top line. If a measurement is currently active, this is indicated by means of a red dot to the right of the line.
2	Energy consumption for the last 60 minutes since the most recent cold start
3	Starts a measurement. **Start measuring** is not available if a measurement is currently active.

Item	Description
4	Stops an active measurement. How the measurement was started (by means of **Start measuring** or via KRL) is irrelevant.
5	■ Check box active: While a measurement is being carried out, the result display is continually refreshed. ■ Check box not active: While a measurement is being carried out, the most recently refreshed value is displayed. The result is not displayed until the measurement is stopped.

4.17.2 Displaying the actual position

Procedure

1. In the main menu, select **Display** > **Actual position**. The Cartesian actual position is displayed.
2. To display the axis-specific actual position, press **Axis-specific**.
3. To display the Cartesian actual position again, press **Cartesian**.

Description

Actual position, Cartesian:

The current position (X, Y, Z) and orientation (A, B, C) of the TCP are displayed. Status and Turn are also displayed.

Actual position, axis-specific:

The current position of axes A1 to A6 is indicated. If external axes are being used, the position of the external axes is also displayed.

The actual position can also be displayed while the robot is moving.

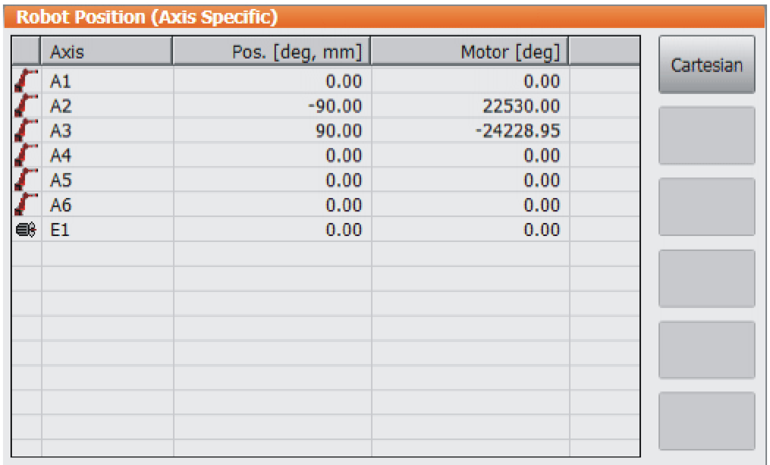


Fig. 4-25: Actual position, axis-specific

4.17.3 Displaying digital inputs/outputs

Procedure

1. In the main menu, select **Display** > **Inputs/outputs** > **Digital I/O**.
2. To display a specific input/output:
   - Click on the **Go to** button. The **Go to:** box is displayed.
   - Enter the number and confirm with the Enter key.

   The display jumps to the input/output with this number.

Description

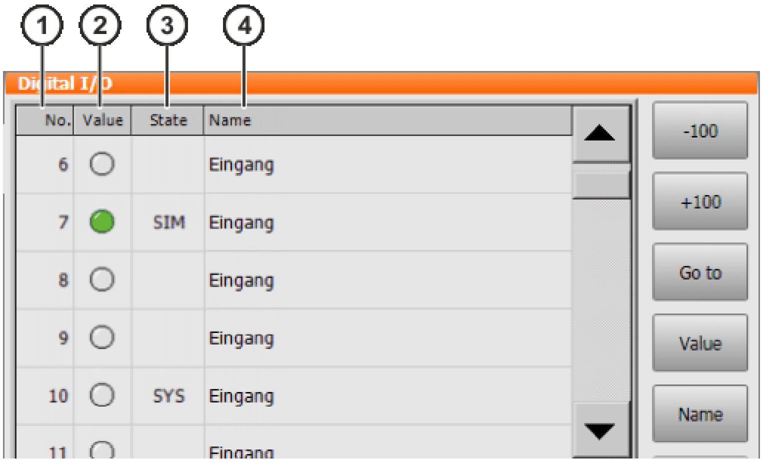


Fig. 4-26: Digital inputs

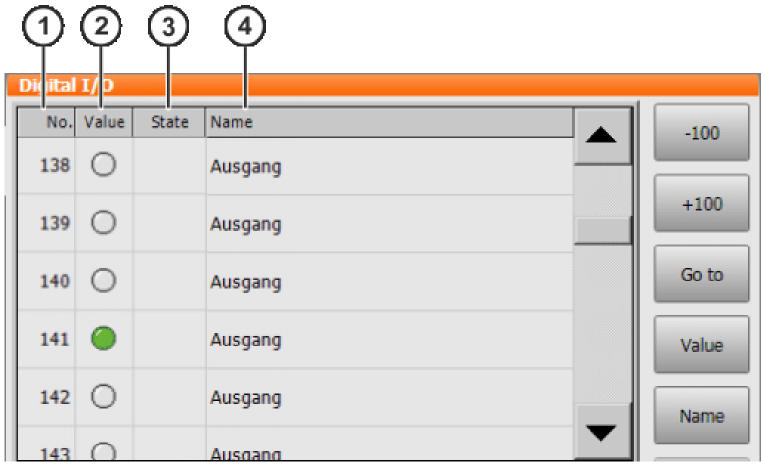


Fig. 4-27: Digital outputs

Item	Description
1	Number of the input/output
2	Value of the input/output. The icon is green if the input or output is TRUE.
3	**SIM**: The input/output is simulated. **SYS**: The value of the input/output is saved in a system variable. This input/output is write-protected.
4	Name of the input/output

The following buttons are available:

Button	Description
-100	Toggles back 100 inputs or outputs in the display.
+100	Toggles forward 100 inputs or outputs in the display.
Go to	The number of the input or output being searched for can be entered.

Button	Description
Value	Toggles the selected input/output between TRUE and FALSE. Precondition: The enabling switch is pressed. ■ **Value** is not available in EXT mode. ■ **Value** is only available for inputs if simulation is activated.
Name	The name of the selected input or output can be changed.

4.17.4 Displaying analog inputs/outputs

Procedure

1. In the main menu, select **Display** > **Inputs/outputs** > **Analog I/O**.
2. To display a specific input/output:
   - Click on the **Go to** button. The **Go to:** box is displayed.
   - Enter the number and confirm with the Enter key.

   The display jumps to the input/output with this number.

The following buttons are available:

Button	Description
Go to	The number of the input or output being searched for can be entered.
Voltage	A voltage can be entered for the selected output. ■ **-10 … 10 V** This button is only available for outputs.
Name	The name of the selected input or output can be changed.

4.17.5 Displaying inputs/outputs for Automatic External

Procedure

- In the main menu, select **Display** > **Inputs/outputs** > **Automatic External**.

Description

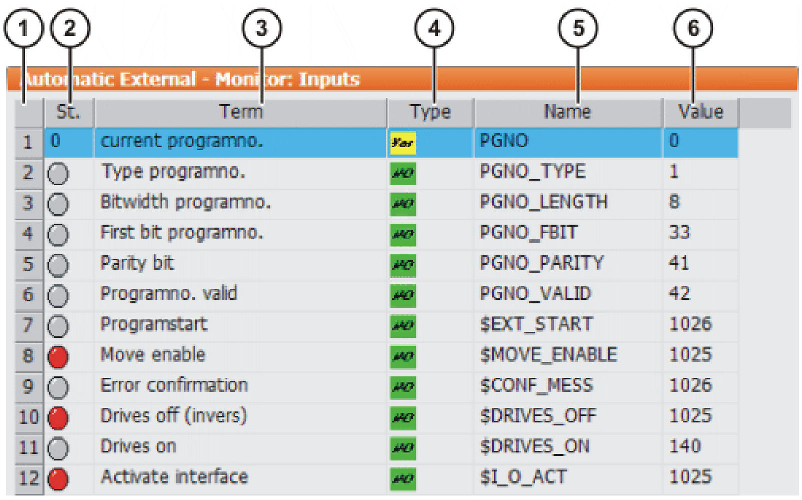


Fig. 4-28: Automatic External inputs (detail view)

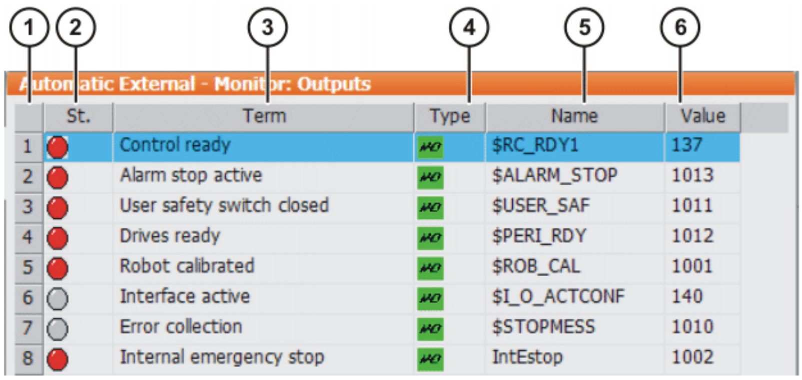


Fig. 4-29: Automatic External outputs (detail view)

Item	Description
1	Number
2	Status ■ **Gray**: inactive (FALSE) ■ **Red**: active (TRUE)
3	Long text name of the input/output
4	Type ■ **Green**: input/output ■ **Yellow**: variable or system variable ($...)
5	Name of the signal or variable
6	Input/output number or channel number

Columns 4, 5 and 6 are only displayed if **Details** has been pressed.

The following buttons are available:

Button	Description
Config.	Switches to the configuration of the Automatic External interface.
Inputs/Outputs	Toggles between the windows for inputs and outputs.
Details/Normal	Toggles between the **Details** and **Normal** views.

4.17.6 Displaying cyclical flags

Procedure

1. In the main menu, select **Display** > **Variable** > **Cyclical flags**. The **Cyclical flags** window is opened.
2. To display a specific flag:
   - Click on the **Go to** button. The **Go to:** box is displayed.
   - Enter the number and confirm with the Enter key.

   The display jumps to the flag with this number.

Description

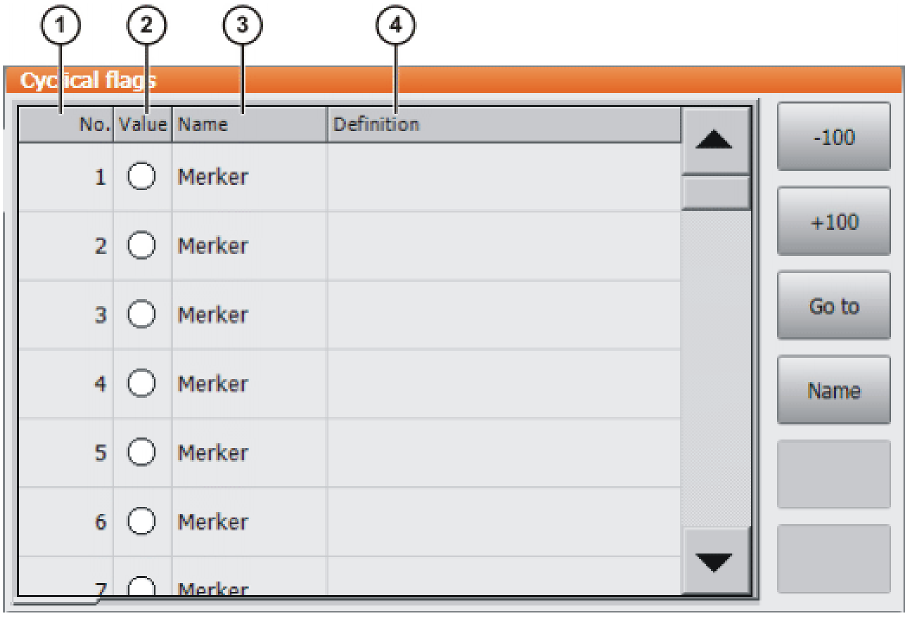


Fig. 4-30: Cyclical flags

Item	Description
1	Flag number
2	Value of the flag. The icon is green if a flag is set.
3	Name of the flag
4	The conditions linked to the setting of a cyclical flag are indicated here.

The following buttons are available:

Button	Description
-100	Toggles back 100 flags in the display.
+100	Toggles forward 100 flags in the display.
Go to	The number of the flag being searched for can be entered.
Name	The name of the selected flag can be modified.

4.17.7 Displaying flags

Procedure

1. In the main menu, select **Display** > **Variable** > **Flags**. The **Flags** window is opened.
2. To display a specific flag:
   - Click on the **Go to** button. The **Go to:** box is displayed.
   - Enter the number and confirm with the Enter key.

   The display jumps to the flag with this number.

Description

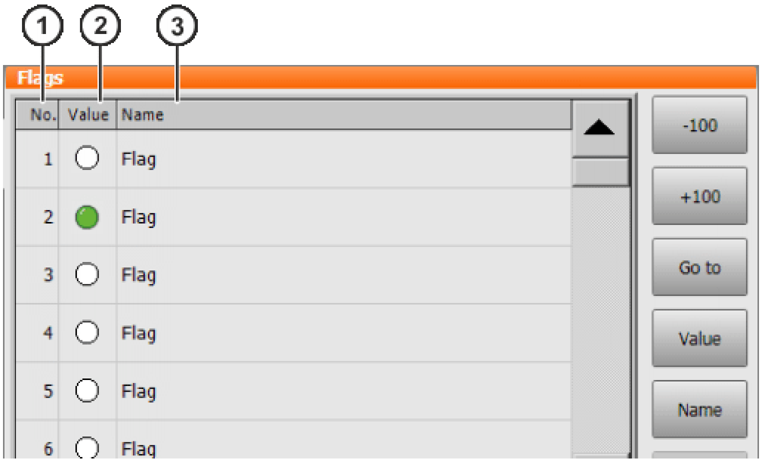


Fig. 4-31: Flags

Item	Description
1	Flag number
2	Value of the flag. The icon is green if a flag is set.
3	Name of the flag

The following buttons are available:

Button	Description
-100	Toggles back 100 flags in the display.
+100	Toggles forward 100 flags in the display.
Go to	The number of the flag being searched for can be entered.
Value	Toggles the selected flag between TRUE and FALSE. Precondition: The enabling switch is pressed. This button is not available in AUT EXT mode.
Name	The name of the selected flag can be modified.

4.17.8 Displaying counters

Procedure

1. In the main menu, select **Display** > **Variable** > **Counter**. The **Counter** window is opened.
2. To display a specific counter:
   - Click on the **Go to** button. The **Go to:** box is displayed.
   - Enter the number and confirm with the Enter key.

   The display jumps to the counter with this number.

Description

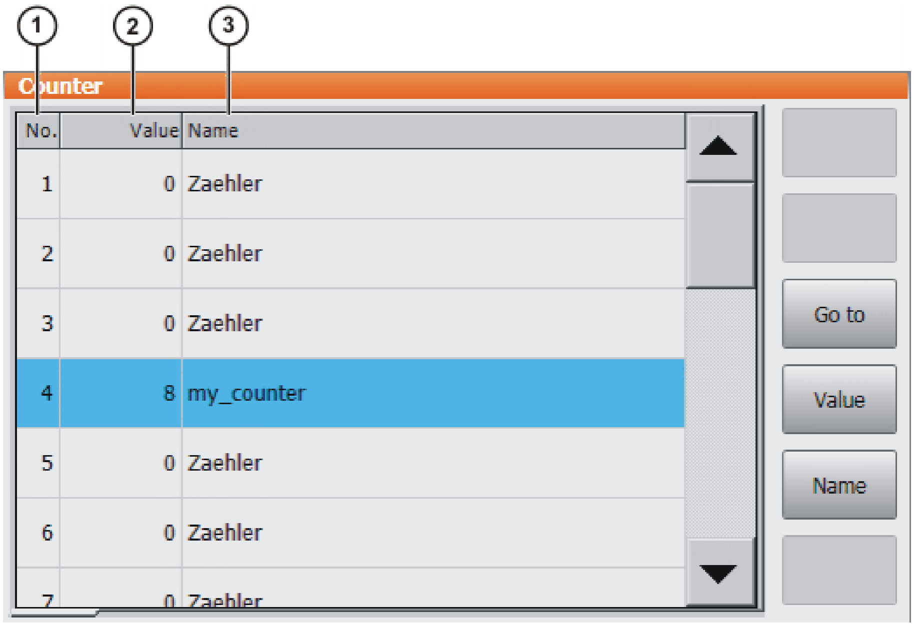


Fig. 4-32: Counter

Item	Description
1	Counter number
4	Value of the counter.
5	Name of counter

The following buttons are available:

Button	Description
Go to	The number of the counter being searched for can be entered.
Value	A value can be entered for the selected counter.
Name	The name of the selected counter can be modified.

4.17.9 Displaying timers

Procedure

1. In the main menu, select **Display** > **Variable** > **Timer**. The **Timer** window is opened.
2. To display a specific timer:
   - Click on the **Go to** button. The **Go to:** box is displayed.
   - Enter the number and confirm with the Enter key.

   The display jumps to the timer with this number.

Description

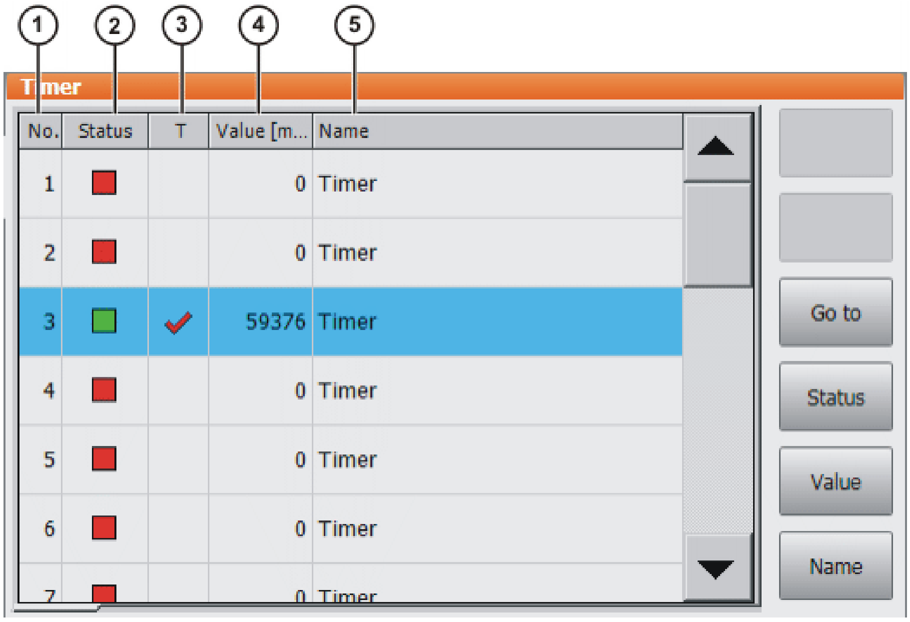


Fig. 4-33: Timer

Item	Description
1	Number of the timer
2	Status of the timer ■ If the timer is activated, this is indicated in green. ■ If the timer is deactivated, this is indicated in red.
3	State of the timer ■ If the value of the timer is > 0, the timer flag is set (red check mark). ■ If the value of the timer is ≤ 0, no timer flag is set.
4	Value of the timer (unit: ms)
5	Name of timer

The following buttons are available:

Button	Description
Go to	The number of the timer being searched for can be entered.
State	Toggles the selected timer between TRUE and FALSE. Precondition: The enabling switch is pressed.
Value	A value can be entered for the selected timer.
Name	The name of the selected timer can be modified.

4.17.10 Displaying calibration data

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Calibration points** and the desired menu item:
   - **Tool type**
   - **Base type**
   - **External axis**
2. Enter the number of the tool, base or external kinematic system.

   The calibration method and the calibration data are displayed.

4.17.11 Displaying information about the robot and robot controller

Procedure

- In the main menu, select **Help** > **Info**.

Description

The information is required, for example, when requesting help from KUKA Customer Support.

The tabs contain the following information:

Tab	Description
Info	■ Robot controller type ■ Robot controller version ■ User interface version ■ Kernel system version
Robot	■ Robot name ■ Robot type and configuration ■ Service life The operating hours meter is running as long as the drives are switched on. Alternatively, the operating hours can also be displayed via the variable $ROB-RUNTIME. ■ Number of axes ■ List of external axes ■ Machine data version
System	■ Control PC name ■ Operating system version ■ Storage capacities
Options	Additionally installed options and technology packages
Comments	Additional comments
Modules	Names and versions of important system files The **Export** button exports the contents of the **Modules** tab to the file C:\KRC\ROBOTER\LOG\FILEVERSIONS.TXT.

4.17.12 Displaying/editing robot data

Precondition

- T1 or T2 operating mode
- No program is selected.

Procedure

- In the main menu, select **Start-up** > **Robot data**.

Description

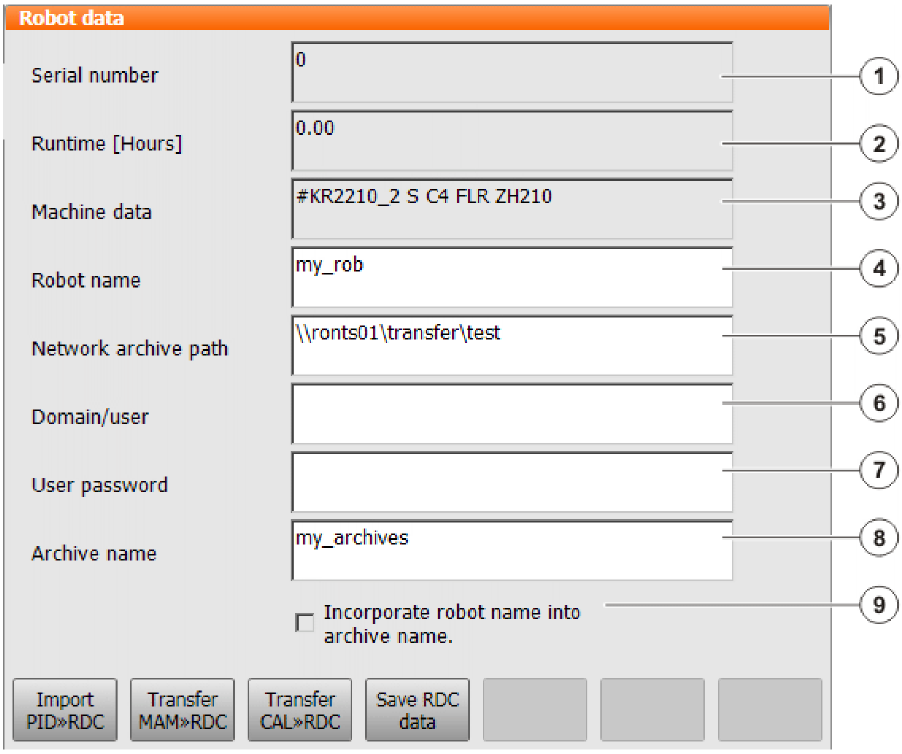


Fig. 4-34: Robot data window

Item	Description
1	Serial number
2	Operating hours. The operating hours meter is running as long as the drives are switched on. Alternatively, the operating hours can also be displayed via the variable $ROBRUNTIME.
3	Machine data name
4	Robot name. The robot name can be changed.
5	Robot controller data can be archived. The target directory can be defined here. It can be a network directory or a local directory. If a directory is defined here, it is also available for importing/exporting long texts.
6 7	If archiving to the network requires a user name and password, these can be entered here. It is then no longer necessary to enter them every time for archiving.
8	This box is only displayed if the check box **Incorporate robot name into archive name.** is not activated. A name for the archive file can be defined here.
9	■ **Check box active**: The robot name is used as the name for the archive file. If no robot name is defined, the name *archive* is used. ■ **Check box not active**: A separate name can be defined for the archive file.

The buttons are not available in the user group “User”.

4.18 Exporting the safety configuration (XML export)

Description

Parts of the safety configuration can be exported. The export creates an XML file. This contains only those parameters which are relevant for the safety options, e.g. SafeOperation.

- Exporting is always possible, irrespective of whether a safety option is installed or not. However, an export only makes sense if a safety option is installed.
- If no safety option is installed on the robot controller, the parameters in the XML file are filled with default values (often “0”).

> In addition to exporting, it is also possible to import a safety configuration when a safety option is installed. More detailed information about exporting and importing can be found in the safety option documentation.
> It is also possible to import or export safety configurations in WorkVisual. Information about this can be found in the WorkVisual documentation.

Procedure

1. In the main menu, select **Configuration** > **Safety configuration**.

   The **Safety configuration** window opens.
2. Press **Export**. The available drives are displayed.
3. Select the desired file path and press **Export**.

   The safety configuration is saved in an XML file. The file name is generated automatically.

5 Start-up and recommissioning

5.1 Start-up wizard

Description

Start-up can be carried out using the Start-up wizard. This guides the user through the basic start-up steps.

Precondition

- No program is selected.
- Operating mode T1

Procedure

- Select **Start-up** > **Start-up wizard** in the main menu.

5.2 Jogging the robot without a higher-level safety controller

Description

To jog the robot without a higher-level safety controller, Start-up mode must first be activated. The robot can then be jogged in T1 mode.

> **DANGER** External safeguards are disabled in Start-up mode. Observe the safety instructions relating to Start-up mode. (>>> 3.8.3.2 "Start-up mode" Page 33)

The robot controller automatically deactivates Start-up mode in the following cases:

- If no operator action has been carried out within 30 min of activation.
- If the smartPAD is switched to passive mode or disconnected from the robot controller.
- If the Ethernet safety interface is used: when a connection to a higher-level safety controller is established.
- If a discrete safety interface is used:

  System Software 8.2 or earlier: The robot controller automatically deactivates Start-up mode if it is no longer the case that all input signals at the discrete interface (and, if used, at the discrete safety interface for safety options) have the state “logic zero”.

  From System Software 8.3 onwards, on the other hand, Start-up mode is not dependent on the inputs at the discrete safety interfaces.

Effect

When the Start-up mode is activated, all outputs are automatically set to the state “logic zero”.

If the robot controller has a peripheral contactor (US2), and if the safety configuration specifies for this to switch in accordance with the motion enable, then the same also applies in Start-up mode. This means that if motion enable is present, the US2 voltage is switched on – even in Start-up mode.

> **NOTICE** The maximum number of switching cycles of the peripheral contactors is 175 per day.

In Start-up mode, the system switches to the following simulated input image:

- The external EMERGENCY STOP is not active.
- The safety gate is open.
- No safety stop 1 has been requested.
- No safety stop 2 has been requested.
- No safe operational stop has been requested.
- Only for VKR C4: E2/E22 is closed.

If SafeOperation or SafeRangeMonitoring is used, Start-up mode also influences other signals.

> Information about the effects of Start-up mode in conjunction with SafeOperation or SafeRangeMonitoring can be found in the documentation **SafeOperation** and **SafeRangeMonitoring**.

Precondition

- T1 mode
- In the case of VKR C4: no E2/E22/E7 signals are activated via a USB stick or retrofit interface.
- In the case of RoboTeam: the local smartPAD is used.
- If the Ethernet safety interface is used: No connection to a higher-level safety controller
- If a discrete safety interface is used:

  For System Software 8.2 only: all input signals have the state “logic zero”. If the additional discrete safety interface for safety options is used, the inputs there must also have the state “logic zero”.

  (From System Software 8.3 onwards, Start-up mode is not dependent on the state of these inputs.)

Procedure

- In the main menu, select **Start-up** > **Service** > **Start-up mode**.

Menu	Description
	Start-up mode is active. Touching the menu item deactivates the mode.
	Start-up mode is not active. Touching the menu item activates the mode.

5.3 Checking the activation of the positionally accurate robot model

Description

If a positionally accurate robot is used, it must be checked that the positionally accurate robot model is activated.

In the case of positionally accurate robots, position deviations resulting from workpiece tolerances and elastic effects of the individual robots are compensated for. The positionally accurate robot positions the programmed TCP anywhere in the Cartesian workspace within the tolerance limits. The model parameters of the positionally accurate robot are determined at a calibration station and permanently saved on the robot (RDC).

> The positionally accurate robot model is only valid for the robot as delivered.
> Following conversion or retrofitting of the robot, e.g. with an arm extension or a new wrist, the robot must be recalibrated.

Functions

A positionally accurate robot has the following functions:

- Increased positioning accuracy, approximately by the factor 10
- Increased path accuracy

> A precondition for the increased positioning and path accuracy is the correct input of the load data into the robot controller.

- Simplified transfer of programs if the robot is exchanged (no reteaching)
- Simplified transfer of programs after offline programming with WorkVisual (no reteaching)

Procedure

1. In the main menu, select **Help** > **Info**.

2. Check on the **Robot** tab that the positionally accurate robot model is activated. (= specification **Positionally accurate robot**).

5.4 Mastering

Overview

Every robot must be mastered. Only if the robot has been mastered can it move to programmed positions and be moved using Cartesian coordinates. During mastering, the mechanical position and the electronic position of the robot are aligned. For this purpose, the robot is moved to a defined mechanical position, the mastering position. The encoder value for each axis is then saved.

The mastering position is similar, but not identical, for all robots. The exact positions may even vary between individual robots of a single robot type.

Fig. 5-1: Mastering position – approximate position

A robot must be mastered in the following cases:

Case	Comment
During commissioning	- - -
After maintenance work during which the robot loses its mastering, e.g. exchange of motor or RDC	(>>> 5.4.8 "Reference mastering" Page 101)
When the robot has been moved without the robot controller (e.g. with the release device)	- - -

Case	Comment
After exchanging a gear unit	Before carrying out a new mastering procedure, the old mastering data must first be deleted! Mastering data are deleted by manually unmastering the axes. (>>> 5.4.10 "Manually unmastering axes" Page 109)
After an impact with an end stop at more than 250 mm/s	
After a collision.	

5.4.1 Mastering methods

Overview The mastering methods that can be used for a robot depend on the type of gauge cartridge with which it is equipped. The types differ in terms of the size of their protective caps.

Type of gauge cartridge	Mastering methods
Gauge cartridge for **SEMD** (Standard Electronic Mastering Device) Protective cap with fine thread, M20	Mastering with the probe, type **SEMD** (>>> 5.4.5 "Mastering with the SEMD" Page 95)
	Mastering with the dial gauge (>>> 5.4.6 "Mastering with the dial gauge" Page 100)
	Reference mastering Only for mastering after certain maintenance work (>>> 5.4.8 "Reference mastering" Page 101)
Gauge cartridge for **MEMD** (Micro Electronic Mastering Device) Protective cap with fine thread, M8	Mastering with the probe, type **MEMD** On A6 in certain cases: mastering to the mark (>>> 5.4.9 "Mastering with the MEMD and mark" Page 102)

SEMD/MEMD SEMD and/or MEMD are contained in the KUKA mastering kit. There are several variants of the mastering kit.

Fig. 5-2: Mastering kit with SEMD and MEMD

1	Mastering box	4	SEMD
2	Screwdriver for MEMD	5	Cables
3	MEMD		

The thinner cable is the signal cable. It connects the SEMD or MEMD to the mastering box.

The thicker cable is the EtherCAT cable. It is connected to the mastering box and to the robot at X32.

NOTICE

- Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.
- In the case of probes to which the signal cable is not permanently attached, always screw the device onto the gauge cartridge without the signal cable. Only then may the cable be attached to the device. Otherwise, the cable could be damaged. Similarly, when removing the device, the signal cable must always be removed from the device first. Only then may the device be removed from the gauge cartridge.
- After mastering, remove the EtherCAT cable from connection X32. Failure to do so may result in interference signals or damage.

5.4.2 Moving axes to the pre-mastering position using mastering marks

Description

The axes must be moved to the pre-mastering position before every mastering operation. To do so, each axis is moved so that the mastering marks line up.

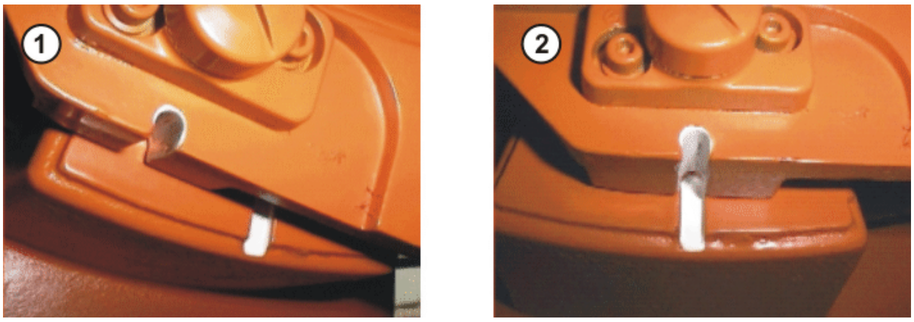


Fig. 5-3: Moving an axis to the pre-mastering position

In some cases it is not possible to align the axes using the mastering marks, e.g. because the marks can no longer be recognized due to fouling. The axes can also be mastered using the probe instead of the mastering marks.
(>>> 5.4.3 "Moving axes to the pre-mastering position using the probe" Page 93)

The following figure shows where on the robot the mastering marks are situated. Depending on the specific robot model, the positions may deviate slightly from those illustrated.

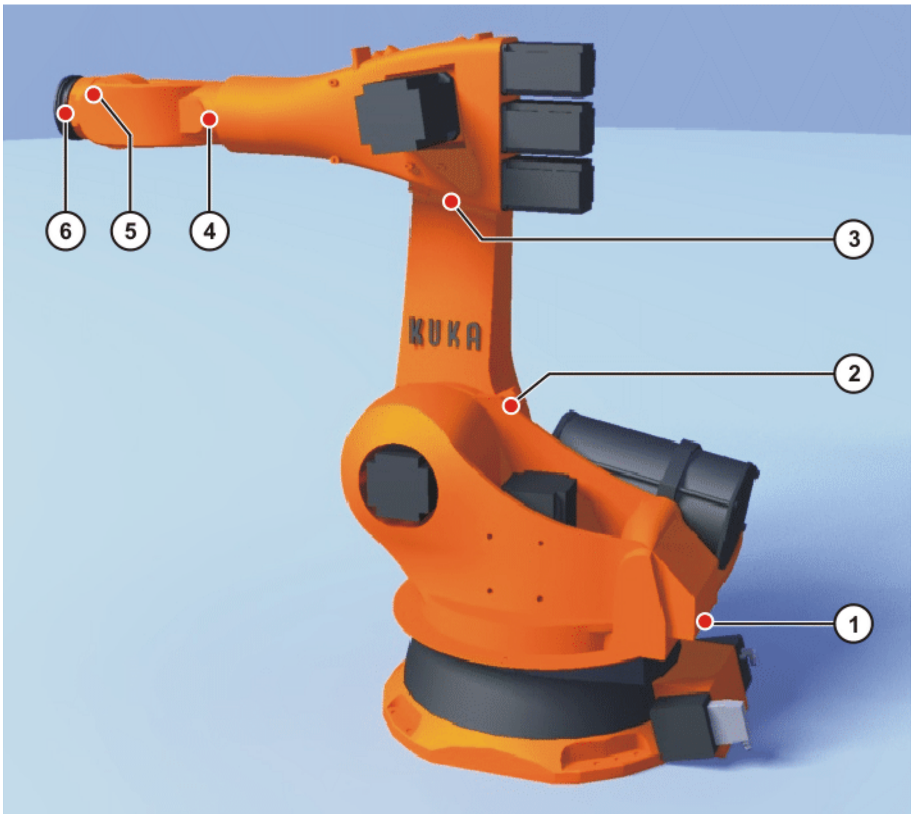


Fig. 5-4: Mastering marks on the robot

Precondition

- The jog mode “Jog keys” is active.
- Operating mode T1

NOTICE Before A4 and A6 are moved to the pre-mastering position, ensure that the energy supply system – if present – is in its correct position and not rotated through 360°.

Procedure

1. Select **Axes** as the coordinate system for the jog keys.

2. Press and hold down the enabling switch.

   Axes A1 to A6 are displayed next to the jog keys.
3. Press the plus or minus jog key to move an axis in the positive or negative direction.
4. Move each axis, starting from A1 and working upwards, so that the mastering marks line up. (An exception is made for A6 of robots for which this axis is mastered using the mark.)

5.4.3 Moving axes to the pre-mastering position using the probe

Description

The axes must be moved to the pre-mastering position before every mastering operation. This is generally done using the mastering marks.

It is sometimes not possible, however, e.g. because the marks can no longer be recognized due to fouling. The axes can also be mastered using the probe instead of the mastering marks. An LED on the smartHMI indicates when the pre-mastering position has been reached.

Precondition

- The jog mode “Jog keys” is active.
- Operating mode T1
- No program is selected.
- The user knows the approximate pre-mastering position of the axes.

> **NOTICE** Before A4 and A6 are moved to the pre-mastering position, ensure that the energy supply system – if present – is in its correct position and not rotated through 360°.

Procedure

1. Jog the robot to a position in which the axes are close to their pre-mastering position. It should subsequently be possible to move them in the minus direction to the pre-mastering position.
2. In the main menu, select **Start-up** > **Master** > **EMD** > **With load correction**.

   Depending on the method for which the axes are to be aligned, the option **First mastering** or **Teach offset** or **With offset** is now selected.
3. Proceed in accordance with the instructions for the relevant mastering procedure until the probe is attached to A1 and connected via the mastering box to X32.

> Thereafter, do NOT continue to follow the description of the mastering procedure!
> i.e. do NOT press **Master** or **Learn** or **Check**!

4. The LED **EMD in mastering range** is displayed on the smartHMI. It must now be red. Observe this LED closely.

   (>>> 5.4.4 "Mastering LEDs" Page 94)
5. Jog the robot in the minus direction. As soon as the LED switches from red to green, stop the robot.

   A1 is now in the pre-mastering position.

> The axes indicated next to the LEDs do not disappear one after the other in the usual way. This does not occur until the actual mastering.

> Do not yet master the axis. The actual mastering operation must not be carried out until all axes are in the pre-mastering position. If this is not observed, correct mastering cannot be achieved.

6. Remove the probe from the gauge cartridge as described in the mastering procedure and replace the protective cap.

7. Move the remaining axes to the pre-mastering position in the same way in ascending order. (An exception is made for A6 of robots for which this axis is mastered using the mark.)
8. Close the window containing the mastering LEDs.
9. Disconnect the EtherCAT cable from X32 and the mastering box.

NOTICE Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.4 Mastering LEDs

For most mastering operations, the smartHMI displays a list of axes. There are 2 LEDs to the right of the list.

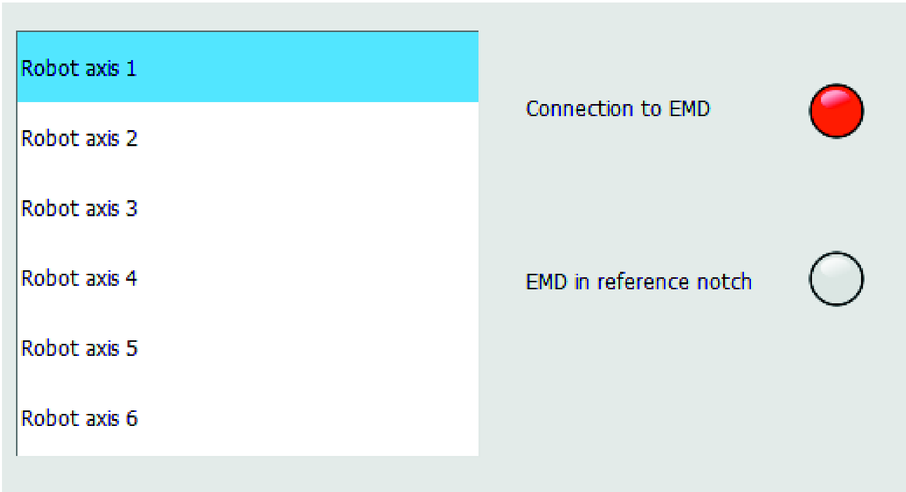


Fig. 5-5: Mastering LEDs

LED	Description
Connection to EMD	■ **Red**: The probe is not connected to connection X32. ■ **Green**: The probe is connected to connection X32. If this LED is red, the LED **EMD in mastering range** is gray.
EMD in mastering range	■ **Gray**: The probe is not connected to connection X32. ■ **Red**: The probe is in a position where mastering is not possible. ■ **Green**: The probe is either immediately next to or in the mastering notch.

The LED **EMD in mastering range** can be used to move the axes to the pre-mastering position with the aid of the probe. The pre-mastering position is reached at the moment when the LED changes from red to green during jogging in the minus direction.

(>>> 5.4.3 "Moving axes to the pre-mastering position using the probe" Page 93)

5.4.5 Mastering with the SEMD

Overview

In SEMD mastering, the axis is automatically moved by the robot controller to the mastering position. Mastering is carried out first without and then with a load. It is possible to save mastering data for different loads.

Step	Description
1	**First mastering** (>>> 5.4.5.1 "First mastering (with SEMD)" Page 95) First mastering is carried out without a load.
2	**Teach offset** (>>> 5.4.5.2 "Teach offset (with SEMD)" Page 98) “Teach offset” is carried out with a load. The difference from the first mastering is saved.
3	If required: **Load mastering with offset** (>>> 5.4.5.3 "Check load mastering with offset (with SEMD)" Page 99) “Load mastering with offset” is carried out with a load for which an offset has already been taught. Area of application: ■ Checking first mastering ■ Restoring first mastering if it has been lost (e.g. following exchange of motor or collision). Since an offset that has been taught is retained, even if mastering is lost, the robot controller can calculate the first mastering.

5.4.5.1 First mastering (with SEMD)

Precondition

- There is no load on the robot; i.e. there is no tool, workpiece or supplementary load mounted.
- All axes are in the pre-mastering position.
- No program is selected.
- Operating mode T1

Procedure

NOTICE The SEMD must always be screwed onto the gauge cartridge without the signal cable attached. Only then may the cable be attached to the SEMD. Otherwise, the cable could be damaged. Similarly, when removing the SEMD, the signal cable must always be removed from the SEMD first. Only then may the SEMD be removed from the gauge cartridge.
After mastering, remove the EtherCAT cable from connection X32. Failure to do so may result in interference signals or damage.

The SEMD actually used need not necessarily look exactly like the model illustrated in the figures. The procedure for using it is the same, however.

1. In the main menu, select **Start-up** > **Master** > **EMD** > **With load correction** > **First mastering**.

   A window opens. All axes to be mastered are displayed. The axis with the lowest number is highlighted.
2. Remove the cover from connection X32.

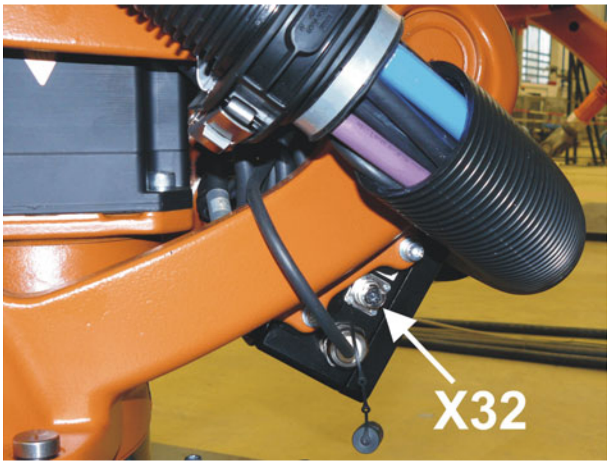


Fig. 5-6: Removing cover from X32

3. Connect the EtherCAT cable to X32 and to the mastering box.

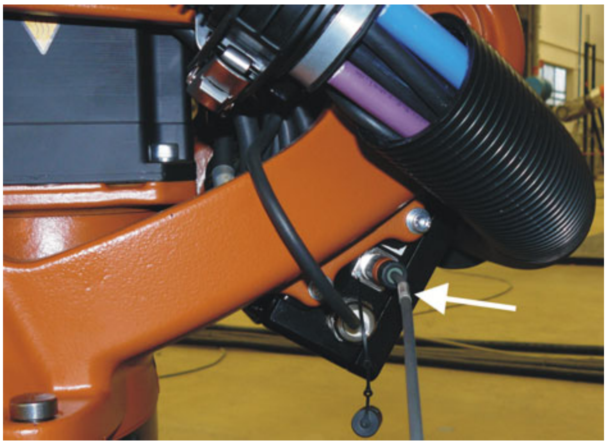


Fig. 5-7: Connecting the EtherCAT cable to X32

4. Remove the protective cap of the gauge cartridge on the axis highlighted in the window. (Turned around, the SEMD can be used as a screwdriver.)

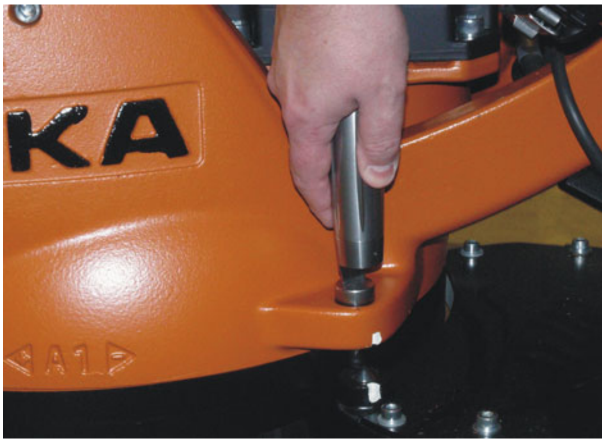


Fig. 5-8: Removing protective cap from gauge cartridge

5. Screw the SEMD onto the gauge cartridge.

Fig. 5-9: Screwing SEMD onto gauge cartridge

6. Attach the signal cable to the SEMD. It is possible to see from the cable socket which way round it has to be on the connector pins at the SEMD.

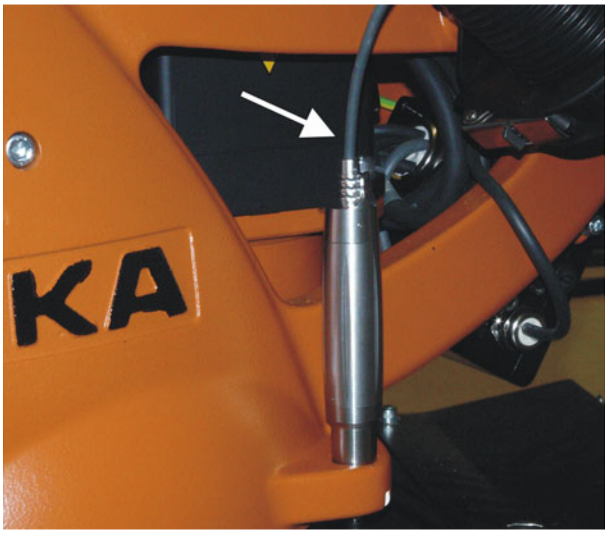


Fig. 5-10: Attaching signal cable to SEMD

7. Connect the signal cable to the mastering box if it is not already connected.
8. Press **Master**.
9. Press an enabling switch and the Start key.

   When the SEMD has passed through the reference notch, the mastering position is calculated. The robot stops automatically. The values are saved. The axis is no longer displayed in the window.
10. Remove the signal cable from the SEMD. Then remove the SEMD from the gauge cartridge and replace the protective cap.
11. Repeat steps 4 to 10 for all axes to be mastered.
12. Close the window.
13. Disconnect the EtherCAT cable from X32 and the mastering box.

NOTICE Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.5.2 Teach offset (with SEMD)

Description

Teach offset is carried out with a load. The difference from the first mastering is saved.

If the robot is operated with different loads, **Teach offset** must be carried out for every load. In the case of grippers used for picking up heavy workpieces, **Teach offset** must be carried out for the gripper both with and without the workpiece.

Precondition

- Same ambient conditions (temperature, etc.) as for first mastering.
- The load is mounted on the robot.
- All axes are in the pre-mastering position.
- No program is selected.
- Operating mode T1

Procedure

NOTICE The SEMD must always be screwed onto the gauge cartridge without the signal cable attached. Only then may the cable be attached to the SEMD. Otherwise, the cable could be damaged. Similarly, when removing the SEMD, the signal cable must always be removed from the SEMD first. Only then may the SEMD be removed from the gauge cartridge.
After mastering, remove the EtherCAT cable from connection X32. Failure to do so may result in interference signals or damage.

1. In the main menu, select **Start-up** > **Master** > **EMD** > **With load correction** > **Teach offset**.
2. Enter tool number. Confirm with **Tool OK**.

   A window opens. All axes for which the tool has not yet been taught are displayed. The axis with the lowest number is highlighted.
3. Remove the cover from connection X32. Connect the EtherCAT cable to X32 and to the mastering box.
4. Remove the protective cap of the gauge cartridge on the axis highlighted in the window. (Turned around, the SEMD can be used as a screwdriver.)
5. Screw the SEMD onto the gauge cartridge.
6. Attach the signal cable to the SEMD. It is possible to see from the cable socket which way round it has to be on the connector pins at the SEMD.
7. Connect the signal cable to the mastering box if it is not already connected.
8. Press **Learn**.
9. Press an enabling switch and the Start key.

   When the SEMD has passed through the reference notch, the mastering position is calculated. The robot stops automatically. A window opens. The deviation of this axis from the first mastering is indicated in degrees and increments.
10. Confirm with **OK**. The axis is no longer displayed in the window.
11. Remove the signal cable from the SEMD. Then remove the SEMD from the gauge cartridge and replace the protective cap.
12. Repeat steps 4 to 11 for all axes to be mastered.
13. Close the window.
14. Disconnect the EtherCAT cable from X32 and the mastering box.

NOTICE Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.5.3 Check load mastering with offset (with SEMD)

Description

Area of application:

- Checking first mastering
- Restoring first mastering if it has been lost (e.g. following exchange of motor or collision). Since an offset that has been taught is retained, even if mastering is lost, the robot controller can calculate the first mastering.

An axis can only be checked if all axes with lower numbers have been mastered.

Precondition

- Same ambient conditions (temperature, etc.) as for first mastering.
- A load for which **Teach offset** has been carried out is mounted on the robot.
- All axes are in the pre-mastering position.
- No program is selected.
- Operating mode T1

Procedure

> **NOTICE** The SEMD must always be screwed onto the gauge cartridge without the signal cable attached. Only then may the cable be attached to the SEMD. Otherwise, the cable could be damaged. Similarly, when removing the SEMD, the signal cable must always be removed from the SEMD first. Only then may the SEMD be removed from the gauge cartridge.
> After mastering, remove the EtherCAT cable from connection X32. Failure to do so may result in interference signals or damage.

1. In the main menu, select **Start-up** > **Master** > **EMD** > **With load correction** > **Load mastering** > **With offset**.
2. Enter tool number. Confirm with **Tool OK**.

   A window opens. All axes for which an offset has been taught with this tool are displayed. The axis with the lowest number is highlighted.
3. Remove the cover from connection X32. Connect the EtherCAT cable to X32 and to the mastering box.
4. Remove the protective cap of the gauge cartridge on the axis highlighted in the window. (Turned around, the SEMD can be used as a screwdriver.)
5. Screw the SEMD onto the gauge cartridge.
6. Attach the signal cable to the SEMD. It is possible to see from the cable socket which way round it has to be on the connector pins at the SEMD.
7. Connect the signal cable to the mastering box if it is not already connected.
8. Press **Check**.
9. Hold down an enabling switch and press the Start key.

   When the SEMD has passed through the reference notch, the mastering position is calculated. The robot stops automatically. The difference from “Teach offset” is displayed.
10. If required, press **Save** to save the values. The old mastering values are deleted.

    To restore a lost first mastering, always save the values.

> Axes A4, A5 and A6 are mechanically coupled. This means:
> If the values for A4 are deleted, the values for A5 and A6 are also deleted.
> If the values for A5 are deleted, the values for A6 are also deleted.

11. Remove the signal cable from the SEMD. Then remove the SEMD from the gauge cartridge and replace the protective cap.

12. Repeat steps 4 to 11 for all axes to be mastered.
13. Close the window.
14. Disconnect the EtherCAT cable from X32 and the mastering box.

> **NOTICE** Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.6 Mastering with the dial gauge

Description

In dial mastering, the axis is moved manually by the user to the mastering position. Mastering is always carried out with a load. It is not possible to save mastering data for different loads.

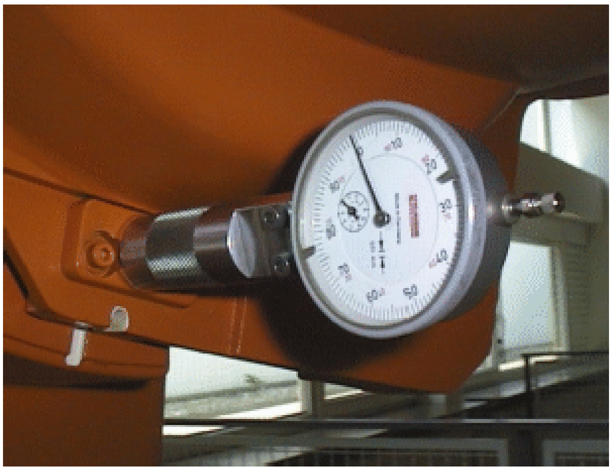


Fig. 5-11: Dial gauge

Precondition

- The load is mounted on the robot.
- All axes are in the pre-mastering position.
- The jog mode “Jog keys” is active and the coordinate system **Axis** has been selected.
- No program is selected.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Master** > **Dial**.

   A window opens. All axes that have not been mastered are displayed. The axis that must be mastered first is selected.
2. Remove the protective cap from the gauge cartridge on this axis and mount the dial gauge on the gauge cartridge.

   Using the Allen key, loosen the screws on the neck of the dial gauge. Turn the dial so that it can be viewed easily. Push the pin of the dial gauge in as far as the stop.

   Using the Allen key, tighten the screws on the neck of the dial gauge.
3. Reduce jog override to 1%.
4. Jog axis from “+” to “-”. At the lowest position of the reference notch, recognizable by the change in direction of the pointer, set the dial gauge to 0.

   If the axis inadvertently overshoots the lowest position, jog the axis backwards and forwards until the lowest position is reached. It is immaterial whether the axis is moved from “+” to “-” or from “-” to “+”.
5. Move the axis back to the pre-mastering position.

6. Move the axis from “+” to “-” until the pointer is about 5-10 scale divisions before zero.
7. Switch to incremental jogging.
8. Move the axis from “+” to “-” until zero is reached.

> If the axis overshoots zero, repeat steps 5 to 8.

9. Press **Master**. The axis that has been mastered is removed from the window.
10. Remove the dial gauge from the gauge cartridge and replace the protective cap.
11. Switch back from incremental jogging to the normal jog mode.
12. Repeat steps 2 to 11 for all axes to be mastered.
13. Close the window.

5.4.7 Mastering external axes

Description

- KUKA external axes can be mastered using either the probe or the dial gauge.
- Non-KUKA external axes can be mastered using the dial gauge. If mastering with the probe is desired, the external axis must be fitted with gauge cartridges.

Procedure

- The procedure for mastering external axes is the same as that for mastering robot axes. Alongside the robot axes, the configured external axes now also appear in the axis selection window.

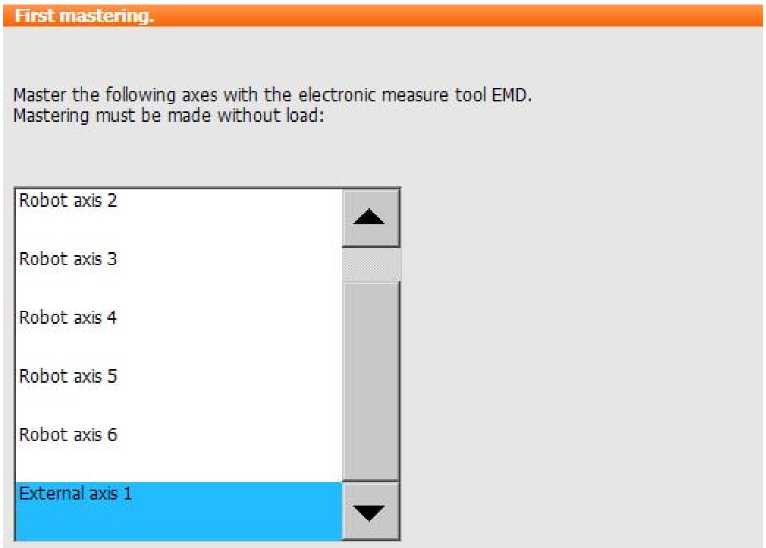


Fig. 5-12: Selection list of axes to be mastered

> Mastering in the case of industrial robots with more than 2 external axes: if the system contains more than 8 axes, it may be necessary to connect the signal cable of the probe to the second RDC.

5.4.8 Reference mastering

> The procedure described here must not be used when commissioning the robot.

Description

Reference mastering is suitable if maintenance work is due on a correctly mastered robot and it is to be expected that the robot will lose its mastering. Examples:

- Exchange of RDC
- Exchange of motor

The robot is moved to the $MAMES position before the maintenance work is commenced. Afterwards, the axis values of this system variable are reassigned to the robot by means of reference mastering. The state of the robot is then the same as before the loss of mastering. Taught offsets are retained. No EMD or dial gauge is required.

In the case of reference mastering, it is irrelevant whether or not there is a load mounted on the robot. Reference mastering can also be used for external axes.

Preparation

- Move the robot to the $MAMES position before commencing the maintenance work. To do so, program a point PTP $MAMES and move the robot to it. This is only possible in the user group “Expert”!

> ⚠ **WARNING** The robot must not move to the default HOME position instead of to $MAMES. $MAMES may be, but is not always, identical to the default HOME position. Only in the $MAMES position will the robot be correctly mastered by means of reference mastering. If the robot is reference mastered at any position other than $MAMES, this may result in injury and material damage.

Precondition

- No program is selected.
- Operating mode T1
- The position of the robot was not changed during the maintenance work.
- If the RDC has been exchanged: the robot data have been transferred from the hard drive to the RDC.

Procedure

1. In the main menu, select **Start-up** > **Master** > **Reference**.

   The option window **Reference mastering** is opened. All axes that have not been mastered are displayed. The axis that must be mastered first is selected.
2. Press **Master**. The selected axis is mastered and removed from the option window.
3. Repeat step 2 for all axes to be mastered.

5.4.9 Mastering with the MEMD and mark

Overview

In MEMD mastering, the axis is automatically moved by the robot controller to the mastering position. Mastering is carried out first without and then with a load. It is possible to save mastering data for different loads.

- In the case of robots with line marks on A6 instead of conventional mastering marks, A6 is mastered without MEMD.

  (>>> 5.4.9.1 "Moving A6 to the mastering position (with line mark)" Page 103)
- In the case of robots with mastering marks on A6, A6 is mastered in the same way as the other axes.

Step	Description
1	**First mastering** (>>> 5.4.9.2 "First mastering (with MEMD)" Page 104) First mastering is carried out without a load.
2	**Teach offset** (>>> 5.4.9.3 "Teach offset (with MEMD)" Page 107) “Teach offset” is carried out with a load. The difference from the first mastering is saved.
3	If required: **Load mastering with offset** (>>> 5.4.9.4 "Check load mastering with offset (with MEMD)" Page 108) “Load mastering with offset” is carried out with a load for which an offset has already been taught. Area of application: ■ Checking first mastering ■ Restoring first mastering if it has been lost (e.g. following exchange of motor or collision). Since an offset that has been taught is retained, even if mastering is lost, the robot controller can calculate the first mastering.

5.4.9.1 Moving A6 to the mastering position (with line mark)

Description

In the case of robots with line marks on A6 instead of conventional mastering marks, A6 is mastered without MEMD.

Before mastering, A6 must be moved to its mastering position. (This means before the overall mastering process, not directly before mastering A6 itself). For this purpose, A6 has fine marks in the metal.

- To move A6 to the mastering position, the marks must be aligned exactly.

> **i** When moving to the mastering position, it is important to look at the fixed mark in a straight line from in front. If the mark is observed from the side, the movable mark cannot be aligned accurately enough. This results in incorrect mastering.

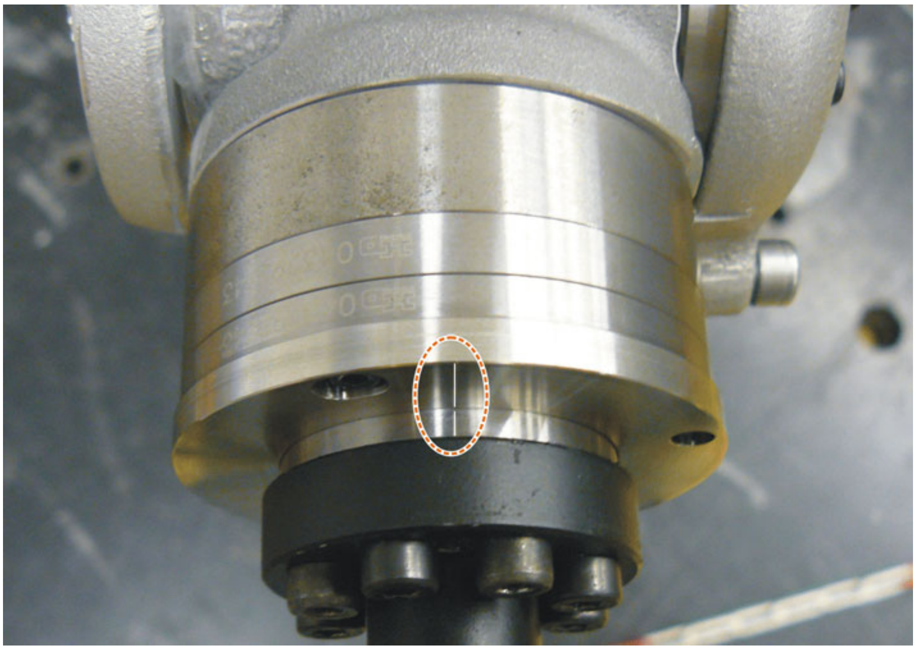


Fig. 5-13: Mastering position A6 – view from above

Mastering fixture

A mastering fixture is available for mastering A6 of the KR AGILUS. Use of this fixture is optional. Using the fixture allows mastering with greater accuracy and greater repeatability.

More information about the mastering fixture is contained in the **Mastering fixture A6** documentation.

5.4.9.2 First mastering (with MEMD)

Precondition

- There is no load on the robot; i.e. there is no tool, workpiece or supplementary load mounted.
- The axes are in the pre-mastering position.

  Exception A6, if this axis has a line mark: A6 is in the mastering position.
- No program is selected.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Master** > **EMD** > **With load correction** > **First mastering**.

   A window opens. All axes to be mastered are displayed. The axis with the lowest number is highlighted.
2. Remove the cover from connection X32.

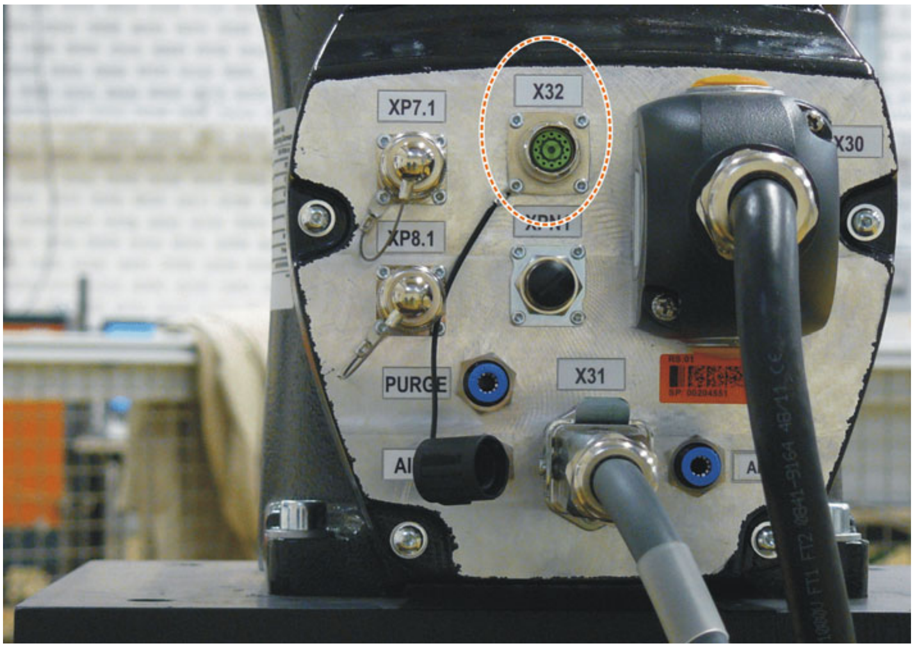


Fig. 5-14: X32 without cover

3. Connect the EtherCAT cable to X32 and to the mastering box.

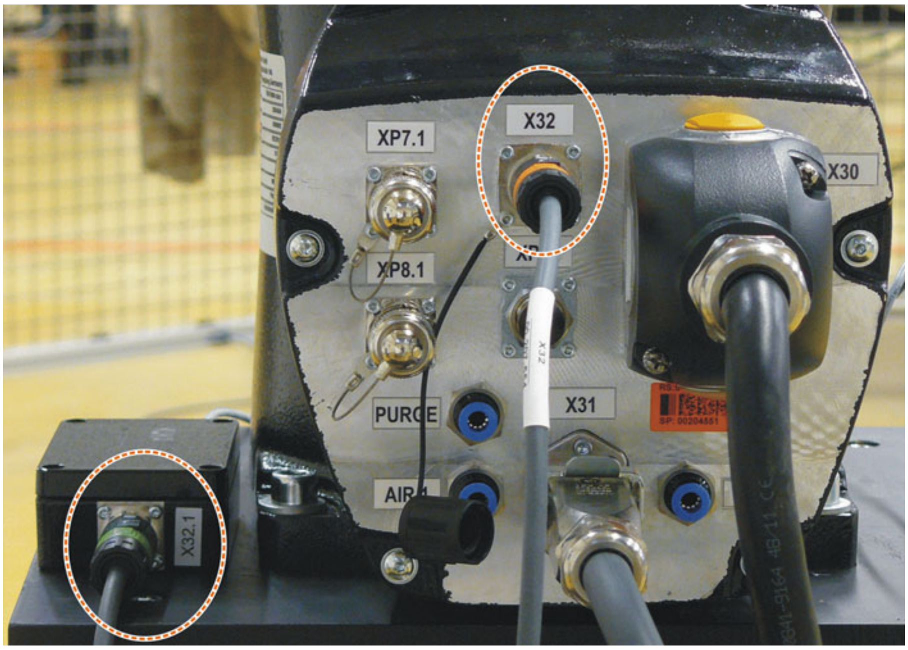


Fig. 5-15: Connecting the cable to X32

4. Remove the protective cap of the gauge cartridge on the axis highlighted in the window.

Fig. 5-16: Removing protective cap from gauge cartridge

5. Screw the MEMD onto the gauge cartridge.

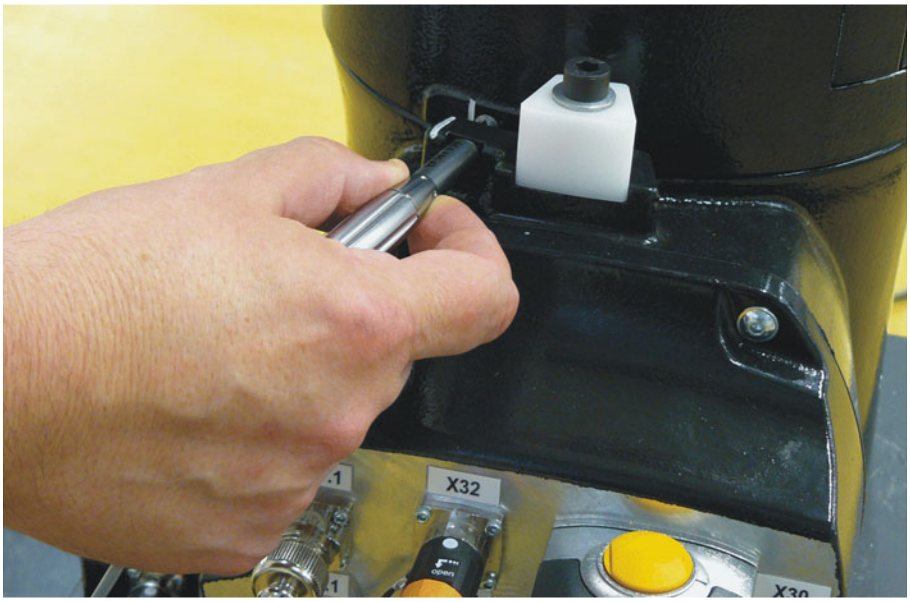


Fig. 5-17: Screwing MEMD onto gauge cartridge

6. Connect the signal cable to the mastering box if it is not already connected.
7. Press **Master**.
8. Press an enabling switch and the Start key.

   When the MEMD has passed through the reference notch, the mastering position is calculated. The robot stops automatically. The values are saved. The axis is no longer displayed in the window.
9. Remove the MEMD from the gauge cartridge and replace the protective cap.
10. Repeat steps 4 to 9 for all axes to be mastered.

    Exception: Not for A6 if this axis has a line mark.
11. Close the window.
12. This step is only to be performed if A6 has a line mark.
    a. In the main menu, select **Start-up** > **Master** > **Reference**.

The option window **Reference mastering** is opened. A6 is displayed and is selected.

b. Press **Master**. A6 is mastered and removed from the option window.

c. Close the window.

13. Disconnect the EtherCAT cable from X32 and the mastering box.

> **NOTICE** Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.9.3 Teach offset (with MEMD)

Description

Teach offset is carried out with a load. The difference from the first mastering is saved.

If the robot is operated with different loads, **Teach offset** must be carried out for every load. In the case of grippers used for picking up heavy workpieces, **Teach offset** must be carried out for the gripper both with and without the workpiece.

Precondition

- Same ambient conditions (temperature, etc.) as for first mastering.
- The load is mounted on the robot.
- The axes are in the pre-mastering position.

  Exception A6, if this axis has a line mark: A6 is in the mastering position.
- No program is selected.
- Operating mode T1

Procedure

1. Select **Start-up** > **Master** > **EMD** > **With load correction** > **Teach offset** in the main menu.
2. Enter tool number. Confirm with **Tool OK**.

   A window opens. All axes for which the tool has not yet been taught are displayed. The axis with the lowest number is highlighted.
3. Remove the cover from connection X32.
4. Connect the EtherCAT cable to X32 and to the mastering box.
5. Remove the protective cap of the gauge cartridge on the axis highlighted in the window.
6. Screw the MEMD onto the gauge cartridge.
7. Connect the signal cable to the mastering box if it is not already connected.
8. Press **Learn**.
9. Press an enabling switch and the Start key.

   When the MEMD has passed through the reference notch, the mastering position is calculated. The robot stops automatically. A window opens. The deviation of this axis from the first mastering is indicated in degrees and increments.
10. Confirm with **OK**. The axis is no longer displayed in the window.
11. Remove the MEMD from the gauge cartridge and replace the protective cap.
12. Repeat steps 5 to 11 for all axes to be mastered.

    Exception: Not for A6 if this axis has a line mark.
13. Close the window.
14. This step is only to be performed if A6 has a line mark.

    a. In the main menu, select **Start-up** > **Master** > **Reference**.

The option window **Reference mastering** is opened. A6 is displayed and is selected.

b. Press **Master**. A6 is mastered and removed from the option window.

c. Close the window.

15. Disconnect the EtherCAT cable from X32 and the mastering box.

> **NOTICE** Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.9.4 Check load mastering with offset (with MEMD)

Description

Area of application:

- Checking first mastering
- Restoring first mastering if it has been lost (e.g. following exchange of motor or collision). Since an offset that has been taught is retained, even if mastering is lost, the robot controller can calculate the first mastering.

An axis can only be checked if all axes with lower numbers have been mastered.

In the case of robots where A6 has a line mark, the value determined for this axis is not displayed, i.e. first mastering cannot be checked for A6. It is possible to restore lost first mastering, however.

Precondition

- Same ambient conditions (temperature, etc.) as for first mastering.
- A load for which **Teach offset** has been carried out is mounted on the robot.
- The axes are in the pre-mastering position.

  Exception A6, if this axis has a line mark: A6 is in the mastering position.
- No program is selected.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Master** > **EMD** > **With load correction** > **Master load** > **With offset**.
2. Enter tool number. Confirm with **Tool OK**.

   A window opens. All axes for which an offset has been taught with this tool are displayed. The axis with the lowest number is highlighted.
3. Remove the cover from connection X32.
4. Connect the EtherCAT cable to X32 and to the mastering box.
5. Remove the protective cap of the gauge cartridge on the axis highlighted in the window.
6. Screw the MEMD onto the gauge cartridge.
7. Connect the signal cable to the mastering box if it is not already connected.
8. Press **Check**.
9. Hold down an enabling switch and press the Start key.

   When the MEMD has passed through the reference notch, the mastering position is calculated. The robot stops automatically. The difference from “Teach offset” is displayed.
10. If required, press **Save** to save the values. The old mastering values are deleted.

    To restore a lost first mastering, always save the values.

Axes A4, A5 and A6 are mechanically coupled. This means:
If the values for A4 are deleted, the values for A5 and A6 are also deleted.
If the values for A5 are deleted, the values for A6 are also deleted.

11. Remove the MEMD from the gauge cartridge and replace the protective cap.
12. Repeat steps 5 to 11 for all axes to be mastered.

    Exception: Not for A6 if this axis has a line mark.
13. Close the window.
14. This step is only to be performed if A6 has a line mark.
    a. In the main menu, select **Start-up** > **Master** > **Reference**.

       The option window **Reference mastering** is opened. A6 is displayed and is selected.
    b. Press **Master** to restore lost first mastering. A6 is removed from the option window.
    c. Close the window.
15. Disconnect the EtherCAT cable from X32 and the mastering box.

NOTICE Leave the signal cable connected to the mastering box and disconnect it as little as possible. The pluggability of the M8 sensor connector is limited. Frequent connection/disconnection can result in damage to the connector.

5.4.10 Manually unmastering axes

Description

The mastering values of the individual axes can be deleted. The axes do not move during unmastering.

Axes A4, A5 and A6 are mechanically coupled. This means:
If the values for A4 are deleted, the values for A5 and A6 are also deleted.
If the values for A5 are deleted, the values for A6 are also deleted.

NOTICE The software limit switches of an unmastered robot are deactivated. The robot can hit the end stop buffers, thus damaging the robot and making it necessary to exchange the buffers. An unmastered robot must not be jogged, if at all avoidable. If it must be jogged, the jog override must be reduced as far as possible.

Precondition

- No program is selected.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Master** > **Unmaster**. A window opens.
2. Select the axis to be unmastered.
3. Press **Unmaster**. The mastering data of the axis are deleted.
4. Repeat steps 2 and 3 for all axes to be unmastered.
5. Close the window.

5.5 Modifying software limit switches

There are 2 ways of modifying the software limit switches:

- Enter the desired values manually.

- Or automatically adapt the limit switches to one or more programs.

  The robot controller determines the minimum and maximum axis positions occurring in the program. These values can then be set as software limit switches.

Precondition

- "Expert" user group
- T1, T2 or AUT mode

Procedure

Modifying software limit switches manually:

1. In the main menu, select **Start-up** > **Service** > **Software limit switch**. The **Software limit switch** window is opened.
2. Modify the limit switches as required in the columns **Negative** and **Positive**.
3. Save the changes with **Save**.

Adapting software limit switches to a program:

1. In the main menu, select **Start-up** > **Service** > **Software limit switch**. The **Software limit switch** window is opened.
2. Click on **Auto detection**. The following message is displayed: *Auto detection is running.*
3. Start the program to which the limit switches are to be adapted. Execute the program completely and then cancel it.

   The maximum and minimum position reached by each axis is displayed in the **Software limit switch** window.
4. Repeat step 3 for all programs to which the limit switches are to be adapted.

   The maximum and minimum position reached by each axis in all executed programs is displayed in the **Software limit switch** window.
5. Once all desired programs have been executed, press **End** in the **Software limit switch** window.
6. Press **Save** to save the determined values as software limit switches.
7. If required, modify the automatically determined values manually.

> Recommendation: Reduce the determined minimum values by 5°. Increase the determined maximum values by 5°.
> This margin prevents the axes from reaching the limit switches during program execution and thus triggering a stop.

8. Save the changes with **Save**.

Description

Software limit switch window:

Fig. 5-18: Before automatic determination

Item	Description
1	Current negative limit switch
2	Current position of the axis
3	Current positive limit switch

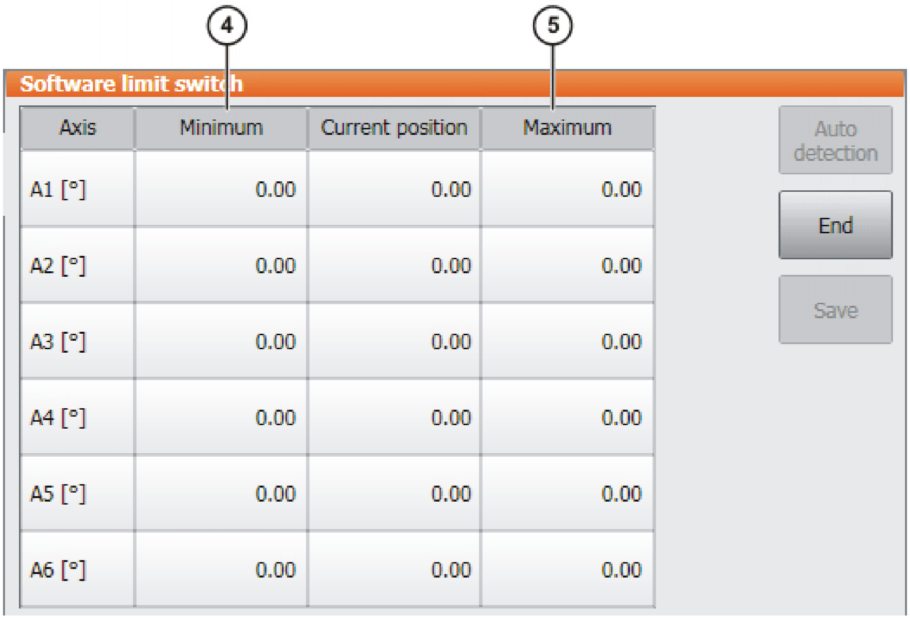


Fig. 5-19: During automatic determination

Item	Description
4	Minimum position of the axis since the start of determination
5	Maximum position of the axis since the start of determination

Buttons

The following buttons are available (only in the “Expert” user group):

Button	Description
Auto detection	Starts the automatic determination: The robot controller writes the minimum and maximum positions adopted by the axes from now on to the columns **Minimum** and **Maximum** in the **Software limit switch** window.
End	Ends the automatic determination. Transfers the calculated minimum/maximum positions to the columns **Negative** and **Positive**, but does not yet save them.
Save	Saves the values in the columns **Negative** and **Positive** as software limit switches.

5.6 Calibration

5.6.1 Tool calibration

Description

During tool calibration, the user assigns a Cartesian coordinate system to the tool mounted on the mounting flange. This coordinate system is called the TOOL coordinate system.

The TOOL coordinate system has its origin at a user-defined point. This is called the TCP (Tool Center Point). The TCP is generally situated at the working point of the tool.

> In the case of a fixed tool, the type of calibration described here must not be used. A separate type of calibration must be used for fixed tools. (>>> 5.6.3 "Fixed tool calibration" Page 122)

Advantages of tool calibration:

- The tool can be moved in a straight line in the tool direction.
- The tool can be rotated about the TCP without changing the position of the TCP.
- In program mode: The programmed velocity is maintained at the TCP along the path.

The number of TOOL coordinate systems that can be saved depends on the configuration in WorkVisual. Default: 16 TOOL coordinate systems. Variable: TOOL_DATA[1 … 16].

The following data are saved:

- X, Y, Z:
  Origin of the TOOL coordinate system relative to the FLANGE coordinate system
- A, B, C:
  Orientation of the TOOL coordinate system relative to the FLANGE coordinate system

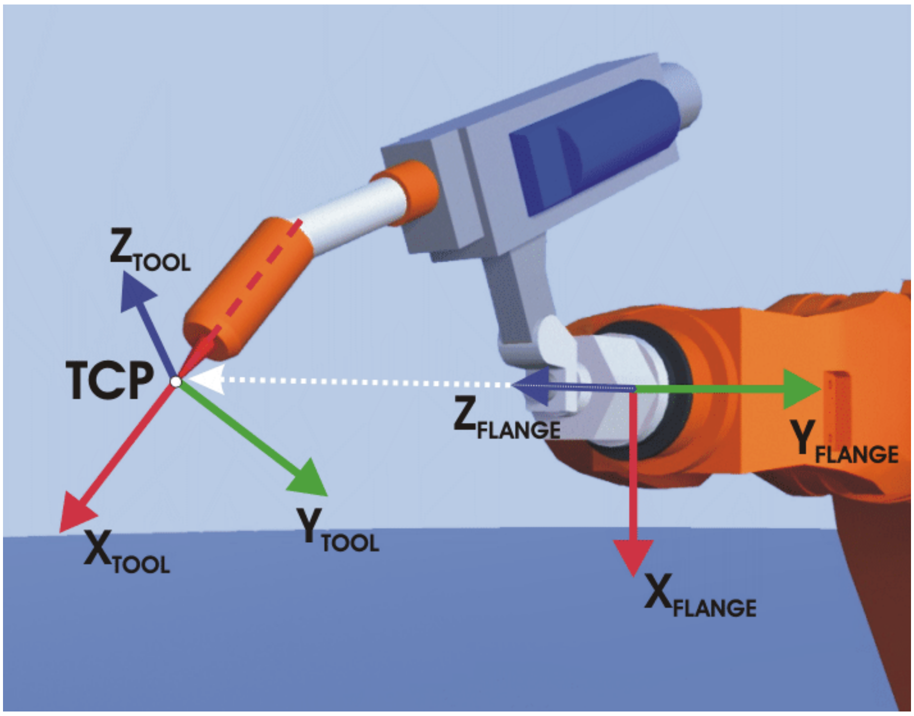


Fig. 5-20: TCP calibration principle

Overview

Tool calibration consists of 2 steps:

Step	Description
1	**Definition of the origin of the TOOL coordinate system** The following methods are available: ■ XYZ 4-point ■ XYZ Reference
2	**Definition of the orientation of the TOOL coordinate system** The following methods are available: ■ ABC 2-point ■ ABC World

If the calibration data are already known, they can be entered directly.
(>>> 5.6.1.5 "Numeric input" Page 118)

5.6.1.1 TCP calibration: XYZ 4-point method

The XYZ 4-point method cannot be used for palletizing robots.

Description

The TCP of the tool to be calibrated is moved to a reference point from 4 different directions. The reference point can be freely selected. The robot controller calculates the TCP from the different flange positions.

The 4 flange positions at the reference point must be sufficiently different from one another and must not lie in a plane.

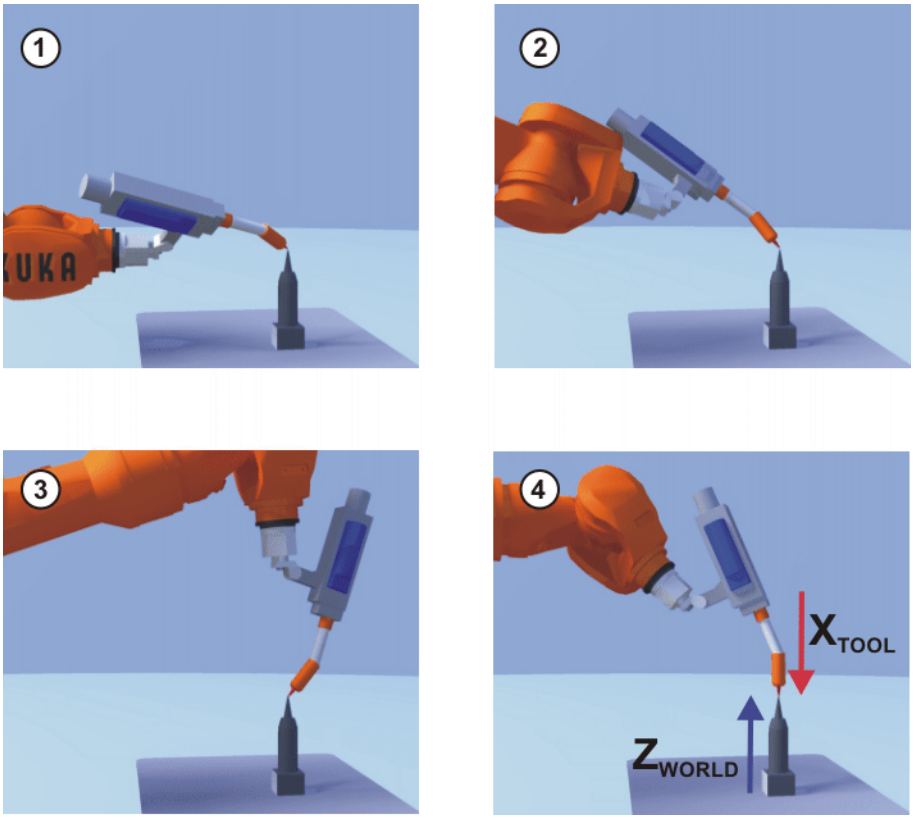


Fig. 5-21: XYZ 4-point method

Precondition

- The tool to be calibrated is mounted on the mounting flange.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** > **XYZ 4-point**.
2. Select a number for the tool that is to be calibrated and assign a tool name. Confirm with **Next**.
3. Move the TCP to a reference point. Press **Calibrate**. Answer the request for confirmation with **Yes**.
4. Move the TCP to the reference point from a different direction. Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Repeat step 4 twice.
6. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
7. Confirm with **Next**.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
9. Either: Press **Save** and then close the window via the **Close** icon.

   Or: Press **ABC 2-point** or **ABC World**. The previous data are automatically saved and a window is opened in which the orientation of the TOOL coordinate system can be defined.

   (>>> 5.6.1.4 "Defining the orientation: ABC 2-point method" Page 116)

   (>>> 5.6.1.3 "Defining the orientation: ABC World method" Page 116)

5.6.1.2 TCP calibration: XYZ Reference method

Description

In the case of the XYZ Reference method, a new tool is calibrated with a tool that has already been calibrated. The robot controller compares the flange positions and calculates the TCP of the new tool.

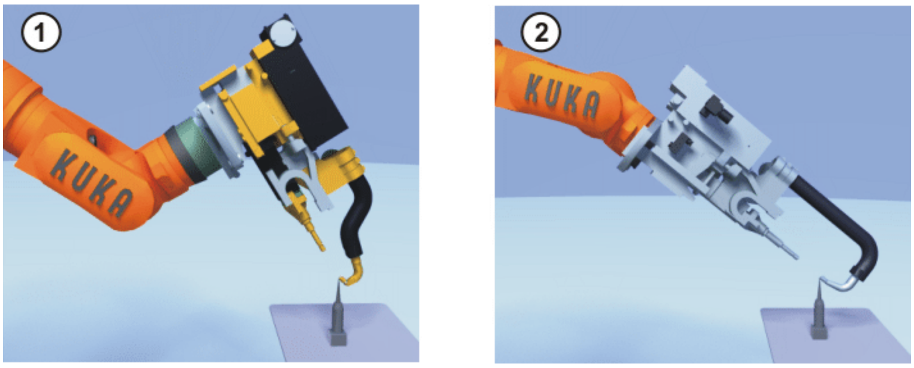


Fig. 5-22: XYZ Reference method

Precondition

- A previously calibrated tool is mounted on the mounting flange.
- Operating mode T1

Preparation

Calculate the TCP data of the calibrated tool:

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** > **XYZ Reference**.
2. Select the number of the calibrated tool.
3. The tool data are displayed. Note the X, Y and Z values.
4. Close the window.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** > **XYZ Reference**.
2. Select a number for the new tool and assign a tool name. Confirm with **Next**.
3. Enter the TCP data of the calibrated tool. Confirm with **Next**.
4. Move the TCP to a reference point. Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Move the tool away and remove it. Mount the new tool.
6. Move the TCP of the new tool to the reference point. Press **Calibrate**. Answer the request for confirmation with **Yes**.
7. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
8. Confirm with **Next**.
9. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
10. Either: Press **Save** and then close the window via the **Close** icon.

    Or: Press **ABC 2-point** or **ABC World**. The previous data are automatically saved and a window is opened in which the orientation of the TOOL coordinate system can be defined.

    (>>> 5.6.1.4 "Defining the orientation: ABC 2-point method" Page 116)
    (>>> 5.6.1.3 "Defining the orientation: ABC World method" Page 116)

5.6.1.3 Defining the orientation: ABC World method

Description

The user aligns the axes of the TOOL coordinate system parallel to the axes of the WORLD coordinate system. This communicates the orientation of the TOOL coordinate system to the robot controller.

There are 2 variants of this method:

- **5D**: The user communicates the tool direction to the robot controller. By default, the tool direction is the X axis. The orientation of the other axes is defined by the system and cannot be influenced by the user.

  The system always defines the orientation of the other axes in the same way. If the tool subsequently has to be calibrated again, e.g. after a crash, it is therefore sufficient to define the tool direction again. Rotation about the tool direction need not be taken into consideration.
- **6D**: The user communicates the direction of all 3 axes to the robot controller.

Precondition

- The tool to be calibrated is mounted on the mounting flange.
- The TCP of the tool has already been measured.
- T1 mode

> The following procedure applies if the tool direction is the default tool direction (= X axis). If the tool direction has been changed to Y or Z, the procedure must also be changed accordingly.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** > **ABC World**.
2. Select the number of the tool to be calibrated. Confirm with **Next**.
3. Select a variant in the **5D / 6D** box. Confirm with **Next**.
4. If **5D** is selected:

   Align $+X_{TOOL}$ parallel to $-Z_{WORLD}$. ($+X_{TOOL}$ = tool direction)

   If **6D** is selected:

   Align the axes of the TOOL coordinate system as follows.

   - $+X_{TOOL}$ parallel to $-Z_{WORLD}$. ($+X_{TOOL}$ = tool direction)
   - $+Y_{TOOL}$ parallel to $+Y_{WORLD}$.
   - $+Z_{TOOL}$ parallel to $+X_{WORLD}$.
5. Press **Calibrate**. Answer the request for confirmation with **Yes**.

> The following two steps are not required if the procedure is not called via the main menu, but by means of the **ABC World** button after TCP calibration.

6. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
7. Confirm with **Next**.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
9. Press **Save**.

5.6.1.4 Defining the orientation: ABC 2-point method

Description

The axes of the TOOL coordinate system are communicated to the robot controller by moving to a point on the X axis and a point in the XY plane.

This method is used if it is necessary to define the axis directions with particular precision.

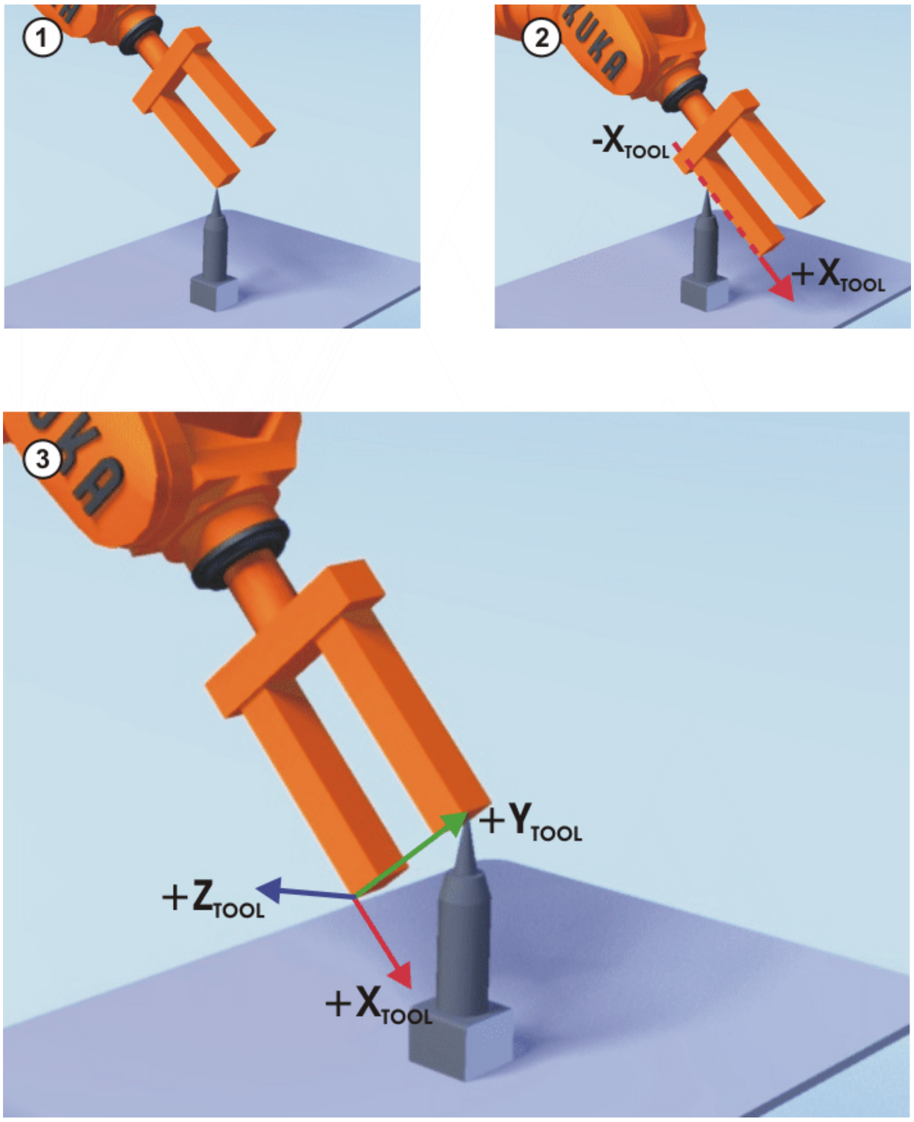


Fig. 5-23: ABC 2-point method

Precondition

- The tool to be calibrated is mounted on the mounting flange.
- The TCP of the tool has already been measured.
- T1 mode

> The following procedure applies if the tool direction is the default tool direction (= X axis). If the tool direction has been changed to Y or Z, the procedure must also be changed accordingly.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** > **ABC 2-point**.
2. Select the number of the tool to be calibrated. Confirm with **Next**.
3. Move the TCP to any reference point. Press **Calibrate**. Answer the request for confirmation with **Yes**.
4. Move the tool so that the reference point on the X axis has a negative X value (i.e. move against the tool direction). Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Move the tool so that the reference point in the XY plane has a positive Y value. Press **Calibrate**. Answer the request for confirmation with **Yes**.

The following two steps are not required if the procedure is not called via the main menu, but by means of the **ABC 2-point** button after TCP calibration.

6. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
7. Confirm with **Next**.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
9. Press **Save**.

5.6.1.5 Numeric input

Description

The tool data can be entered manually.

Possible sources of data:

- CAD
- Externally calibrated tool
- Specifications from the tool manufacturer

In the case of palletizing robots with 4 axes, the tool data must be entered numerically. The different XYZ and ABC methods cannot be used, as reorientation of these robots is highly restricted.

Precondition

- The following values are known:
  - X, Y and Z relative to the FLANGE coordinate system
  - A, B and C relative to the FLANGE coordinate system
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** >**Numeric input**.
2. Select a number for the tool and assign a tool name. Confirm with **Next**.
3. Enter the tool data. Confirm with **Next**.
4. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
5. If online load data checking is available (this depends on the robot type): configure as required.

   (>>> 5.7.5 "Online load data check" Page 137)
6. Confirm with **Next**.
7. Press **Save**.

5.6.2 Base calibration

Description

During base calibration, the user assigns a Cartesian coordinate system (BASE coordinate system) to a work surface or the workpiece. The BASE coordinate system has its origin at a user-defined point.

If the workpiece is mounted on the mounting flange, the type of calibration described here must not be used. A separate type of calibration must be used for workpieces mounted on the mounting flange.
(>>> 5.6.3 "Fixed tool calibration" Page 122)

Advantages of base calibration:

- The TCP can be jogged along the edges of the work surface or workpiece.
- Points can be taught relative to the base. If it is necessary to offset the base, e.g. because the work surface has been offset, the points move with it and do not need to be retaught.

The number of BASE coordinate systems that can be saved depends on the configuration in WorkVisual. Default: 32 BASE coordinate systems. Variable: BASE_DATA[1 … 32].

Overview

There are 2 ways of calibrating a base:

- 3-point method
  (>>> 5.6.2.1 "Base calibration: 3-point method" Page 119)
- Indirect method
  (>>> 5.6.2.2 "Base calibration: indirect method" Page 121)

If the calibration data are already known, they can be entered directly.
(>>> 5.6.2.3 "Entering the base numerically" Page 122)

5.6.2.1 Base calibration: 3-point method

Description

The robot moves to the origin and 2 further points of the new base. These 3 points define the new base.

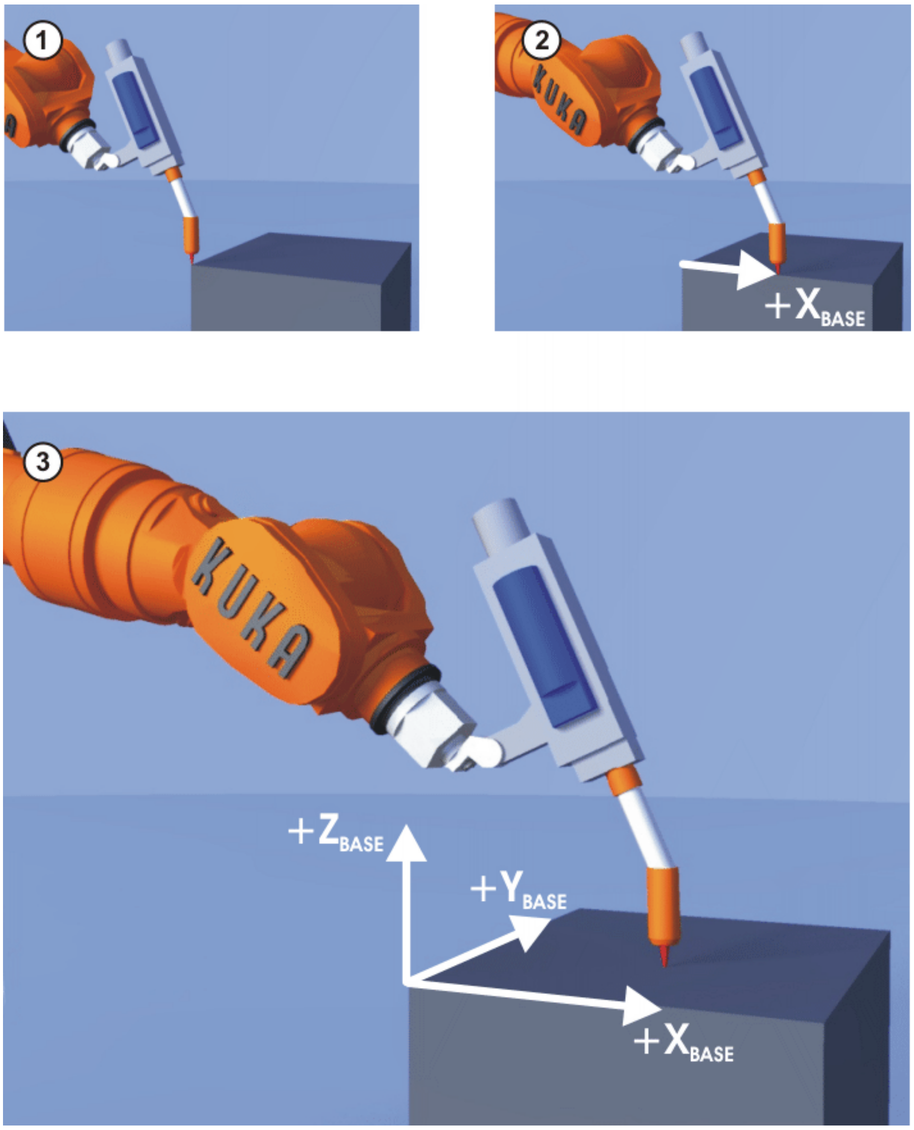


Fig. 5-24: 3-point method

Precondition

- A previously calibrated tool is mounted on the mounting flange.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Base** > **3-point**.
2. Assign a number and a name for the base that is to be calibrated. Confirm with **Next**.
3. Select the number of the tool that has already been calibrated. Confirm with **Next**.
4. Move the TCP to the origin of the new base. Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Move the TCP to a point on the positive X axis of the new base. Press **Calibrate**. Answer the request for confirmation with **Yes**.
6. Move the TCP to a point in the XY plane with a positive Y value. Press **Calibrate**. Answer the request for confirmation with **Yes**.
7. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
8. Press **Save**.

5.6.2.2 Base calibration: indirect method

Description

The indirect method is used if it is not possible to move to the origin of the base, e.g. because it is inside a workpiece or outside the workspace of the robot.

The TCP is moved to 4 points in the base, the coordinates of which must be known. The robot controller calculates the base from these points.

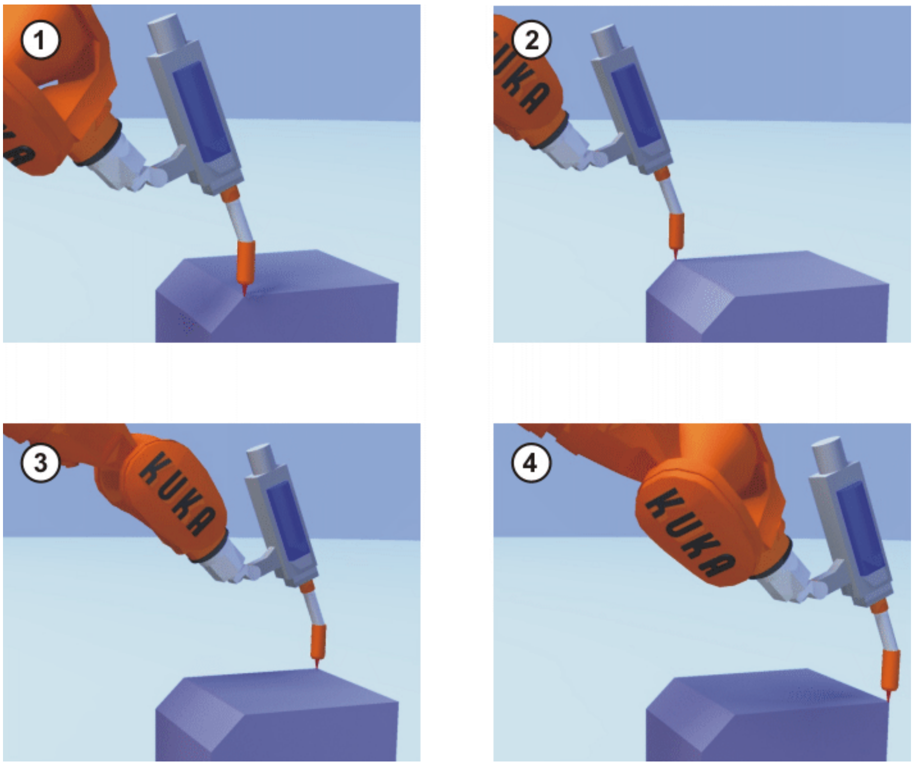


Fig. 5-25: Indirect method

Precondition

- A previously calibrated tool is mounted on the mounting flange.
- The coordinates of 4 points in the new base are known, e.g. from CAD data. The 4 points are accessible to the TCP.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Base** > **Indirect**.
2. Assign a number and a name for the base that is to be calibrated. Confirm with **Next**.
3. Select the number of the tool that has already been calibrated. Confirm with **Next**.
4. Enter the coordinates of a known point in the new base and move the TCP to this point. Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Repeat step 4 three times.
6. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
7. Press **Save**.

5.6.2.3 Entering the base numerically

Precondition

- The following numerical values are known, e.g. from CAD data:
  Distance between the origin of the base and the origin of the WORLD co-ordinate system
  Rotation of the base axes relative to the WORLD coordinate system
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Base** > **Numeric input**.
2. Assign a number and a name for the base that is to be calibrated. Confirm with **Next**.
3. Enter data. Confirm with **Next**.
4. Press **Save**.

5.6.3 Fixed tool calibration

Overview

Calibration of a fixed tool consists of 2 steps:

Step	Description
1	**Calibration of the TCP of the fixed tool** The TCP of a fixed tool is called an external TCP. (>>> 5.6.3.1 "Calibrating an external TCP" Page 122) If the calibration data are already known, they can be entered directly. (>>> 5.6.3.2 "Entering the external TCP numerically" Page 124)
2	**Calibration of the workpiece** The following methods are available: ▪ Direct method ▪ Indirect method

The robot controller saves the external TCP as the BASE coordinate system and the workpiece as the TOOL coordinate system.

5.6.3.1 Calibrating an external TCP

Description

First of all, the TCP of the fixed tool is communicated to the robot controller. This is done by moving a calibrated tool to it.

Then, the orientation of the coordinate system of the fixed tool is communicated to the robot controller. For this purpose, the coordinate system of the calibrated tool is aligned parallel to the new coordinate system. There are 2 variants:

- **5D**: Only the tool direction of the fixed tool is communicated to the robot controller. By default, the tool direction is the X axis. The orientation of the other axes is defined by the system and cannot be detected easily by the user.
- **6D**: The orientation of all 3 axes is communicated to the robot controller.

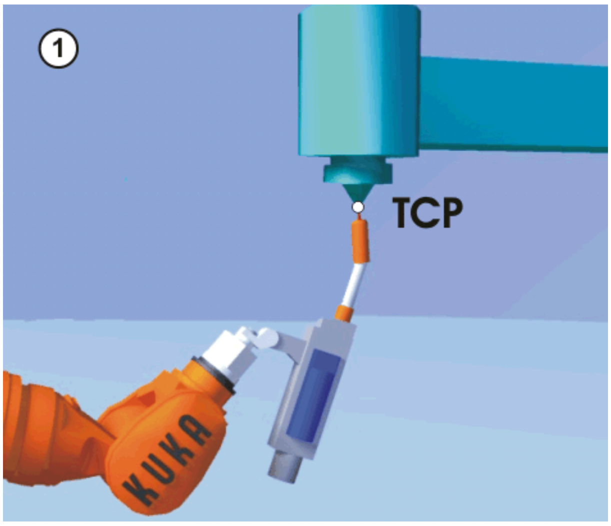


Fig. 5-26: Moving to the external TCP

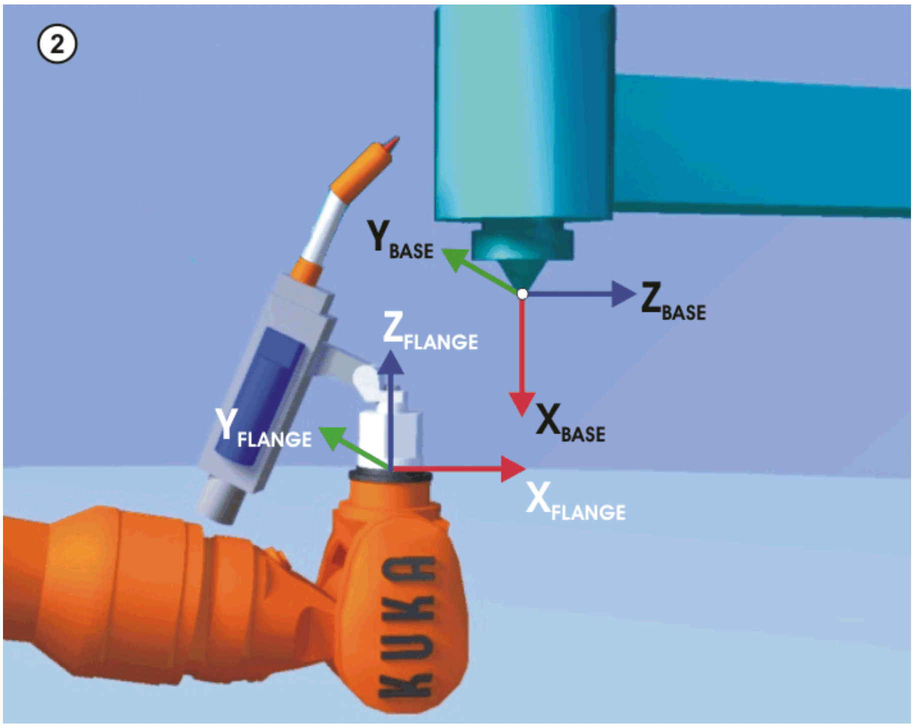


Fig. 5-27: Aligning the coordinate systems parallel to one another

Precondition

- A previously calibrated tool is mounted on the mounting flange.
- Operating mode T1

> The following procedure applies if the tool direction is the default tool direction (= X axis). If the tool direction has been changed to Y or Z, the procedure must also be changed accordingly.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Fixed tool** > **Tool**.
2. Select a number for the fixed tool that is to be calibrated and assign a tool name. Confirm with **Next**.
3. Select the number of the tool that has already been calibrated. Confirm with **Next**.
4. Select a variant in the **5D / 6D** box. Confirm with **Next**.
5. Move the TCP of the calibrated tool to the TCP of the fixed tool. Press **Calibrate**. Answer the request for confirmation with **Yes**.

6. If **5D** is selected:

   Align $+X_{BASE}$ parallel to $-Z_{FLANGE}$.

   (i.e. align the mounting flange perpendicular to the tool direction of the fixed tool.)

   If **6D** is selected:

   Align the mounting flange so that its axes are parallel to the axes of the fixed tool:

   - $+X_{BASE}$ parallel to $-Z_{FLANGE}$

     (i.e. align the mounting flange perpendicular to the tool direction.)
   - $+Y_{BASE}$ parallel to $+Y_{FLANGE}$
   - $+Z_{BASE}$ parallel to $+X_{FLANGE}$
7. Press **Calibrate**. Answer the request for confirmation with **Yes**.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
9. Press **Save**.

5.6.3.2 Entering the external TCP numerically

Precondition

- The following numerical values are known, e.g. from CAD data:
  - Distance between the TCP of the fixed tool and the origin of the WORLD coordinate system (X, Y, Z)
  - Rotation of the axes of the fixed tool relative to the WORLD coordinate system (A, B, C)
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Fixed tool** > **Numeric input**.
2. Select a number for the fixed tool and assign a tool name. Confirm with **Next**.
3. Enter data. Confirm with **Next**.
4. Press **Save**.

5.6.3.3 Workpiece calibration: direct method

Description

The origin and 2 further points of the workpiece are communicated to the robot controller. These 3 points uniquely define the workpiece.

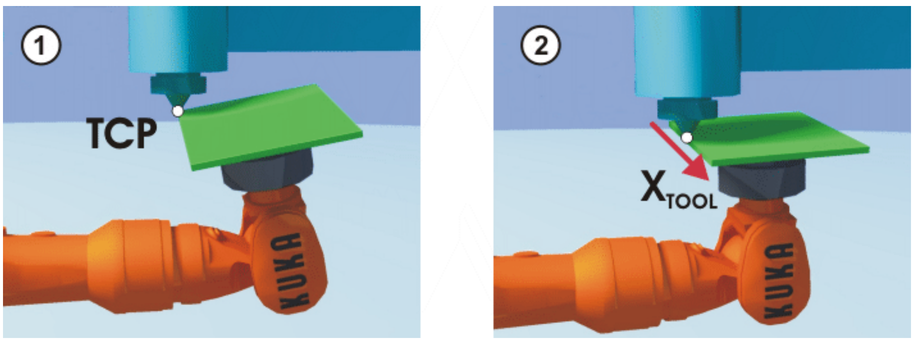


Fig. 5-28

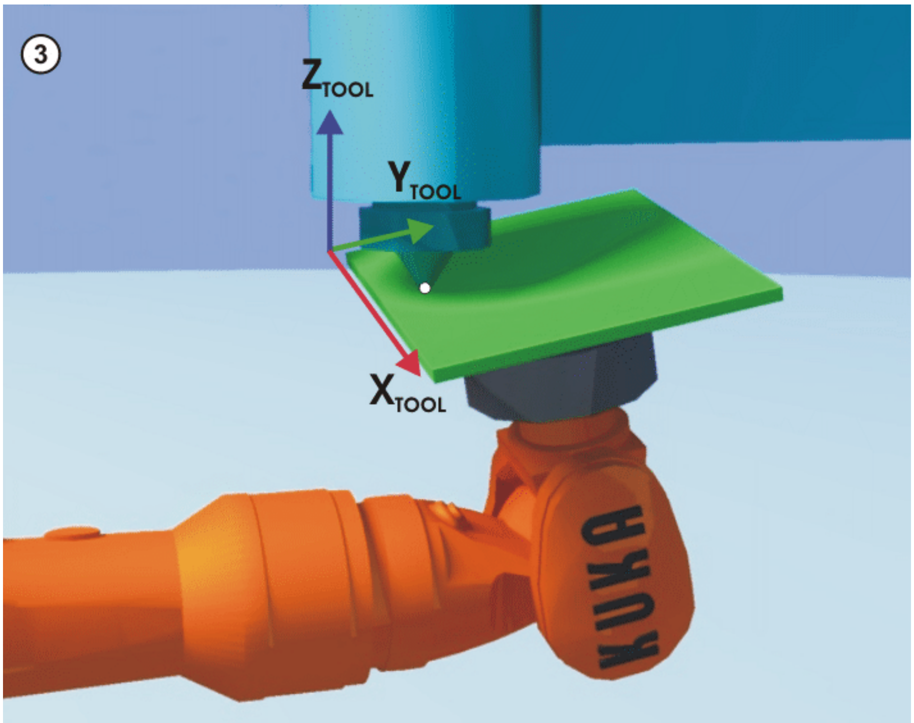


Fig. 5-29: Workpiece calibration: direct method

Precondition

- The workpiece to be calibrated is mounted on the mounting flange.
- A previously calibrated fixed tool is mounted.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Fixed tool** > **Workpiece** > **Direct calibration**.
2. Select a number for the workpiece to be calibrated and assign a workpiece name. Confirm with **Next**.
3. Select the number of the fixed tool. Confirm with **Next**.
4. Move the origin of the workpiece coordinate system to the TCP of the fixed tool. Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Move a point on the positive X axis of the workpiece coordinate system to the TCP of the fixed tool. Press **Calibrate**. Answer the request for confirmation with **Yes**.
6. Move a point with a positive Y value in the XY plane of the workpiece coordinate system to the TCP of the fixed tool. Press **Calibrate**. Answer the request for confirmation with **Yes**.
7. Enter the load data of the workpiece. (This step can be skipped if the load data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
8. Confirm with **Next**.
9. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
10. Press **Save**.

5.6.3.4 Workpiece calibration: indirect method

Description

The robot controller calculates the workpiece on the basis of 4 points whose coordinates must be known. The robot does not move to the origin of the workpiece.

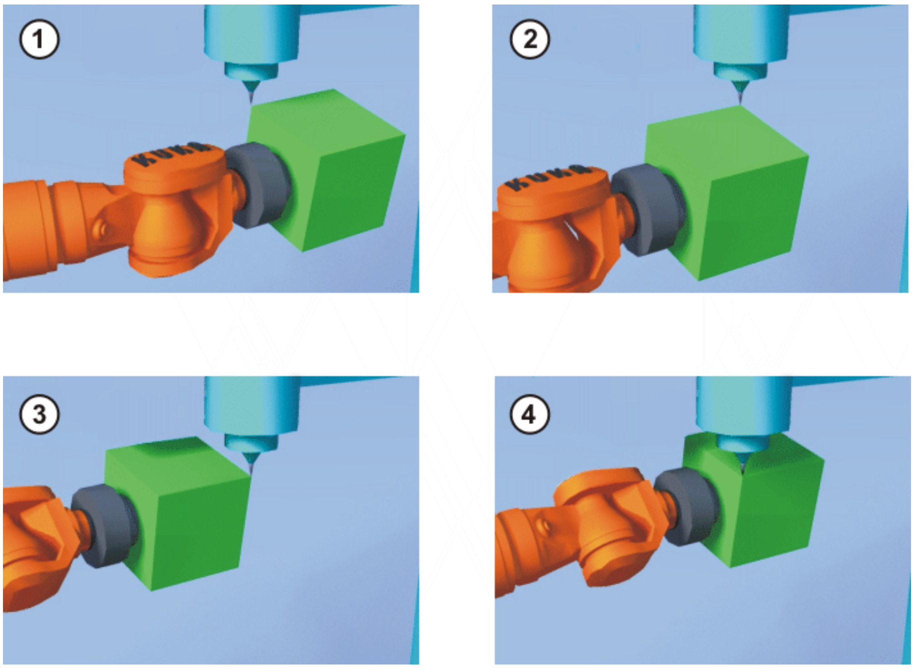


Fig. 5-30: Workpiece calibration: indirect method

Precondition

- The workpiece to be calibrated is mounted on the mounting flange.
- The coordinates of 4 points of the new workpiece are known, e.g. from CAD data. The 4 points are accessible to the TCP.
- A previously calibrated fixed tool is mounted.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Fixed tool** > **Workpiece** > **Indirect calibration**.
2. Select a number for the workpiece to be calibrated and assign a workpiece name. Confirm with **Next**.
3. Select the number of the fixed tool. Confirm with **Next**.
4. Enter the coordinates of a known point on the workpiece and move this point to the TCP of the fixed tool. Press **Calibrate**. Answer the request for confirmation with **Yes**.
5. Repeat step 4 three times.
6. Enter the load data of the workpiece. (This step can be skipped if the load data are entered separately instead.)

   (>>> 5.7.3 "Entering payload data" Page 136)
7. Confirm with **Next**.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press **Meas. points**. Then return to the previous view by pressing **Back**.
9. Press **Save**.

5.6.4 Renaming the tool/base

Precondition

- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** or **Base** > **Change name**.
2. Select the tool or base and press **Name**.
3. Enter the new name and confirm with **Save**.

5.6.5 Linear unit

The KUKA linear unit is a self-contained, one-axis linear unit mounted on the floor or ceiling. It is used for linear traversing of the robot and is controlled by the robot controller as an external axis.

The linear unit is a ROBROOT kinematic system. When the linear unit is moved, the position of the robot in the WORLD coordinate system changes. The current position of the robot in the WORLD coordinate system is defined by the vector $ROBROOT_C.

$ROBROOT_C consists of:

- $ERSYSROOT (static component)

  Root point of the linear unit relative to $WORLD. The root point is situated by default at the zero position of the linear unit and is not dependent on $MAMES.
- #ERSYS (dynamic component)

  Current position of the robot on the linear unit relative to the $ERSYS-ROOT

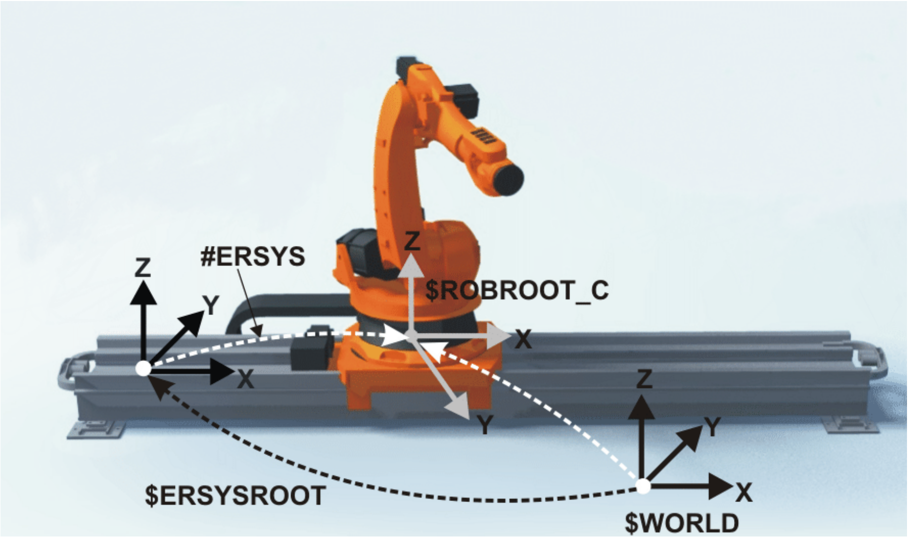


Fig. 5-31: ROBROOT kinematic system – linear unit

5.6.5.1 Checking whether the linear unit needs to be calibrated

Description

The robot is standing on the flange of the linear unit. Ideally, the ROBROOT coordinate system of the robot should be identical to the FLANGE coordinate system of the linear unit. In reality, there are often slight discrepancies which mean that positions cannot be moved to correctly. Calibration allows mathematical correction of these discrepancies. (Rotations about the direction of motion of the linear unit cannot be corrected. They do not, however, cause errors when moving to positions.)

If there are no discrepancies, the linear unit does not need to be calibrated. The following procedure can be used to determine whether calibration is required.

Precondition

- The machine data of the linear unit have been configured and loaded into the robot controller.
- A previously calibrated tool is mounted on the mounting flange.
- No program is open or selected.
- Operating mode T1

Procedure

1. Align the TCP against a freely selected point and observe it.
2. Execute a Cartesian (not axis-specific) motion with the linear unit.
   - If the TCP remains stationary: the linear unit does not require calibration.
   - If the TCP moves: the linear unit does require calibration.

If the calibration data are already known (e.g. from CAD), they can be entered numerically.

5.6.5.2 Calibrating the linear unit

Description

During calibration, the TCP of a tool that has already been calibrated is moved to a reference point 3 times.

- The reference point can be freely selected.
- The position of the robot on the linear unit from which the reference point is approached must be different all 3 times. The 3 positions must be far enough apart.

The correction values determined by the calibration are factored into the system variable $ET*x*_TFLA3.

Precondition

- The machine data of the linear unit have been configured and loaded into the robot controller.
- A previously calibrated tool is mounted on the mounting flange.
- No program is open or selected.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **External kinematic system** > **Linear unit**.
2. Select the number of the tool that has already been calibrated. Confirm with **Next**.

   The robot controller detects the linear unit automatically and displays the following data:

   - **Ext. kinematic system no.**: number of the external kinematic system (1 … 6) ($EX_KIN)
   - **Axis**: number of the external axis (1 … 6) ($ET*x*_AX)
   - **Name of ext. kinematic system:** ($ET*x*_NAME)

   (If the robot controller is unable to determine these values, e.g. because the linear unit has not yet been configured, calibration cannot be continued.)
3. Move the linear unit with the jog key “+”.
4. Specify whether the linear unit has moved to “+” or “-”. Confirm with **Next**.
5. Move the TCP to the reference point.
6. Press **Calibrate**.
7. Repeat steps 5 and 6 twice, but move the linear unit first each time in order to address the reference point from different positions.
8. Press **Save**. The calibration data are saved.
9. The system asks whether the positions that have already been taught are to be corrected.

- If no positions have been taught prior to the calibration, it makes no difference whether the question is answered with **Yes** or **No**.
- If positions have been taught prior to the calibration:

  Answering **Yes** will cause positions with base 0 to be corrected automatically. Other positions will not be corrected.

  Answering **No** will cause no positions to be corrected.

NOTICE After calibration of a linear unit, the following safety measures must be carried out:

1. Check the software limit switches of the linear unit and adapt them if required.
2. Test programs in T1.

Damage to property may otherwise result.

5.6.5.3 Entering the linear unit numerically

Precondition

- The machine data of the linear unit have been configured and loaded into the robot controller.
- No program is open or selected.
- The following numerical values are known, e.g. from CAD data:
  - Distance between the robot base flange and the origin of the ERSYSROOT coordinate system (X, Y, Z)
  - Orientation of the robot base flange relative to the ERSYSROOT coordinate system (A, B, C)
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **External kinematic system** > **Linear unit (numeric)**.

   The robot controller detects the linear unit automatically and displays the following data:
   - **Ext. kinematic system no.**: number of the external kinematic system (1 … 6)
   - **Axis**: number of the external axis (1 … 6)
   - **Name of ext. kinematic system:**

   (If the robot controller is unable to determine these values, e.g. because the linear unit has not yet been configured, calibration cannot be continued.)
2. Move the linear unit with the jog key “+”.
3. Specify whether the linear unit has moved to “+” or “-”. Confirm with **Next**.
4. Enter data. Confirm with **Next**.
5. Press **Save**. The calibration data are saved.
6. The system asks whether the positions that have already been taught are to be corrected.
   - If no positions have been taught prior to the calibration, it makes no difference whether the question is answered with **Yes** or **No**.
   - If positions have been taught prior to the calibration:

     Answering **Yes** will cause positions with base 0 to be corrected automatically. Other positions will not be corrected.

     Answering **No** will cause no positions to be corrected.

NOTICE After calibration of a linear unit, the following safety measures must be carried out:

1. Check the software limit switches of the linear unit and adapt them if required.
2. Test programs in T1.

Damage to property may otherwise result.

5.6.6 Calibrating an external kinematic system

Description

Calibration of the external kinematic system is necessary to enable the motion of the axes of the kinematic system to be synchronized and mathematically coupled with the robot axes. An external kinematic system can be a turn-tilt table or positioner, for example.

For linear units, the type of calibration described here must not be used. A separate type of calibration must be used for linear units. (>>> 5.6.5 "Linear unit" Page 127)

Overview

Calibration of an external kinematic system consists of 2 steps:

Step	Description
1	Calibrate the root point of the external kinematic system. (>>> 5.6.6.1 "Calibrating the root point" Page 130) If the calibration data are already known, they can be entered directly. (>>> 5.6.6.2 "Entering the root point numerically" Page 132)
2	If there is a workpiece on the external kinematic system: calibrate the base of the workpiece. (>>> 5.6.6.3 "Calibrating a workpiece base" Page 132) If the calibration data are already known, they can be entered directly. (>>> 5.6.6.4 "Entering the workpiece base numerically" Page 134)
	If there is a tool mounted on the external kinematic system: calibrate the external tool. (>>> 5.6.6.5 "Calibrating an external tool" Page 134) If the calibration data are already known, they can be entered directly. (>>> 5.6.6.6 "Entering the external tool numerically" Page 135)

5.6.6.1 Calibrating the root point

Description

In order to be able to move the robot with a mathematical coupling to a kinematic system, the robot must know the precise location of the kinematic system. This location is determined by means of root point calibration.

The TCP of a tool that has already been calibrated is moved to a reference point on the kinematic system 4 times. The position of the reference point must be different each time. This is achieved by moving the axes of the kinematic system. The robot controller uses the different positions of the reference point to calculate the root point of the kinematic system.

In the case of external kinematic systems from KUKA, the reference point is configured in the system variable $ET*x*_TPINFL in the machine data. This contains the position of the reference point relative to the FLANGE coordinate system of the kinematic system. (*x* = number of the kinematic system.) The reference point is also marked on the kinematic system. During calibration, this reference point must be addressed.

In the case of non-KUKA external kinematic systems, the reference point must be configured in the machine data.

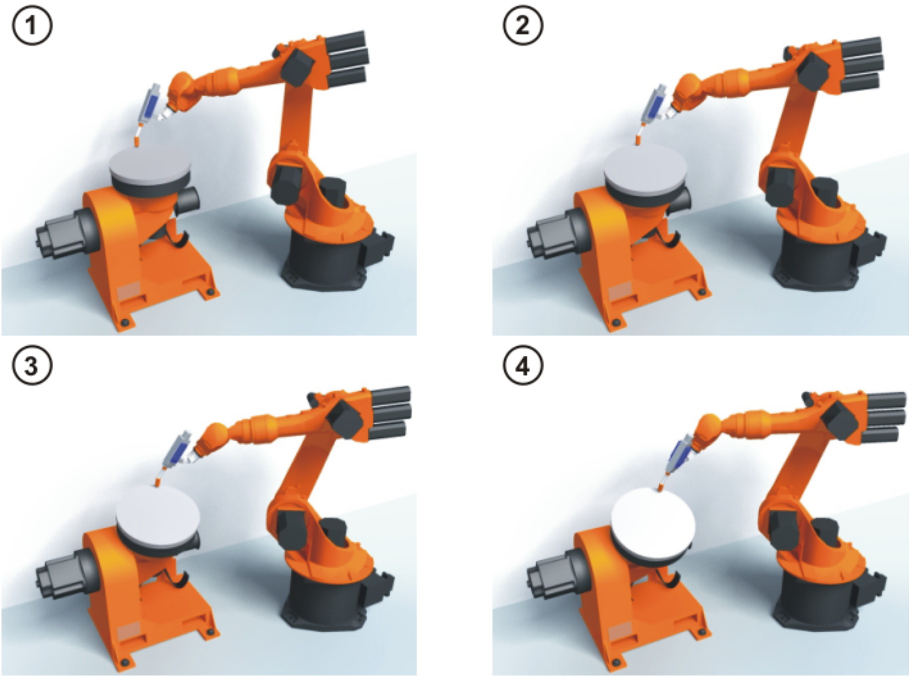


Fig. 5-32: Root point calibration principle

Precondition

- The machine data of the kinematic system have been configured and loaded into the robot controller.
- A previously calibrated tool is mounted on the mounting flange.
- If $ET*x*_TPINFL is to be modified: user group “Expert”
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **External kinematic system** > **Root point**.
2. Select the external kinematic system and assign a new kinematic name if required. Confirm with **Next**.
3. Select the number of the tool that has already been calibrated. Confirm with **Next**.
4. The value of $ET*x*_TPINFL is displayed.
   - If the value is not correct: the value can be modified here in the user group “Expert”.
   - If the value is correct: confirm with **Next**.
5. Move the TCP to the reference point.
6. Press **Calibrate**. Confirm with **Next**.
7. Repeat steps 5 and 6 three times. Each time, move the kinematic system first so that the reference point is approached from different positions.
8. Press **Save**.

5.6.6.2 Entering the root point numerically

Precondition

- The machine data of the kinematic system have been configured and loaded into the robot controller.
- The following numerical values are known, e.g. from CAD data:
  - Distance between the origin of the ROOT coordinate system and the origin of the WORLD coordinate system (X, Y, Z)
  - Orientation of the ROOT coordinate system relative to the WORLD coordinate system (A, B, C)
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **External kinematic system** > **Root point (numeric)**.
2. Select the external kinematic system and enter a new kinematic name if required. Confirm with **Next**.

   (The name is automatically also assigned to the BASE coordinate system.)
3. Enter the data of the ROOT coordinate system. Confirm with **Next**.
4. Press **Save**.

5.6.6.3 Calibrating a workpiece base

Description

During this calibration, the user assigns a BASE coordinate system to a workpiece located on the kinematic system. This BASE coordinate system is relative to the FLANGE coordinate system of the kinematic system. The base is thus a moving base that moves in the same way as the kinematic system.

It is not strictly necessary to calibrate a base. If none is calibrated, the FLANGE coordinate system of the kinematic system is taken as the base.

During calibration, the TCP of a calibrated tool is moved to the origin and 2 other points of the desired base. These 3 points define the base. Only one base can be calibrated per kinematic system.

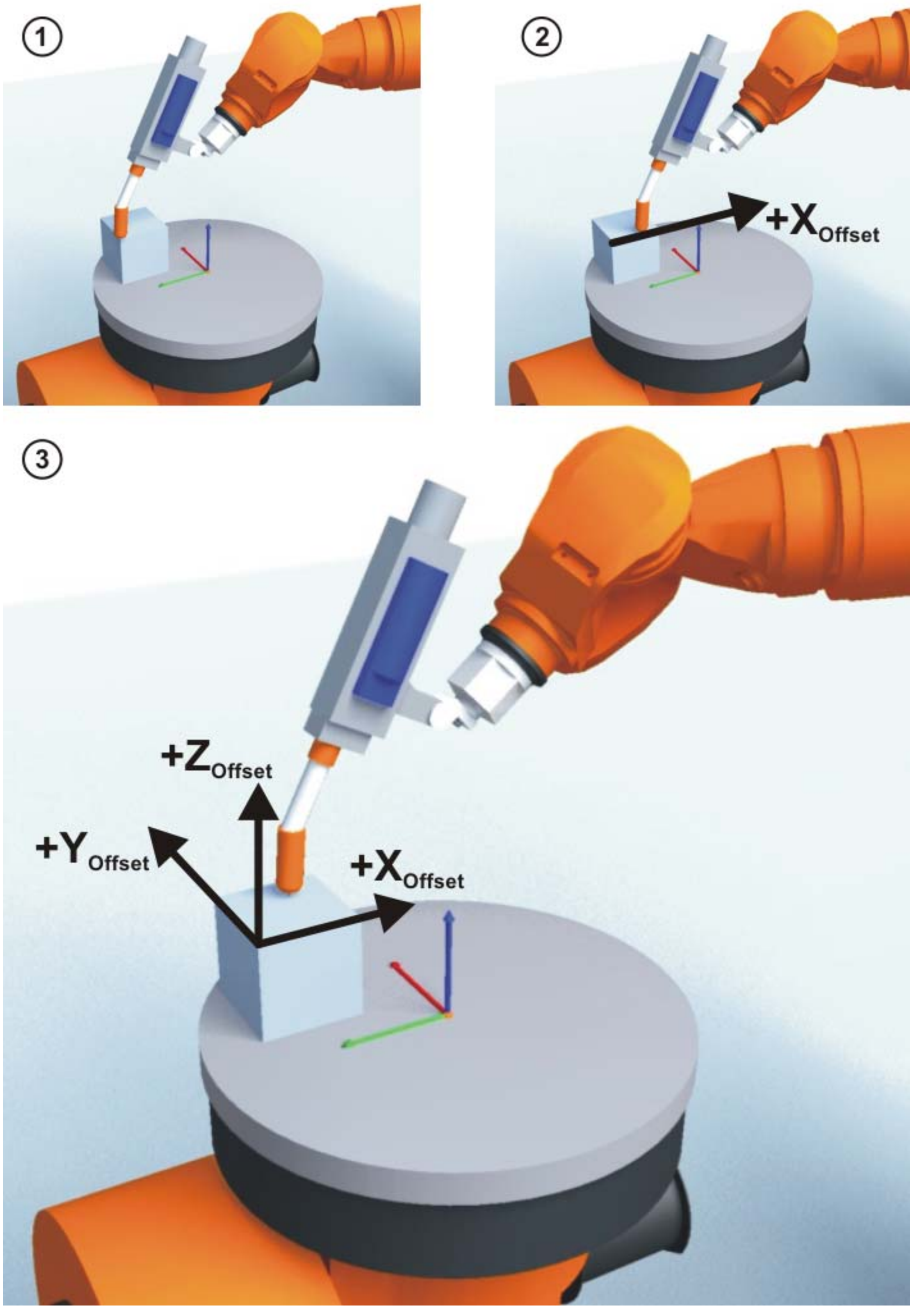


Fig. 5-33: Base calibration principle

Precondition

- The machine data of the kinematic system have been configured and loaded into the robot controller.
- A previously calibrated tool is mounted on the mounting flange.
- The root point of the external kinematic system has been calibrated.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **External kinematic system** > **Offset**.
2. Select the BASE coordinate system for which the offset is to be calibrated. Confirm with **Next**.
3. Select the external kinematic system. Confirm with **Next**.
4. Select the number of the tool that has already been calibrated. Confirm with **Next**.
5. Move the TCP to the origin of the workpiece base. Press **Calibrate** and confirm with **Next**.

6. Move the TCP to a point on the positive X axis of the workpiece base. Press **Calibrate** and confirm with **Next**.
7. Move the TCP to a point in the XY plane with a positive Y value. Press **Calibrate** and confirm with **Next**.
8. Press **Save**.

5.6.6.4 Entering the workpiece base numerically

Precondition

- The machine data of the kinematic system have been configured and loaded into the robot controller.
- The following numerical values are known, e.g. from CAD data:
  - Distance between the origin of the workpiece base and the origin of the FLANGE coordinate system of the kinematic system (X, Y, Z)
  - Rotation of the axes of the workpiece base relative to the FLANGE coordinate system of the kinematic system (A, B, C)
- The root point of the external kinematic system has been calibrated.
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **External kinematic system** > **Offset (numeric)**.
2. Select the BASE coordinate system for which the offset is to be calibrated. Confirm with **Next**.
3. Select the external kinematic system. Confirm with **Next**.
4. Enter data. Confirm with **Next**.
5. Press **Save**.

5.6.6.5 Calibrating an external tool

Description

During calibration of the external tool, the user assigns a coordinate system to the tool mounted on the kinematic system. This coordinate system has its origin in the TCP of the external tool and is relative to the FLANGE coordinate system of the kinematic system.

First of all, the user communicates to the robot controller the TCP of the tool mounted on the kinematic system. This is done by moving a calibrated tool to the TCP.

Then, the orientation of the coordinate system of the tool is communicated to the robot controller. For this purpose, the user aligns the coordinate system of the calibrated tool parallel to the new coordinate system. There are 2 variants:

- **5D**: The user communicates the tool direction to the robot controller. By default, the tool direction is the X axis. The orientation of the other axes is defined by the system and cannot be influenced by the user.

  The system always defines the orientation of the other axes in the same way. If the tool subsequently has to be calibrated again, e.g. after a crash, it is therefore sufficient to define the tool direction again. Rotation about the tool direction need not be taken into consideration.
- **6D**: The user communicates the direction of all 3 axes to the robot controller.

> If **6D** is selected: it is advisable to document the alignment of all axes. If the tool subsequently has to be calibrated again, e.g. after a crash, the axes must be aligned the same way as the first time in order to be able to continue moving to existing points correctly.

The robot controller saves the coordinates of the external tool as the BASE coordinate system.

Precondition

- The machine data of the kinematic system have been configured and loaded into the robot controller.
- A previously calibrated tool is mounted on the mounting flange.
- The root point of the external kinematic system has been calibrated.
- Operating mode T1

> The following procedure applies if the tool direction is the default tool direction (= X axis). If the tool direction has been changed to Y or Z, the procedure must also be changed accordingly.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Fixed tool** > **External kinematic offset**.
2. Select the number of the BASE coordinate system in which the coordinates of the external tool are saved and assign a name for the BASE coordinate system. Confirm with **Next**.
3. Select the external kinematic system. Confirm with **Next**.
4. Select the number of the tool that has already been calibrated. Confirm with **Next**.
5. Select a variant in the **5D / 6D** box. Confirm with **Next**.
6. Move the TCP of the calibrated tool to the TCP of the external tool. Press **Calibrate** and confirm with **Next**.
7. If **5D** is selected:

   Align $+X_{BASE}$ parallel to $-Z_{FLANGE}$.

   (i.e. align the mounting flange perpendicular to the tool direction of the external tool.)

   If **6D** is selected:

   Align the mounting flange so that its axes are parallel to the axes of the external tool:

   - $+X_{BASE}$ parallel to $-Z_{FLANGE}$

     (i.e. align the mounting flange perpendicular to the tool direction of the external tool.)
   - $+Y_{BASE}$ parallel to $+Y_{FLANGE}$
   - $+Z_{BASE}$ parallel to $+X_{FLANGE}$
8. Press **Calibrate** and confirm with **Next**.
9. Press **Save**.

5.6.6.6 Entering the external tool numerically

Precondition

- The following numerical values are known, e.g. from CAD data:
  - Distance between the TCP of the external tool and the origin of the FLANGE coordinate system of the kinematic system (X, Y, Z)
  - Rotation of the axes of the external tool relative to the FLANGE coordinate system of the kinematic system (A, B, C)
- Operating mode T1

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Fixed tool** > **Numeric input**.
2. Select a number for the external tool and assign a tool name. Confirm with **Next**.
3. Enter data. Confirm with **Next**.
4. Press **Save**.

5.7 Load data

The load data are factored into the calculation of the paths and accelerations and help to optimize the cycle times. The load data must be entered in the robot controller.

Sources

Load data can be obtained from the following sources:

- Software option KUKA.LoadDataDetermination (only for payloads on the flange)
- Manufacturer information
- Manual calculation
- CAD programs

5.7.1 Checking loads with KUKA.Load

All load data (payload and supplementary loads) must be checked with the KUKA.Load software. Exception: If the payload is checked with KUKA.LoadDataDetermination, it is not necessary to check it with KUKA.Load.

A sign-off sheet can be generated for the loads with KUKA.Load. KUKA.Load can be downloaded free of charge, complete with the documentation, from the KUKA website www.kuka.com.

> More information is contained in the **KUKA.Load** documentation.

5.7.2 Calculating payloads with KUKA.LoadDataDetermination

Description

KUKA.LoadDataDetermination can be used to calculate payloads exactly and transfer them to the robot controller.

Precondition

- T1 or T2 operating mode
- No program is selected.

Procedure

- In the main menu, select **Start-up** > **Service** > **Load data determination**.

> More information is contained in the **KUKA.LoadDataDetermination** documentation.

5.7.3 Entering payload data

Description

For a tool or workpiece on the flange, the payload data must be communicated to the robot controller.

The payload data can be entered numerically or determined with KUKA.LoadDataDetermination and transferred to the robot controller.

Precondition

- The payload data have been checked with KUKA.Load or KUKA.LoadDataDetermination and the robot is suitable for these payloads.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Tool** > **Tool load data**.
2. In the **Tool no.** box, select the number of the tool. Confirm with **Next**.
3. Enter the payload data:
   - Box **M**: Mass
   - Boxes **X**, **Y**, **Z**: Position of the center of gravity relative to the flange

- Boxes **A**, **B**, **C**: Orientation of the principal inertia axes relative to the flange
- Boxes **JX**, **JY**, **JZ**: Mass moments of inertia

  (JX is the inertia about the X axis of the coordinate system that is rotated relative to the flange by A, B and C. JY and JZ are the analogous moments of inertia about the Y and Z axes.)

Or, if the default values for this robot type are to be used: press **Default**.

4. If online load data checking is available (this depends on the robot type): configure as required.

   (>>> 5.7.5 "Online load data check" Page 137)

5. Confirm with **Next**.
6. Press **Save**.

5.7.4 Entering supplementary load data

Description

The supplementary load data must be entered in the robot controller.

Reference systems of the X, Y and Z values for each supplementary load:

Load	Reference system
Supplementary load A1	ROBROOT coordinate system A1 = 0°
Supplementary load A2	ROBROOT coordinate system A2 = -90°
Supplementary load A3	FLANGE coordinate system A4 = 0°, A5 = 0°, A6 = 0°

Precondition

- The supplementary loads have been verified with KUKA.Load and are suitable for this robot type.

Procedure

1. In the main menu, select **Start-up** > **Calibrate** > **Supplementary load data**.
2. Enter the number of the axis on which the supplementary load is to be mounted. Confirm with **Continue**.
3. Enter the load data. Confirm with **Continue**.
4. Press **Save**.

5.7.5 Online load data check

Configuration

OLDC can be configured as follows:

- During manual entry of the tool data

  (>>> 5.6.1.5 "Numeric input" Page 118)
- During separate entry of the payload data

  (>>> 5.7.3 "Entering payload data" Page 136)

The following boxes are displayed in the same window in which the payload data are also entered:

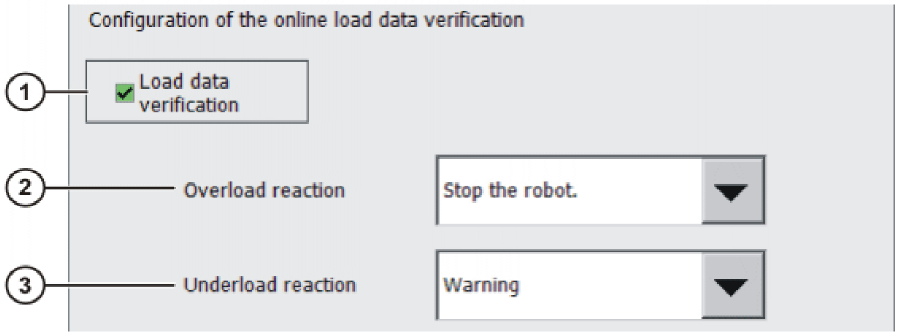


Fig. 5-34: Online load data check

Item	Description
1	**TRUE**: OLDC is activated for the tool displayed in the same window. The defined reactions are carried out in the case of an overload or underload. **FALSE**: OLDC is deactivated for the tool displayed in the same window. There is no reaction in the case of an overload or underload.
2	The overload reaction can be defined here. ■ **None**: No reaction. ■ **Warning**: The robot controller generates the following status message: *Check of robot load (Tool {No.}) calculated overload.* ■ **Stop robot**: The robot controller generates an acknowledgement message with the same content as that generated under **Warning**. The robot stops with a STOP 2.
3	The underload reaction can be defined here. The possible reactions are analogous to those for an overload.

The reactions can be modified in the KRL program using the system variable $LDC_CONFIG.

5.8 Exporting/importing long texts

Description

If names have been assigned to inputs/outputs, flags, etc., these names (so-called “long texts”) can be exported to a file. It is also possible to import a file with long text names. In this way, the long texts do not need to be re-entered manually for each robot after reinstallation.

The long texts can be exported to a USB stick or to the directory defined in the **Network archive path** box in the **Robot data** window. The same directories are also available as sources for the import function.

Precondition

- Either: USB stick
- Or: The target is configured in the **Network archive path** box in the **Robot data** window.

For import only:

- The long text names are present in a TXT or CSV file.
- The file is structured in such a way that it can be imported.

A file that originated as a long text export is automatically structured in such a way that it can be re-imported. If a file is to be filled with names manually, it is advisable first to assign a few dummy long texts in the robot controller, then to perform an export and fill the exported file.

Procedure

1. If a USB stick is used, connect it to the cabinet or smartPAD.

2. In the main menu, select **Start-up** > **Service** > **Long texts**. The **Long texts** window opens.
3. Select the **Export** or **Import** tab as required. Make the required settings.
4. Press the **Export** or **Import** button.

   When the import is finished, the message *Import successful.* is displayed.

   When the export is finished, the message *Export successful.* is displayed.

“Export” tab

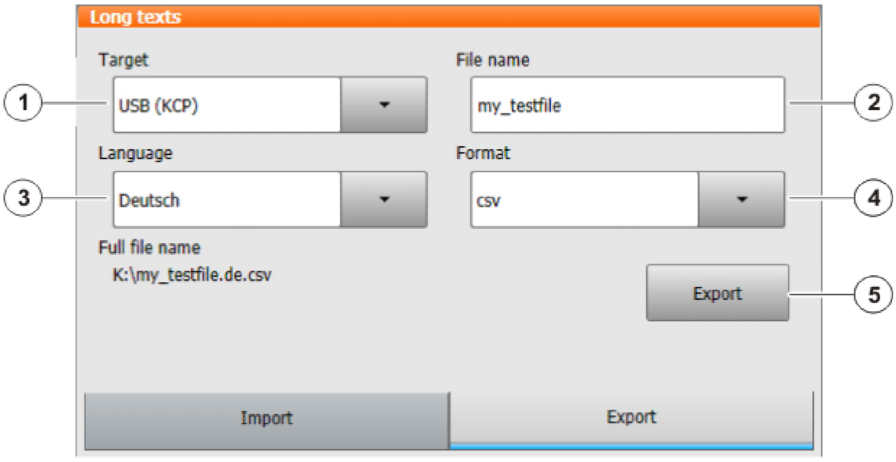


Fig. 5-35: Exporting long texts

Item	Description
1	Select the destination for the exported file. The entry **Network** is only available here if a path has been configured in the **Robot data** window.
2	Specify the desired file name. If **Network** is selected under item 1, the archive name configured in the **Robot data** window is displayed. The name can be changed here. This does not change it in the **Robot data** window. A suffix corresponding to the language selected is automatically appended to the name.
3	Select the language from which the long texts are to be exported. If, for example, the smartHMI is set to “English” and “Italiano” is selected here, a file with the suffix “it” is created. It contains the long texts that have been stored on the Italian smartHMI. It is also possible to select **All languages**.
4	Select the desired file format.
5	Starts the export.

"Import" tab

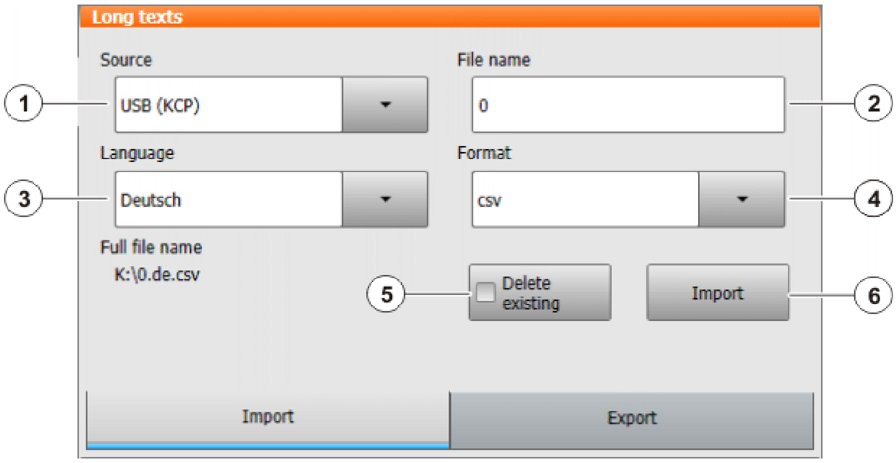


Fig. 5-36: Importing long texts

Item	Description
1	Specify the source from which files are to be imported. The entry **Network** is only available here if a path has been configured in the **Robot data** window.
2	Specify the name of the file to be imported, but without the language suffix. If **Network** is selected under item 1, the archive name configured in the **Robot data** window is displayed. The name can be changed here. This does not change it in the **Robot data** window.
3	Specify the language matching the language suffix of the file.
4	Specify the format of the file.
5	■ **Activated:** All existing long texts are deleted. The contents of the file are applied. ■ **Deactivated:** Entries in the file overwrite existing long texts. Existing long texts for which there is no entry in the file are retained.
6	Starts the import.

5.9 Maintenance handbook

The **Maintenance handbook** functionality is available in the KUKA System Software. The maintenance handbook enables logging of the maintenance work. The logged maintenance work can be displayed in an overview.

The robot controller uses messages to indicate when maintenance is due:

- A message is generated one month before the maintenance work is due. This message can be acknowledged.
- At the end of the month, the robot controller generates a message indicating that the maintenance is now due. This message cannot be acknowledged. Additionally, LED4 on the Controller System Panel flashes (= first LED from the left in the bottom row).

  Only when the corresponding maintenance work has been logged does the robot controller reset the message and the LED stops flashing.

> The controller variant "KR C4 compact" has no Controller System Panel and no flashing lights to indicate when maintenance work is due.

The due dates are determined by the maintenance intervals specified in the KUKA maintenance agreements. The intervals are counted from the initial start-up of the robot controller. The operating hours of the robot are counted.

5.9.1 Logging maintenance

Description

It is not possible to log multiple maintenance activities of the same kind on one day.

> Once saved, changes can no longer be made.

Precondition

- "Expert" user group

Procedure

1. In the main menu, select **Start-up** > **Service** > **Maintenance handbook**.
   The **Maintenance handbook** window is opened.
2. Select the **Maintenance input** tab and enter the maintenance details. Entries must be made in all boxes.
3. Press **Save**. A request for confirmation is displayed.
4. If all entries are correct, answer the request for confirmation with **Yes**.

The entries are now saved. Switching to the **Maintenance overview** tab causes the maintenance to be displayed there.

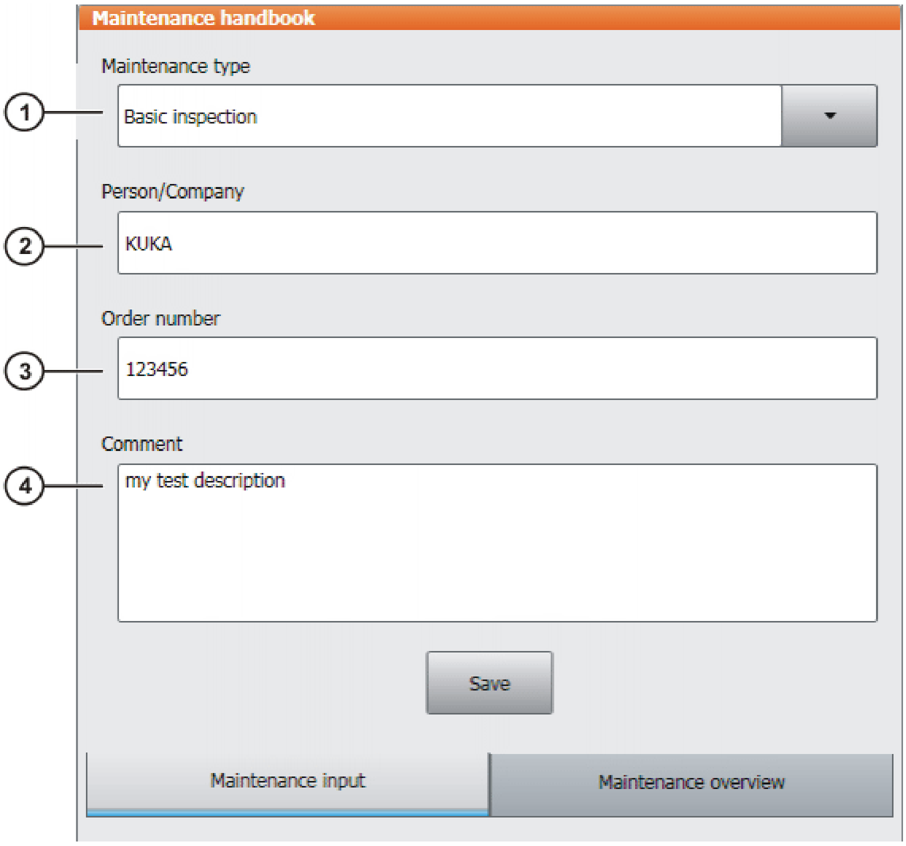


Fig. 5-37: Maintenance input

Item	Description
1	Select which type of maintenance has been carried out.
2	Enter who performed the maintenance.

Item	Description
3	For maintenance carried out and logged by KUKA employees: enter the order number. For other maintenance: enter any number.
4	Enter a comment.

Maintenance types

By default, the following maintenance types can be selected:

- **Basic inspection**
- **In-line wrist maintenance**
- **Main axis maintenance**
- **Gear backlash measurement**
- **Minor electrical maintenance**
- **Major electrical maintenance**
- **Data backup with spare hard drive**
- **Repair**

These maintenance types correspond to those in the KUKA maintenance agreements. Depending on the options used (e.g. linear axis or technology packages), other maintenance types may be available for selection.

5.9.2 Displaying a maintenance log

Description

The logged maintenance work can be displayed in an overview. If the KUKA System Software is updated, this overview is retained.

When archiving is carried out, the maintenance logs are also archived. If, when the data are restored, other maintenance work has been logged on the robot controller in the meantime, these logs are not overwritten; instead, the restored logs are added to the overview.

Procedure

1. In the main menu, select **Start-up** > **Service** > **Maintenance handbook**. The **Maintenance handbook** window is opened.
2. Select the **Maintenance overview** tab.

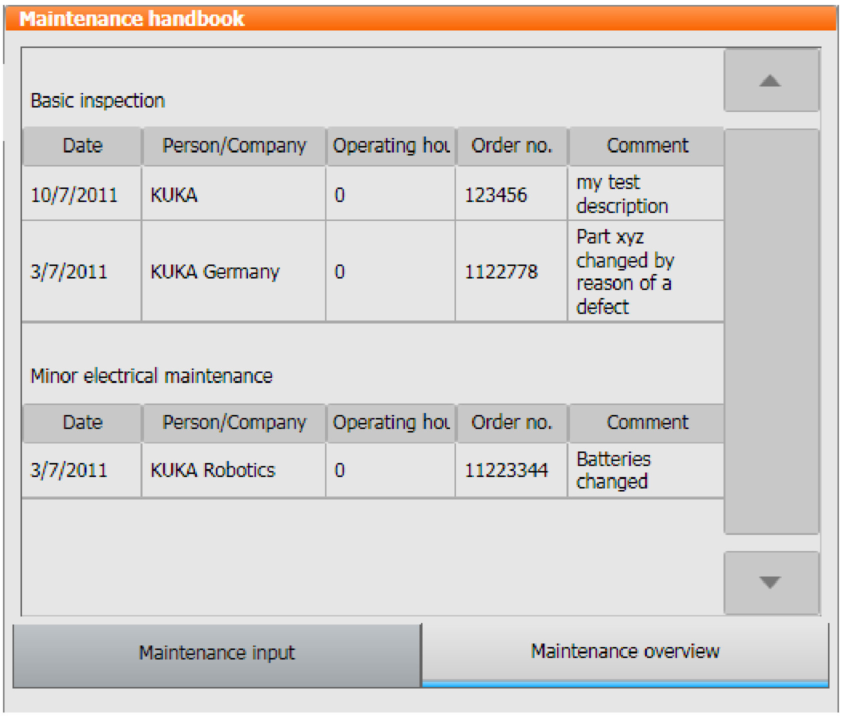


Fig. 5-38: Maintenance overview

6 Program and project management

6.1 Creating a new program

Description

It is not possible to select a template in the user group “User”. By default, a program of type “Module” is created.

Precondition

- The Navigator is displayed.

Procedure

1. In the directory structure, select the folder in which the program is to be created, e.g. the folder **Program**. (Not all folders allow the creation of programs within them.)
2. Press **New**.
3. Only in the user group “Expert”:

   The **Template selection** window is opened. Select the desired template and confirm with **OK**.
4. Enter a name for the program and confirm it with **OK**.

6.2 Creating a new folder

Precondition

- The Navigator is displayed.

Procedure

1. In the directory structure, select the folder in which the new folder is to be created, e.g. the folder **R1**.

   Not all folders allow the creation of new folders within them. In the user groups “Operator” and “User”, new folders can only be created in the folder **R1**.
2. Press **New**.
3. Enter a name for the folder and confirm it with **OK**.

6.3 Renaming a file or folder

Precondition

- The Navigator is displayed.

Procedure

1. In the directory structure, select the folder in which the file or folder to be renamed is located.
2. Select the file or folder in the file list.
3. Select **Edit** > **Rename**.
4. Overwrite the name with the new name and confirm with **OK**.

6.4 Navigator file manager

Overview

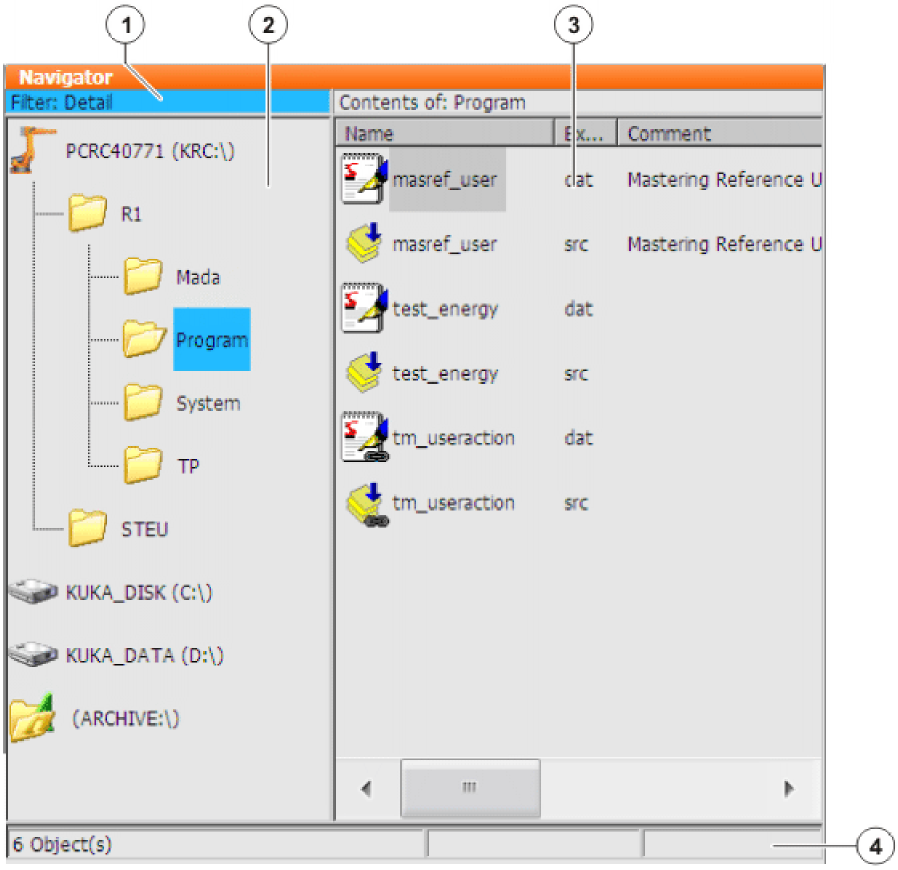


Fig. 6-1: Navigator

1	Header	3	File list
2	Directory structure	4	Status bar

Description

In the Navigator, the user manages programs and system-specific files.

Header

- Left-hand area: the selected filter is displayed.
  (>>> 6.4.1 "Selecting filters" Page 145)
- Right-hand area: the directory or drive selected in the directory structure is displayed.

Directory structure

Overview of directories and drives. Exactly which directories and drives are displayed depends on the user group and configuration.

File list

The contents of the directory or drive selected in the directory structure are displayed. The manner in which programs are displayed depends on the selected filter.

The file list has the following columns:

Column	Description
Name	Directory or file name
Extension	File extension This column is not displayed in the user group “User”.

Column	Description
Comment	Comment
Attribute	Attributes of the operating system and kernel system This column is not displayed in the user group “User”.
Size	File size in kilobytes This column is not displayed in the user group “User”.
#	Number of changes made to the file
Changed	Date and time of the last modification
Created	Date and time of file creation This column is not displayed in the user group “User”.

Status bar

The status bar can display the following information:

- Selected objects
- Action in progress
- User dialogs
- User entry prompts
- Requests for confirmation

6.4.1 Selecting filters

Description

This function is not available in the user group “User”.

The filter defines how programs are displayed in the file list. The following filters are available:

- **Detail**
  Programs are displayed as SRC and DAT files. (Default setting)
- **Modules**
  Programs are displayed as modules.

Precondition

- “Expert” user group

Procedure

1. Select the menu sequence **Edit** > **Filter**.
2. Select the desired filter in the left-hand section of the Navigator.
3. Confirm with **OK**.

6.5 Selecting or opening a program

Overview

A program can be selected or opened. Instead of the Navigator, an editor is then displayed with the program.

(>>> 6.5.1 "Selecting and deselecting a program" Page 146)

(>>> 6.5.2 "Opening a program" Page 147)

It is possible to toggle backwards and forwards between the program display and the Navigator.

(>>> 6.5.3 "Toggling between the Navigator and the program" Page 148)

Differences

Program is selected:

- The block pointer is displayed.
- The program can be started.

- The program can be edited to a certain extent.

  Selected programs are particularly suitable for editing in the user group "User".

  Example: KRL instructions covering several lines (e.g. LOOP … ENDLOOP) are not permissible.
- When the program is deselected, modifications are accepted without a request for confirmation. If impermissible modifications are programmed, an error message is displayed.

Program is open:

- The program cannot be started.
- The program can be edited.

  Opened programs are particularly suitable for editing in the user group "Expert".
- A request for confirmation is generated when the program is closed. Modifications can be accepted or rejected.

6.5.1 Selecting and deselecting a program

Precondition

- T1, T2 or AUT mode

Procedure

1. Select the program in the Navigator and press **Select**.

   The program is displayed in the editor. It is irrelevant whether a module, an SRC file or a DAT file is selected. It is always the SRC file that is displayed in the editor.
2. Start or edit the program.
3. Deselect the program again:

   Select **Edit** > **Cancel program**.

   or: In the status bar, touch the **Robot interpreter** status indicator. A window opens. Select **Cancel program**.

> When the program is deselected, modifications are accepted without a request for confirmation!

If the program is running, it must be stopped before it can be deselected.

Description

If a program is selected, this is indicated by the **Robot interpreter** status indicator.

(>>> 7.6 "Robot interpreter status indicator" Page 167)

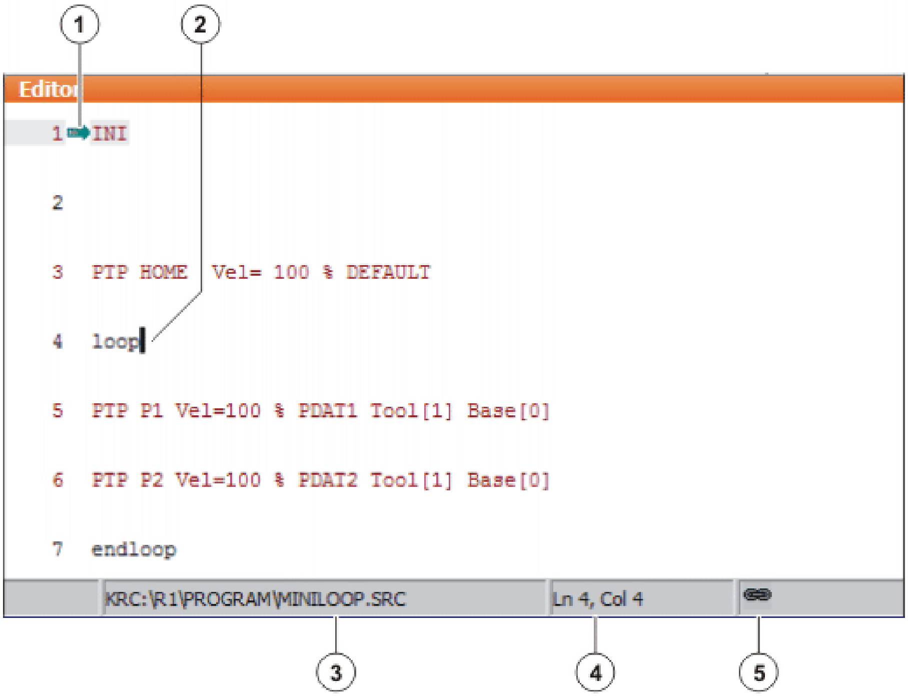


Fig. 6-2: Program is selected

1	Block pointer
2	Cursor
3	Program path and file name
4	Position of the cursor in the program
5	The icon indicates that the program is selected.

6.5.2 Opening a program

Precondition

- T1, T2 or AUT mode

A program can be opened in AUT EXT mode, but not edited.

Procedure

1. Select the program in the Navigator and press **Open**. The program is displayed in the editor.

   If a module has been selected, the SRC file is displayed in the editor. If an SRC file or DAT file has been selected, the corresponding file is displayed in the editor.
2. Edit the program.
3. Close the program.
4. To accept the changes, answer the request for confirmation with **Yes**.

Description

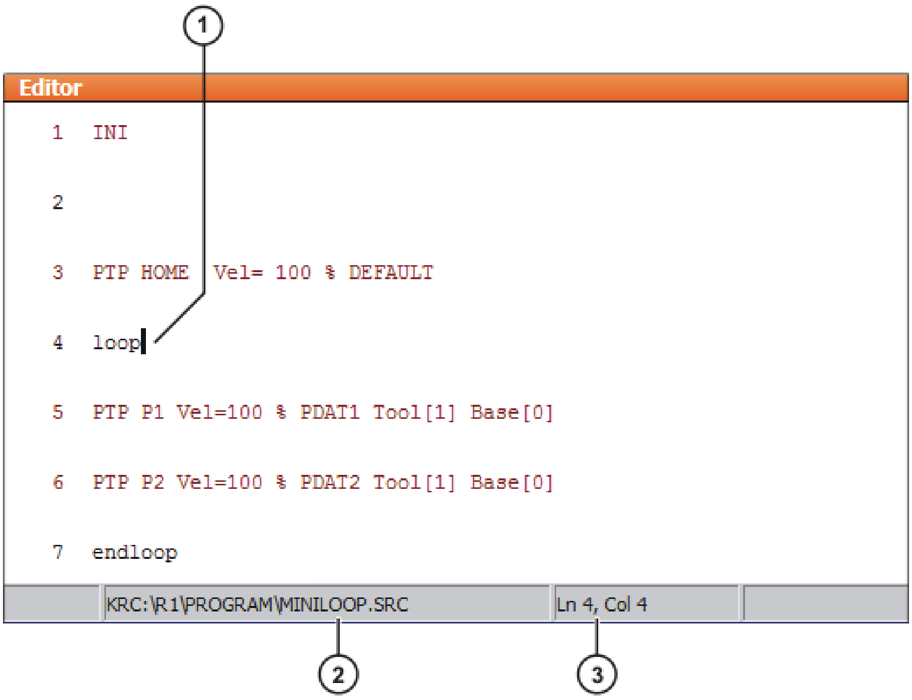


Fig. 6-3: Program is open

1	Cursor
2	Program path and file name
3	Position of the cursor in the program

6.5.3 Toggling between the Navigator and the program

Description

If a program is selected or open, it is possible to display the Navigator again without having to deselect or close the program. The user can then return to the program.

Procedure

Program is selected:

- Toggling from the program to the Navigator: select the menu sequence **Edit** > **Navigator**.
- Toggling from the Navigator to the program: press **PROGRAM**.

Program is open:

- Toggling from the program to the Navigator: select the menu sequence **Edit** > **Navigator**.
- Toggling from the Navigator to the program: press **EDITOR**.

> Programs that are running or have been interrupted must first be stopped before the menu sequences and buttons referred to above are available.

6.6 Structure of a KRL program

```
DEF my_program( )
INI

PTP HOME  Vel= 100 % DEFAULT
...
LIN point_5 CONT Vel= 2 m/s CPDAT1 Tool[3] Base[4]
```

```
   ...
14 PTP point_1 CONT Vel= 100 % PDAT1 Tool[3] Base[4]
   ...
20 PTP HOME  Vel= 100 % DEFAULT
21
22 END
```

Line	Description
1	The DEF line indicates the name of the program. If the program is a function, the DEF line begins with “DEFFCT” and contains additional information. The DEF line can be displayed or hidden. (>>> 6.7.1 "Displaying/hiding the DEF line" Page 150)
2	The INI line contains initializations for internal variables and parameters.
4	HOME position (>>> 6.6.1 "HOME position" Page 149)
8	LIN motion
14	PTP motion
20	HOME position
22	The END line is the last line in any program. If the program is a function, the wording of the END line is “ENDFCT”. The END line must not be deleted!

The first motion instruction in a KRL program must define an unambiguous starting position. The HOME position, which is stored by default in the robot controller, ensures that this is the case.

If the first motion instruction is not the default HOME position, or if this position has been changed, one of the following statements must be used:

- Complete PTP instruction of type POS or E6POS
- Complete PTP instruction of type AXIS or E6AXIS

“Complete” means that all components of the end point must be specified.

⚠ WARNING If the HOME position is modified, this affects all programs in which it is used. Injuries or damage to property may result.

In programs that are used exclusively as subprograms, different statements can be used as the first motion instruction.

6.6.1 HOME position

The HOME position is not program-specific. It is generally used as the first and last position in the program as it is uniquely defined and uncritical.

The HOME position is stored by default with the following values in the robot controller:

Axis	A1	A2	A3	A4	A5	A6
Pos.	0°	- 90°	+ 90°	0°	0°	0°

Additional HOME positions can be taught. A HOME position must meet the following conditions:

- Good starting position for program execution
- Good standstill position. For example, the stationary robot must not be an obstacle.

WARNING If the HOME position is modified, this affects all programs in which it is used. Injuries or damage to property may result.

6.7 Displaying/hiding program sections

6.7.1 Displaying/hiding the DEF line

Description

By default, the DEF line is hidden. Declarations can only be made in a program if the DEF line is visible.

The DEF line is displayed and hidden separately for opened and selected programs. If detail view (ASCII mode) is activated, the DEF line is visible and does not need to be activated separately.

Precondition

- User group “Expert”
- Program is selected or open.

Procedure

1. Select the menu sequence **Edit** > **View**. The subitem **DEF line** displays the current status:
   - **Check box not active**: The DEF line is hidden.
   - **Check box active**: The DEF line is displayed.

2. To change the status, touch the menu item **DEF line**.

   The menu then closes automatically.

6.7.2 Activating detail view

Description

Detail view (ASCII mode) is deactivated by default to keep the program transparent. If detail view is activated, hidden program lines, such as the FOLD and ENDFOLD lines and the DEF line, are displayed.

Detail view is activated and deactivated separately for opened and selected programs.

Precondition

- “Expert” user group

Procedure

1. Select the menu sequence **Edit** > **View**. The subitem **Detail view (ASCII)** displays the current status:
   - **Check box not active**: Detail view is deactivated.
   - **Check box active**: Detail view is activated.

2. To change the status, touch the menu item **Detail view (ASCII)**.

   The menu then closes automatically.

6.7.3 Activating/deactivating the line break function

Description

If a line is wider than the program window, the line is broken by default. The part of the line after the break has no line number and is marked with a black, L-shaped arrow.

```
8  EXT  IBGN (IBGN_COMMAND  :IN,BOOL  :IN,REAL  :IN,REAL
 ↳ :IN,BOOL  :IN,E6POS  :OUT )
```

Fig. 6-4: Line break

The line break function can be deactivated. If a line is wider than the program window, the line is no longer visible in its entirety. A scroll bar is displayed underneath the program window.

The line break function is activated and deactivated separately for opened and selected programs.

Precondition

- User group “Expert”
- Program is selected or open.

Procedure

1. Select the menu sequence **Edit** > **View**. The subitem **Line break** displays the current status:
   - **Check box not active**: Line break function is deactivated.
   - **Check box active**: Line break function is activated.

2. To change the status, touch the menu item **Line break**.

   The menu then closes automatically.

6.8 Editing programs

Overview

- A running program cannot be edited.
- Programs cannot be edited in AUT EXT mode.

Action	Possible in user group ...?
Insert comment or stamp	**User**: Yes **Expert**: Yes
Create folds	**User**: No **Expert**: Yes
Cut	**User**: No **Expert**: Yes
Copy	**User**: No **Expert**: Yes
Insert blank lines (press the Enter key)	**User**: No **Expert**: Yes
Paste	**User**: No **Expert**: Yes
Delete program lines	**User**: Yes **Expert**: Yes
Search	**User**: Yes **Expert**: Yes Possible for all user groups in an open program, even in AUT EXT mode.
Replace	**User**: No **Expert**: Yes (program is open, not selected)

Action	Possible in user group …?
Mark region	**User**: No **Expert**: Yes
Programming with inline forms	**User**: Yes **Expert**: Yes
KRL programming	**User**: Possible to a certain extent. KRL instructions covering several lines (e.g. LOOP … ENDLOOP) are not permissible. **Expert**: Yes

6.8.1 Inserting a comment or stamp

Precondition

- Program is selected or open.
- Operating mode T1

Procedure

1. Select the line after which the comment or stamp is to be inserted.
2. Select the menu sequence **Commands** > **Comment** > **Normal** or **Stamp**.
3. Enter the desired data. If a comment or stamp has already been entered previously, the inline form still contains the same entries.
   - In the case of a comment, the box can be cleared using **New text** ready for entry of a new text.
   - In the case of a stamp, the system time can also be updated using **New time** and the **NAME** box can be cleared using **New name**.
4. Save with **Cmd Ok**.

Description Comment

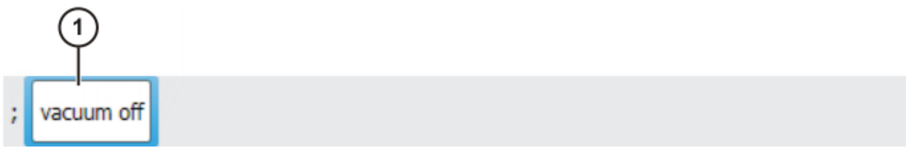


Fig. 6-5: Inline form “Comment”

Item	Description
1	Any text

Description Stamp

A stamp is a comment that is extended to include the system date and time and the user ID.

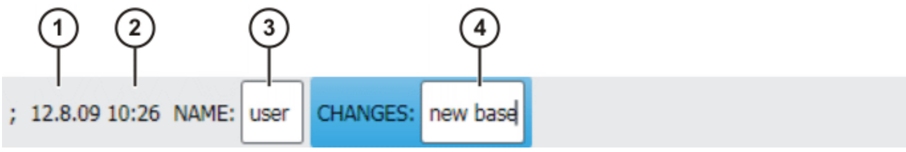


Fig. 6-6: Inline form “Stamp”

Item	Description
1	System date (cannot be edited)
2	System time
3	Name or ID of the user
4	Any text

6.8.2 Deleting program lines

Lines cannot be restored once they have been deleted!

Description

If a program line containing a motion instruction is deleted, the point name and coordinates remain saved in the DAT file. The point can be used in other motion instructions and does not need to be taught again.

Precondition

- Program is selected or open.
- Operating mode T1

Procedure

1. Select the line or range that is to be deleted.

   If only one line is to be deleted, it does not have to have a color background. It is sufficient for the cursor to be positioned in the line.
2. Select the menu sequence **Edit** > **Delete**.
3. Confirm the request for confirmation with **Yes**.

6.8.3 Additional editing functions

The following additional program editing functions can be called using **Edit**:

Copy

Precondition:

- Program is selected or open.
- “Expert” user group
- Operating mode T1

Paste

Precondition:

- Program is selected or open.
- “Expert” user group
- Operating mode T1

Cut

Precondition:

- Program is selected or open.
- “Expert” user group
- Operating mode T1

Find

Precondition:

- Program is selected or open.

Replace

Precondition:

- Program has been opened.
- “Expert” user group
- Operating mode T1

6.9 Printing a program

Procedure

1. Select the program in the Navigator. Multiple program selection is also possible.
2. Select the menu sequence **Edit** > **Print**.

6.10 Archiving and restoring data

6.10.1 Archiving overview

Target locations

Archiving can be performed to the following target destinations:

- USB stick in smartPAD or robot controller
- Network

Menu items

The following menu items are available:

(“*.*” means all files and subdirectories.)

Menu item	Archives the directories/files
All	■ KRC:*.* ■ C:\KRC\Roboter\Config\User*.* ■ C:\KRC\Roboter\Config\System\Common\Mada*.* ■ C:\KRC\Roboter\Init*.* ■ C:\KRC\Roboter\Ir_Spec*.* ■ C:\KRC\Roboter\Template*.* ■ C:\KRC\Roboter\Rdc*.* ■ C:\KRC\User*.* ■ C:\KRC\Roboter\log\Mastery.log ■ Some additional log data
Applications	■ KRC:\R1\Program*.* ■ KRC:\R1\System*.* ■ KRC:\R1\cell*.* ■ KRC:\Steu\$config*.*
System data	■ KRC:\R1\Mada*.* ■ KRC:\R1\System*.* ■ KRC:\R1\TP*.* ■ KRC:\Steu\Mada*.* ■ C:\KRC\Roboter\Config\User*.* ■ C:\KRC\Roboter\Config\System\Common\Mada*.* ■ C:\KRC\Roboter\Init*.* ■ C:\KRC\Roboter\Ir_Spec*.* ■ C:\KRC\Roboter\Template*.* ■ C:\KRC\Roboter\Rdc*.* ■ C:\KRC\User*.*

Menu item	Archives the directories/files
Log data	■ C:\KRC\Roboter\log*.* Except: Poslog.xsl and files with the extension DMP ■ Some additional log data
KrcDiag	If it is necessary for an error to be analyzed by KUKA Deutschland GmbH, this menu item can be used to compress the data for sending to KUKA. A screenshot of the current view of the smartHMI is automatically generated for the data package. For this reason, display error-relevant information on the smartHMI before starting the procedure: For example, expand the message window or display the logbook. What information is useful here depends on the specific circumstances. In addition to the menu sequence **File** > **Archive**, there are other methods available for compressing these data. (>>> 6.10.6 "Automatically compressing data for error analysis (KRCDiag)" Page 157)

If archiving is carried out using the menu item **All** and there is an existing archive present, this will be overwritten.

If archiving is carried out using a menu item other than **All** or **KrcDiag** and an archive is already available, the robot controller compares its robot name with that in the archive. If the names are different, a request for confirmation is generated.

If archiving is carried out repeatedly via **KrcDiag**, a maximum of 10 archives can be created. Further archives will overwrite the oldest existing archive.

The logbook can also be activated. (>>> 6.10.4 "Archiving the logbook" Page 156)

6.10.2 Archiving to a USB stick

Description

This procedure generates a ZIP file on the stick. By default, this file has the same name as the robot. A different name can be defined for the file, however, under **Start-up** > **Robot data**.

The archive is displayed in the ARCHIVE:\ directory in the Navigator. Archiving is also carried out automatically to D:\ as well as to the stick. The file INTERN.ZIP is generated here.

Special case **KrcDiag**:

This menu item generates the folder **KRCDiag** on the stick. This contains a ZIP file. The ZIP file is also automatically archived in C:\KUKA\KRCDiag.

> **NOTICE** A non-bootable USB stick must be used.
> We recommend using a non-bootable KUKA stick. Data may be lost if a stick from a different manufacturer is used.

Procedure

1. Connect the USB stick (to smartPAD or cabinet).
2. In the main menu, select **File** > **Archive** > **USB (KCP)** or **USB (cabinet)** and then the desired menu item.
3. Confirm the request for confirmation with **Yes**. The archive is created.

   Once the archiving is completed, this is indicated in the message window.

Special case **KrcDiag**: If archiving is carried out using this menu item, a separate window indicates when archiving has been completed. The window is then automatically hidden again.

4. The stick can now be removed.

6.10.3 Archiving on the network

Description

This procedure generates a ZIP file on the network path. By default, this file has the same name as the robot. A different name can be defined for the file, however, under **Start-up** > **Robot data**.

The network path to which the data are to be archived must be configured under **Start-up** > **Robot data**. If a user name and password are required for archiving to this path, these can also be entered here.

The archive is displayed in the ARCHIVE:\ directory in the Navigator. Archiving is also carried out automatically to D:\ as well as to the network path. The file INTERN.ZIP is generated here.

Special case **KrcDiag**:

This menu item generates the folder **KRCDiag** on the network path. This contains a ZIP file. The ZIP file is also automatically archived in C:\KUKA\KRCDiag.

Precondition

- The network path to which the data are to be archived is configured.

Procedure

1. In the main menu, select **File** > **Archive** > **Network** and then the desired menu item.
2. Confirm the request for confirmation with **Yes**. The archive is created.

   Once the archiving is completed, this is indicated in the message window. Special case **KrcDiag**: If archiving is carried out using this menu item, a separate window indicates when archiving has been completed. The window is then automatically hidden again.

6.10.4 Archiving the logbook

Description

The file “Logbuch.txt” is generated as an archive in the directory C:\KRC\ROBOTER\LOG.

Procedure

- In the main menu, select **File** > **Archive** > **Logbook**.

  The archive is created. Once the archiving is completed, this is indicated in the message window.

6.10.5 Restoring data

⚠ WARNING Only KSS 8.3 archives may be loaded into KSS 8.3. If other archives are loaded, the following may occur:

- Error messages
- Robot controller is not operable.
- Personal injury and damage to property.

Description

The following menu items are available for restoring data:

- **All**
- **Applications**
- **System data**

If the archived files are not the same version as the files present in the system, an error message is generated during restoration.

Similarly, if the version of the archived technology packages does not match the installed version, an error message is generated.

Precondition

- If data are to be restored from the USB stick: A USB stick with the archive is connected.

  The stick can be connected to the smartPAD or robot controller.

NOTICE A non-bootable USB stick must be used. We recommend using a non-bootable KUKA stick. Data may be lost if a stick from a different manufacturer is used.

Procedure

1. In the main menu, select **File** > **Restore** and then the desired subitems.
2. Confirm the request for confirmation with **Yes**. Archived files are restored to the robot controller. A message indicates completion of the restoration process.
3. If data have been restored from a USB stick: the stick can now be removed.
4. Reboot the robot controller.

6.10.6 Automatically compressing data for error analysis (KRCDiag)

Description

If it is necessary for an error to be analyzed by KUKA Deutschland GmbH, this procedure can be used to compress the required data. The procedure generates a ZIP file in the directory C:\KUKA\KRCDiag. This contains the data which KUKA Deutschland GmbH requires to analyze an error. This includes information about the system resources, screenshots and much more.

Preparation

A screenshot of the current view of the smartHMI is automatically generated for the data packet.

- Therefore, if possible, display the information related to errors on the smartHMI before starting the procedure:

  e.g. expand the message window or display the logbook. What information is useful here depends on the specific circumstances.

Procedure via “Diagnosis”

- In the main menu, select **Diagnosis** > **KrcDiag**.

  The data are compressed. Progress is displayed in a window. Once the operation has been completed, this is also indicated in the window. The window is then automatically hidden again.

Procedure via smartPAD

This procedure uses keys on the smartPAD instead of menu items. It can thus also be used if the SmartHMI is not available, due to Windows problems for example.

Precondition:

- The smartPAD is connected to the robot controller.
- The robot controller is switched on.

The keys must be pressed within 2 seconds. Whether or not the main menu and keypad are displayed in the smartHMI is irrelevant.

1. Press the “Main menu” key and hold it down.
2. Press the keypad key twice.
3. Release the “Main menu” key.

The data are compressed. Progress is displayed in a window. Once the operation has been completed, this is also indicated in the window. The window is then automatically hidden again.

Procedure via "Archive"

Alternatively, the data can also be compressed via **File** > **Archive** > **[...]**. In this way, the data can be stored on a USB stick or network path.

(>>> 6.10 "Archiving and restoring data" Page 154)

6.11 Project management

6.11.1 Project management window

Overview

The **Project management** window is opened using the WorkVisual icon on the smartHMI.

In addition to the regular projects, the **Project management** window contains the following special projects:

Project	Description
Initial project	The initial project is always present. It cannot be changed by the user. It contains the initial state of the robot controller as shipped.
Base project	The user can save the active project as the base project. This functionality is generally used to save a functional, tried-and-tested project state. The base project cannot be activated, but copied. The base project can no longer be changed by the user. It can, however, be overwritten by saving a new base project (after a request for confirmation). If a project is activated which does not contain all the configuration files, the missing information is inserted from the base project. This is the case, for example, if a project is activated from an earlier version of WorkVisual. (The configuration files include machine data files, safety configuration files and many others.)

In the case of a KSS/VSS update, the initial project and base project are overwritten by copies of the active project.

Description

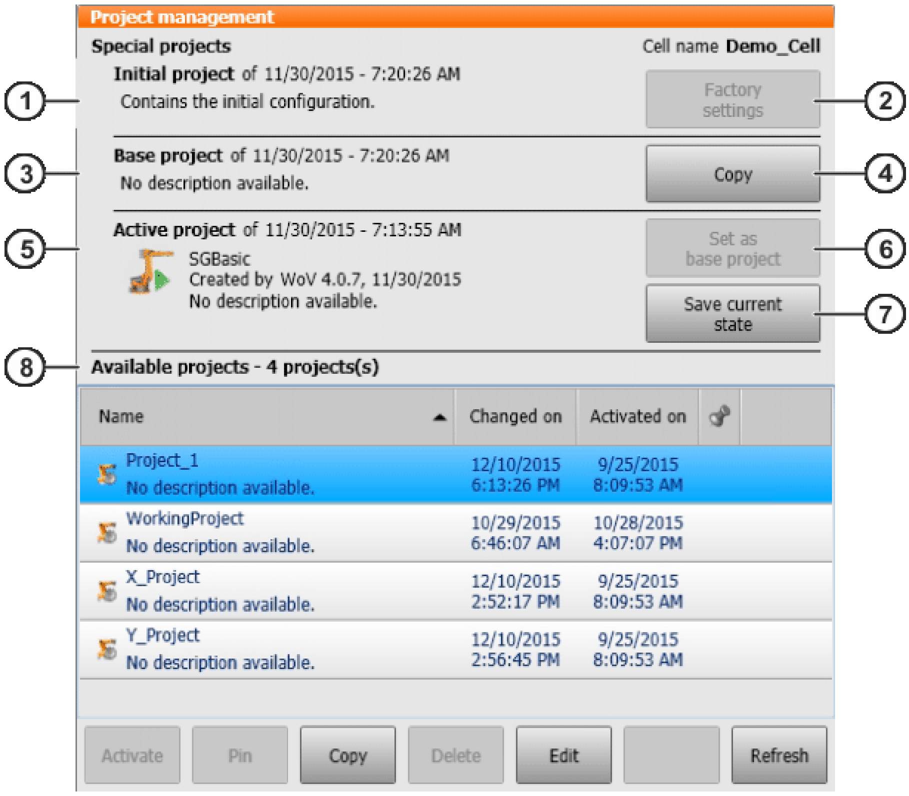


Fig. 6-7: “Projects management” window

Item	Description
1	The initial project is displayed.
2	Restores the factory settings of the robot controller. Only available to the user group “Expert” or higher.
3	The base project is displayed.
4	Creates a copy of the base project.
5	The active project is displayed.
6	Saves the active project as the base project. The active project remains active and the previous base project is deleted. Only available to the user group “Expert” or higher.
7	Creates a pinned copy of the active project.
8	List of inactive projects (except base and initial project)

With all copying operations, a window opens in which a name and a description can be entered for the copy.

Buttons

The following buttons are available:

Button	Description
Activate	Activates the selected project. If the selected project is pinned: Creates a copy of the selected project. (A pinned project cannot be activated itself, only a copy of it.) The user can then decide whether to activate this copy or whether the current project should remain active. Only available to the user group “Expert” or higher.
Pin	Pins the project. Only available if an unpinned project is selected. Only available to the user group “Expert” or higher.
Unpin	Unpins the project. Only available if a pinned project is selected. Only available to the user group “Expert” or higher.
Copy	Copies the selected project.
Delete	Deletes the selected project. Only available if a non-activated, unpinned project is selected. Only available to the user group “Expert” or higher.
Edit	Opens a window in which the name and/or description of the selected project can be changed.
Refresh	Refreshes the project list. This enables, for example, projects to be displayed which have been transferred to the robot controller since the display was opened.

6.11.2 Backing up projects, option packages and RDC data

Projects

The following projects are backed up by default:

- Active project
- Initial project
- Base project

Option packages

Option packages will be backed up under the following conditions:

- The option package has a KOP file.
- The option package was originally added to the project in WorkVisual. The project is now active on the robot controller.

  Or:

  The option package has been installed in the active project via **Start-up** > **Additional software**. The option package was available during installation as a single KOP file (not as a directory structure!).

RDC data

Every time a backup is made, a file with the name *[Robot_serial_number]*.RDC is created. It contains the CAL, MAM and PID files. Not all files are present in all cases (dependent on the robot).

Precondition

- There is currently no backup or restoration already running.

Procedure

Back up:

- In the main menu, select **File** > **Backup Manager** > **Back up**.

Save as... :

1. In the main menu, select **File** > **Backup Manager** > **Save as...**.
2. Enter the target in the **Target path for project backup** box.
3. If necessary, enter a separate path for the option packages.

   To do so, activate the check box **Divert path for option packages** and enter the desired path in the **Target path for KOP backup** box.
4. Start the backup with **Save**.

The backup is carried out. The robot controller displays messages when the backup has been successfully completed. It generates one message per project or option package and one message relating to the RDC data.

However, option packages will not be backed up if the same package version already exists in the target directory.

6.11.3 Restoring projects, option packages and RDC data

Detailed information about restoration is contained in the Operating and Programming Instructions for System Integrators.

Projects and option packages on the one hand and RDC data on the other can only be restored separately from one another.

Projects, option packages

No special preconditions are required for restoration as such.

After a restoration, however, an option package must generally be installed and/or a project activated. For this, it is necessary for the user to be in the user group “Expert” or higher.

RDC data

For the restoration of RDC data, it is necessary for the user to be in the user group “Expert” or higher.

7 Program execution

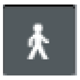
7.1 Selecting the program run mode

Procedure

1. Touch the **Program run mode** status indicator. The **Program run mode** window is opened.
2. Select the desired program run mode.

   The window closes and the selected program run mode is applied.

7.2 Program run modes

Designation	Status indicator	Description
Go #GO		The program is executed through to the end without stopping.
Motion #MSTEP		The program is executed with a stop at each point, including auxiliary points and the points of a spline segment. The Start key must be pressed again for each point. The program is executed without advance processing.
Single Step #ISTEP		The program is executed with a stop after each program line. The motion is also stopped after program lines that cannot be seen and after blank lines. The Start key must be pressed again for each line. The program is executed without advance processing. **Single Step** is only available to the "Expert" user group.
Backwards #BSTEP		This program run mode is automatically selected if the Start backwards key is pressed. It is not possible to select a different mode. This mode works in the same way as **Motion**, but with the following exception: CIRC motions are executed backwards in the same way as they were last executed forwards, i.e. if the forward motion was not stopped at the auxiliary point, the backward motion will not be stopped there either. This exception does not apply in the case of SCIRC motions. Here, the backward motion is always stopped at the auxiliary point.

7.3 Advance run

The advance run is the maximum number of motion blocks that the robot controller calculates and plans in advance during program execution. The actual number is dependent on the capacity of the computer. The default value is 3. The advance run refers to the current position of the block pointer. The advance run is required, for example, in order to be able to calculate approximate positioning motions.

Certain statements trigger an advance run stop. These include statements that influence the periphery, e.g. OUT statements.

7.4 Block pointer

Overview

During program execution, the block pointer indicates various items of information:

- Which motion the robot is currently executing or has completed
- Whether an auxiliary point or end point is currently being approached
- The direction in which the robot is executing the program

Pointer	Direction	Description
	Forwards	The end point is being approached.
	Backwards	
	Forwards	The end point has been reached with exact positioning.
	Backwards	
	Forwards	The auxiliary point is being approached.
	Backwards	
	Forwards	The auxiliary point has been reached with exact positioning.
	Backwards	

Examples for forward motion

```
5  PTP P3 Vel=100 % PDAT1 Tool[1] Base[0]
6  PTP P4 Vel=100 % PDAT2 Tool[1] Base[0]
7  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
```

Fig. 7-1: The robot is moving from P3 to P4

```
5  PTP P3 Vel=100 % PDAT1 Tool[1] Base[0]
6  PTP P4 Vel=100 % PDAT2 Tool[1] Base[0]
7  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
```

Fig. 7-2: The robot has reached P4 with exact positioning

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-3: The robot is moving from P5 to auxiliary point P6

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-4: The robot has reached auxiliary point P6 with exact positioning

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-5: The robot is moving from auxiliary point P6 to P7

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-6: The robot has reached P7 with exact positioning

Examples for backward motion

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-7: The robot is moving from P8 to P7

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-8: The robot has reached P7 with exact positioning

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-9: The robot is moving from P7 to auxiliary point P6

```
6  PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]
7  CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]
8  PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Fig. 7-10: The robot has reached auxiliary point P6 with exact positioning

Double upward/downward arrow

If the program window shows a section in which the block pointer is not currently located, a double arrow indicates the direction in which it is to be found.

```
7  PTP P6 Vel=100 % PDAT4 Tool[1] Base[0]
8  PTP P7 Vel=100 % PDAT5 Tool[1] Base[0]
```

Fig. 7-11: The block pointer is located higher up in the program

```
14  PTP P13 Vel=100 % PDAT11 Tool[1] Base[0]
15  PTP P14 Vel=100 % PDAT12 Tool[1] Base[0]
```

Fig. 7-12: The block pointer is located lower down in the program

7.5 Setting the program override (POV)

Description

Program override is the velocity of the robot during program execution. The program override is specified as a percentage of the programmed velocity.

In T1 mode, the maximum velocity is 250 mm/s, irrespective of the value that is set.

Procedure

1. Touch the **POV/HOV** status indicator. The **Overrides** window is opened.
2. Set the desired program override. It can be set using either the plus/minus keys or by means of the slider.
   - Plus/minus keys: The value can be set to 100 %, 75 %, 50 %, 30 %, 10 %, 3 %, 1 %
   - Slider: The override can be adjusted in 1 % steps.
3. Touch the **POV/HOV** status indicator again. (Or touch the area outside the window.)

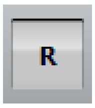
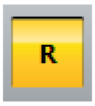
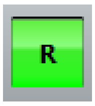
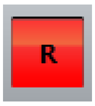
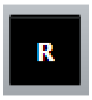
   The window closes and the selected override value is applied.

The **Jog options** window can be opened via **Options** in the **Overrides** window.

Alternative procedure

Alternatively, the override can be set using the plus/minus key on the lower right-hand side of the smartPAD.

7.6 Robot interpreter status indicator

Icon	Color	Description
	Gray	No program is selected.
	Yellow	The block pointer is situated on the first line of the selected program.
	Green	The program is selected and is being executed.
	Red	The selected and started program has been stopped.
	Black	The block pointer is situated at the end of the selected program.

7.7 Starting a program forwards (manual)

Precondition

- A program is selected.
- Operating mode T1 or T2

Procedure

1. Select the program run mode.
2. Hold the enabling switch down and wait until the status bar indicates “Drives ready”:

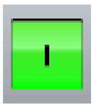


Fig. 7-13

3. Carry out a BCO run: Press Start key and hold it down until the message “*Programmed path reached (BCO)*” is displayed in the message window. The robot stops.

CAUTION The BCO run is executed as a LIN or PTP motion from the actual position to the target position. The velocity is automatically reduced. The path of the motion cannot be predicted reliably. Observe the motion during the BCO run so that the robot can be stopped in time if a collision becomes imminent.

4. Press Start key and hold it down.

   The program is executed with or without stops, depending on the program run mode.

To stop a program that has been started manually, release the Start key.

7.8 Starting a program forwards (automatic)

Precondition

- A program is selected.
- Operating mode Automatic (not Automatic External)

Procedure

1. Select the program run mode **Go**.
2. Switch on the drives.
3. Carry out a BCO run:

   Press Start key and hold it down until the message “Programmed path reached (BCO)” is displayed in the message window. The robot stops.

> **⚠ CAUTION** The BCO run is executed as a LIN or PTP motion from the actual position to the target position. The velocity is automatically reduced. The path of the motion cannot be predicted reliably. Observe the motion during the BCO run so that the robot can be stopped in time if a collision becomes imminent.

4. Press the Start key. The program is executed.

To stop a program that has been started in Automatic mode, press the STOP key.

7.9 Carrying out a block selection

Description

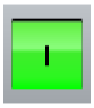
A program can be started at any point by means of a block selection.

Precondition

- A program is selected.
- Operating mode T1 or T2

Procedure

1. Select the program run mode.
2. Select the motion block at which the program is to be started.
3. Press **Block selection**. The block pointer indicates the motion block.
4. Hold the enabling switch down and wait until the status bar indicates “Drives ready”:

5. Carry out a BCO run: Press the Start key and hold it down until the message “*Programmed path reached (BCO)*” is displayed in the message window. The robot stops.

> **⚠ CAUTION** The BCO run is executed as a LIN or PTP motion from the actual position to the target position. The velocity is automatically reduced. The path of the motion cannot be predicted reliably. Observe the motion during the BCO run so that the robot can be stopped in time if a collision becomes imminent.

6. The program can now be started manually or automatically. It is not necessary to carry out a BCO run again.

7.10 Resetting a program

Description

In order to restart an interrupted program from the beginning, it must be reset. This returns the program to the initial state.

Precondition

- Program is selected.

Procedure

- Select the menu sequence **Edit** > **Reset program**.

Alternative procedure

- In the status bar, touch the **Robot interpreter** status indicator. A window opens.
  Select **Reset program**.

7.11 Starting Automatic External mode

> **NOTICE** There is no BCO run in Automatic External mode. This means that the robot moves to the first programmed position after the start at the programmed (not reduced) velocity and does not stop there.

Precondition

- Operating mode T1 or T2
- Inputs/outputs for Automatic External are configured.
- The program CELL.SRC is configured.

Procedure

1. Select the program CELL.SRC in the Navigator. (This program is located in the folder “R1”.)
2. Set program override to 100%. (This is the recommended setting. A different value can be set if required.)
3. Carry out a BCO run:
   Hold down the enabling switch. Then press the Start key and hold it down until the message “Programmed path reached (BCO)” is displayed in the message window.

> **⚠ CAUTION** The BCO run is executed as a LIN or PTP motion from the actual position to the target position. The velocity is automatically reduced. The path of the motion cannot be predicted reliably. Observe the motion during the BCO run so that the robot can be stopped in time if a collision becomes imminent.

4. Select “Automatic External” mode.
5. Start the program from a higher-level controller (PLC).

To stop a program that has been started in Automatic mode, press the STOP key.

7.12 Backward motion using the Start backwards key

7.12.1 Executing motions backwards (using the “Start backwards” key)

Description

Backward motion via the “Start backwards” key is often used if a sequence of motions is to be optimized and individual points are to be re-taught for this purpose. The user executes the path backwards until the point that is to be corrected has been reached. Once the point has been re-taught, backward motion is continued if required in order to correct further points.

The program run mode #BSTEP is automatically applied for backward motion.

Approximate positioning and weaving are not possible during backward motion. If approximate positioning or weaving were carried out for points during forward execution, the backward path will thus differ from the forward path. It is thus possible that the robot may have to perform a BCO run after starting backward motion, even though it did not leave the path during forward motion.

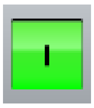
> **⚠ CAUTION** The BCO run is executed as a LIN or PTP motion from the actual position to the target position. The velocity is automatically reduced. The path of the motion cannot be predicted reliably. Observe the motion during the BCO run so that the robot can be stopped in time if a collision becomes imminent.

Precondition

- A program is selected.
- The motions that are to be executed backwards have been executed forwards.
- T1 or T2 mode

Procedure

1. Hold the enabling switch down and wait until the status bar indicates “Drives ready”:

2. Press and hold down the Start backwards key.
   - If the robot is already on the backward path, it now moves backwards.
   - If the robot is not on the backward path, it now moves to it. When “*Programmed path reached (BCO)*” is displayed in the message window, it has reached the path. The robot stops.

     Press the Start backwards key again. The robot now moves backwards.
3. Press the Start backwards key again for each motion block.

7.12.2 Functional principle and characteristics of backward motion

Functional principle

During forward motion, the robot controller saves the executed motions in a ring buffer. During backward motion, the motions are executed on the basis of the saved information.

No backward motion possible once the buffer has been deleted:

The contents of the buffer are deleted in the following cases. Backward motion is not possible again until motions have been executed in the forward direction again.

- Program is reset.
- Program is deselected.
- Lines are inserted into the program or deleted.
- KRL instruction RESUME
- Block selection to a motion other than the current one.

  What is possible without restriction, however, is a block selection to any segment point within the current spline block. This counts as block selection to the current motion, as the robot controller plans and executes the spline block as one motion.

The robot controller deletes the buffer without generating a corresponding message.

Properties

- Backward motion is only possible in modes T1 and T2.
- Only motions are executed during backward motion, and no control structures or control instructions.
- Outputs and flags are not recorded during forward motion. For this reason, their previous states are not restored during backward motion.

- The velocity is the same as for forward motion.

  In T2, it is possible that monitoring functions may be triggered during backward motion that are not triggered during forward motion. In this case, the program override must be reduced.
- Backward motion is active by default. It can be deactivated or prevented for specific motions in the user group “Expert”.

Further information is contained in the Operating and Programming Instructions for System Integrators.

Torque/force mode, VectorMove

The following applies to motions with torque or force mode or VectorMove:

- Backward motion is possible for conventional motions, but force/torque mode or VectorMove is automatically deactivated.
- Spline motions cannot be executed backwards.

7.12.2.1 Response in the case of subprograms

- Motions executed forwards in an interrupt program are not recorded. They cannot, therefore, be executed backwards.
- If a subprogram has been completely executed during forward motion, it cannot be executed with backward motion.
- If the forward motion was stopped in a subprogram, the response depends on the position of the advance run pointer:

Position of the advance run pointer	Response
Advance run pointer is in the subprogram.	Backward motion is possible.
Advance run pointer has already left the subprogram.	Backward motion is not possible. Prevention: Trigger an advance run stop before the END of the subprogram, e.g. with WAIT SEC 0. However, it is then no longer possible to carry out approximate positioning at this point. Or set $ADVANCE to "1". This does not always prevent the error message, but it reduces the probability. Approximate positioning is still possible.

7.12.2.2 Approximate positioning response

Description

Approximate positioning is not possible during backward motion. If approximate positioning was carried out for points during forward execution, the backward path will thus differ from the forward path. It is thus possible that the robot may have to perform a BCO run for the backward path after starting backward motion, even though it did not leave the path during forward motion.

Example 1

Backward start outside an approximate positioning range:

The Start backwards key is pressed while the robot is on the path, but not in an approximate positioning range. The robot now moves backwards on the path to the end point of the previous motion.

P_{BACK} = position of the robot at the moment at which the Start backwards key is pressed

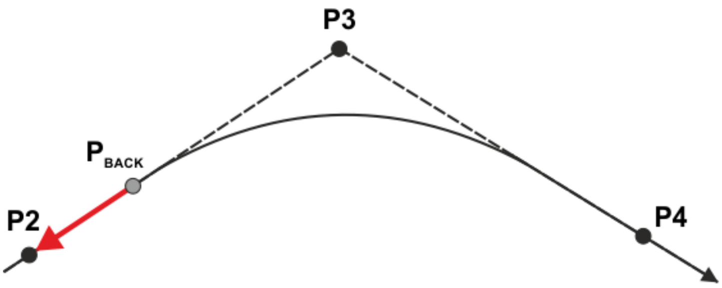


Fig. 7-14: Case 1: Backward start outside an approximate positioning range

If the end point of the previous motion is approximated, it is nonetheless addressed with exact positioning.

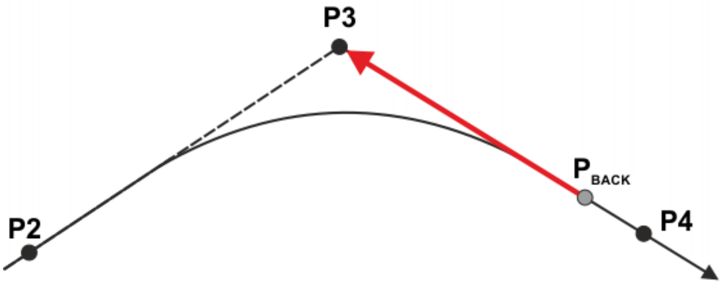


Fig. 7-15: Case 2: Backward start outside an approximate positioning range

Example 2

Backward start in the approximate positioning range:

The Start backwards key is pressed while the robot is in an approximate positioning range. The robot now performs a BCO run to the start of the approximate positioning range and stops there. If the Start backwards key is now pressed again, the actual backward motion begins, i.e. the robot moves backwards along the path to the end point of the previous motion.

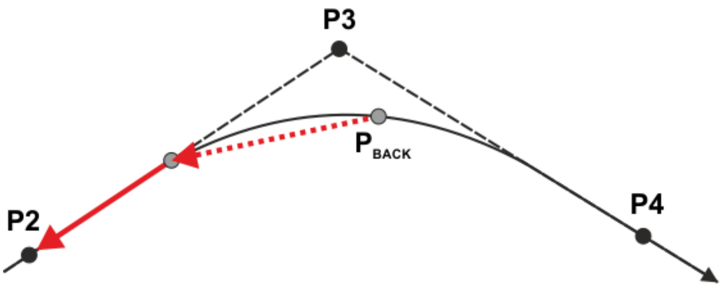


Fig. 7-16: Backward start in the approximate positioning range

7.12.2.3 Response in the case of weave motions

Description

Weaving is not possible during backward motion. If weaving was carried out during forward execution, the backward path will thus differ from the forward path. The robot must therefore perform a BCO run for the backward path after starting backward motion, even though it did not leave the path during forward motion.

Example

Backward start on weave path:

The Start backwards key is pressed while the robot is weaving. The robot now performs a BCO run to the taught path and stops there. If the Start backwards key is now pressed again, the actual backward motion begins, i.e. the robot moves backwards along the path to the end point of the previous motion.

P_{BACK} = position of the robot at the moment at which the Start backwards key is pressed

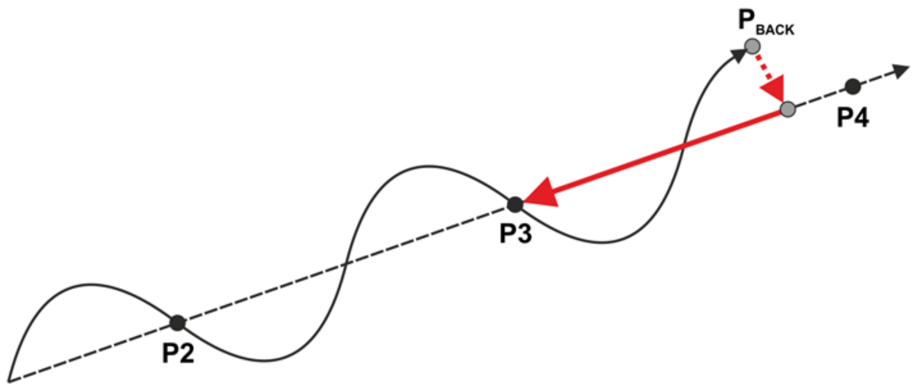


Fig. 7-17: Backward start on weave path

7.12.2.4 Switching from backwards to forwards

Precondition

It is only possible to resume forward motion following backward motion if the following preconditions are met:

- Block selection is possible to the program line on which the backward block pointer is currently located.
- If the first motion to be executed forwards again is a conventional motion, it must be completely programmed.

  It is thus not possible, for example, to switch from backward motion to forward motion if the first motion is a PTP_REL motion.

  With few exceptions, this restriction does not apply in the case of spline motions.

Response

When the Start forward key is pressed for the first time following backward motion, the response is as follows:

- If BCO exists, the program run mode most recently used in the forward direction is automatically restored and the robot moves forwards on the path.
- If BCO does not exist, a BCO run is carried out. The program run mode meanwhile remains set to #BSTEP. After the BCO run, the robot stops. The Start forwards key must now be pressed again. The program run mode most recently used in the forward direction is automatically restored and the robot now moves forwards on the path.

If the switch from backwards to forwards occurs in a control structure, the robot first moves forwards to the end of the control structure. It then stops with the message *Control structure next block {Block number}*. The block number specifies the first block after the control structure.

8 Basic principles of motion programming

8.1 Overview of motion types

The following motion types can be programmed:

- **Point-to-point motion (PTP)**
  (>>> 8.2 "Motion type PTP" Page 175)
- **Linear motions (LIN)**
  (>>> 8.3 "Motion type LIN" Page 176)
- **Circular motion (CIRC)**
  (>>> 8.4 "Motion type CIRC" Page 176)
- **Spline motions**
  Spline motions have a number of advantages over conventional PTP, LIN and CIRC motions.
  (>>> 8.7 "Spline motion type" Page 179)

The start point of a motion is always the end point of the previous motion.

The following motions are known as CP ("Continuous Path") motions.

- LIN, CIRC, CP spline blocks, SLIN, SCIRC

8.2 Motion type PTP

The robot guides the TCP along the fastest path to the end point. The fastest path is generally not the shortest path and is thus not a straight line. As the motions of the robot axes are rotational, curved paths can be executed faster than straight paths.

The exact path of the motion cannot be predicted.

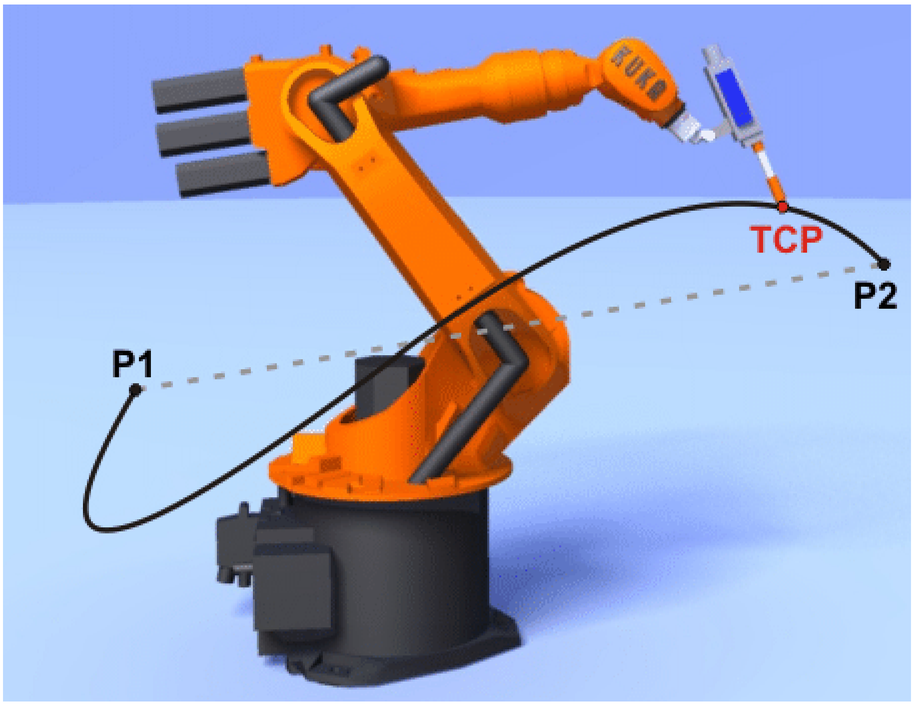


Fig. 8-1: PTP motion

8.3 Motion type LIN

The robot guides the TCP at a defined velocity along a straight path to the end point.

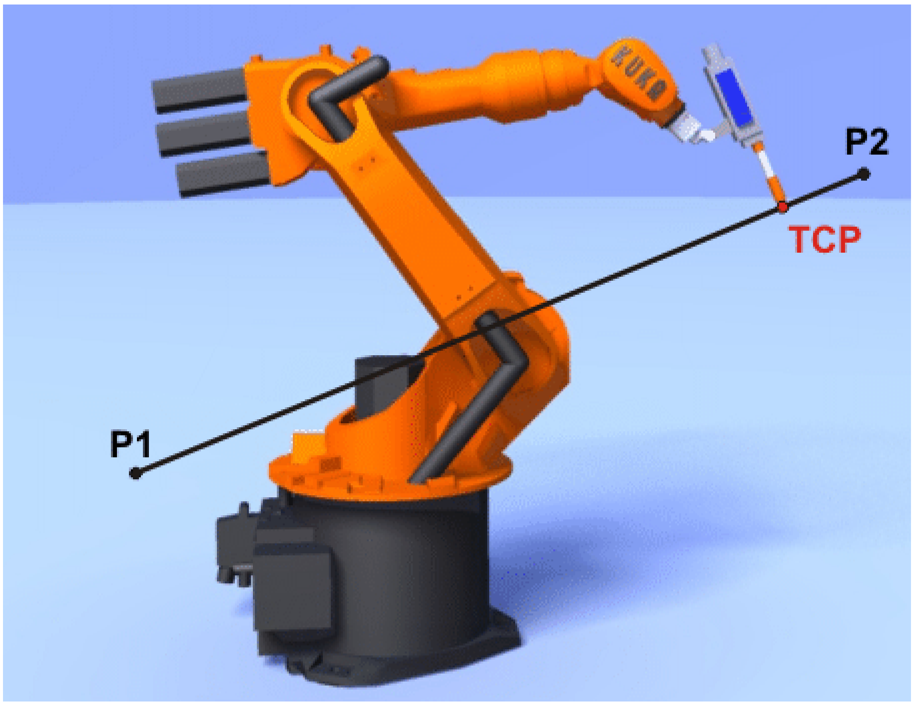


Fig. 8-2: LIN motion

8.4 Motion type CIRC

The robot guides the TCP at a defined velocity along a circular path to the end point. The circular path is defined by a start point, auxiliary point and end point.

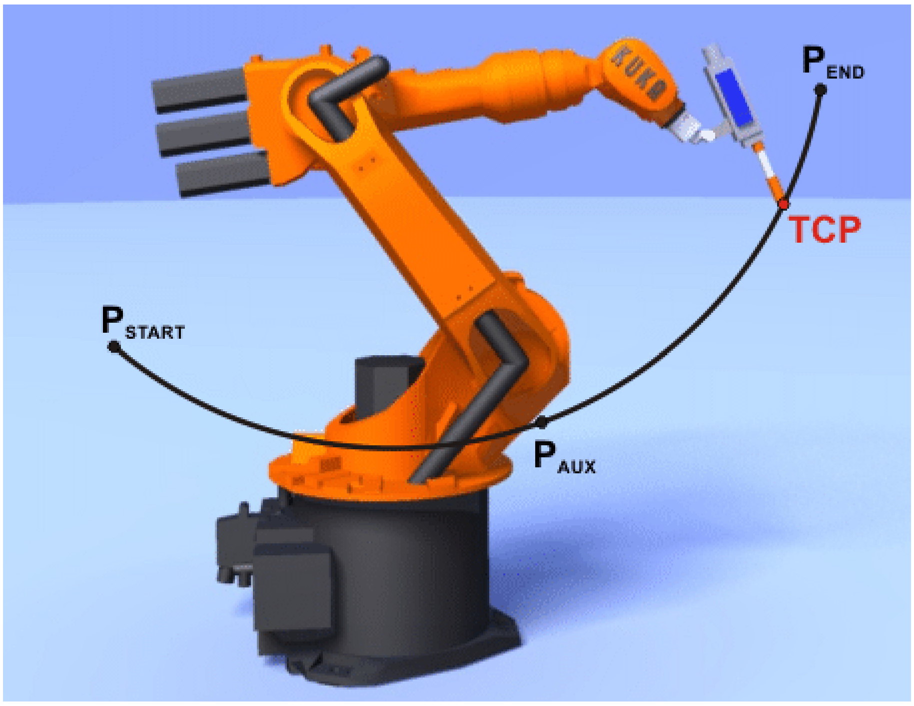


Fig. 8-3: CIRC motion

8.5 Approximate positioning

Approximate positioning means that the motion does not stop exactly at the programmed point. Approximate positioning is an option that can be selected during motion programming.

Approximate positioning is not possible if the motion instruction is followed by an instruction that triggers an advance run stop.

Approximate positioning with a PTP motion

The TCP leaves the path that would lead directly to the end point and moves along a faster path. During programming of the motion, the maximum distance from the end point at which the TCP may deviate from its original path is defined.

The path of an approximated PTP motion cannot be predicted. It is also not possible to predict on which side of the approximated point the path will run.

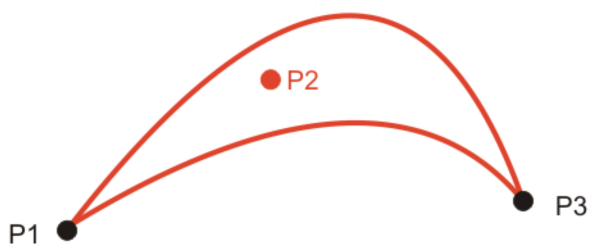


Fig. 8-4: PTP motion, P2 is approximated

Approximate positioning with a LIN motion

The TCP leaves the path that would lead directly to the end point and moves along a shorter path. During programming of the motion, the maximum distance from the end point at which the TCP may deviate from its original path is defined.

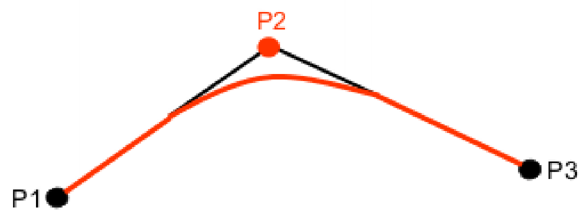


Fig. 8-5: LIN motion, P2 is approximated

Approximate positioning with a CIRC motion

The TCP leaves the path that would lead directly to the end point and moves along a shorter path. During programming of the motion, the maximum distance from the end point at which the TCP may deviate from its original path is defined.

The motion passes exactly through the auxiliary point.

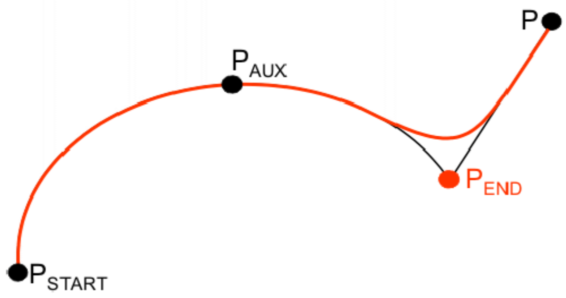


Fig. 8-6: CIRC motion, P_{END} is approximated

8.6 Orientation control LIN, CIRC

Description

The orientation of the TCP can be different at the start point and end point of a motion. There are several different types of transition from the start orientation to the end orientation. A type must be selected when a CP motion is programmed.

The orientation control for LIN and CIRC motions is defined as follows:

- In the option window **Motion parameter**

LIN motion

Orientation control	Description
Constant orientation	The orientation of the TCP remains constant during the motion. The programmed orientation is disregarded for the end point and that of the start point is retained.
Standard	The orientation of the TCP changes continuously during the motion. **Note:** If, with **Standard**, the robot passes through a wrist axis singularity, use **Wrist PTP** instead.
Wrist PTP	The orientation of the TCP changes continuously during the motion. This is done by linear transformation (axis-specific motion) of the wrist axis angles. **Note:** Use **Wrist PTP** if, with **Standard**, the robot passes through a wrist axis singularity. The orientation of the TCP changes continuously during the motion, but not uniformly. **Wrist PTP** is thus not suitable if a specific orientation must be maintained exactly, e.g. in the case of laser welding.

If a wrist axis singularity occurs with **Standard** and the desired orientation cannot be maintained exactly enough with **Wrist PTP**, the following remedy is recommended:
Re-teach start and/or end point. Select orientations that prevent a wrist axis singularity from occurring and allow the path to be executed with **Standard**.

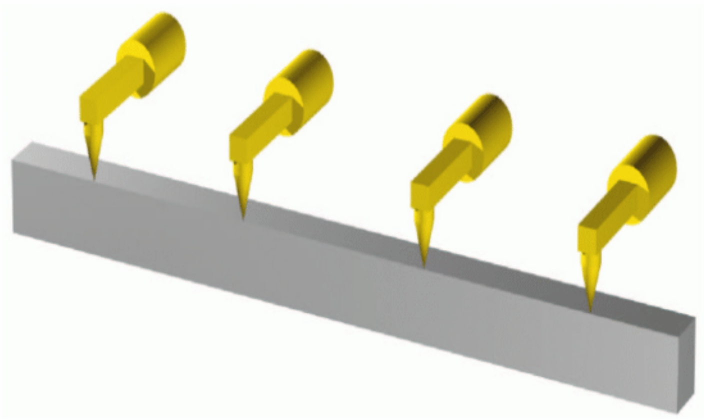


Fig. 8-7: Constant orientation

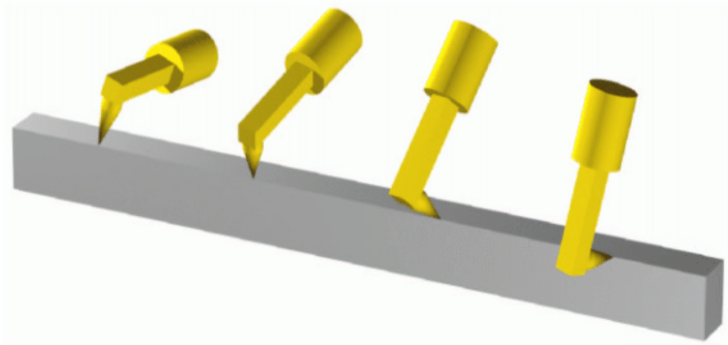


Fig. 8-8: Standard or Wrist PTP

CIRC motion

The same orientation control options are available for selection for CIRC motions as for LIN motions.

During CIRC motions, the robot controller only takes the programmed orientation of the end point into consideration. The programmed orientation of the auxiliary point is disregarded.

8.7 Spline motion type

Spline is a motion type that is particularly suitable for complex, curved paths. Such paths can also be generated using approximated LIN and CIRC motions, but splines have advantages, however.

The most versatile spline motion is the spline block. A spline block is used to group together several motions as an overall motion. The spline block is planned and executed by the robot controller as a single motion block.

The motions that may be included in a spline block are called spline segments. They are taught separately.

- A CP spline block can contain SPL, SLIN and SCIRC segments.
- A PTP spline block can contain SPTP segments.

In addition to spline blocks, individual spline motions can also be programmed: SLIN, SCIRC and SPTP.

Advantages of spline blocks

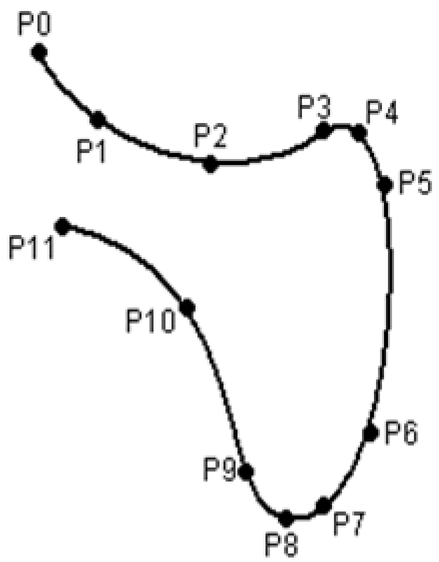


Fig. 8-9: Curved path with spline block

- The path is defined by means of points that are located on the path. The desired path can be generated easily.
- The programmed velocity is maintained better than with conventional motion types. There are few cases in which the velocity is reduced.

  (>>> 8.7.1 "Velocity profile for spline motions" Page 181)

  Furthermore, special constant velocity ranges can be defined in CP spline blocks.
- The path always remains the same, irrespective of the override setting, velocity or acceleration.
- Circles and tight radii are executed with great precision.

Disadvantages of LIN/CIRC

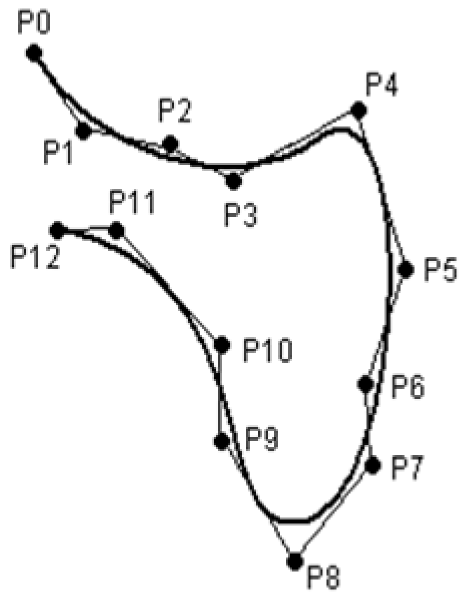


Fig. 8-10: Curved path with approximated LIN motions

- The path is defined by means of approximated points that are not located on the path. The approximate positioning ranges are difficult to predict. Generating the desired path is complicated and time-consuming.

- In many cases, the velocity may be reduced in a manner that is difficult to predict, e.g. in the approximate positioning ranges and near points that are situated close together.
- The path changes if approximate positioning is not possible, e.g. for time reasons.
- The path changes in accordance with the override setting, velocity or acceleration.

8.7.1 Velocity profile for spline motions

The path always remains the same, irrespective of the override setting, velocity or acceleration.

The robot controller already takes the physical limits of the robot into consideration during planning. The robot moves as fast as possible within the constraints of the programmed velocity, i.e. as fast as its physical limits will allow. This is an advantage over conventional LIN and CIRC motions for which the physical limits are not taken into consideration during planning. It is only during motion execution that these limits have any effect and can cause stops to be triggered.

Reduction of the velocity

Prime examples of cases in which the velocity has to fall below the programmed velocity include:

- Tight corners
- Major reorientation
- Large motions of the external axes
- Motion in the vicinity of singularities

Reduction of the velocity due to major reorientation can be avoided with spline segments by selecting the orientation control option **Ignore orientation**.

Reduction of the velocity to 0

This is the case for:

- Successive points with the same coordinates.
- Successive SLIN and/or SCIRC segments. Cause: inconstant velocity direction.

  In the case of SLIN-SCIRC transitions, the velocity is also reduced to 0 if the straight line is a tangent of the circle, as the circle, unlike the straight line, is curved.

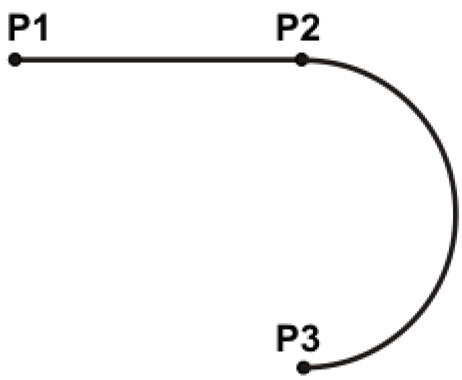


Fig. 8-11: Exact positioning at P2

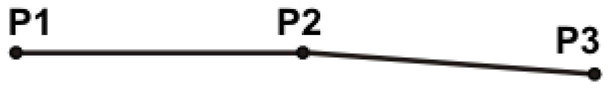


Fig. 8-12: Exact positioning at P2

Exceptions:

- In the case of successive SLIN segments that result in a straight line and in which the orientations change uniformly, the velocity is not reduced.

Fig. 8-13: P2 is executed without exact positioning.

- In the case of a SCIRC-SCIRC transition, the velocity is not reduced if both circles have the same center point and the same radius and if the orientations change uniformly. (This is difficult to teach, so calculate and program points.)

> Circles with the same center point and the same radius are sometimes programmed to obtain circles ≥ 360°. A simpler method is to program a circular angle.

8.7.2 Block selection with spline motions

Spline block

Block selection can be made to the segments of a spline block.

- CP spline block:

  The BCO run is executed as a conventional LIN motion. This is indicated by means of a message that must be acknowledged.
- PTP spline block:

  The BCO run is executed as a conventional PTP motion. This is not indicated by a message.

Following a block selection, the path is generally the same as if the spline were to be executed during normal program execution.

Exceptions are possible if the spline has never yet been executed prior to the block selection and the block selection is made here to the start of the spline block:

The start point of the spline motion is the last point before the spline block, i.e. the start point is outside the block. The robot controller saves the start point during normal execution of a spline. In this way, it is already known in the case of a block selection being carried out subsequently. If the spline block has never yet been executed, however, the start point is unknown.

If the Start key is pressed after the BCO run, the modified path is indicated by means of a message that must be acknowledged.

Example: modified path in the case of block selection to P1

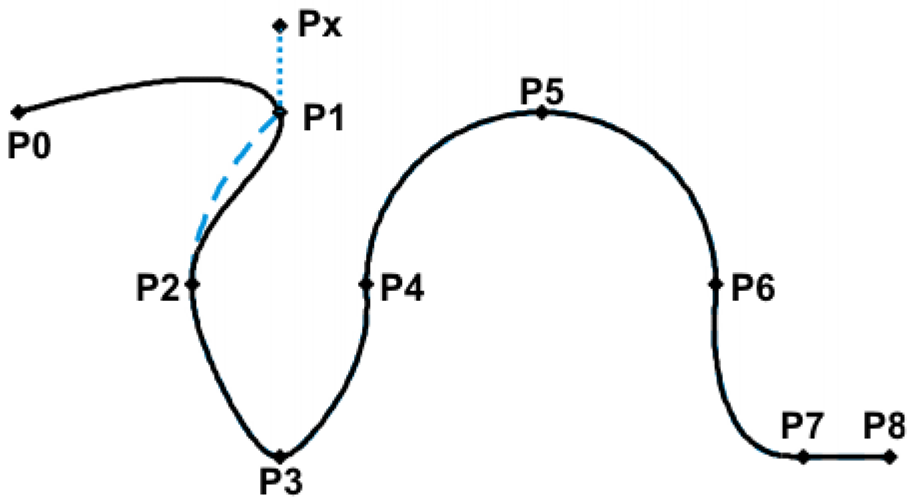


Fig. 8-14: Example: modified path in the case of block selection to P1 with unknown P0

```
1  PTP P0
2  SPLINE
3   SPL P1
4   SPL P2
5   SPL P3
6   SPL P4
7   SCIRC P5, P6
8   SPL P7
9   SLIN P8
10 ENDSPLINE
```

Line	Description
2	Header/start of the CP spline block
3 … 9	Spline segments
10	End of the CP spline block

SCIRC

In the case of block selection to a SCIRC segment for which a circular angle has been programmed, the motion is executed to the end point, taking into consideration the circular angle, provided that the robot controller knows the start point.

If the robot controller does not know the start point, the motion is executed to the programmed end point. In this case, a message is generated, indicating that the circular angle is not being taken into consideration.

In the case of a block selection to an individual SCIRC motion, the circular angle is never taken into consideration.

8.7.3 Modifications to spline blocks

Description

- Modification of the position of the point:

  If a point within a spline block is offset, the path is modified, at most, in the 2 segments before this point and the 2 segments after it.

  Small point offsets generally result in small modifications to the path. If, however, very long segments are followed by very short segments or vice versa, small modifications can have a very great effect.

- Modification of the segment type:
  If an SPL segment is changed into an SLIN segment or vice versa, the path changes in the previous segment and the next segment.

Example 1

Original path:

```
PTP P0
SPLINE
 SPL P1
 SPL P2
 SPL P3
 SPL P4
 SCIRC P5, P6
 SPL P7
 SLIN P8
ENDSPLINE
```

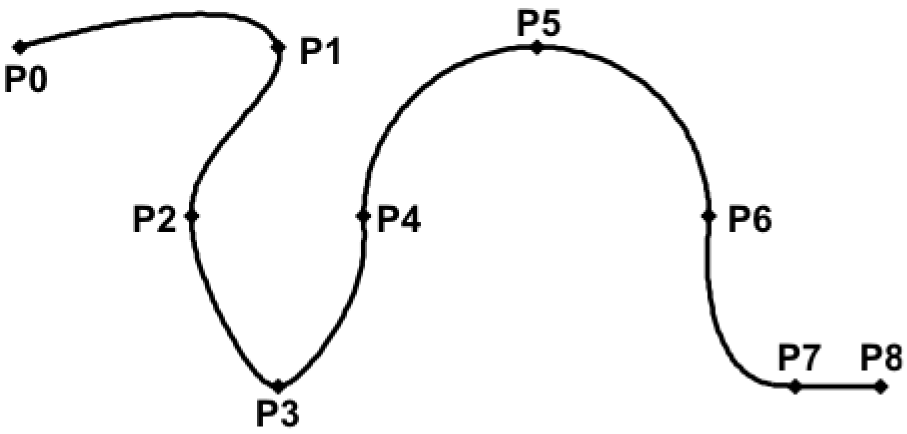


Fig. 8-15: Original path

A point is offset relative to the original path:

P3 is offset. This causes the path to change in segments P1 - P2, P2 - P3 and P3 - P4. Segment P4 - P5 is not changed in this case, as it belongs to an SCIRC and a circular path is thus defined.

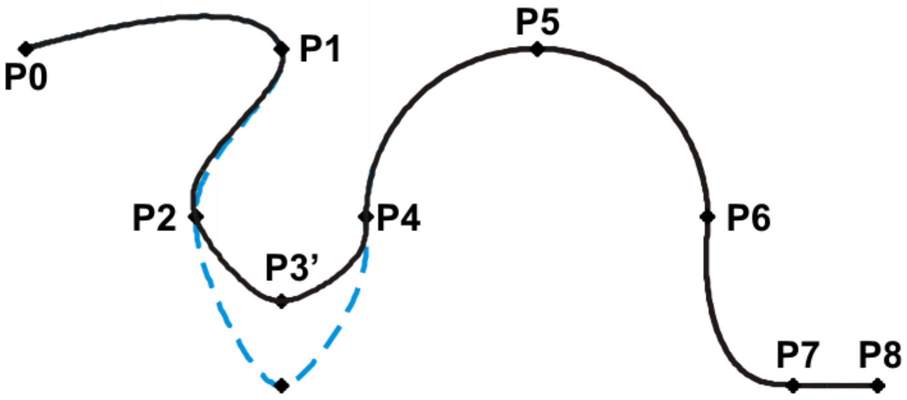


Fig. 8-16: Point has been offset

The type of a segment is changed relative to the original path:

In the original path, the segment type of P2 - P3 is changed from SPL to SLIN. The path changes in segments P1 - P2, P2 - P3 and P3 - P4.

```
PTP P0
SPLINE
 SPL P1
 SPL P2
 SLIN P3
 SPL P4
 SCIRC P5, P6
 SPL P7
 SLIN P8
ENDSPLINE
```

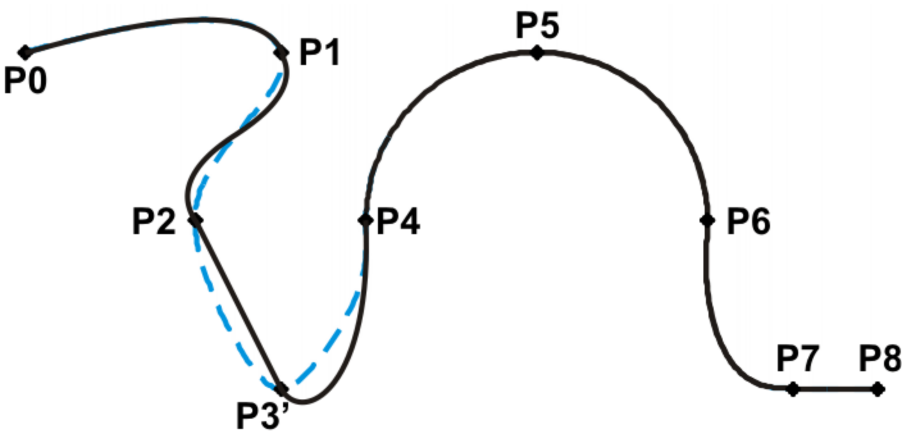


Fig. 8-17: Segment type has been changed

Example 2

Original path:

```
...
SPLINE
 SPL {X 100, Y 0, ...}
 SPL {X 102, Y 0}
 SPL {X 104, Y 0}
 SPL {X 204, Y 0}
ENDSPLINE
```

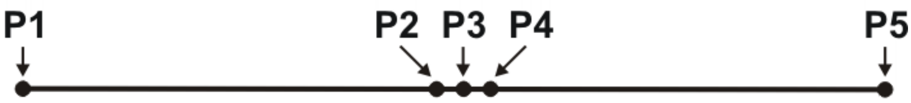


Fig. 8-18: Original path

A point is offset relative to the original path:

P3 is offset. This causes the path to change in all the segments illustrated. Since P2 - P3 and P3 - P4 are very short segments and P1 - P2 and P4 - P5 are long segments, the slight offset causes the path to change greatly.

```
...
SPLINE
 SPL {X 100, Y 0, ...}
 SPL {X 102, Y 1}
 SPL {X 104, Y 0}
 SPL {X 204, Y 0}
ENDSPLINE
```

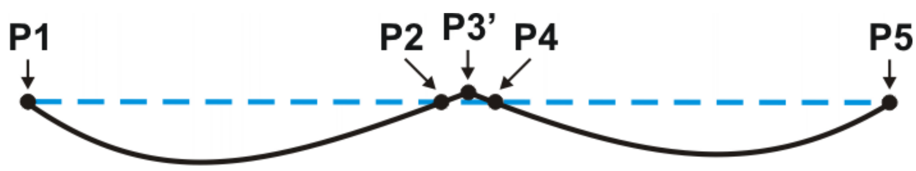


Fig. 8-19: Point has been offset

Remedy:

- Distribute the points more evenly
- Program straight lines (except very short ones) as SLIN segments

8.7.4 Approximation of spline motions

All spline blocks and all individual spline motions can be approximated with one another. It makes no difference whether they are CP or PTP spline blocks, nor is the motion type of the individual motion relevant.

The motion type of the approximate positioning arc always corresponds to the second motion. In the case of SPTP-SLIN approximation, for example, the approximate positioning arc is of type CP.

Spline motions cannot be approximated with conventional motions (LIN, CIRC, PTP).

Approximation not possible due to time or advance run stops:

If approximation is not possible for reasons of time or due to an advance run stop, the robot waits at the start of the approximate positioning arc.

- In the case of time reasons: the robot moves again as soon as it has been possible to plan the next block.
- In the case of an advance run stop: the end of the current block is reached at the start of the approximate positioning arc. This means that the advance run stop is canceled and the robot controller can plan the next block. Robot motion is resumed.

In both cases, the robot now moves along the approximate positioning arc. Approximate positioning is thus technically possible; it is merely delayed.

This response differs from that for LIN, CIRC or PTP motions. If approximate positioning is not possible for the reasons specified, the motion is executed to the end point with exact positioning.

No approximate positioning in MSTEP and ISTEP:

In the program run modes MSTEP and ISTEP, the robot stops exactly at the end point, even in the case of approximated motions.

In the case of approximate positioning from one spline block to another spline block, the result of this exact positioning is that the path is different in the last segment of the first block and in the first segment of the second block from the path in program run mode GO.

In all other segments of both spline blocks, the path is identical in MSTEP, ISTEP and GO.

8.7.5 Replacing an approximated CP motion with a spline block

Description

In order to replace approximated conventional CP motions with spline blocks, the program must be modified as follows:

- Replace LIN - LIN with SLIN - SPL - SLIN.

- Replace LIN - CIRC with SLIN - SPL - SCIRC.

  Recommendation: Allow the SPL to project a certain way into the original circle. The SCIRC thus starts later than the original CIRC.

In approximated motions, the corner point is programmed. In the spline block, the points at the start and end of the approximation are programmed instead.

The following approximated motion is to be reproduced:

```
LIN P1 C_DIS
LIN P2
```

Spline motion:

```
SPLINE
 SLIN P1A
 SPL P1B
 SLIN P2
ENDSPLINE
```

P1A = start of approximation, P1B = end of approximation

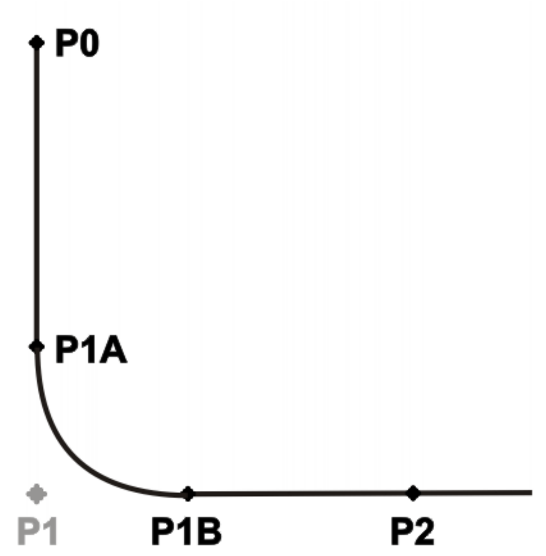


Fig. 8-20: Approximated motion - spline motion

Ways of determining P1A and P1B:

- Execute the approximated path and save the positions at the desired point by means of Trigger.
- Calculate the points in the program with KRL.
- The start of the approximation can be determined from the approximate positioning criterion. Example: If C_DIS is specified as the approximate positioning criterion, the distance from the start of the approximation to the corner point corresponds to the value of $APO.CDIS.

  The end of the approximation is dependent on the programmed velocity.

The SPL path does not correspond exactly to the approximate positioning arc, even if P1A and P1B are exactly at the start/end of the approximation. In order to recreate the exact approximate positioning arc, additional points must be inserted into the spline. Generally, one point is sufficient.

Example

The following approximated motion is to be reproduced:

```
$APO.CDIS=20
$VEL.CP=0.5
```

```
LIN {Z 10} C_DIS
LIN {Y 60}
```

Spline motion:

```
SPLINE WITH $VEL.CP=0.5
 SLIN {Z 30}
 SPL {Y 30, Z 10}
 SLIN {Y 60}
ENDSPLINE
```

The start of the approximate positioning arc has been calculated from the approximate positioning criterion.

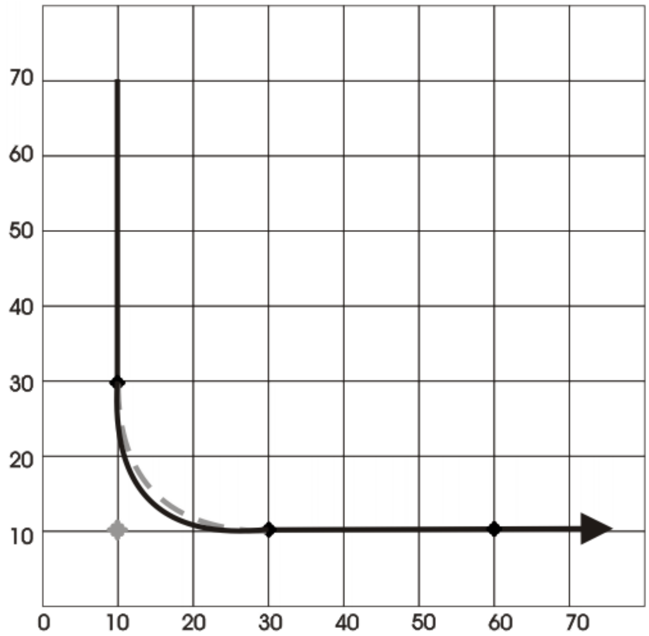


Fig. 8-21: Example: Approximated motion - spline motion 1

The SPL path does not yet correspond exactly to the approximate positioning arc. For this reason, an additional SPL segment is inserted into the spline.

```
SPLINE WITH $VEL.CP=0.5
 SLIN {Z 30}
 SPL {Y 15, Z 15}
 SPL {Y 30, Z 10}
 SLIN {Y 60}
ENDSPLINE
```

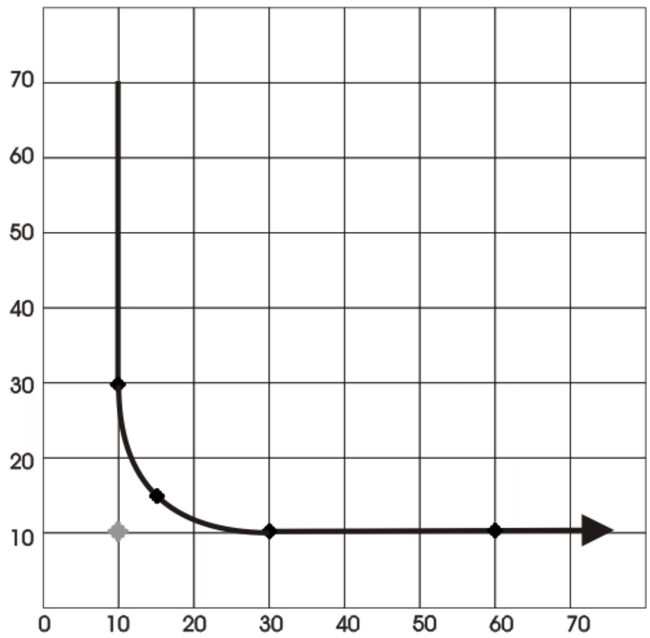


Fig. 8-22: Example: Approximated motion - spline motion 2

With the additional point, the path now corresponds to the approximate positioning arc.

8.7.5.1 SLIN-SPL-SLIN transition

In the case of a SLIN-SPL-SLIN segment sequence, it is usually desirable for the SPL segment to be located within the smaller angle between the two straight lines. Depending on the start and end point of the SPL segment, the path may also move outside this area.

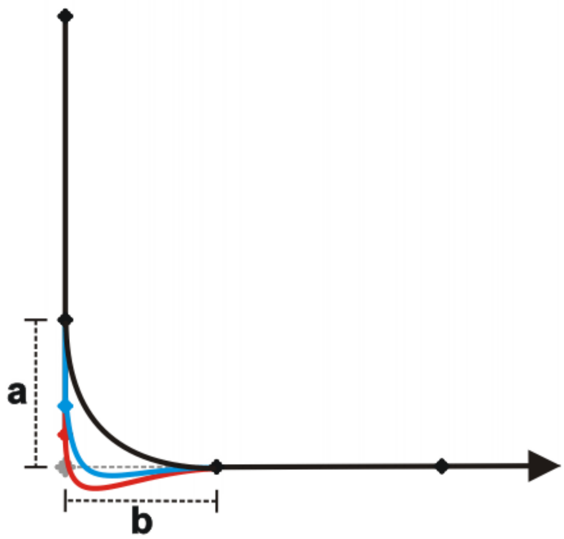


Fig. 8-23: SLIN-SPL-SLIN

The path remains inside if the following conditions are met:

- The extensions of the two SLIN segments intersect.

- 2/3 ≤ a/b ≤ 3/2

  a = distance from start point of the SPL segment to intersection of the SLIN segments

  b = distance from intersection of the SLIN segments to end point of the SPL segment

8.8 Orientation control for CP spline motions

Description

The orientation of the TCP can be different at the start point and end point of a motion. When a CP spline motion is programmed, it is necessary to select how to deal with the different orientations.

The orientation control type is defined in the option window **Motion parameters**.

Orientation control	Description
Constant orientation	The orientation of the TCP remains constant during the motion. The orientation of the start point is retained. The programmed orientation of the end point is not taken into consideration.
Default	The orientation of the TCP changes continuously during the motion. At the end point, the TCP has the programmed orientation.
Wrist PTP	The orientation of the TCP changes continuously during the motion. This is done by linear transformation (axis-specific motion) of the wrist axis angles. **Note**: Use **Wrist PTP** if, with **Default**, the robot passes through a wrist axis singularity. The orientation of the TCP changes continuously during the motion, but not uniformly. **Wrist PTP** is thus not suitable if a specific orientation must be maintained exactly, e.g. in the case of laser welding.
Ignore orientation	This option is only available for CP spline segments (not for the spline block or for individual spline motions). This option is used if no specific orientation is required at a specific point. (>>> "Ignore orientation" Page 191)

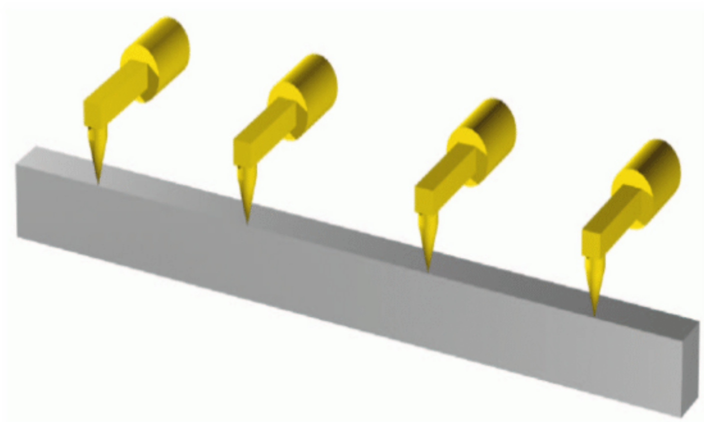


Fig. 8-24: Constant orientation

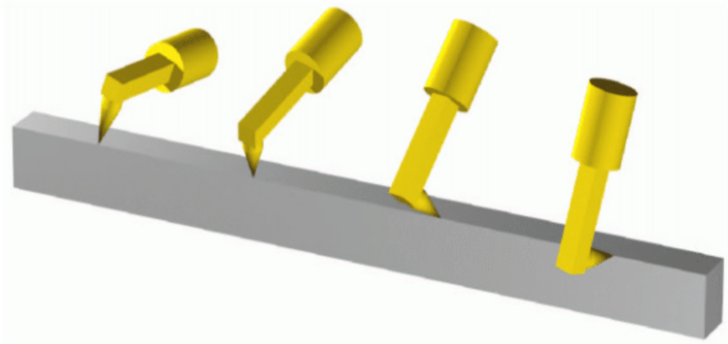


Fig. 8-25: Default orientation control

Ignore orientation

Ignore orientation is used if no specific orientation is required at a specific point. If this option is selected, the robot controller ignores the taught or programmed orientation of the point. Instead, it calculates the optimal orientation for this point on the basis of the orientations of the surrounding points. This reduces the cycle time.

Properties of **Ignore orientation**:

- In the program run modes MSTEP and ISTEP, the robot stops with the orientations calculated by the robot controller.
- In the case of a block selection to a point with **Ignore orientation**, the robot adopts the orientation calculated by the robot controller.

Ignore orientation is not allowed for the following segments:

- The last segment in a spline block
- SCIRC segments with the circle orientation control type **path-oriented**
- Segments followed by a SCIRC segment with **path-oriented**
- Segments followed by a segment with **Constant orientation**

SCIRC

The same orientation control options are available for selection for SCIRC motions as for SLIN motions. It is also possible to define for SCIRC motions whether the orientation control is to be space-related or path-related.

Orientation control	Description
base-related	Base-related orientation control during the circular motion
path-oriented	Path-related orientation control during the circular motion

(>>> 8.8.1 "Combinations of Orientation control and Circle orientation control" Page 192)

The option **path-oriented** is not allowed for the following motions:

- SCIRC segments for which **Ignore orientation** applies
- SCIRC motions preceded by a spline segment for which **Ignore orientation** applies

Orientation of the auxiliary point:

During SCIRC motions with the orientation control type **Default**, the robot controller takes the programmed orientation of the auxiliary point into consideration, but only to a certain extent.

The transition from the start orientation to the end orientation passes through the programmed orientation of the auxiliary point, i.e. the orientation of the auxiliary point is accepted at some point during the transition, but not necessarily at the auxiliary point.

8.8.1 Combinations of Orientation control and Circle orientation control

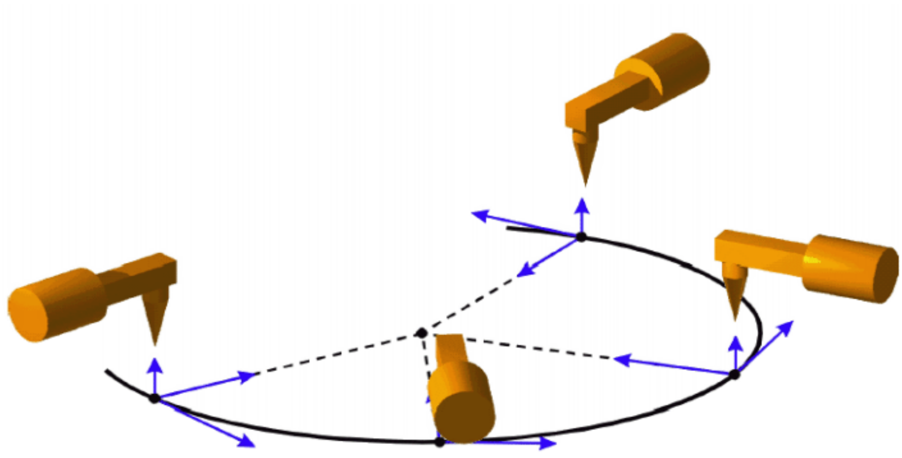


Fig. 8-26: Constant orientation control + path-related

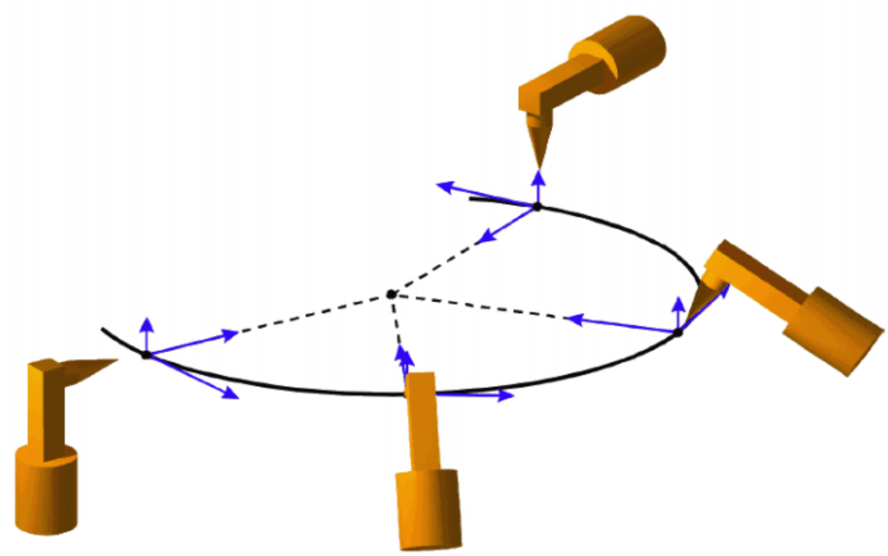


Fig. 8-27: Standard + path-related

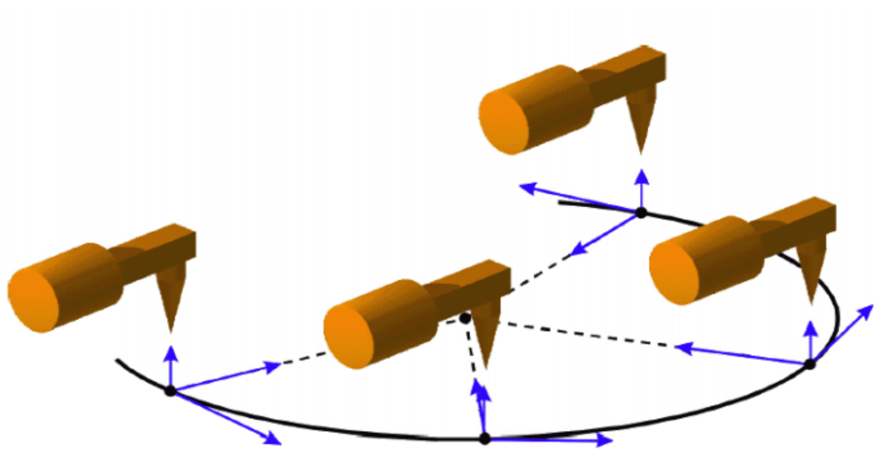


Fig. 8-28: Constant orientation control + base-related

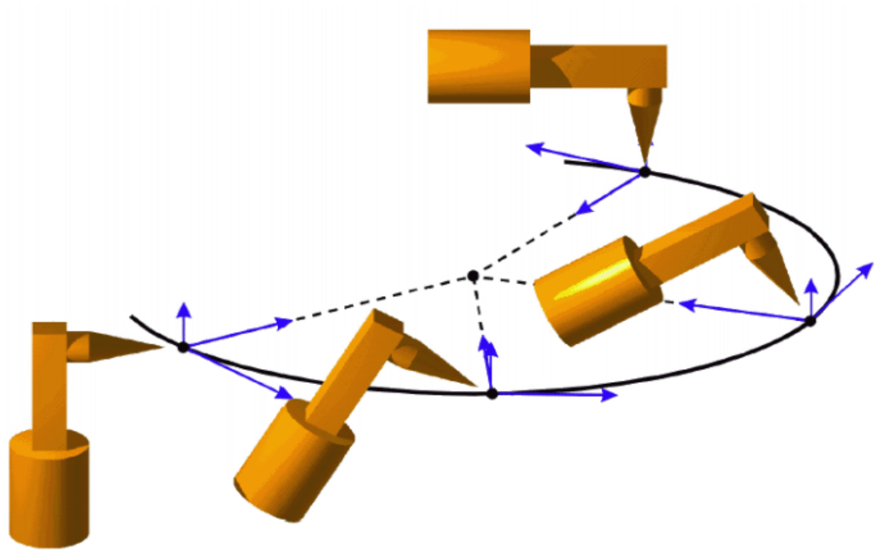


Fig. 8-29: Standard + base-related

8.9 Circular angle

A circular angle can be programmed for most circular motions.

> Information about whether this is possible for a specific circular motion can be found in the description of the motion in the programming section of this documentation.

The circular angle specifies the overall angle of the motion. This makes it possible to extend the motion beyond the programmed end point or to shorten it. The actual end point thus no longer corresponds to the programmed end point.

Unit: degrees. Circular angles greater than +360° and less than -360° can be programmed.

The preceding sign determines the direction in which the circular path is executed:

- Positive: direction **Start point › Auxiliary point › End point**
- Negative: direction **Start point › End point › Auxiliary point**

8.10 Singularities

KUKA robots with 6 degrees of freedom have 3 different singularity positions.

- Overhead singularity
- Extended position singularity
- Wrist axis singularity

A singularity position is characterized by the fact that unambiguous reverse transformation (conversion of Cartesian coordinates to axis-specific values) is not possible, even though Status and Turn are specified. In this case, or if very slight Cartesian changes cause very large changes to the axis angles, one speaks of singularity positions.

Overhead

In the overhead singularity, the wrist root point (intersection of axes A4, A5 and A6) is located vertically above axis 1.

The position of axis A1 cannot be determined unambiguously by means of reverse transformation and can thus take any value.

If the end point of a PTP motion is situated in this overhead singularity position, the robot controller may react as follows by means of the system variable $SINGUL_POS[1]:

- **0**: The angle for axis A1 is defined as 0 degrees (default setting).
- **1**: The angle for axis A1 remains the same from the start point to the end point.

Extended position

In the extended position singularity, the wrist root point (intersection of axes A4, A5 and A6) is located in the extension of axes A2 and A3 of the robot.

The robot is at the limit of its work envelope.

Although reverse transformation does provide unambiguous axis angles, low Cartesian velocities result in high axis velocities for axes A2 and A3.

If the end point of a PTP motion is situated in this extended position singularity, the robot controller may react as follows by means of the system variable $SINGUL_POS[2]:

- **0**: The angle for axis A2 is defined as 0 degrees (default setting).
- **1**: The angle for axis A2 remains the same from the start point to the end point.

Wrist axes

In the wrist axis singularity position, the axes A4 and A6 are parallel to one another and axis A5 is within the range ±0.01812°.

The position of the two axes cannot be determined unambiguously by reverse transformation. There is an infinite number of possible axis positions for axes A4 and A6 with identical axis angle sums.

If the end point of a PTP motion is situated in this wrist axis singularity, the robot controller may react as follows by means of the system variable $SINGUL_POS[3]:

- **0**: The angle for axis A4 is defined as 0 degrees (default setting).
- **1**: The angle for axis A4 remains the same from the start point to the end point.

9 Programming for user group “User” (inline forms)

9.1 Instructions for programming

NOTICE When programming motions, it must be ensured that the energy supply system is not wound up or damaged during program execution.

NOTICE In the case of programs with the following axis motions or positions, the film of lubricant on the gear units of the axes may break down:

- Motions <3°
- Oscillating motions
- Areas of gear units permanently facing upwards

It must be ensured that the gear units have a sufficient supply of oil. For this, in the case of oscillating motions or short motions (<3°), programming must be carried out in such a way that the affected axes regularly move more than 40° (e.g. once per cycle).
In the case of areas of gear units permanently facing upwards, sufficient oil supply must be achieved by programming re-orientations of the in-line wrist. In this way, the oil can reach all areas of the gear units by means of gravity.
Required frequency of re-orientations:

- With low loads (gear unit temperature <+35 °C): daily
- With medium loads (gear unit temperature +35 °C to 55 °C): hourly
- With heavy loads (gear unit temperature >+55 °C): every 10 minutes

Failure to observe this precaution may result in damage to the gear units.

9.2 Names in inline forms

Names for data sets can be entered in inline forms. These include, for example, point names, names for motion data sets, etc.

The following restrictions apply to names:

- Maximum length 23 characters
- No special characters are permissible, with the exception of $.
- The first character must not be a number.

The restrictions do not apply to output names.

Other restrictions may apply in the case of inline forms in technology packages.

9.3 Programming PTP, LIN and CIRC motions

9.3.1 Programming a PTP motion

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the position that is to be taught as the end point.
2. Position the cursor in the line after which the motion instruction is to be inserted.
3. Select the menu sequence **Commands** > **Motion** > **PTP**.
4. Set the parameters in the inline form.

(>>> 9.3.2 "Inline form “PTP”" Page 196)

5. Save instruction with **Cmd OK**.

9.3.2 Inline form “PTP”

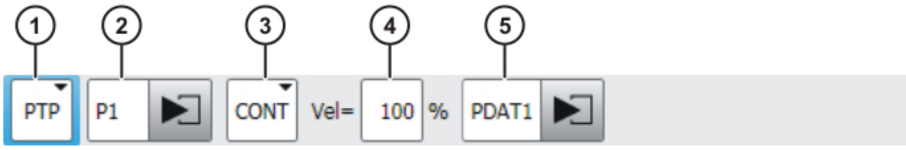


Fig. 9-1: Inline form for PTP motions

Item	Description
1	Motion type **PTP**
2	Name of the end point The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.7 "Option window “Frames”" Page 198)
3	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
4	Velocity ■ **1 … 100%**
5	Name for the motion data set The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.8 "Option window “Motion parameters” (LIN, CIRC, PTP)" Page 199)

9.3.3 Programming a LIN motion

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the position that is to be taught as the end point.
2. Position the cursor in the line after which the motion instruction is to be inserted.
3. Select the menu sequence **Commands** > **Motion** > **LIN**.
4. Set the parameters in the inline form.

   (>>> 9.3.4 "Inline form “LIN”" Page 197)
5. Save instruction with **Cmd OK**.

9.3.4 Inline form “LIN”

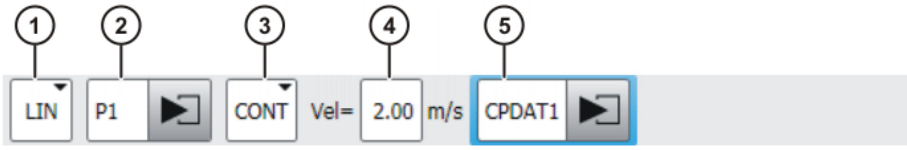


Fig. 9-2: Inline form for LIN motions

Item	Description
1	Motion type **LIN**
2	Name of the end point The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.7 "Option window “Frames”" Page 198)
3	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
4	Velocity ■ **0.001 … 2 m/s**
5	Name for the motion data set The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.8 "Option window “Motion parameters” (LIN, CIRC, PTP)" Page 199)

9.3.5 Programming a CIRC motion

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the position that is to be taught as the auxiliary point.
2. Position the cursor in the line after which the motion instruction is to be inserted.
3. Select the menu sequence **Commands** > **Motion** > **CIRC**.
4. Set the parameters in the inline form.
   (>>> 9.3.6 "Inline form “CIRC”" Page 198)
5. Press **Teach Aux**.
6. Move the TCP to the position that is to be taught as the end point.
7. Save instruction with **Cmd OK**.

9.3.6 Inline form “CIRC”

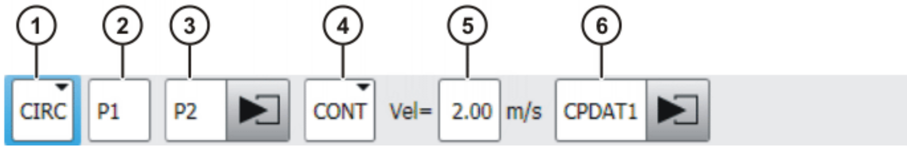


Fig. 9-3: Inline form for CIRC motions

Item	Description
1	Motion type **CIRC**
2	Name of the auxiliary point The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195)
3	Name of the end point The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.7 "Option window “Frames”" Page 198)
4	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
5	Velocity ■ **0.001 … 2 m/s**
6	Name for the motion data set The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.8 "Option window “Motion parameters” (LIN, CIRC, PTP)" Page 199)

9.3.7 Option window “Frames”

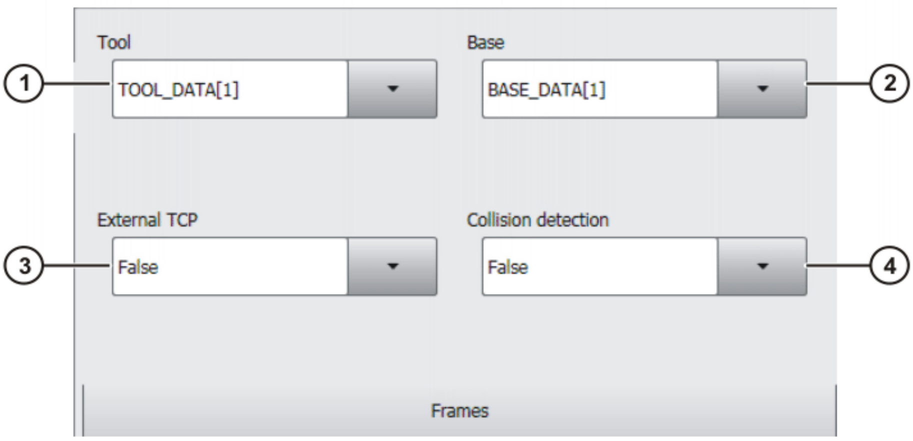


Fig. 9-4: Option window “Frames”

Item	Description
1	Tool selection. If **True** in the box **External TCP**: workpiece selection. Range of values: [1] … [16]
2	Base selection. If **True** in the box **External TCP**: fixed tool selection. Range of values: [1] … [32]
3	Interpolation mode ■ **False**: The tool is mounted on the mounting flange. ■ **True**: The tool is a fixed tool.
4	■ **True**: For this motion, the robot controller calculates the axis torques. These are required for collision detection. ■ **False**: For this motion, the robot controller does not calculate the axis torques. Collision detection is thus not possible for this motion.

9.3.8 Option window “Motion parameters” (LIN, CIRC, PTP)

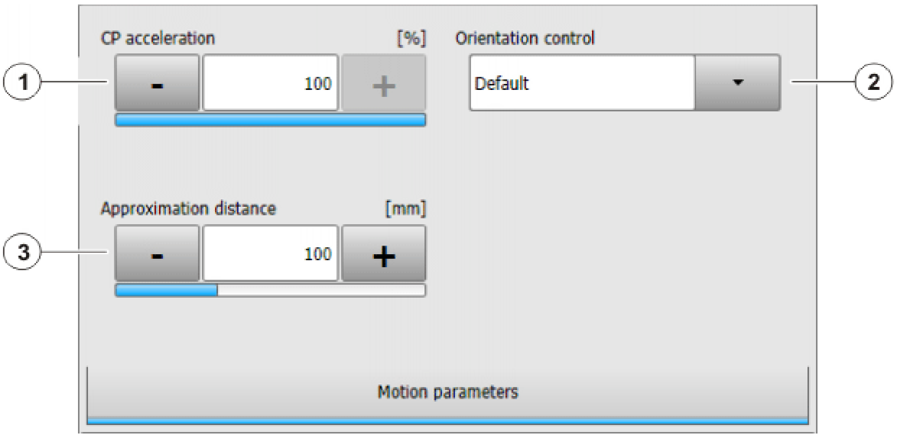


Fig. 9-5: Option window “Motion parameters” (LIN, CIRC, PTP)

Item	Description
1	Acceleration Refers to the maximum value specified in the machine data. The maximum value depends on the robot type and the selected operating mode.

Item	Description
2	This box is only displayed if it is specified in the inline form that the point is to be approximated. Furthest distance before the end point at which approximate positioning can begin The maximum permissible value is half the distance between the start point and the end point. If a higher value is entered, this is ignored and the maximum value is used.
3	This box is only displayed for LIN and CIRC motions. Orientation control selection. ■ **Standard** ■ **Wrist PTP** ■ **Constant orientation** (>>> 8.6 "Orientation control LIN, CIRC" Page 178)

9.4 Programming spline motions

9.4.1 Programming tips for spline motions

- It is only possible to exploit the advantages of the spline motion type to the full if spline blocks are used.
- A spline block should cover only one process (e.g. an adhesive seam). More than one process in a spline block leads to a loss of structural clarity within the program and makes changes more difficult.
- Use SLIN and SCIRC segments in cases where the workpiece necessitates straight lines and arcs. (Exception: use SPL segments for very short straight lines.) Otherwise, use SPL segments, particularly if the points are close together.
- Procedure for defining the path:
  a. First teach or calculate a few characteristic points. Example: points at which the curve changes direction.
  b. Test the path. At points where the accuracy is still insufficient, add more SPL points.
- Avoid successive SLIN and/or SCIRC segments, as this often reduces the velocity to 0.

  Program SPL segments between SLIN and SCIRC segments. The length of the SPL segments must be at least > 0.5 mm. Depending on the actual path, significantly larger SPL segments may be required.
- Avoid successive points with identical Cartesian coordinates, as this reduces the velocity to 0.
- The parameters (tool, base, velocity, etc.) assigned to the spline block have the same effect as assignments before the spline block. The assignment to the spline block has the advantage, however, that the correct parameters are read in the case of a block selection.
- Use the option **Ignore orientation** if no specific orientation is required for a SLIN, SCIRC or SPL segment. The robot controller calculates the optimal orientation for this point on the basis of the orientations of the surrounding points. This improves the cycle time.
- The jerk can be modified The jerk is the change in acceleration. Procedure:
  a. Use the default values initially.
  b. If vibrations occur at tight corners: reduce values.

If the velocity drops or the desired velocity cannot be reached: increase values or increase acceleration.

- If the robot executes points on a work surface, a collision with the work surface is possible when the first point is addressed.

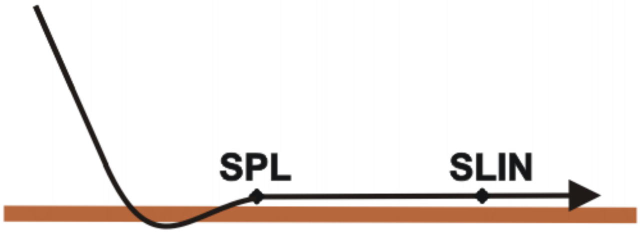


Fig. 9-6: Collision with work surface

In order to avoid a collision, observe the recommendations for the SLIN-SPL-SLIN transition.

(>>> 8.7.5.1 "SLIN-SPL-SLIN transition" Page 189)

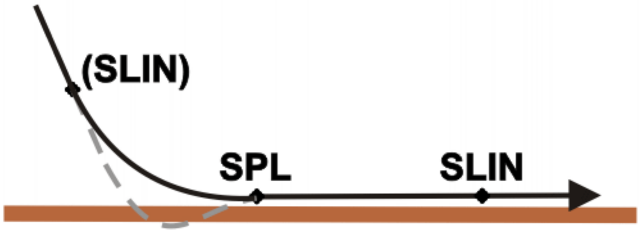


Fig. 9-7: Avoiding a collision with the work surface

- In the case of PTP spline blocks with multiple SPTP segments, it is possible that the software limit switches may be violated even though the points are within the limits!

  In this case, the points must be re-taught, i.e. they must be moved further away from the software limit switches. Alternatively, the software limit switches can be modified, provided that the required machine protection is still assured.

9.4.2 Programming a spline block

Description

A spline block can be used to group together several motions as an overall motion. The motions that may be included in a spline block are called spline segments. They are taught separately.

A spline block is planned and executed by the robot controller as a single motion block.

- A CP spline block may contain SPL, SLIN and SCIRC segments.
- A PTP spline block may contain SPTP segments.

A spline block that contains no segments is not a motion statement. The number of segments in the block is only limited by the memory capacity. Apart from the segments, a spline block may also contain the following elements:

- Inline commands from technology packages that support the spline functionality
- Comments and blank lines

A spline block must not include any other instructions, e.g. variable assignments or logic statements.

The start point of a spline block is the last point before the spline block.
The end point of a spline block is the last point in the spline block.
A spline block does not trigger an advance run stop.

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the spline block is to be inserted.
2. Select the menu sequence **Commands** > **Motion**.
   - Then select **SPLINE block** for a CP spline block.
   - Or select **PTP SPLINE block** for a PTP spline block.
3. Set the parameters in the inline form.
   (>>> 9.4.2.1 "Inline form for CP spline block" Page 202)
   (>>> 9.4.2.2 "Inline form “PTP SPLINE block”" Page 203)
4. Press **Cmd OK**.
5. Press **Open/close fold**. Spline segments can now be inserted into the block.

9.4.2.1 Inline form for CP spline block

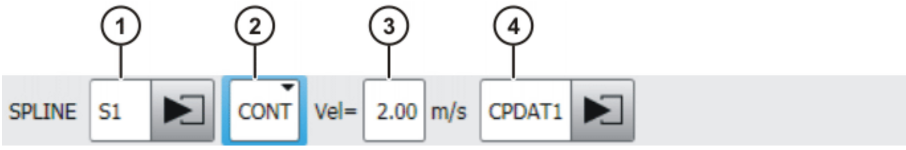


Fig. 9-8: Inline form for CP spline block

Item	Description
1	Name of the spline block. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Position the cursor in this box to edit the motion data. The corresponding option window is opened. (>>> 9.4.2.3 "Option window “Frames” (CP and PTP spline block)" Page 203)
2	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
3	Cartesian velocity ■ **0.001 … 2 m/s**
4	Name for the motion data set. The system automatically generates a name. The name can be overwritten. Position the cursor in this box to edit the motion data. The corresponding option window is opened. (>>> 9.4.2.4 "Option window “Motion parameters” (CP spline block)" Page 204)

9.4.2.2 Inline form “PTP SPLINE block”

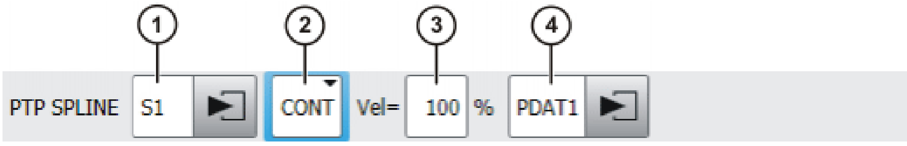


Fig. 9-9: Inline form “PTP SPLINE block”

Item	Description
1	Name of the spline block. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Position the cursor in this box to edit the motion data. The corresponding option window is opened. (>>> 9.4.2.3 "Option window “Frames” (CP and PTP spline block)" Page 203)
2	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
3	Axis velocity ■ **1 … 100%**
4	Name for the motion data set. The system automatically generates a name. The name can be overwritten. Position the cursor in this box to edit the motion data. The corresponding option window is opened. (>>> 9.4.2.5 "Option window “Motion parameters” (PTP spline block)" Page 205)

9.4.2.3 Option window “Frames” (CP and PTP spline block)

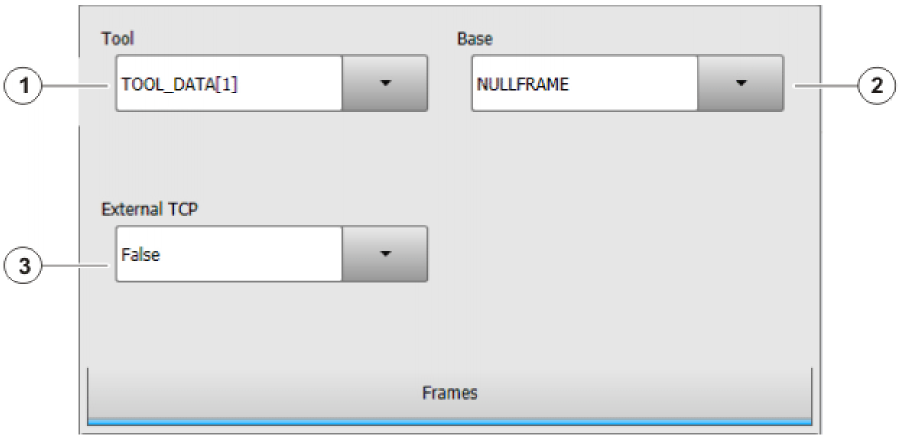


Fig. 9-10: Option window “Frames” (CP and PTP spline block)

Item	Description
1	Tool selection. Or: If **True** in the box **External TCP**: workpiece selection. ■ **[1]** … **[16]**
2	Base selection. Or: If **True** in the box **External TCP**: fixed tool selection. ■ **[1]** … **[32]**
3	Interpolation mode ■ **False**: The tool is mounted on the mounting flange. ■ **True**: The tool is a fixed tool.

9.4.2.4 Option window "Motion parameters" (CP spline block)

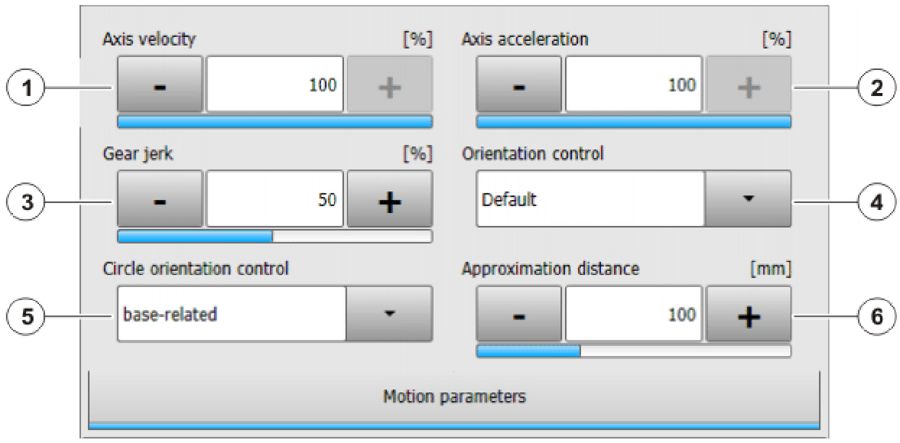


Fig. 9-11: Option window "Motion parameters" (CP spline block)

Item	Description
1	Axis velocity. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
2	Axis acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
3	Gear jerk. The jerk is the change in acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
4	Orientation control selection.
5	Orientation control reference system selection. This parameter only affects SCIRC segments (if present).
6	This box is only displayed if **CONT** was selected in the inline form. Furthest distance before the end point at which approximate positioning can begin. The maximum distance is that of the last segment in the spline. If there is only one segment present, the maximum distance is half the segment length. If a higher value is entered, this is ignored and the maximum value is used.

9.4.2.5 Option window "Motion parameters" (PTP spline block)

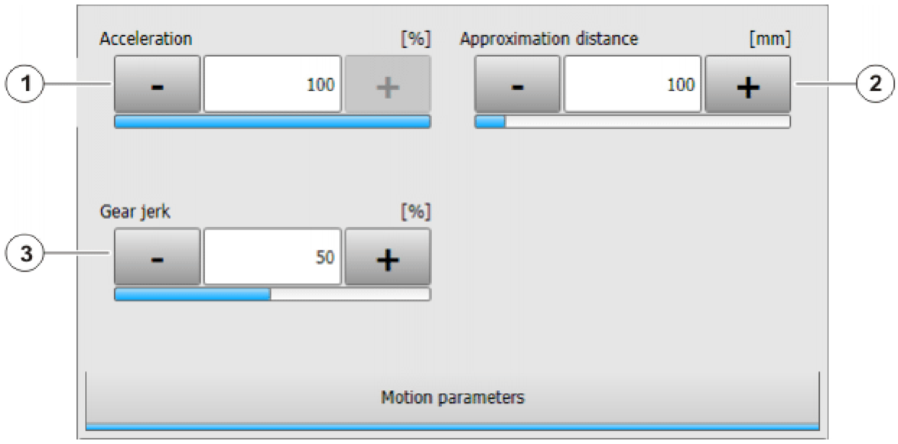


Fig. 9-12: Option window "Motion parameters" (PTP spline block)

Item	Description
1	Axis acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
2	This box is only displayed if **CONT** was selected in the inline form. Furthest distance before the end point at which approximate positioning can begin. The maximum distance is that of the last segment in the spline. If there is only one segment present, the maximum distance is half the segment length. If a higher value is entered, this is ignored and the maximum value is used.
3	Gear jerk. The jerk is the change in acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**

9.4.3 Programming segments for a spline block

9.4.3.1 Programming an SPL or SLIN segment

Precondition

- A program is selected.
- Operating mode T1
- The CP spline block fold is open.

Procedure

1. Move the TCP to the end point.
2. Position the cursor in the line after which the segment is to be inserted in the spline block.
3. Select the menu sequence **Commands** > **Motion** > **SPL** or **SLIN**.
4. Set the parameters in the inline form.
5. Press **Cmd OK**.

9.4.3.2 Programming an SCIRC segment

Precondition

- A program is selected.
- Operating mode T1

- The CP spline block fold is open.

Procedure

1. Move the TCP to the auxiliary point.
2. Position the cursor in the line after which the segment is to be inserted in the spline block.
3. Select the menu sequence **Commands** > **Motion** > **SCIRC**.
4. Set the parameters in the inline form.
   (>>> 9.4.3.3 "Inline form for CP spline segment" Page 206)
5. Press **Teach Aux**.
6. Move the TCP to the end point.
7. Press **Cmd OK**.

9.4.3.3 Inline form for CP spline segment

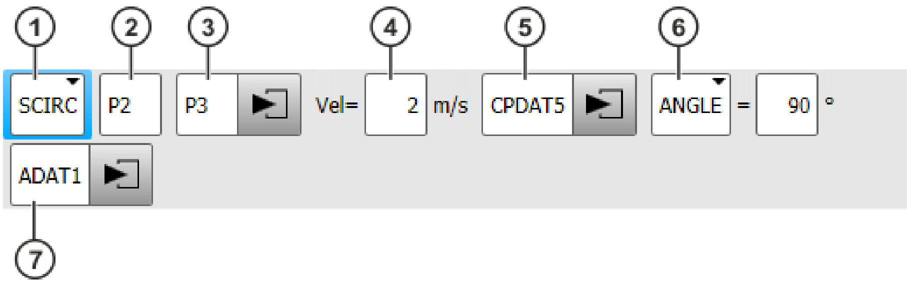


Fig. 9-13: Inline form for CP spline segment

By default, not all boxes of the inline form are displayed. The boxes can be displayed or hidden using the **Switch parameter** button.

Item	Description
1	Motion type ■ **SPL**, **SLIN** or **SCIRC**
2	Only for **SCIRC**: Point name for the auxiliary point. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195)
3	Point name for the end point. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.4.3.6 "Option window “Frames” (CP and PTP spline segments)" Page 208)
4	Cartesian velocity By default, the value that is valid for the spline block is also valid for the segment. A separate value can be assigned here for the segment if required. The value applies only for this segment. ■ **0.001 … 2 m/s**

Item	Description
5	Name for the motion data set. The system automatically generates a name. The name can be overwritten. By default, the values that are valid for the spline block are also valid for the segment. Separate values can be assigned here for the segment if required. The values apply only for this segment. Touch the arrow to edit the data. The corresponding option window is opened. (>>> 9.4.3.7 "Option window “Motion parameters” (CP spline segment)" Page 209)
6	Circular angle Only available if the motion type **SCIRC** has been selected. ■ **- 9,999° … + 9,999°** If a value less than -400° or greater than +400° is entered, a request for confirmation is generated when the inline form is saved asking whether entry is to be confirmed or rejected.
7	Name of the data set containing logic parameters. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the data. The corresponding option window is opened. (>>> 9.4.3.9 "Option window “Logic parameters”" Page 210)

9.4.3.4 Programming an SPTP segment

Precondition

- A program is selected.
- Operating mode T1
- The PTP spline block fold is open.

Procedure

1. Move the TCP to the end point.
2. Position the cursor in the line after which the segment is to be inserted in the spline block.
3. Select the menu sequence **Commands** > **Motion** > **SPTP**.
4. Set the parameters in the inline form.
   (>>> 9.4.3.5 "Inline form for SPTP segment" Page 207)
5. Press **Cmd OK**.

9.4.3.5 Inline form for SPTP segment

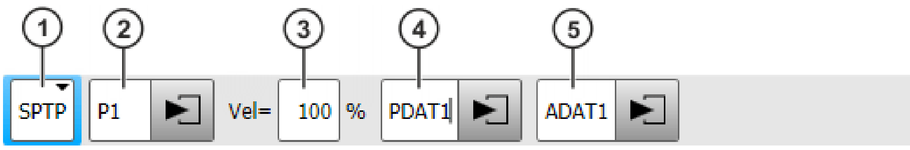


Fig. 9-14: Inline form for SPTP segment

By default, not all boxes of the inline form are displayed. The boxes can be displayed or hidden using the **Switch parameter** button.

Item	Description
1	Motion type **SPTP**
2	Point name for end point. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.4.3.6 "Option window “Frames” (CP and PTP spline segments)" Page 208)
3	Axis velocity By default, the value that is valid for the spline block is also valid for the segment. A separate value can be assigned here for the segment if required. The value applies only for this segment. ■ **1 … 100%**
4	Name for the motion data set. The system automatically generates a name. The name can be overwritten. By default, the values that are valid for the spline block are also valid for the segment. Separate values can be assigned here for the segment if required. The values apply only for this segment. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.4.3.8 "Option window “Motion parameters” (SPTP)" Page 210)
5	Name of the data set containing logic parameters. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the data. The corresponding option window is opened. (>>> 9.4.3.9 "Option window “Logic parameters”" Page 210)

9.4.3.6 Option window “Frames” (CP and PTP spline segments)

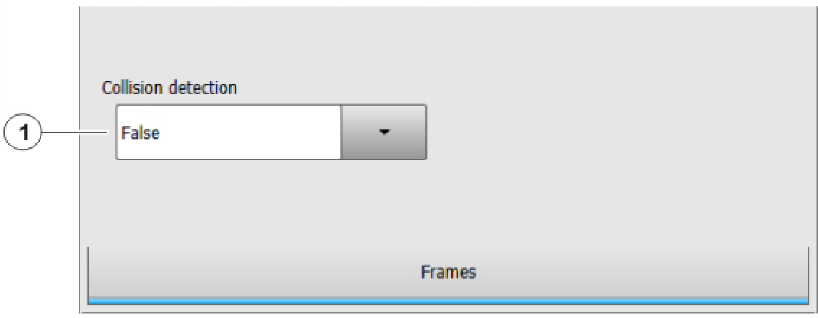


Fig. 9-15: Option window “Frames” (CP and PTP spline segments)

Item	Description
1	■ **True**: For this motion, the robot controller calculates the axis torques. These are required for collision detection. ■ **False**: For this motion, the robot controller does not calculate the axis torques. Collision detection is thus not possible for this motion.

9.4.3.7 Option window “Motion parameters” (CP spline segment)

Motion parameters

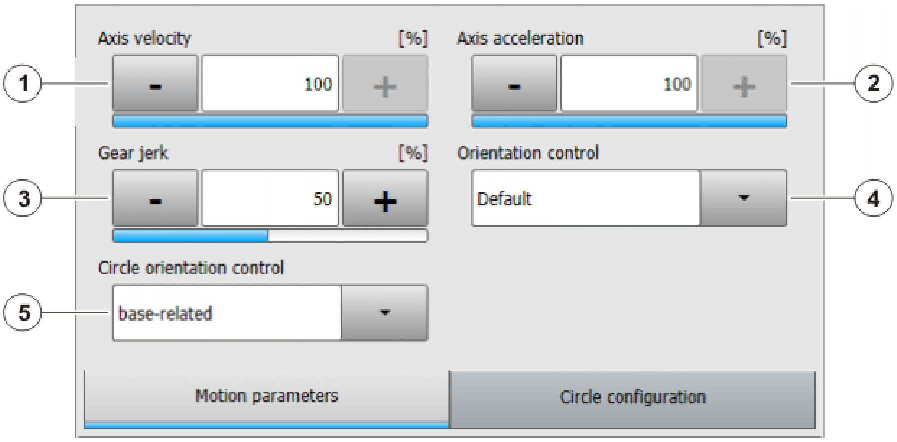


Fig. 9-16: Option window “Motion parameters” (CP spline segment)

Item	Description
1	Axis velocity. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
2	Axis acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
3	Gear jerk. The jerk is the change in acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
4	Orientation control selection.
5	Only in the case of SCIRC segments: Orientation control reference system selection.

Circle configuration

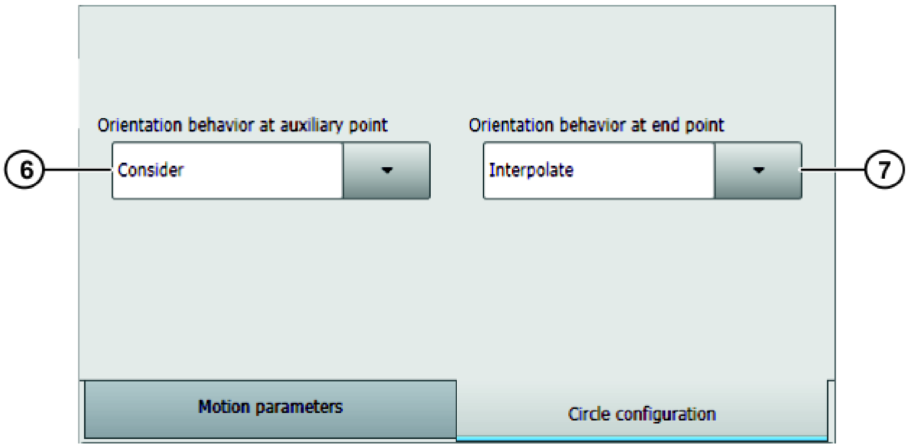


Fig. 9-17: Circle configuration (SCIRC segment)

Item	Description
6	Only in the case of SCIRC segments: Selection of orientation behavior at auxiliary point.
7	Only in the case of SCIRC segments: This box is only displayed if **ANGLE** was selected in the inline form. Selection of orientation behavior at end point.

9.4.3.8 Option window “Motion parameters” (SPTP)

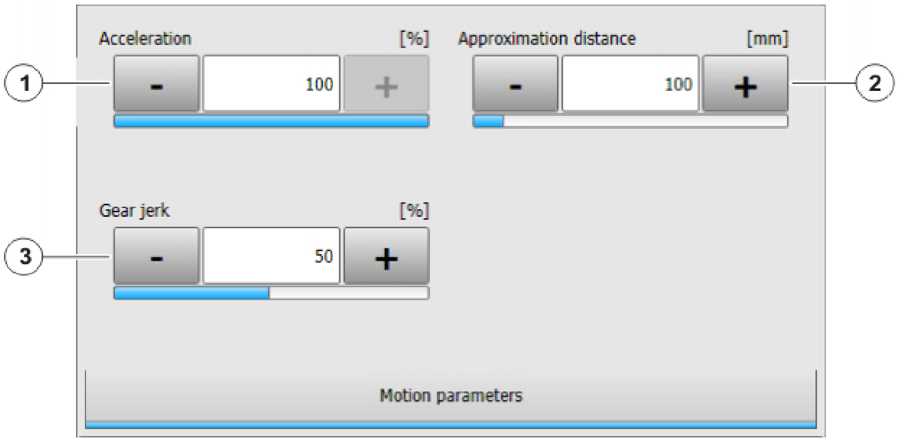


Fig. 9-18: Option window “Motion parameters” (SPTP)

Item	Description
1	Axis acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
2	This box is not available for SPTP segments. In the case of individual SPTP motions, this box is only displayed if **CONT** was selected in the inline form. Furthest distance before the end point at which approximate positioning can begin. The maximum permissible value is half the distance between the start point and the end point. If a higher value is entered, this is ignored and the maximum value is used.
3	Gear jerk. The jerk is the change in acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**

9.4.3.9 Option window “Logic parameters”

Trigger

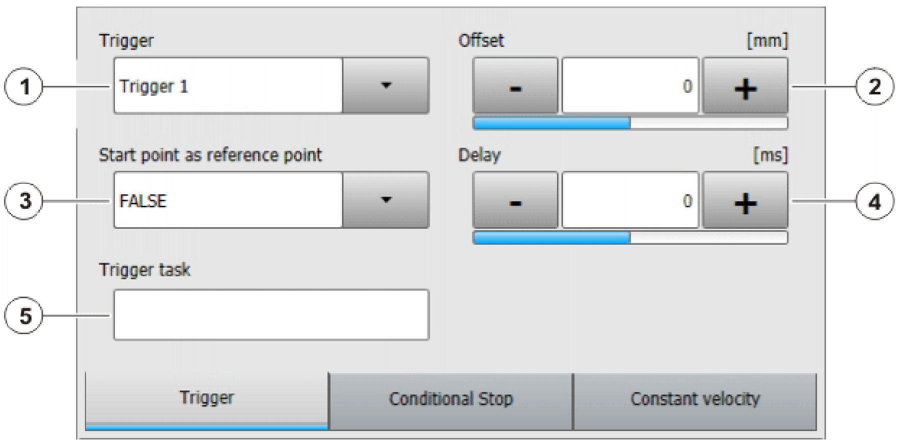


Fig. 9-19: Trigger

Information about the location of the switching point if the reference point is approximated is contained in the Operating and Programming Instructions for System Integrators. Information about maximum offset limits can also be found there.

Item	Description
1	A (further) trigger can be assigned to the motion by means of the button **Select action** > **Add trigger**. If it is the first trigger for this motion, this command also causes the **Trigger** box to be displayed. A maximum of 8 triggers per motion are possible. (A trigger can be removed again by means of **Select action** > **Remove trigger**.)
2	Reference point of the trigger ■ **TRUE**: Start point ■ **FALSE**: End point
3	Spatial offset relative to the end or start point ■ Negative value: Shift towards the start of the motion ■ Positive value: Shift towards the end of the motion The shift in space can also be taught. In this case, the box **Start point is reference point** is automatically set to **FALSE**. (>>> 9.4.3.10 "Teaching the shift in space for logic parameters" Page 213)
4	Shift in time relative to **Offset** ■ Negative value: Shift towards the start of the motion. ■ Positive value: Trigger is switched after *Time* has elapsed.
5	Statement that is to be initiated by the trigger. The following are possible: ■ Assignment of a value to a variable **Note:** There must be no runtime variable on the left-hand side of the assignment. ■ OUT statement; PULSE statement; CYCFLAG statement ■ Subprogram call. In this case, the priority must be specified. Example: `my_subprogram() PRIO = 81` Priorities 1, 2, 4 to 39 and 81 to 128 are available. Priorities 40 to 80 are reserved for cases in which the priority is automatically assigned by the system. If the priority is to be assigned automatically by the system, the following is programmed: `PRIO = -1`.. If several triggers call subprograms at the same time, the trigger with the highest priority is processed first, then the triggers of lower priority. 1 = highest priority.

Conditional stop

Further information about the conditional stop can be found in this documentation.
(>>> 9.4.5 "Conditional stop" Page 219)

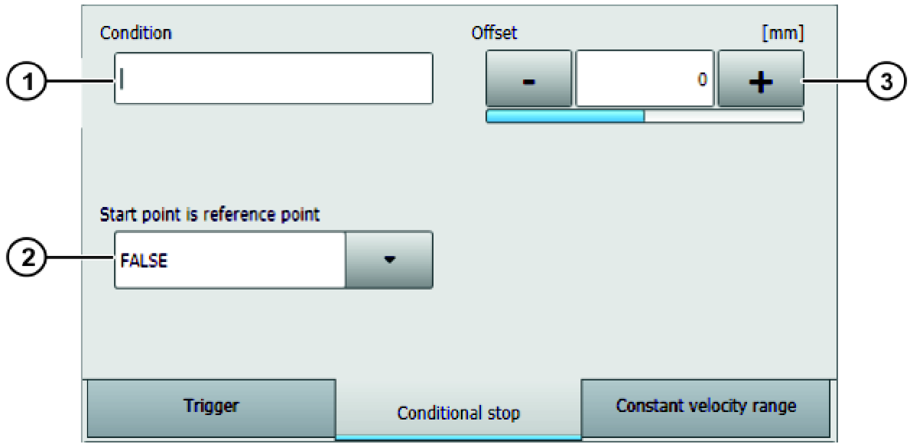


Fig. 9-20: Conditional stop

Item	Description
1	Stop condition. The following are permitted: ■ a global Boolean variable ■ a signal name ■ a comparison ■ a simple logic operation: NOT, OR, AND or EXOR
2	The conditional stop can refer to either the start point or the end point of the motion. ■ **TRUE**: Start point ■ **FALSE**: End point If the reference point is approximated, the same rules apply as for the PATH trigger. **Note:** Information about approximate positioning with the PATH trigger is contained in the Operating and Programming Instructions for System Integrators.
3	The stop point can be shifted in space. For this, the desired distance from the start or end point must be specified. If no shift in space is desired, enter “0”. ■ Positive value: Shift towards the end of the motion ■ Negative value: Shift towards the start of the motion There are limits to the distance the stop point can be offset. The same limits apply as for the PATH trigger. The shift in space can also be taught. In this case, the box **Start point is reference point** is automatically set to **FALSE**. (>>> 9.4.3.10 "Teaching the shift in space for logic parameters" Page 213) **Note:** Information about the offset limits with the PATH trigger is contained in the Operating and Programming Instructions for System Integrators.

Constant velocity range

Constant velocity range is only available for CP spline segments.

Basic information about the constant velocity range can be found here:
(>>> 9.4.6 "Constant velocity range in the CP spline block" Page 223)

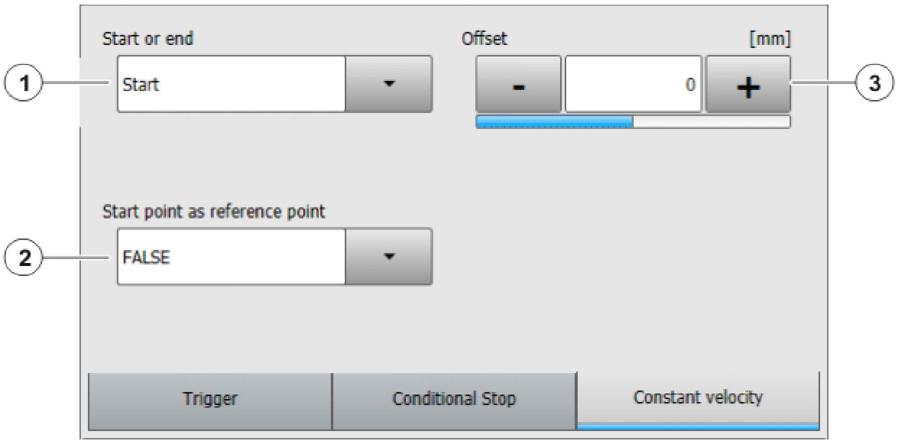


Fig. 9-21: Constant velocity range

Item	Description
1	■ **Start**: Defines the start of the constant velocity range. ■ **End**: Defines the end of the constant velocity range.
2	**Start** and **End** can refer to either the start point or the end point of the motion. ■ **TRUE**: **Start** or **End** refers to the start point. If the start point is approximated, the reference point is generated in the same way as for homogenous approximate positioning with the PATH trigger. **Note:** Information about approximate positioning with the PATH trigger is contained in the Operating and Programming Instructions for System Integrators. ■ **FALSE**: **Start** or **End** refers to the end point. If the end point is approximated, **Start** or **End** refers to the start of the approximate positioning arc.
3	The start or end of the constant velocity range can be shifted in space. For this, the desired distance must be specified. If no shift in space is desired, enter “0”. ■ Positive value: Offset towards the end of the motion ■ Negative value: Offset towards the start of the motion (>>> 9.4.6.2 "Maximum limits" Page 224) The shift in space can also be taught. In this case, the box **Start point is reference point** is automatically set to **FALSE**. (>>> 9.4.3.10 "Teaching the shift in space for logic parameters" Page 213)

9.4.3.10 Teaching the shift in space for logic parameters

Description

Shifts in space can be specified in the option window **Logic parameters** for trigger, conditional stop and constant velocity range. Instead of entering these offsets numerically, they can also be taught.

> If an offset is taught, the box **Start point is reference point** in the corresponding tab is automatically set to **FALSE**, as the taught distance refers to the end point of the motion.

Precondition

- A program is selected.
- Operating mode T1
- The point for which the offset is to apply has already been taught.

Procedure

1. Move the TCP to the desired position.
2. Position the cursor in the line containing the motion instruction for which the offset is to be taught.
3. Press **Change**. The inline form for this instruction is opened.
4. Open the option window **Logic parameters** and select the required tab.
5. Press **Select action** then one of the following buttons depending on what the offset is to be taught for:
   - **Teach trigger path**
   - **Teach conditional stop path**
   - **Teach constant velocity range path**

   The distance from the end point of the current motion statement is now applied as the value for the offset.
6. Save the change by pressing **Cmd OK**.

9.4.4 Programming individual spline motions

9.4.4.1 Programming an individual SLIN motion

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the end point.
2. Position the cursor in the line after which the motion is to be inserted.
3. Select **Commands** > **Motion** > **SLIN**.
4. Set the parameters in the inline form.

   (>>> 9.4.4.2 "Inline form “SLIN”" Page 214)
5. Press **Cmd OK**.

9.4.4.2 Inline form “SLIN”

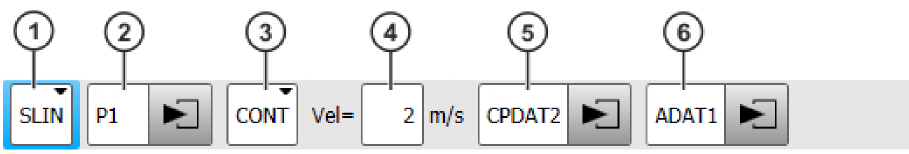


Fig. 9-22: Inline form “SLIN” (individual motion)

Item	Description
1	Motion type **SLIN**
2	Point name for end point. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.7 "Option window “Frames”" Page 198)
3	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
4	Velocity ■ **0.001 … 2 m/s**
5	Name for the motion data set. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.4.4.3 "Option window “Motion parameters” (SLIN)" Page 215)
6	This box can be displayed or hidden by means of **Switch parameter**. Name of the data set containing logic parameters. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the data. The corresponding option window is opened. (>>> 9.4.3.9 "Option window “Logic parameters”" Page 210)

9.4.4.3 Option window “Motion parameters” (SLIN)

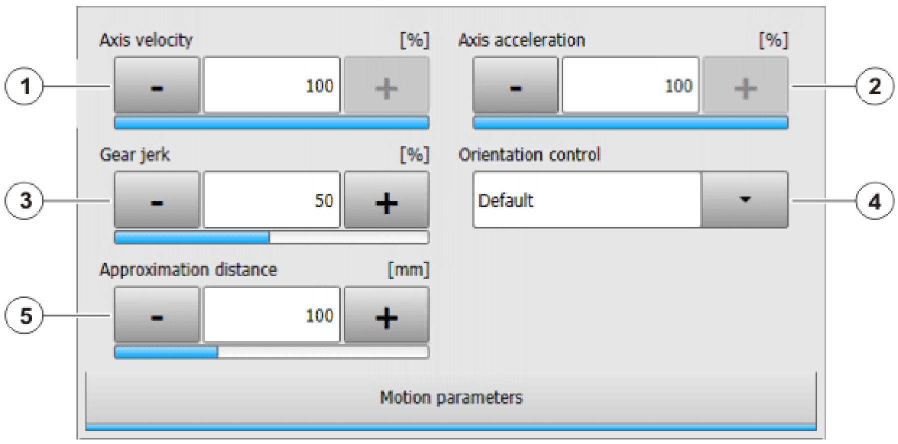


Fig. 9-23: Option window “Motion parameters” (SLIN)

Item	Description
1	Axis velocity. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
2	Axis acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**

Item	Description
3	Gear jerk. The jerk is the change in acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
4	Orientation control selection.
5	This box is only displayed if **CONT** was selected in the inline form. Furthest distance before the end point at which approximate positioning can begin. The maximum permissible value is half the distance between the start point and the end point. If a higher value is entered, this is ignored and the maximum value is used.

9.4.4.4 Programming an individual SCIRC motion

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the auxiliary point.
2. Position the cursor in the line after which the motion is to be inserted.
3. Select the menu sequence **Commands** > **Motion** > **SCIRC**.
4. Set the parameters in the inline form.
   (>>> 9.4.4.5 "Inline form “SCIRC”" Page 216)
5. Press **Teach Aux**.
6. Move the TCP to the end point.
7. Press **Cmd OK**.

9.4.4.5 Inline form “SCIRC”

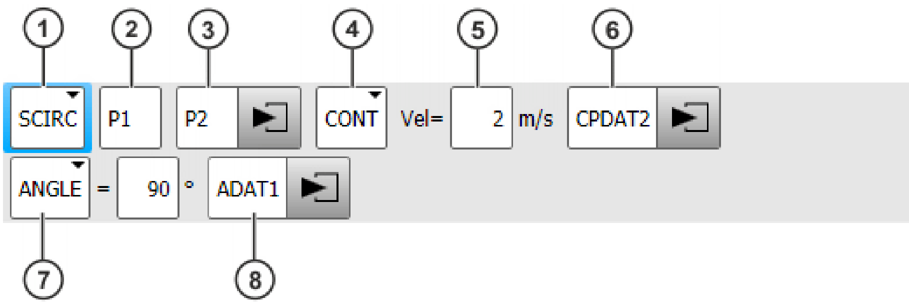


Fig. 9-24: Inline form “SCIRC” (individual motion)

Item	Description
1	Motion type **SCIRC**
2	Point name for the auxiliary point. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195)

Item	Description
3	Point name for the end point. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.7 "Option window “Frames”" Page 198)
4	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
5	Velocity ■ **0.001 … 2 m/s**
6	Name for the motion data set. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.4.4.6 "Option window “Motion parameters” (SCIRC)" Page 217)
7	Circular angle ■ **- 9,999° … + 9,999°** If a circular angle less than -400° or greater than +400° is entered, a request for confirmation is generated when the inline form is saved asking whether entry is to be confirmed or rejected.
8	This box can be displayed or hidden by means of **Switch parameter**. Name of the data set containing logic parameters. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the data. The corresponding option window is opened. (>>> 9.4.3.9 "Option window “Logic parameters”" Page 210)

9.4.4.6 Option window “Motion parameters” (SCIRC)

Motion parameters

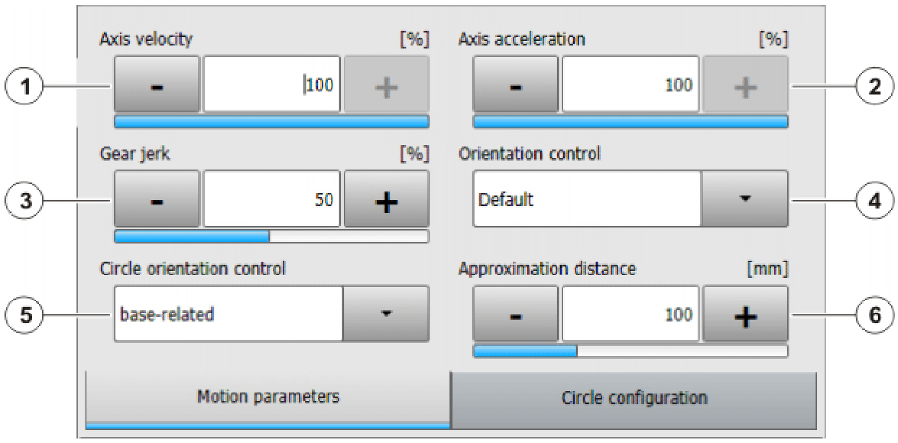


Fig. 9-25: Motion parameters (SCIRC)

Item	Description
1	Axis velocity. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
2	Axis acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
3	Gear jerk. The jerk is the change in acceleration. The value refers to the maximum value specified in the machine data. ■ **1 … 100%**
4	Orientation control selection.
5	Orientation control reference system selection.
6	This box is only displayed if **CONT** was selected in the inline form. Furthest distance before the end point at which approximate positioning can begin. The maximum permissible value is half the distance between the start point and the end point. If a higher value is entered, this is ignored and the maximum value is used.

Circle configuration

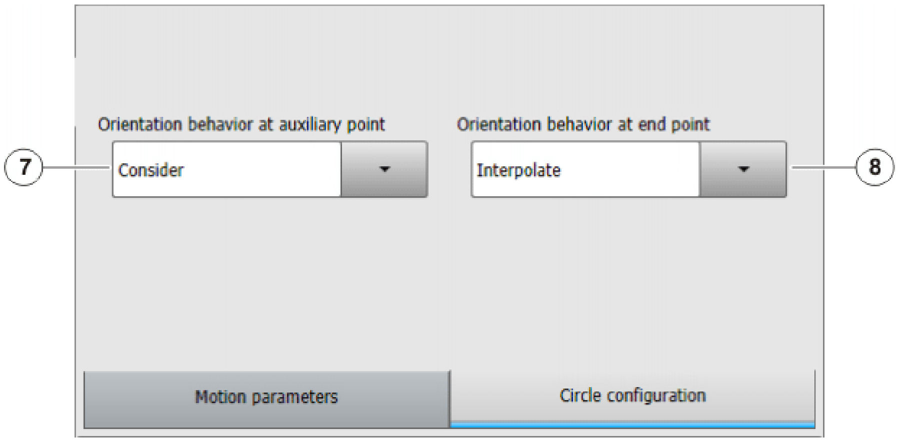


Fig. 9-26: Circle configuration (SCIRC)

Item	Description
7	Selection of orientation behavior at auxiliary point.
8	This box is only displayed if **ANGLE** was selected in the inline form. Selection of orientation behavior at end point.

9.4.4.7 Programming an individual SPTP motion

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the end point.
2. Position the cursor in the line after which the motion is to be inserted.
3. Select **Commands** > **Motion** > **SPTP**.
4. Set the parameters in the inline form.
   (>>> 9.4.4.8 "Inline form “SPTP”" Page 219)

5. Press **Cmd OK**.

9.4.4.8 Inline form “SPTP”

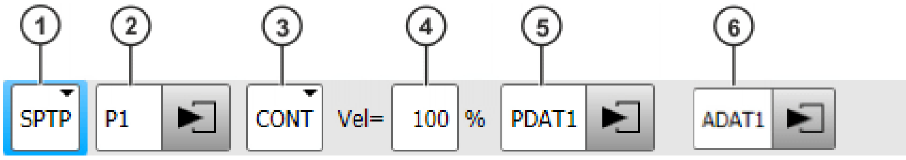


Fig. 9-27: Inline form “SPTP” (individual motion)

Item	Description
1	Motion type **SPTP**
2	Point name for end point. The system automatically generates a name. The name can be overwritten. (>>> 9.2 "Names in inline forms" Page 195) Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.3.7 "Option window “Frames”" Page 198)
3	■ **CONT**: End point is approximated. ■ **[blank]**: The motion stops exactly at the end point.
4	Velocity ■ **1 … 100%**
5	Name for the motion data set. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 9.4.3.8 "Option window “Motion parameters” (SPTP)" Page 210)
6	This box can be displayed or hidden by means of **Switch parameter**. Name of the data set containing logic parameters. The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the data. The corresponding option window is opened. (>>> 9.4.3.9 "Option window “Logic parameters”" Page 210)

9.4.5 Conditional stop

Description

The conditional stop enables the user to define a point on the path at which the robot stops if a certain condition is met. This point is called the “stop point”. As soon as the condition is no longer met, the robot resumes its motion.

During the runtime, the robot controller calculates the latest point at which the robot must brake in order to be able to stop at the stop point. From this point (braking point) onwards, it monitors whether or not the condition is met.

- If the condition is met at the braking point, the robot brakes in order to stop at the stop point.

  If, however, the condition then switches back to “not met” before the stop point is reached, the robot accelerates again and does not stop.
- If the condition is not met at the braking point, the robot motion is continued without braking.

Essentially, any number of conditional stops can be programmed. A maximum of 10 “braking point → stop point” paths may overlap, however.

While the robot is braking, the robot controller displays the following message in T1/T2 mode: *Conditional stop active (line {Line number}).*

(>>> 9.4.5.2 "Stop condition: example and braking characteristics" Page 221)

Programming

Programming with KRL syntax:

- using the statement STOP WHEN PATH

Programming with inline forms:

- In the spline block (CP and PTP) or in the individual spline block:
  in the option window “**Logic parameters**”
- Before a spline block (CP and PTP):
  via the inline form **Spline Stop Condition**

9.4.5.1 Inline form “Spline Stop Condition”

This inline form may only be used before a spline block. There may be other statements between the inline form and the spline block, but no motion instructions.

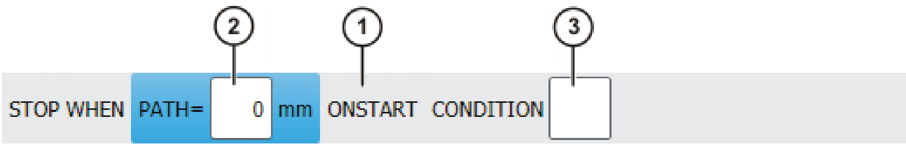


Fig. 9-28: Inline form “Spline Stop Condition”

Item	Description
1	Point to which the conditional stop refers ■ With ONSTART: last point before the spline block ■ Without ONSTART: last point in the spline block If the spline is approximated, the same rules apply as for the PATH trigger. **Note:** Information about approximate positioning with the PATH trigger is contained in the Operating and Programming Instructions for System Integrators. ONSTART can be set or removed using the **Toggle OnStart** button.

Item	Description
2	The stop point can be shifted in space. For this, the desired distance from the reference point must be specified. If no shift in space is desired, enter “0”. ■ Positive value: Offset towards the end of the motion ■ Negative value: Offset towards the start of the motion There are limits to the distance the stop point can be offset. The same limits apply as for the PATH trigger. The shift in space can also be taught. (>>> "Teach path" Page 221)
3	Stop condition The following are permitted: ■ a global Boolean variable ■ a signal name ■ a comparison ■ a simple logic operation: NOT, OR, AND or EXOR

Teach path

Button	Description
Teach path	If an offset is desired, it is not necessary to enter the value into the inline form numerically; the offset can also be taught. This is done via **Teach path**. If an offset is taught, ONSTART is automatically removed, if set in the inline form, as the taught distance always refers to the end point of the motion. The teaching sequence is the same as that for the option window **Logic parameters**. (>>> 9.4.3.10 "Teaching the shift in space for logic parameters" Page 213)

9.4.5.2 Stop condition: example and braking characteristics

Example

The indentations are not present by default and have been inserted here for greater clarity.

```
PTP P0 Vel=100 % PDAT1 Tool[1] Base[0]

SPLINE S1 Vel=2 m/s CPDAT1 Tool[1] Base[0]

SPL P1 ADAT1

  STOP WHEN PATH = 50 IF $in[77]==FALSE

  SPL XP1

SPL P2

SPL P3

ENDSPLINE
```

Fig. 9-29: Inline programming example (folds expanded)

Line	Description
4	If the input $IN[77] is FALSE, the robot stops 50 mm after P2 and waits until $IN[77] is TRUE.

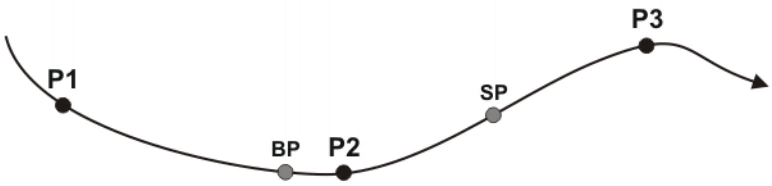


Fig. 9-30: Example of STOP WHEN PATH

Point	Description
BP	**B**raking **P**oint: The robot must start braking here in order to stop at the stop point. From this point onwards, the robot controller monitors whether or not the stop condition is met. The position of **BP** depends on the velocity and the override setting and cannot be identified by the operator.
SP	**S**top **P**oint The distance **P2** → **SP** is 50 mm long.

Braking characteristics

Situation at BP	Behavior of the robot
$IN[77] == FALSE	The robot brakes and stops at **SP**.
$IN[77] == TRUE	The robot does not brake and does not stop at **SP**. The program is executed as if the STOP WHEN PATH statement were not present.

Situation at BP	Behavior of the robot
1. $IN[77] == FALSE at **BP**. 2. The input switches to TRUE between **BP** and **SP**.	1. The robot brakes at **BP**. 2. If the input is TRUE, the robot accelerates again and does not stop at **SP**.
1. $IN[77] == TRUE at **BP**. 2. The input switches to FALSE between **BP** and **SP**.	1. The robot does not brake at **BP**. 2. If the input is FALSE, the robot stops with a path-maintaining EMERGENCY STOP and comes to a standstill at an unpredictable point.

If the stop condition is not met until the robot has already passed **BP**, it is too late to stop at **SP** with a normal braking ramp. In this case, the robot stops with a path-maintaining EMERGENCY STOP and comes to a standstill at an unpredictable point.

- If the EMERGENCY STOP causes the robot to stop after **SP**, the program cannot be resumed until the stop condition is no longer met.
- If the path-maintaining EMERGENCY STOP causes the robot to stop before **SP**, the following occurs when the program is resumed:
  - If the stop condition is no longer met, the robot resumes its motion.
  - If the stop condition is still met, the robot moves as far as **SP** and stops there.

9.4.6 Constant velocity range in the CP spline block

Description

In a CP spline block, a range can be defined in which the robot maintains the programmed velocity constant where possible. This range is called the “constant velocity range”.

- 1 constant velocity range can be defined per CP spline block.
- A constant velocity range is defined by a start statement and an end statement.
- The range cannot extend beyond the spline block.
- There is no lower limit to the size of the range.

If it is not possible to maintain the programmed velocity constant, the robot controller indicates this by means of a message during program execution.

Constant velocity range over several segments:

A constant velocity range can extend over several segments with different programmed velocities. In this case, the lowest of the velocities is valid for the whole range.

Even in the segments with a higher programmed velocity, the motion is executed with the lowest velocity in this case. No message is generated indicating that the velocity has not been maintained. This only occurs if the lowest velocity cannot be maintained.

Programming

A constant velocity range can be programmed in the following ways:

- If programming with KRL syntax: by means of the statement CONST_VEL
- If programming with inline forms:

  The start or end of the range is stored in the corresponding CP segment in the option window **Logic parameters**.

9.4.6.1 Block selection to the constant velocity range

Description

If a block selection to a constant velocity range is carried out, the robot controller ignores it and generates a corresponding message. The motions are executed as if no constant velocity range were programmed.

A block selection to the path section defined by the offset values is considered as a block selection to the constant velocity range. The motion blocks in which the start and end of the range are programmed, however, are irrelevant.

9.4.6.2 Maximum limits

If the start or end point of the spline block is an exact positioning point:

- The constant velocity range starts at the start point at the earliest.
- The constant velocity range ends at the end point at the latest.

If the offset value is such that these limits would be exceeded, the robot controller automatically reduces the offset and generates the following message: *CONST_VEL {Start/End} = {Offset} cannot be implemented; {New offset} will be used.*

The robot controller reduces the offset far enough to create a range in which the constant programmed velocity can be maintained. In other words, it does not necessarily shift the limit exactly to the start or end point of the spline block, but possibly further inwards.

The same message is generated if the range is already in the spline block beforehand, but the defined velocity cannot be maintained due to the offset. In this case, once again, the robot controller reduces the offset.

If the start or end point of the spline block is approximated:

- The constant velocity range starts at the start of the approximate positioning arc of the start point at the earliest.
- The constant velocity range ends at the start of the approximate positioning arc of the end point at the latest.

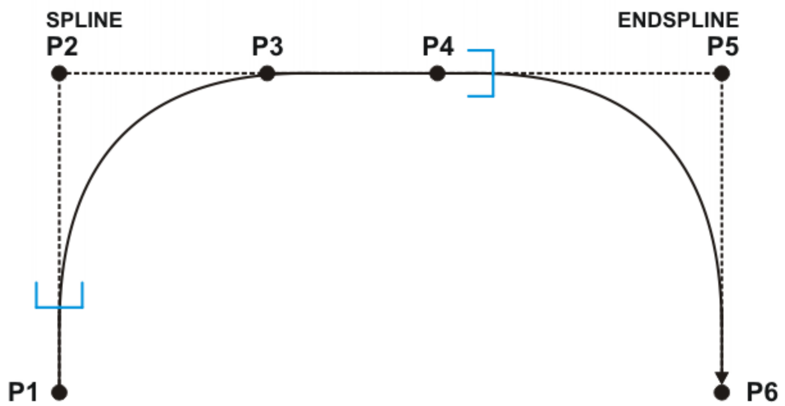


Fig. 9-31: Maximum limits for approximated SPLINE/ENDSPLINE

If the offset is such that these limits would be exceeded, the robot controller automatically sets the limit to the start of the corresponding approximate positioning arc. It does not generate a message.

9.5 Modifying motion parameters

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line containing the instruction that is to be changed.
2. Press **Change**. The inline form for this instruction is opened.
3. Modify parameters.
4. Save changes by pressing **Cmd Ok**.

9.6 Re-teaching a point

Description

The coordinates of a taught point can be modified. This is done by moving to the new position and overwriting the old point with the new position.

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Move the TCP to the desired position.
2. Position the cursor in the line containing the motion instruction that is to be changed.
3. Press **Change**. The inline form for this instruction is opened.
4. For PTP and LIN motions: Press **Touch Up** to accept the current position of the TCP as the new end point.

   For CIRC motions:

   - Press **Teach Aux** to accept the current position of the TCP as the new auxiliary point.
   - Or press **Teach End** to accept the current position of the TCP as the new end point.
5. Confirm the request for confirmation with **Yes**.
6. Save change by pressing **Cmd Ok**.

9.7 Programming logic instructions

9.7.1 Inputs/outputs

Digital inputs/outputs

The robot controller can manage up to 8192 digital inputs and 8192 digital outputs. 4096 inputs/outputs are available by default.

Analog inputs/outputs

The robot controller can manage 32 analog inputs and 32 analog outputs.

The inputs/outputs are managed via the following system variables:

	Inputs	Outputs
Digital	$IN[1] … $IN[8192]	$OUT[1] … $OUT[8192]
Analog	$ANIN[1] … $ANIN[32]	$ANOUT[1] … $ANOUT[32]

$ANIN[…] indicates the input voltage, adapted to the range between -1.0 and +1.0. The actual voltage depends on the settings of the analog module.

$ANOUT[…] can be used to set an analog voltage. $ANOUT[…] can have values from -1.0 to +1.0 written to it. The voltage actually generated depends on the settings of the analog module. If an attempt is made to set voltages outside the range of values, the robot controller displays the following message: *Limit {Signal name}*

9.7.2 Setting a digital output - OUT

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the logic instruction is to be inserted.
2. Select the menu sequence **Commands** > **Logic** > **OUT** > **OUT**.
3. Set the parameters in the inline form.
   (>>> 9.7.3 "Inline form “OUT”" Page 226)
4. Save instruction with **Cmd Ok**.

9.7.3 Inline form “OUT”

The instruction sets a digital output.

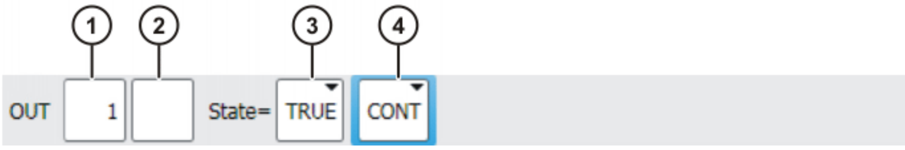


Fig. 9-32: Inline form “OUT”

Item	Description
1	Number of the output
2	If a name exists for the output, this name is displayed. User group “Expert” or higher: A name can be entered by pressing **Long text**. The name is freely selectable.
3	State to which the output is switched ■ **TRUE** ■ **FALSE**
4	■ **CONT**: Execution in the advance run ■ **[Empty box]**: Execution with advance run stop

9.7.4 Setting a pulse output - PULSE

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the logic instruction is to be inserted.
2. Select the menu sequence **Commands** > **Logic** > **OUT** > **PULSE**.
3. Set the parameters in the inline form.
   (>>> 9.7.5 "Inline form “PULSE”" Page 226)
4. Save instruction with **Cmd Ok**.

9.7.5 Inline form “PULSE”

The instruction sets a pulse of a defined length.

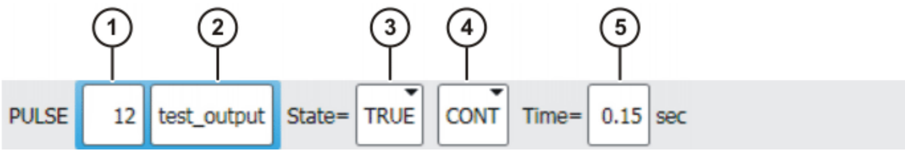


Fig. 9-33: Inline form “PULSE”

Item	Description
1	Number of the output
2	If a name exists for the output, this name is displayed. User group “Expert” or higher: A name can be entered by pressing **Long text**. The name is freely selectable.
3	State to which the output is switched ■ **TRUE**: “High” level ■ **FALSE**: “Low” level
4	■ **CONT**: Execution in the advance run ■ **[Empty box]**: Execution with advance run stop
5	Length of the pulse ■ **0.10 … 3.00 s**

9.7.6 Setting an analog output - ANOUT

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the instruction is to be inserted.
2. Select **Commands** > **Analog output** > **Static** or **Dynamic**.
3. Set the parameters in the inline form.

   (>>> 9.7.7 "Inline form “ANOUT” (static)" Page 227)

   (>>> 9.7.8 "Inline form “ANOUT” (dynamic)" Page 228)
4. Save instruction with **Cmd Ok**.

9.7.7 Inline form “ANOUT” (static)

This instruction sets a static analog output. The voltage is set to a fixed level by means of a factor. The actual voltage level depends on the analog module used. For example, a 10 V module with a factor of 0.5 provides a voltage of 5 V.

ANOUT triggers an advance run stop.

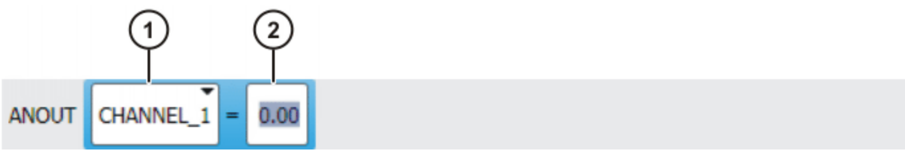


Fig. 9-34: Inline form “ANOUT” (static)

Item	Description
1	Number of the analog output ■ **CHANNEL_1 … CHANNEL_32**
2	Factor for the voltage ■ **0 … 1** (intervals: 0.01)

9.7.8 Inline form “ANOUT” (dynamic)

This instruction activates or deactivates a dynamic analog output.

A maximum of 4 dynamic analog outputs can be activated at any one time. ANOUT triggers an advance run stop.

The voltage is determined by a factor. The actual voltage level depends on the following values:

- Velocity or function generator

  For example, a velocity of 1 m/s with a factor of 0.5 results in a voltage of 5 V.
- Offset

  For example, an offset of +0.15 for a voltage of 0.5 V results in a voltage of 6.5 V.

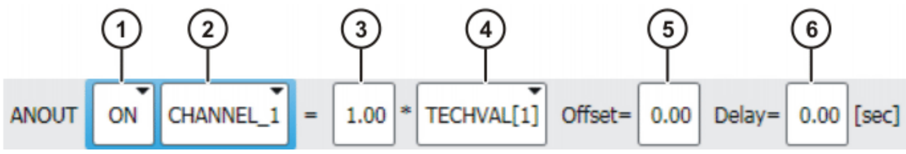


Fig. 9-35: Inline form “ANOUT” (dynamic)

Item	Description
1	Activation or deactivation of the analog output ■ **ON** ■ **OFF**
2	Number of the analog output ■ **CHANNEL_1 … CHANNEL_32**
3	Factor for the voltage ■ **0 … 10** (intervals: 0.01)
4	■ **VEL_ACT**: The voltage is dependent on the velocity. ■ **TECHVAL[1] … TECHVAL[6]**: The voltage is controlled by a function generator.
5	Value by which the voltage is increased or decreased ■ **-1 … +1** (intervals: 0.01)
6	Time by which the output signal is delayed (+) or brought forward (-) ■ **-0.2 … +0.5 s**

9.7.9 Programming a wait time - WAIT

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the logic instruction is to be inserted.
2. Select the menu sequence **Commands** > **Logic** > **WAIT**.
3. Set the parameters in the inline form.
   (>>> 9.7.10 "Inline form “WAIT”" Page 229)
4. Save instruction with **Cmd Ok**.

9.7.10 Inline form “WAIT”

WAIT can be used to program a wait time. The robot motion is stopped for a programmed time. WAIT always triggers an advance run stop.

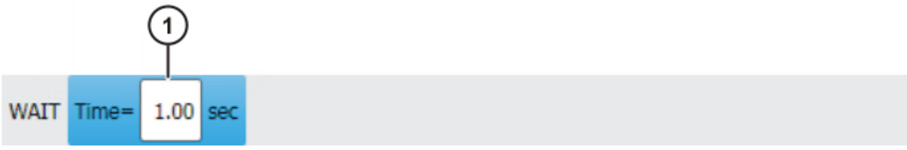


Fig. 9-36: Inline form “WAIT”

Item	Description
1	Wait time ■ **≥ 0** s

9.7.11 Programming a signal-dependent wait function - WAITFOR

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the logic instruction is to be inserted.
2. Select the menu sequence **Commands** > **Logic** > **WAITFOR**.
3. Set the parameters in the inline form.
   (>>> 9.7.12 "Inline form “WAITFOR”" Page 229)
4. Save instruction with **Cmd Ok**.

9.7.12 Inline form “WAITFOR”

The instruction sets a signal-dependent wait function.

If required, several signals (maximum 12) can be linked. If a logic operation is added, boxes are displayed in the inline form for the additional signals and links.

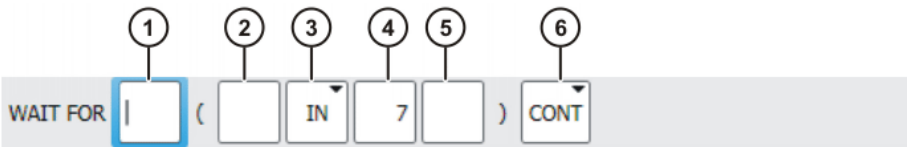


Fig. 9-37: Inline form “WAITFOR”

Item	Description
1	Add external logic operation. The operator is situated between the bracketed expressions. ■ **AND** ■ **OR** ■ **EXOR** Add NOT. ■ **NOT** ■ **[blank]** Enter the desired operator by means of the corresponding button.
2	Add internal logic operation. The operator is situated inside a bracketed expression. ■ **AND** ■ **OR** ■ **EXOR** Add NOT. ■ **NOT** ■ **[blank]** Enter the desired operator by means of the corresponding button.
3	Signal for which the system is waiting ■ **IN** ■ **OUT** ■ **CYCFLAG** ■ **TIMER** ■ **FLAG**
4	Number of the signal
5	If a name exists for the signal, this name is displayed. User group “Expert” or higher: A name can be entered by pressing **Long text**. The name is freely selectable.
6	■ **CONT**: Execution in the advance run ■ **[Empty box]**: Execution with advance run stop

9.7.13 Switching on the path - SYN OUT

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the logic instruction is to be inserted.
2. Select the menu sequence **Commands** > **Logic** > **OUT** > **SYN OUT**.
3. Set the parameters in the inline form.

   (>>> 9.7.14 "Inline form “SYN OUT”, option “START/END”" Page 231)

   (>>> 9.7.15 "Inline form “SYN OUT”, option “PATH”" Page 233)
4. Save instruction with **Cmd Ok**.

9.7.14 Inline form “SYN OUT”, option “START/END”

The switching action can be triggered relative to the start or end point of the motion. The switching action can be delayed or brought forward. The motion can be a LIN, CIRC or PTP motion.

Possible applications include:

- Closing or opening the weld gun during spot welding
- Switching the welding current on/off during arc welding
- Starting or stopping the flow of adhesive in bonding or sealing applications.

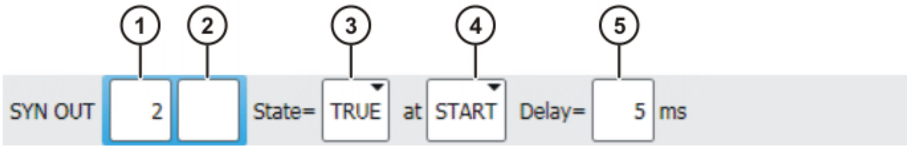


Fig. 9-38: Inline form “SYN OUT”, option “START/END”

Item	Description
1	Number of the output
2	If a name exists for the output, this name is displayed. User group “Expert” or higher: A name can be entered by pressing **Long text**. The name is freely selectable.
3	State to which the output is switched - **TRUE** - **FALSE**
4	Point to which SYN OUT refers: - **START**: Start point of the motion - **END**: End point of the motion
5	Switching action delay - **-1,000 … +1,000 ms** **Note:** The time specification is absolute. In other words, the switching point varies according to the velocity of the robot.

Example 1

Start point and end point are exact positioning points.

```
LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
SYN OUT 1 '' State= TRUE at START Delay=20ms
SYN OUT 2 '' State= TRUE at END Delay=-20ms
LIN P3 VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
```

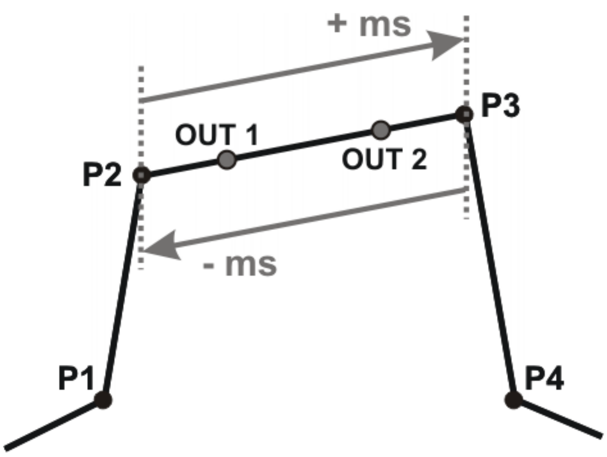


Fig. 9-39

OUT 1 and OUT 2 specify approximate positions at which switching is to occur. The dotted lines indicate the switching limits.

Switching limits:

- START: The switching point can be delayed, at most, as far as exact positioning point P3 (+ ms).
- END: The switching point can be brought forward, at most, as far as exact positioning point P2 (- ms).

If greater values are specified for the delay, the controller automatically switches at the switching limit.

Example 2

Start point is exact positioning point, end point is approximated.

```
LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
SYN OUT 1 '' State= TRUE at START Delay=20ms
SYN OUT 2 '' State= TRUE at END Delay=-20ms
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
```

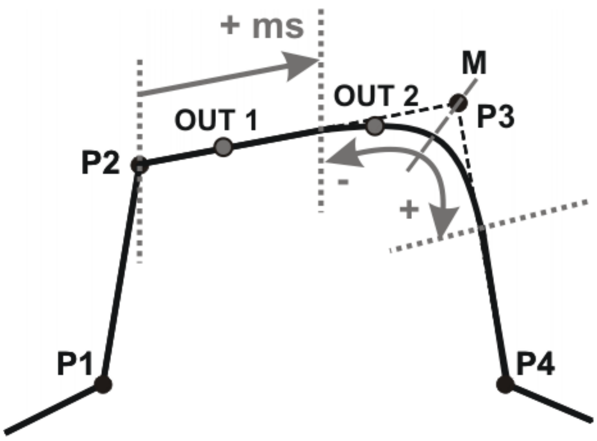


Fig. 9-40

OUT 1 and OUT 2 specify approximate positions at which switching is to occur. The dotted lines indicate the switching limits. M = middle of the approximate positioning range.

Switching limits:

- START: The switching point can be delayed, at most, as far as the start of the approximate positioning range of P3 (+ ms).

- END: The switching point can be brought forward, at most, as far as the start of the approximate positioning range of P3 (-).

  The switching point can be delayed, at most, as far as the end of the approximate positioning range of P3 (+).

If greater values are specified for the delay, the controller automatically switches at the switching limit.

Example 3

Start point and end point are approximated

```
LIN P1 VEL=0.3m/s CPDAT1
LIN P2 CONT VEL=0.3m/s CPDAT2
SYN OUT 1 '' State= TRUE at START Delay=20ms
SYN OUT 2 '' State= TRUE at END Delay=-20ms
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
```

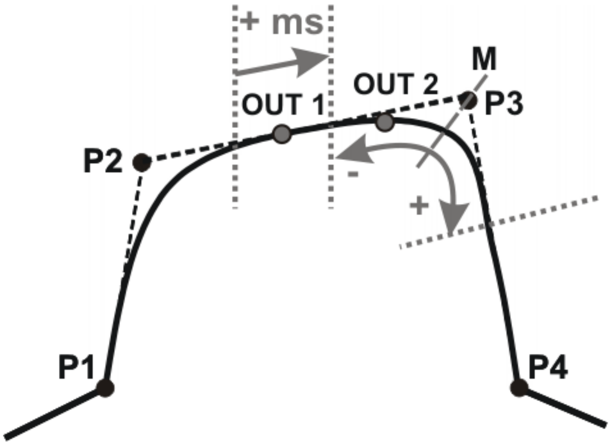


Fig. 9-41

OUT 1 and OUT 2 specify approximate positions at which switching is to occur. The dotted lines indicate the switching limits. M = middle of the approximate positioning range.

Switching limits:

- START: The switching point can be situated, at the earliest, at the end of the approximate positioning range of P2.

  The switching point can be delayed, at most, as far as the start of the approximate positioning range of P3 (+ ms).
- END: The switching point can be brought forward, at most, as far as the start of the approximate positioning range of P3 (-).

  The switching point can be delayed, at most, as far as the end of the approximate positioning range of P3 (+).

If greater values are specified for the delay, the controller automatically switches at the switching limit.

9.7.15 Inline form “SYN OUT”, option “PATH”

The switching action refers to the end point of the motion. The switching action can be shifted in space and delayed or brought forward. The motion can be a LIN or CIRC motion. It must not be a PTP motion.

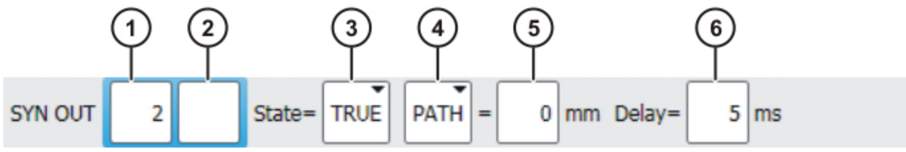


Fig. 9-42: Inline form “SYN OUT”, option “PATH”

Item	Description
1	Number of the output
2	If a name exists for the output, this name is displayed. User group “Expert” or higher: A name can be entered by pressing **Long text**. The name is freely selectable.
3	State to which the output is switched ■ **TRUE** ■ **FALSE**
4	■ **PATH**: SYN OUT refers to the end point of the motion.
5	This box is only displayed if **PATH** has been selected. Distance from the switching point to the end point ■ **-2,000 … +2,000 mm**
6	Switching action delay ■ **-1,000 … +1,000 ms** **Note:** The time specification is absolute. In other words, the switching point varies according to the velocity of the robot.

Example 1

Start point is exact positioning point, end point is approximated.

```
LIN P1 VEL=0.3m/s CPDAT1
SYN OUT 1 '' State= TRUE at START PATH=20mm Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
```

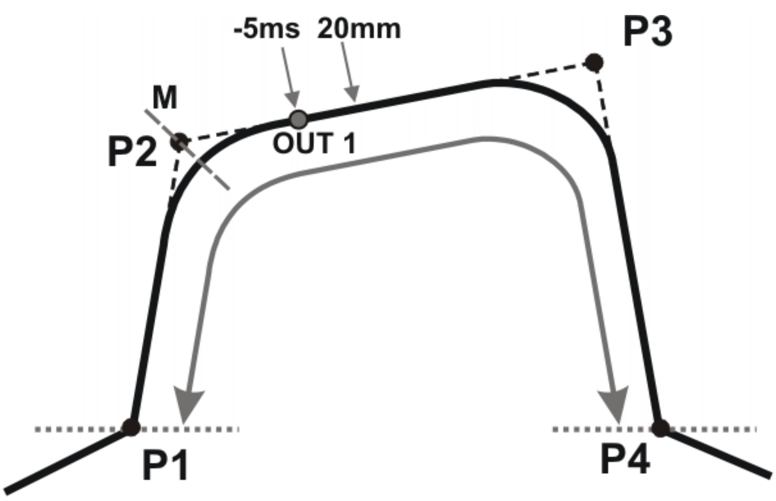


Fig. 9-43

OUT 1 specifies the approximate position at which switching is to occur. The dotted lines indicate the switching limits. M = middle of the approximate positioning range.

Switching limits:

- The switching point can be brought forward, at most, as far as exact positioning point P1.
- The switching point can be delayed, at most, as far as the next exact positioning point P4. If P3 was an exact positioning point, the switching point could be delayed, at most, as far as P3.

If greater values are specified for the shift in space or time, the controller automatically switches at the switching limit.

Example 2

Start point and end point are approximated

```
LIN P1 CONT VEL=0.3m/s CPDAT1
SYN OUT 1 '' State= TRUE at START PATH=20mm Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
```

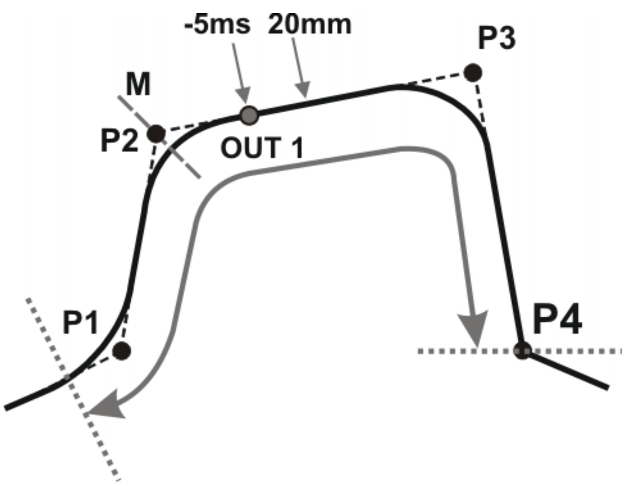


Fig. 9-44

OUT 1 specifies the approximate position at which switching is to occur. The dotted lines indicate the switching limits. M = middle of the approximate positioning range.

Switching limits:

- The switching point can be brought forward, at most, as far as the start of the approximate positioning range of P1.
- The switching point can be delayed, at most, as far as the next exact positioning point P4. If P3 was an exact positioning point, the switching point could be delayed, at most, as far as P3.

If greater values are specified for the shift in space or time, the controller automatically switches at the switching limit.

9.7.16 Setting a pulse on the path - SYN PULSE

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line after which the logic instruction is to be inserted.
2. Select the menu sequence **Commands** > **Logic** > **OUT** > **SYN PULSE**.
3. Set the parameters in the inline form.

   (>>> 9.7.17 "Inline form “SYN PULSE”" Page 236)

4. Save instruction with **Cmd Ok**.

9.7.17 Inline form “SYN PULSE”

SYN PULSE can be used to trigger a pulse at the start or end point of the motion. The pulse can be shifted in time and/or space, i.e. it does not have to be triggered exactly at the point, but can also be triggered before or after it.

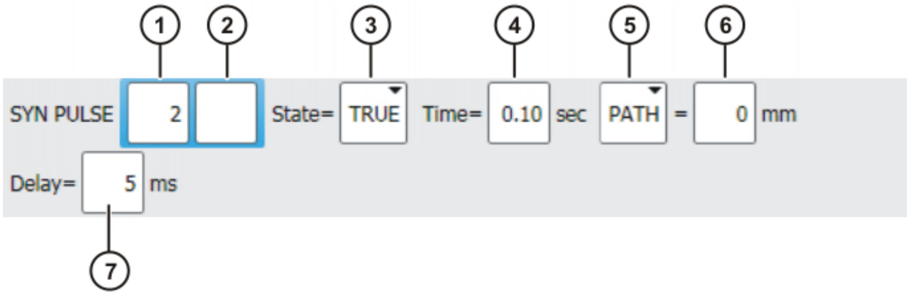


Fig. 9-45: Inline form “SYN PULSE”

Item	Description
1	Number of the output
2	If a name exists for the output, this name is displayed. User group “Expert” or higher: A name can be entered by pressing **Long text**. The name is freely selectable.
3	State to which the output is switched ■ **TRUE** ■ **FALSE**
4	Duration of the pulse ■ **0.1 … 3 s**
5	Point to which SYN PULSE refers: ■ **START**: Start point of the motion ■ **END**: End point of the motion See SYN OUT for examples and switching limits. (>>> 9.7.14 "Inline form “SYN OUT”, option “START/END”" Page 231) ■ **PATH**: SYN PULSE refers to the end point. An offset in space is also possible. See SYN OUT for examples and switching limits. (>>> 9.7.15 "Inline form “SYN OUT”, option “PATH”" Page 233)
6	Distance from the switching point to the end point ■ **-2,000 … +2,000 mm** This box is only displayed if **PATH** has been selected.
7	Switching action delay ■ **-1,000 … +1,000 ms** **Note:** The time specification is absolute. The switching point varies according to the velocity of the robot.

9.7.18 Modifying a logic instruction

Precondition

- A program is selected.
- Operating mode T1

Procedure

1. Position the cursor in the line containing the instruction that is to be changed.
2. Press **Change**. The inline form for this instruction is opened.
3. Change the parameters.
4. Save changes by pressing **Cmd Ok**.

10 Diagnosis

10.1 Logbook

10.1.1 Displaying the logbook

The operator actions on the smartPAD are automatically logged.

Procedure

- In the main menu, select **Diagnosis** > **Logbook** > **Display**.

The following tabs are available:

- Log (>>> 10.1.2 "“Log” tab" Page 239)
- Filter (>>> 10.1.3 "Filter tab" Page 240)

10.1.2 “Log” tab

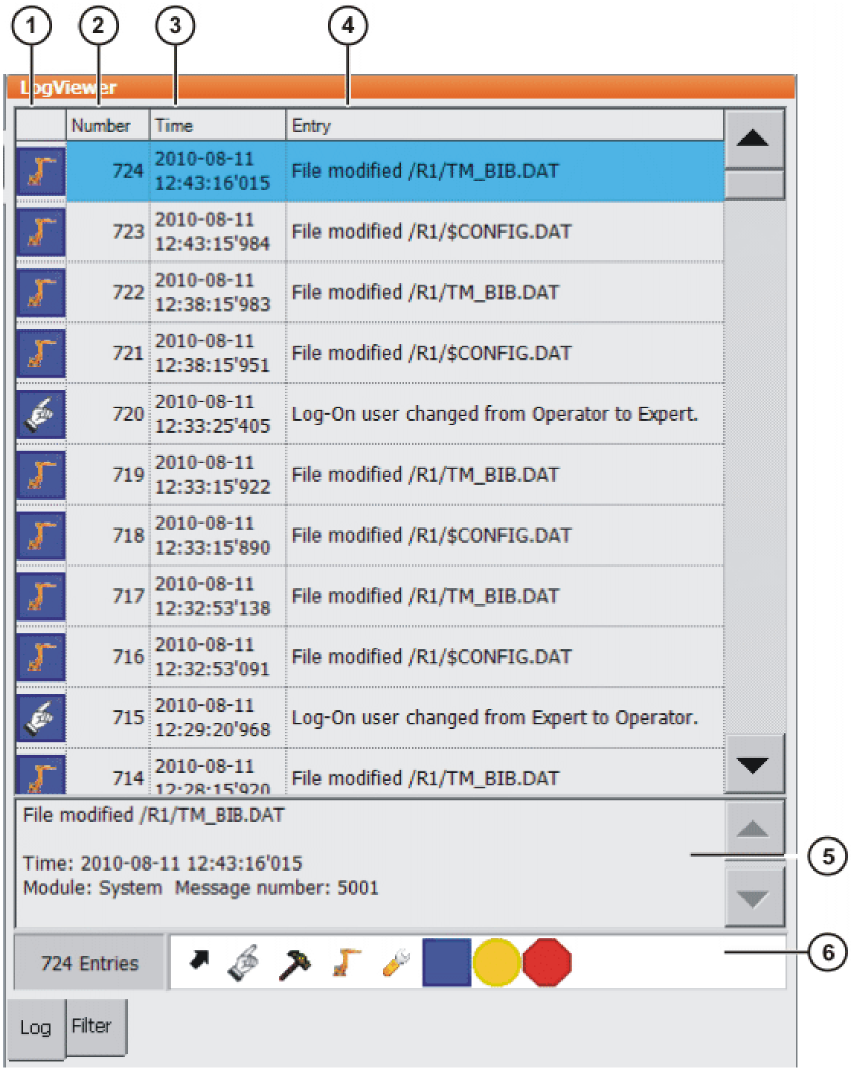
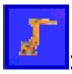


Fig. 10-1: Logbook, Log tab

Item	Description
1	Type of log event Example : Filter type "Information" + filter class "System" = information originated by the kernel system of the robot. The individual filter types and filter classes are listed on the **Filter** tab.
2	Log event number
3	Date and time of the log event
4	Brief description of the log event
5	Detailed description of the selected log event
6	Indication of the active filter

The following buttons are available:

Button	Description
Export	Exports the log data as a text file.
Update	Refreshes the log display.

10.1.3 Filter tab

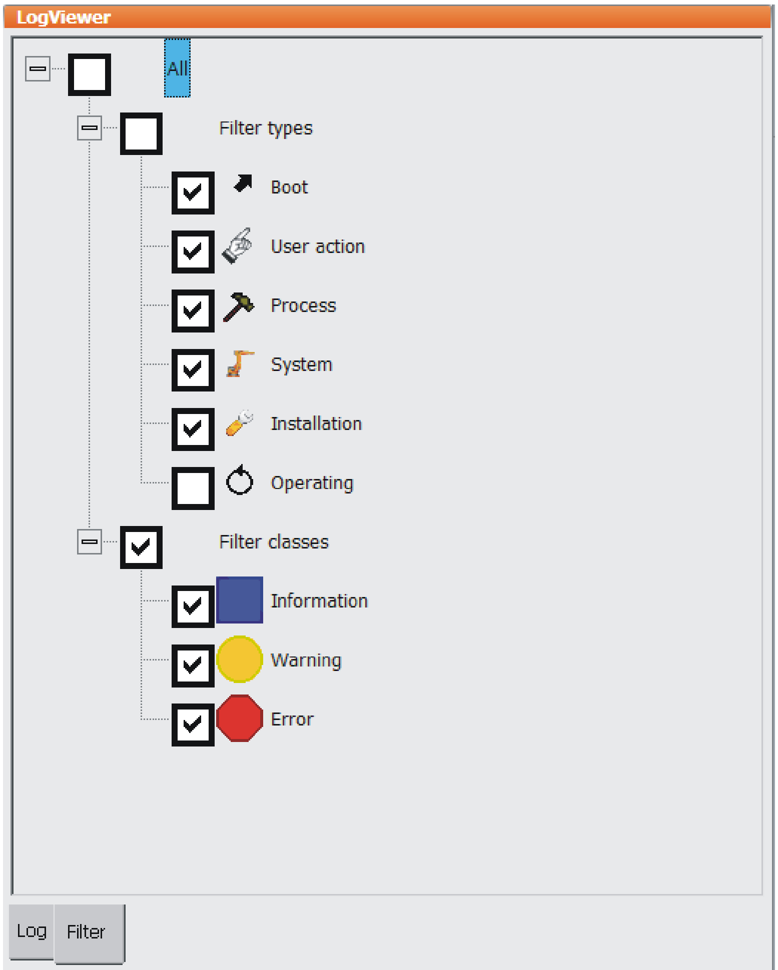


Fig. 10-2: Logbook, Filter tab

10.2 Displaying diagnostic data about the kernel system

Description

The menu item **Diagnostic monitor** makes it possible to display a wide range of diagnostic data concerning numerous software sub-areas of the kernel system.

Examples:

- Area **Kcp3 driver** (= driver for the smartPAD)
- Network driver

The data displayed depend on the selected area. The display includes states, fault counters, message counters, etc.

Procedure

1. In the main menu, select **Diagnosis** > **Diagnostic monitor**.
2. Select an area in the **Module** box.

   Diagnostic data are displayed for the selected area.

10.3 Automatically compressing data for error analysis (KRCDiag)

Description

If it is necessary for an error to be analyzed by KUKA Deutschland GmbH, this procedure can be used to compress the required data. The procedure generates a ZIP file in the directory C:\KUKA\KRCDiag. This contains the data which KUKA Deutschland GmbH requires to analyze an error. This includes information about the system resources, screenshots and much more.

Preparation

A screenshot of the current view of the smartHMI is automatically generated for the data packet.

- Therefore, if possible, display the information related to errors on the smartHMI before starting the procedure:

  e.g. expand the message window or display the logbook. What information is useful here depends on the specific circumstances.

Procedure via “Diagnosis”

- In the main menu, select **Diagnosis** > **KrcDiag**.

  The data are compressed. Progress is displayed in a window. Once the operation has been completed, this is also indicated in the window. The window is then automatically hidden again.

Procedure via smartPAD

This procedure uses keys on the smartPAD instead of menu items. It can thus also be used if the SmartHMI is not available, due to Windows problems for example.

Precondition:

- The smartPAD is connected to the robot controller.
- The robot controller is switched on.

> The keys must be pressed within 2 seconds. Whether or not the main menu and keypad are displayed in the smartHMI is irrelevant.

1. Press the “Main menu” key and hold it down.
2. Press the keypad key twice.
3. Release the “Main menu” key.

   The data are compressed. Progress is displayed in a window. Once the operation has been completed, this is also indicated in the window. The window is then automatically hidden again.

Procedure via “Archive”

Alternatively, the data can also be compressed via **File** > **Archive** > **[...]**. In this way, the data can be stored on a USB stick or network path.

(>>> 6.10 "Archiving and restoring data" Page 154)

11 KUKA Service

11.1 Requesting support

Introduction

This documentation provides information on operation and operator control, and provides assistance with troubleshooting. For further assistance, please contact your local KUKA subsidiary.

Information

The following information is required for processing a support request:

- Description of the problem, including information about the duration and frequency of the fault
- As comprehensive information as possible about the hardware and software components of the overall system

  The following list gives an indication of the information which is relevant in many cases:

  - Model and serial number of the kinematic system, e.g. the manipulator
  - Model and serial number of the controller
  - Model and serial number of the energy supply system
  - Designation and version of the system software
  - Designations and versions of other software components or modifications
  - Diagnostic package KRCDiag

    Additionally for KUKA Sunrise: Existing projects including applications

    For versions of KUKA System Software older than V8: Archive of the software (KRCDiag is not yet available here.)
  - Application used
  - External axes used

11.2 KUKA Customer Support

Availability

KUKA Customer Support is available in many countries. Please do not hesitate to contact us if you have any questions.

Argentina

Ruben Costantini S.A. (Agency)
Luis Angel Huergo 13 20
Parque Industrial
2400 San Francisco (CBA)
Argentina
Tel. +54 3564 421033
Fax +54 3564 428877
ventas@costantini-sa.com

Australia

KUKA Robotics Australia Pty Ltd
45 Fennell Street
Port Melbourne VIC 3207
Australia
Tel. +61 3 9939 9656
info@kuka-robotics.com.au
www.kuka-robotics.com.au

Belgium

KUKA Automatisering + Robots N.V.
Centrum Zuid 1031
3530 Houthalen
Belgium
Tel. +32 11 516160
Fax +32 11 526794
info@kuka.be
www.kuka.be

Brazil

KUKA Roboter do Brasil Ltda.
Travessa Claudio Armando, nº 171
Bloco 5 - Galpões 51/52
Bairro Assunção
CEP 09861-7630 São Bernardo do Campo - SP
Brazil
Tel. +55 11 4942-8299
Fax +55 11 2201-7883
info@kuka-roboter.com.br
www.kuka-roboter.com.br

Chile

Robotec S.A. (Agency)
Santiago de Chile
Chile
Tel. +56 2 331-5951
Fax +56 2 331-5952
robotec@robotec.cl
www.robotec.cl

China

KUKA Robotics China Co., Ltd.
No. 889 Kungang Road
Xiaokunshan Town
Songjiang District
201614 Shanghai
P. R. China
Tel. +86 21 5707 2688
Fax +86 21 5707 2603
info@kuka-robotics.cn
www.kuka-robotics.com

Germany

KUKA Deutschland GmbH
Zugspitzstr. 140
86165 Augsburg
Germany
Tel. +49 821 797-1926
Fax +49 821 797-41 1926
Hotline.robotics.de@kuka.com
www.kuka.com

France	KUKA Automatisme + Robotique SAS Techvallée 6, Avenue du Parc 91140 Villebon S/Yvette France Tel. +33 1 6931660-0 Fax +33 1 6931660-1 commercial@kuka.fr www.kuka.fr
India	KUKA Robotics India Pvt. Ltd. Office Number-7, German Centre, Level 12, Building No. - 9B DLF Cyber City Phase III 122 002 Gurgaon Haryana India Tel. +91 124 4635774 Fax +91 124 4635773 info@kuka.in www.kuka.in
Italy	KUKA Roboter Italia S.p.A. Via Pavia 9/a - int.6 10098 Rivoli (TO) Italy Tel. +39 011 959-5013 Fax +39 011 959-5141 kuka@kuka.it www.kuka.it
Japan	KUKA Japan K.K. YBP Technical Center 134 Godo-cho, Hodogaya-ku Yokohama, Kanagawa 240 0005 Japan Tel. +81 45 744 7531 Fax +81 45 744 7541 info@kuka.co.jp
Canada	KUKA Robotics Canada Ltd. 6710 Maritz Drive - Unit 4 Mississauga L5W 0A1 Ontario Canada Tel. +1 905 670-8600 Fax +1 905 670-8604 info@kukarobotics.com www.kuka-robotics.com/canada

Korea	KUKA Robotics Korea Co. Ltd. RIT Center 306, Gyeonggi Technopark 1271-11 Sa 3-dong, Sangnok-gu Ansan City, Gyeonggi Do 426-901 Korea Tel. +82 31 501-1451 Fax +82 31 501-1461 info@kukakorea.com
Malaysia	KUKA Robot Automation (M) Sdn Bhd South East Asia Regional Office No. 7, Jalan TPP 6/6 Taman Perindustrian Puchong 47100 Puchong Selangor Malaysia Tel. +60 (03) 8063-1792 Fax +60 (03) 8060-7386 info@kuka.com.my
Mexico	KUKA de México S. de R.L. de C.V. Progreso #8 Col. Centro Industrial Puente de Vigas Tlalnepantla de Baz 54020 Estado de México Mexico Tel. +52 55 5203-8407 Fax +52 55 5203-8148 info@kuka.com.mx www.kuka-robotics.com/mexico
Norway	KUKA Sveiseanlegg + Roboter Sentrumsvegen 5 2867 Hov Norway Tel. +47 61 18 91 30 Fax +47 61 18 62 00 info@kuka.no
Austria	KUKA Roboter CEE GmbH Gruberstraße 2-4 4020 Linz Austria Tel. +43 7 32 78 47 52 Fax +43 7 32 79 38 80 office@kuka-roboter.at www.kuka.at

Poland	KUKA Roboter CEE GmbH Poland Spółka z ograniczoną odpowiedzialnością Oddział w Polsce Ul. Porcelanowa 10 40-246 Katowice Poland Tel. +48 327 30 32 13 or -14 Fax +48 327 30 32 26 ServicePL@kuka-roboter.de
Portugal	KUKA Robots IBÉRICA, S.A. Rua do Alto da Guerra n° 50 Armazém 04 2910 011 Setúbal Portugal Tel. +351 265 729 780 Fax +351 265 729 782 info.portugal@kukapt.com www.kuka.com
Russia	KUKA Russia OOO 1-y Nagatinskiy pr-d, 2 117105 Moskau Russia Tel. +7 495 665-6241 support.robotics.ru@kuka.com
Sweden	KUKA Svetsanläggningar + Robotar AB A. Odhners gata 15 421 30 Västra Frölunda Sweden Tel. +46 31 7266-200 Fax +46 31 7266-201 info@kuka.se
Switzerland	KUKA Roboter Schweiz AG Industriestr. 9 5432 Neuenhof Switzerland Tel. +41 44 74490-90 Fax +41 44 74490-91 info@kuka-roboter.ch www.kuka-roboter.ch
Slovakia	KUKA Roboter CEE GmbH organizačná zložka Bojnická 3 831 04 Bratislava Slovakia Tel. +420 226 212 273 support.robotics.cz@kuka.com

Spain	KUKA Iberia, S.A.U. Pol. Industrial Torrent de la Pastera Carrer del Bages s/n 08800 Vilanova i la Geltrú (Barcelona) Spain Tel. +34 93 8142-353 comercial@kukarob.es
South Africa	Jendamark Automation LTD (Agency) 76a York Road North End 6000 Port Elizabeth South Africa Tel. +27 41 391 4700 Fax +27 41 373 3869 www.jendamark.co.za
Taiwan	KUKA Automation Taiwan Co. Ltd. 1F, No. 298 Yangguang ST., Nei Hu Dist., Taipei City, Taiwan 114 Taiwan Tel. +886 2 8978 1188 Fax +886 2 8797 5118 info@kuka.com.tw
Thailand	KUKA (Thailand) Co. Ltd. No 22/11-12 H-Cape Biz Sector Onnut Sukhaphiban 2 road, Prawet Bangkok 10250 Thailand Tel. +66 (0) 90-940-8950 HelpdeskTH@kuka.com
Czech Republic	KUKA Roboter CEE GmbH organizační složka Pražská 239 25066 Zdiby Czech Republic Tel. +420 226 212 273 support.robotics.cz@kuka.com
Hungary	KUKA Robotics Hungaria Kft. Fö út 140 2335 Taksony Hungary Tel. +36 24 501609 Fax +36 24 477031 info@kuka-robotics.hu

USA	KUKA Robotics Corporation 51870 Shelby Parkway Shelby Township 48315-1787 Michigan USA Tel. +1 866 873-5852 Fax +1 866 329-5852 info@kukarobotics.com www.kukarobotics.com
UK	KUKA Robotics UK Ltd Great Western Street Wednesbury West Midlands WS10 7LL UK Tel. +44 121 505 9970 Fax +44 121 505 6589 service@kuka-robotics.co.uk www.kuka-robotics.co.uk

Index