ROBOTICS

Application manual

Functional safety and SafeMove

Trace back information:
Workspace 24D version a7
Checked in 2024-12-18
Skribenta version 5.6.018

Application manual

Functional safety and SafeMove

RobotWare 7.17

Document ID: 3HAC066559-001

Revision: R

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damage to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

Table of contents

This page is intentionally left blank

Overview of this manual

About this manual

This manual describes the safety module, the functional safety options, and the second generation of SafeMove. It contains descriptions of the functionality, and how to configure that functionality. It also describes user interfaces and recommendations on how to use the safety module.

> **Note**
>
> It is the responsibility of the integrator to provide safety and user guides for the robot system.

> **Note**
>
> Screenshots in this manual are generally intended to show a language version corresponding to the language of the manual. In some cases, a translated manual still uses English screenshots if the localized user interface was not available at the time of publishing the manual.

Usage

This manual should be used during installation and configuration of the safety module, SafeMove, and the functional safety options.

> **Note**
>
> Before any work on or with the robot is performed, the safety information in the product manual for the controller and manipulator must be read.

Who should read this manual?

This manual is mainly intended for:

- personnel that are responsible for installations and configurations of hardware/software
- personnel that make configurations of the I/O system
- system integrators

Prerequisites

The reader should have the required knowledge of:

- mechanical installation work
- electrical installation work
- working with industrial robots
- using RobotStudio
- personal safety

Continues on next page

Continued

References

Reference	Document ID
Application manual - Additional axes	*3HAC082287-001*
Application manual - EtherCAT	*3HAC090257-001*
Application manual - EtherNet/IP Scanner/Adapter	*3HAC066565-001*
Application manual - PROFINET Controller/Device	*3HAC066558-001*
Application manual - I/O Engineering	*3HAC082346-001*
Application manual - Scalable I/O	*3HAC070208-001*
Operating manual - RobotStudio	*3HAC032104-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC065038-001*
Technical reference manual - System parameters	*3HAC065041-001*
Product manual - OmniCore C30	*3HAC060860-001*
Product manual - OmniCore C90XT	*3HAC073706-001*
Product manual - OmniCore V250XT Type B	*3HAC087112-001*
Product manual - OmniCore V400XT	*3HAC081697-001*
Safety manual for robot - Manipulator and IRC5 or OmniCore controller	*3HAC031045-001*

Tip

All documents can be found via myABB Business Portal, *www.abb.com/myABB*.

Revisions

Revision	Description
A	Released with RobotWare 7.0.2.
B	Released with RobotWare 7.1. • Stand Still Supervision is now possible in manual mode, see *Configuring Stand Still Supervision on page 161*. • The service mode is updated, see *The Mode tab on page 110*. • Changes regarding Basic joint supervision made in section *Limitations on page 27*, *Overview of SafeMove functions on page 33*, *Manual Operation Supervision on page 36*, *Safe Brake Ramp on page 45* and *The Visual SafeMove browser on page 83*. • Added support for PROFISafe Device.
C	Released with RobotWare 7.2. • Added sections *Tool Force Supervision (TFO) on page 59* and *Human Contact Supervision on page 47*. • Added information about the SafeMove configuration application on FlexPendant, see *The SafeMove configurator app on FlexPendant on page 113*.

Continues on next page

Revision	Description
D	Released with RobotWare 7.2.2 and RobotWare 7.3. • Added guidelines for *Human Contact Supervision*, available for the GoFa robot. • Updated information about SafeMove configurator app on FlexPendant. • Updated the overview of the SafeMove options, see *Functional safety options on page 18*.
E	Released with RobotWare 7.3. • Added *Guidelines for the option Extended working range on page 230*.
F	Released with RobotWare 7.4. • Added information which robot controllers are supported, *Supported robot controllers on page 27*. • Added notes in section *Tool Force Supervision (TFO) on page 59*. • Updated some sections regarding *I/O Engineering Tool*. • Added updates to *The Visual SafeMove user interface in RobotStudio* • Removed the reference section *Connection of external emergency stop*, as this is not specific for the functional safety options. • Added information about the *SafeMove Assistant* functionality. • The title of this document is changed to *Application manual - Functional safety and SafeMove*.
G	Released with RobotWare 7.5. • Added information for commissioning mode. • Added more functionality for the SafeMove configurator app.
H	Released with RobotWare 7.6. • The functionality *SafeMove Assistant* is only active in automatic mode. • Updated information for the SafeMove function **Human Contact Supervision**. • Updated information for the SafeMove function Tool Force Supervision, see *Limitations on page 61*. • The signal names for safety stop configuration are corrected, see *The Stop Configuration button on page 78*. • Added information about response time to external stops, see *Reaction time on page 185*. • Added information about gravity parameters for GoFa. • Information about online user guide added in section *The SafeMove configurator app on FlexPendant on page 113*.
J	Released with RobotWare 7.7. • Added more functionality for the SafeMove configurator app. • The list of supported robots is generalized, see *Robots supported by SafeMove on page 27*. • Move to Sync Position functionality added in sections *The Safety Controller control panel on page 109* and *Software synchronization guidelines on page 197*.
K	Released with RobotWare 7.8. • Minor updates.

Continues on next page

Continued

Revision	Description
L	Released with RobotWare 7.10. • Added limitation regarding TCP speed for Tool Force Supervision. • Added support for *PROFISafe Controller*. • Added missing information about *Cyclic Brake Check* functionality. • Added information about global supervision functions are available in the FlexPendant configurator app for SafeMove, see *Supervision functions on page 116*. • Added information about singularity handling, see *Tool Force Supervision (TFO) on page 59*. • Added information about protected groups in the safety configuration. • Added information about execution order for combinatory logic, see *Pre-logic and post-logic on page 96*.
M	Released with RobotWare 7.12. • Added the section *Display of safety violations on page 115*. • Added the section *Load a safety configuration template on page 117*. • Added example for the *Safe Payload Supervision* functionality, see *Safe Payload Supervision (SPA) on page 48*. • Updated user interface for protected groups. • Improved some of the descriptions, see *Configure the supervision functions on page 148*.
N	Released with RobotWare 7.13. • Added information about the new main computer (DSQC1095) and where this affects the functionality. The computer itself is described in the product manual for the respective controller. • Added information about safety add-ins. • Corrected information about number of operations, see *Safe I/O system rules and limitations on page 104*. • Corrected information about signals for commissioning mode, see *The Safe IO Configurator on page 90*. • Updated number of supported axes, due to the release of OmniCore V400XT. • Corrected the information about the servo lag for OmniCore, see *Servo Delay Factor and Servo Lag on page 210*.
P	Released with RobotWare 7.14. • Updated information about encapsulations in the FlexPendant configurator app for SafeMove, see *The SafeMove configurator app on FlexPendant on page 113*. • Added information about importing protected groups.
Q	Released with RobotWare 7.15. • Added *Safety functions and operating modes on page 35*. • Added the function *Safe Disable of Drive Unit (SDI) on page 63*. • Added note about incompatibility for main computers in section *Load the configuration to the safety controller on page 167*. • Added *Configuring Laser Scanner Calculation on page 165*.
R	Released with RobotWare 7.17. • Corrected the unit for tolerance, see *Configuring Stand Still Supervision on page 161*.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

Tip

All documents can be found via myABB Business Portal, *www.abb.com/myABB*.

Product manuals

Manipulators, controllers, DressPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Troubleshooting.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.
- How to use the application.

Continues on next page

Continued

- Examples of how to use the application.

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

Safety

Safety regulations

Before beginning mechanical and/or electrical installations, ensure you are familiar with the safety information in the product manuals for the robot.

The integrator of the robot system is responsible for the safety of the robot system.

This page is intentionally left blank

1 Introduction

1.1 Overview of functional safety

Purpose

The purpose of the functional safety options is to provide easy-to-use safety functions in the robot. Functional safety includes a complete software and hardware solution that is fully integrated with the robot controller and the RobotStudio programming environment.

SafeMove is the main functional safety option, but the safety module can also be used in various applications without the SafeMove option. For example to communicate with a safety PLC through safe fieldbus communication.

SafeMove functions

SafeMove ensures a high safety level in the robot by using supervision functions that can stop the robot. Note that the functional safety options is one component in the safety system of a complete robot cell, normally complemented by other equipment (for example light barriers) for detecting the whereabouts of the operator.

Function	Description
Tool supervision functions	Protects the operator and enhances machine and equipment safety by supervising the position (*Tool Position Supervision*), speed (*Tool Speed Supervision*) and orientation (*Tool Orientation Supervision*) of the tool. For some robots, the tool force can also be supervised (*Tool Force Supervision*). This function also includes supervision of the axis torque.
Axis supervision functions	Protects the surroundings by supervising the axis position (*Axis Position Supervision*) and the axis speed (*Axis Speed Supervision*).
Stand Still Supervision	Supervises the stand-still of robot axes without having to switch the robot to Motors Off. It enables operators to perform tasks in the immediate vicinity of the robot.
Contact application tolerance	Allows the robot to be in contact with the work-piece in limited areas. This can for example be used in applications where the robot is used for grinding or during tool change.
Power and force limiting	Functions for collaborative applications, for example power and force limiting, and support to calculate speed and force.
Cyclic brake check	Supervises that the brakes are checked with a cyclic interval.
Stop functions	Triggers stop of the robot using safe fieldbus inputs from the safety PLC.
Safe Disable of Drive Unit	Safely disables a robot or additional axis by setting it in a state with all brakes applied and servo control switched off.

Some examples of applications:

- Manual loading stations
- Manual workpiece inspection during operation
- Optimization of cell size
- Protection of sensitive equipment
- Ensuring safe orientation of emitting processes

Continues on next page

Functional safety options

The safety controller is included as standard in the robot controller. To use any of the safety functionality, it is necessary to have the corresponding option:

- [3043-1] *SafeMove Basic*
- [3043-2] *SafeMove Pro*
- [3043-3] *SafeMove Collaborative* I
- [3023-2] *PROFIsafe Device*
- [3023-1] *PROFIsafe Controller*
- [3076-2] *Safety Over EtherCAT Device*
- [3044-1] *3 modes Keyless* II
- [3044-2] *2 modes Keyless*

I *SafeMove Collaborative* is not available for all robots.

II Included for all systems unless 3044-2 is selected. Therefore, this is not visible in **Modify Installation** function.

Visual SafeMove

The functional safety options described above gives you access to the Visual SafeMove configurator in RobotStudio. With Visual SafeMove you can:

- configure and visualize supervision functions in a 3D environment.
- configure stop functions, such as automatic stop.
- configure Cyclic Brake Check.
- configure safe signals (safe Ethernet communication and I/Os are configured in *I/O Engineering Tool*).
- configure signal logics.
- configure system status outputs.

There is also a SafeMove application on the FlexPendant. It has much of the same functionality but the RobotStudio environment has more capabilities.

SafeMove Basic, SafeMove Pro, and SafeMove Collaborative

The below table lists the differences between SafeMove Basic, SafeMove Pro, and SafeMove Collaborative.

Function	SafeMove Basic	SafeMove Pro	SafeMove Collaborative
Supported number of axes[i]	12	12	9
Safe ranges	8	8	8
Safe zones	1	16	16
Tool changer support	-	16 tools	16 tools
Axis Position Supervision	Yes	Yes	Yes
Axis Speed Supervision	-	Yes	Yes
Total Global Supervision	8	8	8
Tool Orientation Supervision	-	Yes	Yes
Tool Force Supervision[ii]	-	-	Yes
Tool Position Supervision	1	Yes	Yes

Continues on next page

Function	SafeMove Basic	SafeMove Pro	SafeMove Collaborative
Tool Speed Supervision	-	Yes	Yes
Stand Still Supervision	-	Yes	Yes
Human Contact Supervision	-	-	Yes
Contact application support	Yes	Yes	Yes
Safe Disable of Drive Unit	Yes	Yes	Yes

i The number of supported axes depends on what controller variant is selected. See the product specification for the robot controller.

ii Includes *Axis Torque Supervision*.

The safety functions *Safe Disable of Drive Unit* and *Axis Position Supervision* are also available if the options *MultiMove Coordinated* or *MultiMove Independent* are present, or if the drive system options contain at least one additional drive unit (ADU).

Basic approach

This is the general approach for setting up the safety module and SafeMove.

This can be done in RobotStudio or on the FlexPendant.

1. Connect the safety controller to other safety hardware and configure the safe I/O connections.
2. Configure the settings for the SafeMove functions.
3. Download the configuration to the the safety controller. Restart the controller.
4. Synchronize the safety controller.
5. Make sure the activation input signals are activating the desired supervision functions.
6. Validate the configuration.
7. Lock the configuration.

For more detailed instructions, see sections *Installation on page 65* and *Configuring SafeMove on page 125*.

Requirements

Robust supervision functionality in SafeMove requires correct settings of payload and additional axes, since this will affect the calculated accepted servo lag. Also note that external forces applied on the manipulator can cause a negative influence on the supervision functions, since the servo lag might differ from the calculated values, due to such external forces.

DANGER

A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

The validation must also consider that the braking starts after a zone is violated, so additional stopping distances may be required, which depend on many factors, for example mass and speed.

1.2 Safety

1.2.1 Safety aspects for the safety module and SafeMove

Overview

The safety controller is integrated in the robot controller, with the purpose of providing safety functionality for the robot. Safe output and input signals are typically connected to cell safety circuitry by safe communication with a safety PLC. The safety PLC can take care of interlocking in the robot cell, for example, in order to prevent robot and operator to enter a the same area at the same time.

In this chapter we describe how the safety controller and SafeMove comply with relevant safety standards and regulations.

Note

The safety controller and SafeMove is only a part of the robot system, it is the responsibility of the user to do a risk assessment of the robot system. It is also the responsibility of the user of SafeMove to ensure that the robot system is designed and installed in accordance with the safety requirements set forth in the standards and regulations of the country where the robot system is installed.

1.2.2 Standards conformance

Standards

The safety controller and SafeMove has been designed to fulfill applicable parts of the following standards.

- EN ISO 12100:2010 Safety of machinery - General principles for design - Risk assessment and risk reduction
- EN 60204-1:2006/A1:2009 Safety of machinery - Electrical equipment of machines - Part 1: General requirements
- EN ISO 10218-1:2011, Robots for industrial environments - Safety requirements - Part 1: Robot
- EN 61000-6-2:2005 EMC, Generic immunity
- EN 61000-6-4:2007/A1:2011 EMC, Generic emission
- EN ISO 13849-1:2015 Safety of machinery - Electrical equipment of machines - Part 1: General requirements

  EN ISO 13849-2:2012 Safety of machinery - Safety-related parts of control systems - Part 2: Validation

1.2.3 Specific safety requirements

Specific safety requirements for SafeMove

SafeMove complies with EN ISO 10218-1 in general and specifically complies with chapter 5.4.2, that is, the following requirements.

When safety related control systems are required, the safety related parts shall be designed so that:

- A single fault in any of these parts shall not lead to the loss of the safety function.
- Whenever reasonably practicable, the single fault shall be detected at or before the next demand upon the safety function.
- When the single fault occurs, the safety function is always performed and a safe state shall be maintained until the detected fault is corrected.
- All reasonably foreseeable faults shall be detected.

This requirement is considered to be equivalent to structure category 3 as described in ISO 13849-1. Category 3 is normally fulfilled by redundant circuits, such as dual channels, which is the case for SafeMove. SafeMove together with the safety module and the robot controller also complies with performance level (PL) "d" according to ISO 13849-1. This safety level is equivalent to SIL 2 as defined in IEC 61508.

Note

It is the responsibility of the integrator to do a risk assessment for the final application, also for a collaborative application. The safety needs to be configured depending on the application, to fulfill the requirements in ISO/TS 15066.

1.2.4 Safe design of SafeMove

Overview

SafeMove has two important types of supervision functionality.

The first one being to ensure that the drive system is working correctly, making the robot follow the ordered value from the main computer as expected.

The second being to supervise the robot position and speed and stopping the robot or setting output signals to indicate a hazard.

Supervision of the drive system

The main computer calculates the absolute motor position values sent as reference to the drive system, and simultaneously sends them to the safety controller. The drive system reports the actual rotational motor position values via the main computer to the safety controller, as a separate process from the reference value. Since these values are within one revolution, the absolute position is calculated by adding values from internal revolution counters in both the drive system and in SafeMove.

By comparing the ordered motor position and the actual motor position, SafeMove can detect any difference (outside a permitted lag deviation) between the two positions, thereby ensuring that the drive system is working properly according to the first supervision function as described above.

It is important to ensure that the safety controller and the robot controller are synchronized. The safe sync position is defined during configuration and stored in the safety controller. In this position, SafeMove will calculate the robot joint positions and check against a stored value to confirm that the synchronization is correct, covering the following points:

- SafeMove is working correctly with the right revolution counter value.
- The right manipulator is used.
- The calibration value is correct.
- The SMB is working correctly.

Category 3 supervision

The supervision complies with category 3, that is, two separate channels shall always give the same result. One channel consists of the drive system, motors, resolvers, and measurement system. The second channel consists of the ordered value from the main computer. These channels are compared using the SafeMove evaluation circuits, which in itself is dual channel.

Additional safety design

Additional safety, over and above what is formally required, is brought to the concept by the inherent dual channel character of the resolver, thanks to its dual sine and cosine output, where the square sum is supervised to be close to 1.

Continues on next page

1.2.4 Safe design of SafeMove
Continued

Supervision of robot position and speed

The second type of supervision functionality (to supervise the robot position and speed) is fulfilled by letting SafeMove compare the robot position and speed with limit values configured by an authorized user (so called Safety User). If any value is outside its defined safe area, the supervision functions will stop the robot (or set an output signal).

To ensure that also this supervision complies with the category 3 requirement, SafeMove is internally working with a two channel microprocessor based system. Both processors make parallel calculations comparing the actual position and the reference position.

1.2.5 Certifications

Overview

The functional safety and SafeMove options have been certified by external organizations as described below.

Certifications by RISE Research Institutes of Sweden

RISE Research Institutes of Sweden has made an assessment of the safety functionality included in DSQC1015/DSQC1095, including its software, according to EN ISO 13849-1:2015 and issued a certificate with regard to 2006/42/EC, Annex V, item 4, as a logic unit.

RISE has also assessed that the software blocks, implementing safety functions as defined in EN-ISO 10218-1:2011, are correctly implemented in the robot controller.

Certifications by UL

The safety functionality included in the robot controller, in DSQC1015/DSQC1095, including its software, is approved by UL according to the following standards:

- UL 1740, Standard for Robots and Robotic Equipment
- ANSI/RIA R15.06, Industrial Robots and Robotic Systems
- CAN/CSA Z434-14, Industrial Robots and Robot Systems - General Safety Requirements

1.2.6 Conclusion

Conclusion

As has been shown above and confirmed by third party certifications, the robot controller and SafeMove fulfill all relevant current safety standards globally.

1.3 Limitations

Supported robot controllers

The functional safety options and the safe fieldbus options are available for the following controllers:

- OmniCore C30
- OmniCore C90XT
- OmniCore V250XT
- OmniCore V400XT

Note

Safety configurations created on a controller with main computer DSQC1029 cannot be loaded in main computer DSQC1095.

Robots supported by SafeMove

SafeMove is supported for the articulated and collaborative manipulators, if they are connected to a controller that has functional safety or SafeMove options enabled.

Some robot models are not supported, for example:

- SafeMove does not support delta robots (parallel arm robots).
- SafeMove does not support SCARA robots[I].

[I] Only support for *Basic joint supervision* mode, see *Basic joint supervision mode on page 34*.

Note

YuMi robots with SafeMove requires using the enabling device to enable the motors in manual mode. For more information, see:

Product manual - IRB 14050

WARNING

Even if a model is supported by SafeMove, each installed robot must be verified individually to ensure that no mechanical or other deviations exists which would make SafeMove position measurements incorrect.

This is normally done during safety function verification, see *Validate the configuration on page 168*.

Supported mounting angles

SafeMove supports any mounting angle. For example floor mounted, tilted, inverted, etcetera.

Supported tracks

SafeMove supports all ABB track motion units.

Continues on next page

Supported positioners

SafeMove supports positioners that are single axis mechanical units. Positioners with several axes are treated as multiple single axes, for example two axes positioners will be treated by SafeMove as two single axes and can be monitored as such using axis supervision.

Positioners that are used with activation/deactivation feature, is not supported. The axes of the positioner must be active at all times.

Servo welding gun

SafeMove does not support supervision of servo welding guns. That is, the robot axes can be supervised, but not the axis of the servo welding gun.

Non ABB additional axes

Non ABB track motion units, non ABB positioners, and other additional axis may be supported by the SafeMove option but this needs to be verified case by case.

To verify if a non ABB additional axis can be used with SafeMove, tune the additional axis before configuring the SafeMove parameters. If a properly tuned and configured non ABB additional axis still generates error messages regarding servo lag, then it cannot be used with SafeMove. For more information about tuning an additional axis see *Application manual - Additional axes*.

Work area for additional axes

There are always the following upper and lower work area limitations for additional axes:

- Track unit length (arm side) max ± 448 m
- Rotating axis (arm side) max ± 25700 degrees or ± 448 radians

On the motor side there is also a limitation of ± 32000 revolutions.

Combined external axes and robot with gantry

SafeMove does not support supervision of combined external axes, such as a gantry, or supervision of robots moved by the gantry.

It is possible to supervise robots moved by a gantry using the Basic joint supervision mode. In Basic joint supervision mode, only SafeMove functions working on axes are available.

Basic joint supervision mode

In Basic joint supervision mode, SafeMove supports supervision of some robots and external axes that are not normally allowed. Examples are robots mounted on a gantry, or other non-track external axes.

In Basic joint supervision mode, no safe zones or tool supervision functions are allowed. In addition, there are changes to some internal SafeMove supervision functions:

- The manual mode speed supervision of the TCP, wrist center point, and elbow are disabled. Instead, axis speed limits are set to approximate a Cartesian speed limit. The user is responsible for validating that this speed limit satisfies the requirements for safe manual mode operation.

Continues on next page

- The Safe Brake Ramp supervision for the robot during a stop category 1 is changed, so that SafeMove supervises that all robot axes will stop within 1 second.

Tool changer

SafeMove Pro supports up to 16 different tools. All included tools must have their appropriate settings in the configuration file. The selection of tool must be supervised using a safe fieldbus.

Robot mounted on rotational axis

SafeMove does not support supervision of a robot mounted on a rotational axis. Using the Basic joint supervision mode, axis monitoring will be possible but the tool supervision functions are not supported.

No deactivation

Additional axes that are used with activation/deactivation feature are not supported. If additional axes are to be used, they must also be active at all times.

Independent joint

Independent joint cannot be monitored by SafeMove.

Electronically linked motors

SafeMove supports supervision of additional axes using Electronically Linked Motors. Using this feature may cause servo lag problems, particularly if the follower axis is included in the safety configuration. It may be necessary to modify the tuning of the axes, or increase the tolerances in the SafeMove configuration. The performance should be verified case by case.

RAPID non motion execution

This test feature cannot fully be used together with the SafeMove option.

Responsive jogging

Stop category 1 is deactivated in manual mode when responsive jogging is active. This is because a robot that is stopped with a stop category 1 follows its programmed path while decelerating. When using responsive jogging there is no defined path available.

Note

Responsive jogging can be deactivated by changing the parameter *Jog Mode* from *Responsive* to *Standard*.

For more information about parameter *Jog Mode*, see type *Jog Parameters*, topic *Motion* in *Technical reference manual - System parameters*.

Continues on next page

SoftMove

When SafeMove is used together with SoftMove there is a risk for servo lag problems. The recommended action is to add a Contact Application Tolerance (CAP) in the area where SoftMove is active.

For more information about SoftMove, see *Application manual - SoftMove*.

1.4 Terminology

About these terms

Some words have a specific meaning when used in this manual. It is important to understand what is meant by these words. This manual's definitions of these words are listed below.

Term list

Term	Definition
Stop category 0	Stop by immediate removal of power to the actuators. Mechanical brakes are applied. A robot that is stopped with a stop category 0 does not follow its programmed path while decelerating.
Stop category 1	Controlled stop with power available to the actuators to achieve the stop. Power is removed from the actuators when the stop is achieved. A robot that is stopped with a stop category 1 follows its programmed path while decelerating.
Checksum	A checksum is a unique identifier created during the safety configuration. It is possible to protect groups of elements in the safety configuration, therefore there can be more than one checksum.
Occupationally safe	Safe for a person to be in an area.
Operationally safe	Safe for the machinery but not safe for persons to enter the area.
Safe input	Dual monitored digital input.
Safe output	Dual monitored digital output.
Safety controller	A safety feature in the robot controller, handling SafeMove functionality.

1.5 Abbreviations and acronyms

Overview

This section specifies typical abbreviations and acronyms used in this manual.

Abbreviations/acronyms list

Abbreviation/acronym	Description
APO	Axis Position Supervision
ASP	Axis Speed Supervision
CAP	Contact Application Tolerance
CBC	Cyclic Brake Check
SPA	Safe Payload Supervision
SDI	Safe Disable of Drive Unit
SST	Stand Still Supervision
TFO	Tool Force Supervision
TOR	Tool Orientation Supervision
TPO	Tool Position Supervision
TSP	Tool Speed Supervision

2 Safety functions provided by SafeMove

2.1 Overview of SafeMove functions

Overview

The SafeMove functions can be divided into the following categories:

- General functions, see:
  - *Manual Operation Supervision on page 36*
  - *Contact Application Tolerance (CAP) on page 37*
- Synchronization functions, see:
  - *Software synchronization on page 41*
  - *Hardware synchronization on page 43*
- Supporting functions, for example verification of brakes, see:
  - *Safe Brake Ramp on page 45*
  - *Cyclic Brake Check guidelines on page 200*
  - *Human Contact Supervision on page 47*
  - *Safe Payload Supervision (SPA) on page 48*
- Supervision functions, can stop the robot or set a safe output signal, see:
  - *Stand Still Supervision (SST) on page 50*
  - *Axis Speed Supervision (ASP) on page 52*
  - *Tool Speed Supervision (TSP) on page 53*
  - *Axis Position Supervision (APO) on page 55*
  - *Tool Position Supervision (TPO) on page 57*
  - *Tool Orientation Supervision (TOR) on page 58*
  - *Tool Force Supervision (TFO) on page 59*
  - *Control Error Supervision on page 62*
  - *Safe Disable of Drive Unit (SDI) on page 63*

About the supervision functions

Supervision functions can be activated and deactivated with safe input signals or be configured to be permanently active.

The supervision functions can stop the robot and additional axes, or set a safe output signal, if a violation occurs.

Signal	Status
Activation	The signal is set to 0 for activation.
Function active status	The signal is set to 1 when active.

Continues on next page

Signal	Status
Violation action • **Signal**	The signal is set to 0 at violation.  **Note** When a signal is set to 0 at violation, it will remain 0 for at least 250 ms even after the violation has ended.

Local and global functions

There are two types of functions in SafeMove, local functions and global function.

Local functions are active when the robot is in defined parts of its working area, it could be inside a safe zone or within specified ranges. Those functions are used for setting a speed limitation in specific areas (Tool Speed Supervision) or for protecting equipment being hit by the robot (Tool Position Supervision).

Global functions are general functions that are active regardless of the position of the robot. It could for example be a general speed limitation activated when the cell door is opened.

For more information, see *The Global Functions button on page 77*.

Combining functions

The supervision functions can be used separately or in a variety of combinations.

Basic joint supervision mode

Certain combinations of robots and external axes are normally not supported by SafeMove, such as robots mounted on a gantry or other non-track external axes.

In this case, Basic joint supervision mode can be used, which will allow a limited set of safety supervision functions. In Basic joint supervision mode, no safe zones or tool supervision functions are allowed.

In addition, there are changes to some internal SafeMove supervision functions:

- The manual reduced speed supervision of the TCP, wrist center point, and elbow are disabled. Instead, axis speeds are supervised, with limits set to approximate a Cartesian speed limit.

  The user is responsible for validating that this speed limit satisfies the requirements for safe manual mode operation.
- The Safe Brake Ramp supervision for the robot during a stop category 1 is changed, so that SafeMove supervises that all robot axes will stop within 1 second.

For information on how to select Basic joint supervision mode, see *Robot parameters node on page 83*.

Exclude from configuration

In the implementation of the safety controller, some safety supervision functions are included in the system even if the SafeMove option is not selected. This includes Safe Brake Ramp, Manual Operation Supervision, and Control Error Supervision. The included safety supervision functions require synchronization of the safety controller and that the configuration is validated and locked.

Continues on next page

In some cases it is necessary to exclude the safety settings from the configuration. For example when configuring a safe fieldbus on robots that are not supported by SafeMove. This is done with the setting *Exclude from configuration*. This setting can also be used to exclude supported robots, for example individual robots in a MultiMove application, and supported robots running only a safe fieldbus.

When excluding the safety supervision from the configuration, the robot behaves as if no safety module is installed. That means:

- Instead of Safe Brake Ramp a one second delay is used between the request of a stop category 1 and the completing stop category 0.
- No safety supervision of manual reduced speed.
- No safety supervision on the position data received from the serial measurement board, SMB, and the drive system, (resolver input and revolution counter).
- No need to synchronize and lock the configuration.

For information on how to exclude a robot from the safety configuration, see *Robot parameters node on page 83*.

DANGER

A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

Safety functions and operating modes

The following table describes what functions are active in manual mode and in automatic mode.

Function	Manual mode	Automatic mode
Contact Application Tolerance (CAP)	Active	Active
Safe Brake Ramp	Active	Active
Cyclic Brake Check (CBC)	Active	Active
Safe Payload Supervision (SPA)	Not active	Active
Safe Disable of Drive Unit (SDI)	Active	Active
Stand Still Supervision (SST)	No stop, only status signal	Active
Axis Speed Supervision (ASP)	No stop, only status signal	Active
Tool Speed Supervision (TSP)	No stop, only status signal	Active
Axis Position Supervision (APO)	Active, stop temporarily disabled after violation	Active
Tool Position Supervision (TPO)	Active, stop temporarily disabled after violation	Active
Tool Orientation Supervision (TOR)	Active, stop temporarily disabled after violation	Active
Tool Force Supervision (TFO)	No stop, only status signal	Active

Stand Still Supervision (SST) can be configured to be active in manual mode.

2.2 General functions

2.2.1 Manual Operation Supervision

Manual Operation Supervision

Manual Operation Supervision is a function that is active in manual operating mode and supervises that motion is below the configured manual mode supervision speed.

Functionality

While Manual Operation Supervision is active, a supervision makes sure that the tool center point (TCP), wrist center point (WCP), and elbow speed does not exceed 250 mm/s (unless a lower value is configured).

Manual Operation Supervision overrides safety functions by muting stops from the safety controller due to supervision functions.

Note

If Manual Operation Supervision is active and the robot is jogged to a non-violation position and then into a supervision violation position again, the robot will stop again. The new violation must be confirmed by releasing the enabling device on the FlexPendant before the jogging can be resumed.

DANGER

If Basic joint supervision mode is used, the supervision of the TCP, WCP, and elbow speed is disabled.

Instead, axis speeds are supervised with limits set to approximate a Cartesian speed limit, corresponding to the parameter **Max speed in manual mode**.

The user is responsible for validating that this speed limit satisfies the requirements for safe manual mode operation.

If the speed in manual mode is determined to be too high, decrease the parameter **Max speed in manual mode**.

Settings

The following parameters can to be configured for Manual Operation Supervision:

- Max speed in manual mode.

See *Configure Manual Operation Supervision on page 133*.

Function activation

Manual Operation Supervision is activated by switching to manual mode.

Dependencies to other supervision functions

Manual Operation Supervision can be used in combination with all other SafeMove functions.

2.2.2 Contact Application Tolerance (CAP)

Contact Application Tolerance

Contact Application Tolerance relaxes the supervision of the servo lag if either:

- all configured axes are within the corresponding safe axis range,
- the TCP is within the corresponding safe zone,
- the activation signal for the Contact Application Tolerance function is 0 (if used).

Functionality

Contact Application Tolerance relaxes the Control Error Supervision (servo lag) to a higher value if all configured axes are within the defined axis range, or the TCP is within the defined zone, and the activation signal is 0 (if used).

Contact Application Tolerance can be used, for instance, in machine tending, when the servo loop gain is reduced (soft servo), or during Force Control. It is also useful when external forces are applied to the robot, for example during tool change.

If the robot is within the defined range/zone, then the safety level is considered to be operationally safe rather than occupationally safe. That means it is not safe for personnel to be in the range/zone defined for Contact Application Tolerance.

For axis ranges, both reference value and measured value for all axes must be within the defined range to be able to activate the relaxed control error. For zones, both reference value and measured value for the TCP must be within the defined zone to be able to activate the relaxed control error.

Up to 9 axes can be supervised simultaneously.

WARNING

When the Contact Application Tolerance is active then the dual channel safety tolerance is reduced with the configured value. This must be considered in the design of the robot application.

Settings

The following settings can be configured for Contact Application Tolerance:

- An axis range or a zone to apply Contact Application Tolerance for.
- Permissible control error for each axis, in degrees or mm on arm side.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.

How to define these settings is described in *Configuring Contact Application Tolerance on page 159*.

Dependencies to other supervision functions

If Contact Application Tolerance is active, it overrides the Control Error Supervision function. That means that all other active safety controller functions work with relaxed Control Error Supervision.

Continues on next page

2.2.2 Contact Application Tolerance (CAP)
Continued

Contact Application Tolerance can be used in combination with all other SafeMove functions.

Limitations

Contact Application Tolerance is not considered to be active if the run chain is open.

Related information

Control Error Supervision on page 62.

2.2.3 SafeMove Assistant

Purpose

SafeMove Assistant is a functionality in RobotWare that helps users to program their application when there is an active SafeMove configuration. The assistant will read the active configuration and plan the trajectories according to the limits and settings in that configuration. It will set the speed so that SafeMove will not trigger violations etc. It will also stop with error message in case the robot is programmed to enter a forbidden zone etc.

SafeMove Assistant will automatically adjust robot behavior to adopt to the active SafeMove configuration, the robot will adopt to speed limited zones and stop before entering forbidden zones.

CAUTION

SafeMove Assistant is not a safety function.

For example, if using a fence, then a safety distance is required between the safe cartesian zone and the fence.

Note

In case of SafeMove Assistant fails, the SafeMove supervision will trigger an emergency stop.

Description

SafeMove Assistant will check if any SafeMove speed limit is active for any Cartesian speed checkpoint (TCP, tool points, and elbow). If this is the case, a corresponding speed limit is applied in the path planner. For technical reasons, only the speed of the TCP, the wrist center point (WCP), and the elbow are limited by the path planner. Therefore, in cases where other tool points move faster than the TCP, SafeMove may trigger a Tool Speed violation. To avoid this, change the program or decrease the value of the parameter *SafeMove assistance speed factor* (see below).

SafeMove Assistant is not active in manual mode.

SafeMove Assistant does not take path corrections generated at lower level into account. It is therefore an increased risk of SafeMove violations when running applications like Externally Guided Motion or conveyor tracking.

Continues on next page

2.2.3 SafeMove Assistant
Continued

System parameters

SafeMove Assistant can be disabled for the SafeMove validation etc. This is done with the parameter *Disable SafeMove Assistance*, in the type in *Motion System*.

There are some parameters that can be changed in case robot system has minor overshoot or in any other way triggers SafeMove violations.

Parameter	Description
SafeMove Assistance Speed Factor	That has a default setting of 0.96 which corresponds to 96% of speed supervision will be the speed that path planner will use. This parameter can be decreased to reduce that risk but can in most cases be left at default value.
SafeMove assistance zone margin	When robot is running on a zone border there is a small risk that SafeMove can trigger violations when going in and out of the zone. This parameter can be increased to reduce that risk but can in most cases be left at default value.

For more information, see the parameters in the type *Motion System* described in *Technical reference manual - System parameters*.

2.3 Synchronization functions

2.3.1 Software synchronization

Software synchronization

Software synchronization is a function that makes sure that the safety controller has the correct information regarding robot position.

Unsynchronized state can, for example, occur:

- If one or more axes were moving during shutdown or power off.
- After a failed synchronization.

> **Note**
>
> On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

> **Note**
>
> The CRB 15000 (GoFa) has a limitation of allowed SafeMove software synchronization position of all axes. The allowed software synchronization range is +-179 degrees.

Functionality

Software synchronization is initiated from the FlexPendant. How to execute a software synchronization is described in section *Performing a synchronization on page 197*.

If the synchronization attempt is unsuccessful, the synchronization procedure must be executed again until successful.

> **Note**
>
> The supervision functions can only be active while SafeMove is synchronized. When unsynchronized, only manual mode operation with reduced speed is possible until synchronization is executed successfully.

Settings

The following settings need to be configured for software synchronization:

- Status signal.
- Angles and positions of robot (and additional axes) at the synchronization position.

Dependencies to other supervision functions

Software synchronization is always available even if hardware synchronization is configured.

Continues on next page

2.3.1 Software synchronization
Continued

Related information

Configure the synchronization position on page 137

Software synchronization guidelines on page 197.

Recovery after safety violation on page 187.

2.3.2 Hardware synchronization

Hardware synchronization

Hardware synchronization is a function that makes sure that the robot calibration is correct by using a physical synchronization switch.

Unsynchronized state can, for example, occur:

- If one or more axes were moving during shutdown or power off.
- After a failed synchronization.

> **Note**
>
> On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Functionality

The robot must move to a safe synchronization position to ensure that the safety controller and the robot controller are synchronized. The safe synchronization position is defined during configuration and stored in the safety controller.

The robot must move to the safe synchronization position and activate a switch. When the switch is activated, the safety controller assumes that the robot revolution counters are correct. It also calculates the arm position from the motor positions, the gear ratio, and its internal revolution counter. If the position matches the stored synchronization position within half a motor revolution, then the synchronization is assumed to be correct.

If the synchronization is correct, the safety controller then sends a message to the robot controller, confirming that the safety controller is synchronized to its mechanical units, and continues with its regular operation.

> **Note**
>
> The supervision functions can only be active while SafeMove is synchronized. When unsynchronized, only manual mode operation with reduced speed is possible until synchronization is executed successfully.

Settings

The following settings need to be configured for hardware synchronization:

- Synchronization signal.
- Angles and positions of robot (and additional axes) at the synchronization position.

Dependencies to other supervision functions

Software synchronization is always available even if hardware synchronization is configured.

Continues on next page

2.3.2 Hardware synchronization
Continued

Limitations

- The safe sync position must be within reach for the robot. It must not be a singularity, that is all six axis must have unique positions.

Related information

Configure the synchronization position on page 137
Hardware synchronization guidelines on page 199.
Recovery after safety violation on page 187.

2.4 Supporting functions

2.4.1 Safe Brake Ramp

Safe Brake Ramp

Safe Brake Ramp is an active supervision function that supervises stop category 1 initiated by the safety controller.

Functionality

When a stop category 1 is triggered by the safety controller, the motors are used for a controlled deceleration along the planned motion path. Safe Brake Ramp supervises this deceleration. If the deceleration is too slow, a stop category 0 is triggered. After 1 second, a stop category 0 is always triggered regardless.

A stop category 1 usually stops faster than the margins for Safe Brake Ramp, so under normal circumstances Safe Brake Ramp does not trigger.

Note

Depending on the application, Safe Brake Ramp may trigger more often, for example for tilted robot or heavy load. This results in a stop category 0.

Note

If Basic joint supervision mode is used, the Safe Brake Ramp supervision is changed to supervise that all robot axes will stop within 1 second.

Settings

For track motions and other additional axes, the parameters **Brake Ramp Limit** and **Ramp Delay** have to be set. The parameter **Start Speed Offset** is used for both manipulator and all additional axes.

Function activation

Safe Brake Ramp cannot be dynamically activated/deactivated. If it is configured to be active, it is always active.

Dependencies to other supervision functions

Safe Brake Ramp will be used in combination with all other SafeMove functions using stop category 1.

Limitations

- Safe Brake Ramp only supervises stop category 1 initiated by the safety controller. Stops initiated elsewhere, e.g. by the robot controller, are not supervised.

Related information

Stop category 1, see *Terminology on page 31*

Continues on next page

2.4.1 Safe Brake Ramp
Continued

Stop category 0, see *Terminology on page 31*

Explanation of Safe Brake Ramp on page 88

2.4.2 Human Contact Supervision

Human Contact Supervision

The *Human Contact Supervision* supporting function will in quasi-static case give suggested values of speed and force to not exceed any desired peak level. These values must be validated.

> **CAUTION**
>
> The functionality is based on the recommendations in ISO/TS 15066. A risk assessment of the final application must always be done, where the calculations are reviewed and verified by test.

Functionality

Human Contact Supervision calculates the maximum allowed tool speed and maximum allowed tool force and torque, for the functions Tool Speed Supervision and Tool Force Supervision. The calculation can be done for transient contact or quasi-static contact.

The speed, force, and applied torque on the tool and each joint shall be below the limit for injuring the operator.

See ISO/TS 15066 for more information.

Settings

The following parameters can be configured for Human Contact Supervision:

- The mass of the tooling and workpiece.
- The surface area of the workpiece.

How to define these settings is described in *Configuring Human Contact Supervision on page 148*. See also *Guidelines for transient and quasi-static contact, CRB 15000 on page 207*.

> **Note**
>
> This function is implemented for the reducing the risk in a clamping situation. The speed must be taken into consideration for not causing injury in case of a collision.

Limitations

- Human Contact Supervision is only available for CRB 15000 (GoFa).
- The calculation cannot be used for surface areas smaller than 1 cm^2.

2.4.3 Safe Payload Supervision (SPA)

Safe Payload Supervision

Safe Payload Supervision is a safety function that can be used to safely supervise the change of the payload held by the robot gripper. In this way, the payload mass data can be taken into account by the dynamic models used to calculate the external force and torque supervised by TFO, resulting in improved accuracy.

Functionality

Safe Payload Supervision can be used to define payload-change zones, where changing the payload held by the robot is allowed. When the robot enters a payload-change zone, SafeMove waits for the robot RAPID program to execute a `GripLoad` instruction selecting the new payload, followed by a linear verification movement. During this movement, SafeMove supervises the joint torques to verify that the payload held by the robot matches a payload configured in SafeMove. A payload is considered as a match if the torques predicted by an internal dynamic model of the robot with payload match the measured joint torques (to within a tolerance that cannot be modified).

- If a unique configured payload matches the measured joint torques, and this payload is identical to the selected payload, the selected payload is accepted.
- If several configured payloads match the measured joint torques, and one of these payloads is identical to the selected payload, the selected payload is accepted only if the difference between the mass of each matching payload and the mass of the selected payload is less than a configured maximum payload uncertainty.

WARNING

While the robot is inside the payload-change zone, SafeMove does not guarantee that the selected payload is correct. If Tool Force Supervision is used inside a payload-change zone, the total payload uncertainty must be taken into account when configuring the supervision limits.

DANGER

External forces acting on the robot during the verification movement could cause Safe Payload Supervision to verify the payload incorrectly. The application must be designed to ensure that no external forces are acting on the robot during the verification movement.

DANGER

Safe Payload Supervision does not detect changes of payload occurring outside a payload-change zone. The application must be designed so that the payload can only change inside the payload-change zones, for instance by ensuring that a held payload cannot be accidentally dropped by the robot.

Continues on next page

DANGER

All payloads that could be held by the robot must be configured in SafeMove. Using Safe Payload Supervision when the robot is holding a payload that is not configured in SafeMove will lead to undefined results.

Settings

The following parameters can be configured for Safe Payload Supervision:

- Up to 8 different payloads, including mass data expressed in either wrist or tool coordinates.
- Up to 4 payload-change zones.
- The reference speed and tool orientation during the verification movement.
- A maximum payload uncertainty.

Function activation

Safe Payload Supervision is always active, preventing operation in automatic mode unless a payload has been selected and verified.

Limitations

Safe Payload Supervision can only be used with CRB 15000.

Example of Safe Payload Supervision

In the following code example, the new load (`load5`) is identified by the `GripLoad` instruction, and then verified in the following `MoveL` instruction.

```
! Move to pick position, in the payload-change zone
MoveL p_pick, v1000, fine, toolGripper;
! Pick up payload
WaitTime\InPos,0.1;
close_gripper();
GripLoad load5;
! Move vertically at 100 mm/s to a position outside of the
     payload-change zone, to verify that the joint torques match
     the selected payload
MoveL offs(p_pick,0,0,60), v100, z5, toolGripper;
! Continue program
MoveL p10, v2000, z5, toolGripper;
```

2.5 Supervision functions

2.5.1 Stand Still Supervision (SST)

Stand Still Supervision

Stand Still Supervision is an active supervision function ensuring that all supervised axes are standing still.

Functionality

Stand Still Supervision can supervise that a robot is standing still even if the servo and drive system are in regulation. If any supervised axis starts to move, Stand Still Supervision will cause a stop category 0.

When Stand Still Supervision is active for all axes (including all additional axes), the operator is safe from harm related by robot motion when entering the working space of the robot.

8 different sets of up to 9 axes can be defined. When Stand Still Supervision is activated for a set, all axes in that set are supervised.

DANGER

Working under an axis affected by gravity which has no balancing may require a safety performance level (PL) "e", which is not provided by SafeMove. If this kind of work is intended, the risk must be added to the risk analysis of the installation and eliminated by other means (for example additional mechanical stops).

DANGER

It is not allowed to enter the working space of the robot if a Contact Application Tolerance function is active, even if a Stand Still Supervision function is active at the same time.

DANGER

For additional axes, a standstill reference tolerance must be configured.

Note

If the robot tries to move due to an error during active Stand Still Supervision, SafeMove will detect this and initiate a stop. Since there is a certain reaction time involved a slight jerk may occur.

Settings

The following parameters can be configured for Stand Still Supervision:

- Assignment of safe inputs for activation of Stand Still Supervision.

Continues on next page

- Which axes to supervise, with specified stand still measurement tolerance, for each stand still set.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.

See *Configuring Stand Still Supervision on page 161*.

Function activation

Stand Still Supervision is activated by a safe input signal, or is permanently active if only output signal and no stop is used.

> **Note**
>
> If SafeMove becomes unsynchronized, the robot will stop and the Stand Still Supervision function will be deactivated. A movement with reduced speed is possible.

Limitations

- Stand Still Supervision is only available for SafeMove Pro, see *Functional safety options on page 18*.

2.5.2 Axis Speed Supervision (ASP)

Axis Speed Supervision

Axis Speed Supervision is an active supervision function that supervises the speed of robot axes and additional axes.

Functionality

Supervision of the speed for up to 9 axes (robot axes and additional axes). Up to 8 sets can be configured.

If any of the supervised axes is outside its allowed speed, the safety controller triggers. This violation will cause a stop category 0, a stop category 1, or set an output signal, depending on the configuration.

WARNING

Axis Speed Supervision is not active in manual operating mode and shall therefore not be used to guarantee operator safety in that mode.

Settings

The following parameters can be configured for Axis Speed Supervision:

- An optional axis range or a zone for which the Axis Speed Supervision is applied.
- Which axes to supervise.
- Maximum speed and minimum speed, defined per axis.
- Stop category 0, stop category 1, or no stop if a violation occurs.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.
- Assignment of safe input for activation of Axis Speed Supervision.

How to define these settings is described in *Configuring Axis Speed Supervision on page 152*.

Function activation

Axis Speed Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Axis Speed Supervision is only available for SafeMove Pro, see *Functional safety options on page 18*.
- The highest speed limit that can be configured is 600 degrees/s for rotational axes and 10000 mm/s for linear axes.

2.5.3 Tool Speed Supervision (TSP)

Tool Speed Supervision

Tool Speed Supervision is a supervision function that supervises the speed of the active safety tool, arm check point, and configured speed supervised points.

Functionality

Tool Speed Supervision supervises the linear speed (in mm/s) for:

- TCP for the active safety tool.
- Arm check point, "elbow".

  (The position is depending on robot type and can be user defined, but is located around axis 3).
- Wrist center point (WCP), in manual mode only.
- A number of configurable speed supervised points on the current tool.

If any of these points exceed the maximum speed, the safety controller triggers. If the TCP moves slower than the minimum speed, the safety controller will also trigger. The speed violation will cause a stop category 0, a stop category 1, or set an output signal, depending on the configuration.

There can be up to 8 global sets of Tool Speed Supervision plus one for each zone and axis range (up to 16 zones and 8 axis ranges).

WARNING

Tool Speed Supervision is not active in manual operating mode and shall therefore not be used to guarantee operator safety in that mode.

CAUTION

Since the TCP speed is determined by the programmed speed it is very important that the TCP of the active tool in SafeMove corresponds to the active tool of the robot program.

Note

The resultant robot TCP speed can in some situations be higher than the programmed TCP speed. This could happen for some robot types if the move instructions are of type `MoveJ` or `MoveAbsJ`. If this occurs, either increase the *Max Speed* for Tool Speed Supervision, or try to add intermediate robot targets in the RAPID program.

Continues on next page

2.5.3 Tool Speed Supervision (TSP)
Continued

Note

When the robot is running in manual mode, neither the elbow point nor the TCP point will exceed 250 mm/s. When the robot is running in auto mode, the robot controller will not consider the elbow speed when generating the path, only the defined TCP speed and reorientation speed. (If additional axis exists in the system, the speed data for this will also be considered.)

The result from this is that the elbow speed is sometimes higher than the programmed TCP speed. Since Tool Speed Supervision supervises the TCP, the elbow, and the speed supervision points on the tool, the speed of these points must be taken into account when creating the RAPID program.

Note

For collaborative applications, use ISO/TS 15066 for guidance.

Settings

The following parameters can be configured for each set of Tool Speed Supervision:

- An optional axis range or a zone for which the Tool Speed Supervision is applied.
- Maximum allowed speed (in mm/s) for TCP, elbow, and speed supervised points.
- An optional minimum speed for the TCP.
- Stop category 0, stop category 1, or no stop if a violation occurs.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.
- Assignment of a safe input for activation, or set as permanently active.

How to define these settings is described in *Configuring Tool Speed Supervision on page 150*.

Function activation

Tool Speed Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Speed Supervision is only available for SafeMove Pro, see *Functional safety options on page 18*.
- The highest speed limit that can be configured is 600 degrees/s for rotational axes and 10000 mm/s for linear axes.

2.5.4 Axis Position Supervision (APO)

Axis Position Supervision

Axis Position Supervision is an active supervision function that triggers a violation if any axis is outside of the defined ranges.

Functionality

Supervision of up to 9 axes (robot axes and additional axes) in each set. Up to 8 sets can be configured, one for each safe axis range.

If an axis in an active set exceeds its allowed range, the safety controller triggers. This violation will cause a stop category 0, a stop category 1, and/or set an output signal, depending on the configuration.

Settings

The following parameters can be configured for Axis Position Supervision:

- A safe range to which it should be applied.
- Stop category 0, stop category 1, or no stop if an axis is outside its range.
- Set an output signal if an axis is outside its range.
- Set a status signal when the function is active.
- Assignment of safe inputs for activation of each set of axis ranges, or set as permanently activated.

How to define these settings is described in *Configuring Axis Position Supervision on page 158*.

Function activation

Axis Position Supervision is activated by a safe input signal, or is permanently active.

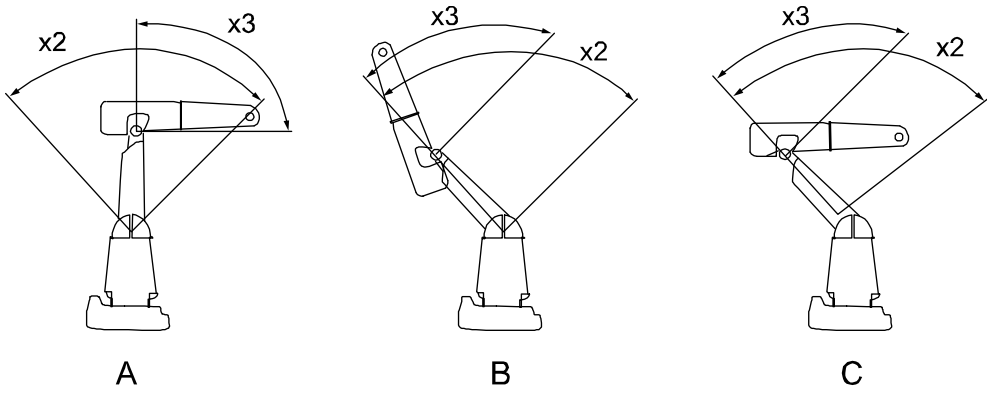
Continues on next page

2.5.4 Axis Position Supervision (APO)
Continued

Examples

This example shows a robot with defined axis ranges for axes 2 and 3 in three different positions. The function Axis Position Supervision supervises that axis 2 is within range x2 and that axis 3 is within range x3.

In positions A and B, all supervised axes are within the allowed ranges. In position C, axis 3 is not within the defined range.



xx0600003331

x2	Allowed axis position range for axis 2.
x3	Allowed axis position range for axis 3.
A	Robot position A. Both axis 2 and axis 3 are within the allowed ranges.
B	Robot position B. Both axis 2 and axis 3 are within the allowed ranges.
C	Robot position C. Axis 2 is within the allowed range but axis 3 is not within its allowed range. This will trigger a violation.

Note

The ranges define axis angles, not the position of the TCP. In robot position C, the TCP is still within what seems to be a safe range, but axis 3 is outside its defined range.

Limitations

WARNING

Be aware of that the braking starts when the axis exceeds the configured limit value.

The braking distance depends on robot type, load, position and speed, and therefore an additional stopping distance may sometimes be required to achieve the desired safety.

2.5.5 Tool Position Supervision (TPO)

Tool Position Supervision

Tool Position Supervision is a supervision function that supervises that the robot is within the allowed safe zone.

Functionality

Tool Position Supervision supervises that the robot and the active safety tool (and any configured encapsulation around them) are within the defined zone. Up to 32 sets can be configured, max two per safe zone.

If the robot is outside its allowed zone, the safety controller triggers. This violation will cause a stop category 0, a stop category 1, and/or set an output signal, depending on the configuration.

Settings

The following parameters can be configured for Tool Position Supervision:

- A safe zone to which it should be applied.
- Assignment of a safe input for activation, or set as permanently active.
- Stop category 0, stop category 1, or no stop if the robot violates its zone limits.
- Set an output signal if the robot violates its zone limits.
- Set a status signal when the function is active.
- If the upper arm should be included in the supervision, or only the tool.
- If the robot should be allowed only inside or only outside of the zone.

How to define these settings is described in *Configuring Tool Position Supervision on page 148*.

Function activation

Tool Position Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Position Supervision is only available for SafeMove Pro, see *Functional safety options on page 18*.

WARNING

Be aware of that the braking starts when the tool or robot exceeds the configured limit value.

The braking distance depends on robot type, load, position and speed, and therefore an additional stopping distance may sometimes be required to achieve the desired safety.

2.5.6 Tool Orientation Supervision (TOR)

Tool Orientation Supervision

Tool Orientation Supervision is an active supervision function that supervises that the tool orientation of the active safety tool is within the allowed tolerance.

Functionality

Tool Orientation Supervision supervises the tool orientation. If the tool orientation is outside its allowed tolerance, the safety controller triggers. This violation will cause a stop category 0, a stop category 1, or set an output signal, depending on the configuration.

Up to 8 sets can be configured.

CAUTION

Since the tool orientation is determined by the programmed tool orientation, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.

Settings

The following parameters can be configured for Tool Orientation Supervision:

- An optional axis range or a zone for which the Tool Orientation Supervision is applied.
- Allowed orientation of the tool in x and z directions including a tolerance.
- Assignment of a safe input for activation, or set as permanently active.
- Stop category 0, stop category 1, or no stop if a violation occurs.
- Set an output signal if the tool orientation violates its limits.
- Set a status signal when the function is active.

How to define these settings is described in *Configuring Tool Orientation Supervision on page 154*.

Function activation

Tool Orientation Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Orientation Supervision is only available for SafeMove Pro, see *Functional safety options on page 18*.

2.5.7 Tool Force Supervision (TFO)

Tool Force Supervision

The Tool Force Supervision monitors that supervises that the external force on the tool and the external torque on each joint does not exceed a given limit, to prevent injury in case of a clamping situation.

Tool Force Supervision is only available for CRB 15000.

Functionality

Tool Force Supervision supervises that the tool force and the joint torques are below the specified limits.

Up to 8 sets can be configured.

If the force or torque exceed the allowed limit, the safety controller triggers. This violation will cause a stop category 1, and/or set an output signal, depending on the configuration.

To fully utilize the TFO functionality, a configured tool including inertia must be used. Avoid using tool0 when testing TFO.

The tool force is calculated from measurements of the joint torques, assuming a general force and torque acting at an arbitrary point on the tool. However, near the manipulator singularities, the calculation is changed to assume that the tool force is a pure linear force applied at the TCP (that is, the applied external torque about the TCP is zero). This assumption is necessary since the calculated force would otherwise become very inaccurate near singularities.

WARNING

Near singular configurations of the robot arm, the calculated force becomes inaccurate if the contact force is not a pure linear force acting at the TCP. This could result in cases where the true force is significantly greater than the force calculated and supervised by SafeMove, in particular when using a large tool with several different clamping points. During validation of TFO it must be ensured that the true force is below the desired limit in every potential clamping situation, in particular when the robot is close to a singularity, see *Tool Force Supervision validation on page 176*.

Note

For collaborative applications, use ISO/TS 15066 for guidance.

Settings

The following parameters can be configured for Tool Force Supervision:

- The limit of the external force applied on the tool.
- An optional limit for the external torque applied on each joint.
- Assignment of a safe input for activation, or set as permanently active.
- Stop category 1, or no stop if the force/torque limits are exceeded.

Continues on next page

- Set an output signal if the force/torque limits are exceeded.
- Set a status signal when the function is active.
- If the axis torque supervision should be included.

How to define these settings is described in *Configuring Tool Force Supervision on page 156*.

Note

When calculating the forces and torques, by default the *Tool Force Supervision* safety function accounts only for the tool mass (see *Configure the supervision functions on page 148*) but not any load held by the tool. Proper margins should therefore be included when setting the limits for the function, to compensate for any possible payload held by the tool.

If the application requires that the force calculation compensates also for the payload mass, the function *Safe Payload Supervision* (see *Safe Payload Supervision (SPA) on page 48*) must be used to update the payload held be the robot.

Note

This function is implemented for the reducing the risk in a clamping situation. The speed must be taken into consideration to prevent injury in case of a collision. This can be set up with assistance of the supporting function *Human Contact Supervision on page 47*.

When speed supervision is higher than 500 mm/s, the robot will have a different brake behavior. That is, it is never recommend to have speed supervision with a limit higher than 500 mm/s when there is a risk for clamping situation.

Note

The external force on the tool is calculated from the torques at each joint using a dynamic model of the robot. The accuracy of the calculated tool force depends on the current position of each robot joint, as well as the movement and dynamics of the robot.

Note

Near singular configurations of the robot arm, the robot might stop frequently with and event log (90544). It is recommended not to rely on the TFO force limit to avoid dangerous contacts near singularities, and instead to use TFO with axis torque limits or other safety functions such as TPO or APO. For a description of the different types of singularities, see *Technical reference manual - RAPID Overview*, section *Singularities*.

Continues on next page

Note

Due to the robot stopping distance, the tool force and axis torques will increase after TFO has triggered a stop. This must be taken into account when setting the limits for the function. Proper speed limits can be used to limit the increase in force/torque during the stop.

Note

In the presence of external forces acting on the robot arm, the accuracy of the calculated tool force is reduced.

To avoid external forces exceeding the configured tool force limit, the user must ensure that all external forces act exclusively on the tool, or use properly defined axis torque limits to limit the forces acting on the robot arm.

Function activation

Tool Force Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Force Supervision is only available for SafeMove Collaborative, see *Functional safety options on page 18*.
- The speed of the tool center point (TCP) must never exceed 250 mm/s.
- Near-singular configurations should be avoided.
- Positions where axis 4 and axis 6 are aligned and the robot pushes in flange-z direction should be avoided.
- A global Tool Force Supervision (TFO) has a high risk of causing false triggers. TFO should only be set up in zones where clamping can occur.
- Tool Force Supervision should be turned off when running on a virtual controller.

2.5.8 Control Error Supervision

Control Error Supervision

Control Error Supervision is a function that supervises the difference between the reference value and the measured value of the motor position of each axis. Control Error Supervision is required to ensure the accuracy in the supervision functions and the structural category 3 of the safety system, i.e. dual channel supervision.

Functionality

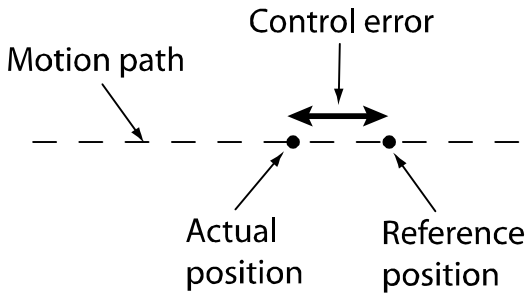
The control error (servo lag) is the absolute value of the difference between the reference value and the measured value of the motor position of each axis.

Control Error Supervision is activated automatically after the safety controller has been synchronized with the robot position.

When Control Error Supervision triggers, the following happens:

- The robot is stopped with a stop category 1.
- An event log message (90511) is sent to the robot controller.

Illustration of control error



en0700000723

Function activation

Control Error Supervision is always active. It can only be relaxed by Contact Application Tolerance.

Dependencies to other functions

If Contact Application Tolerance is active, then Control Error Supervision is relaxed according to user definitions.

2.5.9 Safe Disable of Drive Unit (SDI)

Safe Disable of Drive Unit

Safe Disable of Drive Unit is a function that can be used to disable a manipulator or additional axis, by safely setting its drive unit in a state with all brakes applied and servo control switched off. This can, for example, be used to prevent unexpected start-up.

Functionality

The function is activated by a safe input signal. It shall be used when there is no movement on the mechanical unit. If the function is activated while one or more axes of the mechanical unit are moving, a violation is triggered with a stop category 0 or stop category 1, depending on the configuration.

Settings

The following parameters can be configured for Safe Disable of Drive Unit.

- Assignment of safe inputs for activation of Safe Disable of Drive Unit.
- Stop category 0 or stop category 1 if a violation occurs, for example, attempting to disable the mechanical unit while it is moving.
- Set a status signal when the mechanical unit has been safely disabled. This signal can be used to read the status of the mechanical unit.

Function activation

Safe Disable of Drive Unit is activated by a safe input signal.

Limitations

- Safe Disable of Drive Unit cannot be used together with Contact Application Tolerance (CAP).
- Safe Disable of Drive Unit is only supported for controllers that do not have any axis computer.
- Safe Disable of Drive Unit is not available for CRB 15000.
- Safe Disable of Drive Unit cannot be used to disable mechanical units with axes controlled by more than one drive unit, such as IRB 8700.

This page is intentionally left blank

3 Installation

3.1 Software installation

Install RobotStudio

The configuration environment, *Visual SafeMove* is a free add-in in RobotStudio.

	Action
1	Install RobotStudio. Visual SafeMove is included in both the *Minimal* and *Full* version of RobotStudio.
2	Start RobotStudio.
3	Start Visual SafeMove by going to the **Controller** tab on the ribbon, click **Safety** and select **Visual SafeMove**.

For more information see *Operating manual - RobotStudio*.

Note

RobotStudio must be of the same version or later than the RobotWare used.

Modify RobotWare with SafeMove

A RobotWare license with the option *SafeMove Basic*, or *SafeMove Pro*, or *SafeMove Collaborative* is required to run SafeMove on the robot controller.

Use RobotStudio to modify the RobotWare system on the robot controller. A system with *SafeMove Collaborative* can also be configured using only the SafeMove configurator application on the FlexPendant.

When configuring the system in the **Modify RobotWare** function, select the checkbox **Enable Add-In Installation of Safety Parameters** if you plan to install safety data via add-ins. For example, to install protected groups using an add-in, see *Protected basic configuration on page 106*. For more information about creating add-ins, see *Application manual - RobotWare Add-Ins*.

For more information about how to modify a system, see *Operating manual - RobotStudio*.

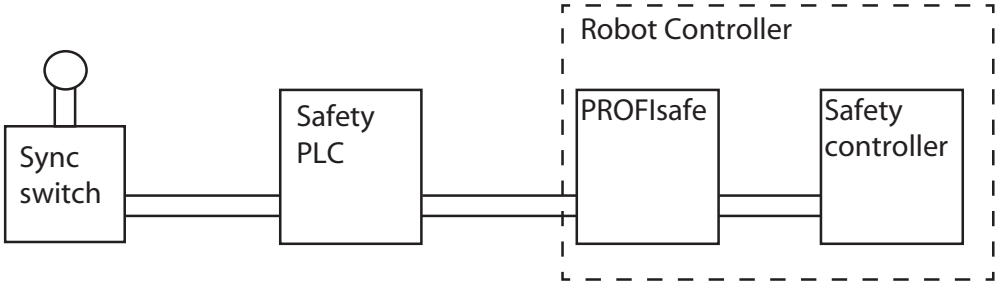
3.2 Hardware installation

Safety controller

The safety controller is part of the main computer in the robot controller, and is always installed at delivery. For more information, see the product manual and the circuit diagram for the robot controller. See *References on page 10*.

Sync switch connection

If hardware synchronization is used, a sync switch should be connected to a safety PLC and communicate with the robot controller via PROFIsafe.



xx1600000249

3.3 About safe fieldbuses

Introduction

A safe fieldbus must be used for all I/O communication with the safety module. The following safe fieldbus configurations are supported by the safety module:

- *PROFIsafe F-Device*
- *PROFIsafe F-Host*
- *ABB Scalable IO*
- *Safety Over EtherCAT Device*

For information on the corresponding options numbers, see *Functional safety options on page 18*.

Consider the safety of the entire solution

If a safety PLC is used, it is the responsibility of the installer to make sure the safety is maintained throughout the solution (e.g. signal redundancy, safe PLC programming, safe state at power failure, etc.).

DANGER

An emergency stop on the controller will not be forwarded to the safety PLC or any other equipment if the robot controller is disconnected.

DANGER

Lost communication between the robot controller and external safe fieldbus will not stop the robot, but any signals configured for the device will go low (0). This can be used to activate safety functions to stop the robot, for example by configuring SST.

DANGER

The emergency stop buttons on a disconnected robot controller must be clearly marked or covered.

CAUTION

The replacement of safety devices requires that the replacement device is configured properly, and the operation of the replacement device must be verified by the user.

Connections

The safe fieldbuses are connected directly to the Ethernet ports.

For more information on the connectors, such as placement and configuration, see the product manual for the robot controller and the application manuals for the industrial networks. All referenced manuals are listed in section *References on page 10*.

This page is intentionally left blank

4 The Visual SafeMove user interface in RobotStudio

4.1 About Visual SafeMove

What is Visual SafeMove

Visual SafeMove is the configuration tool for SafeMove and the functional safety options. The tool is completely integrated into the RobotStudio user interface and takes full advantage of the user interface elements such as tabs, browsers, and 3D graphics.

Visual SafeMove is enabled for robots with the safety module. It offers an intuitive way to visualize and configure safety zones. Zones can be adjusted by direct manipulation in the 3D window. Users with previous experience from SafeMove will recognize the same terminology used as before.

Visual SafeMove is used to configure safety stops. For this purpose, the SafeMove options are not required, that is, this functionality is available for all robots. More information about the configuration is available in the product manual for the robot controller.

Visual SafeMove works both with the real controller and the virtual controller. For a virtual controller, a RobotStudio station should be used, which allows zones to be generated automatically. When not running a RobotStudio station, **Online Monitor** is used to visualize the robot.

Starting Visual SafeMove

	Action
1	Start RobotStudio with a virtual controller (with or without a station) or connect a real controller.
2	In the **Controller** tab, click **Safety**, then select **Visual SafeMove**.

Prerequisites

- Some functionality is only available for SafeMove Pro, see *Functional safety options on page 18*.
- Only a user with the grant **Safety Services** is allowed to download a configuration. See *Set up safety user grants on page 130*.

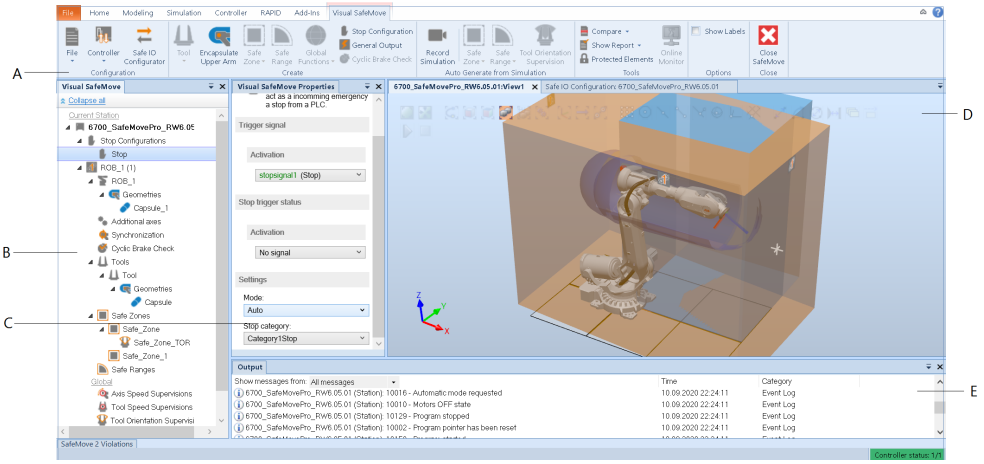
Limitations

The simulation functions are only available when running a RobotStudio station.

4.2 The user interface

Overview of the user interface

This section presents an overview of the Visual SafeMove graphical user interface.



xx2000001847

	Parts	Description
A	The **Visual SafeMove** ribbon	Displays groups of icons organized in a logical sequence of function.
	The **Modify** tab	Is used by some functions in the **Visual SafeMove** ribbon to display additional functionality.
B	**Visual SafeMove** browser	Displays all available SafeMove functions.
C	**Visual SafeMove Properties** browser	Displays all available properties and settings of the selected SafeMove function.
D	Graphics window	Is used to to visualize and configure safety zones in the RobotStudio station. When not running a RobotStudio station, **Online Monitor** is used to visualize the robot.
E	**Output** window	The output window displays information about events that occur in RobotStudio, both general events and Visual SafeMove events.
	SafeMove Violations window	Displays all violations that has occurred since the last restart. Click a violation in the list to show a detailed view of all geometries that are related to the violation. The geometries that are not related are automatically hidden. The violations window can also be used to view a violation when RobotStudio is connected to a real robot.

The tabs

The **Visual SafeMove** tab and the **Modify** tab contains groups of commands organized in a logical sequence of functions that simplifies the configuration of SafeMove, see *The Visual SafeMove tab on page 72* and *The Modify tab on page 80*.

Continues on next page

The browsers

The configured safety functions are available from the **Visual SafeMove** browser. When a function is selected by clicking the node in the browser, the properties and settings are displayed in the **Visual SafeMove Properties** browser, see *The Visual SafeMove browser on page 83*.

Use the **Visual SafeMove** browser, or standard keyboard shortcuts, to cut, copy, and paste zones both between zone types and between robots.

The graphics window

In general the Visual SafeMove graphics window is navigated using the same commands as in RobotStudio. A few additional navigation tools are available in Visual SafeMove for editing zones, those are listed below.

For more information on navigating RobotStudio, see *Operating manual - RobotStudio*.

Editing zones in the graphics window

Zones are displayed as semi-transparent (opaque) planes.

The following navigation options are available in Visual SafeMove for editing zones:

- Drag and drop on spheres in the graphics window (at corners and surfaces) to modify a zone.

  Tip

  To lock an axis while dragging, press **x** or **y** on the keyboard.

- Double click vertices or surfaces in the graphics window to create new corners/vertices at that location.
- Delete vertices that are no longer desired by selecting and deleting them from the graphics window.
- Move, rotate, and change the size of a zone by dragging the zone frame arrows in the graphics window.

Symbols in the graphics window

The following symbols are used in the graphics window to illustrate the properties of the zones.

 xx1600000241	 xx1600000242	 xx1600000243	 xx1600000244	 xx1600000245	 xx1600000246
Robot number	Upper arm included in Tool Position Supervision	Zone minimum speed limit	Zone maximum speed limit	Robot allowed inside zone	Robot not allowed inside zone

4.3 The Visual SafeMove tab

4.3.1 About the Visual SafeMove tab

Layout of the ribbon

The **Visual SafeMove** tab contains groups of commands organized in a logical sequence of functions that simplifies the configuration of SafeMove.

xx2000001831

The ribbon consists of the following groups:

Group	Functions used for
Configuration	Creating, saving and loading SafeMove configurations. *Configuration group on page 72*
Create	Adding and configuring SafeMove functions. *Adding SafeMove functions on page 75*
Auto Generate From Simulation	Creating a SafeMove function based on a simulated path in RobotStudio. *Simulating SafeMove on page 105* **Note** The simulation functions are only available when running a RobotStudio station.
Tools	Different tools for analyzing SafeMove configurations. *The Tools group on page 74*
Options	Display options for the Visual SafeMove user interface. *Options group on page 74*

Tip

Most commands in the ribbon are also available from the **Visual SafeMove** browser by right-clicking the nodes in the tree.

Configuration group

The **Configuration** group is used when creating, saving and loading SafeMove configurations.

Continues on next page

The safety configuration in RobotStudio that has not yet been written to the controller is referred to as "local configuration". The safety configuration on the controller is referred to as "controller configuration".

The **File** button

Button	Description
New / **Controller Configuration**	Create a new controller configuration.
New / **Drive Module Configuration**	Create a new drive module configuration. The new configuration will be in the latest format version.
Open configuration	Open a controller configuration. The format version of the opened configuration will be kept.
Save configuration as	Save the safety or geometry configuration to file. For more information about the geometry configuration file, see *SafeMove geometry configuration file on page 212*.
Import geometries	Imports safety geometries such as zones, tools and robot upper arm geometry from a safety configuration file or a simplified configuration file generated by an external tool. Geometries with the same name will be overwritten.
Import protected elements	Imports the protected elements from the selected file as a group, including the checksum for the group.

The **Controller** button

Button	Description
Read from controller	Reads the configuration from the safety controller.
Write to controller	Writes the configuration to the safety controller.
Upgrade configuration to latest version	Upgrades the safety configuration to the latest format version. The checksums will be updated.
Reset to factory settings	Resets the configuration in the safety controller to default settings. See also *Reset safety controller to factory settings on page 183*.
Restore configuration	Restores the configuration from file. The file is restored as it is. It is not opened in RobotStudio.

The **Safe IO Configurator** button

The **Safe IO Configurator** button is used to start the SafeMove IO configurator.

The safe IO configurator is described in a separate chapter, see *The Safe IO Configurator on page 90*.

Continues on next page

The Tools group

The **Tools** group contains different tools for analyzing and protecting SafeMove configurations.

Button	Description
Compare	The **Compare** button starts a tool that is used to compare the differences between two configuration files. This button provides the following. • **Compare two configurations**: Compare two selected configuration files. • **Compare with RobotStudio configuration**: Compare the selected configuration file with configuration file in the RobotStudio user interface. • **Compare with controller configuration**: Compare the selected configuration file with the controller configuration file.
Show report	Shows the report for the safety configuration. This button provides the following options • **Controller configuration**: Shows the safety configuration report for the current controller configuration. • **RobotStudio configuration**: Shows the safety configuration report for the configuration available in the RobotStudio user interface. To show the local safety configuration report, click **Controller** and select **Write to controller**. The local report is shown and if you do not want to write it to the controller, select **Cancel**. Select **Write to controller** to overwrite the controller configuration with the RobotStudio configuration.
Protected Elements	The **Protected Elements** button opens the **Protected Elements Group Editor** browser. From here, it is possible to create protected groups of elements in the SafeMove configuration, so that the user cannot change them. It is possible to create configuration templates to import to a new system, see *Protected basic configuration on page 106*. The write-protected element are visualized with a padlock icon in the **Visual SafeMove** browser.
Setup Laser Scanner	The **Setup Laser Scanner** button opens a guide for setting up laser scanners for collaborative robots. This is currently only available for the GoFa robot. See *Configuring Laser Scanner Calculation on page 165*.

Options group

The **Options** group contains display options for the Visual SafeMove user interface.

Checkbox	Description
Show Labels	Displays labels in the graphics window.

4.3.2 Adding SafeMove functions

About the Create group

The **Create** group is used when creating and configuring SafeMove functions.

Tip

Most commands in the ribbon are also available from the **Visual SafeMove** browser, by right-clicking the nodes in the tree.

The Tool button

Button	Description
New	Create a new tool.
Encapsulate	Creates a geometry that encapsulates the selected tool. Up to four different geometries can be used to create a more accurate encapsulation of the tool.

Settings

The following settings are available in the **Visual SafeMove Properties** browser after the tool has been created.

Setting	Description
Make default Tool	If more than one tool is configured, one of them must be selected as default tool. The default tool will be used at startup of the controller when no tool is selected by input signals. The last selected tool will still be active if the signal combination becomes faulty. The tool that already is the default tool has a greyed out button with the text **Is default Tool**.
Activation	Specifies the safety signal that activates the selected tool. The setting **Permanently active** is used for a tool that is always active. This setting needs to be deactivated to be able to create more than one tool.
Function active status	Specifies a safety signal that indicates that the selected tool is active. The setting **No signal** is used for a tool that is always active.
Tool data	Modify the tool data, or load a tool data from the RobotStudio station. **CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
Speed supervised points	The tool can use up to eight points to supervise the speed of the tool. By default all eight points are created. It is recommended to reduce the number of points if all points are not required depending on the shape of the tool.

Continues on next page

The Encapsulate Upper Arm button

The **Encapsulate Upper Arm** button is used to create a geometry that encapsulates the upper arm. The geometry can then be adapted to fit the needs of the current application.

Up to two different geometries can be used to create a more accurate encapsulation of the upper arm.

The Safe Zone button

The **Safe Zone** button is used to create safe zones. The safe zone is displayed in the graphics window as a rectangular box with vertices.

Button	Description
Safe Zone	Creates a default safety zone in the shape of a rectangular box.
Wrapped Safe Zone	Crates a safe zone by wrapping a part in the RobotStudio station. The part must first be selected.

After a zone has been created, the next step is to add safety functions to the zone. This is described in section *The Modify tab on page 80*.

Tip

It is recommended to change the default name of the safe zone in the **Visual SafeMove** browser to a name that better corresponds to the current installation.

Settings

The following settings are available in the **Visual SafeMove Properties** browser after the zone has been created.

Setting	Description
Tool Speed Supervision Priority	Used to set the priority for overlapping zones. The zone with the highest priority will set the speed limit for the overlapping space. • **BASE** - lowest priority. • **NORMAL** - normal priority. • **OVERRIDE** - highest priority. For more information, see *Configure the zones on page 143*. **Tip** Create a **BASE** zone with a low speed limit that covers the entire robot cell. This zone is then used to protect the fence. Add **NORMAL** zones for the areas where the robot performs most of the work. Additionally, add **OVERRIDE** zones for any areas where the robot is allowed to move fast.
Reference	The reference coordinate system in the RobotStudio station. **Note** The base frame cannot be used for reference if it is rotated around the controller world x or y-axes, for example when using a tilted robot.
Height	The height of the zone.

Continues on next page

Setting	Description
Vertices	Displays the vertices of the safe zone. It is possible to add or delete vertices from the **Visual SafeMove Properties**.  **Tip** Add a vertex by double-clicking the edge of a zone in the graphics window. Delete a vertex by first selecting the vertex in the graphics window and then pressing delete.

The Safe Range button

Click the **Safe Range** button to create a safe range. The safe ranges are displayed in the graphics window as a discs around the rotation centers of the selected axes.

After a range has been created, the next step is to add safety functions to the range. This is described in section *The Modify tab on page 80*.

Settings

The following settings are available in the **Visual SafeMove Properties** browser after the range has been created.

Setting	Description
Joint	The joints of the robot.
Enabled	Safe range is enabled for the joint.
Lower bound	The lower bound limit in degrees.
Upper bound	The upper bound limit in degrees.
Invert	If selected, the supervised angles for that axis will be below the **Lower bound** and above the **Upper bound**.

The Global Functions button

The **Global Functions** button is used to create global safety functions. Global safety functions are always active and are not connected to a specific safe zone or safe range.

The below functions, except Stand Still Supervision, can also be assigned to individual safe zones and safe ranges.

The following global functions are available:

Button	Description
Global Tool Orientation Supervision	*Tool Orientation Supervision (TOR) on page 58*
Global Tool Speed Supervision	*Tool Speed Supervision (TSP) on page 53*
Global Axis Speed Supervision	*Axis Speed Supervision (ASP) on page 52*
Stand Still Supervision	*Stand Still Supervision (SST) on page 50*

Settings

The settings for the global safety functions are the same as for the safe zones and safe ranges. For information about the settings see *The Add Safety Function group on page 80*.

Continues on next page

The Stop Configuration button

The **Stop Configuration** button is used configure a safety stop that is sent on the safe fieldbus from the safety PLC to the robot controller.

The **Mode** setting defines which stop mode that shall be activated on the robot controller (General Stop, Automatic Stop, or Emergency Stop). The corresponding warning will be written to the event log and presented on the FlexPendant.

Note

Depending on the controller variant and RobotWare version, the configuration options are different. See the product manual for the controller.

Not all configurations can be modified.

Setting	Description
Trigger signal	Specifies the safety signal that activates the safety stop. The signal is set to 0 for activation. The trigger signal can only be selected for user-created stop configurations.
Stop trigger status	Specifies a safety signal that indicates that the the safety stop is active. The signal is set to 0 when triggered. The setting **No signal** is used if no status signal should be used. The trigger status signal can only be selected for user-created stop configurations.
Settings • **Mode**	The **Mode** setting defines which stop mode that shall be activated on the robot controller. The corresponding warning will be written to the event log and presented on the FlexPendant.
Settings • **Stop category**	• **Category0Stop** - Stop by immediate removal of power to the actuators. Mechanical brakes are applied. A robot that is stopped with a stop category 0 does not follow its programmed path while decelerating. • **Category1Stop** - Controlled stop with power available to the actuators to achieve the stop. Power is removed from the actuators when the stop is achieved. A robot that is stopped with a stop category 1 follows its programmed path while decelerating.

The Cyclic Brake Check button

The **Cyclic Brake Check** button is used to setup the cyclic brake check function.

For more information, see *Cyclic Brake Check guidelines on page 200*.

Setting	Description
Warning only, no stop	If this checkbox is selected, the robot will not stop if a cyclic brake check has not been performed on time. Only a warning will be written to the event log and presented on the FlexPendant.
Max CBC test interval	The maximum allowed time between cyclic brake checks.
Pre warning time	The warning time before a cyclic brake check must be performed.
Standstill tolerance	The maximum allowed movement during a cyclic brake check.
Supervision threshold	The minimum servo lag used for detecting that the cyclic brake check is performed.

Continues on next page

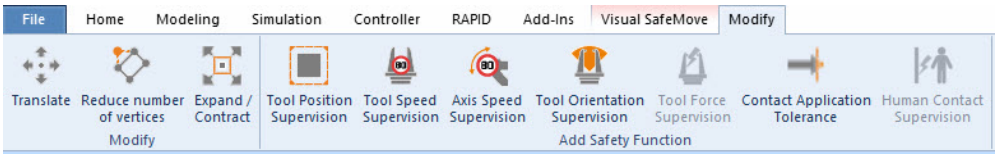
Setting	Description
ROB1	If a checkbox is deactivated, then cyclic brake check is deactivated for that joint.

4.4 The Modify tab

Layout of the ribbon

The **Modify** tab is a sub-tab to the **Visual SafeMove** tab and contains additional functions and settings for the selected object.

The functions are different depending on which object that is selected. The below picture show the available functions for a safe zone.



xx2000001830

Tip

Most commands in the ribbon are also available from the **Visual SafeMove** browser, by right-clicking the nodes in the tree.

The Modify group

The **Modify** group is only available when modifying a safe zone.

Button	Description
Translate	Translates the zone.
Reduce number of vertices	Reduces the number of vertices for the zone without reducing the area covered by the safe zone. Note that the shape of the safe zone will change, and possibly grow, but it will still cover the original area.
Expand/Contract	Expands the zone in all directions except the height.

The Add Safety Function group

The **Add Safety Function** group is used when creating and configuring safe zone and safe range safety functions.

The following functions are available for both safe zones and safe ranges:

Button	Description
Axis Speed Supervision	*Axis Speed Supervision (ASP) on page 52*
Contact Application Tolerance	*Contact Application Tolerance (CAP) on page 37*
Tool Orientation Supervision	*Tool Orientation Supervision (TOR) on page 58*
Tool Speed Supervision	*Tool Speed Supervision (TSP) on page 53*

The following function is only available for safe zones:

Button	Description
Tool Position Supervision	*Tool Position Supervision (TPO) on page 57*

Continues on next page

The following function is only available for safe ranges:

Button	Description
Axis Position Supervision	*Axis Position Supervision (APO) on page 55*

The following function is only available as a global function:

Button	Description
Stand Still Supervision	*Stand Still Supervision (SST) on page 50*

Note

For information about global functions, see *The Global Functions button on page 77*.

Settings

The following settings are available in the **Visual SafeMove Properties** browser, and are common for all safety functions.

Setting	Description
Activation	Specifies the safety signal that activates the supervision. The signal is set to 0 for activation. The setting **Permanently active** is used for a supervision that is always active.
Function active status	Specifies a safety signal that indicates that the selected safety function is active. The signal is set to 1 when active. The setting **No signal** is used if no signal should be used.
Violation action • **Stop type**	• **Category0Stop** - Stop by immediate removal of power to the actuators. Mechanical brakes are applied. A robot that is stopped with a stop category 0 does not follow its programmed path while decelerating. • **Category1Stop** - Controlled stop with power available to the actuators to achieve the stop. Power is removed from the actuators when the stop is achieved. A robot that is stopped with a stop category 1 follows its programmed path while decelerating. • **NoStop** - Passive monitoring with signaling function only.
Violation action • **Signal**	Specifies a safety signal that indicates that the the axis speed supervision is violated. The signal is set to 0 at violation. The setting **No signal** is used if no violation signal should be used. **Note** When a signal is set to 0 at violation, it will remain 0 for at least 250 ms even after the violation has ended.

The following settings are available in the **Visual SafeMove Properties** browser, and are individual for each safety function.

Function	Setting	Description
ASP	**Speed limits**	The minimum and maximum speed for each robot joint.

Continues on next page

4.4 The **Modify** tab

Continued

Function	Setting	Description
TOR	**X Vector** **Z Vector**	The tolerance around the x and z axis of the tool.
	Get vectors for active tool (button)	Gets the vectors from the active tool in RAPID. The tool tip shows which tool is active. **CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
TSP	**Speed limits**	The minimum and maximum speed for the tool.
SST	**Tolerances**	Specifies if Stand Still Supervision is enabled and the tolerance for each robot joint.

4.5 The Visual SafeMove browser

About the Visual SafeMove browser

When adding a safety function from the **Visual SafeMove** tab or the **Modify** tab, it is automatically displayed in the **Visual SafeMove** browser. These functions and settings are described in the corresponding section where they are added.

In addition to that, the **Visual SafeMove** browser also has some system related settings that are described in this section.

Robot properties node

Setting	Description
Max speed in manual mode	The max speed cannot be higher than the default value of 250 mm/s, but a lower value can be set.

Note

If reducing the max speed that SafeMove allows in manual mode, the jogging speed of the robot has to be reduced to the same value. Change the parameter *Teach Mode Max Speed*, topic *Motion* and type *Motion Planner*. See *Technical reference manual - System parameters*.

Stop Configurations node

The stop configuration for OmniCore is done in Visual SafeMove. This does not require any SafeMove license. More information about the configuration is available in the product manual for the robot controller.

Robot parameters node

Setting	Description
Elbow offset • **Position X, Y, Z**	If any extra equipment is attached to the upper arm, a point on this equipment can be defined as a check point. The robot will then monitor the speed of this point so that it does not exceed 250 mm/s in manual reduced speed mode. See *Explanation of Elbow offset on page 84*.
Safe Brake Ramp Data • **Start Speed Offset**	A speed offset for the Safe Brake Ramp function. For track motions and other additional axes, the parameters **Brake Ramp Limit** and **Ramp Delay** have to be set. The parameter **Start Speed Offset** is used for both manipulator and all additional axes, see *Explanation of Safe Brake Ramp on page 88*.

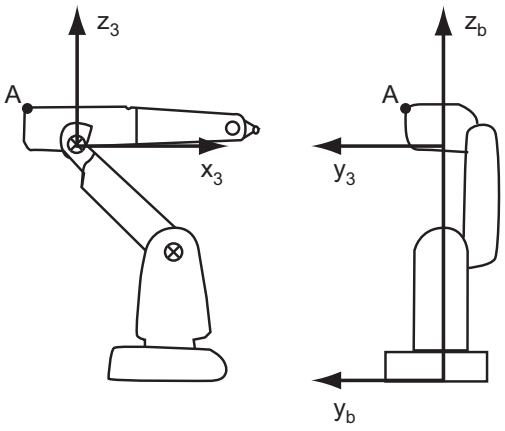
Continues on next page

The following settings are available when right-clicking the robot parameters node:

Setting	Description
Basic joint supervision mode	Set Basic joint supervision mode for the robot. This setting is used with some combinations of robots and external axes that are normally not supported by SafeMove, such as robots mounted on a gantry or other non-track external axes. In Basic joint supervision mode, no safe zones or tool supervision functions are allowed. See *Overview of SafeMove functions on page 33*.
Exclude from configuration	Excludes the robot from the safety configuration. This setting must be used when configuring a safe fieldbus on robots that are not supported by SafeMove. This setting can also be used to exclude supported robots, for example individual robots in a MultiMove application and robots running only a safe fieldbus. See *Overview of SafeMove functions on page 33*.  **DANGER** A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

Explanation of **Elbow offset**

An elbow point is considered for Tool Speed Supervision. The elbow point is given a default value based on robot model. This value can be changed in the configuration. Specify the elbow point's x, y and z offsets relative to the center of robot axis 3. Note that the X value should always be negative.



xx1300002628

A	Elbow point
x_3, y_3, z_3	Axis 3
y_b, z_b	Robot base

Continues on next page

Note

The values that are entered into **Elbow Offset** should also be entered into the parameter *Arm Check Point*, topic *Motion*. This is to avoid different speed calculations between the robot controller and SafeMove in manual reduced speed mode.

For more information, see *Technical reference manual - System parameters*.

Explanation of **Base Frame**

All values for the base frame are automatically loaded from the robot controller and cannot be changed from the **Visual SafeMove** browser.

Setting	Description
Base frame • **Reference**	Zone can be defined in either task frame, world coordinate system, user coordinate system, or robot base frame. These coordinate systems are often identical, but for MultiMove systems it may be desired to do the configuration in the robot base frame.
Base frame • **Position X, Y, Z**	X, Y and Z values for the selected reference frame's origin, expressed in the world coordinate system.
Base frame • **Orientation**	Defines the orientation of the selected reference frame, compared to the world coordinate system.

Gravity parameters for rotated robots

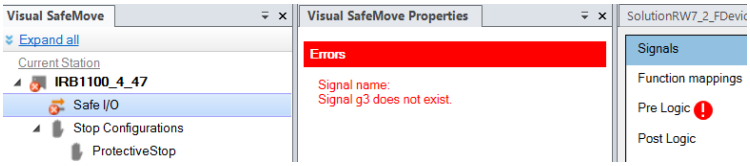
Inverted robots with power force limiting functionality, must have the gravity parameters in the motion configuration correctly defined.

Safe Disable of Drive Unit parameters

Setting	Description
Activation signal	The signal used to disable the drive unit of the robot.
Status signal	The signal used to indicate that the drive unit of the robot has been safely disabled. There will be a delay of 500 ms between the activation is triggered and the status is changed.
Stop category	Stop category 0 or category 1 in case of a violation, for example, if the robot is moving when the function is activated.

Safe I/O node

The **Safe IO Configurator** window can be opened from the **Safe I/O** node. Right-click the **Safe I/O** node and then click **Safe IO Configurator** from the context menu. The Visual SafeMove Properties browser also displays errors in the Safe I/O configuration.

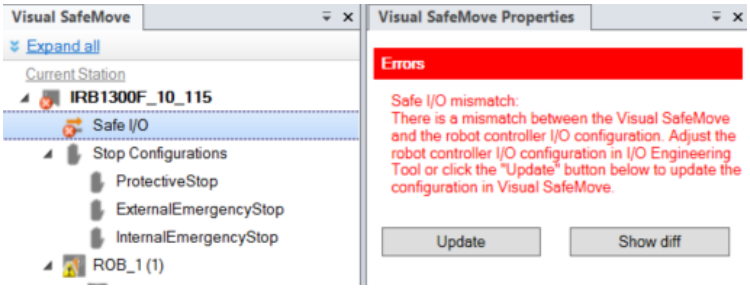


xx2000002721

Devices, modules and signals from internal and external devices are configured using the I/O Engineering tool and written to the controller I/O configuration. This

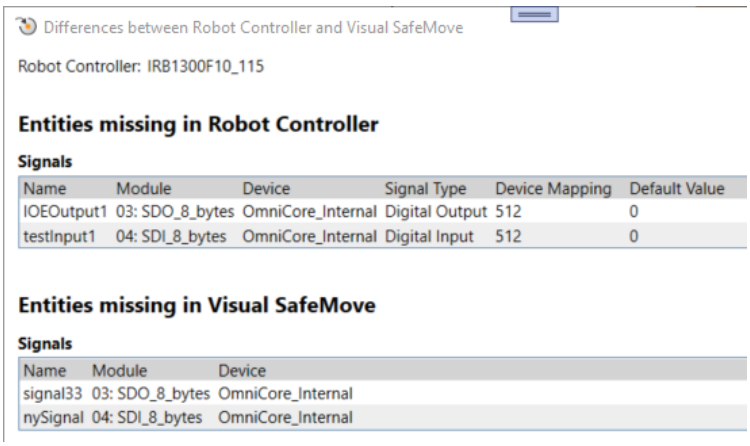
Continues on next page

will result in a mismatch between the controller I/O configuration and the Visual SafeMove configuration, which can be visualized and then updated from the **Safe I/O** node.



xx2000002719

The differences between the two configurations can be updated according to the System Parameters. To view the difference between these configurations, click the **Show diff** button and to update the Visual SafeMove configuration, click the **Update** button.



xx2000002720

Use the following procedures to remove mismatches to ensure that the robot controller configuration and the Safety Controller configuration are synchronized.

Update Visual SafeMove according to robot controller I/O configuration

1. In the **Visual SafeMove** browser, select the **Safe I/O node** to open the **Visual SafeMove Properties** window.
2. In the **Visual SafeMove Properties** window, click the **Update** button to update the safety configuration according to the controller I/O configuration.
3. In the **Configuration** group, in the **Controller** drop-down menu, click **Write to controller**.

Update robot controller I/O configuration according to Visual SafeMove

1. In the **Visual SafeMove** browser, select the **Safe I/O node** to open the **Visual SafeMove Properties** window.
2. In the **Visual SafeMove Properties** window, click **Show diff** to view the differences between the controller I/O configuration and Visual SafeMove.

Continues on next page

3 Open the I/O Engineering Tool and manually update the configuration according to the **Differences between Robot Controller and Visual SafeMove** window.
4 In the **Access** group, click **Write config** to write the changes to the robot controller.

Additional axes parameters node

If the axis should be part of the SafeMove supervision, select the check box **Is supervised**.

Joint parameters

Setting	Description
Servo lag	**Servo lag** is the estimated lag (in radians on motor side) for the additional axis. For more information, see *Servo Delay Factor and Servo Lag on page 210*.
Servo delay factor	Estimated delay factor between reference position and measured position (number of 4 ms units) when moving the additional axis. (See TuneMaster, signal number 17 and 18.) For more information, see *Servo Delay Factor and Servo Lag on page 210*.
Max speed manual mode	The maximum speed in manual mode.

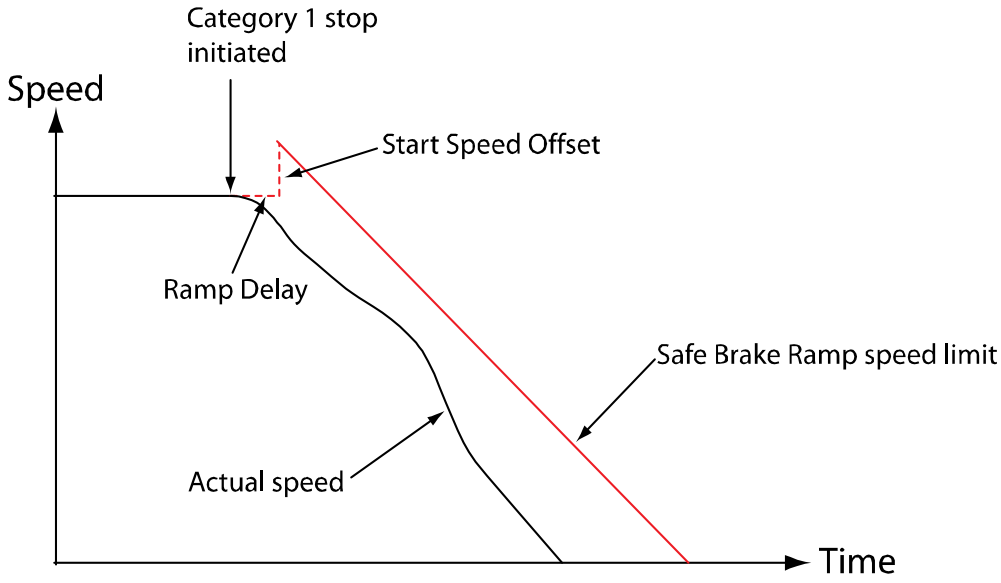
Brake ramp supervision parameters

Setting	Description
Safe brake ramp enabled (check box)	Selects if safe brake ramp should be used for the additional axis.
Ramp delay	Delays the Safe Brake Ramp function. See figure below. Default value: 200 ms.
Brake ramp limit	If the actual deceleration is lower than the specified **Brake Ramp Limit**, then Safe Brake Ramp will cause a stop category 0. The value is specified for the arm side.
Start speed offset	A speed offset for the Safe Brake Ramp function.

Continues on next page

Explanation of Safe Brake Ramp

Safe Brake Ramp supervises that a stop category 1 decelerates fast enough. To avoid that Safe Brake Ramp triggers every time, some margins are necessary. For additional axes, it is possible to set the margins both in time and start speed, and the slope of the speed limit ramp. For the robot, it is only possible to set the margin for the start speed.



en0700000724

Safe Disable of Drive Unit parameters

Setting	Description
Activation signal	The signal used to disable the drive unit of the additional axis.
Status signal	The signal used to indicate that the drive unit of the robot has been safely disabled. There will be a delay of 500 ms between the activation is triggered and the status is changed.
Stop category	Stop category 0 or category 1 in case of a violation, for example, if the additional axis is moving when the function is activated.

DANGER

For additional axes that are mechanical units with multiple joints, every joint required to be disabled for personal safety must be configured for Safe Disable of Drive Unit. The reason is that each joint is controlled by a separate additional drive unit (ADU). Although activating Safe Disable of Drive Unit for one joint sets the entire mechanical unit in a state where motion is prevented, only the joints for which Safe Disable of Drive Unit is active can be relied on for personal safety. Therefore, it is recommended to configure the same activation signal for all joints in the mechanical unit.

Continues on next page

Synchronization node

Setting	Description
Activation	To use software synchronization, select **Software synchronization**. To use hardware synchronization, select the input signal from the sync switch.
Synchronization status	Specifies a safety signal that indicates that the safety controller is synchronized The signal is set to 1 when synchronized. The setting **No signal** is used if no signal should be used.
ROB1 • **Joint** • **Position**	The axis position values of the robot.
Additional axis • **Joint** • **Position**	The axis position value of the additional axis.

4.6 The Safe IO Configurator

4.6.1 Introduction

About the Safe IO Configurator

The **Safe I/O Configurator** has the following main purposes:

- View safe fieldbuses and safe I/O signals.
- Connect signals to general purpose safety functions.
- Configure combinatory logic.

To open the **Safe IO Configurator**, on the **Visual SafeMove** tab, in the **Configuration** group, click **Safe IO Configurator**.

Alternatively, in the **Visual SafeMove** browser, right-click the **Safe I/O** node and then click **Safe IO Configurator** from the context menu.

The **Safe IO Configuration** window consists of the following views:

View	Description
Signals	Configuration of safety signals for safe fieldbuses is done in **I/O Engineering**, while global (virtual) safety signals are configured in **Safe IO Configurator**.
Function Mappings	Configuration for mapping of signals to specific safety status information and functions.
Pre Logic	Configuration of combinatory logic for the safety signals that is to be executed before the safety application (AND, OR, XOR, etc.).
Post Logic	Configuration of combinatory logic for the safety signals that is to be executed after the safety application (AND, OR, XOR, etc.).

For instructions on how to perform the configuration, see *Configure safe I/O on page 141*. For information on general rules and limitations see *Safe I/O system rules and limitations on page 104*.

Signals for commissioning mode

Input signals selected for commissioning mode are active (high) during commissioning mode.

This means that supervision functions and stop functions will not be triggered or activated by safety signals when commissioning mode is active, when trigger or activation signals are selected for commissioning mode.

4.6.2 Signals view

Introduction

The **Signals** view contains configuration of global safety signals (virtual signals) to be used internally in RAPID and in the safety controller, see *Global signals on page 91*.

The safety signals can be read from RAPID, but they can only be set from the safety controller, see *Accessing safe signals and feedback signals on page 92*.

For more information on how to configure safe I/O, see *Configure safe I/O on page 141*.

Global signals

The global signals are virtual signals that are to be used internally in RAPID and in the safety controller. The user can create up to 2048 global signals. The signals can be read from RAPID, but they can only be set from the safety controller, see *Accessing safe signals and feedback signals on page 92*.

There is a set of predefined global signals that corresponds to safety functions with the same names. It is allowed to change the names of these signals as long as the signal is still mapped to the corresponding safety function. It is also allowed to delete them as long as the function is replaced by another signal.

Predefined global signals

The following global signals are predefined and corresponds to safety functions with the same names:

- AutomaticMode
- BrakeReleaseActivated[I]
- BrakeReleasePending[I]
- DriveEnable
- DriveEnableFeedback
- EmergencyStopActivated
- EnableSwitch
- ExternalEmergencyStop
- LocalEmergencyStop
- ManualFullSpeedMode
- ManualMode
- ProtectiveStop
- SafetyEnable

I Only for GoFa

For a description of the safety functions, see *Description of safety functions on page 93*.

Continues on next page

4.6.2 Signals view
Continued

Accessing safe signals and feedback signals

The input, output, and global signals are automatically made available in RAPID. The signals can be found together with all other I/O-signals in the I/O-configuration views both in RobotStudio and on the FlexPendant.

The signals can be read from RAPID, but they can only be set from the safety controller.

The global signals are made available in RAPID as feedback signals. The feedback signals are inputs located on the simulated I/O device *SC_Feedback_Dev* on the simulated bus *SC_Feedback_Net*.

Safe signals

See *Application manual - I/O Engineering* for information about the safe input and output signals for the safe fieldbuses.

4.6.3 Function mappings

Configuring function mappings

Some safety functions, for example the system modes, need defined safety signals to interact with the rest of the system.

To allow the user to choose their own name for signals, these safety functions use predefined signal aliases instead, thus ensuring that the correct signal is read.

The role of the function mappings is to connect safety signals to aliases.

Setting	Description
Function	The name of the safety function or state.
Direction	Defines if the safety function writes the signal or reads the signal. → Writes Safety function writes to signal. ← Reads Safety function reads signal.
Signal	The signal that is currently mapped to the function (input, output, or global).
Source	The source gives information on which part of the system writes value to the signal. This can be either **Safe local I/O**, **Safety system**, or user defined. The source also gives information on when a signal/alias gets its value updated, i.e. **Safe local I/O** is first and **Safety system** is after the pre-logic.
Mandatory	True or false. Some mappings are mandatory since they are used by the safety controller.
Description	See *Description of safety functions on page 93*.
Errors	Displays error, for example when the mapping for a mandatory function is missing.

Note

ExtComShutdownAck and *SafetyEnable* has no source if they are not mapped to a signal which is written. These signals should be written to before the safety application (e.g. device input or from pre-logic) or used with their default values.

Description of safety functions

The following safety functions are available:

Function	Description
AutomaticMode	True if **Automatic mode** is selected.
CommissioningModeActive	True if **Commissioning mode** is selected. Use this to check, in the robot controller I/O system, if **Commissioning mode** is active.

Continues on next page

4.6.3 Function mappings
Continued

Function	Description
ConfigurationLocked[i]	True if configuration is locked. Can, for example, be used as extra precaution by connecting to a PLC that disables robot operation in automatic mode when the configuration is unlocked.
DriveEnable	True if power is enabled to the actuators. DriveEnable is the safety controllers way of ensuring that there is only power enabled to the actuators when there are no safety violations. This is done through the superior stop input on the panel board.
DriveEnableFeedback	True if the power to the motors is enabled.
EmergencyStopActivated	False if any emergency stop has been triggered. This can be used to send the local emergency stop status to other devices.
EnableSwitch	The status of the connected three-position enabling switch.
ExtComShutdownAck	Only used only with ExtComShutdownReq. If True, the safety controller is allowed to shut down before 1 second has passed after the ExtComShutdownReq was set. For example, this can be set to True by the PLC when preparations have been made after ExtComShutdownReq is set to True.
ExtComShutdownReq	Set to True when the safe external communication is about to be terminated, beginning in no less than 1 second. This will happen in case of controlled shutdown, for example during robot controller shutdown.
ExternalEmergencyStop-Status	The status of the emergency stop input on the front panel. False indicates that emergency stop is active.
LocalEmergencyStopStatus	False if the emergency stop on the FlexPendant is pressed. This can be used to send local emergency stop status to other devices.
ManualFullSpeedMode	True if manual full speed mode is selected.
ManualMode	True if manual mode is selected.
ProtectiveStop	The status of the protective stop input on the front panel. False indicates that the protective stop is active.
SafetyControllerOperational	True if the safety controller is running without errors. It is False during start-up before SafetyEnable is set to True, during shutdown of the controller or if there is a critical failure of the safety controller. **CAUTION** When SafetyControllerOperational is set to 0 all supervisions are disabled. This will happen during startup and shutdown of the controller and also if, for example, a critical error occurs inside the safety controller, or during feedback disconnect for paint robots.
SafetyEnable	The SafetyEnable function should, when possible, be connected to a safe input signal from, for example, a PLC. This signal shall be set to True as soon as communication is up and running. When False, all signal activated supervision functions, such as stop configurations, will be deactivated, and DriveEnable will be set to False, disabling robot motion.
ServiceModeActive[i]	True if **Service mode** is activated.

Continues on next page

Function	Description
Stop0Status	False if a stop category 0 is active.
Stop1Status	False if a stop category 1 is active. Note that this will be a very brief indication since a stop category 1 is converted to a stop category 0 when all robot motion has ceased.

i Only used for option *SafeMove Basic* or *SafeMove Pro*.

4.6.4 Pre-logic and post-logic

Configuring combinatory logic

The combinatory logic has two layers:

- Pre-logic - executed before the safety application
- Post-logic - executed after the safety application

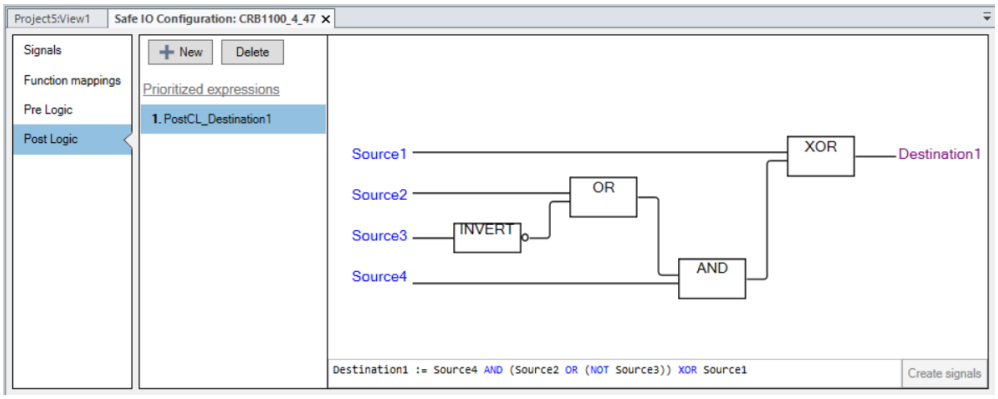
The user defines logic diagrams by writing statements. The combinatory logic operators can be grouped into logical operators, arithmetic operators, and complex operators.

For more detailed information including rules and limitations, see *Safe I/O system rules and limitations on page 104*.

Creating statements

Operators can be combined into complex statements.

Global signals can automatically be created from the statements by clicking **Create signals**.



xx2300000727

Note

The combinatory logic expressions are always executed by name order, so if there are dependencies between expressions (where output from one is used as input for another), make sure that they are executed in the intended order, by setting appropriate names.

The sort order will not handle values with different number of digits so EXPR_17 will be executed before EXPR_2. Therefore, if using numbers to set the order, make sure that the same number of digits are used, for example, name them EXPR_17 and EXPR_02 instead.

When creating expressions, they are shown in the order in which they are created. When loading saved expressions, they are shown in name order.

Continues on next page

Note

Signals between blocks are created internally and will not be available to the user. For example the signal between the *OR* and the *AND* block in the above picture.

Tip

First use the button **Create signals** to automatically create global signals from the statement, then use cut and paste to move signals to safe inputs or outputs.

Tip

When viewing large statements, press the left mouse button to pan and use the scroll wheel to zoom.

Overview of the combinatory logic operators

The following operators can be used:

Logical (Boolean) operators

The logical operators work only on actuators and resultants of type `bool`.

Operator	Description	Syntax example
AND	Logical AND operator. 1 if all source signals are 1.	Destination := Source1 AND Source2 ... AND Source8
OR	Logical OR operator. 1 if at least one of the source signals is 1.	Destination := Source1 OR Source2 ... OR Source8
XOR	Logical exclusive OR operator. 1 if odd number of source signals are 1.	Destination := Source1 XOR Source2 ... XOR Source8
NOT	Inverted value	Destination := NOT Source

Arithmetic (integer) operators

The arithmetic operators work only on actuators and resultants of type `INT32`.

Operator	Description	Syntax example
+	Add integer value.	Destination := Source1 + Source2
-	Subtract integer value.	Destination := Source1 - Source2
*	Multiply integer value.	Destination := Source1 * Source2
/	Divide integer value. See also the operator DIV. **Note** Division by zero is undefined. If the denominator is zero, then the quotient keeps its current value.	Destination := Source1 / Source2

Continues on next page

4.6.4 Pre-logic and post-logic
Continued

Operator	Description	Syntax example
<, <=, ==, >, >=, !=	Compares two integer values and produces a Boolean result. (smaller than, smaller/equal, equal, larger than, larger/equal, not equal)	Destination := Source1 < Source2 Destination := Source1 != Source2 ...
ABS	Absolute value of integer.	Destination := ABS Source
COPY	Copy integer value. Copy integer value from source to up to eight destination signals.	COPY(Source, Destination1, Destination2,...,Destination8)

Complex operators

These operators use a mix of `bool` and `INT32` actuators and resultants. For a more detailed description, see *Description of the complex operators on page 99*.

Operator	Description
COUNT	Counts number of pulses.
DELAY	Timer
DECODE4LOW	Integer to Boolean converter. Decodes the binary value when for example selecting a tool.
DECODE4HIGH	Integer to Boolean converter. Decodes the binary value when for example selecting a tool.
DIV	Divides two integer values with boolean error flag when dividing with zero.  **Note** Division by zero is undefined. If the denominator is zero, then the quotient keeps its current value.
EDGE	Filters signal on edge. 0 when trigger signal is 0. Stays 0 until trigger signal is set to 1 followed by the reset signal set to 1.
MUX	Multiplexer between two integer values.
REM	Remainder, modulo, after division with boolean error flag when dividing with zero.

Continues on next page

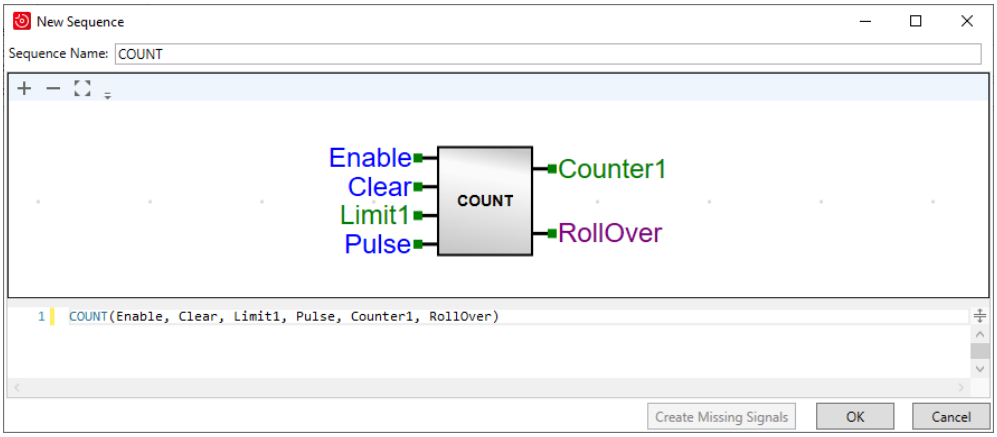
4.6.4.1 Description of the complex operators

Operator COUNT

This operator counts pulses on an actuator. If the Boolean actuator `enable` is inactive then this operator does nothing. The integer resultant `count` retains its current value.

If `enable` is active then `count` counts the number of inactive-to-active transitions of the Boolean actuator `pulse` since the last active-to-inactive transition of `clear`.

The integer resultant `count` is limited by the value of the integer actuator `limit`. When `count` reaches `limit`, then `count` will restart from 0 (zero) on the next activation of the Boolean actuator `pulse`. When the resultant `count` is restarted from zero the `rollover` resultant is activated and then deactivated on the following execution.



xx2300000728

Operator DELAY

The delay operator is similar to the count operator, but instead of counting pulses on an actuator it counts execution loops (`rollover`).

If the Boolean actuator `enable` is inactive then this operator does nothing. The integer resultant `count` retains its current value.

If the Boolean actuator `reset` is activated (and `enable` is active) then the integer resultant `count` is set to 0 (zero).

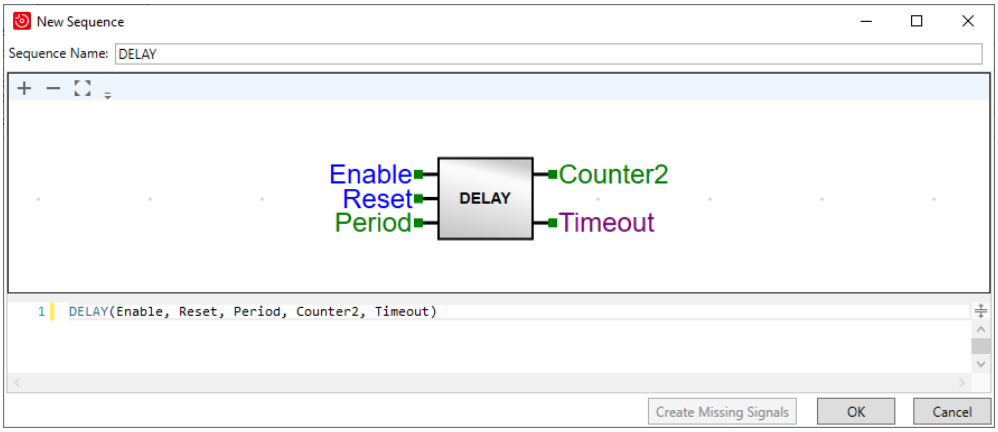
While `enable` is active, and `reset` is inactive, then the integer resultant count counts the number of executions loops since the last active-to-inactive transition of `reset`.

The integer resultant `count` is limited by the value of the integer actuator `period`. When `count` reaches `period`, then `count` will restart from 0 (zero) on the next

Continues on next page

4.6.4.1 Description of the complex operators
Continued

execution. When the resultant `count` is restarted from zero the `timeout` resultant is activated and then deactivated on the following execution.



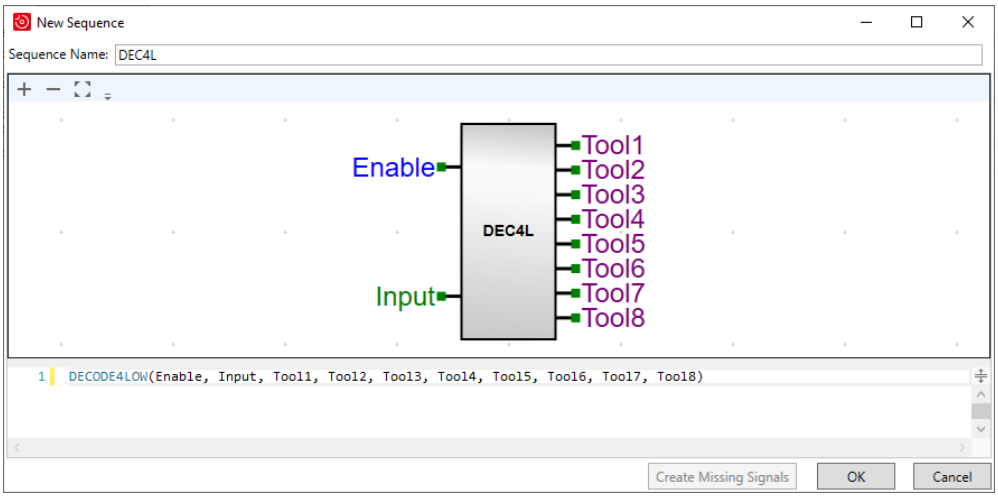
xx2300000729

Operator DECODE4LOW

Decodes the three least significant bits of an integer actuator into eight individual Boolean resultants if the fourth least significant bit of the integer actuator `input` is low. I.e. for integer values between 0 and 7, (0xxx).

Maximum one output will ever be activated at the same time.

DECODE4LOW works in conjunction with the DECODE4HIGH operator to create a 4 bits-to-16 decoder.



xx2300000730

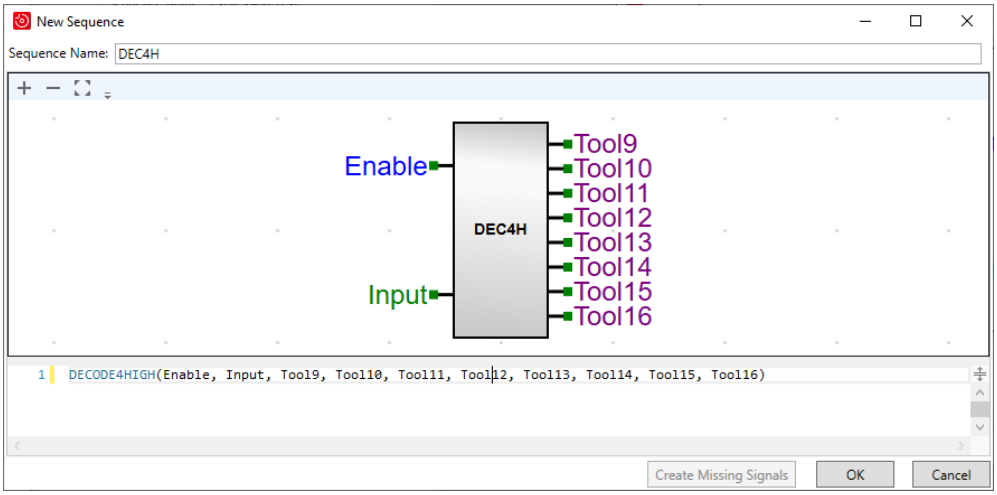
Operator DECODE4HIGH

Decodes the three least significant bits of an integer actuator into eight individual Boolean resultants, if the fourth least significant bit of the integer actuator `input` is high. I.e. for integer values between 8 and 15, (1xxx).

Maximum one output will ever be activated at the same time.

Continues on next page

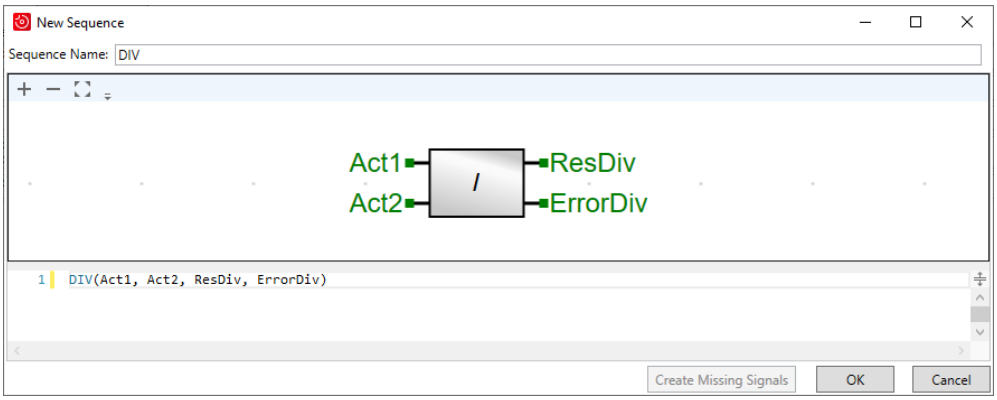
DECODE4HIGH works in conjunction with the DECODE4LOW operator to create a 4 bits-to-16 decoder.



xx2300000731

Operator DIV

This operator divides two integer actuators and produces the quotient as an integer resultant. If the denominator `act2` is zero, then `error` is set and `res1` keeps its current value.



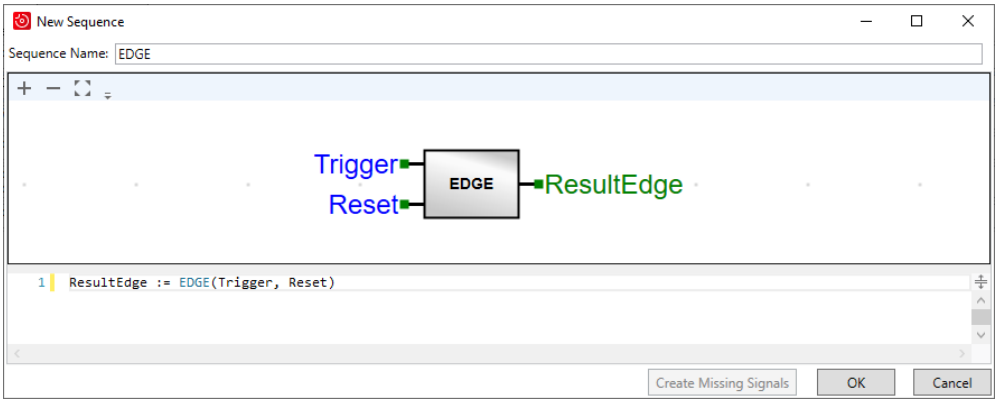
xx2300000736

Continues on next page

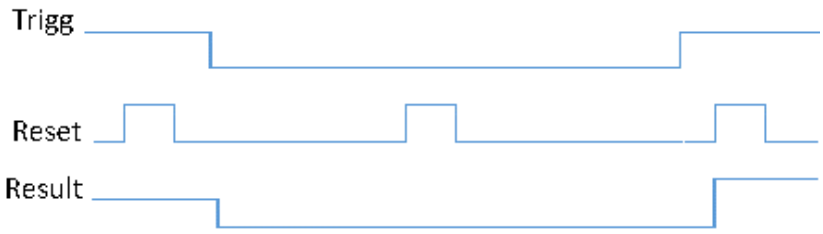
4.6.4.1 Description of the complex operators
Continued

Operator EDGE

The EDGE operator has two Boolean actuators, `trigg` and `reset`, and one Boolean resultant `result`. When `trigg` is inactive then `result` is also inactive. When `trigg` is active then the EDGE operator sets `result` to active upon a transition, from inactive to active, on `reset`.



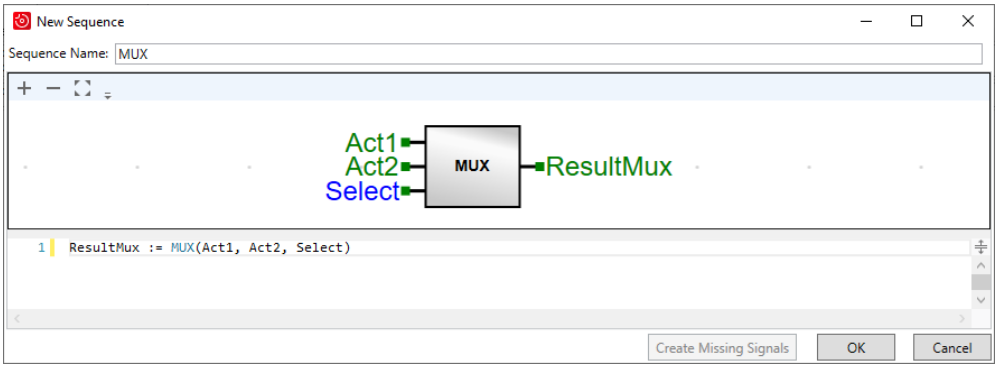
xx2300000732



xx1700000469

Operator MUX

The MUX operator has two integer actuators, `input1` and `input2`, one Boolean actuator `select`, and one integer resultant `result`. When `select` is inactive then `result` obtains the value of `input1`. When `select` is active then `result` obtains the value of `input2`.

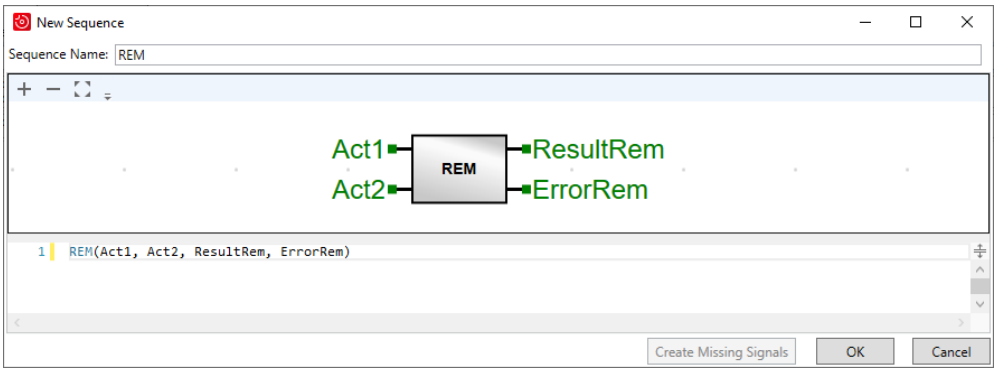


xx2300000733

Continues on next page

Operator REM

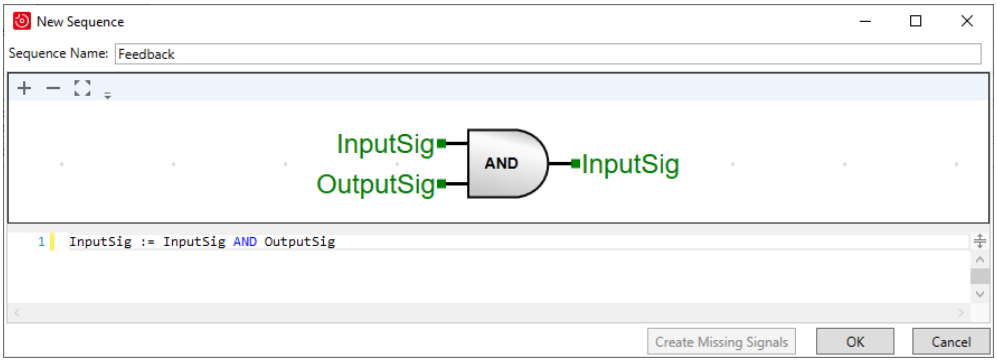
This operator divides two integer actuators and produces the remainder, modulo, as an integer resultant. If the denominator `act2` is zero, then `error` is set and `res1` keeps its current value.



xx2300000734

Feedback coupling

The feedback coupling feature refers to using the resultant integer output as an input to the same operator.



xx2300000735

4.6.5 Safe I/O system rules and limitations

General rules and limitations

- Signal names must be unique.
- Signals must have a name.
- Offsets do not have to have a corresponding signal, for example it is OK to only define offset 1, 7, and 63 as signals.
- Each signal is assigned a default value which is given at startup.
- Each signal, and equivalent alias, can have at most one writer.
- Each signal can have any number of readers.
- Expressions are executed in order by name, indicated by the leading number in the list.

See also *Application manual - I/O Engineering*,*Application manual - PROFINET Controller/Device*, *Application manual - EtherCAT*, *Application manual - Scalable I/O*.

Rules and limitations of combinatory logic

- A maximum of 256 combinatory logic operations in total are allowed.

  (256 operations in pre-logic and 256 operations in post-logic.)
- In some cases the configurator creates operations which are not shown to the user, for example in case of more than 8 operands to an AND operator. The configuration will in these cases internally split the operation into two (or more if required), resulting in more operations than visible in the UI.

4.7 Simulating SafeMove

About the Auto Generate From Simulation group

Automatic generation of safety parameters from simulation is a quick and easy way of creating zones and axes ranges based on a simulated path in RobotStudio. The function creates a zone or axes range that encapsulate the recorded simulated path.

A prerequisite is that there has to be at least one SafeMove tool in the configuration, and a simulation has to be defined.

> **Note**
>
> The simulation functions are only available when running a RobotStudio station.

The Record Simulation button

A simulation is setup, started, and stopped using the standard RobotStudio commands. For more information, see *Operating manual - RobotStudio*.

The simulation has to be recorded to be able to automatically generate safe functions from the simulation. Before the simulation is started, the **Record simulation** button in the Visual SafeMove tab must be pressed.

When there is a recorded simulation available, the **Safe Zone**, **Safe Range**, and **Tool Orientation Supervision** buttons become active.

The Safe Zone button

From the simulation it is possible to create a **Tool Position Supervision** and/or a **Tool Speed Supervision**.

The function creates a safe zone that encapsulates the path of the tool, or the tool and elbow, from the recorded simulation.

Function	Description
Tool only	Create a new safe zone that encapsulates the path of the tool from the recorded simulation.
Tool and elbow	Create a new safe zone that encapsulates the path of the tool and elbow from the recorded simulation.

The Safe Range button

From the simulation it is possible to create an **Axis Position Supervision** and/or an **Axis Speed Supervision**.

The function creates a safe range with the max and min values of each joint set to the max and min value from the recorded simulation.

The Tool Orientation Supervision button

From the simulation it is possible to create a **Tool Orientation Supervision**.

The function creates a **Tool Orientation Supervision** where the tolerances around the x and z axis of the tool are calculated from the simulation.

4.8 Protected basic configuration

Introduction

When configuring a large number of robot systems in a factory or a large production line, it is often desired to make the signal configurations similar or identical.

If every system has different signal names and different setups, then it will be very difficult for the operator to understand the systems.

Therefore it is possible to create a basic configuration that can be used as a template when configuring a new system. It is also possible to protect elements in the configuration.

> **Note**
>
> If you have protected groups configured outside RobotStudio (for example, for tools or equipment other than the robot and I/O), then when configuring the system in the **Modify RobotWare** function, select the checkbox **Enable Add-In Installation of Safety Parameters**. This way, these protected groups are not deleted on system reset.

The protected elements checksum

When RobotStudio saves the configuration, a separate checksum is calculated for the protected groups. The protected group's checksum is placed in the checksum protected part of the configuration file. If using protected groups, there is one checksum per group.

The customer can verify that the protected basic configuration is unchanged by reading the protected elements checksum.

When the configuration is loaded into the safety controller, the protected elements' checksums are extracted and made available for the safety PLC using a RAPID variable in the same way as the checksum of the complete safety configuration, see *RAPID components on page 195*.

By reading the checksum from RAPID, the safety PLC can compare the checksum with the expected checksum given from RobotStudio when the configuration was created.

> **Note**
>
> If loading a safety configuration from RobotWare 7.8 or earlier to a later RobotWare, then the protected elements' checksum is made into a protected group.

Creating a protected basic configuration

Use this procedure to create a protected basic configuration. The procedure can be repeated for each protected group.

	Action	Note/illustration
1	Create a basic configuration using Visual SafeMove in RobotStudio.	

Continues on next page

	Action	Note/illustration
2	When the basic configuration is completed, open the **Protected Elements** browser from the **Tools** group.	It is possible to protect groups of elements, known as protected groups.
3	Create a new group and select it. Select networks, signals, function mappings, stop configurations, and/or drive modules and assign them to the selected group.	The checksum for each protected group is shown. Click **Show** to only see the elements that are included in the selected protected group. There can be maximum 10 protected groups. A group cannot be empty.
4	Save the controller configuration from the **File** button in the **Visual SafeMove** ribbon.	

Tip

Click **Save** to save the protected group as an xml file. This can be imported on another robot.

Click **Report** to display the safety configuration report for the protected group.

Importing a protected group configuration

It is only possible to import a protected element group if the configuration version is the same as the currently used version. To see the current version, click the root node in the tree browser in **Visual SafeMove** and then **Safety configuration version**. To see the version in the file to import, open the file in a text editor and read the `version` attribute at the top.

If the file to import is older than the current configuration, it must first be upgraded. Because a group is not a full configuration it cannot be upgraded by itself, but must instead be upgraded as part of a full configuration. To make an upgrade from version A (the current group version) to version B (the current system version), use the following procedure:

1. Setup a system in version A with similar settings as the current system, with version B.
2. Save configuration (full configuration A).
3. Load the current system (with version B) and open the full configuration A.
4. Import the protected group (version A).
5. Upgrade the controller configuration to the latest version.
6. Save the group, which now has version B.
7. Restore the configuration in the current system (read from controller) to version B.
8. Import the group.

Continues on next page

Note

Upgrading configuration may change the global checksum, which may also include the checksum in the group. This depends on what the group contains and what has changed between versions A and B.

Using a protected basic configuration

After creating a basic configuration that should be used as a template in all robots, the configuration should be distributed to the persons creating the robot specific configurations.

Note

A user can change protected elements if he deselects them in the **Protected Elements** browser. However, this will affect the protected elements checksum. Therefore it is necessary to verify the protected elements checksum for each system.

Tip

A safety configuration can suitably be divided so that logical parts of the configuration are grouped, known as protected groups. For example, zones is one protected group, supervisions is one protected group, etc.

Example with safety PLC

Use this procedure to import and use a protected basic configuration with a safety PLC.

	Action	Note/illustration
1	Open, or import, the protected basic configurations into Visual SafeMove in RobotStudio.	Multiple files can be imported at the same time.
2	Add the system specific parts of the configuration.	The write-protected elements are visualized with a padlock icon in the **Visual SafeMove** browser.
3	Save, download, and verify the safety configuration.	
4	Make sure that the safety PLC is programmed with the checksums of all of the installed protected groups. The checksums should be the same in all systems where the protected groups are installed.	
5	By reading the checksum from RAPID, the safety PLC can compare the checksum with the expected checksum given from RobotStudio when the basic configuration was created.	

5 SafeMove in the FlexPendant user interface

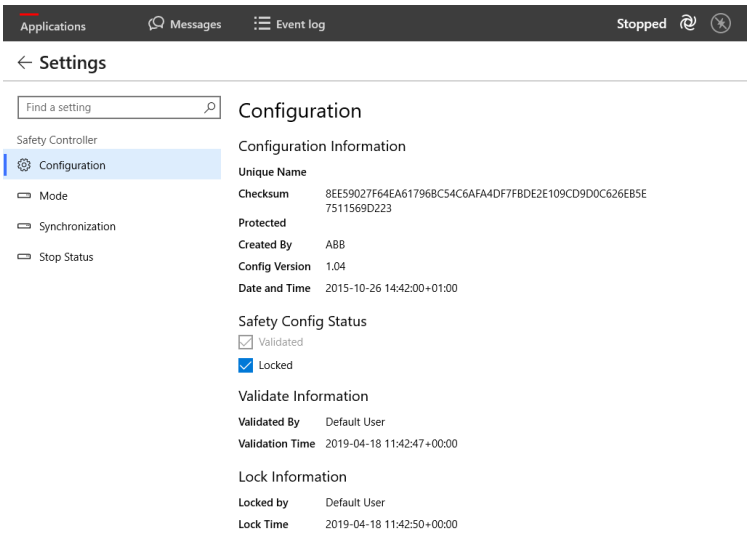
5.1 The Safety Controller control panel

Introduction

This section gives an overview of the **Safety Controller** control panel on the FlexPendant. Views, buttons, and other parts of the user interface are described in respect to their content and how they are accessed.

For more information on how to use the FlexPendant in general, see *Operating manual - OmniCore*.

The safety configuration can be viewed from the **Settings** app.



xx1900000976

User authorization (UAS)

The following UAS settings related to the safety controller are available for setting the user access on the FlexPendant.

For more information about creating user grants, see *Set up safety user grants on page 130*.

User grant	Description
Commissioning mode	Required for changing to commissioning mode, see *The Mode tab on page 110*.
Lock Safety Controller Configuration	Required for locking the safety configuration.
Safety Services	Required for the following actions: • Loading a safety configuration. • Setting the safety configuration status to **Validated**. • Changing the operating mode of the safety controller from the **Mode** tab.
Software synchronization	Required for synchronizing the controller from the **Synchronization** tab.

Continues on next page

The Configuration tab

The **Configuration** tab shows the status of the safety configuration file. From the configuration tab it is possible to view the safety configuration, validate, and lock the configuration file.

For more information, see *Validate the configuration on page 168*.

Safety Config Status	Description
Validated	When the safety technician has validated the configuration, the status of the configuration is changed to **Validated** on the FlexPendant.
Locked	When the safety responsible has approved the configuration, the status of the configuration is changed to **Locked** on the FlexPendant.

Tip

Tap the checksum to view the complete string in a larger window.

View safety controller data

Click the **View** button to view the safety configuration in a separate window.

The view function shows the safety configuration as it is defined in the xml-format, which means that structure and function names etc. does not have a one-to-one mapping with the configurator or the safety report.

For a more user friendly view, use RobotStudio to view configuration, or create a new report.

The Mode tab

The **Mode** tab is used to change the operating mode of the safety controller.

Mode	Description
Safety Supervision Mode	The safety supervision mode is the default mode where the safety supervision is active.
Commissioning Mode	Commissioning mode is intended to be used before the safety PLC is connected and functional. In commissioning mode, the selected signals are set high for not triggering stops. Commissioning mode is allowed in automatic operating mode, but the power to the actuators is disabled and the robot is prohibited from movement. The status bar on the FlexPendant displays that the safety controller is in commissioning mode. **DANGER** Using the function commissioning mode compromises the safety.

Continues on next page

Mode	Description
Service Mode	Service mode is intended to be used during service and commissioning. In service mode all safety supervision is deactivated, so it is possible to jog and run the robot without limitations. The violation output signals are set high, indicating no violation. The communication to the safety PLC is active so the configured safety stops are active. Service mode is only allowed in manual mode. If the operating modes manual full speed or automatic mode is selected, then service mode is deactivated and safety supervision mode is activated by the safety controller. The status bar on the FlexPendant displays that the safety controller is in service mode.

The Synchronization tab

The safety controller has to be synchronized with the robot controller before the safety supervision can be used. For more information, see *Configure the synchronization position on page 137* and *Synchronization guidelines on page 197*.

The **Status** LED has the following indication:

- Green - Synchronized
- Red - Unsynchronized
- Grey - Undefined, no information is present.

Button	Description
Press and hold to move	If **Move to Sync Position** is enabled, press and hold this button to move to the synchronization position without having to jog the robot manually.
Synchronize	Synchronizes the safety controller with the robot controller.

The Stop Status tab

The **Stop Status** tab is only present when a *Stop configuration* is configured in the safety configuration. Up to 8 stop configurations are shown with a green or red LED indicating the status.

For more information, see *The Stop Configuration button on page 78*.

The Cyclic Brake Check tab

The **Cyclic Brake Check** tab is only present when *Cyclic Brake Check* is configured in the safety configuration.

The tab displays the current status, the interval for the next brake check, and the status of the previous brake check. For more information see *The Cyclic Brake Check button on page 78* and *Cyclic Brake Check guidelines on page 200*.

The **Status** LED has the following indication:

- Green - OK
- Yellow - Pre warning
- Red - CBC required
- Grey - Undefined, no information is present.

The **Last Brake Check** LED has the following indication:

- Green - OK

Continues on next page

- Yellow - Warning
- Red - Error
- Grey - Undefined, no information is present.

Button	Description
Execute	Starts the service routine and performs a Cyclic Brake Check. **Note** All drive modules with a CBC configuration are targeted when executing the service routine.

5.2 The SafeMove configurator app on FlexPendant

Introduction

The application **SafeMove** on the FlexPendant offers an intuitive way to visualize and configure a safety configuration for systems with the option *SafeMove Collaborative*. This includes stop functions and *Cyclic Brake Check*. To get started, see *Use cases on page 117*.

Tip

Use the online user guide tool, included in the SafeMove configurator app, for help with the SafeMove configuration setup process.

Note

The SafeMove configurator app is available for the following robots:

- CRB 1100
- CRB 1300
- CRB 15000

The configuration follows the same principles as when using Visual SafeMove in RobotStudio but the functionality is not as extensive. For full configuration of SafeMove, see *Configuring SafeMove on page 125*.

For more information about transient contact, quasi-static contact, and body areas, see *Guidelines for transient and quasi-static contact, CRB 15000 on page 207*.

Overview of the user interface

The user interface consists of a configurator and a 3D model that visualizes the robot with the configured encapsulations and zones. The first time that the app is opened, a default factory setting is loaded. If a safety configuration is loaded, this will be shown.

- The tab **Robot Encapsulation** contains the configuration of the encapsulations of the robot itself.
- The tab **Tool Encapsulation** contains the configuration of the encapsulations of the tools.
- The tab **Tool Data** contains the configuration for the tools.
- The tab **Safe Zones** contains the configuration of the safe zones.
- The tab **Global Settings** contains the configuration for Cyclic Brake Check and supervision settings.
- The tab **Synchronization** contains functions for software synchronization.
- The **Context** menu (...) contains functionality for loading, saving, and viewing configurations, and to reset the configuration.

The functionality is described in detail in section *Safety functions provided by SafeMove on page 33*.

Continues on next page

Prerequisites

- The option *SafeMove Collaborative* is required.
- To edit a configuration, the grant *Safety Services* is required. A user without this grant can view a configuration, but not modify, write it to the controller, or apply it to the controller. See *Set up safety user grants on page 130*.

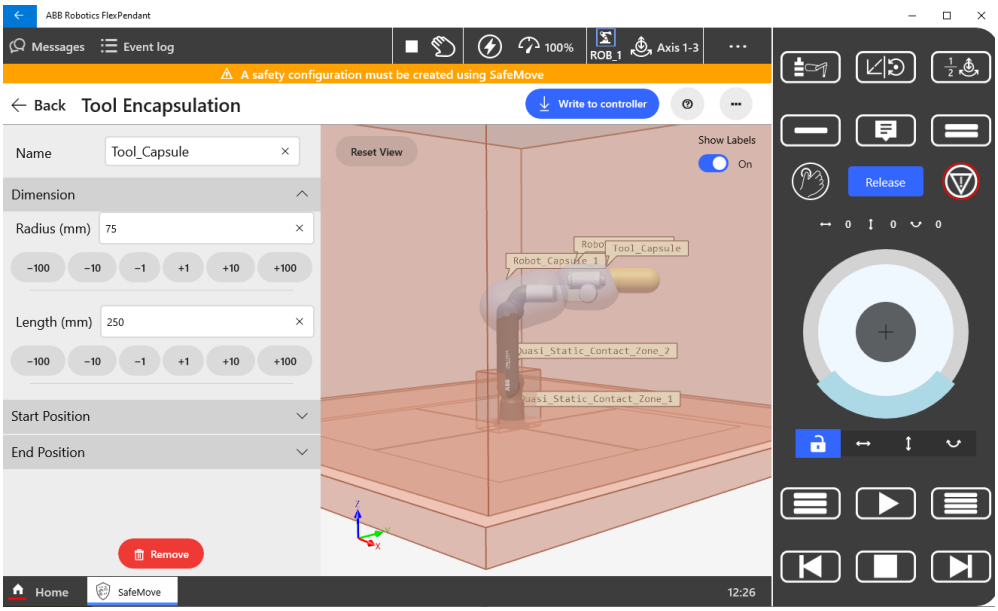
Template configurations

The template configuration is adapted for the specific manipulator, and typically contains one or two encapsulations of the arm, one encapsulation of the wrist (intended for the tool), one or two safe zones, and a Cyclic Brake Check setting. This configuration is typically a good start for a generic application with a smaller tool.

The factory setting is an empty safety configuration. A loaded configuration can be removed and the system is then reset to the factory setting.

Encapsulations

The encapsulations are geometries that can be in the shape of a sphere, capsule, or lozenge. A sphere or capsule encapsulation can be modified in dimension, length, and position. A lozenge capsule can also be modified in rotation.



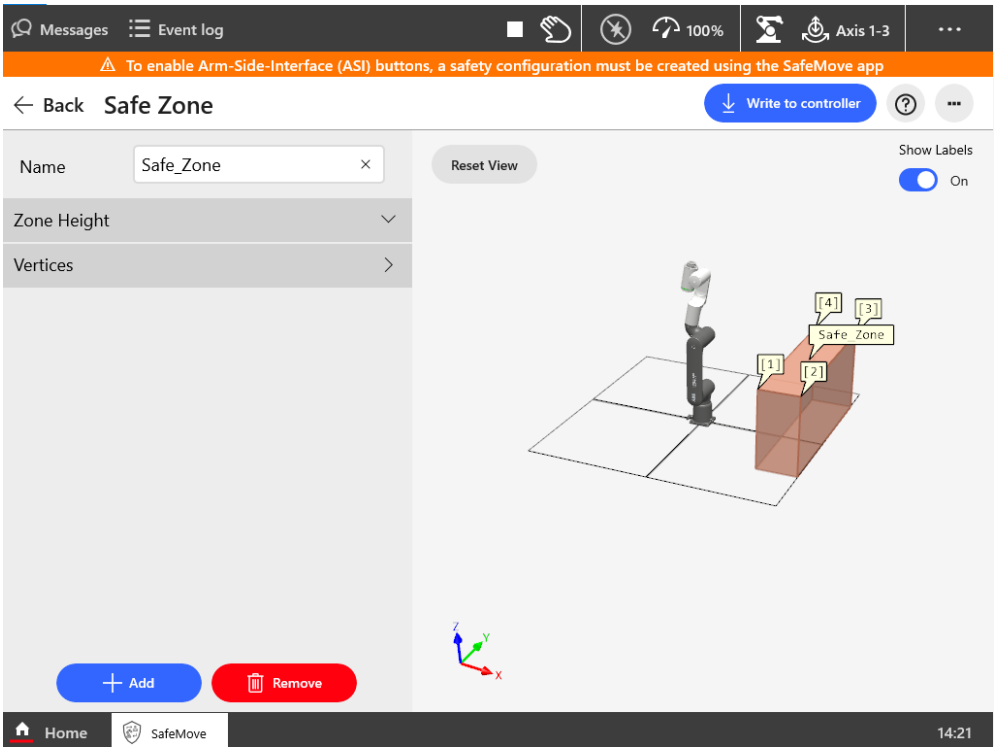
xx2100000712

Continues on next page

Safe zones

The default safe zone is a rectangular box with four vertices. The vertices defines the shape of the safe zone, and the position in space. More vertices can be added to define the safe zone. The minimum number of vertices is 4, and the maximum is 24.

Each vertex can be edited in x and y values.



xx2100000714

Each vertex is numbered, from 1 and up. When a new vertex is added between two existing vertices the vertex numbers will be automatically adjusted so that they come in order. For example, if a new vertex is added between vertices 2 and 3, the vertex with index 3 will change to 4 and the new vertex will be indexed 3.

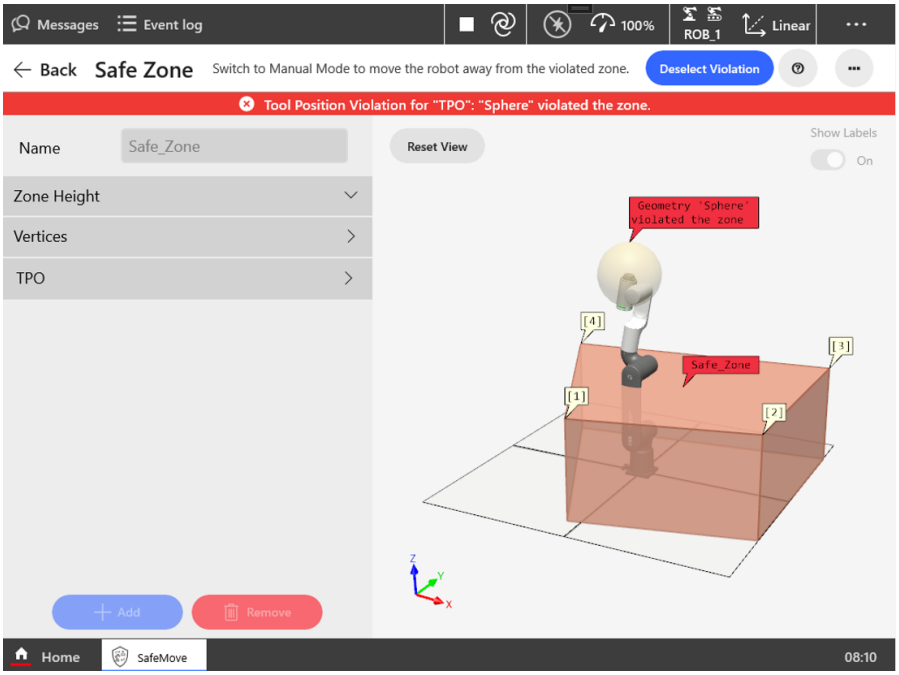
Display of safety violations

During the validation of a robot cell using the SafeMove app, it is possible to check whether the robot is committing a safety violation. For example, robot crossing a forbidden zone, robot speed or force exceeding a certain value, and so on. Once a violation is detected and displayed on the SafeMove app, it is possible to take the necessary actions.

When a safety violation has been triggered, the name of the violation zone and the type of violation that was triggered are displayed in the 3D view and at the top bar

Continues on next page

of the SafeMove app. Violation messages will disappear when the motors are on or when the controller is restarted.



xx2300001440

From the safety violation messages, you can read details about the violated supervision, which zone it is, suggested corrective action, and so on. Tap the **Deselect Violation** button to make the message about the current violation disappear until the SafeMove app is opened again. But to completely recover from the violation, perform the action suggested beside the **Deselect violation** button and then go to motors on.

Display of safety violations in zone is available for the following supervisions:

- Tool Position Supervision, see *Tool Position Supervision (TPO) on page 57*.
- Tool Speed Supervision, see *Tool Speed Supervision (TSP) on page 53*.
- Tool Force Supervision, see *Tool Force Supervision (TFO) on page 59*.
- Tool Orientation Supervision, see *Tool Orientation Supervision (TOR) on page 58*.

Supervision functions

The global supervision functions are not connected to a specific safe zone or safe range. They can be added, modified, and deactivated.

For more information about the global supervision functions, see *Supervision functions on page 50*.

Synchronization

The **Synchronization** tab is used to manually set the current joint positions for the robot.

For more information about synchronization, see *Software synchronization on page 41*.

Continues on next page

Recommended working procedure

Use this procedure when configuring SafeMove in the configurator app on FlexPendant.

1. Log in as a user with safety user grants.
2. Start the SafeMove configurator app.
3. Load a safety configuration template or an existing configuration from the **Context** menu (...).
4. Configure encapsulations.
5. Configure zones and the supervision functions.
6. Load the configuration to the safety controller.
7. Restart the controller.
8. In the **Settings** app, tap **Safety Controller** and validate the configuration.

   See *Validate the configuration on page 168*.
9. In the **Settings** app, tap **Safety Controller** and set the safety configuration to validated and then lock it.

   See *Setting the configuration to locked on page 181*.

For more details, see *Use cases on page 117*.

For functionality not supported in the SafeMove configurator app, use Visual SafeMove in RobotStudio.

Use cases

Start the SafeMove configurator app

The SafeMove configurator app is available on the home screen of the FlexPendant for systems with the option *SafeMove Collaborative*. If the app is not shown, then review the system settings using the **Modify Installation** function in RobotStudio and add that option.

The first time that the app is opened, a default factory setting is loaded. This contains only the manipulator with *Cyclic Brake Check* activated. There are no encapsulations, safe zones, or tool data defined.

The factory setting can always be resumed, if needed.

To continue and create a safety configuration, see *Load a safety configuration template on page 117*.

Load a safety configuration template

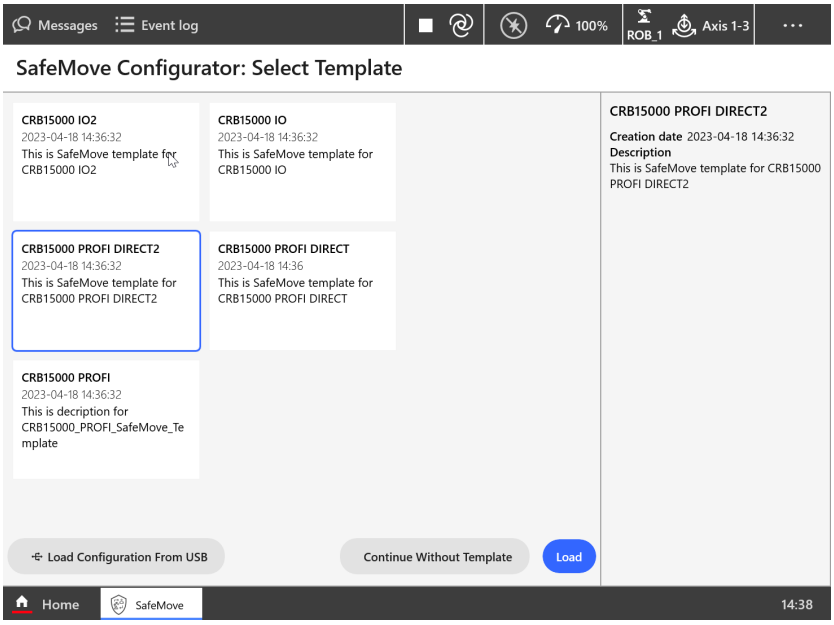
The safety configuration template feature is available from RW 7.12 onwards. Systems with RW 7.10 or earlier will still have the default template solution.

Use the following procedure to load a predefined safety configuration template and apply it to the robot controller.

1. Log in as a user with safety user grants.
2. Open the SafeMove app.
3. Tap **Enable Edit Mode**.

Continues on next page

The **SafeMove Configurator: Select Template** page is displayed with a list of available templates.



xx2300001391

4 Select a template from the list.

The metadata of the selected template is displayed on the right side panel.

5 Tap **Load**.

The **Load Safety Configuration** dialogue is displayed.

6 Tap **Yes**.

The selected safety configuration template is loaded on the FlexPendant.

7 Review that the selected template configuration is suitable for the intended application.

If modifications are needed, see *Modify a loaded safety configuration on page 119*.

> **Note**
>
> A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

8 If the template configuration is suitable, select **Write to controller**.

The safety report is presented on the screen.

9 Save the safety report. Take a print out and sign this safety report.

See *Validate the configuration on page 168*.

10 Tap **Apply to controller**.

The **Saved** dialogue is displayed

11 Tap **Restart Controller**.

Continues on next page

The controller is restarted and loads the newly saved safety configuration template.

> **Note**
>
> To change the loaded safety configuration template, tap the **Context** menu, select **Open Template Selector**, select the required template from the list, and follow the rest of the steps.

Modify a loaded safety configuration

Use the following procedure to modify a loaded safety configuration and apply it to the robot controller.

1. Log in as a user with safety user grants.
2. Open the SafeMove app.

   The **SafeMove Configurator** page is displayed along with the saved safety configuration.
3. Select **Enable Edit Mode** to edit the loaded safety configuration.
4. To add or modify an encapsulation, tap **Add** and select a geometry for **Robot Encapsulation** or **Tool Encapsulation**.

   To modify the encapsulation, select it and modify the attributes.
5. To add or modify a zone, tap **Add** and **Add Zone**.

   Select the safe zone and modify the attributes. See *Modify a safe zone on page 120*.
6. To add or modify a global setting, tap **Add** and select which supervision to modify.

   For a detailed description of global settings, see *Global supervision functions on page 147*.
7. When the configuration is done, select **Write to controller**.

   The safety report is presented on the screen.

   > **Note**
   >
   > A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.
8. Save the safety report. Take a print out and sign this safety report.

   See *Validate the configuration on page 168*.
9. Tap **Apply to controller**.

   The **Saved** dialogue is displayed
10. Tap **Restart Controller**.

    The controller is restarted and loads the newly saved safety configuration.

Continues on next page

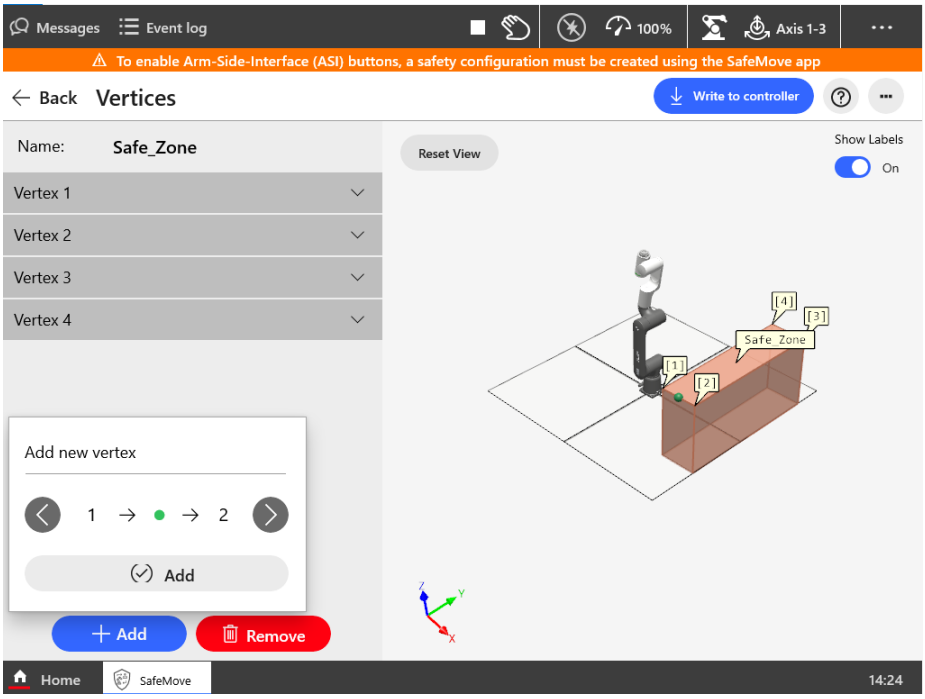
Modify a safe zone

Use the following procedure to modify a safe zone.

1. Add a new safe zone or select an existing safe zone.
2. Tap **Safe Zones** to open the attributes.
3. Add, modify, or remove vertices as needed to create the desired shape of the safe zone.

   The green dot in the 3D visualization shows where the new vertex is located. Use the arrows to change the position (index).

   Tap the grey **Add** button to place the vertex.

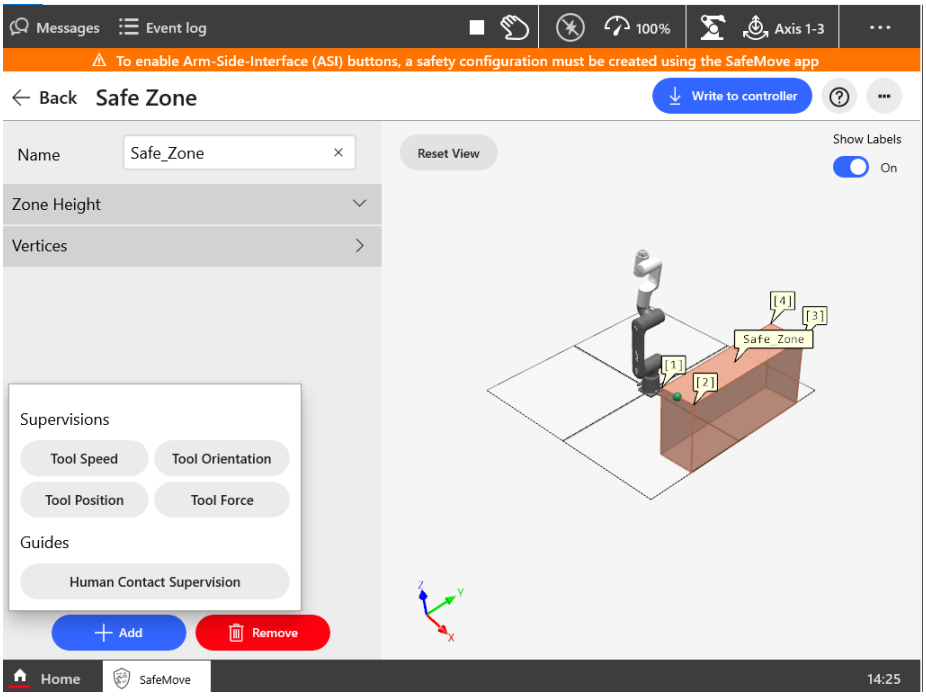


xx2100000710

4. To add a supervision to a safe zone, tap to select the safe zone in the 3D view, then tap **Add**.
5. Select a supervision function or guide.

Continues on next page

For a detailed description of global settings, see *Configuring SafeMove on page 125*.



xx2100000713

6 For supervision functions, select stop category, signal, and any other available setting applicable for the function.

   The functionality is described in detail in section *Safety functions provided by SafeMove on page 33*.

7 For the guide **Human Contact Supervision**, select contact type, tooling properties, and body contact areas.

   See *Use the Human Contact Supervision settings on page 121*.

Use the Human Contact Supervision settings

Use the following procedure for Human Contact Supervision.

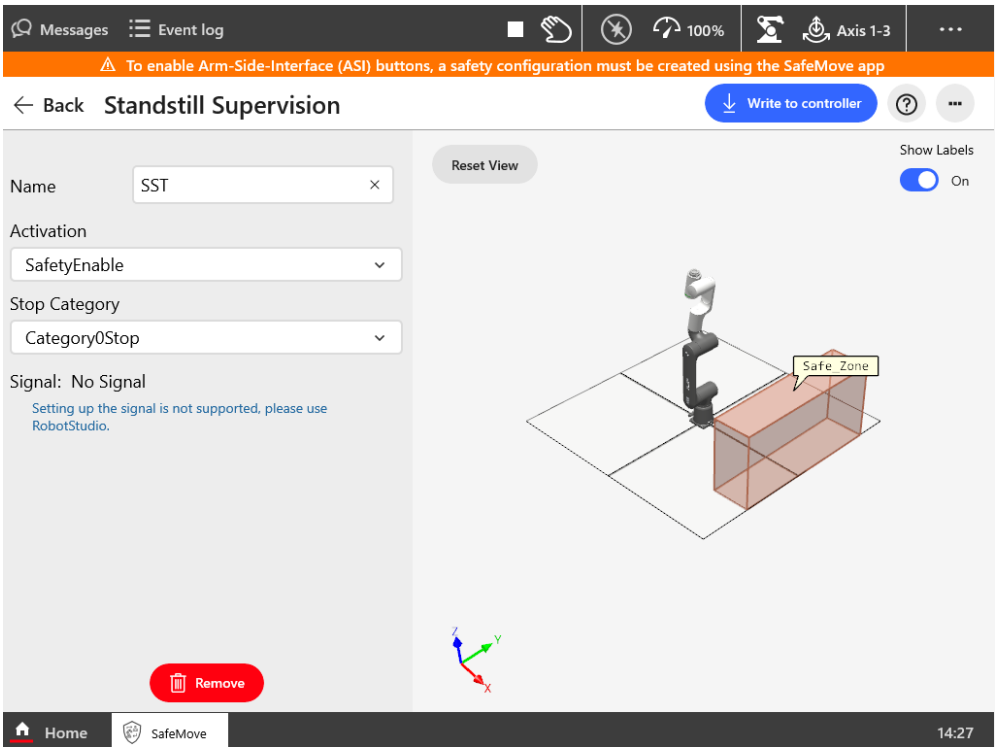
1 Select **Human Contact Supervision**.
2 Select contact type.
3 Define the tooling properties.
4 Select body contact areas. This is only used for transient contact.
5 Review the suggested supervisions.
6 When the supervision is applied, the data is transferred to *Tool Speed Supervision* and *Tool Force Supervision*.

For more details, see *Guidelines for transient and quasi-static contact, CRB 15000 on page 207*.

Continues on next page

Modify the Standstill Supervision settings

The Standstill Supervision functionality is not active by default. It can be added, modified, and deactivated.



xx2100000251

Modify the global supervision settings

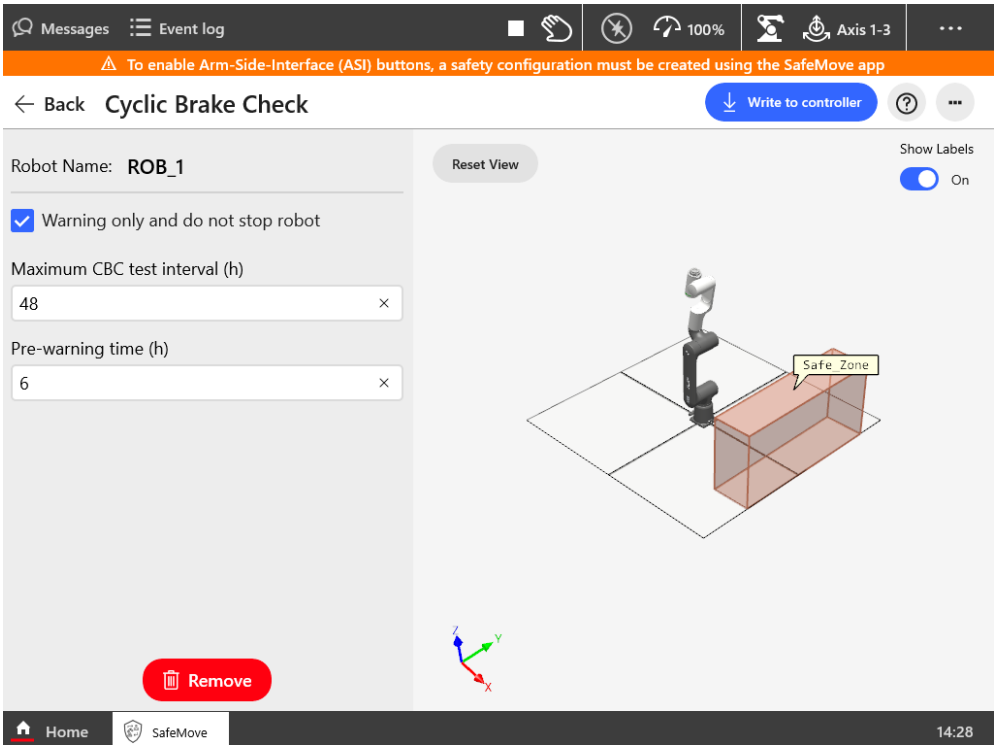
The global supervision functions are not connected to a specific safe zone or safe range. They can be added, modified, and deactivated.

For more information about the global supervision functions, see *Supervision functions on page 50*.

Continues on next page

Modify the Cyclic Brake Check settings

The Cyclic Brake Check functionality is active by default. It can be modified and deactivated.



xx2100000717

Viewing the configuration report

The configuration report is available both on the FlexPendant and on the controller. It can be viewed from the **Context** menu.

Loading and exporting a safety configuration

An existing safety configuration on the FlexPendant can be exported from the **Context** menu, **Save Configuration To File**. It is also possible to load a safety configuration from a file.

Validate and lock the configuration

How to validate and lock the configuration is done in a similar way as when using Visual SafeMove. See *Validate the configuration on page 168*.

This page is intentionally left blank

6 Configuring SafeMove

6.1 Recommended working procedure

General

This section describes the recommended working procedure when configuring SafeMove for the first time. The working procedure helps to understand the dependencies between the different steps. A good approach when creating a new configuration is to start with the basic functionality. When that works as expected, expand the application.

A prerequisite is that all steps in the hardware and software installation procedures must have been performed, see chapter *Installation on page 65*.

For more information on what can be done with the configuration tool, see chapter *The Visual SafeMove user interface in RobotStudio on page 69*.

Note

For systems with the option *SafeMove Collaborative*, much of the configuration can also be done in the SafeMove configurator application on FlexPendant, but the RobotStudio version is more capable. The descriptions in this section is based on RobotStudio, but can be used also on the FlexPendant for the functions that are available there. See *The SafeMove configurator app on FlexPendant on page 113*.

Note

Some functionality is only available for SafeMove Pro or SafeMove Collaborative, see *Functional safety options on page 18*.

Basic steps

Use this procedure when configuring SafeMove.

	Action	See
1	Make some initial preparations.	*Preparations on page 127*
2	Configure system parameters.	*Configure system parameters on page 128*
3	Set the input and output size and name of the internal device.	*Application manual - I/O Engineering*
4	Set up safety user grants.	*Set up safety user grants on page 130*
5	Configure robot properties.	*Configure the robot on page 134*
6	Configure the synchronization position.	*Configure the synchronization position on page 137*
7	Configure the SafeMove tool definitions.	*Configure the tools on page 138*
8	Configure safe I/O signals.	*Configure safe I/O on page 141* See also *Application manual - I/O Engineering*

Continues on next page

6.1 Recommended working procedure

Continued

	Action	See
9	Configure zones and/or ranges.	*Configure the zones on page 143* *Configure the ranges on page 145*
10	Configure the supervision functions.	*Configure the supervision functions on page 148*
11	Configure other functions.	*Configure other functions on page 164*
12	Load the configuration to the safety controller.	*Load the configuration to the safety controller on page 167*
13	Restart the robot controller.	
14	Validate the configuration. Use the checksum to verify that the right configuration is used.	*Validate the configuration on page 168*
15	Set the safety configuration to validated and lock it.	*Setting the configuration to validated on page 181* *Setting the configuration to locked on page 181*

6.2 Preparations

Preparations

Experience shows that when starting with a clean system it is good to first configure the robot system and make some initial preparations before configuring SafeMove.

- Create a robot system with the option **SafeMove Basic**, or **SafeMove Pro**, or **SafeMove Collaborative**.
- Define the coordinate systems that should be used in the robot system.
- Create tool data for all needed tools, and define the TCPs, tool loads, payloads, and arm loads.
- Create work object data for all needed fixtures and define them.

6.3 Configure system parameters

About the system parameters

The configuration of system parameters required for a robot system should be made before starting with the safety configuration.

In addition to the system parameters that need to be configured for a robot system without SafeMove, there are a few parameters in topic *Motion* that are specific for SafeMove. These are described in this section.

> **Note**
>
> It is important to restart the robot controller after changing system parameters, before configuring SafeMove.

Type Mechanical Unit

All mechanical units for additional axes that shall be supervised must have the parameters *Activate at Start Up* and *Deactivation Forbidden* set to On. (Supervised mechanical units must always be active.)

Type Arm

Before configuring SafeMove, ensure that each joint in a drive module configuration has a unique *Joint Id* number. By default, the logical axis number for each joint will be used as the *Joint Id*. If more than one joint in the same cabinet share the same logical axis number, the parameter *Joint Id* in type *Arm*, topic *Motion*, can be used to change the *Joint Id* to a non-default value. This is only possible for joints belonging to an additional axis, for example, an ABB positioner, not for robot joints. See *Technical reference manual - System parameters*, section *Type Arm* for a description of the parameter *Joint Id*.

If an axis should be excluded from Cyclic Brake Check, set the parameter *Deactivate Cyclic Brake Check for axis* to On.

The maximum working area for additional axes has to be limited according to limitations specified in section *Work area for additional axes on page 28*. This must be taken into consideration when entering the parameters *Upper Joint Bound* and *Lower Joint Bound*. (The parameter values in radians or meters on arm side.)

Type Brake

If Cyclic Brake Check is executed on an additional axis, a lowest safe brake torque must be defined. A 5% margin is added during the test for setting the fail limit. The parameter used is *Max Static Arm Torque* defined in Nm on motor side. A warning limit is set with a higher torque value (depending on the brake).

Type Motion Planner

If reducing the max speed that SafeMove allows in manual mode, the jogging speed of the robot has to be reduced to the same value by changing the parameter *Teach Mode Max Speed*.

Continues on next page

The parameter *Use checkpoint limitation in world* enables the robot to limit checkpoint speed in world coordinate system in teach mode. In this way the additional speed from, for example, a track motion is added to the checkpoint speed and the robot speed is reduced. This parameter is useful when combining SafeMove with a robot on track motion. SafeMove supervises the speed of check points in world coordinate system and if this parameter is not active there is a risk that SafeMove will trigger overspeed error when robot and track is moved simultaneously.

6.4 Set up safety user grants

Why do you need safety users

Configuring SafeMove is normally done initially and then never changed (until the robot is used for a different purpose). It is vital that the safety configuration is not changed by unauthorized personnel. It is therefore recommended to have specific safety users with grants permitting the respective assignments.

Grants specific for SafeMove

User grant	Description
Commissioning mode	Required for changing to commissioning mode, see *The Mode tab on page 110*.
Lock Safety Controller Configuration	Required for locking the safety configuration.
Safety Services	Required for the following actions: • Loading a safety configuration. • Setting the safety configuration status to **Validated**. • Changing the operating mode of the safety controller from the **Mode** tab.
Software synchronization	Required for performing software synchronization from the FlexPendant.

Note

Anyone is allowed to create or view a SafeMove configuration, but only a user with the grant **Safety Services** is allowed to download it to the controller.

Tip

Create different user groups as described in *Operating manual - RobotStudio*. Make sure that one administrator has the grant *Manage UAS settings* and that the regular users (operators, Default user, etc.) do not have any the following grants:

- **Commissioning mode**
- **Lock Safety Controller configuration**
- **Safety services**
- **Lockable mode selector**
- **Write access to controller**
- **Manage UAS settings**

WARNING

Users must be educated appropriately before given any grants related to safety configuration or operation.

Continues on next page

Additional grants that may be required when working with SafeMove

User grant	Description
Remote restart	Required when configuring from RobotStudio (remote restart of the controller is required after downloading a new configuration).
Program debug	Required when configuring from RobotStudio (allows request write access).
Read access to controller disks	Required when loading a configuration, already stored on controller disk, from the FlexPendant.
Write access to controller	Required for writing a safety configuration to the controller.
Read access to controller	Required for reading an existing safety configuration from the controller.

Example of recommended users

User	Service technician (line builder)	Safety responsible	Safety operator	Supervisor
Safety Services	X	X		
Lock Safety Controller Configuration		X		
Software synchronization			X	
				X
Commissioning mode	X			
Remote restart	X	X		
Program debug	X	X		
Read access to controller disks	X			

6.5 Starting Visual SafeMove

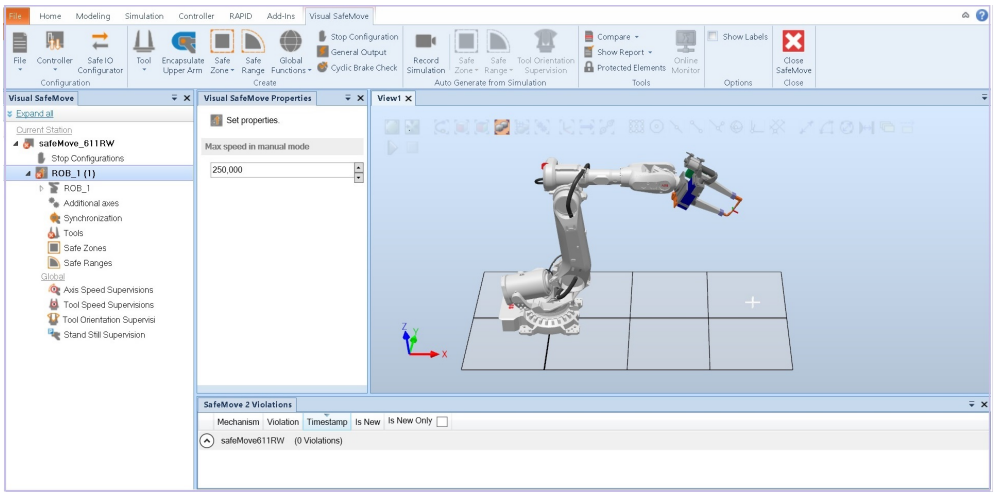
Starting Visual SafeMove in RobotStudio

	Action
1	Start RobotStudio with a virtual controller (with or without a station) or connect a real controller.
2	In the **Controller** tab, click **Online Monitor**. (Not needed when running a RobotStudio station.)
3	In the **Controller** tab, click **Safety**, then select **Visual SafeMove**.

6.6 Configure Manual Operation Supervision

Setting the max speed in manual mode

	Action	Note/illustration
1	Click on **ROB_1 (1)** in the SafeMove.	
2	In the SafeMove Properties browser, set **Max speed in manual mode**.	The max speed cannot be higher than the default value of 250 mm/s, but a lower value can be set. See also system parameters *Teach Mode Max Speed* and *Use checkpoint limitation in world* in section *Configure system parameters on page 128*.



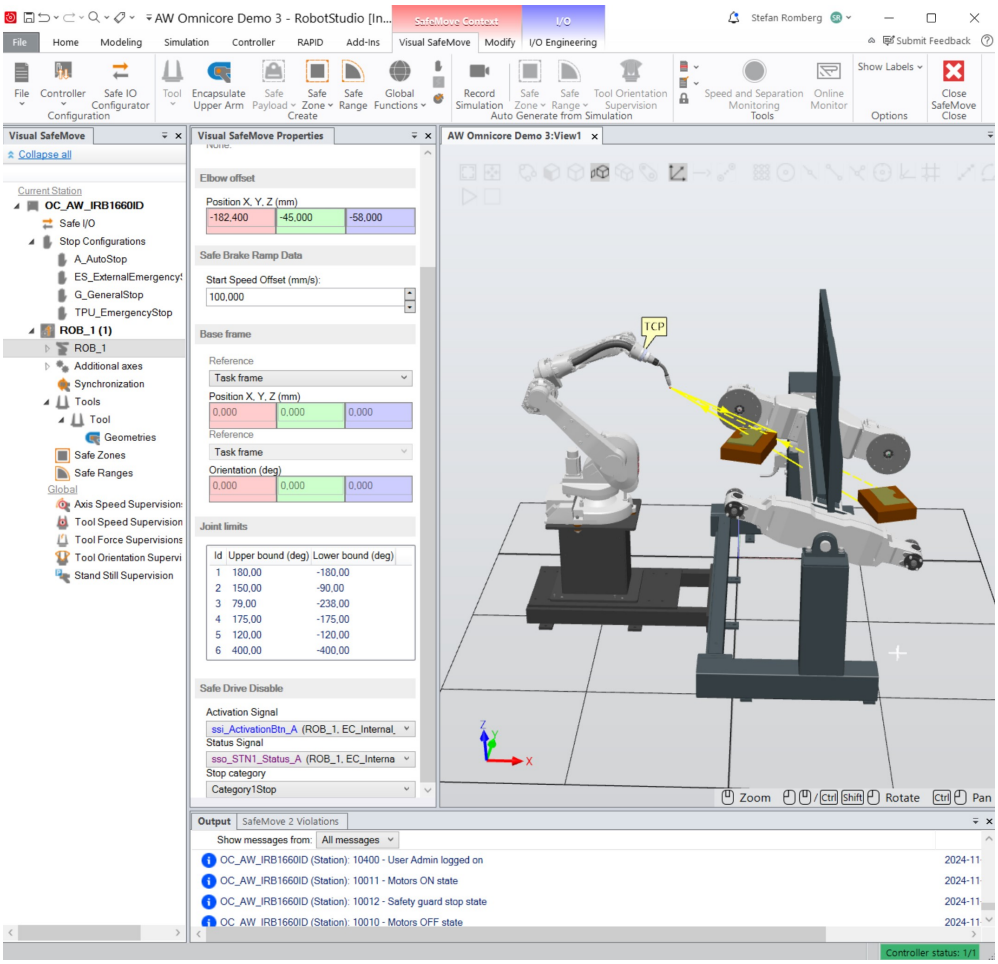
xx2000001832

6.7 Configure the robot

Setting the robot properties

	Action	Note/illustration
1	Click on **ROB_1** in the SafeMove browser to specify the robot properties in the SafeMove Properties browser.	
2	Under **Elbow offset**, set the **X**, **Y** and **Z** values for the elbow point.	See *Explanation of Elbow offset on page 84*.
3	Under **Safe Brake Ramp Data**, set the **Start Speed Offset**.	See *Explanation of Safe Brake Ramp on page 88*.
4	Under **Base frame**, select reference co-ordinate system.	See *Explanation of Base Frame on page 85*.
5	If Safe Disable of Drive Unit should be used for the robot, set the parameters **Activation Signal**, **Status Signal**, and **Stop Category** under **Safe Drive Disable**.	See *Safe Disable of Drive Unit parameters on page 88*.

The joint limits are shown here but cannot be changed.



xx2400001904

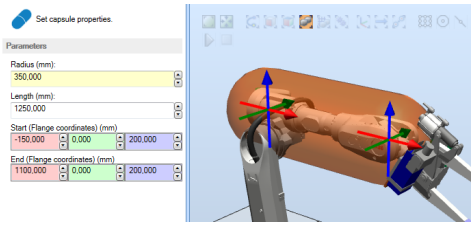
Continues on next page

Encapsulate upper arm

Automatically create encapsulation

In the SafeMove ribbon, click on **Encapsulate Upper Arm**. This will create a geometry surrounding the upper arm.

Manually create encapsulation

	Action	Note/illustration
1	In the **Visual SafeMove** browser, select the robot (for example **ROB_1**).	
2	In the **Visual SafeMove** ribbon, click on the **Capsule**.	
3	Click on the shape you have just created to select it.	
4	Fill in the parameters for the shape to specify the position and size of the shape.	 xx1600000129

Configuring robot stopping functions

Note

Depending on the controller variant and RobotWare version, the configuration options are different. See the product manual for the controller.

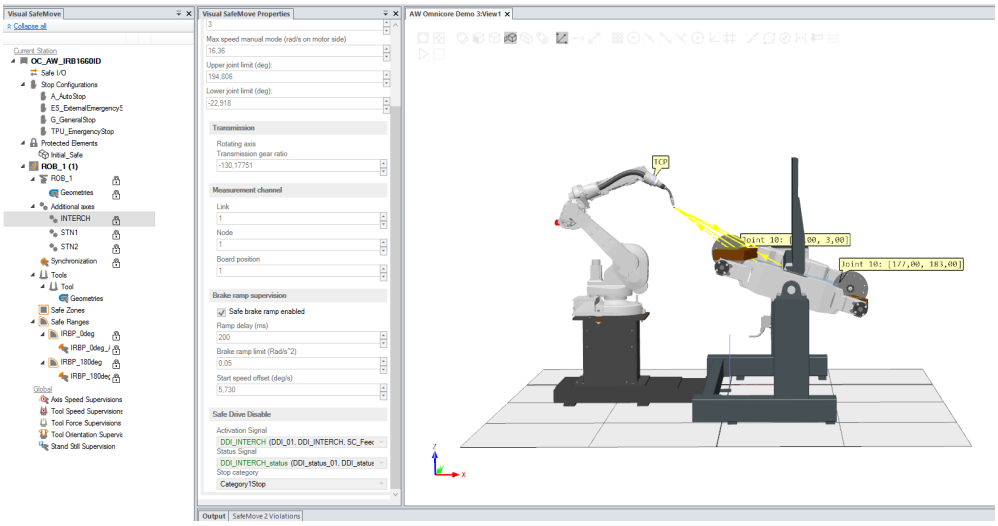
	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click **Stop Configuration**.	
2	Select a stop configuration or right-click to create a new configuration.	Not all configurations can be modified.
3	For user-created stop configurations, select the signal that should trigger the stop in the **Trigger signal** dropdown menu.	Trigger signal definitions: 0 = activate stop 1 = deactivate stop
4	For user-created stop configurations, if a status signal should be set when the functionality is active, select the signal to use in **Stop trigger status** dropdown menu. If no output signal should be used, select **No signal**.	**Stop trigger status** is a status signal telling if the configured stop has triggered. Signal definitions: 0 = stop triggered 1 = stop not triggered
5	Define the mode (automatic or manual) in the list **Mode**.	The available modes depends on the controller variant and RobotWare version. See the product manual for the controller for more details.
6	In **Stop category**, select if the function should stop the robot with stop category 0 or stop category 1.	Not all configurations can be modified.

6.8 Configure additional axes

Setting the additional axis properties

	Action	Note/illustration
1	If the axis should be part of the SafeMove supervision, in the **Visual SafeMove** browser right-click the axis and then select **Include in configuration**.	
2	Specify **Servo lag**.	**Servo lag** is the estimated lag (in radians on motor side) for the additional axis. For more information, see *Servo Delay Factor and Servo Lag on page 210*.
3	Specify **Servo delay factor**.	Estimated delay factor between reference position and measured position (number of 4 ms units) when moving the additional axis. (See TuneMaster, signal number 17 and 18.) For more information, see *Servo Delay Factor and Servo Lag on page 210*.
4	If safe brake ramp should be used for the axis, select the checkbox **Safe brake ramp enabled**. Set the values for **Ramp delay**, **Brake ramp limit** and **Start speed offset**.	See *Explanation of Safe Brake Ramp on page 88*.
5	If Safe Disable of Drive Unit should be used for the axis, set the parameters **Activation Signal**, **Status Signal**, and **Stop Category** under **Safe Drive Disable**.	
6	If the additional axis is used as an interchange positioner, then the safe range for the 0° and 180° position of the interchange should be configured.	See *Configure the ranges on page 145*.

Some information about the additional axis is shown, but cannot be changed. This includes joint limits, transmission gear ratio and measurement channel information.



xx2400001847

6.9 Configure the synchronization position

Setting the synchronization position

	Action	Note/illustration
1	Click on **Synchronization** in the **Visual SafeMove** browser.	
2	Select **Activation**. To use software synchronization only, select **Software synchronization**. To also use hardware synchronization, select the input signal from the sync switch.	Software synchronization is always available even if hardware synchronization is configured.
3	If a status signal should be set when the safety controller is synchronized, select the signal to use in **Synchronization status**. If no output signal should be used, select **No signal**.	Signal definitions: 0 = not synchronized 1 = synchronized
4	Jog the robot to the synchronization position. In the **Visual SafeMove Properties** browser, click on **Read current values**.	It is also possible to specify the axis position values manually.

6.10 Configure the tools

About tool configuration

A SafeMove tool can have one or both of the following:

- Up to 4 shapes surrounding the tool. This is used by the function Tool Position Supervision, which supervises that no part of the shape is outside its allowed zone.
- Up to 8 speed supervision points. The function Tool Speed Supervision always supervises that the TCP and robot elbow do not move faster than allowed speed. By adding speed supervision points, Tool Speed Supervision will supervise these points as well. If speed supervision points surrounds the tool, no part of the tool can move faster than allowed max speed.

Up to 16 SafeMove tools per drive module can be defined, in case a tool changer is used. Input signals are used to specify which tool is used. In order to configure more than one tool in SafeMove, an activation signal for each tool must be specified (if the first tool is set to be permanently active, no new tool can be created).

Orientation of tool shapes

The user can set the orientation of the shapes surrounding the tool using Euler ZYX angles. This rotation can also be found in the safety report.

The same rotation can be described as Euler ZYX angles in multiple ways. For example Euler ZYX angles (90, 90, 0) is the same rotation as (0, 90, -90). The orientation control updates the Euler angles immediately when the user is done editing the orientation. This may cause the Euler angles to change to another representation of the same rotation.

The underlying mathematical representation of the rotation in the controller is in the form of a quaternion which has a unique representation.

Creating a SafeMove tool

Automatically create a SafeMove tool

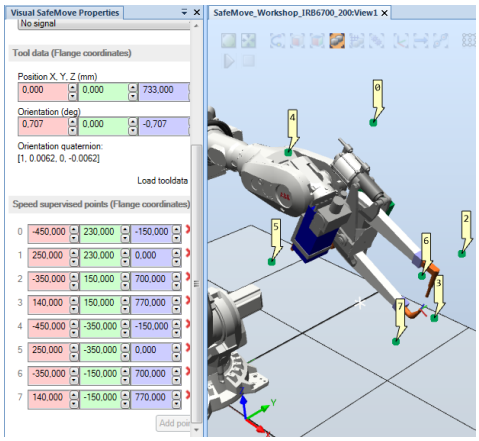
In the SafeMove ribbon, click on **Tool**, select **Encapsulate** and then select the tool you want to represent with a SafeMove tool. This will create 8 speed supervision points around the tool and also create a geometry surrounding the tool.

> **Note**
>
> The tool must be correctly defined in RobotStudio for the automatic encapsulation to work.

Manually create a SafeMove tool

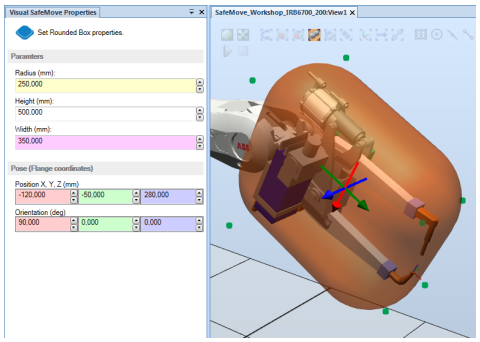
	Action	Note/illustration
1	In the SafeMove ribbon, click on the **Tool** menu and select **New**.	

Continues on next page

	Action	Note/illustration
2	Select **Activation**. If no tool changer is used, select **Permanently active**. If a tool changer is used, select which input signal should be used to activate this SafeMove tool.	Tool signal definitions: 0 = deactivate tool 1 = activate tool **Note** One, and only one, tool must be active at all times.
3	If a status signal should be set when the tool is active, select the output signal to use in **Function active status**. If no output signal should be used, select **No signal**.	
4	Under **Tool data**, fill in the position and orientation of the tool's TCP in relation to tool0 (the mounting flange).	**CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
5	Specify **Speed supervision points**. Click on **Add point** until you have the number of speed supervision points you want (max. 8). Fill in the X, Y, and Z values for each speed supervised point. A speed supervised point is specified by its X, Y and Z coordinates in the tool0 coordinate systems (mm from the mounting flange).	 xx1500000237
6	In the ribbon, click on a geometry (**Sphere**, **Capsule**, or **Rounded box**).	
7	Click on the shape you have just created to select it.	

Continues on next page

6.10 Configure the tools
Continued

	Action	Note/illustration
8	Fill in the parameters for the shape to specify the position and size of the geometry.	 xx1500000238 **Note** The minimum size of a geometry is 50 mm.

Rename a SafeMove tool

To change the name of a tool, right-click on the tool in the SafeMove browser, select **Rename** and type the desired name.

Change default tool

If more than one tool is configured, one of them must be selected as default tool. The default tool will be used at startup of the controller when no tool is selected by input signals. The last selected tool will still be active if the signal combination becomes faulty.

Select the tool that should be default. Click on the button **Make default Tool**.

Copying a tool

	Action	Note/illustration
1	Right-click on a tool and select **Copy**.	
2	Right-click on the tool again and select **Paste**.	A new tool is created with exactly the same data as the first tool. The parameter values can then be adjusted.

Deleting a tool

	Action
1	Right-click on a tool and select **Delete object**.

6.11 Configure safe I/O

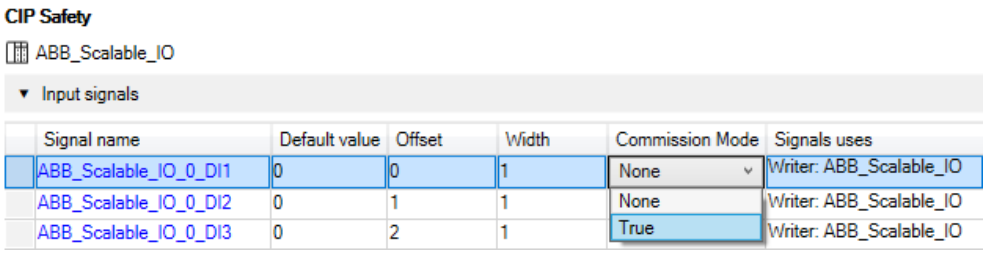
Configure safe signals

Any signals used by the SafeMove functions must be configured before they can be used in the function configurations.

The signals are configured in **IO Engineering** tool, see *Application manual - I/O Engineering*. The signals can be viewed from **Safe IO Configurator**.

Configure safe signals for commissioning mode

Any signal that should be high in commissioning mode must be configured. Set **Commissioning Mode** to *True*.



xx2100002332

Tip

For configurations created in older versions than RobotWare 7.10, click **Apply** when all changes are done.

Configure function mappings

Use the **Function mappings** view of the **Safe IO Configuration** to select which signal to use for each function.

xx1600000139

Continues on next page

> **Tip**
>
> For configurations created in older versions than RobotWare 7.10, click **Apply** when all changes are done.

Configure pre logic and post logic

For more detailed information about configuring combinatory logic including rules and limitations, see *Configuring combinatory logic on page 96* and *Safe I/O system rules and limitations on page 104*.

	Action	Note/illustration
1	Select the **Pre Logic** or **Post Logic** view in **Safe IO Configuration**.	
2	Click on **New expression**.	
3	At the bottom of the **Safe IO Configuration**, type the logical expression.	If the result signal is not already configured, click **Create signal** to automatically define it as a global signal.

> **Tip**
>
> It is recommended to configure the pre logic before configuring the supervision functions.
>
> It is recommended to configure the post logic after configuring the supervision functions. An error message is shown if a signal value is used in a logical expression without having a defined writer.

> **Tip**
>
> For configurations created in older versions than RobotWare 7.10, click **Apply** when all changes are done.

6.12 Configure the zones

Creating a zone

DANGER

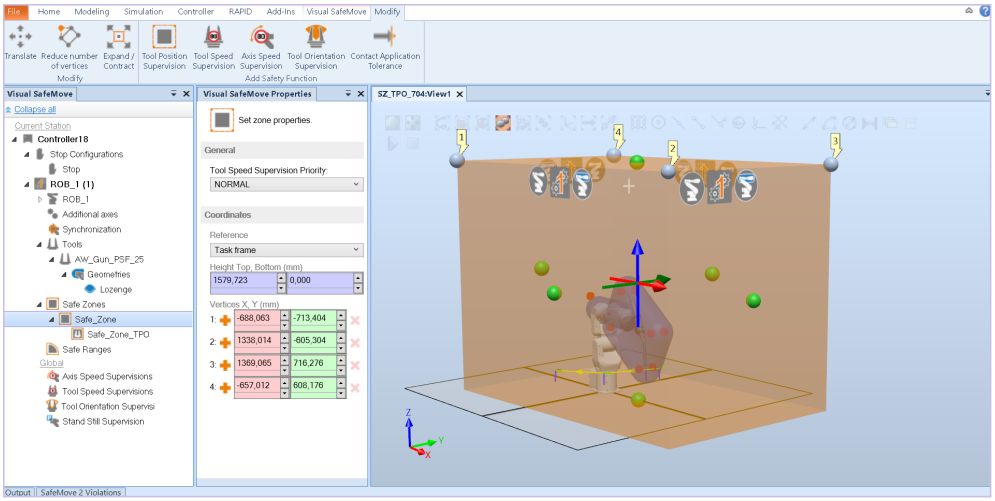
When setting zone limits, brake distances must be taken into consideration, so that the SafeMove functions are configured with enough margin. If the robot hits the zone limit, it starts to brake and needs the brake distance to stop. This occurs outside the allowed zone.

Note that if the robot starts accelerating strongly just before reaching a configured zone limit, there will occur a speed overshoot before decelerating. This may result in a somewhat increased speed and extended braking distance compared to a smoother speed situation.

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click on **Safe Zone**.	A zone of default size is shown in the graphics window. The minimum size shall be 40mm.
2	Specify **Tool Speed Supervision Priority** for the zone. If two zones are overlapping, the zone with the highest priority will set the speed limits for the overlapping space. If two overlapping zones have the same priority, the most restrictive values will be used in the overlapping space (lowest max speed and highest min speed).	**Tip** If there is a small zone with higher allowed speed (e.g. 1000 mm/s) inside a larger zone with lower allowed speed (e.g. 250 mm/s), use higher priority for the small zone. **Note** For a too thin zone, the robot may pass through its corners without reaction if the speed is too high.
3	Fill in the height of the box and the X and Y values for each corner.	If you want to state the coordinates in another coordinate system, select it in the field **Reference**. If the zone base should have another shape than square, add a corner point by clicking on a + button. For example, to add a new point between point 2 and 3, click on the + button at point 2. An alternative to writing coordinates in the **Visual SafeMove Properties** browser, is to click and drag on the corners or sides. By pressing the key X while dragging, the dragging is done along the X-axis while the Y value remains unchanged. By pressing the key Y while dragging, the dragging is done along the Y-axis while the X value remains unchanged.

Continues on next page

6.12 Configure the zones
Continued



xx2000001846

Tip

To change the name of a zone, right-click on the zone in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Zone supervision functions

About zone supervision functions

Each zone can be used by several zone supervision functions. There can be two Tool Position Supervision functions per zone, but only one function of each type per zone for all other zone supervision functions.

The default names of the functions are the zone name combined with:

- TPO for Tool Position Supervision
- TSP for Tool Speed Supervision
- ASP for Axis Speed Supervision
- TOR for Tool Orientation Supervision
- TFO for Tool Force Supervision
- CAP for Contact Application Tolerance

To change the name of a function, right-click on the function in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Creating zone supervision functions

	Action	Note
1	In the **Visual SafeMove** browser, click on the zone to select it.	
2	In the **Visual SafeMove** ribbon, click on the function to create.	
3	Configure the function according to *Configure the supervision functions on page 148*.	

6.13 Configure the ranges

Creating a range

DANGER

When setting range limits, brake distances must be taken into consideration, so that the SafeMove functions are configured with enough margin. If an axis hits the range limit, the robot starts to brake and needs the brake distance to stop. This occurs outside the allowed range.

Note that if the robot starts accelerating strongly just before reaching a configured range limit, there will occur a speed overshoot before decelerating. This may result in a somewhat increased speed and extended braking distance compared to a smoother speed situation.

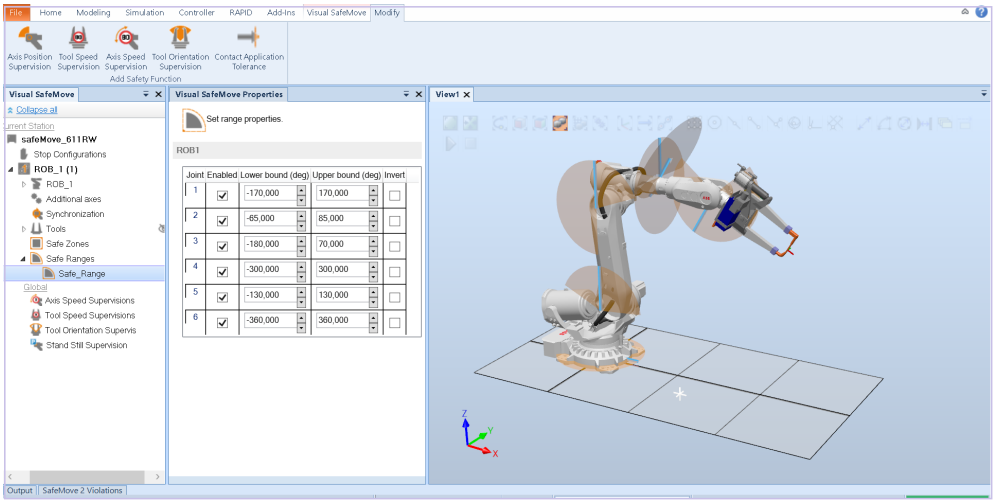
Tip

When configuring a positioner with interchange, then the safe range for the 0° and the 180° position of the interchange need to be configured.

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click on **Safe Range**.	A range of default size is shown for each axis in the graphics window.
2	Specify **Upper bound** and **Lower bound** for each axis.	For an interchange, for example, specify -1° and +1° for one safe range, and the other safe range 179° and 181°.
3	If an axis should be excluded from the safe range, clear the checkbox **Enabled** for that axis.	
4	To supervise the inverted range for an axis (below **Lower bound** and above **Upper bound**) select the checkbox **Inverted** for that axis.	

Continues on next page

6.13 Configure the ranges
Continued



xx2000001835

Tip

To change the name of a range, right-click on the range in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Range supervision functions

About range supervision functions

Each range can be used by several range supervision functions, but only one function of each type (for example, there cannot be two Axis Position Supervision functions for the same range).

The default names of the functions are the range name combined with:

- APO for Axis Position Supervision
- ASP for Axis Speed Supervision
- TSP for Tool Speed Supervision
- TOR for Tool Orientation Supervision
- CAP for Contact Application Tolerance

To change the name of a function, right-click on the function in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Creating range supervision functions

	Action	Note
1	In the **Visual SafeMove** browser, click on the range to select it.	
2	In the **Visual SafeMove** ribbon, click on the function to create.	
3	Configure the function according to the descriptions in section *Configure the supervision functions on page 148*.	

6.14 Global supervision functions

About global supervision functions

The default names of the global functions are:

- Global_ASP for Axis Speed Supervision
- Global_TSP for Tool Speed Supervision
- Global_TOR for Tool Orientation Supervision
- Global_TFO for Tool Force Supervision
- SST for Stand Still Supervision

There can be several global functions of the same type. A number is then added to the name, for example Global_ASP_1.

To change the name of a function, right-click on the function in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Creating global supervision functions

	Action	Note
1	In the **Visual SafeMove** browser, make sure that no zone or range is selected.	
2	In the **Visual SafeMove** ribbon, click **Global Functions** and select function.	
3	Configure the function according to *Configure the supervision functions on page 148*.	See also, *Local and global functions on page 34*.

6.15 Configure the supervision functions

Configuring Human Contact Supervision

Human Contact Supervision is only used as a calculation function for the tool speed and tool force/torque.

Configure the following settings.

	Action	Note
1	Select **Transient contact** or **Quasi-static contact**.	
2	Define the mass of the tooling and workpiece.	The function considers an estimate of the effective mass of the manipulator.
3	Define the surface area of the workpiece.	The calculation cannot be used for surface areas smaller than 1 cm^2.
4	Select which body areas are assumed to be exposed to hazard.	This is only used for transient contact.
5	Click **Create supervisions**. This creates a speed supervision and a tool force supervision.	There can be only one configuration for each zone.

See *Guidelines for transient and quasi-static contact, CRB 15000 on page 207*.

Configuring Tool Position Supervision

Tool Position Supervision is only used as a zone supervision function.

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

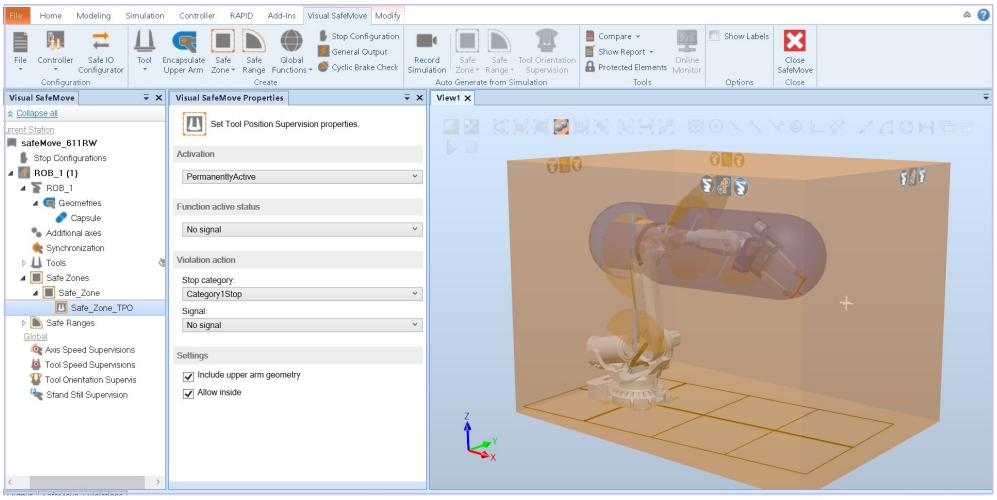
Continues on next page

	Action	Note
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
4	In the **Settings** part, configure if only the tool, or the tool and the upper arm geometry shall be supervised: • Select **Include upper arm geometry** if both the tool and upper arm geometry shall be supervised. If only the tool shall be supervised, clear the checkbox. • Select **Allow Inside** if the selection above (tool, or tool and upper arm geometry) is allowed to be inside the zone. If only outside the zone is allowed, clear the checkbox.	

Continues on next page

6.15 Configure the supervision functions
Continued

The following screenshot shows an example configuration of a permanently active tool position supervision where no signal is used.



xx2000001836

Configuring Tool Speed Supervision

Tool Speed Supervision can be used as:

- zone supervision function
- range supervision function
- global supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

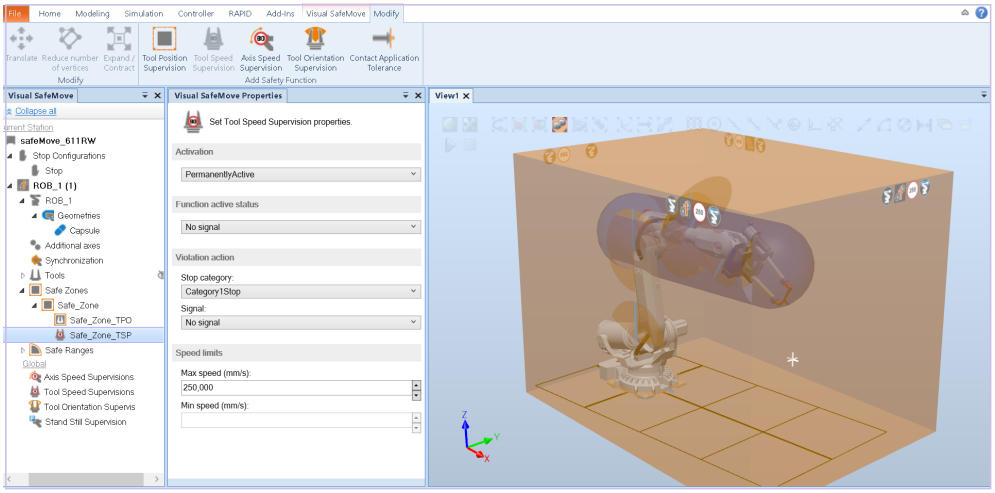
Continues on next page

	Action	Note/illustration
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
4	In the **Speed limits** part, configure the following: • Specify the maximum allowed speed in **Max speed**. The maximum speed limit must always be larger than the minimum speed limit. • To use a minimum TCP speed, specify the minimum allowed speed in **Min speed**.	If a minimum TCP speed shall be used, the function cannot be both permanently active and stopping the robot. The minimum speed limit has a lower supervision limit of 2 mm/s. The minimum speed limit supervision is disabled by entering a limit of 0 mm/s.

Continues on next page

6.15 Configure the supervision functions
Continued

The following screenshot shows an example configuration of a permanently active tool speed supervision where no signal is used.



xx2000001837

Configuring Axis Speed Supervision

Axis Speed Supervision can be used as:

- zone supervision function
- range supervision function
- global supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

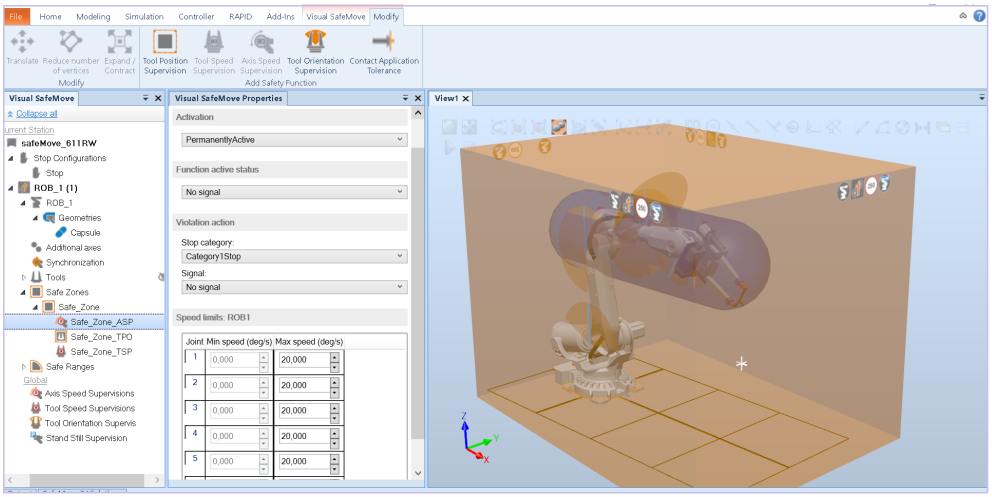
Continues on next page

	Action	Note/illustration
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
4	In the **Speed limits** part, configure the allowed minimum and maximum speed for each joint. The values are given for arm side in deg/s for rotating axes and mm/s for linear axes.	If minimum axis speeds shall be used, the function cannot be both permanently active and stopping the robot.

Continues on next page

6.15 Configure the supervision functions
Continued

The following screenshot shows an example configuration of a permanently active axis speed supervision where no signal is used, and only the maximum speed is configured.



xx2000001838

Configuring Tool Orientation Supervision

Tool Orientation Supervision can be used as:

- zone supervision function
- range supervision function
- global supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

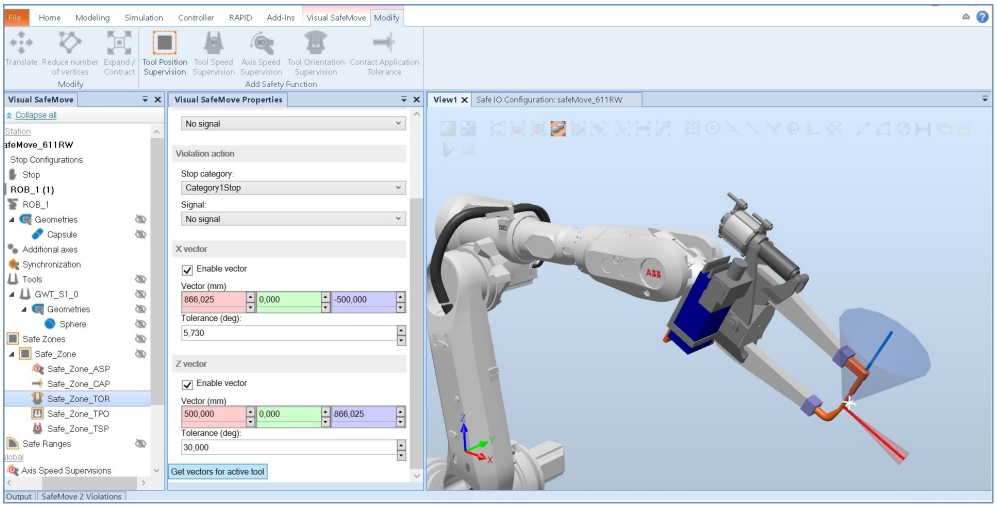
Continues on next page

	Action	Note/illustration
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
4	Jog the robot so that the tool is in desired orientation and click **Get vectors for active tool**. This will set both X and Z vector for the tool orientation. or In the **X vector** and **Z vector** parts, configure the allowed minimum and maximum speed for each joint.	To exclude supervision for either X vector or Z vector, clear the checkbox **Enable vector**. **CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
5	Specify the **Tolerance** for both the X vector and the Z vector.	The allowed orientation of X and Z are shown as cones in the graphics window.

Continues on next page

6.15 Configure the supervision functions
Continued

The following screenshot shows an example configuration of a permanently active tool orientation supervision where no signal is used, and both x and z vectors are configured.



xx2000001840

Configuring Tool Force Supervision

Tool Force Supervision can be used as:

- zone supervision function
- range supervision function

Note

If only torque limits are required, the force limit can be set to a high value.

Torque limits are relative an expected value.

See also *Limitations on page 61*.

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

Continues on next page

	Action	Note/illustration
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
4	In the **Force limits** part, configure the allowed maximum force applied to the tool.	The force applied by the work object needs to be taken into account when setting the limit.
5	In the **Axis Torque Supervision** part, select **Enable Axis Torque** if there are possible contact points between the robot arm and the operator.	
6	If axis torque supervision is enabled, configure**Torque Limits** for each joint.	

Continues on next page

6.15 Configure the supervision functions
Continued

Configuring Axis Position Supervision

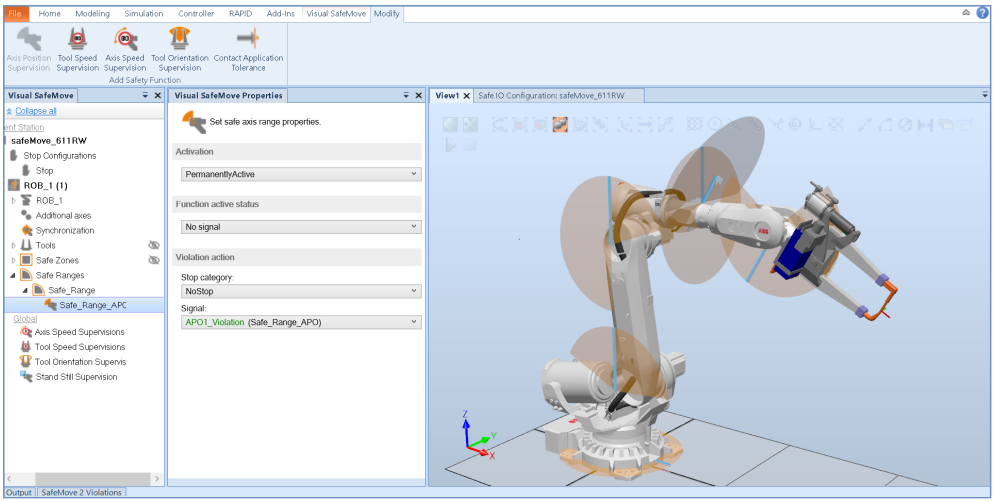
Axis Position Supervision is only used as a range supervision function.

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation

Continues on next page

The following screenshot shows an example configuration of a permanently active axis position supervision where a signal is set on violation.



xx2000001841

Configuring Contact Application Tolerance

Contact Application Tolerance can be used as:

- zone supervision function
- range supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

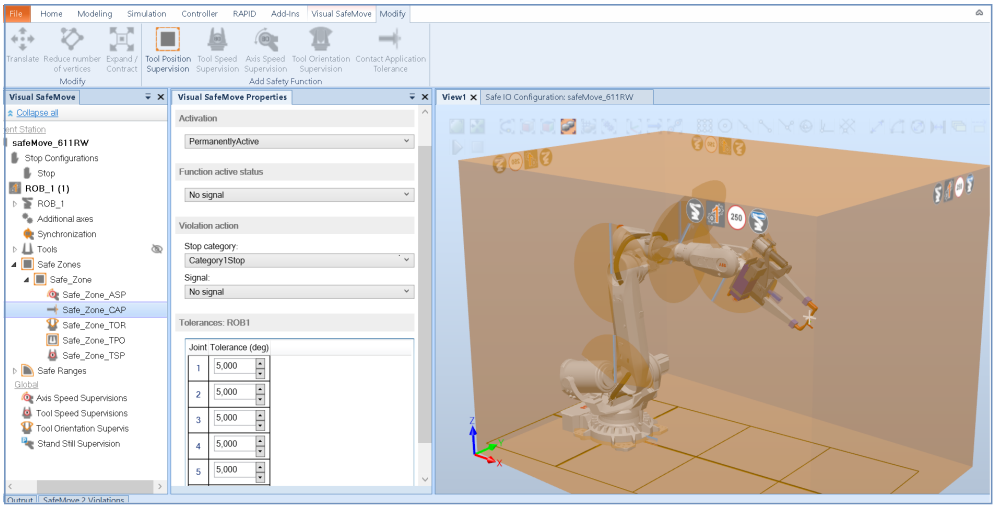
Continues on next page

6.15 Configure the supervision functions
Continued

	Action	Note/illustration
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Signal definitions: 0 = violation 1 = no violation
4	In the **Tolerances** part, configure how much deviation from ordered position that is tolerated for each axis.	Set as low tolerances as the applications permit, to keep the deviations as small as possible.

Continues on next page

The following screenshot shows an example configuration of a permanently active contact application supervision where no signal is used.



xx2000001842

Configuring Stand Still Supervision

Stand Still Supervision can be used as:

- global supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	In the **Activation** part, configure if the supervision function should be permanently active or activated/deactivated by a signal. If permanently active, then select **Permanently active**, otherwise select a signal.	Input signal definitions: 0 = activate function 1 = deactivate function
2	In the **Function active status** part, configure if the active status for the supervision function should be reflected on a signal. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (for example, stop and output signal) will be activated at violation. Signal definitions: 0 = at least one violation action is disabled 1 = all violation actions are enabled

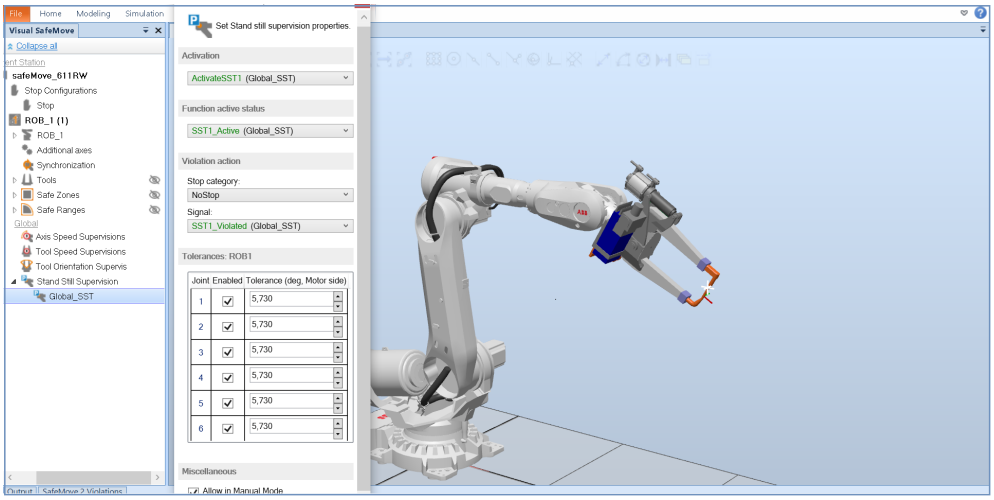
Continues on next page

6.15 Configure the supervision functions
Continued

	Action	Note/illustration
3	In the **Violation action** part, configure the stop category and if a signal should be used if a violation occurs. • In the **Stop category** dropdown menu, select if violation of the function should stop the robot with stop category 0, stop category 1, or not stop the robot. • In the **Signal** dropdown menu, select if a signal should reflect a violation. **CAUTION** The violation signal and the function active status shall be used in a post logic. This to ensure a valid signal when the safety controller is unsynchronized, see also *Software synchronization on page 41*, and *Hardware synchronization on page 43*.  xx2300001386 **CAUTION** The violation signal and *SafetyControllerOperational* shall be used in a post logic. This to ensure a valid signal, see also the description for *SafetyControllerOperational* in section *Function mappings on page 93*.  xx2300001387	Note that if **Activation** is **Permanently active**, then **Stop category** must be **NoStop**. Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
4	In the **Tolerances** part, enable supervision for desired joints by selecting **Enabled** and specify a maximum allowed tolerance (in degrees on motor side).	
5	In the **Miscellaneous** part, select **Allow in Manual Mode** if the supervision should be active in manual mode.	

Continues on next page

The following screenshot shows an example configuration where *ActiveSST1* is used, and *SST1_Violated* is set on violation.



xx2000001843

6.16 Configure other functions

Configuring Cyclic Brake Check

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click **Cyclic Brake Check**.	
2	If the robot should not be stopped when the test interval has elapsed, select the checkbox **Warning only, no stop**.	
3	In **Max CBC test interval**, set the maximum allowed time (in hours) between brake checks.	A value between 2 and 720 hours.
4	In **Pre warning time**, set how long before the end of the interval a warning should be shown on the FlexPendant.	A value between 1 and 11 hours.
5	Do not change the default value for **Standstill tolerance** unless absolutely necessary.	**Standstill tolerance** is used for Stand Still Supervision during brake test. The motor is in regulation during brake test, and a small movement may be allowed. The size of the allowed movement is specified in **Standstill tolerance** (in radians on motor side). Typical value is 2 radians.
6	Do not change the default value for **Supervision threshold** unless absolutely necessary.	**Supervision threshold** defines the threshold to verify that a brake check has been made.
7	If one axis should be excluded from the Cyclic Brake Check, clear the checkbox **Enabled** for that axis.	This must correspond with the axes that has the system parameter *Deactivate Cyclic Brake Check for axis* set to *On*. For axes not included in SafeMove, deactivation of the axes must be done by setting the parameter *Deactivate Cyclic Brake Check for axis* to *On* via RobotStudio for all axes not included.

For more information, see *Cyclic Brake Check guidelines on page 200*.

Configuring Safe Payload Supervision

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click **Safe Payload Supervision**.	
2	Select **Add Zone** to create up to four payload-change zones, where changing the payload should be allowed.	
3	Configure up to 8 payloads. Each payload contains mass and load data, and a flag to define whether the payload mass data is specified relative to wrist or tool coordinates.	**Note** The zero load load0 is always configured by default and should not be added to the safety configuration.
4	Configure the **Payload Verification Reference TCP Speed** of the movement used to verify the current payload. A value between 0 and 250 mm/s.	

Continues on next page

	Action	Note/illustration
5	Configure the **Payload Verification Reference Tool Orientation** for the movement used to verify the current payload. The orientation is defined relative to the world coordinate system.	
6	Optionally, configure **Max Payload Uncertainty**.	**Note** An acceptable uncertainty for the mass of the payload. This can be useful if there are two or more payloads with similar mass data, so that *Safe Payload Supervision* cannot reliably determine which payload is held by the robot, see *Safe Payload Supervision (SPA) on page 48*. The user is responsible for taking the configured uncertainty into account when configuring the force and torque limits in TFO.

Configuring Laser Scanner Calculation

To enable the add-in **Laser Scanner Calculation** for GoFa robots, the following must be configured first.

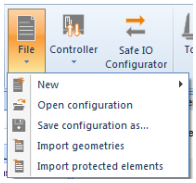
1. Create a tool with collision geometry.
2. Create a global tool speed supervision.
3. Set the global tool speed supervision to be activated by a signal.

   This enables the button for the add-in.
4. Click **Laser Scanner Calculation** in the **Tools** group in the ribbon.

The add-in has its own documentation, found in the help section of RobotStudio.

6.17 Save the configuration

Saving the configuration

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click on **File** and then select **Save configuration as**.	 xx1500000802
2	Select a file name and location for the file. Click on **Save**.	

Loading a saved configuration

Note

Safety configurations created on a controller with main computer DSQC1029 cannot be loaded in main computer DSQC1095.

	Action
1	In the **Visual SafeMove** ribbon, click on **File** and then select **Open configuration**.
2	Browse and select a file. Click on **Open**.

Note

The configuration file can only be edited using RobotStudio. Changing the configuration file in any other way will make the file invalid and it is not possible to run the robot if this file is loaded.

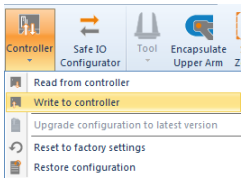
6.18 Load the configuration to the safety controller

6.18.1 Loading a safety configuration using RobotStudio

Writing configuration to the safety controller

Note

Safety configurations created on a controller with main computer DSQC1029 cannot be loaded in main computer DSQC1095.

	Action	Note/illustration
1	Log in as a user with the grant **Safety Services**.	See *Set up safety user grants on page 130*.
2	In the **Visual SafeMove** ribbon, click on **Controller** and then select **Write to controller**.	 xx1500000801
3	A report of the safety configuration is shown. The report can be printed by clicking on **Print** (it is recommended to print the report since it should be used when validating the configuration as described in *Validate the configuration on page 168*). Click OK to close the report.	
4	Answer **Yes** when asked if you want to restart the controller.	After the restart, the downloaded configuration is active. Before running in auto mode, the configuration should be validated and locked, see *Validate the configuration on page 168*.

Reading the configuration from safety controller

It is possible to upload the configuration from the safety controller to Visual SafeMove. This makes it easy to view the configuration or to make changes to it and download it again.

	Action
1	In the **Visual SafeMove** ribbon, click on **Controller** and then select **Read from controller**.

6.19 Validate the configuration

DANGER

A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety. It is the responsibility of the owner of the robot to perform the safety validation.

About the validation

A modification done to the safety configuration *must* be validated before running in production. If the configuration is set up with protected groups, then the validation must validate the modified part. Hence, a protected group with a checksum that is unchanged, does not need a new validation. However, it is recommended to at least visually inspect the resulting configuration in the configurator to see that it is correct. Any modified checksum will result in an event log being issued. For example, modifications to any of the following will require a validation of the respective functionality.

- All settings for Safe Disable of Drive Unit of robots and additional axes
- All I/O settings and signals used for safety interlocking including connected functionality
- All Pre and Post logic used for safety interlocking
- All function mappings used for safety interlocking
- All Stop configuration functions
- All safety ranges with connected supervision functions including connected signals used for safety interlocking
- All safety zones with connected supervision functions and signals used for safety interlocking
- All global supervision functions
- Upper arm encapsulation in combination with Tool Position Supervision (if used)
- All tools with corresponding supervision functions (Tool Position Supervision, Tool Speed Supervision and Tool Orientation Supervision)

Note

Depending on the combination of functions and how they are activated, the validation procedures described below may have to be modified for the specific configuration.

Tip

Use RAPID programs in order to perform testing faster and to be able to repeat them.

Continues on next page

Preparations before validation

Do the following checks before you start the validation procedure:

1. Carry out the synchronization procedure. See *Synchronization guidelines on page 197*.
2. If configured, run the service routine for the function Cyclic Break Check. See *Cyclic Brake Check guidelines on page 200*.
3. Turn off the *SafeMove Assistant* functionality, with the system parameter *Disable SafeMove Assistant*.
4. Turn off collision detection during validation of any tool force supervision
5. Start the validation procedure.

   If using protected groups in the safety configuration, only the modified parts must be validated.

ABB Safety Configuration Report

The validation of each function should be documented in the safety report by signature of the validator.

The safety configuration report lists all parameters that are set for the installation. The report also includes a visual representation of the installation, a floor plan. This shows the robot and safety zones as seen from above.

The configuration report includes the checksum (multiple checksums if using protected groups in the safety configuration). The checksum can also be read using the RAPID function `SafetyControllerGetChecksum` or `SafetyControllerGetGroupChecksum`.

Recovery after safety violation

The validation procedures test when the safety functions trigger. If the functions Axis Position Supervision, Tool Position Supervision or Tool Orientation Supervision triggers with a stop, recovery is achieved by performing the following:

	Action
1	Select manual operating mode.
2	Press the enable device and jog the robot into a position in which the supervision functions are not in violation.

Validate the safe fieldbus and signal configuration

Validate the safe fieldbus parameters, including I/O settings and signals used for safety interlocking, by comparing the safety report with the configured values.

The user must visually verify that the data in the safety report is correct and that it is the same as entered in the Visual SafeMove configuration.

After controller restart the configuration is applied. The user must verify that no safe fieldbus related event logs were generated, that the status of the connection in the originator indicates "running", and that the connection to the intended adapter device has been established.

Continues on next page

6.19 Validate the configuration
Continued

Validate range limits

> **Tip**
>
> To validate that the ranges have enough margins, let the robot move with maximum allowed speed when reaching the range limits. Verify that the limits are set with enough margin with respect to the breaking distance.

Validate the range limits for all configured ranges. It is enough to validate the limits of each range for one function, since the range limits are treated equally for all functions in the range. Perform the range limits validation using one of the following functions (preferably Axis Position Supervision).

Validate range using Axis Position Supervision

If a range has an Axis Position Supervision function, perform the validation for that function. See *Axis Position Supervision validation on page 177*.

Validate range using Axis Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Axis Speed Supervision for the range. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves one configured axis at a speed that is allowed outside the range but not allowed inside the range. At the same time the program shall test one axis at a time by moving it from outside the range to inside the range. Note that the violating speed axis does not have to be the same axis as the one you are testing the range limits for.	Axis Position Supervision will trigger when the tested axis reach its range limit.
3	Repeat this for both lower and upper limit of the axis range.	
4	Repeat this for all axes configured in the range, including additional axes.	

Validate range using Tool Orientation Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Orientation Supervision for the range. Deactivate all other supervision functions that are signal activated.	
2	Jog the robot (reorient jogging) to an orientation that is allowed outside the range but not inside the range.	
3	Move the robot, one axis at a time, from outside the range to inside the range.	Tool Orientation Supervision will trigger when the tested axis reach its range limit.
4	Repeat this for both lower and upper limit of the axis range.	
5	Repeat this for all axes configured in the range, including additional axes.	

Continues on next page

Validate range using Tool Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Speed Supervision for the range. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves the tool at a speed that is allowed outside the range but not allowed inside the range. At the same time the program shall test one axis at a time by moving it from outside the range to inside the range.	Tool Speed Supervision will trigger when the tested axis reach its range limit.
3	Repeat this for both lower and upper limit of the axis range.	
4	Repeat this for all axes configured in the range, including additional axes.	

Validate range using Contact Application Tolerance

	Action	Expected result
1	Make sure that Soft Servo is active and set the stiffness to a value that is allowed outside the range but not allowed inside the range.	
2	If not permanently active, activate the activation input signal for Contact Application Tolerance for the range. Deactivate all other supervision functions that are signal activated.	
3	Run a RAPID program moving the TCP with maximum speed allowed (e.g. `vmax`). At the same time the program shall test one axis at a time by moving it from outside the range to inside the range.	Control Error Supervision will trigger when the tested axis reach its range limit.
4	Repeat this for both lower and upper limit of the axis range.	
5	Repeat this for all axes configured in the range, including additional axes.	

Validate zone limits

Tip

To validate that the zones have enough margins, let the robot move with maximum allowed speed when reaching the zone limits. Verify that the limits are set with enough margin with respect to the braking distance.

Validate the zone limits for all configured zones. It is enough to validate the limits of each zone for one function and one tool, since the zone limits are treated equally for all functions and tools in the zone. Perform the zone limits validation using one of the following functions (preferably Tool Position Supervision).

Validate zone using Tool Position Supervision

If a zone has a Tool Position Supervision function, perform the validation for that function. See *Tool Position Supervision validation on page 174*.

Continues on next page

6.19 Validate the configuration
Continued

Validate zone using Tool Orientation Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Orientation Supervision for the zone. Deactivate all other supervision functions that are signal activated.	
2	Jog the robot (reorient jogging) to an orientation that is allowed outside the zone but not inside the zone.	
3	Move the TCP from outside the zone to inside the zone.	Tool Orientation Supervision will trigger when the TCP reach the zone limit.
4	Repeat this for at least two points on each side of the zone, including top and bottom.	
5	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Validate zone using Tool Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Speed Supervision for the zone. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves the tool at a speed that is allowed outside the zone but not allowed inside the zone. The tool shall be moved at this speed from outside the zone to inside the zone.	Tool Speed Supervision will trigger when one of the configured speed supervision points (or the TCP) reach the zone limit.
3	Repeat this for at least two points on each side of the zone, including top and bottom.	
4	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Validate zone using Axis Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Axis Speed Supervision for the zone. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves one configured axis at a speed that is allowed outside the zone but not allowed inside the zone. At the same time the program shall move the TCP from outside the zone to inside the zone.	Axis Position Supervision will trigger when the TCP reach the zone limit.
3	Repeat this for at least two points on each side of the zone, including top and bottom.	
4	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Continues on next page

Validate zone using Contact Application Tolerance

	Action	Expected result
1	Make sure that Soft Servo is active and set the stiffness to a value that is allowed outside the zone but not allowed inside the zone.	
2	If not permanently active, activate the activation input signal for Contact Application Tolerance for the zone. Deactivate all other supervision functions that are signal activated.	
3	Run a RAPID program moving the TCP with maximum speed allowed (e.g. `vmax`). The tool shall be moved at this speed from outside the zone to inside the zone.	Control Error Supervision will trigger when the reference position of the TCP reach the zone limit.
4	Repeat this for at least two points on each side of the zone, including top and bottom.	
5	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Validate the tools

If no tool changer is used, the tool geometry, speed supervision points and TCP are validated in the respective functions. If more than one tool is configured, each tool must be validated according to the following.

Note

It is not necessary to validate each tool with each range/zone. It is enough if all ranges/zones are validated with one tool and all tools are validated with one range/zone.

Validate tool geometry

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Position Supervision set to use. Deactivate all other supervision functions that are signal activated.	
2	Move the robot so that the configured geometry surrounding the tool crosses the border of the configured zone. Verify that Tool Position Supervision triggers when the border is crossed.	Tool Position Supervision will trigger.
3	Rotate the tool and repeat the test with different parts of the tool crossing the zone border. Make sure Tool Position Supervision always triggers on the surrounding geometry before the tool itself reach the zone border.	

Validate tool speed supervision point

For each tool, perform the validation for Tool Speed Supervision with that tool. See *Tool Speed Supervision validation on page 175*.

Continues on next page

6.19 Validate the configuration
Continued

Validate the TCP position

For each tool, validate the TCP position if possible. It is possible to validate the TCP position with the following zone supervision functions:

- If a tool does not have speed supervision points surrounding the TCP, it is possible to validate the TCP position using Tool Speed Supervision. See *Tool Speed Supervision validation on page 175*.
- Tool Orientation Supervision can be used to validate the TCP position. See *Validate zone using Tool Orientation Supervision on page 172*.
- Contact Application Tolerance can be used to validate the TCP position. See *Validate zone using Contact Application Tolerance on page 173*.

If none of these criteria is met for a tool, the TCP position cannot be validated for that tool, but then it is not necessary to validate it.

Note

The TCP for the SafeMove tool must correspond with the TCP of the active tool for jogging and RAPID instructions.

Verify tool mass data

For each tool, perform the validation for Tool Force Supervision with that tool. See *Tool Force Supervision validation on page 176*.

If Tool Force Supervision is not used, validation of the tool mass data is not required.

Validate payload mass data

For each payload, including the default zero load `load0`, perform the validation for *Safe Payload Supervision* using the payload. See *Safe Payload Supervision validation on page 179*. Then perform the validation for *Tool Force Supervision* using the payload. See *Tool Force Supervision validation on page 176*.

If the function *Safe Payload Supervision* is not used, validation of the payload mass data is not required.

Tool Position Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Position Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
3	Move the robot so that the configured geometry surrounding the tool crosses the border of the configured zone. Verify that Tool Position Supervision triggers when the border is crossed.	Tool Position Supervision will trigger.
4	Repeat this for at least two points on each side of the zone, including top and bottom.	

Continues on next page

	Action	Expected result
5	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Tool Orientation Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Orientation Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If applicable, make sure the TCP is inside the Tool Orientation Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
4	Jog the robot (reorient jogging) to the tolerance limits of the tool orientation. Verify that Tool Orientation Supervision triggers for violation of both the tool's x direction and the tool's z direction.	Tool Orientation Supervision will trigger.

Tool Speed Supervision validation

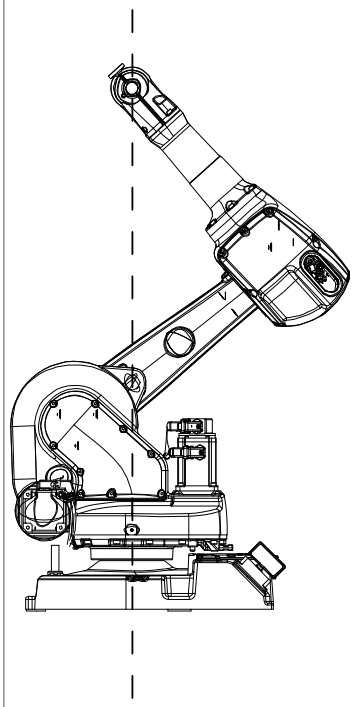
Validate all points supervised by Tool Speed Supervision:

- tool center point (TCP)
- robot elbow (near axis 3 according to configuration)
- any configured speed supervision points for the tool

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Speed Supervision. Deactivate all other supervision functions that are signal activated.	
2	If applicable, make sure the TCP is inside the Tool Speed Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
4	Create and run a RAPID program with a `MoveL` instruction. The `Speed` argument should be slightly higher than the configured max speed. The `Tool` argument should be set to the tool that is to be supervised by Tool Speed Supervision.	Tool Speed Supervision will trigger.
5	Test the speed supervision of all configured speed supervision points of the tool. How to do this test depends on the position of the points. Usually a rotation of the tool can make a configured speed supervision point move faster than the TCP. Use `MoveAbsJ`, moving axis 5 or 6, to rotate the tool.	Tool Speed Supervision will trigger.

Continues on next page

6.19 Validate the configuration
Continued

	Action	Expected result
6	Jog the robot to a position where the elbow is pointing out as much as possible, while the tool is close to the rotation axis of axis 1.  xx0700000696 Create and run a RAPID program with a `MoveAbsJ` instruction moving axis 1 fast enough for the elbow to exceed the configured max speed.	Tool Speed Supervision will trigger.

Tool Force Supervision validation

For the validation of Tool Force Supervision, the collision detection functionality should be disabled. This can be done by temporarily setting the system parameter *Path Collision Detection* to *Off*, in the type *Motion Supervision* in the topic *Motion*.

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Force Supervision set that you want to validate. Deactivate all other supervision functions that are activated by signals.	
2	If applicable, make sure that the TCP is inside the Tool Force Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
4	Program a movement in such a way that the robot tool applies force against some type of force measuring device. Verify that Tool Force Supervision triggers when the configured force limit is exceeded.	Tool Force Supervision will trigger.

Continues on next page

	Action	Expected result
5	For each joint with a configured torque limit, program a movement in such a way that the robot arm applies a force against some type of force measuring device, causing a well-defined torque on the specific joint. The position of the robot arm should be chosen so that the joint torque can be calculated from the measured force and the lever arm vector connecting the contact point and the joint axis. Verify that Tool Force Supervision triggers with a joint torque violation when the joint torque exceeds the configured torque limit for the joint.	Tool Force Supervision will trigger with a joint torque violation for the joint.

Axis Position Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Axis Position Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
3	Move the robot, one axis at a time, from inside the allowed range to outside the range. Verify that Axis Position Supervision triggers when the axis reaches the range limit.	Axis Position Supervision will trigger.
4	Repeat this for both lower and upper limit of the axis range.	
5	Repeat this for all axes configured for Axis Position Supervision, including additional axes.	

Axis Speed Supervision validation

Tip

To determine which `speeddata` settings to use for validation, use an `MoveAbsJ` instruction to move relevant axis and check the axis speed in signal analyzer in RobotStudio. Change the speed of the `MoveAbsJ` instruction until the desired axis speed is achieved.

	Action	Expected result
1	If not permanently active, activate the activation input signal for Axis Speed Supervision. Deactivate all other supervision functions that are signal activated.	
2	If applicable, make sure the TCP is inside the Axis Speed Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.

Continues on next page

6.19 Validate the configuration
Continued

	Action	Expected result
4	Create a RAPID program with a `MoveAbsJ` instruction moving the first configured axis with a speed slower than the configured Max Speed for that axis. Run the program in auto or manual full speed mode.	No triggered function.
5	Change the program so that the axis is moved with a speed higher than the configured Max Speed. Run the program in auto or manual full speed mode.	Axis Speed Supervision will trigger.
6	Repeat this for all axes configured for Axis Speed Supervision.	

Stand Still Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Stand Still Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
3	Create a RAPID program with `MoveJ` instructions that move one axis at a time at maximum allowed speed. Let the program wait for user input between each move instruction to let you validate each move instruction. If there are additional axes in the system, include instructions that move these axes, one at a time, as well.	
4	Run the program in auto or manual full speed mode and verify that Stand Still Supervision triggers for every move instruction.	Stand Still Supervision will trigger.
5	If the Stand Still Supervision is configured to stop at violation, determine that the movement until the robot is stopped again is within a tolerable limit.	
6	If the Stand Still Supervision is configured to set an output signal at violation, verify that the signal is 0 when the robot is moving and 1 when the robot is standing still.	

Contact Application Tolerance validation

Contact Application Tolerance only needs to be configured when using Soft Servo, Force Control or when external forces are applied to the robot. It cannot be verified unless it is used in one of these ways.

	Action	Expected result
1	Make sure that Soft Servo is active and set the stiffness to a reduced value (but not lower than needed by the application).	
2	If not permanently active, activate the activation input signal for Contact Application Tolerance. Deactivate all other supervision functions that are signal activated.	
3	Make sure the TCP is inside the Contact Application Tolerance zone or all axes inside the range.	

Continues on next page

	Action	Expected result
4	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
5	Run the application and verify that Contact Application Tolerance does not trigger within the range/zone to verify that the tolerance is high enough.	
6	To verify that the tolerance values are as low as possible, change the Soft Servo stiffness to a lower value (e.g. 10% lower). Run a RAPID program moving the TCP with maximum speed allowed in the range/zone (e.g. `vmax`) inside the range/zone for Contact Application Tolerance.	Control Error Supervision will trigger.

Safe Payload Supervision validation

	Action	Expected result
1	Place a payload to be picked up by the robot inside the payload-change zone.	
2	In automatic mode, perform payload selection and verification of the payload as described in *Using Safe Payload Supervision to change payload on page 189*.	Event log 90556 is correctly indicating the new payload.

Configured stop validation

	Action	Expected result
1	Deactivate all supervision functions that are signal activated.	
2	Move the robot, for example with a move instruction.	
3	Set the signal configured to stop the robot in relevant operating modes. Relevant operating modes are: • **Auto**: Automatic mode • **General**: All modes • **EmergencyStop**: All modes	The robot will stop.

Safe Disable of Drive Unit validation for additional axes

For every additional axis with Safe Disable of Drive Unit configured, perform the following steps.

	Action	Expected result
1	Deactivate all supervision functions that are signal activated.	
2	Move the additional axis while all other axes in the system are stationary, for instance with a Move instruction.	
3	While moving, activate Safe Disable of Drive Unit for the moving additional axis.	Event log 90560 correctly indicating the moving axis.
4	While Safe Disable of Drive Unit is still active, move the program pointer, and go to state motors on.	

Continues on next page

	Action	Expected result
5	While Safe Disable of Drive Unit is still active, move the additional axis, for instance with a move instruction.	The additional axis does not move. Other axes in the system can be jogged.
6	If used, verify that the corresponding **Status Signal** is set.	Status signal is 1.

Safe Disable of Drive Unit validation for robots

For every robot with Safe Disable of Drive Unit configured, perform the following steps.

	Action	Expected result
1	Deactivate all supervision functions that are signal activated.	
2	Move one axis of the robot while all other axes in the system are stationary, for instance with a Move instruction.	
3	While moving, activate Safe Disable of Drive Unit for the robot.	Event log 90560 correctly indicating the moving axis.
4	While Safe Disable of Drive Unit is still active, move the program pointer, and go to state motors on.	
5	While Safe Disable of Drive Unit is still active, move the same robot axis as before, for instance with a move instruction.	The robot does not move. Other axes in the system can be jogged.
6	If used, verify that the corresponding **Status Signal** is set.	Status signal is 1.
7	Repeat these steps for each axis of the robot.	

Verify the checksum

The checksum covers the entire installation. If using protected groups in the configuration, then there is one checksum for each protected group, and any modified checksum must be verified. This step verifies that the report corresponds to the loaded configuration.

Tip

Any modified checksum will result in the event log *90869 Protected elements group was modified* being issued. Therefore, it is helpful to check the event logs to verify that only the expected changes are there.

	Action	Expected result
1	Open the report and locate the checksum.	
2	Open the configuration on the FlexPendant.	
3	Compare the checksum in the report with the checksum shown on the FlexPendant.	The checksums are identical.

Continues on next page

Setting the configuration to validated

When the safety technician has validated the configuration and signed the safety report, the status of the configuration shall be changed to **Validated** on the FlexPendant.

1. Log in as a user with the grant **Safety Services**.
2. In the **Settings** app, select the **Safety Controller**, and then **Configuration**.
3. Select the checkbox **Validated**.

Setting the configuration to locked

When the responsible safety user has approved the validation of the configuration, the status of the configuration should be changed to **Locked** on the FlexPendant.

Running the robot in auto mode with the configuration unlocked will result in a warning message.

1. Log in as a user with the grant **Lock Safety Controller Configuration**.
2. In the **Settings** app, select the **Safety Controller**, and then **Configuration**.
3. Select the checkbox **Locked**.

Concluding steps

After the validation is concluded, turn on the the *SafeMove Assistant* functionality, with the system parameter *Disable SafeMove Assistant*.

6.20 Restore configuration

Restore configuration from backup

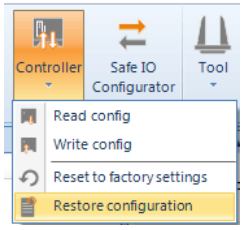
When performing a system backup, a SafeMove safety configuration file is included. This file gives the possibility to restore the SafeMove safety configuration without changing it. The advantage is that the configuration is identical, so the SafeMove safety configuration does not have to be validated and no new safety report has to be generated.

The safety settings includes the safety configuration, the EIO configuration, and the version information.

When restoring a backup, select the **Include Safety Settings** check box to also restore the safety configuration.

Restore configuration from RobotStudio

The file can also be restored separately from the system backup by using the **Restore Configuration** function in **Visual SafeMove**.

	Action	Note/illustration
1	Open **Visual SafeMove**.	
2	Click on **Controller** in the **Visual SafeMove** ribbon and select **Restore Configuration**.	 xx1500000533
3	Browse to the *BACKINFO* folder in the backup.	
4	Select the file *sc_cfg.xml* and click **Open**.	
5	The **Restart Controller** dialog is shown when the safety configuration has been downloaded. Click **Yes** to restart the controller.	

6.21 Reset safety controller to factory settings

Introduction

The function **Reset safety controller to factory settings** clears all user settings and loads a default configuration. The firmware of the safety controller is not affected.

In rare cases the safety controller can get locked in safety state, for example when loading an incompatible or poorly configured safety configuration or replacing the robot controller. Then it is not possible to load another safety configuration without first resetting the safety controller to factory settings.

Resetting the safety controller

Use this procedure to reset the safety controller from the FlexPendant:

1. On the start screen, tap **Settings**, and then select **Backup & Recovery** from the menu.
2. On the sidebar tap **Reset user data**.
3. Select the check box **Reset safety settings**.
4. Tap **Reset**.
5. Synchronize the safety controller with the robot controller, see *Synchronization guidelines on page 197*.
6. A new safety configuration can now be loaded and validated, see *Load the configuration to the safety controller on page 167*.

> **Note**
>
> It is also possible to reset the safety controller to factory settings from RobotStudio, see *Configuration group on page 72*.

6.22 Upgrading and installing new systems

Upgrading RobotWare

If an upgrade of RobotWare is done by using the **Modify Installation** function, the safety configuration is kept as it is. A restore of the safety configuration is not needed.

Since the configuration is kept as it is, the checksum is the same and the configuration is automatically locked after upgrade.

Downgrading RobotWare

SafeMove is not forward compatible. A safety configuration from a newer system is not possible to use in an older system.

Installing a new system

After installing a new system, or after a factory reset, the safety configuration is empty and no safety functionality is active.

Use this procedure when installing a new system:

	Action
1	Backup the old system.
2	Install the new system. If the configuration was locked in the old system, there will be an error message at startup.
3	Restore the backup and include the safety settings, see *Restore configuration on page 182*.
4	Restart the system. If the configuration was locked in the old system, then it will also be locked in the new system.

7 Running in production

7.1 Reaction time

Stopping response time

When a stopping supervision function is triggered, the reaction time until a stop is ordered is maximum 4 ms. This reaction time must be added to the stopping time of the manipulator, specified in the product specification for stopping distances and times (see *References on page 10*), to get the complete stopping time for the manipulator.

Response time to external stops (discrete)

When a discrete external stop is triggered, the maximum reaction time until a stop is ordered is 24 ms. This reaction time must be added to the stopping time of the manipulator, specified in the product specification for stopping distances and times (see *References on page 10*), to get the complete stopping time for the manipulator.

Response time to external stops (safe bus)

When an external stop is triggered via safe bus, the maximum reaction time until a stop is ordered is 10 ms. This reaction time must be added to both the cycle time of the safe bus and the stopping time of the manipulator, specified in the product specification for the respective manipulator, to get the complete stopping time for the manipulator.

Output signal response time

The time from when the robot makes a safety violation, for example entering a forbidden zone, until a safe Ethernet signal is generated shall not exceed 24 ms.

Note

The time depends on the cycle time on the communication with the PLC. 24 ms is the case when the cycle time is set to 8 ms.

Note

When a signal is set to 0 at violation, it will remain 0 for at least 250 ms even after the violation has ended.

7.2 Restarting the controller

Restart modes

None of the restart modes **Restart**, **Reset RAPID**, or **Reset system** will affect the safety configuration.

For more information about restart procedures, see *Operating manual - Integrator's guide OmniCore*.

Removing the current system

If the current system is deleted, all safety configurations for that system will also be deactivated and deleted. After installing a new system, the safety configuration must be downloaded to the safety controller again by an authorized user, and the configuration must be validated.

If a backup is available, then the SafeMove safety configuration can be restored without the need of a validation, see *Restore configuration on page 182*.

DANGER

Installing a new system, without downloading or restoring the safety configuration to the safety controller, leaves the robot system without any of SafeMove's safety functions. It can easily be perceived as if the robot system still has SafeMove active, which causes a dangerous situation.

Tip

Set up the User Authorization System so that only the safety user is allowed to administrate installed systems.

Backup and restore

Performing a backup and restore of the system does not affect the SafeMove safety configuration, unless that is selected in the user interface.

The SafeMove safety configuration file is included in the backup. For information on how to restore the SafeMove safety configuration, see *Restore configuration on page 182*.

7.3 Recovery after safety violation

Recovery after a supervision function has triggered in Automatic mode

When a supervision function triggers and the robot stops, perform the following to be able to move the robot again.

Speed and stand still violations

Press the motors on button on the robot controller to confirm the violation.

Position and orientation violations

	Action
1	Switch to Manual mode on the robot controller.
2	Jog the robot back to a position that does not trigger any supervision function.

Recovery after a supervision function has triggered in Manual mode

When a supervision function triggers and the robot stops, perform the following to be able to move the robot again.

Speed and stand still violations

Release and activate the three-position enabling device on the FlexPendant.

Position and orientation violations

	Action
1	Release and activate the three-position enabling device on the FlexPendant.
2	Jog the robot back to a position that does not trigger any supervision function.

Recovery after a two-channel fault

If a two-channel fault has occurred, then the functionality of the safety inputs shall be validated.

Recovery from unsynchronized state

	Action	Note
1	Press the motors on button on the robot controller.	This allows the robot to be moved at reduced speed.
2	Perform a synchronization.	

7.4 Changes to robot or robot cell

Always update safety configuration

If the following is done the safety configuration must be updated and validated again:

- A new version of the safety controller software is installed.

Evaluate if the safety configuration needs to be updated

If any of the following is done, the safety responsible person must evaluate if the safety configuration needs to be updated and validated again:

- The tool is replaced.
- Any robot part is replaced.
- The robot cell is rebuilt in any way.
- The relation between the world coordinate system and the robot base coordinate system is changed.
- The tool coordinate system is changed.
- Changes to system parameters.

Perform synchronization

If any of the following is done, a new synchronization is required:

- Revolution counter update
- Fine calibration
- Axis calibration

7.5 Using Safe Payload Supervision to change payload

Selecting a new payload

When picking up or releasing an object, use the RAPID instruction `GripLoad` to select the new payload. The `loaddata` in the `GripLoad` instruction must be identical to a configured payload in SafeMove. When executing a `GripLoad` instruction, SafeMove will start monitoring the joint torques and robot movement to safely identify the correct payload, as described in the next section. The `GripLoad` instruction is only allowed by SafeMove when the TCP is inside one of the payload-change zones configured in the safety function *Safe Payload Supervision*. See *Configuring Safe Payload Supervision on page 164*.

Note

If a `GripLoad` instruction is executed from a RAPID `Start` or `Reset` event routine, the `GripLoad` command is ignored by SafeMove. This can be used to make sure that the correct payload is used in RAPID before the SafeMove payload has been selected and verified, for example, after a restart of the controller, or if the payload in RAPID has been reset. Unlike the payload in RAPID, the payload in SafeMove is NOT reset when loading a new program or module, starting program execution from the beginning, or when moving the program pointer.

Verifying a new payload using Safe Payload Supervision

Use the RAPID instruction `MoveL` to execute a linear verification movement out of the payload-change zone in the positive z-direction, and with the **Payload Verification Reference TCP speed** and **Tool Orientation** configured in the safety function *Safe Payload Supervision*. For information on configuration parameters, see *Configuring Safe Payload Supervision on page 164*.

SafeMove will monitor the joint torque measurements compared to the torque values predicted by its internal dynamic models, in order to determine which of the configured payloads that could potentially match the measurements during the movement. If the payload is verified correctly, the active payload is changed to the selected payload when exiting the payload-change zone.

Example RAPID code

The following RAPID code shows an example of how to change the payload in SafeMove.

```
MoveL p0, speed1, fine, tool1;
Set doGripper;
GripLoad load1;
MoveL p_start, speed1, fine, tool1;
MoveL p_end, speed2, zone1, tool1;
```

The same procedure can be used to release and reset the payload to zero load:

```
MoveL p0, speed1, fine, tool1;
Reset doGripper;
GripLoad load0;
MoveL p_start, speed1, fine, tool1;
```

Continues on next page

```
MoveL p_end, speed2, zone1, tool1;
```

The `loaddata` in `GripLoad` must correspond to the mass data of a configured payload in SafeMove. To disconnect a payload, the default load0 (0 kg) can always be used.

The position `p_start` is the starting point of the verification movement. The orientation of `p_start` must match to the **Payload Verification Reference Tool Orientation** configured in *Safe Payload Supervision*.

The position `p_end` is a position outside of the payload-change zone, having the same orientation as `p_start`. The movement from `p_start` to `p_end` must be a pure linear movement out of the zone in the positive z-direction, and the `speeddata speed2` must match the **Payload Verification Reference TCP Speed** configured in *Safe Payload Supervision*.

Recovering from failure

If the correct payload could not be verified, or if the verification movement was not executed correctly, the robot will stop and prevent movement in automatic mode. To recover from this state, a new payload selection and verification must be performed in manual reduced speed mode.

> **Note**
>
> In manual reduced speed mode, failure to verify the payload result in a warning event log without a stop. Operation in automatic mode will not be allowed after a failed verification of the payload in manual mode.

Updating the payload after a controller restart

After restarting the controller, the payload in SafeMove is considered undefined. A payload selection and verification must then be performed in manual reduced speed mode.

Selecting the verification movement

The start and end point of the verification movement should be chosen such that there is a significant torque on several joints during the movement, in particular the wrist joints. Otherwise, the joint torques may be too small for SafeMove to differentiate between the configured payloads, especially if their mass data is similar. It may be necessary to reorient the tool inside the payload-change zone after picking up a new payload, before executing the verification movement.

The start position and reference TCP speed of the verification movement must be chosen so that the TCP will move at a constant speed during a minimum of 100 ms prior to exiting the payload-change zone. To be able to perform payload verification in manual mode, it may be necessary to configure a reference TCP speed that is significantly slower than the maximum value 250 mm/s.

7.6 Using Safe Disable of Drive Unit

RAPID execution

When activating Safe Disable of Drive Unit (SDI) during RAPID execution, all axes in the corresponding RAPID task must be stationary in a programmed finepoint. If one or more axes are moving when SDI is activated, all mechanical units connected to the controller will stop with stop category 0 or stop category 1, depending on the configuration. If axes are stopped but not in a programmed finepoint there will be a system stop, RAPID execution will be blocked, and the program pointer must be moved.

While SDI is active, the corresponding mechanical unit remains active in RAPID. Although programmed positions for these units will be ignored, the mechanical unit can still be used, such as in coordinated motion. In this scenario, the current position of the mechanical unit will be used instead of the programmed position. If it is required that the programmed position of a mechanical unit is used, for example, for coordinated motion, it is recommended to monitor the SDI status signal in RAPID. This allows for appropriate actions to be taken in RAPID, preventing issues that may arise if a mechanical unit is stationary while other units expect it to move.

Deactivating of Safe Disable of Drive Unit can be done at any time, however, the corresponding mechanical unit will not start moving until the next move instruction after a finepoint. The recommendation is to only deactivate Safe Disable of Drive Unit when the corresponding RAPID task is in a finepoint.

Mechanical units with multiple axes

When mechanical units having multiple axes are included in the safety configuration, this is treated by SafeMove as multiple single axes, where each joint requires a separate configuration of Safe Disable of Drive Unit. Only the joints with Safe Disable of Drive Unit active are in a state that can be relied on for personal safety, even though the robot controller will prevent motion for the entire mechanical unit.

Positioners

Before configuring Safe Disable of Drive Unit for ABB positioners (IRP), it will in some cases be necessary to configure system parameters so that all positioner axes have a unique *Joint Id* number. See section *Configure system parameters on page 128*, for a description of how to change *Joint Id* number.

This page is intentionally left blank

8 Maintenance

8.1 Required maintenance activities

Test the motor contactors for a protective stop

Verify that a protective stop opens the motor contactors.

	Action	Note
1	In automatic mode, make SafeMove trigger a protective stop. For example, a stop issued from a safe fieldbus or a zone violation.	
2	Verify that the event log *90523 Safety Controller Protective Stop triggered* is shown on the FlexPendant.	
3	Verify that it is not possible to close the motor contactors, that is, go to state motors on.	

Internal functions are self tested

All internal functionality in the SafeMove safety controller is subject to self tests and requires no maintenance activities.

This page is intentionally left blank

9 RAPID components

About the RAPID components

This is an overview of all instructions, functions, and data types for functional safety and SafeMove.

For more information, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Instructions

Instructions	Description
SafetyControllerSyncRequest	`SafetyControllerSyncRequest` is used to initiate the hardware synchronization procedure.

Functions

Functions	Description
SafetyControllerGetOpModePinCode	`SafetyControllerGetOpModePinCode` is used to get the operating mode pin code for the keyless mode selector.
SafetyControllerGetChecksum	`SafetyControllerGetChecksum` is used to get the safety controller checksum for the user configuration file.
SafetyControllerGetGroupChecksum	`SafetyControllerGetGroupChecksum` is used to get the safety controller checksum for the protected group.
SafetyControllerGetSWVersion	`SafetyControllerGetSWVersion` is used to get the safety controller firmware version.
SafetyControllerGetUserChecksum	`SafetyControllerGetUserChecksum` is used to get the safety controller checksum for the area with protected parameters in the user configuration file.

Data types

There are no RAPID data types for functional safety and SafeMove.

This page is intentionally left blank

10 Reference information

10.1 Synchronization guidelines

10.1.1 Software synchronization guidelines

Uniquely defined position

The robot position for the synchronization must be chosen so that the position of the robot axes are unambiguously defined. One way to make sure the synchronization position is well-defined for all axes is to use the instruction `MoveAbsJ` to move to the sync position. See *Technical reference manual - RAPID Instructions, Functions and Data types*.

Note that the sync position should be allowed by all active functions. For example, the robot must be inside the allowed zones for all active Tool Position Supervision functions.

Note

On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Note

Software synchronization is always available even if hardware synchronization is configured.

Use easily verified sync position

Select a sync position where it is easy to verify the position of the robot axes. It is helpful to use a position where the TCP touches a spike or something where it is easy to see if the robot is in the correct position or not.

Performing a synchronization

WARNING

If the robot position is not visually verified, to make sure all robot axes are in correct position, the synchronization can jeopardize the safety.

Note

Synchronization can only be performed by a user with the grant **Software synchronization**. See *Set up safety user grants on page 130*.

Continues on next page

10.1.1 Software synchronization guidelines
Continued

	Action	
1	Move the robot to its sync position (for example with `MoveAbsJ`).	 **Tip** If **Move to Sync Position** is enabled on the tab **Synchronization**, press and hold the button to move to synchronization positions without having to jog the robot manually.
2	Visually verify that the robot is in its sync position (all axes must be in correct position).	If an axis is in wrong position, the revolution counters are most likely incorrect.
3	Go to the **Safety Controller** view.	
4	On the tab **Synchronization**, tap **Synchronize**.	

10.1.2 Hardware synchronization guidelines

Uniquely defined position

The robot position for the synchronization must be chosen so that the position of the robot axes are unambiguously defined. One way to make sure the synchronization position is well-defined for all axes is to use the instruction `MoveAbsJ` to move to the sync position. See *Technical reference manual - RAPID Instructions, Functions and Data types*.

Note that the sync position should be allowed by all active functions. For example, the robot must be inside the allowed zones for all active Tool Position Supervision functions.

Note

On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Note

Software synchronization is always available even if hardware synchronization is configured.

Small sync switch surface

For hardware synchronization, the sync switch surface that the robot touches must be small. The surface of the tool touching the sync switch must also be small. If any robot axis moves one motor revolution, the robot must be out of reach for the sync switch.

Always activate sync switch in the same way

For hardware synchronization, always use the same tool. The robot should always touch the sync switch with the same point on the tool.

Performing a synchronization

	Action	Note
1	Run a RAPID program with the instruction `SafetyControllerSyncRequest`.	This will make the controller responsive to the synchronization input signal for 30 seconds.
2	Move the robot to a position close to the sync switch.	If an axis is in wrong position, the revolution counters are most likely incorrect.
3	Slowly press the sync switch from the desired direction.	If the approach is too fast, the accuracy of the robot position may be too low.

Note

If the RAPID instruction `SafetyControllerSyncRequest` is executed and no sync signal is received within 30 seconds, the robot is stopped with an error message and the safety controller becomes unsynchronized.

10.2 Cyclic Brake Check guidelines

10.2.1 Introduction

About Cyclic Brake Check

Cyclic Brake Check is a function that verifies that the brakes work correctly.

If using Cyclic Brake Check on a SafeMove system, then Cyclic Brake Check should be configured in the safety configuration, see *Configuring Cyclic Brake Check on page 164*. The brake check must then be performed regularly (within the configured time interval).

Prerequisites for brake test

- The robot and all additional axes must be moved to a safe and relaxed position (away from people, equipment and not too much stretched) before performing Cyclic Brake Check. Normally the robot moves only a few centimeters during the brake tests.
- Move the robot to a stop point before performing Cyclic Brake Check.
- Cyclic Brake Check can only be performed at normal execution level (not from a trap routine, error handler, event routine or store path level).
- Brakes are tested in consecutive order and each test takes 10-15 seconds.
- Do not change the speed from the FlexPendant and do not use `VelSet`, `AccSet`, `SpeedRefresh`, or any other instruction that affects motion performance in `TRAPS` or event routines while `CyclicBrakeCheck` is active.

Note

The RAPID function `IsBrakeCheckActive` can be used to check if `CyclicBrakeCheck` is active.

Note

If Cyclic Brake Check is called together with the below combinations, then Cyclic Brake Check will be executed for all axes and the safety controller will not take any actions when the Cyclic Brake Check interval has passed or if the Cyclic Brake Check fails:

- SafeMove system with an empty safety configuration.
- SafeMove system with a safety configuration, but without a configured Cyclic Brake Check.

Performing Cyclic Brake Check

Start the brake check from RAPID

Call the procedure `CyclicBrakeCheck`.

Continues on next page

Start the brake check from the FlexPendant

	Action	Note/illustration
1	Move the robot to a safe brake check position.	
2	On the FlexPendant, open the **Safety Controller** view.	
3		

Cyclic Brake Check output

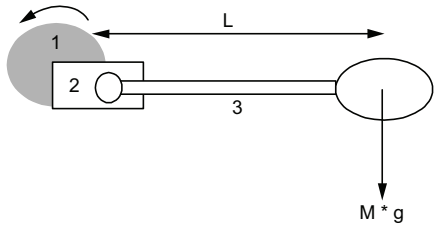
An error or warning message is logged for each axis with low brake torque. A status message is also logged for each complete brake cycle.

Cyclic Brake Check on additional axes

To be able to run CBC on additional axes, the parameter *Max Static Arm Torque* (in topic *Motion* and type *Brake*) needs to be calculated for the additional axis and entered into the configuration. CBC uses this value when testing the brake at error-level.

The parameter should be the maximum static torque that the brake needs to withstand when the additional axis is positioned in maximum gravity. The following formula should be used:

*Max Static Arm Torque = (M*L*g)/n*



xx1600001289

Length to mass (L)
Mass (M) * Gravity constant (g)
Transmission (n)
Motor (1)
Gearbox (2)
Arm (3)

To calculate the parameter for an axis that has no gravity, for example a track, the below formula may be used:

Max Static Arm Torque = Tbrake min/1.35

Tbrake min for ABB motor units can be found in the product specification for the specific motor unit, see *Product specification - Motor Units and Gear Units*.

Note

Note that the calculated value should be entered in [Nm] and calculated to the motor side.

Brake maintenance

Brake maintenance is a feature in the `CyclicBrakeCheck` functionality.

Continues on next page

10.2.1 Introduction
Continued

`CyclicBrakeCheck` automatically detects if maintenance of the mechanical brakes is needed and activates the *Brake maintenance* functionality during execution. *Brake maintenance* applies the brake and turns the motor shaft 1 radian five times, which gives a movement of the robot arm of less than 1 degree.

There are event logs that tell if *Brake maintenance* is needed, and if it has been run.

For more information see parameter *Brake Maintenance*, type *General Rapid*, topic *Controller*, in *Technical reference manual - System parameters*.

10.2.2 Cyclic Brake Check signal description

Introduction

Description of different signal states for Cyclic Brake Check (CBC).

Timing sequence for CBC signals

Description of which signals are set at different times during the Cyclic Brake Check.

Beginning of CBC

The following signals are set in the beginning of the CBC.

Signal	Set to
SC1CBCOK	0
SC1CBCACT	1
SC1CBCERR	0
SC1CBCWAR	0

End of CBC

The following signals are set in the end of the CBC.

Signal	CBC test OK Set to	CBC test WARNING Set to	CBC test ERROR Set to
SC1CBCOK	1	0	0
SC1CBCREQ	0	0	1
SC1CBCERR	0	0	1
SC1CBCWAR	0	1	0
SC1CBCACT	0	0	0
SC1CBCPREWARN	0	0	No change

Program pointer moved to Main after interrupted CBC

When the program pointer is moved to Main after an interrupted CBC the following signals are set.

Signal	Set to
SC1CBCOK	0
SC1CBCREQ	1
SC1CBCACT	0

New configuration downloaded, synchronization done, no CBC done

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	1
SC1CBCERR	0
SC1CBCWAR	0

Continues on next page

10.2.2 Cyclic Brake Check signal description
Continued

Signal	Signal state
SC1CBCACT	0
SC1CBCPREWARN	0

Max. allowed speed: 250 mm/s

During the first CBC test

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	No change
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	1
SC1CBCPREWARN	No change

CBC done with the result OK

Signal	Signal state
SC1CBCOK	1
SC1CBCREQ	0
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	0
SC1CBCPREWARN	0

Max. allowed speed: Max speed

CBC done with the result WARNING

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	0
SC1CBCERR	0
SC1CBCWAR	1
SC1CBCACT	0
SC1CBCPREWARN	0

Max. allowed speed: Max speed

CBC done with the result ERROR

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	1
SC1CBCERR	1

Continues on next page

Signal	Signal state
SC1CBCWAR	0
SC1CBCACT	0
SC1CBCPREWARN	No change

Max. allowed speed: 250 mm/s

Prewarning time has expired

Signal	Signal state
SC1CBCOK	No change
SC1CBCREQ	No change
SC1CBCERR	No change
SC1CBCWAR	No change
SC1CBCACT	No change
SC1CBCPREWARN	1

Max. allowed speed: Max speed

Max CBC test interval has elapsed

Signal	Signal state
SC1CBCOK	No change
SC1CBCREQ	1
SC1CBCERR	No change
SC1CBCWAR	No change
SC1CBCACT	No change
SC1CBCPREWARN	1

Max. allowed speed: 250 mm/s

Interrupted CBC test, program pointer still in CBC routine

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	No change
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	1
SC1CBCPREWARN	No change

Interrupted CBC test, program pointer moved from CBC routine

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	1

Continues on next page

10.2.2 Cyclic Brake Check signal description
Continued

Signal	Signal state
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	0
SC1CBCPREWARN	No change

10.3 Guidelines for transient and quasi-static contact, CRB 15000

About *Human Contact Supervision*

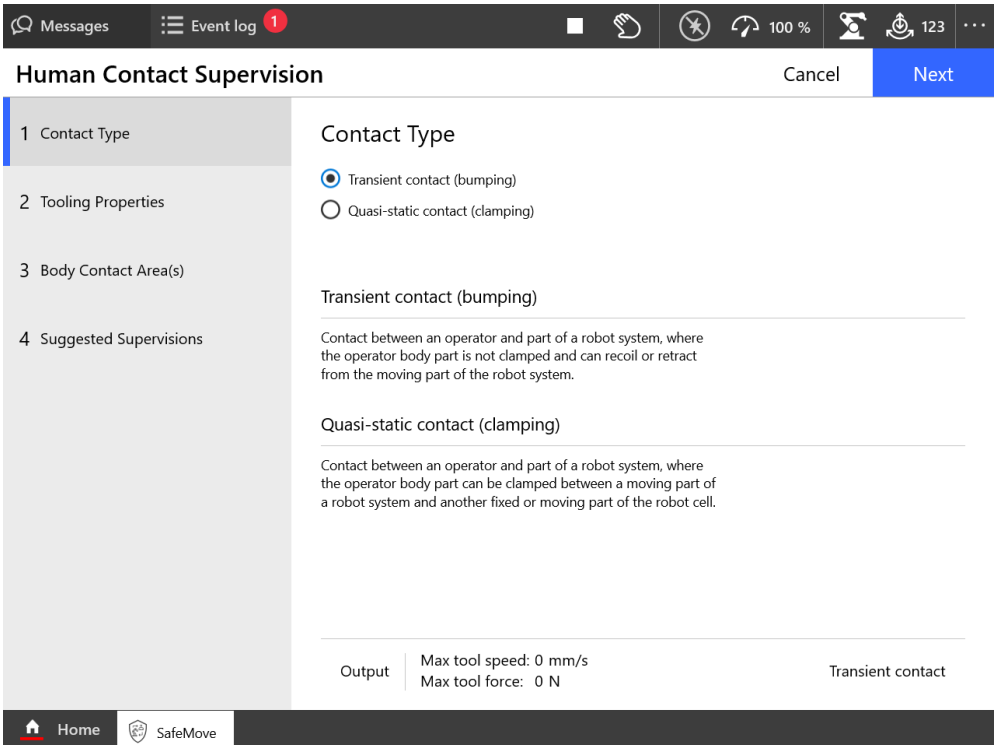
The CRB 15000 robot is designed with collaborative applications in focus, where occasional contact between the human and the robot are foreseen. This is according to ISO/TS 15066.

The supporting function *Human Contact Supervision* in SafeMove can be used to calculate maximum allowed tool force and tool speed.

Transient contact and quasi-static contact

Transient contact is contact between an operator and part of a robot system, where the operator body part is not clamped and can recoil or retract from the moving part of the robot system.

Quasi-static contact is contact between an operator and part of a robot system, where the operator body part can be clamped between a moving part of a robot system and another fixed or moving part of the robot cell.



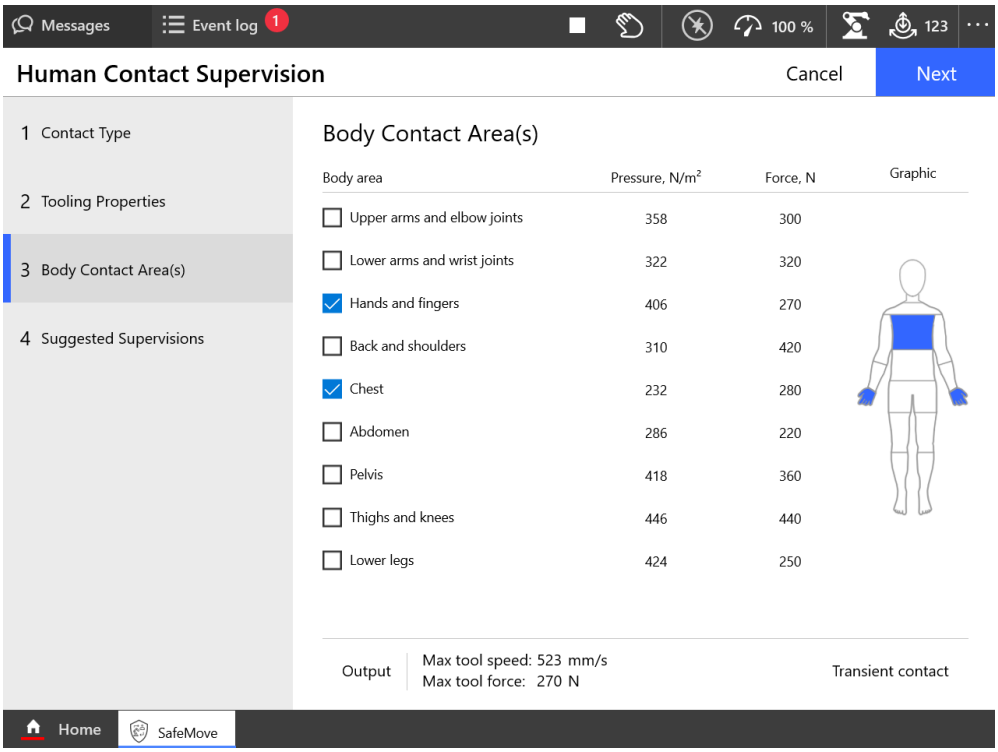
xx2100000711

Continues on next page

Body model

As defined in ISO/TS 15066, the body model is a representation of the human body consisting of individual body segments characterized by biomechanical properties.

The segments of the body model has different sensitivity. In general, the application should be designed so that the human head and neck is never exposed to hazards.



xx2100000709

Recommendations from ISO/TS 15066

A key process in the design of the collaborative robot system and the associated cell layout is the elimination of hazards and reduction of risks, and can include or influence the design of the working environment. The following factors shall be taken into consideration:

1. The established limits (three dimensional) of the collaborative workspace.
2. Collaborative workspace, access, and clearance.
3. Ergonomics and human interface with equipment.
4. Use limits.
5. Transitions.

For more information, see ISO/TS 15066.

Continues on next page

Conclusion

The values calculated in the **Human Contact Supervision** function are conservative. However, if the risk assessment for the final application shows that these values can be changed to a higher value, for example, by using padding on the arm, then the values can be changed in the settings for the tool force supervision and tool speed supervision.

CAUTION

The functionality is based on the recommendations in ISO/TS 15066. A risk assessment of the final application must always be done, where the calculations are reviewed and verified by test.

10.4 Servo Delay Factor and Servo Lag

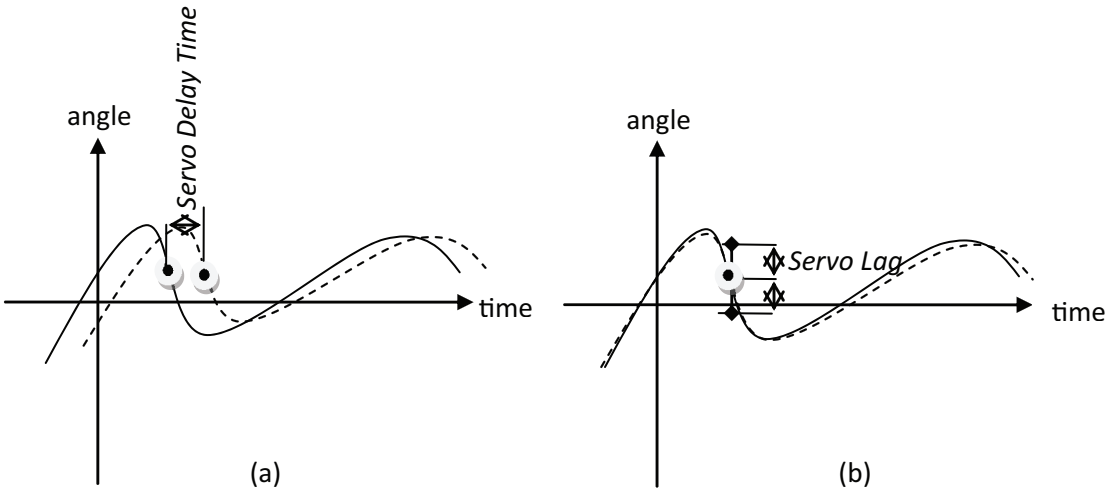
System parameters Servo Delay Factor and Servo Lag

To explain what is affected by the parameters *Servo Delay Factor* and *Servo Lag* in the SafeMove configuration (see *Configure additional axes on page 136*) it is best to consider an example.

In graph (a) in the figure below, the *Servo Delay Time* is illustrated using a constructed example where a reference and a corresponding measured angular motor position is shown. The corresponding *Servo Delay Factor* can be computed using:

Servo Delay Factor = Servo Delay Time / 4

In graph (b) in the figure below, the *Servo Lag* parameter is illustrated when the measured signal is shifted with the *Servo Delay Time*. The measured position should now be within a distance of +/- *Servo Lag* from the reference at every time instance. If the measured position is outside the specified region the SafeMove position supervision is triggered.



en1100000626

Figure 10.1: Illustration of Servo Delay Time (a) and Servo Lag (b). Solid line is motor angular position reference (test signal 17) and dashed is the corresponding measured motor angular position (test signal 18). In (b) the measured motor angular position signal is shifted Servo Delay Time in order to illustrate that it is the time shifted signal (samples) that should lie within +/- Servo Lag radians from the reference.

Note

The signals in the figure are only for illustrative purpose. The *Servo Delay Time* in a real system is small, typically in the range 8-12 ms, which means a *Servo Delay Factor* in the range 2-3. If the *Servo Delay Factor* is incorrect the *Servo Lag* has to compensate this since the difference between reference and measured motor angular position will be large when the axis accelerates, runs at high speed, or decelerates.

Continues on next page

Tuning the parameters

If the SafeMove position supervision is triggered for the axis, take the following actions to verify and possibly tune the parameters:

1. Check that the load of the robot and additional load on the axis is included in the configuration.
2. Make sure no mechanical problem is present, for example giving abnormal friction.
3. If possible, use TuneMaster and log the test signals 17 (motor angular position reference) and 18 (measured motor angular position) for the axis. If TuneMaster is not available, the *Servo Lag* can be increased (according to step *3b*).
   a. Move the axis and measure the *Servo Delay Time*. Compute the corresponding *Servo Delay Factor*:

      Servo Delay Factor = Servo Delay Time / 4

      where it is assumed that *Servo Delay Time* is measured in milliseconds. Use the value in the SafeMove configuration.
   b. If the supervision is still triggered, increase the *Servo Lag* until supervision is not triggered.

Tuning a non ABB additional axis

If attempting to use a non ABB additional axis, start with doing a tuning of the axis. To be able to reduce servo lag below the configured maximum allowed value during the tuning, make sure to set the parameter *FFW Mode* (feed forward mode) to *Spd* (speed) or *Trq* (torque). For more information about tuning an additional axis see *Application manual - Additional axes*.

WARNING

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

10.5 SafeMove geometry configuration file

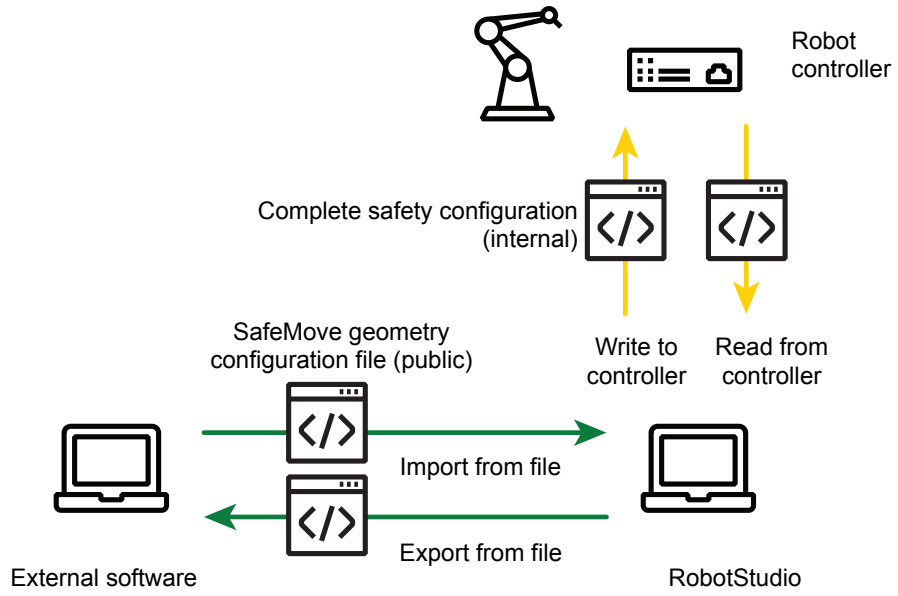
10.5.1 Introduction

Overview

This section describes the SafeMove geometry configuration file that can be used to import and export safety zones and related geometry to and from Visual SafeMove.

The intended use of the file is to provide means to define and analyze the geometry related part of a safety configuration in a tool external to RobotStudio and Visual SafeMove.

The illustration below shows how the safety related geometry information can be read from a file and written to a robot controller via Visual SafeMove. Vice versa, the safety-related geometry information can be read from the controller by Visual SafeMove and exported to a geometry configuration file.



xx1700000702

Unsupported functions

The following functions must be configured using Visual SafeMove in RobotStudio and cannot be configured through the geometry configuration file.

- Contact Application Tolerance
- Stand Still Supervision
- Axis Speed Supervision
- Axis Position Supervision
- Tool Orientation Supervision
- External Power Supply

Continues on next page

- Cyclic Brake Check
- Stop Configuration

10.5.2 Use cases

Use case 1: Import of nominal safety related geometry information

If the planning and geometry layout is created in a tool other than RobotStudio and Visual SafeMove, the file can be used to transfer the safety-related geometry information to RobotStudio and Visual SafeMove.

The safety configuration must be completed in Visual SafeMove by adding non-geometry related safety configuration, for example brake check and activation signals, before writing it to a real controller on the shop floor.

Procedure:

1. Create the cell layout in the external planning tool.
2. Create the safety geometry in the external planning tool, for example safety zones, tool encapsulation, and speed supervision points.
3. Export the safety related geometries from the external planning tool to a SafeMove geometry configuration file.
4. Import the SafeMove geometry configuration file into Visual SafeMove in RobotStudio.
5. Complete the safety configuration by adding remaining non-geometry related safety configurations, for example brake check and activation signals.
6. Load the safety configuration to the safety controller.

Use case 2: Import of calibrated safety related geometry information

This use case is very similar to use case 1, the difference being that the safety-related geometries are adjusted with respect to measurement data of the robot cell.

Steps 3A to 3C are added to the procedure in use case 1:

A. Import the SafeMove geometry configuration file into a second tool that can adjust zones based on measurement data from the real robot cell.
B. Adjust the zones relative to the offsets of the corresponding real objects in the robot cell, for example robot base frame, fixtures, tip dressers, material supply units, etc.
C. Export the adjusted zones to a new SafeMove geometry configuration file.

Use case 3: Export of real safety related geometry information

In this use case, the actual zones and related geometry information of the real controller are exported to a SafeMove geometry configuration file using Visual SafeMove in RobotStudio. The purpose may be to update the nominal robot cell in the external planning tool.

Procedure:

1. Connect RobotStudio to the robot controller.
2. Use Visual SafeMove to read the safety configuration from the robot controller.
3. Export the configuration to a SafeMove geometry configuration file.

Continues on next page

4 Import the SafeMove geometry configuration file into the external planning tool so that the nominal safety zones are updated with the values from the real robot.

10.5.3 Explanation of the configuration file

10.5.3.1 Introduction

Overview

The purpose of the SafeMove geometry configuration file is to support data related to the supervision functions:

- Tool Position Supervision
- Tool Speed Supervision
- Tool Orientation Supervision

The file can contain the tool encapsulation, speed supervision points, and the zone definitions. This data is necessary, but not sufficient, to specify the functions. To completely specify the functions then also activation signals, speed limits, and whether a zone is an inside or an outside zone must be specified and added in Visual SafeMove.

Summary

XML-tag	Description
`DriveModuleConfiguration`	1-2 per controller
`Robot`	0-1 per `DriveModuleConfiguration`
`Baseframe`	0-1 per `Robot`
`ElbowOffset`	0-1 per `Robot`
`UpperArmGeometry`	0-2 per `Robot`
`ExternalAxis`	0-3 per `DriveModuleConfiguration`
`BaseFrame`	1 per `ExternalAxis`
`Tool`	0-16 per `DriveModuleConfiguration`
`TCP`	1 per `Tool`
`ToolOrientation`	1 per `Tool`
`ToolGeometry`	0-4 per `Tool`
`SpeedSupervisionPoint`	0-8 points per `Tool`
`Name`	1 per `Tool`
`SafeZone`	0-16 per `DriveModuleConfiguration`
`Point`	4-24 per `SafeZone`
`Name`	1 per `SafeZone`

XML schema

There is an XML schema available that defines the exact syntax and content of the interface file. This is stored in an xsd-file and can be used to validate the content of the xml-file.

The template file *SimplifiedSafetyControllerSchema.xsd* can be obtained from the PC or the robot controller.

- **In the RobotWare installation folder in RobotStudio**: *...\RobotPackages\RobotWare_RPK_<version>\utility\SafeMove2*

Continues on next page

- **On the robot controller:**
  \products\<RobotControl_xx.xx.xxxx>\utility\SafeMove2

> **Note**
>
> Navigate to the RobotWare installation folder from the RobotStudio **Add-Ins** tab, by right-clicking on the installed RobotWare version in the **Add-Ins** browser and selecting **Open Package Folder**.

10.5.3.2 Drive module configuration

Overview

The `DriveModuleConfiguration` tag contains information about the motion task.

There is a drive module configuration for each motion task of a controller system monitored by SafeMove.

ID

The `driveModuleId` tag contains the ID of the motion task.

Each motion task must have a unique ID between 1 and 4. The ID is not visible to the user.

10.5.3.3 Robot data

Overview

The `Robot` tag contains information about the robot.

Name

The `name` tag is `ROB_1` for a controller with only one manipulator. For a MultiMove system the subsequent robot is called `ROB_2`.

Robot moved by

For track mounted robots, the `movedBy` tag specifies the name of the external axis that moves the robot.

Base frame

The `Baseframe` tag specifies the robot mounting position relative to the controller world coordinate system.

When importing a SafeMove geometry configuration file into Visual SafeMove, the base frame values contained in the file will be replaced by the base frame of the connected controller when creating the complete safety configuration. When exporting the file, the `Baseframe` tag will contain the values of the actual base frame of the connected robot.

Elbow offset

The `ElbowOffset` tag defines the point on the robot elbow that is speed monitored. It should be the top-most point of the upper arm including any cable packages or additional equipment.

Upper arm geometry

The `UpperArmGeometry` tag defines the encapsulation of the upper arm.

The upper arm may optionally be encapsulated by up to two shapes, so-called sphere-swept volumes, and used in combination with a tool encapsulation for the function Tool Position Supervision. The shapes should include the upper arm of the robot including any cable packages or additional arm-mounted equipment. The

Continues on next page

10.5.3.3 Robot data
Continued

reference point of the upper arm geometry is the same as the elbow offset of the motion configuration file of the controller, *moc.cfg*.

xx1700000703

10.5.3.4 External axis

Overview

The `ExternalAxis` tag contains information about external axes.

Name

The `name` tag contains the name of the external axis. For example `TRACK_1`.

Axis type

The `axisType` tag must be `Track` for a track motion.

Base frame

The `Baseframe` tag specifies the position and orientation of the track motion.

10.5.3.5 Tool

Overview

The `Tool` tag contains information about the tools.

A tool consists of a set of shapes and a set of points. The shapes are used by the function Tool Position Supervision and the points by the function Tool Speed Supervision.

Up to 16 tools can be defined.

ID

The `id` tag contains the ID of the tool.

Each tool must have a unique ID between 1 and 16. The ID is not visible to the user.

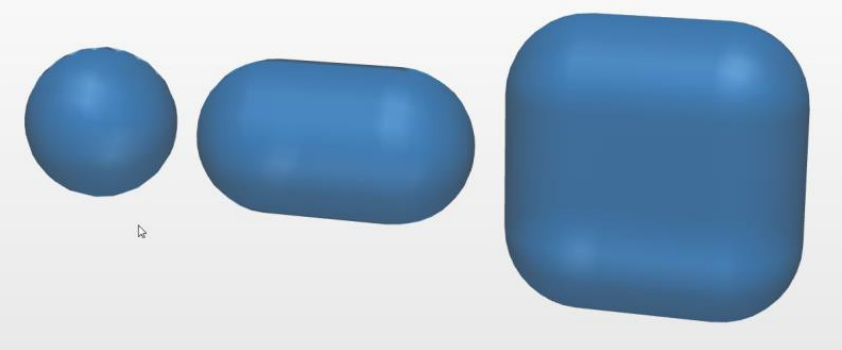
Name

The `name` tag contains the name of the tool.

Tool geometry

The `ToolGeometry` tag defines the number of shapes.

A tool consists of up to four shapes, so called sphere-swept volumes. A shape can be a sphere, a capsule, or a rounded box (lozenge).



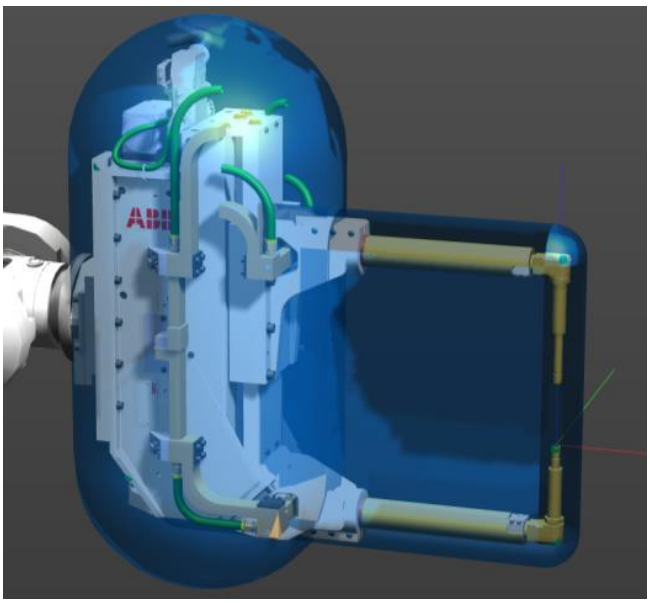
xx1700000705

The shapes are generated by a sphere with a certain radius in combination with a point, a line, or a plane, respectively.

The sphere is trivially generated by placing the generating sphere on a point. A capsule is generated by letting the center of the generating sphere travel along a line. A rounded box is generated by letting the center point of the generating sphere travel along a plane.

Continues on next page

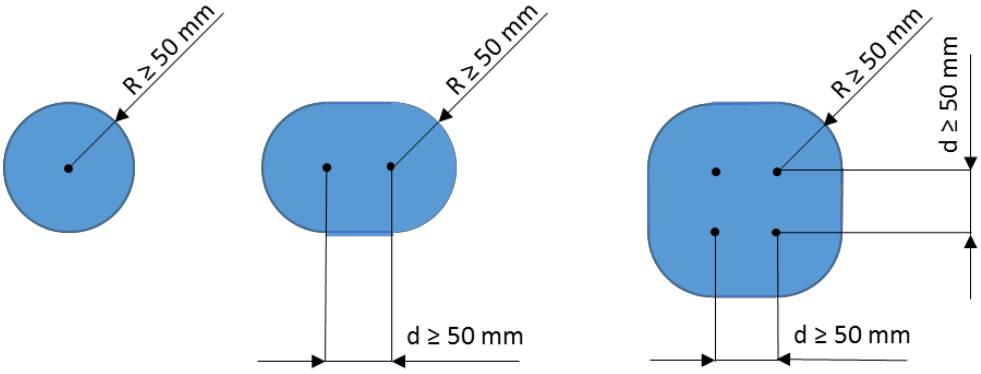
The below figure shows an example of a tool encapsulation with an ABB FlexGun covered by a capsule and a rounded box.



xx1700000706

Limitations of the tool shapes

The radius of any tool shape must be at least 50 mm. The distance between the two end points of the generating line of a capsule must be at least 50 mm. The length and width of the generating plane of a rounded box must be at least 50 mm.



xx1700000707

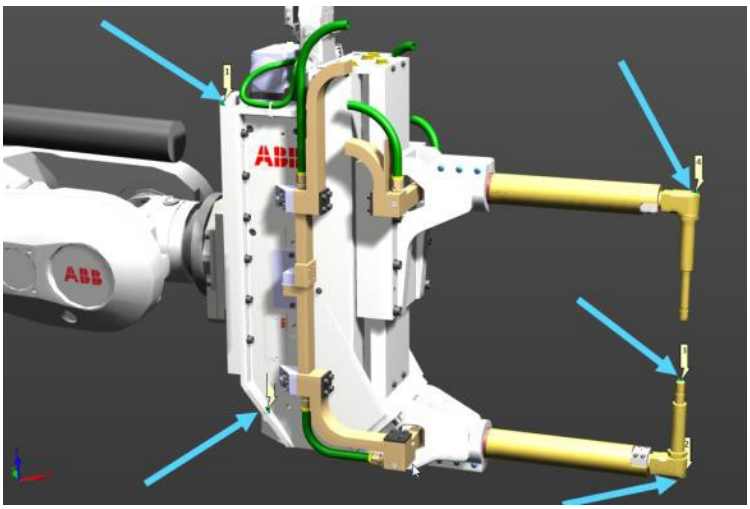
Continues on next page

10.5.3.5 Tool
Continued

Speed supervision points

The `SpeedSupervisionPoint` tag defines the number of speed supervision points.

Up to eight points can be defined on the robot tool. The points are defined in relation to the center point of the flange (*tool0*).



xx1700000708

TCP and tool orientation

The `TCP` and `ToolOrientation` tags defines the TCP and orientation of the current `tooldata` and must be specified.

10.5.3.6 Safe zones

Overview

The `SafeZone` tag contains information about the safe zones.

The functions Tool Position Supervision, Tool Orientation Supervision, and Tool Speed Supervision are based on zones. Up to 16 safe zones can be defined.

ID

The `id` tag contains the ID of the zone.

Each zone must have a unique ID between 1 and 16. The ID is not visible to the user.

Name

The `name` tag contains the name of the zone.

The name will be visible to the user and presented in the controller event log in case of a safety violation.

Top and bottom

The `top` and `bottom` tags defines the top and bottom of the zone and must be specified.

Points

The `Point` tag defines the points of the zone.

A zone is defined by minimum 4 up to 24 points in the X-Y plane of the controller world coordinate system in addition to the bottom and top in the Z direction.

Speed limit priority

The `speedLimitPriority` tag defines the speed limit priority of the zone.

Zones can have three different speed limit priorities: `BASE`, `NORMAL`, and `OVERRIDE`.

Zones can be nested, which means that one zone can overlap or be contained in another zone. Nested zones with different speed priorities can be used to define tighter robot cells without sacrificing cycle time. In particular, the stopping distances can be kept low by defining a large `BASE` zone with low speed limits and interior zones with increasing priorities and speed limits.

The interior, high speed zones, ensure high robot productivity and low cycle times.

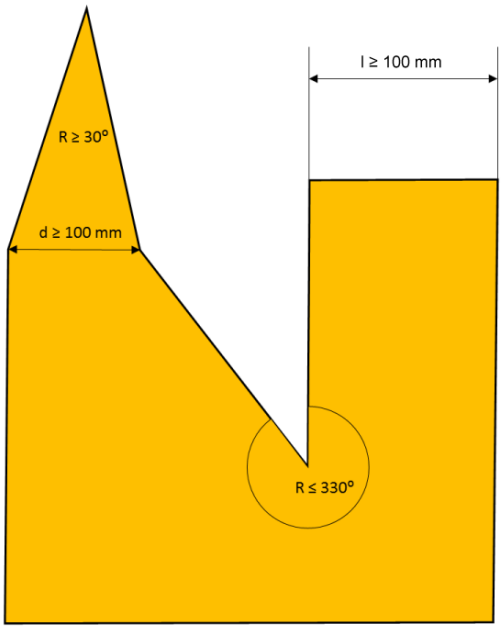
Continues on next page

10.5.3.6 Safe zones
Continued

Limitations

Any of the coordinates must not exceed ± 1000 meters. The edges of a zone must be at least 0.1 m and they must not intersect. The zone must be wider than 0.1 m. A corner of a zone must have an angle between 30 and 330 degrees.

The below illustration shows the constraints that must be met by any zone definition.



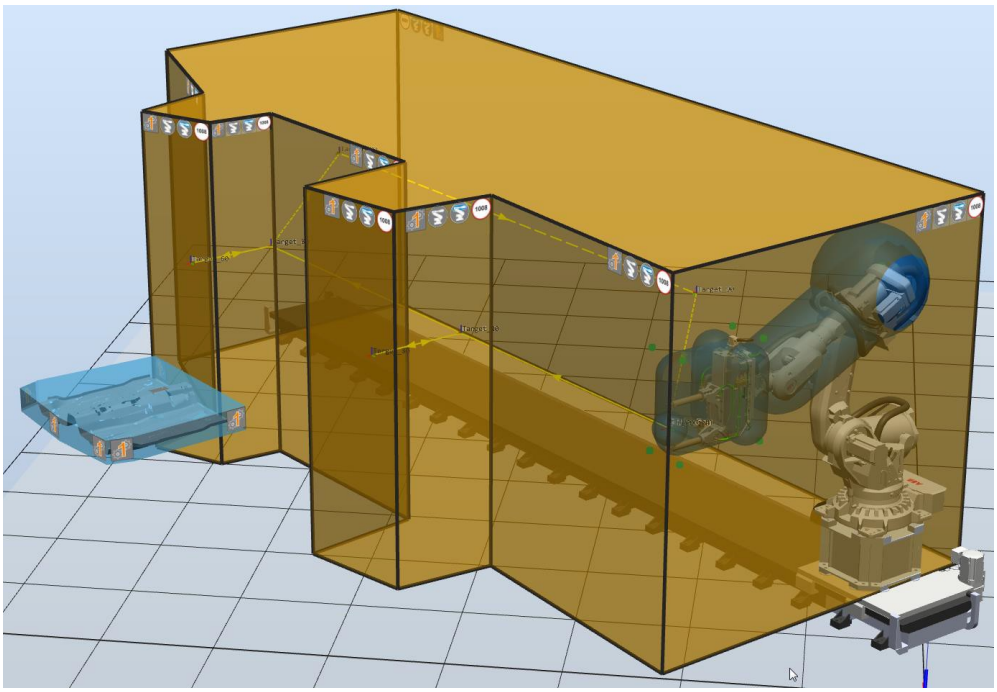
xx1700000704

10.5.4 Example

Overview

This section contains an example of an XML-file for a MultiMove systems with two robots. The first robot, ROB_1, is mounted on a track motion, TRACK_1, whereas the second, ROB_2 is mounted on the floor.

The example file defines the zones in the below illustration.



xx1700000709

Example file

```
<SimplifiedSafetyConfiguration
     xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="urn:abb-robotics-simplified-safety-controller-configuration">
<DriveModuleConfiguration DriveModuleId="1"
     maxSpeedManualMode="0.25">
  <Robot name="ROB_1" startSpeedOffset="0.1" movedBy="TRACK_1">
    <Baseframe>
      <Translation x="0" y="0" z="0" />
      <Quaternion q1="0.707107" q2="0" q3="0" q4="0.707107" />
    </Baseframe>
    <ElbowOffset x="-0.252" y="0" z="0.342" />
    <UpperArmGeometry xs:type="Capsule" name="UpperArm"
         radius="0.357727">
      <Start x="-0.268245" y="-0.017373" z="0.190867" />
      <End x="1.03026" y="-0.096637" z="0.167713" />
    </UpperArmGeometry>
    <UpperArmGeometry xs:type="Sphere" name="HosePackage"
         radius="0.5">
      <Center x="-0.084216" y="-0" z="0.305324" />
```

Continues on next page

10.5.4 Example
Continued

```
  </UpperArmGeometry>
</Robot>
<ExternalAxis name="TRACK_1" axisType="Track">
  <Baseframe>
    <Translation x="0" y="0" z="0" />
    <Quaternion q1="1" q2="0" q3="0" q4="0" />
  </Baseframe>
</ExternalAxis>
<Tool id="1" name="tFlexGun">
  <TCP x="0" y="0.1485" z="0.8015" />
  <ToolOrientation q1="0.5" q2="0.5" q3="-0.5" q4="0.5" />
  <ToolGeometry xs:type="Lozenge" name="GunBody" radius="0.183332"
       width="0.542759" height="0.5">
    <Pose>
      <Translation x="0.063143" y="0" z="0.377878" />
      <Quaternion q1="0.707107" q2="0" q3="-0.707107" q4="0" />
    </Pose>
  </ToolGeometry>
  <ToolGeometry xs:type="Capsule" name="BodyRight"
       radius="0.181797">
    <Start x="0.2" y="0.2" z="0.15" />
    <End x="0.2" y="-0.3" z="0.15" />
  </ToolGeometry>
  <ToolGeometry xs:type="Sphere" name="Electrode"
       radius="0.144212">
    <Center x="0" y="0.162418" z="0.78145" />
  </ToolGeometry>
  <ToolGeometry xs:type="Capsule" name="BodyLeft"
       radius="0.181797">
    <Start x="-0.2" y="0.2" z="0.15" />
    <End x="-0.2" y="-0.3" z="0.15" />
  </ToolGeometry>
  <SpeedSupervisionPoint x="-0.212" y="-0.499717" z="-0.005" />
  <SpeedSupervisionPoint x="-0.212" y="0.3855" z="-0.005" />
  <SpeedSupervisionPoint x="-0.212" y="0.3855" z="0.8515" />
  <SpeedSupervisionPoint x="-0.212" y="-0.499717" z="0.8515" />
  <SpeedSupervisionPoint x="0.15799" y="-0.499717" z="-0.005" />
  <SpeedSupervisionPoint x="0.15799" y="0.3855" z="-0.005" />
  <SpeedSupervisionPoint x="0.15799" y="0.3855" z="0.8515" />
  <SpeedSupervisionPoint x="0.15799" y="-0.499717" z="0.8515" />
</Tool>
<SafeZone id="1" name="RobotZone1" top="2.889806" bottom="-0.5"
     speedLimitPriority="NORMAL">
  <Point x="3.181233" y="3.743311" />
  <Point x="2.13778" y="3.552878" />
  <Point x="2.042189" y="2.624328" />
  <Point x="-0.430474" y="2.750736" />
  <Point x="-0.359415" y="-0.735027" />
  <Point x="8.319961" y="-0.790136" />
  <Point x="8.181677" y="2.193857" />
  <Point x="6.422987" y="2.805629" />
```

Continues on next page

```
    <Point x="6.193726" y="3.606499" />
    <Point x="5.458064" y="3.311171" />
    <Point x="5.399194" y="2.682248" />
    <Point x="3.298018" y="2.624001" />
  </SafeZone>
  <SafeZone id="2" name="Safe_Zone" top="0.931383" bottom="0.710112"
       speedLimitPriority="NORMAL">
    <Point x="3.41184" y="5.366856" />
    <Point x="3.462365" y="4.22674" />
    <Point x="5.035686" y="4.224637" />
    <Point x="5.101747" y="5.381664" />
    <Point x="5.023219" y="5.600441" />
    <Point x="4.195253" y="5.658945" />
    <Point x="3.479269" y="5.592942" />
  </SafeZone>
</DriveModuleConfiguration>
</SimplifiedSafetyConfiguration>
```

10.6 Guidelines for the option *Extended working range*

Introduction

The option *Extended work range* enables an extension of the working range for axis 1, through a software configuration. With this option installed, the working range can exceed the range limited by the mechanical stop on axis 1. The working range shall be limited through the option SafeMove.

A risk analysis must be done to ensure that no risks remain when using option *Extended work range*, to limit the working range, and before removing the mechanical stops.

For information about the option SafeMove, see *Application manual - Functional safety and SafeMove*.

If the mechanical stop is removed, then the manipulator should have a marking for this, for example, a label. If the robot is delivered with the option *Extended work range*, then such a label is included on delivery.

The working limit of axis 1 must be restricted to ensure that the stopping position is not outside the maximum allowed axis angle for the selected manipulator, for example, 220°. This can be done using a physical limitation or using SafeMove.

Example solution with SafeMove

To ensure that the robot can be stopped within the maximum allowed range when there are no mechanical stops on axis 1, the speed must be reduced when the robot is approaching the limit. Use this procedure to set up the speed supervision in SafeMove.

1. Create a safe range that is enabled only for axis 1. Invert the range.
2. Add an *Axis Speed Supervision* to the safe range.
3. Set the activation to permanently activated. Select a stop category.
4. Set the maximum speed for axis 1, that is allowed when axis 1 is within the defined range.
5. Increase the maximum speed for all other axes to maximum value.

With this example configuration, the robot will be stopped when axis 1 passes into the defined safe range (<170° >170°) if axis 1 speed is higher than 10°/s.

Note

The limits for safe range and *Axis Speed Supervision* must be tested case by case using the highest possible speed, highest used payload, and longest possible axis reach. The speed and axis range for the supervision must be defined for ensuring that the robot, in all cases can stop inside the maximum allowed working range of axis 1. The working range depends on the robot, for example on an IRB 7600, the maximum extended working range is +/-220°.

Index

ABB AB
Robotics & Discrete Automation
S-721 68 VÄSTERÅS, Sweden
Telephone +46 10-732 50 00

ABB AS
Robotics & Discrete Automation
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.
Robotics & Discrete Automation
No. 4528 Kangxin Highway
PuDong New District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

ABB Inc.
Robotics & Discrete Automation
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

abb.com/robotics

3HAC066559-001, Rev R, en