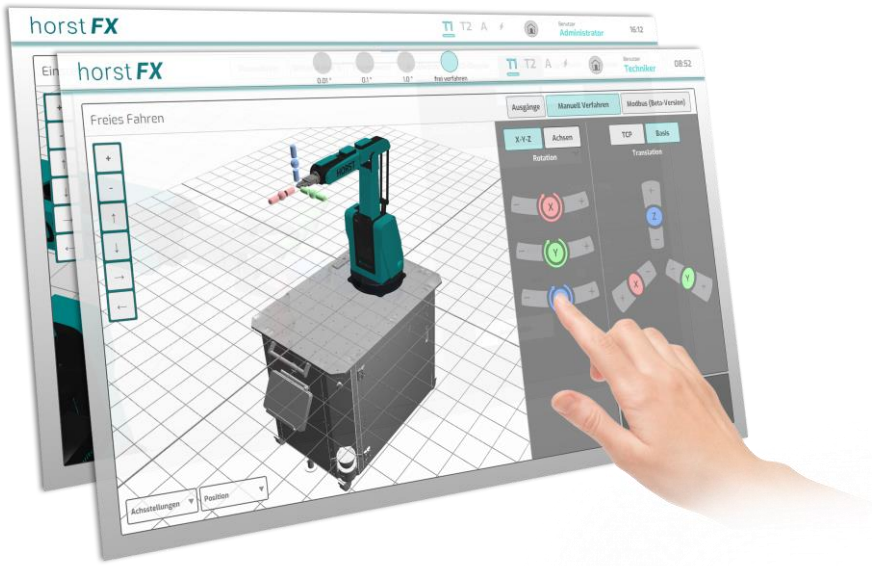
horstFX software



User manual

For installation, operating, and maintenance personnel

Always keep with the product!

Version 2.5 / 24.11.2023 / horstFX 2023.11

A current version of this user manual is always available to view at horstcosmos.com.

Copyright

Original language of the documentation: German

fruitcore robotics GmbH
Macairestr. 3
78467 Konstanz, Germany

Telephone: (+)49 (0)7531 / 945 99-20
Email: info@fruitcore.de
Website: www.fruitcore-robotics.com

Date of issue: November 2023
Subject to design and machine alterations

Foreword

This user manual is intended solely for the efficient and safe use of the horstFX software and does not replace the *assembly instructions* of the respective robot system. Before installation of the robot, it is essential to read, understand, and observe the *assembly instructions*. Proper assembly in compliance with the *assembly instructions* and the applicable standards is also essential.

DANGER!

The robot system may only be operated and used following correct installation in accordance with the enclosed assembly instructions and in compliance with the applicable standards.

- This user manual is to be seen as a supplement and describes exclusively the control of the software without consideration of the installation situation.

Read the assembly instructions of the respective robot system carefully and in full before commissioning the robot system.

Handle the user manual with care. An illegible or missing user manual must be replaced immediately.

fruitcore robotics

Contents

1 Introduction

The horstFX software enables the programming and operation of the robot by means of a touch-sensitive user interface/touchscreen display on the operating panel (horstPANEL). Through the available interfaces on the switch cabinet (horstCONTROL), communication with and control of other machines and external sensors is also possible.

By switching on the main switch on horstCONTROL, the software horstFX launches automatically and is displayed on horstPANEL.

DANGER!
The robot system may only be operated and used following correct installation in accordance with the enclosed assembly instructions for the respective robot system and in compliance with the applicable standards.

- ▶ This user manual is to be seen as a supplement and describes exclusively the control of the software without consideration of the installation situation.

1.1 Principle

This user manual is intended solely for the efficient and safe use of the horstFX software and does not replace the *assembly instructions* of the respective robot system. Before installation of the robot, it is essential to read, understand, and observe the *assembly instructions*. Proper assembly in compliance with the *assembly instructions* and the applicable standards is also essential.

The *assembly instructions* (not this user manual) contain important information on how to operate the robot system safely, properly, and cost-effectively. Observing the *assembly instructions* helps to avoid hazards, reduce repair costs and downtime, and increase the reliability and service life of the robot system.

You should also read this user manual carefully and in full before operating the robot system.

Handle the user manual with care. An illegible or missing user manual must be replaced immediately.

1.2 General information

This user manual provides you with comprehensive instructions for operating the robot system. This manual supplements the *assembly instructions*, in which you will find a detailed description of the robot system, guidelines for transport and installation, fault rectification tips, and information on maintenance.

The robot system delivered may include options that deviate from the text and images shown in this manual. This is due to the individual adaptation and further development of the robot system on the basis of the requests and orders of the individual customers. These deviations do not constitute a basis for claims of any kind.

The robot system must be used only for the permissible purposes listed in the *assembly instructions*. The manufacturer assumes no liability for the improper or unauthorized use of the robot system, operating errors, user errors, or improper or insufficient maintenance.

The *assembly instructions* contain instructions and related information for the safe use of the robot system. The instructions specified here must be followed at all times.

1.3 Signs, symbols, and abbreviations

The following symbols are used in this user manual:

Lists

– Simple lists are marked with "–".

Instructions for action

All instructions for action for a procedure are listed in chronological order.

▶ Instructions are marked with "▶".

⇒ Intermediate results and end results of the action are marked with "⇒".

Note

This symbol stands for information that allows a more effective and economically efficient use of the robot system.

1.4 Marking of the safety and warning signs

The following safety signs mark all actions that present a danger to life and limb of the operator or others around the operator.

Make sure to observe these signs and exercise particular caution in these cases. Also pass on the safety signs to other users.

DANGER!

The sign with the addition DANGER refers to an immediate danger.

The danger will lead to serious injury or death of persons.

▶ The description of the danger is followed by instructions for action that serve to avoid or remove the danger.

WARNING!

The sign with the addition WARNING refers to possible danger.

The danger can lead to serious injury or death of a person.

▶ The description of the danger is followed by instructions for action that serve to avoid or remove the danger.

CAUTION!

The sign with the addition CAUTION refers to a potentially hazardous situation.

The danger can lead to injury of persons.

▶ The description of the danger is followed by instructions for action that serve to avoid or remove the danger.

The safety signs are often used in combination with a pictogram in the text to clarify the source of the danger.

ELECTRICAL VOLTAGE!

This sign is a warning for electricity.

It is posted for all work and operating procedures that are to be observed precisely in order to prevent danger to persons and the system by electricity.

ATTENTION! Danger of damage to robot or property.

This sign indicates information which, if disregarded, presents a danger to the robot system, individual modules, or the operating environment. There is no risk of injury.

Wear protective clothing.

Wear your personal protective equipment:

safety shoes, protective helmet, safety goggles, and work gloves.

Danger of environmental damage.

This sign indicates information which, if disregarded, presents a danger to the environment. There is no risk of injury.

1.4.1 Abbreviations

Fig.	Figure
OI	Operating instructions
I/O	Input and output
HORST	Highly Optimized Robotic Systems Technology
HTTP	Hypertext Transfer Protocol
TCP	Tool Center Point
XML-RPC	Extensible Markup Language Remote Procedure Call

1.5 Robot model illustrations

Some of the illustrations in this user manual show a 3D model of the robot. These are general-purpose illustrations that are valid for all robot types. Unless otherwise stated, these illustrations refer only to the horstFX functions shown therein.

1.6 Changes to the file system of the operating system

In some cases it is necessary to make changes directly to the file system of the operating system, e.g. when importing files needed for horstFX (see section 4.5.1.2 or 4.5.2.1).

ATTENTION!

All other changes that have not been agreed upon with fruitcore robotics GmbH, especially the deletion/moving of folders/files, may result in horstFX no longer being able to be (properly) operated.

It should be noted that fruitcore robotics GmbH does not assume any liability for damages resulting from this.

2 Switching on the System

After successful installation of the robot according to the operating instructions, you can switch on the robot system.

DANGER!

Danger resulting from incorrect installation

- ▶ Installation must be carried out only by persons with technical and electrotechnical training who have also been authorized to do so by fruitcore robotics GmbH.

horstPANEL has a touch-sensitive touchscreen display. It is operated by touching the display with the fingers.

ATTENTION!

Do not operate the display with sharp or pointed objects.

2.1 Switching on the robot system

- ▶ Switch the main switch on the horstCONTROL to ON.

⇒ On horstPANEL, horstFX starts.

⇒ The main menu appears on the display.

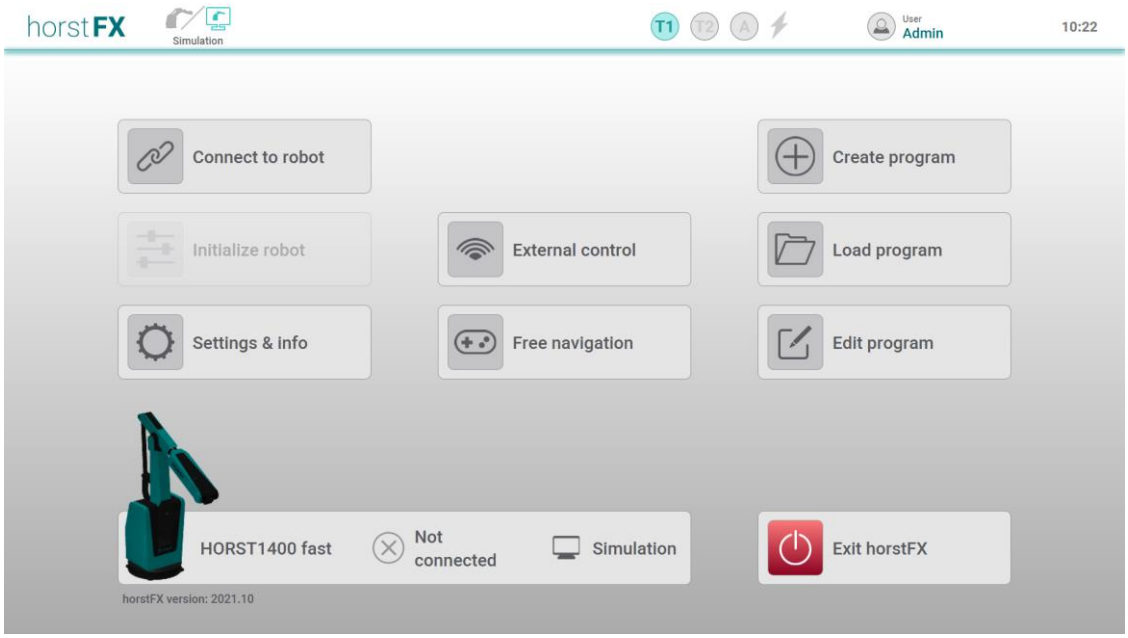


Fig. 2-1: Main menu

⇒ If a user role that is not authorized to remain logged in after a restart was logged in when horstFX was last used, the pop-up window for switching the user role appears.

For information on the user roles, please see section 3.2.

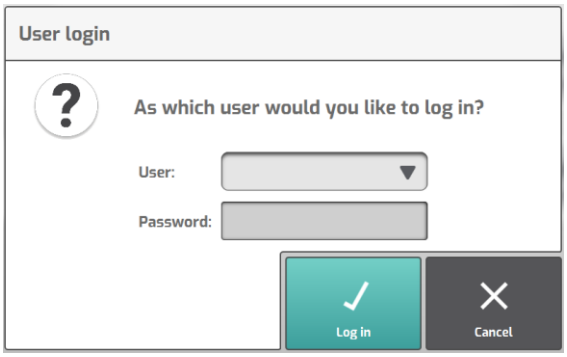


Fig. 2-2: Switching user roles

- ▶ In the main menu, select Connect to robot.
- ▶ Wait until the display shows “Connection with robot established successfully”.

In the next step, the robot must be initialized.

2.2 Initializing the robot

WARNING!
Danger of impact and crushing due to robot movement

The safety stop function is deactivated during the initialization.

- ▶ During initialization operation, close off the area around the robot and protect it against access by unauthorized persons. There must be no persons in the danger zone of the robot.

The initialization must be performed whenever the robot system is switched on after the power supply was interrupted.

During the initialization, you should check the enabling switch for proper function by deliberately letting it go and pressing it down every so often.

- ▶ In the main menu, select Initialize robot.

⇒ The Automatic initialization menu appears.

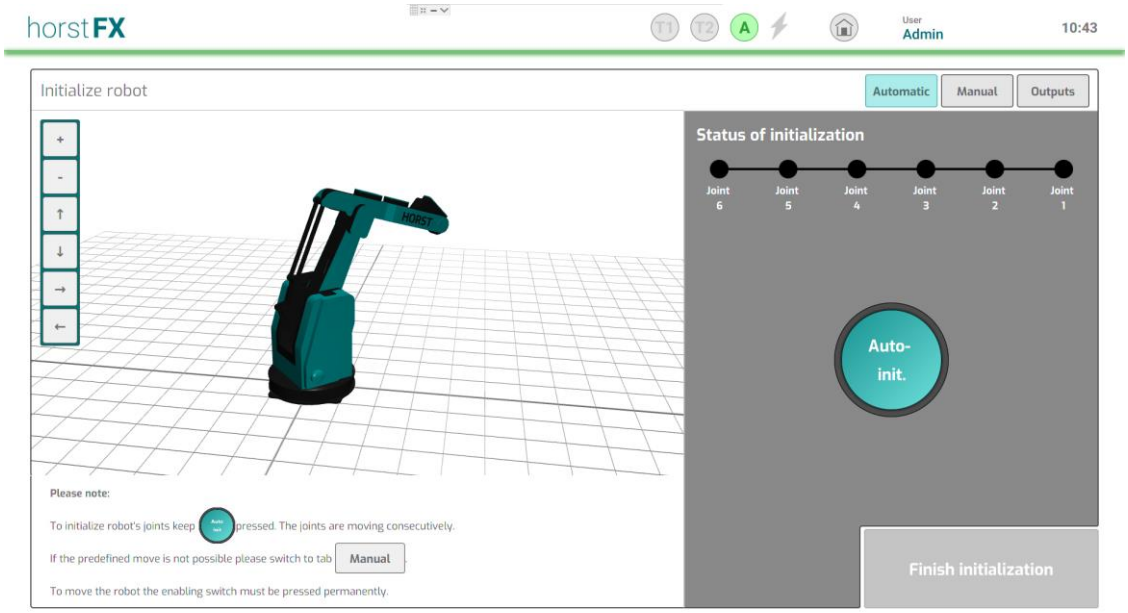


Fig. 2-3: ***Automatic initialization*** *menu*

The menu at the top right displays the initialization status of the six axes of the robot in the form of dots. The axes that have not yet been initialized are shown as black dots. After the initialization, the color changes to turquoise.

- ▶ Press and hold the enabling switch in the center position.
- ▶ Press and hold the **Auto Init** button.

⇒ The automatic initialization of the axes is performed.

⇒ If the initialization was successful, all six dots (initialization status) for the axes are shown in turquoise.

To perform the initialization, the axes (starting with axis 6) must carry out a movement one after the other. If this is not possible due to spatial issues, the axes must be moved manually. In this case, switch to the **Manual initialization** menu.

ATTENTION!

Keep an eye on the robot to prevent collisions.

- ▶ Select the **Manual** button.

⇒ The **Manual initialization** menu appears.

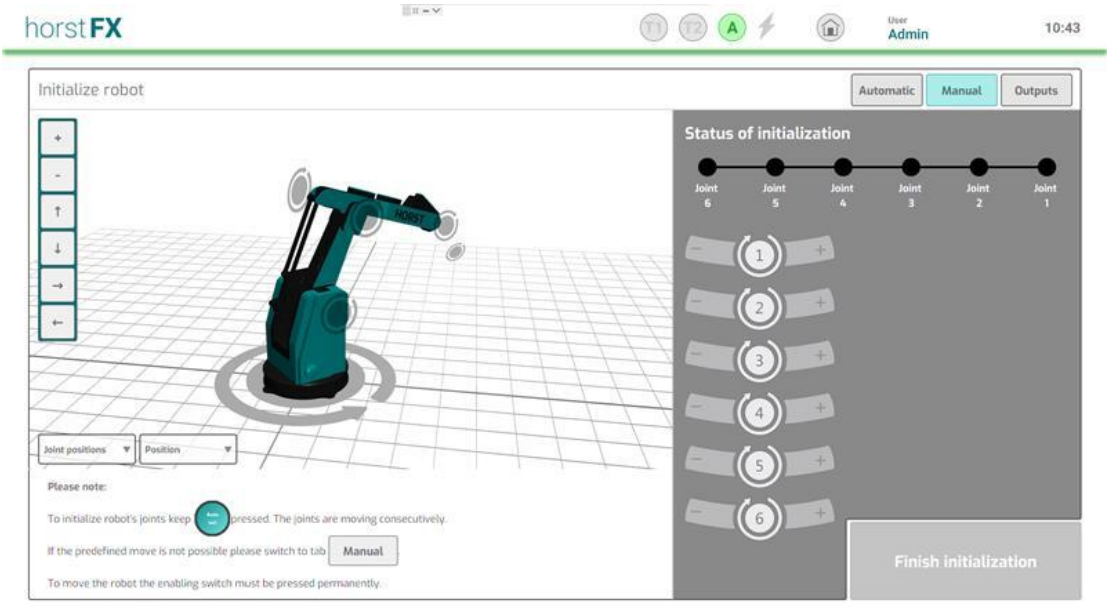


Fig. 2-4: ***Manual initialization*** *menu*

This allows the axes to be moved manually in the case that automatic initialization is not possible.

- ▶ Press and hold the enabling switch in the center position.
- ▶ Select the axes one after the other and move them until successful initialization is displayed.

⇒ If the initialization was successful, the dot (initialization status) for the respective axis is shown in turquoise.

It may be necessary to open a gripper to perform the initialization. In this case, switch to the **Initialize robot – Outputs** menu via the **Outputs** button.

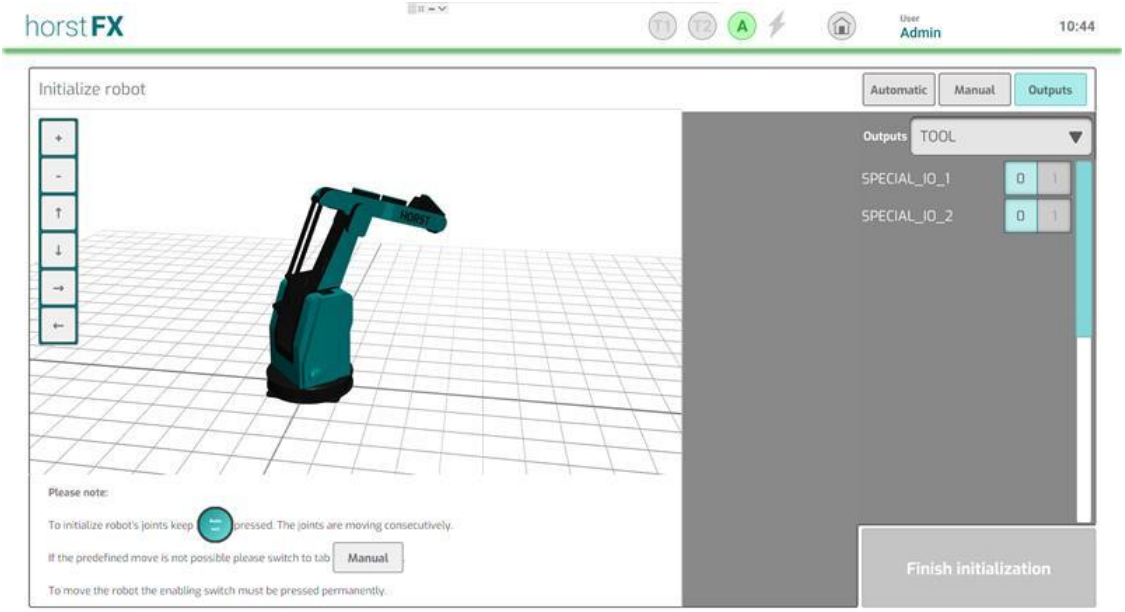


Fig. 2-5: ***Initialize robot – Outputs*** *menu*

Outputs can be switched manually here. For example, a gripper can be opened before the initialization drive is continued.

▶ Switch the desired output via the corresponding toggle button.

The automatic/manual initialization of the axes was successful if all six dots (initialization status) for the axes are shown in turquoise.

⇒ The Complete initialization button is activated.

▶ Press the Complete initialization button.

⇒ The initialization of the robot is completed.

⇒ The main menu is shown again.

⇒ The robot is ready.

3 First Steps

In this chapter, the start screen with the main menu and other elements are described. Other general elements of horstFX are also presented.

3.1 Navigation (menu bar)

General navigation in horstFX is described below.

The following buttons and displays may appear in the menu bar at the top of the screen:

Fig. 3-1: Menu bar view 1

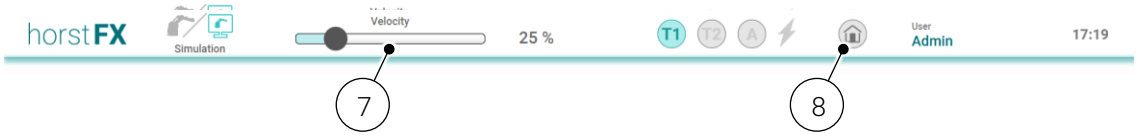


Fig. 3-2: Menu bar view 2

No.	Description
1	**Change control mode** toggle button – switches the control mode between **Real** and **Simulation** In **Simulation** mode, only the movements of the robot model are displayed in the 3D world. In **Real** mode, the robot performs the movements, and the movements of the robot model are displayed in the 3D world.
2	**Operating mode** display – displays the currently selected operating mode T1 – Teaching mode – manual operation at reduced speed T2 – Teaching mode – manual operation at high speed A – automatic mode
3	↯display – warning and error messages The symbol flashes red for unacknowledged messages: emergency stop, safety stop, and system error.
4	**User role** button – switches the user role
5	Display of the current (logged-in) user role
6	Time display
7	**Speed controller** with speed display (in percent) – sets the speed at which a program is executed
8	**Main menu** button – for navigating to the main menu

3.2 User roles

Several user roles are available for the use of horstFX. Each user role has various permissions.

When horstFX is delivered, the Administrator user role is used by default and is automatically logged in. This does not require a password. A password can be set for the Administrator user role via the Settings – Passwords menu (see section 4.1.1). Once a password has been set, the Programmer and Operator user roles can be enabled and a password can also be set for each of them, if desired.

WARNING!
Danger due to incorrect user management!

- ▶ It is essential to ensure that the passwords for the user roles only reach authorised persons. User management is intended to ensure that safety configurations, such as axle restrictions, cannot be changed unnoticed.

ATTENTION!

To save changes to acceptance/safety-relevant configurations (e.g. safety I/O), the Administrator user role should only when necessary.
The Programmer user role should be used by default. By setting a password for the Administrator user role, the Programmer user role can be enabled (see section 4.3.1).

The following table shows the permissions for each individual user role:

User role	Description/authorizations
Operator	Loads, executes and saves programs – controls robot manually – switches outputs – connects with robot – log console
Programmer	As for Operator and also edits/deletes programs – controls robot externally – changes tool – 3D world and 3D objects – uses Modbus – sends data to horstCOSMOS – configures user-specific operating view
Administrator	As for Programmer and also edit programs in automatic operating mode – updates horstFX – sets offset axis 6 – changes passwords and enables user roles – sets controller values – defines axis constraints – changes robot model – configures safety I/O

Due to the different permissions, various menus, submenus, buttons, and similar elements are not visible or available to every user role.

3.3 Main menu

After switching on horstCONTROL or starting horstFX, the main menu appears on the display of horstPANEL.

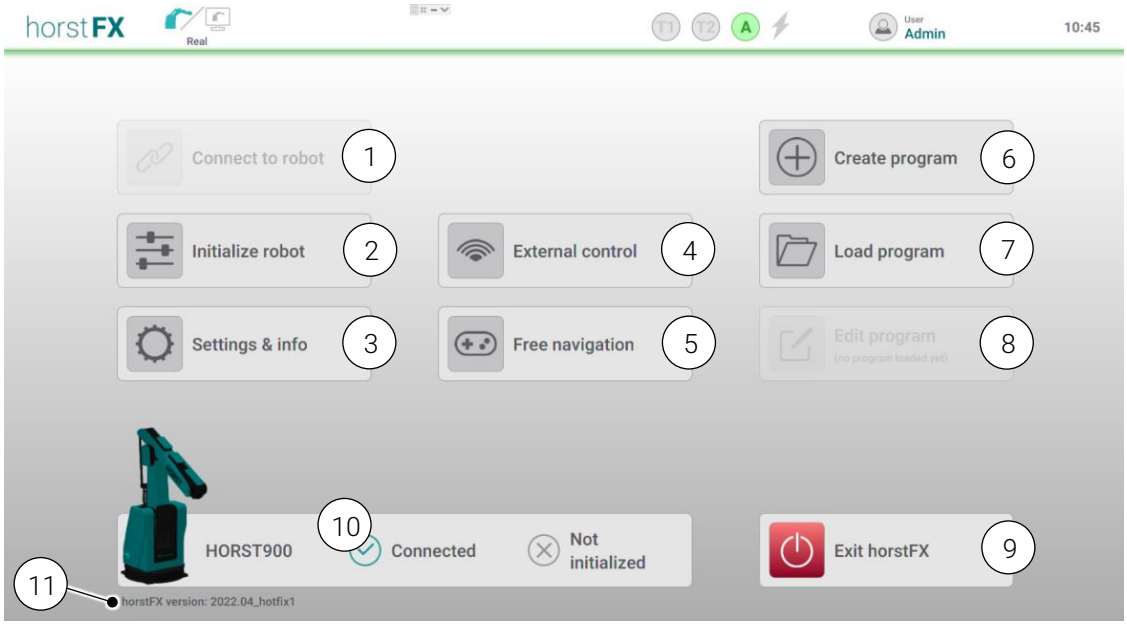


Fig. 3-3: Main menu

No.	Description
1	**Connect to robot** button – establishes a connection between horstFX and the robot.
2	**Initialize robot** button – opens the **Initialize robot** menu (see section 2.2).
3	**Settings & info** button – opens the **Settings & info** menu (see section 4). Various settings can be selected (select/create tool, set controller values, change passwords, etc.).
4	**External control** button – enables the robot to be controlled by commands that have been sent externally (see section 8).
5	**Free navigation** button – the robot can be moved manually without a program (see section 4.9.1).
6	**Create program** button – creates a new program (see section 6.1).
7	**Load program** button – opens the file manager to load a saved program (see section 6.2).
8	**Edit program** button – opens the last edited program (program name in brackets) and allows it to be edited (see section 6.3).
9	**Exit horstFX** button – closes horstFX.
10	**Robot info** display – displays information/status of the robot • Robot model • Status indicating whether the software is connected to the robot • Status indicating whether robot is initialized (if not connected to robot, it instead indicates here that **Simulation** mode is active)
11	**horstFX version** display– displays the version of horstFX

3.4 Robot model view

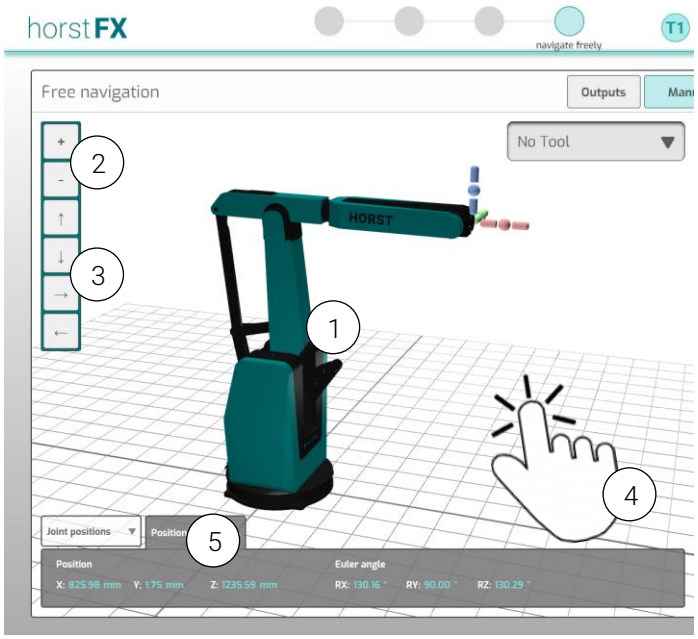


Fig. 3-4: View of the robot model

When the robot model (1) is shown in its current pose in the 3D world, the view of it on the display can be changed:

- Zoom – by tapping the buttons (2) [+] and [-]
- Move – by tapping the buttons (3) [↑], [↓], [→] and [←]
- Rotate – by tapping (holding down) and dragging your finger (4) on the display

In addition, there is a pop-up window (5) with information about both the current values of each axis and its boundaries, and the current coordinate position (X, Y, Z) and orientation (in terms of Euler angles) of the TCP.

3.5 On-screen keyboard

If entries have to be made in horstFX, this is made possible by an on-screen keyboard. There is a standard on-screen keyboard for text input (see Fig. 3-5) and a smaller on-screen keyboard for entering numeric values (see Fig. 3-7). The correct on-screen keyboard appears automatically, depending on the type of input field.

In case of an invalid character or invalid text, a note on horstPANEL and/or corresponding warning symbols appear (see Fig. 3-6), depending on the type of input field.

The keys of the numeric on-screen keyboard have the same functionality as the keys of the standard on-screen keyboard. It just has one other C key, with which the entire input can be deleted (see Fig. 3-7, (1)).

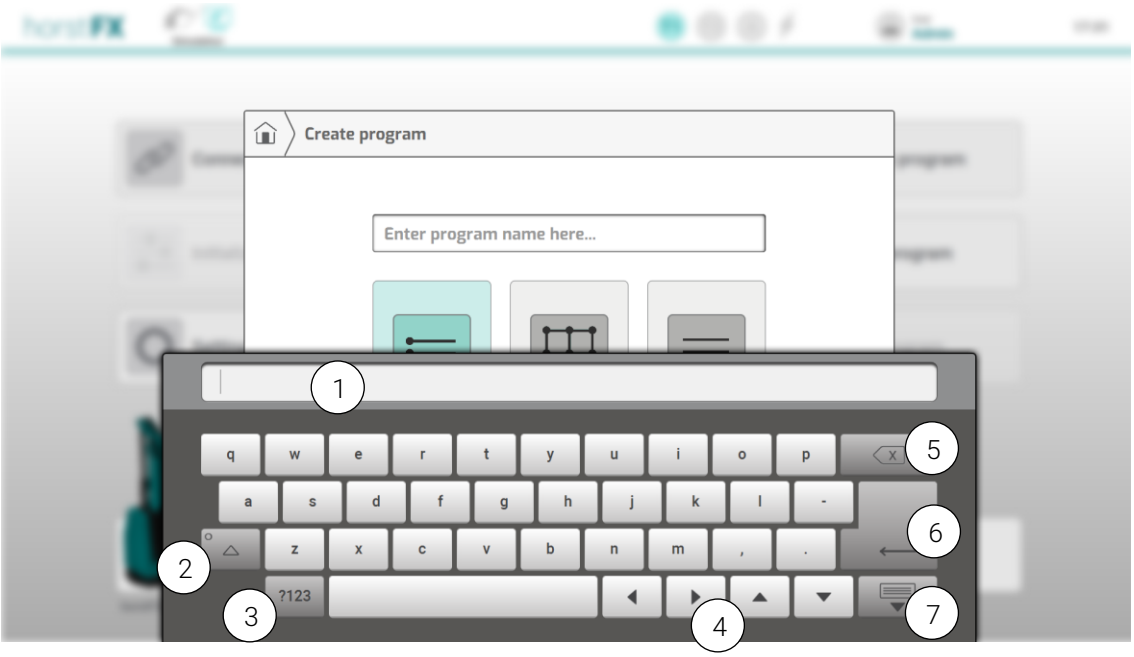


Fig. 3-5: On-screen keyboard for text input

No.	Description
1	Text display – display of the entered text integrated into the on-screen keyboard (the input field may be hidden by the on-screen keyboard depending on its position)
2	Shift key – switches to capital letters By tapping once, the keys automatically switch back to lowercase after the next entry. Tapping twice will lock the Shift key, which can be unlocked by tapping the Shift key again.
3	Numbers/special characters key – switches to numbers and special characters
4	Arrow keys – move the cursor position
5	Delete key – deletes the character to the left of the current cursor position
6	Enter key – confirms the input and hides the on-screen keyboard
7	Hide key – hides the on-screen keyboard (input is confirmed)

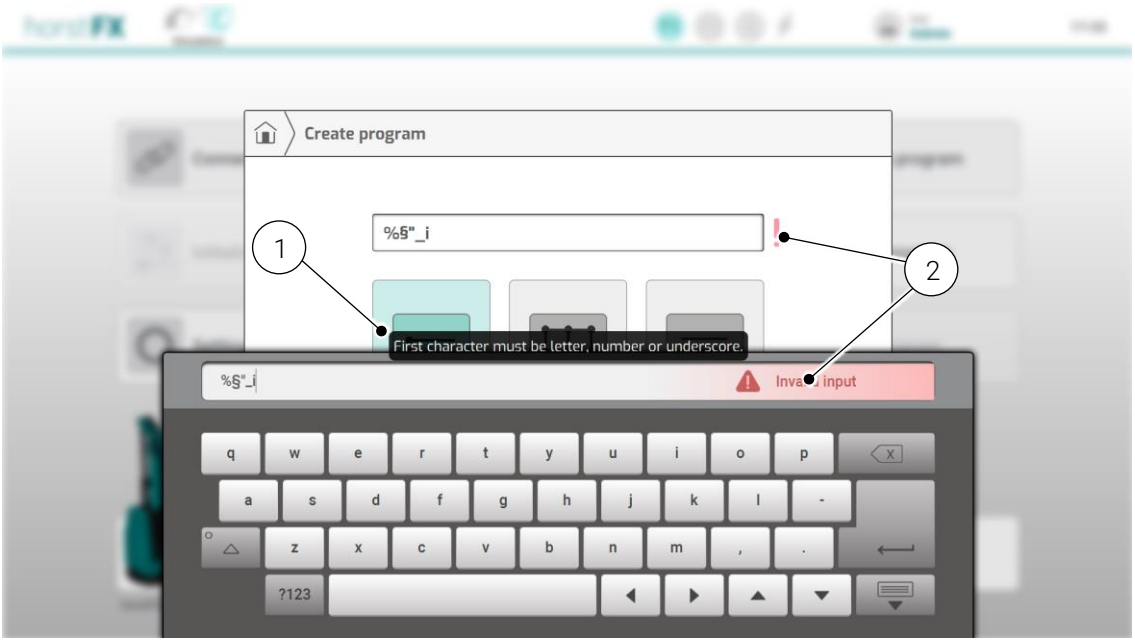


Fig. 3-6: On-screen keyboard for text input with invalid text

No.	Description
1	Note – note about invalid characters (disappears automatically after a few seconds)
2	Warning icons – indicate that an input is invalid

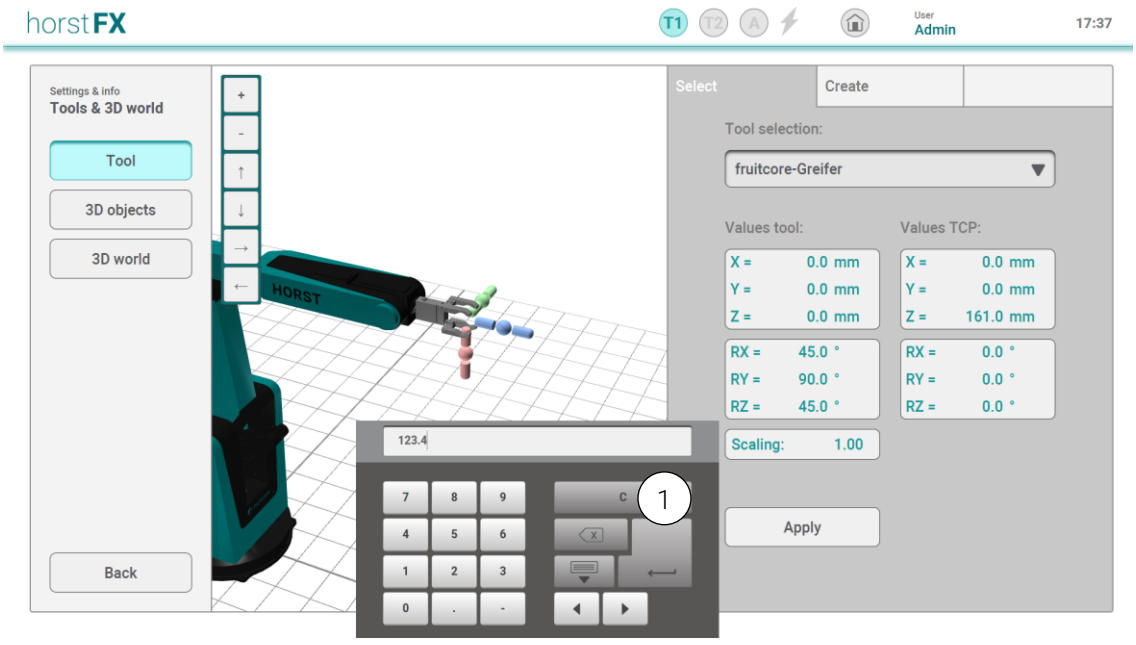


Fig. 3-7: On-screen keyboard for number input

No.	Description
1	C key – deletes the entire input

4 Settings

Pressing the **Settings & info** button in the main menu selects the **Settings & info** menu.

In the settings menus, various settings can be selected and set (e.g. select/create tools and 3D worlds, set controller values, change passwords, add 3D objects, etc.) and various information can be viewed.

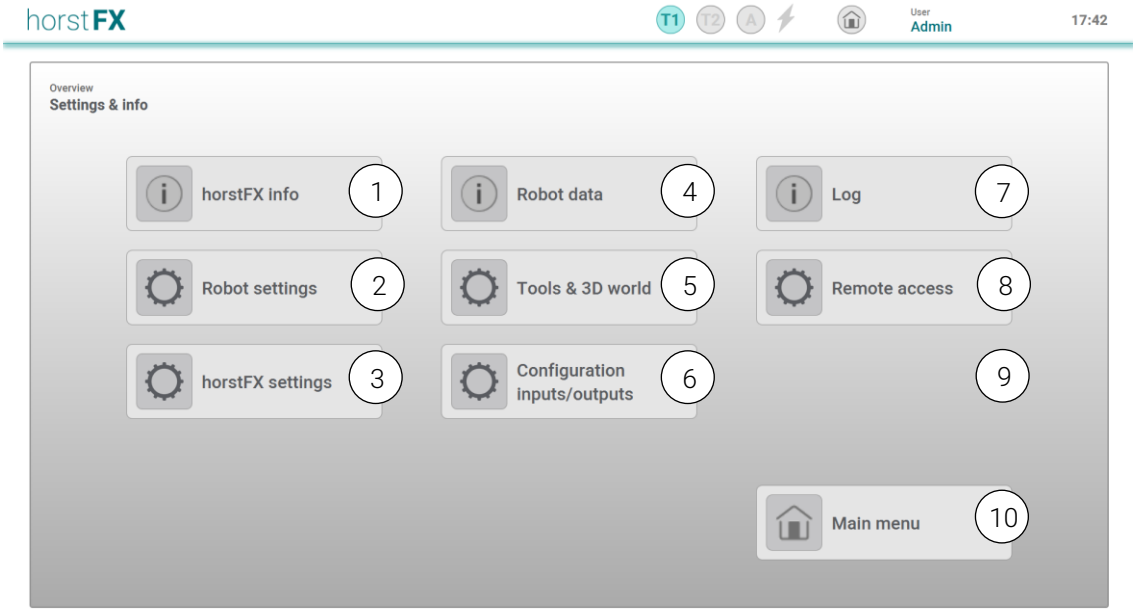


Fig. 4-1: ***Settings & info*** *menu*

No.	Menu/info point	Reference
1	**horstFX info** point	Section 4.1
2	**Robot settings** menu	Section 4.2
3	**horstFX settings** menu	Section 4.3
4	**Robot data** info point	Section 4.4
5	**Tools & 3D world** menu	Section 4.5
6	**Configuration of inputs/outputs** menu	Section 4.5.4
7	**Log console** info point	Section 4.7
8	**Service/remote access** menu	Section 4.8
9	**External interface** menu	Section 4.9
10	**Main menu** button – for navigating to the main menu	Section 3.3

4.1 About horstFX info point

In the **About horstFX** info point, you can see information about individual versions of software and hardware components, changes regarding backward compatibility and new functionalities as well as various licenses.

4.1.1 Version

Under **Version**, you can see the version of horstFX and the versions of various other software and hardware components.

The software version (horstFX version) can also be updated in this menu via the Update horstFX button. Under certain circumstances, the computer integrated in horstCONTROL for horstFX re-starts during the update process.

Here it is also possible to export files and information on a specific status of the robot system (relevant for service cases) via the Export status button.

ATTENTION!

During the update process it is possible that digital outputs are switched off.

▶ After an update, make sure that all outputs are at the status before the update.

4.1.2 Compatibility

The Compatibility info point lists and briefly describes non-backward compatible changes and new functionalities of the individual software versions (horstFX versions).

4.1.3 Licenses

Under Licenses, third-party software licenses used in horstFX are listed.

4.2 Robot settings menu

In the Robot settings menu, the controller values can be adjusted, axis constraint configurations can be created and selected, and an offset for axis 6 can be set.

4.2.1 Controller values

In the Configure values menu, a controller value set can be set to configure the adjustment of the robot. In addition, a user-specific controller value set can be configured where the values can be freely selected within their valid range for the individual axes.

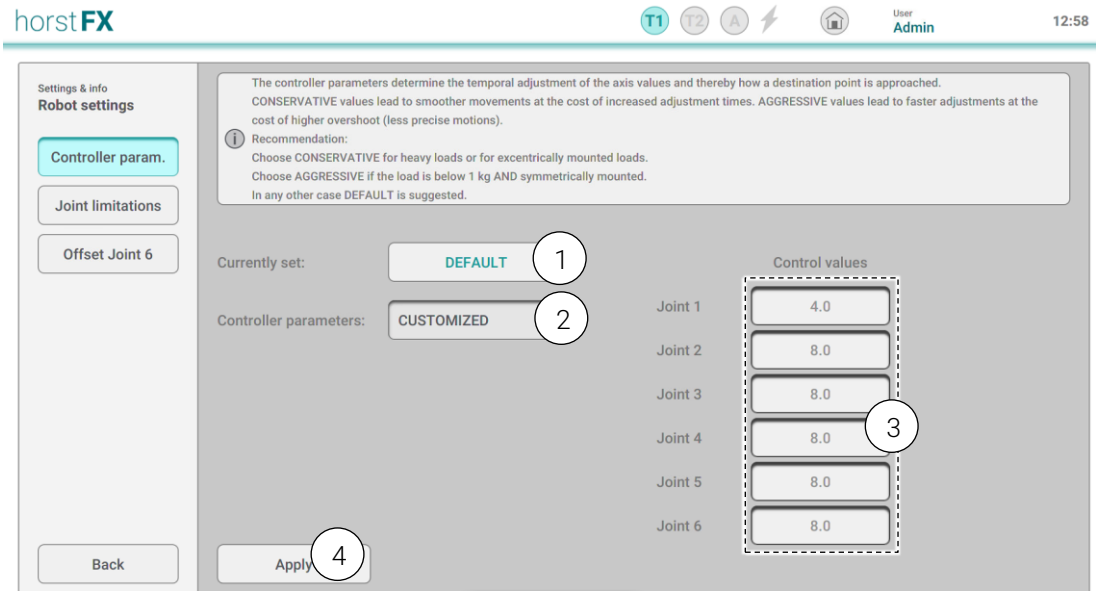


Fig. 4-2: Robot settings – Controller values menu

Pos.	Description
1	Current set display - shows the currently set controller value set.
2	Controller parameters selection field - selection of a controller value set

Pos.	Description
3	**Controller value set** input field – for entering the individual axis values (only active if CUSTOMIZED controller value set is selected)
4	**Apply** button – activates the selected set of controller values

CAUTION!

While updating it is possible that digital outputs are switched off.

▶ After an update, ensure that all outputs are at the state prior to the update..

4.2.2 Axis constraints

In the **Axis constraints** menu, user-defined configurations of axis constraints that differ from the default values can be created and activated. It is also possible to edit or delete existing configurations.

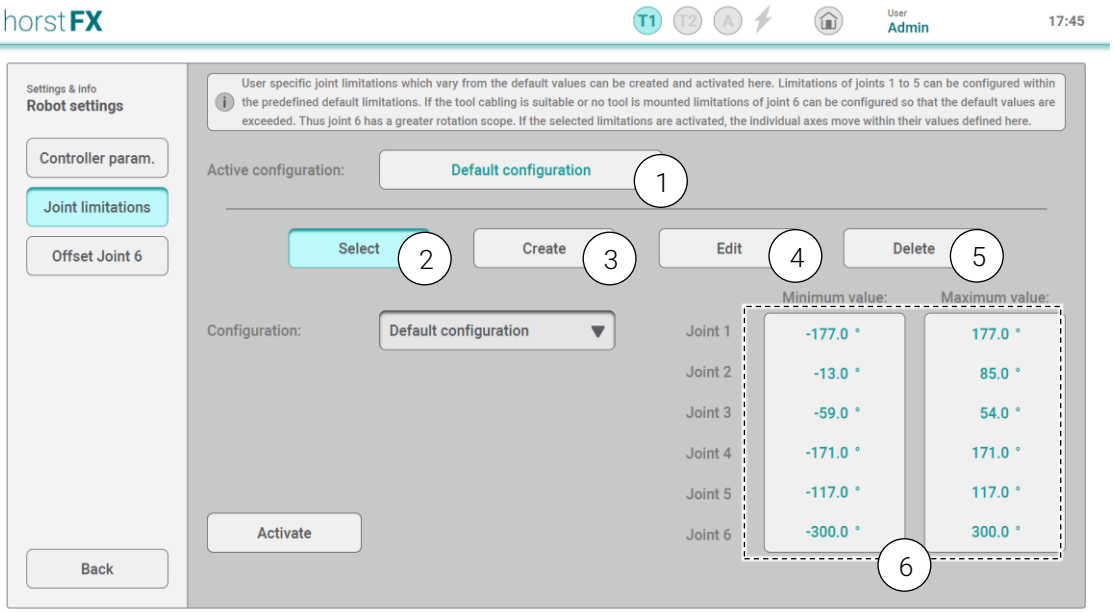


*Fig. 4-3: **Robot settings – Axis constraints** menu*

No.	Description
1	**Active configuration** display – shows the currently active configuration
2	**Select** button – displays the **Select** submenu
3	**Create** button – displays the **Create** submenu
4	**Edit** button – displays the **Edit** submenu
5	**Delete** button – displays the **Delete** submenu
6	Depending on the selected submenu either **Joint values** display – displays the axis values of the selected configuration or **Joint values** text boxes – for entering the desired values

4.2.3 Offset axis 6

For technical reasons, it may happen that axis 6 is correctly aligned after initialization but displays a value outside its valid range. The value deviates by a multiple of 360°.

This incorrect value can be corrected in the **Offset axis 6** menu.

4.3 horstFX settings menu

The **horstFX settings** menu is used to control settings that affect the horstFX software. These include the management of user roles or changing the language, for example.

4.3.1 Passwords

The user roles are managed in the **Passwords** menu. The passwords for the user roles can be changed, and additional user roles can be activated.

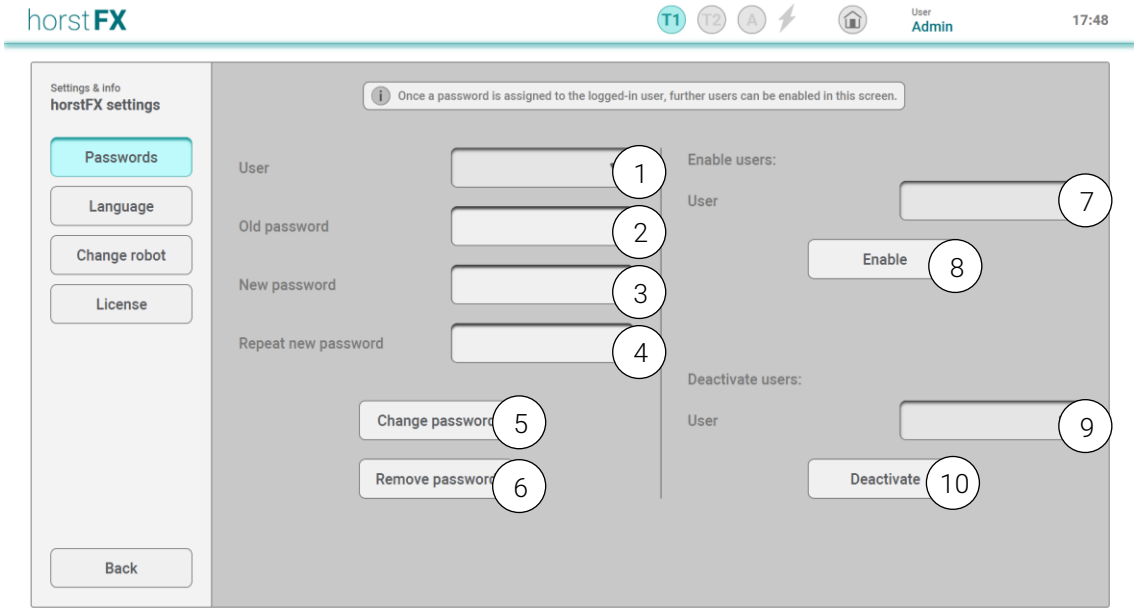


*Fig. 4-4: **horstFX settings – Passwords** menu*

No.	Description
1	**User role** selection field – selects a user role for which the password is to be changed
2	**Old password** input field – for entering the old password
3	**New password** input field – for entering the new password
4	**Repeat new password** input field – for entering the new password again
5	**Change password** button – changes the password
6	**Remove password** button – deletes an existing password
7	**User role** selection field – selects a user role to be unlocked
8	**Enable** button – unlocks the selected user role
9	**User role** selection field – selects a user role to be disabled
10	**Deactivate** button – deactivates the selected user role

4.3.2 Language

In the **Language** menu, the language of horstFX can be changed.

4.3.3 Change robot

In the **Change robot** menu, the robot model used in horstFX can be changed.

4.3.4 License

In the **License** menu, you can see the currently activated license of horstFX and can enter a license key to activate another license.

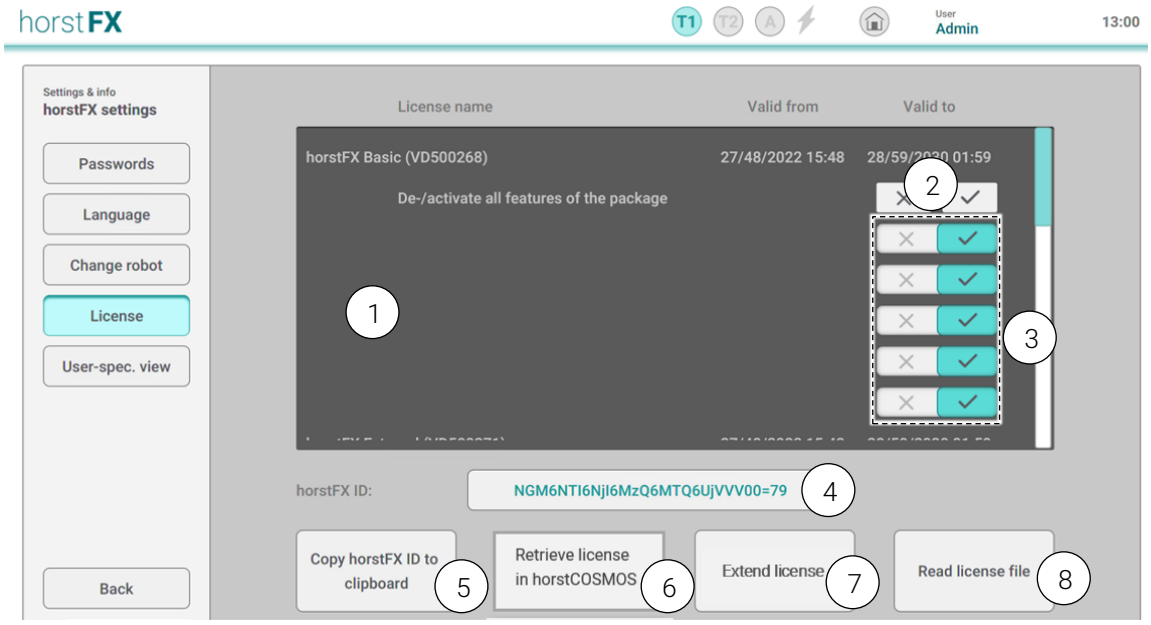


Fig. 4-5: **horstFX settings – License** menu

Pos.	Description
1	Display of the current license as well as overview of all included features
2	**De-/Activate** buttons - de-/activates all features of the package
3	**De-/Activate** toggle button – de-/activates the corresponding feature
4	**horstFX-ID** display – shows the horstFX-ID
5	**Clipboard** button – copies the horstFX-ID to the clipboard
6	**Retrieve license** button – retrieves all licenses reserved in horstCOSMOS and adds them to the display (however, buttons (2) and (3) are only enabled for the currently used license). This button is not available on horstCONTROL.
7	**Extend license** button – extends the currently used license. If no active license is available, this button can be used to reserve a (still free) license for one day.
8	**Read license file** button – loads a license file (necessary only if there is no Internet connection)

4.3.5 User-specific view

In the User-specific view menu, settings/configurations are made for the user-specific operating view menu (see section 6.9).

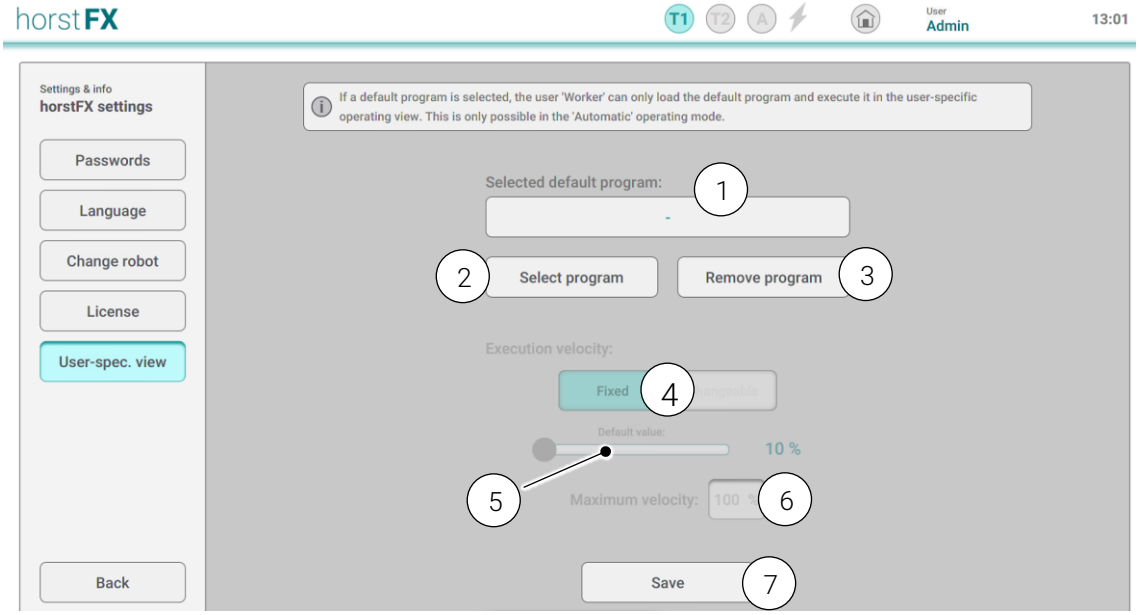


*Fig. 4-6: **horstFX settings – User-specific view** menu*

Pos.	Description
1	**Selected program** display - displays the selected standard program
2	**Select program** button – selection of a standard program
3	**Remove program** button – removes the selected standard program
4	**Execution velocity** toggle button – selection between *Fix* and *Changeable*
5	**Speed controller** with speed display (in percent) – sets the speed at which a program is executed Depending on selection (4): - Fix: Default value (not changeable) - Changeable: Initial value (value is set when the program is loaded)
6	**Maximum velocity** input – sets the upper limit of the speed controller for program execution (only definable if Changeable option (4) is selected)
7	**Save** button – apply and save the set configuration and selection options

4.4 Robot data info point

In the **Robot data** info point, information on robot system data can be seen. A connection to horstCOSMOS can also be established to transfer data.

4.4.1 Robot system data

The **Robot system data** menu displays various data collected since recording began. The total revolutions of the six robot axes are displayed, as well as information on overload (step loss) in the form of the number of step loss errors in the last 24 hours and also the times of the last three occurrences.

4.4.2 horstCOSMOS

In the **horstCOSMOS** menu, a connection to horstCOSMOS can be established by logging in with username and password and data can be transferred to horstCOSMOS.
The data that is transferred consists of various robot system data such as the data mentioned in section 4.4.1 (see section 6.3.24), process data, programs (see below), and error logs.

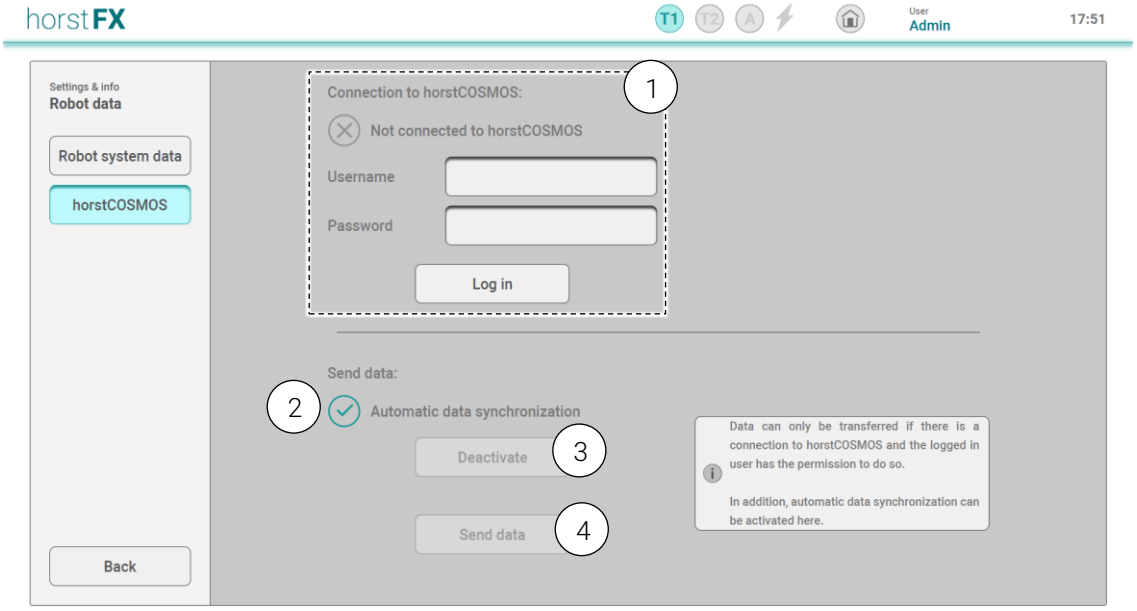


*Fig. 4-7: **Robot data – horstCOSMOS** menu*

No.	Description
1	**Login data** input fields – for entering login data (username and password) to connect to horstCOSMOS
2	**Data synchronization status** display – indicates whether automatic data transmission is activated
3	**(De)activate** button – (de)activates automatic data transmission
4	**Send data** button – manually triggers data transfer if there is a connection to horstCOSMOS

When transferring the data, the programs saved in the default location (*save* folder) are also sent to horstCOSMOS or backed up online. The file manager can be used to check whether a program is saved online in horstCOSMOS (see section 6.2). Programs saved online on horstCOSMOS are marked with a special icon (see Fig. 4-8). As soon as a program is changed after it has been transferred to horstCOSMOS, the changes are **not** automatically saved online on horstCOSMOS, unless

automatic data transfer is activated. In order to also save the changes and thus the current status of modified programs online on horstCOSMOS, the data must be transferred again.

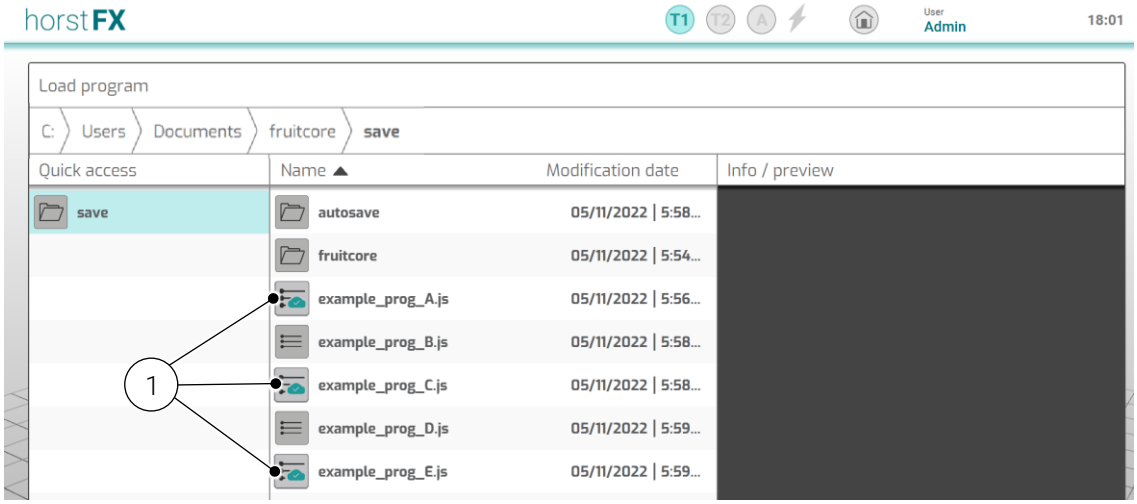


Fig. 4-8: Load program – file manager with programs saved online

No.	Description
1	Special display icon with corresponding symbol for saved programs

4.5 Tools & 3D world menu

The **Tools & 3D world** menu is used to select, create, and edit tools, 3D worlds, and 3D objects.

4.5.1 Tool

In the **Tool** menu, existing tools can be selected, loaded, and adopted in horstFX. New tools can also be created.

4.5.1.1 Select

In the **Select** menu, a tool can be selected, which is then applied in horstFX for all functionalities.

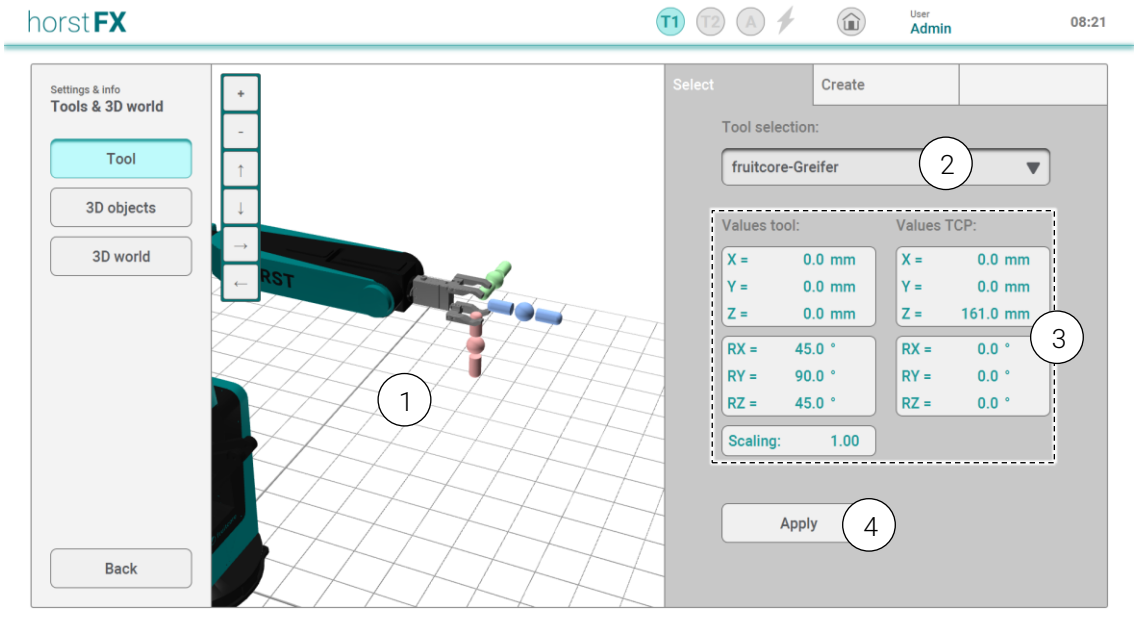


*Fig. 4-9: **Tool – Select** menu*

No.	Description
1	Display of the selected tool and its TCP in the 3D world, as well as the axes of a coordinate system to illustrate the position of the TCP and its orientation
2	**Tool selection** field – for selecting a tool
3	Display of all values of the selected tool and its TCP
4	**Apply** button – the selected tool is added to the robot model in the 3D world and used in horstFX

4.5.1.2 Importing tools

The tool to be imported must be in STL (**.stl**) file format. Other formats are not supported.

In the file system of horstCONTROL, the file must be saved under **/home/fruitcore/fruitcore/tools**. This directory also contains the sample and standard tools saved in horstFX.

If you want to import STL files for tools into horstFX, you can find information and help at **horstcosmos.com**. If you have any further questions, please contact fruitcore robotics GmbH.

4.5.1.3 Create

In the **Create** menu, tools can be created from the imported STL files. This is a four-step process.

Step 1: name and file

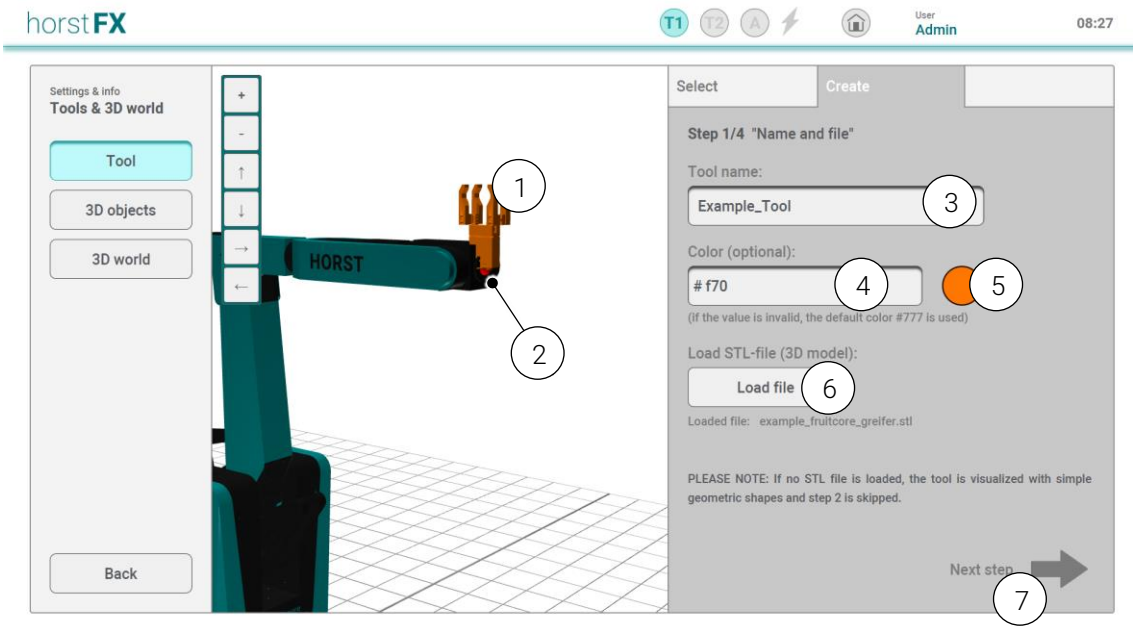


*Fig. 4-10: **Tool – Create** menu (step 1)*

No.	Description
1	Display of the tool in the 3D world
2	Display of the origin of the tool (red sphere) to which translations and rotations of the tool refer
3	**Tool name** input field – for entering the name for the tool to be created

No.	Description
4	**Color** input field – optional input of a color value (hexadecimal)
5	**Color** display – if the color value is valid, the corresponding color will be displayed, otherwise the default color #777 (gray tone) will be used
6	**Load file** button – for selecting an STL file
7	**Next step** button – for continuing to the next step

To create a new tool, it is not absolutely necessary to import and load an STL file. If no STL file is loaded in this step, the tool is visualized with simple geometric shapes and step 2 is skipped.

Step 2: customize tool

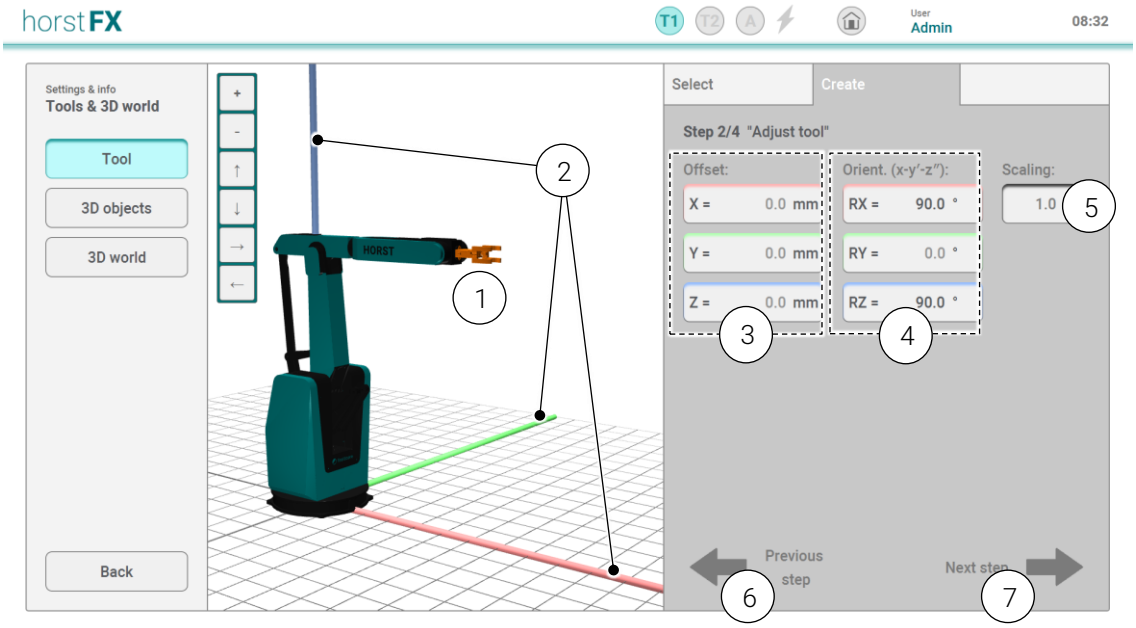


*Fig. 4-11: **Tool – Create** menu (step 2)*

No.	Description
1	Display of the tool in the 3D world
2	Display of the axes of the coordinate system according to which translations and rotations are oriented
3	**Offset** input fields – for entering the X, Y, and Z values (translation)
4	**Orientation** input fields – for entering the RX, RY, and RZ values (rotation)
5	**Scaling** input field – for entering the scaling value
6	**Previous step** button – for going back to the previous step
7	**Next step** button – for continuing to the next step

Each change to one of the values in the input fields triggers an update of the tool, so that the tool in the 3D world always displays the currently defined values.

Step 3: customize TCP

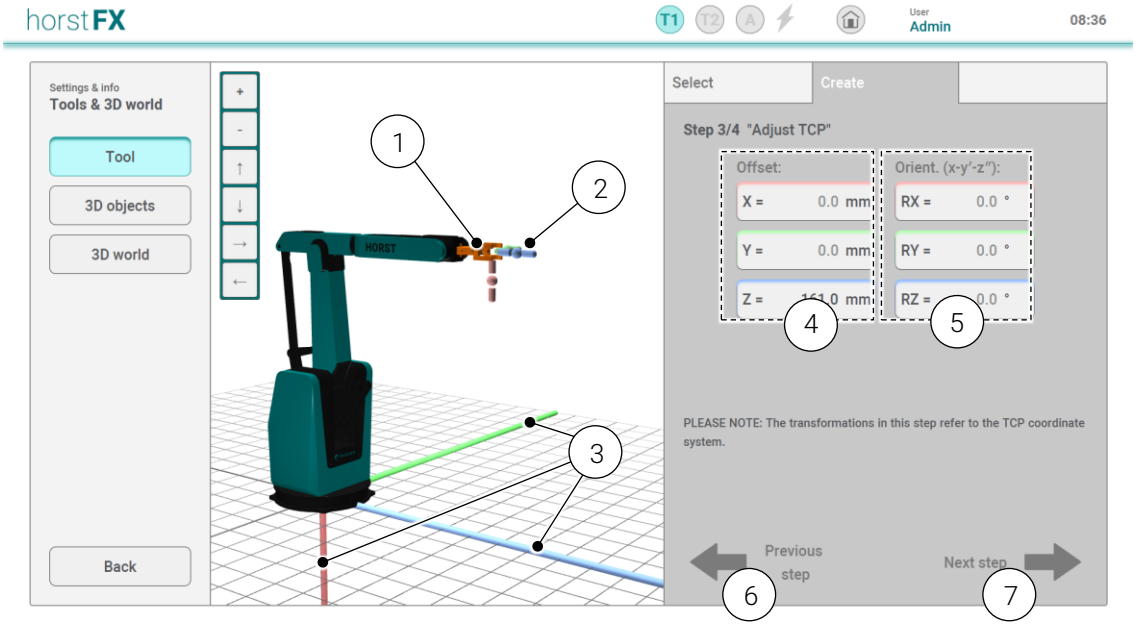


Fig. 4-12: ***Tool – Create*** *menu (step 3)*

No.	Description
1	Display of the tool in the 3D world
2	Display of the axes of a coordinate system for better determination of the position of the TCP and its orientation
3	Display of the axes of the coordinate system according to which translations and rotations are oriented
4	**Offset** input fields – for entering the X, Y, and Z values (translation)
5	**Orientation** input fields – for entering the RX, RY, and RZ values (rotation)
6	**Previous step** button – for going back to the previous step
7	**Next step** button – for continuing to the next step

Each change to one of the values in the input fields triggers an update of the TCP, so that the TCP in the 3D world always represents the currently defined values.

Step 4: overview

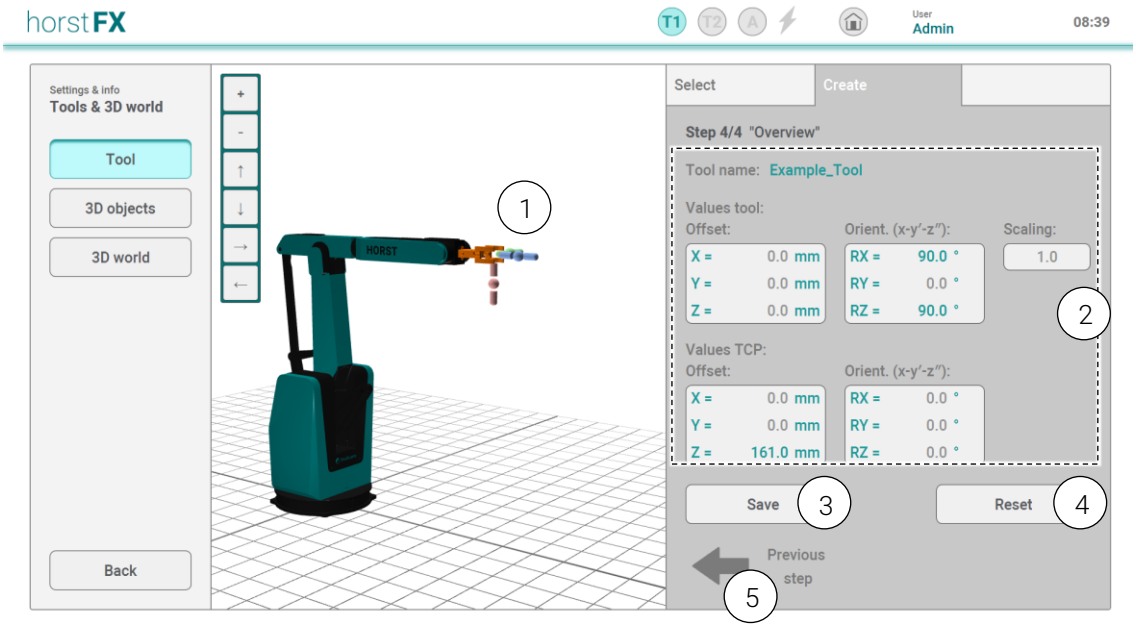


Fig. 4-13: ***Tool – Create*** *menu (step 4)*

No.	Description
1	Display of the tool and the TCP in the 3D world
2	Display of all (entered) values from the previous steps
3	**Save** button – for saving the tool
4	**Reset** button – resets the creation process
5	**Previous step** button – for going back to the previous step

4.5.2 3D objects

In the **3D objects** menu, new 3D objects can be created, which in turn can be used in 3D worlds (see section 4.5.3).

ATTENTION!
horstFX does not automatically consider these 3D objects when limiting the movements of the robot. To avoid collisions, real objects/obstacles located in the workspace must be taken into account during program creation.

4.5.2.1 Importing 3D objects

The 3D object to be imported must be in STL (**.stl**) file format. Other formats are not supported.

In the file system of horstCONTROL, the file must be saved under **/home/fruitcore/fruitcore/objects**. This directory also contains the sample and standard 3D objects saved in horstFX.

If you want to import STL files for 3D objects into horstFX, you can find information and help at **horstcosmos.com**. If you have any further questions, please contact fruitcore robotics GmbH.

4.5.2.2 Create

In the **Create** menu, 3D objects can be created from the imported STL files. This is a three-step process.

Step 1: name and file

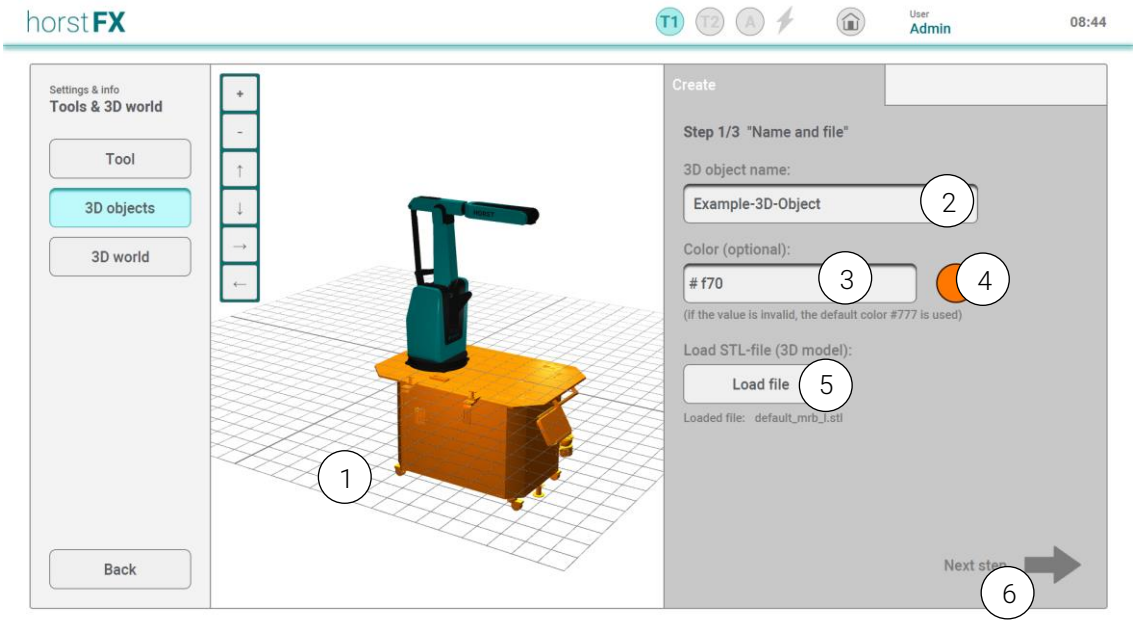


*Fig. 4-14: **3D objects – Create** menu (step 1)*

No.	Description
1	Display of the 3D object in the 3D world including the robot model
2	**3D object name** input field – name for the 3D object to be created
3	**Color** input field – optional input of a color value (hexadecimal)
4	**Color** display – if the color value is valid, the corresponding color will be displayed, otherwise the default color #777 (gray tone) will be used
5	**Load file** button – for selecting an STL file
6	**Next step** button – for continuing to the next step

Step 2: customize 3D object

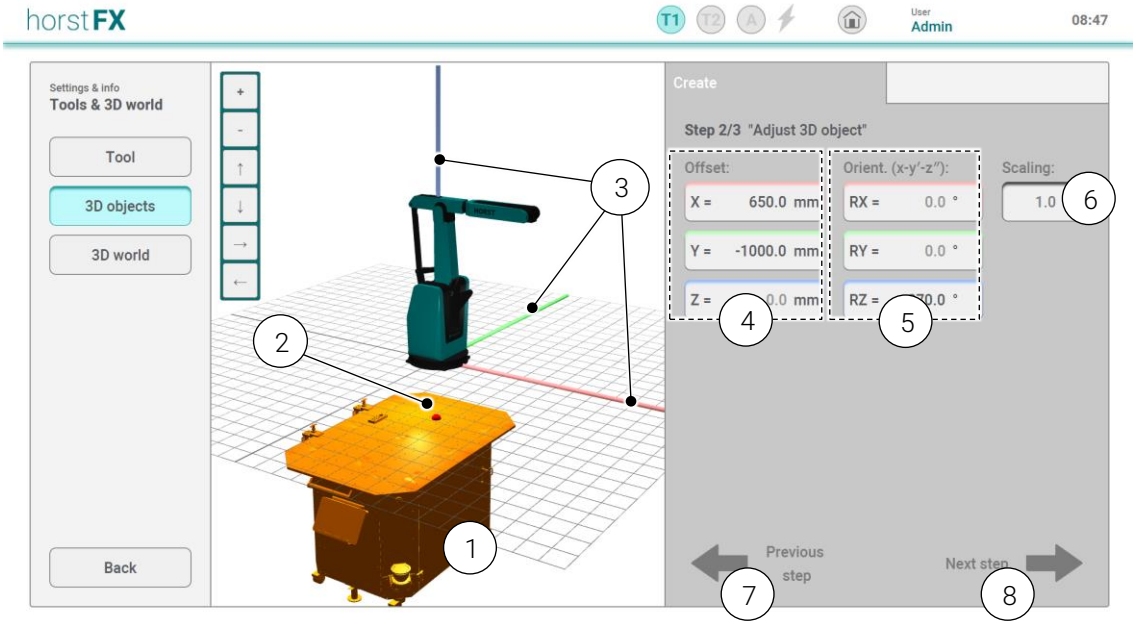


Fig. 4-15: ***3D objects – Create*** *menu (step 2)*

No.	Description
1	Display of the 3D object in the 3D world
2	Display of the origin of the 3D object (red sphere) to which translations and rotations of the 3D object refer
3	Display of the axes of the coordinate system according to which translations and rotations are oriented
4	**Offset** input fields – for entering the X, Y, and Z values (translation)
5	**Orientation** input fields – for entering the RX, RY, and RZ values (rotation)
6	**Scaling** input field – for entering the scaling value
7	**Previous step** button – for going back to the previous step
8	**Next step** button – for continuing to the next step

Each change to one of the values in the input fields triggers an update of the 3D object, so that the 3D object in the 3D world always represents the currently defined values.

Step 3: overview

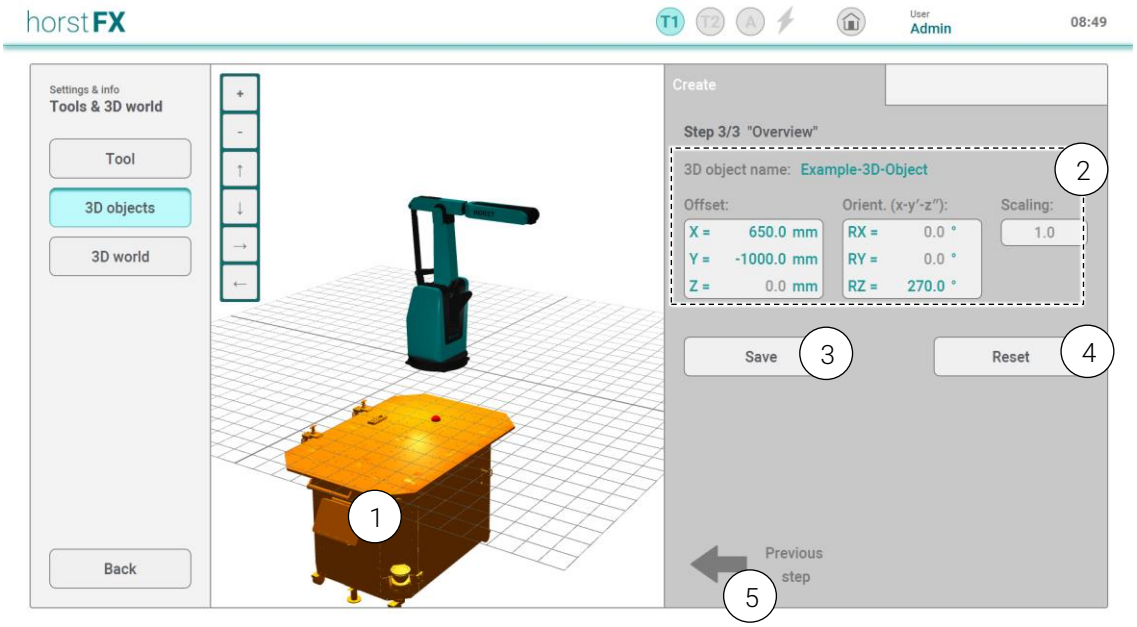


Fig. 4-16: ***3D objects – Create*** *menu (step 3)*

No.	Description
1	Display of the 3D object in the 3D world
2	Display of all (entered) values from the previous steps
3	**Save** button – for saving the 3D object
4	**Reset** button – resets the creation process
5	**Previous step** button – for going back to the previous step

4.5.3 3D world

In the **3D world** menu, an existing 3D world can be adopted, edited, or deleted or a new 3D world can be created. horstFX has the 3D world *EMPTY_WORLD* by default. No 3D objects are added to this world, and it cannot be modified or deleted.

In any other 3D world, existing 3D objects can be added, edited, or removed. This allows process sequences to be prebuilt in the 3D world and, if necessary, programming to be carried out without robots in the simulation.

4.5.3.1 Apply/create

In the **Apply/create** menu, an existing 3D world can be selected, and applied and displayed in horstFX. New 3D worlds can also be created here.

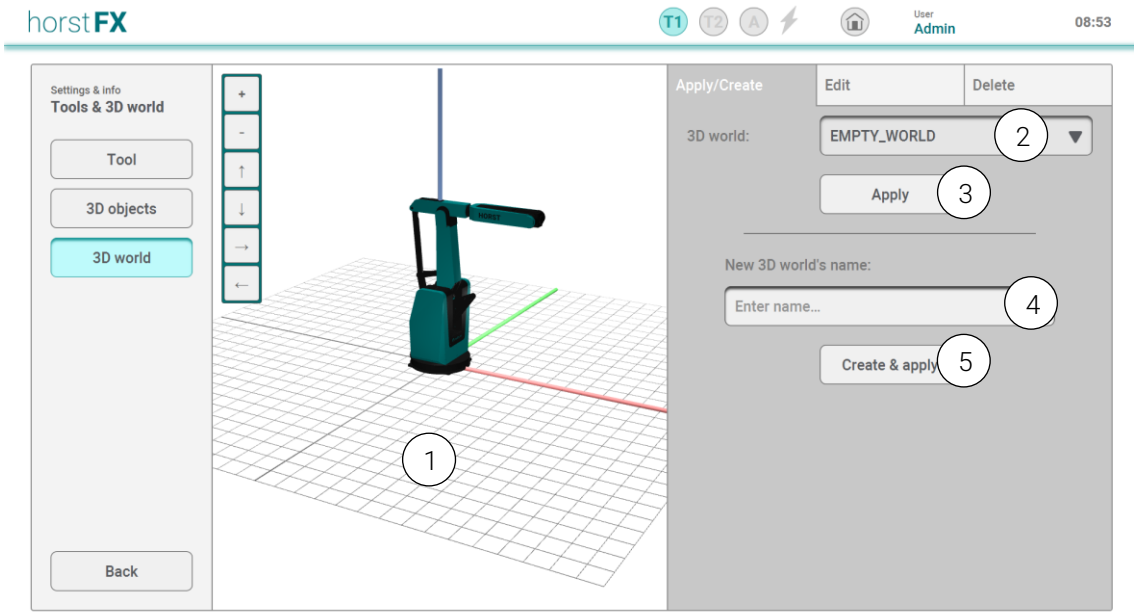


*Fig. 4-17: **3D world – Apply/create** menu*

No.	Description
1	Display of the currently selected 3D world including the robot model and all added 3D objects
2	**3D world** selection field – for selecting an existing 3D world
3	**Apply** button – applies the selected 3D world
4	**Name** input field – name of the new 3D world to be created
5	**Create & apply** button – a new 3D world with name is created and directly applied

4.5.3.2 Edit

In the **Edit** menu, 3D objects of a selected 3D world can be added, edited, and removed.

Adding 3D objects

In the **Add** submenu, 3D objects can be added to the 3D world and their position, orientation, and size can be defined. When a 3D object is selected, it is displayed directly in the 3D world.

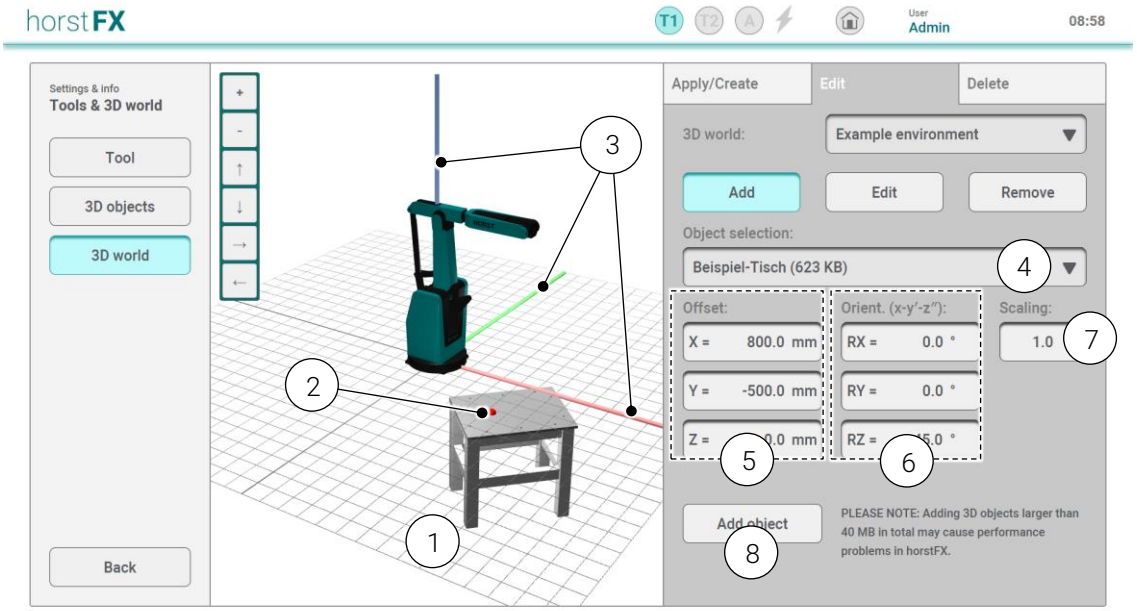


*Fig. 4-18: **3D world – Edit – Add 3D objects** menu*

No.	Description
1	Display of the 3D world including the robot model and all added 3D objects
2	Display of the origin of the 3D object (red sphere) to which translations and rotations of the 3D object refer
3	Display of the axes of the coordinate system according to which translations and rotations are oriented
4	**Select object** selection field – selection of a 3D object (from all default and additionally imported 3D objects)
5	**Offset** input fields – for entering the X, Y, and Z values (translation)
6	**Orientation** input fields – for entering the RX, RY, and RZ values (rotation)
7	**Scaling** input field – for entering the scaling value
8	**Add object** button – loads selected 3D object into the 3D world

Each change to one of the values in the input fields triggers an update of the 3D object, so that the 3D object in the 3D world always represents the currently defined values.

Editing 3D objects

In the **Edit** submenu, the position, orientation, and size of 3D objects that have already been added can be defined. When a 3D object is selected, the selection is displayed in the 3D world in the form of an orange grid model.

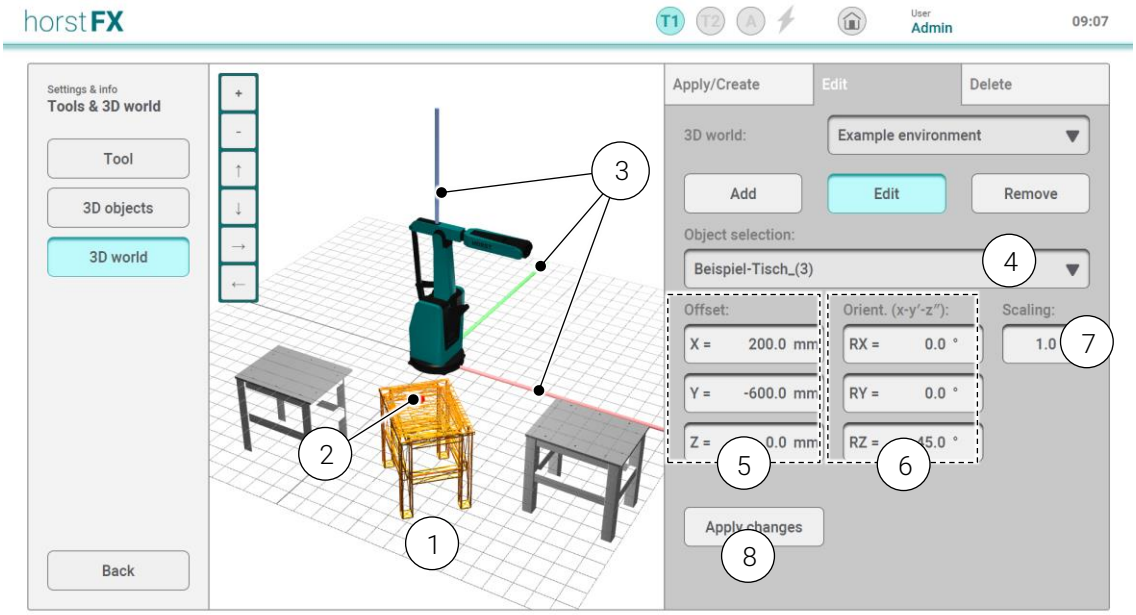


Fig. 4-19: ***3D world – Edit – Edit 3D objects*** *menu*

No.	Description
1	Display of the 3D world including the robot model and all added 3D objects (selected 3D object as orange grid model)
2	Display of the origin of the 3D object (red sphere) to which translations and rotations of the 3D object refer
3	Display of the axes of the coordinate system according to which translations and rotations are oriented
4	**Select object** selection field – selection of a 3D object that has already been added
5	**Offset** input fields – for entering the X, Y, and Z values (translation)
6	**Orientation** input fields – for entering the RX, RY, and RZ values (rotation)
7	**Scaling** input field – for entering the scaling value
8	**Apply changes** button – applies changes to the selected 3D object

Each change to one of the values in the input fields triggers an update of the 3D object, so that the 3D object in the 3D world always represents the currently defined values.

Removing 3D objects

In the **Remove** submenu, 3D objects that have already been added can be removed. When a 3D object is selected, the selection is displayed in the 3D world in the form of an orange grid model.

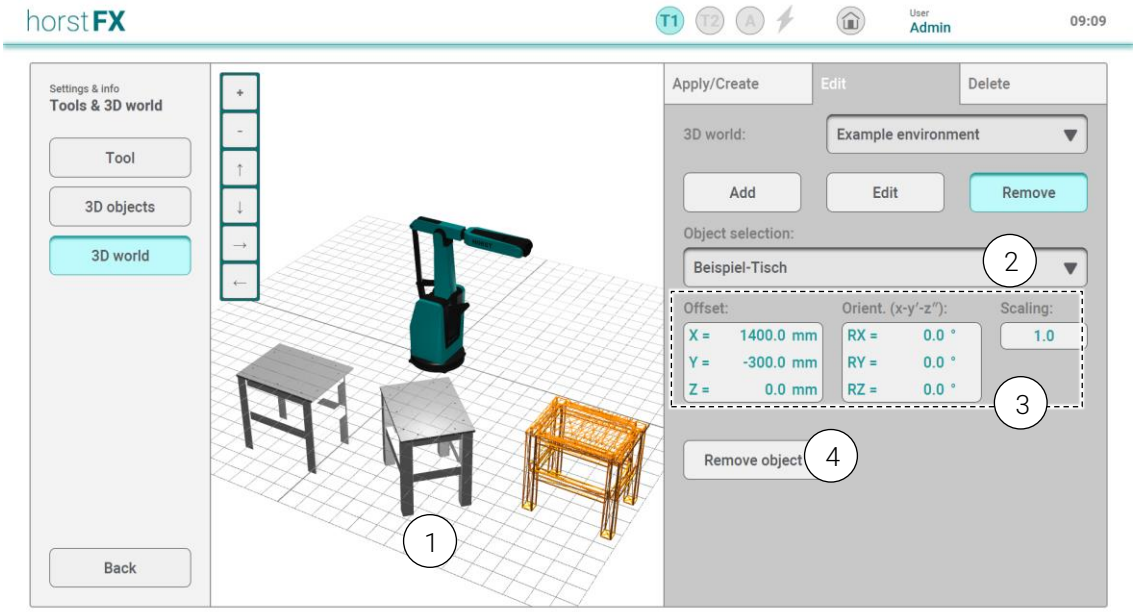


Fig. 4-20: ***3D world – Edit – Remove 3D objects*** *menu*

No.	Description
1	Display of the 3D world including the robot model and all added 3D objects (selected 3D object as orange grid model)
2	**Select object** selection field – selection of a 3D object that has already been added
3	**Offset, orientation**, and **scaling** display – displays the defined values of the selected 3D object
4	**Remove object** button – removes the selected 3D object

4.5.3.3 Delete

In the **Delete** menu, 3D worlds can be deleted.

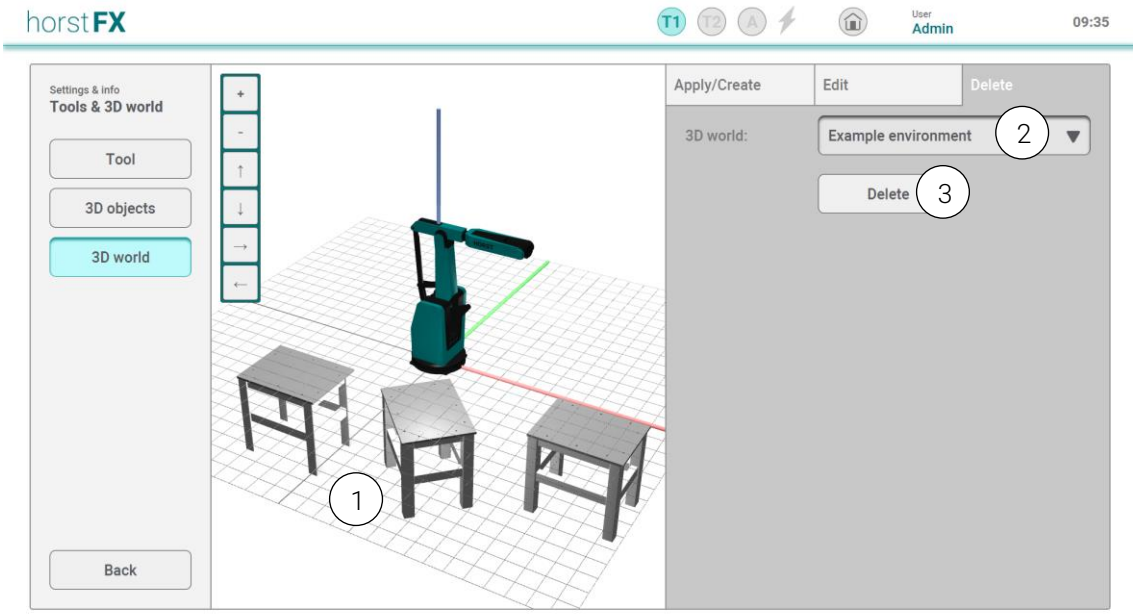


*Fig. 4-21: **3D world – Delete** menu*

No.	Description
1	Display of the currently selected 3D world including the robot model and all added 3D objects
2	**3D world** selection field – for selecting an existing 3D world
3	**Delete** button – deletes the selected 3D world

4.5.4 Coordinate systems (user defined)

In the **Coordinate System** menu, another coordinate system can be created in addition to the basic and ToolkCP coordinate system.
Via the tab "**Create**" a coordinate system can be created in two ways. Either the origin and the orientation are defined directly by input or the coordinate system is taught in by approach.
There are again two possibilities: In the submenu **Axes**, 2 points are defined on the x-axis and the third point on the y-axis. In the submenu **Origin**, first the origin is approached, then a point on the x-axis and then another point, which then lies on the x-y-plane and thus determines the orientation.

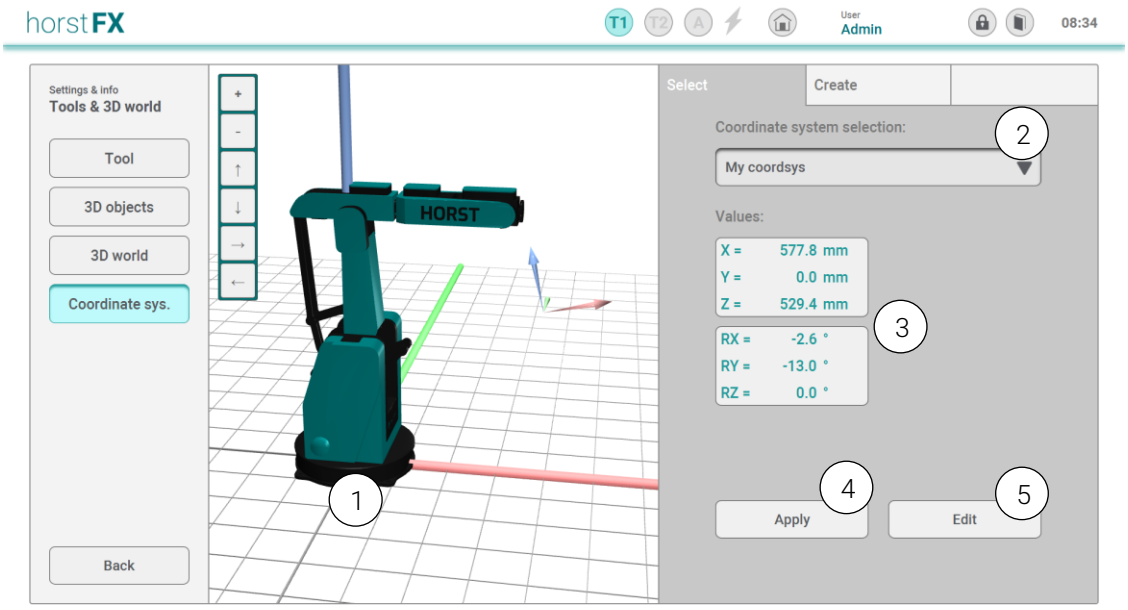


*Fig. 4-22: Menu **Coordinate System** - **Select***

Item	Description
1	Display of the 3D world including the robot model with base coordinate system and selected coordinate system.
2	**Coordinate system** selection field
3	Display of values and orientation of the selected coordinate system
4	**Apply** button - selects coordinate system
5	Button **Edit** - opens editing of the coordinate system

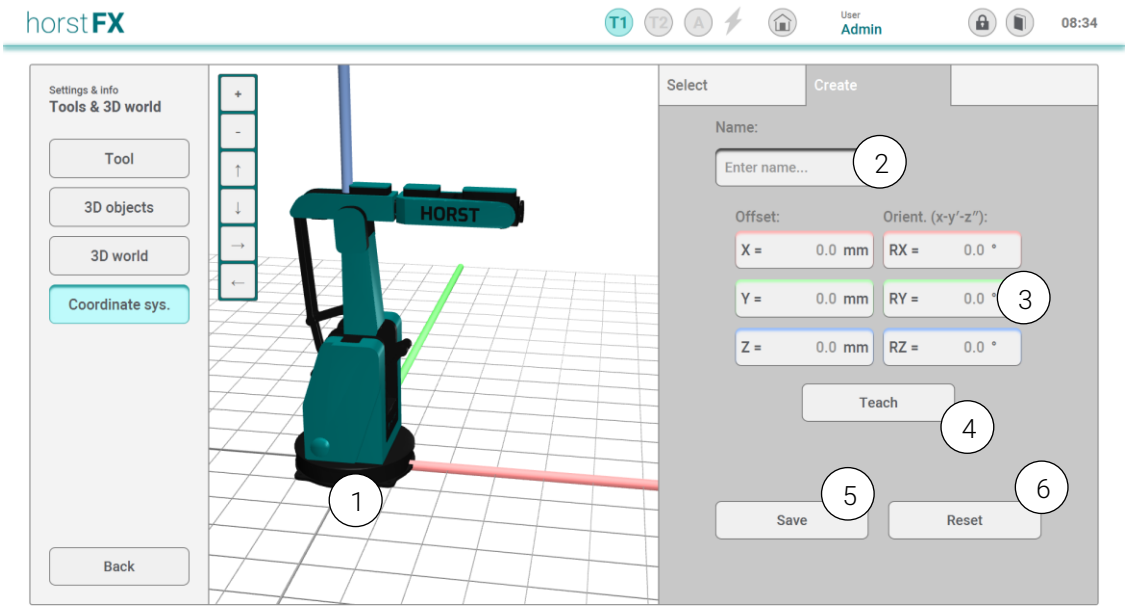


*Fig. 4-23: Menu **Coordinate System** – **Create***

Item	Description
1	Display of the 3D world including the robot model with base coordinate system and preview of the created coordinate system.

Item	Description
2	Enter the name of the coordinate system
3	Display of the values of the origin and orientation of the coordinate system
4	Button **Teach** - opens the following window for defining the coordinate system
5	Button **Save** - saves the coordinate system
6	Button **Reset** - resets values to 0

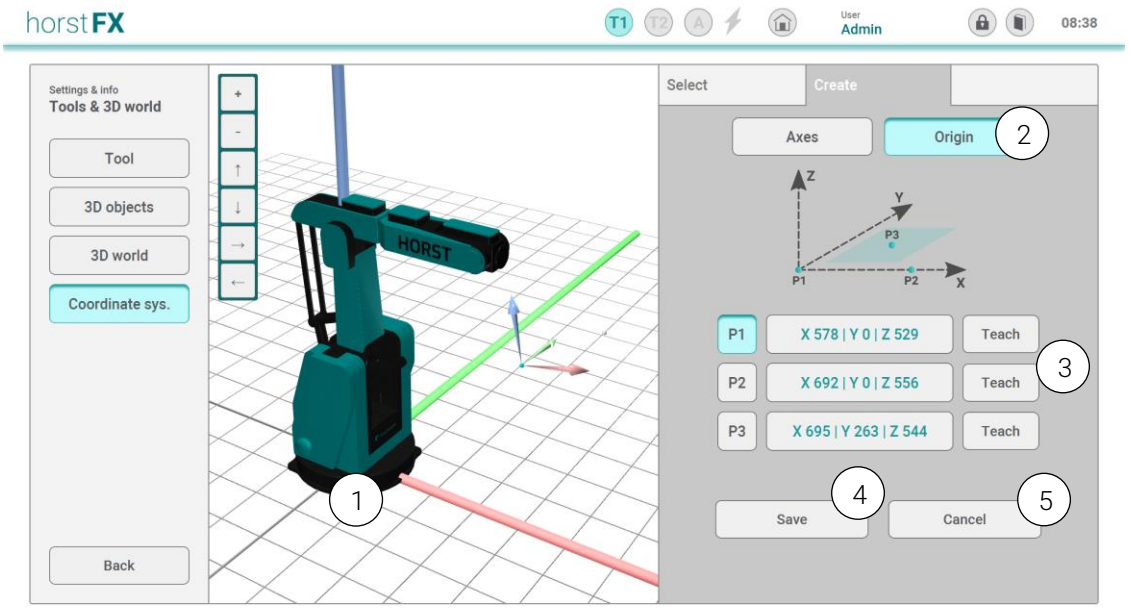


*Fig. 4-24: Menu **Coordinate System – Create – Teach-in***

Item	Description
1	Display of the 3D world including the robot model with base coordinate system and preview of the created coordinate system.
2	Button **Origin** - Coordinate system is created by method 2
3	Display of the coordinates of points 1-3, each with a **Teach** button.
4	Button **Save** - saves taught points, back to previous screen
5	Button **Cancel** - discards taught points, back to previous screen

4.6 Configuration of inputs/outputs menu

In the **Configuration of inputs/outputs** menu, all the settings for the I/O maps and the different inputs and outputs can be set and configured.

The robot system is equipped with two I/O maps *(MAIN* and *Tool)* as standard. Further I/O maps can be connected to horstCONTROL, thus extending the number of inputs and outputs.

Some configuration options are permanently defined in the I/O maps, which results in the corresponding configuration items in sections 4.6.2, 4.6.3, and 4.6.4 being disabled, as these cannot be changed.

More detailed information on some configuration options can be found in the corresponding assembly instructions.

4.6.1 I/O naming

In the **I/O naming** menu, aliases can be assigned for inputs and outputs as well as for the I/O maps themselves. These aliases are used to assign connected appliances/machines to the robot system and to improve readability at all locations in horstFX where inputs and outputs are used or displayed.

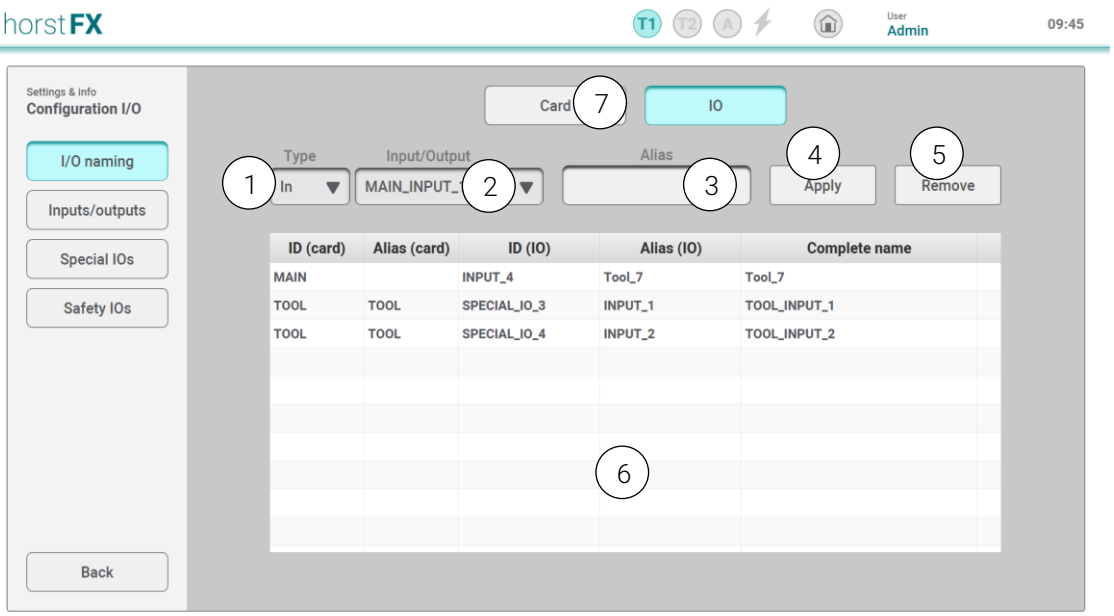


*Fig. 4-25: **Configuration of inputs/outputs – I/O naming** menu*

No.	Description
1	**Type** selection field – for selecting the I/O type The following types can be selected: *On* – inputs *Off* – outputs *Other* – other inputs/outputs (e.g. safety I/O)
2	**Input/output** selection field – for selecting a specific input/output
3	**Alias** input field – for entering an alias
4	**Apply** button – applies the alias entered for the selected input/output
5	**Remove** button – removes the alias of the selected input/output

No.	Description
6	**Inputs/outputs** display – list of all inputs/outputs for which an alias is assigned
7	**Card** button – for switching to the map naming view An alias for a map is assigned in the same way as an alias for inputs/outputs, as described here.

4.6.2 General I/O

In the **General I/O** menu, inputs and outputs can be configured using various setting options.

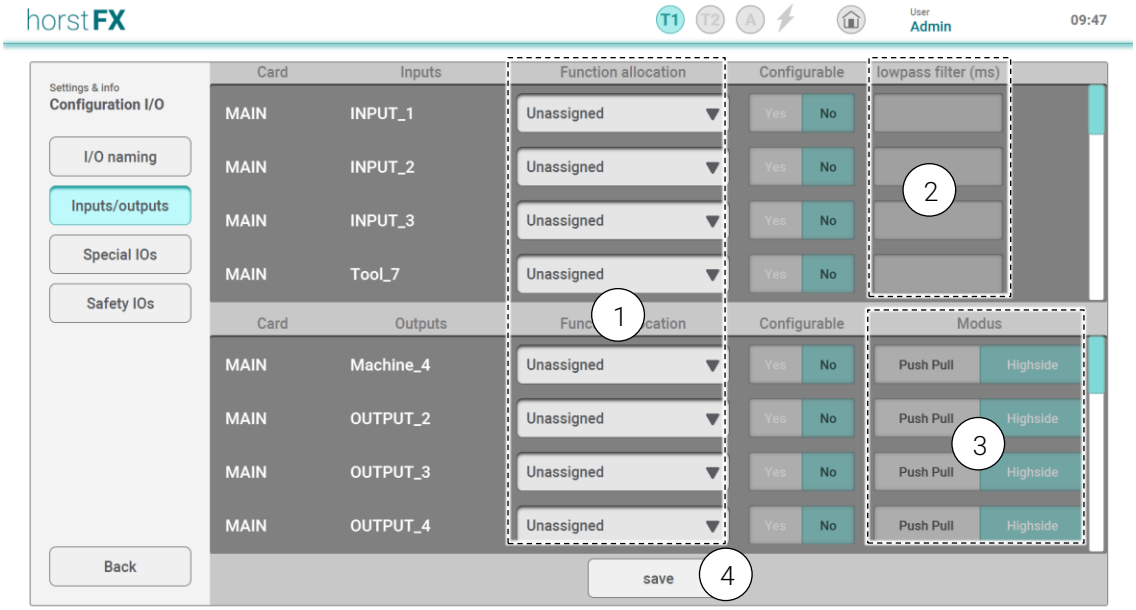


Fig. 4-26: Configuration of inputs/outputs – General I/O menu

No.	Description
1	**Function allocation** selection field – for selecting a function that will be assigned to the corresponding input/output
2	**Low-pass filter** input field – low-pass filter value in ms
3	**Mode** toggle button – for selecting *Push Pull* or *Highside* switching mode
4	**Save** button – saves the changes made to the configurations

4.6.3 Special I/O

In the **Special I/O** menu, the special I/Os can be defined as digital inputs or outputs.

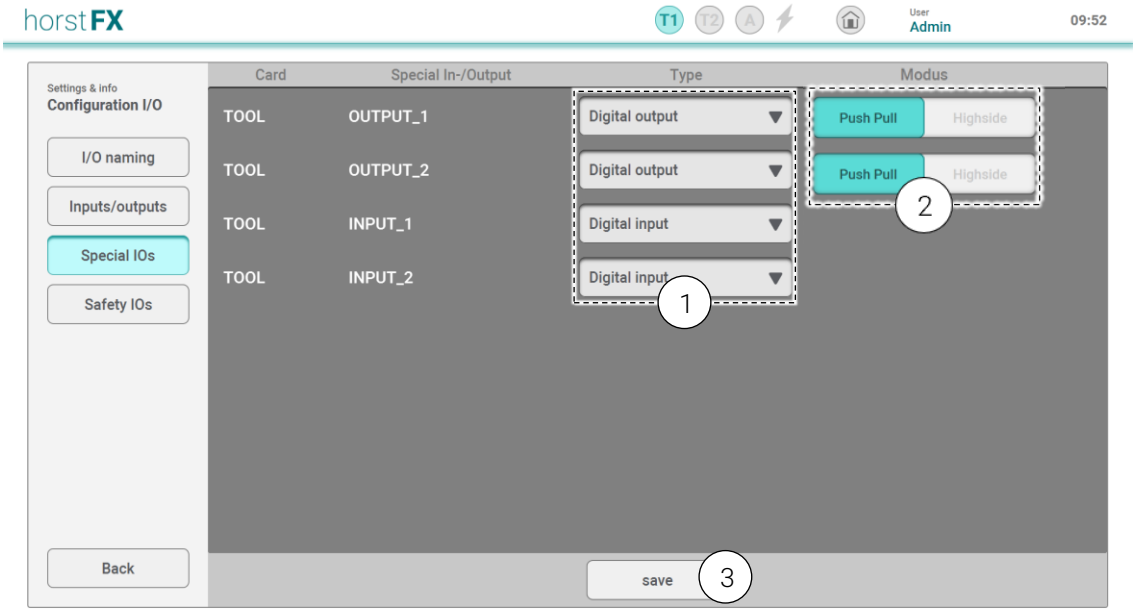


Fig. 4-27: Configuration of inputs/outputs – Special I/O menu

No.	Description
1	**Type** selection field – for selecting between *Digital input* and *Digital output*
2	**Mode** toggle button - selects *Push Pull* or *Highside* switching mode
3	**Save** button – saves the changes made to the configurations

4.6.4 Safety I/O

In the **Safety I/O** menu, the safety I/O can be configured using various setting options.

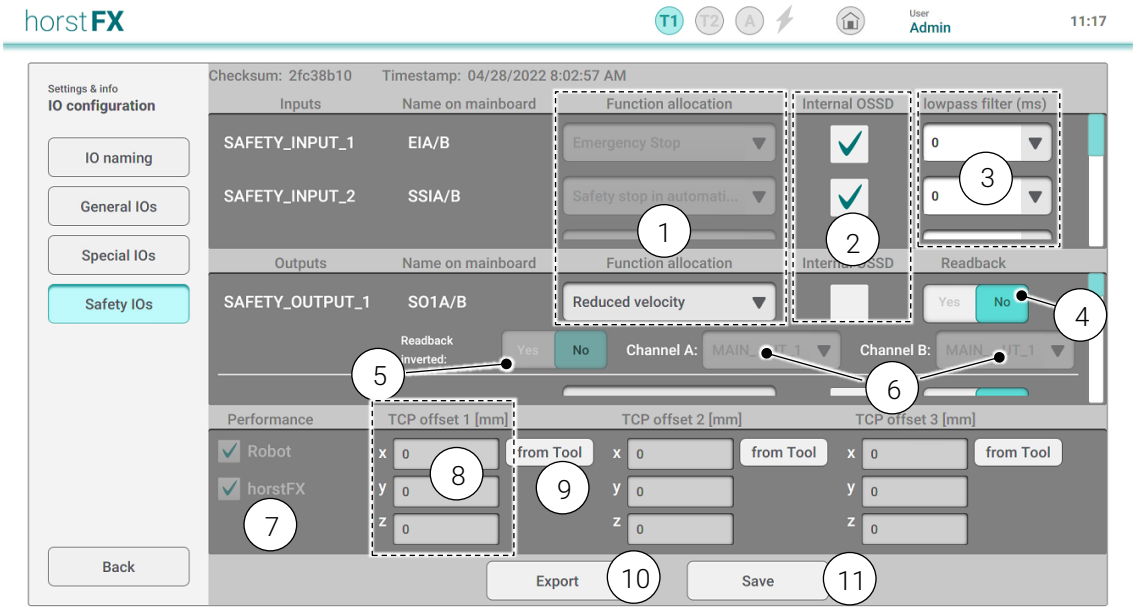


Fig. 4-28: Configuration of inputs/outputs – Safety I/O menu

No.	Description
1	Function allocation selection field – for selecting a function that is assigned to the corresponding safety I/O
2	Internal OSSD toggle button – for selecting whether an internal OSSD is used
3	Low-pass filter input field – low-pass filter value in ms
4	Readback channel toggle button – for selecting whether to activate the readback channel
5	Invert readback channel toggle button – for selecting whether to invert the activated readback channel
6	Channel A/B selection fields – for selecting two inputs for an activated readback channel, which are assigned to channel A and channel B
7	Robot model 'fast' display – shows whether the connected robot is of type 'fast' and whether a robot model of type 'fast' is used in horstFX or not
8	TCP offset input – to monitor the TCP speed, the offsets for X, Y and Z for up to three tools can be specified here
9	From tool button – for selection of a tool from which the offset values for X, Y and Z are taken over
10	Export button – for export of the configuration into a new file
11	Save button – saves the changes made to the configurations

4.7 Log console info point

The Log console info point displays the internally logged program outputs (logs) of horstFX. Normally, these are only relevant for service cases.

4.8 Service/remote access menu

In the Service/remote access menu, remote access can be enabled for a service employee of fruitcore robotics GmbH. An Internet connection is required for this.

WARNING!
During remote access, increased caution is required! For persons not involved in remote access, robot movements may be unpredictable.

- ▶ Never leave the control panel unattended during remote access.
- ▶ Be sure to follow the instructions of the service personnel.

4.9 External interface menu

In the External interface menu, the available interfaces can be activated and configured.

For more information on how to use the external interfaces, see horstcosmos.com. If you have any further questions, please contact the service department of fruitcore robotics GmbH.

4.9.1 XML-RPC

In the XML-RPC menu, the interface can be activated, and the user credentials can be accepted (see also section 8).

4.9.2 Profinet and Modbus

In the Profinet and Modbus menus, the interfaces can be activated respectively. In the Profinet menu, *software error* can also be activated, whereby horstFX reacts to communication errors via Profinet.

In addition, the addressing of the inputs and outputs can be defined in both menus.

5 Free travel

Pressing the **Free travel** button in the main menu selects the **Free travel** menu.

The robot can travel freely through:

- movement of the individual robot axes
- rotation around the axes of the base coordinate system
- linear movement in the direction of the axes of the base coordinate system
- rotation around the axes of the TCP coordinate system (tool coordinate system)
- linear movement in the direction of the axes of the TCP coordinate system (tool coordinate system)

Switching to a different operating mode causes the robot to stop. A warning message appears on the display. The message must be confirmed in order to proceed. The enabling switch must be released during this time.

The robot can only be moved manually in two-handed operation. To move the robot, the enabling switch must always be kept pressed in the center position in operating modes T1 and T2. The desired direction of travel must also be kept pressed on the display. As soon as one of the two conditions is no longer fulfilled, the robot immediately brakes until it comes to a standstill.

In **Simulation** mode, only the movements of the robot model are displayed in the 3D world.

In **Real** mode, the robot performs the movements, and the movements of the robot model are displayed in the 3D world.

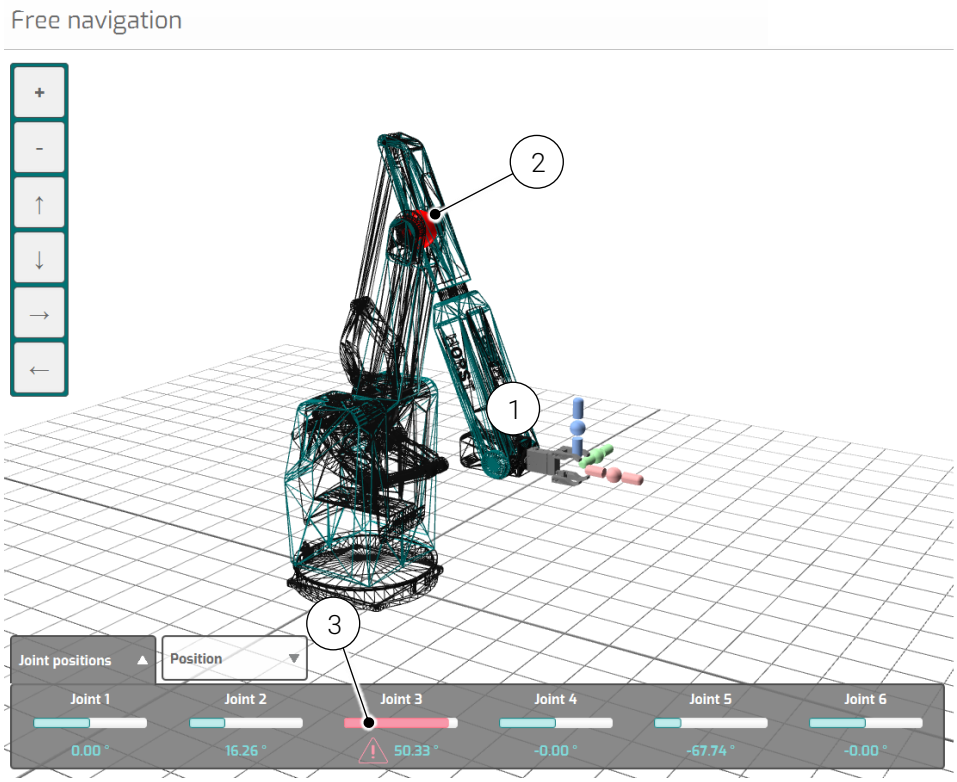


Fig. 5-1: Display when approaching the permissible constraint of an axis

If a value of a robot axis reaches the limit of the permissible axis constraint of the respective robot axis during a movement, then the movement of the robot stops. As soon as a value of a robot axis approaches the limit of the permissible axis constraint, the robot model display on the display changes to a wireframe model (1). A flashing red sphere (2) and a warning symbol (3) indicate which axis has reached the end of its permissible axis constraint or is approaching its limit.

5.1 Movements of the individual robot axes

Pressing the **Axes** button (2) in the **Free travel** menu selects the **Free travel – Robot axes** menu.

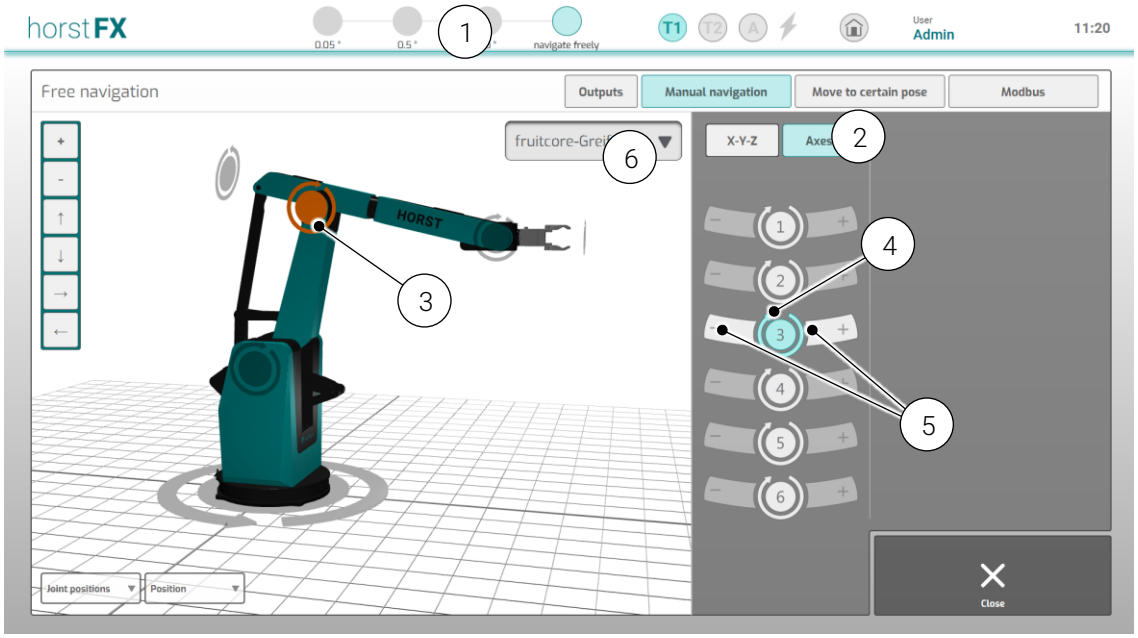


*Fig. 5-2: **Free travel – Robot axes** menu*

For the movement of individual axes of the robot, these can be selected on the display:

- by tapping the axis marking (3) on the robot model

or

- by selecting the button (4) in the axis overview

The selected robot axis is highlighted on the display (here axis 3).

When the **+** or **–** buttons are pressed (5), the robot moves the selected axis. The arrow of the axis marking (3) indicates the **+** direction.

In the menu bar (1), you can set how far the robot travels when the **+** or **–** button is pressed. Move in **Jogging mode** (stepwise in degrees) or continuously in **Free** mode.

Another tool can be selected using the tool change selection field (6). All available tools in horstFX can be selected here. Tool selection uses the same process described in section 4.5.1.1.

5.2 Movements in the coordinate systems

There are two coordinate systems:

- the **base coordinate system**, which does not change its orientation and remains fixed in space.
- the **TCP coordinate system** refers exclusively to the loaded and displayed tool or to the robot flange if no tool has been selected. It changes its orientation depending on the particular pose of the robot.

For a movement in the respective coordinate system, this is selected in the **Free travel** menu.

For movements in a coordinate system, all robot axes move simultaneously.

In the menu bar (Fig. 5-3) (1) you can set how far the robot will move when you hold down the **+** or **–** button. Move in **Jogging mode** (step by step in millimeters) or continuously in **Free** mode.

To illustrate the directions of movement, the selected coordinate system is always shown on the robot model with the corresponding axis colors (Fig. 5-3) (2). Here, the origin of the represented coordinate system is in the TCP of the robot model.

5.2.1 Movements in the base coordinate system

Pressing the **X-Y-Z** button (3) in the **Free travel** menu selects the **Free travel – X-Y-Z** menu. By pressing the **Base** button (4), the base coordinate system is selected and displayed in the robot model (2).

Linear movements (translation) and rotational movements in the base coordinate system can be executed here.

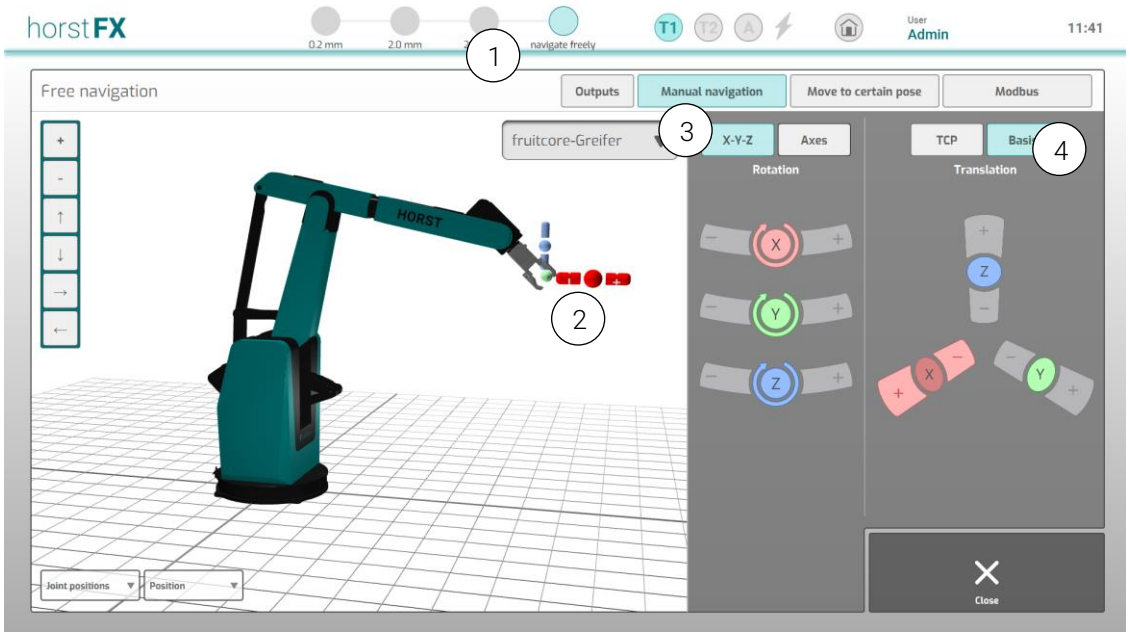


Fig. 5-3: ***Free travel – X-Y-Z – Base*** *menu*

Linear movements in the base coordinate system

For the linear movement (translation) of the robot along the individual coordinate axes, these can be selected on the display by:

- tapping the coordinate axis on the robot model (1)

or

- selecting the coordinate axis in the axis overview (2)

The respective coordinate axis selected is displayed in enlarged form on the robot model and the symbols **+** and **–** (1) appear for the corresponding assignment to the **+** and **–** buttons, which are highlighted in color on the display.

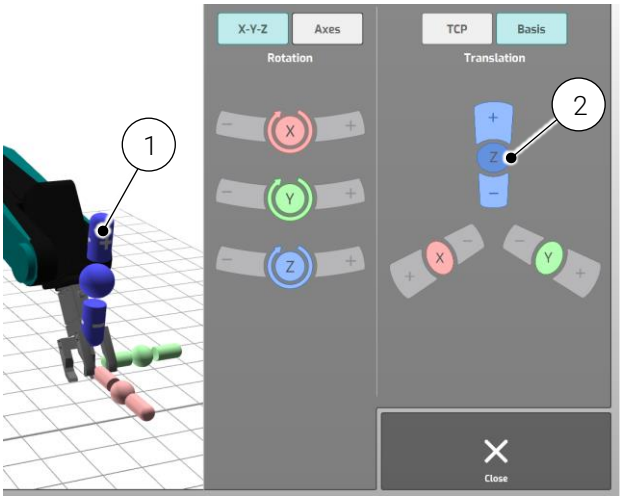


Fig. 5-4: Example – selecting translation on axis Z in the base coordinate system

By pressing the **+** or **–** buttons, the robot moves in linear fashion from the TCP in the desired direction.

Tapping the axis in the robot model again switches the robot to the rotational movement.

Rotational movements in the base coordinate system

For the rotation of the robot around individual coordinate axes, these can be selected on the display by:

- tapping the coordinate axis on the robot model (1)

or

- selecting the coordinate axis in the axis overview (2)

The respective coordinate axis selected is displayed in enlarged form on the robot model and a rotation symbol with an arrow indicating the **+** direction appears. The **+** and **–** buttons are highlighted in color on the display.

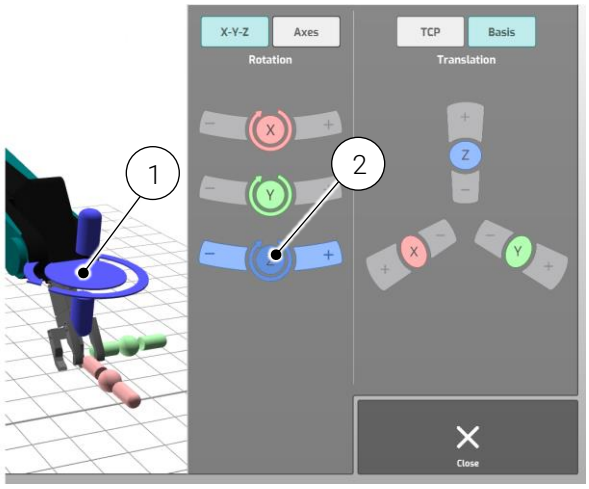


Fig. 5-5: Example – selecting rotation around axis Z in the base coordinate system

By pressing the **+** or **–** buttons, the robot rotates in the TCP around the selected axis in the desired direction.

Tapping the axis in the robot model again switches the robot to linear motion.

5.2.2 Movements in the TCP coordinate system

Pressing the X-Y-Z button (1) in the Free travel menu selects the Free travel – X-Y-Z menu.

By pressing the TCP button (2), the TCP coordinate system is selected and displayed in the robot model.

Linear movements (translation) and rotational movements in the TCP coordinate system can be executed here.

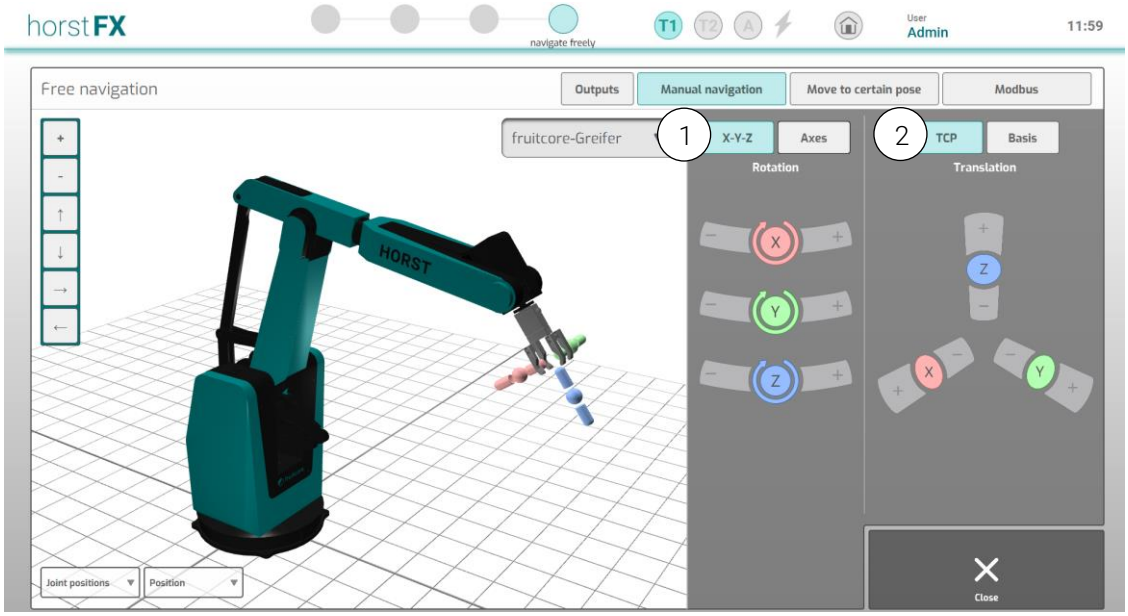


Fig. 5-6: Free travel – X-Y-Z – TCP menu

Linear movements in the TCP coordinate system

For the linear movement (translation) of the robot along the individual coordinate axes, these can be selected on the display by:

- tapping the coordinate axis on the robot model (1)

or

- selecting the coordinate axis in the axis overview (2)

The respective coordinate axis selected is displayed in enlarged form on the robot model and the symbols **+** and **–** (1) appear for the corresponding assignment to the **+** and **–** buttons, which are highlighted in color on the display.

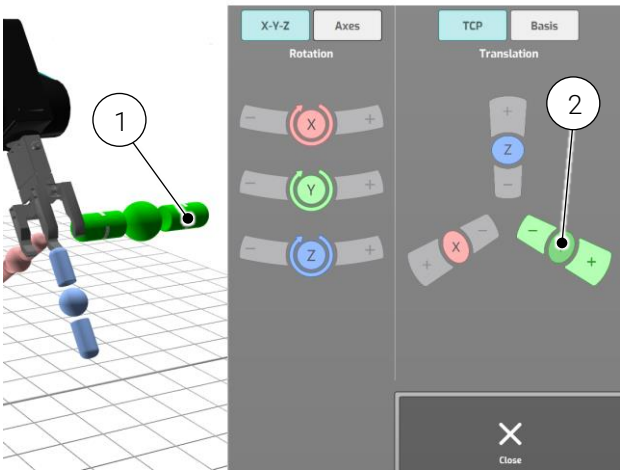


Fig. 5-7: Example – selecting translation around axis Y in the TCP coordinate system

By pressing the **+** or **–** buttons, the robot moves in linear fashion from the TCP in the desired direction.

Tapping the axis in the robot model again switches the robot to the rotational movement.

Rotational movements in the TCP coordinate system

For the rotation of the robot around individual coordinate axes, these can be selected on the display by:

- tapping the coordinate axis on the robot model (1)

or

- selecting the coordinate axis in the axis overview (2)

The respective coordinate axis selected is displayed in enlarged form on the robot model and a rotation symbol appears with an arrow indicating the **+** direction appears. The **+** and **–** buttons are highlighted in color on the display.

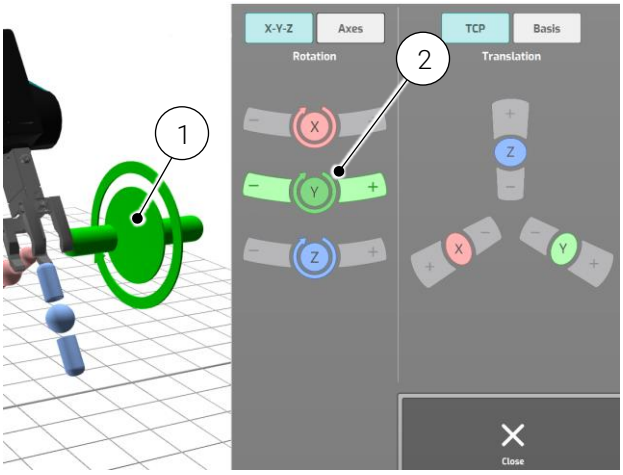


Fig. 5-8: Example – selecting rotation around axis Y in the TCP coordinate system

By pressing the **+** or **–** buttons, the robot rotates in the TCP around the selected axis in the desired direction.

Tapping the axis in the robot model again switches the robot to linear motion.

5.3 Free travel – Outputs

Pressing the **Outputs** button in the **Free travel** menu selects the **Free travel (outputs)** menu.

Here you can both view the current status of all inputs and outputs and change the status of the outputs. The status of the inputs is signaled by OFF or ON. Each output can be toggled directly via the corresponding toggle button 0/1. For example, grippers can be opened or closed manually.

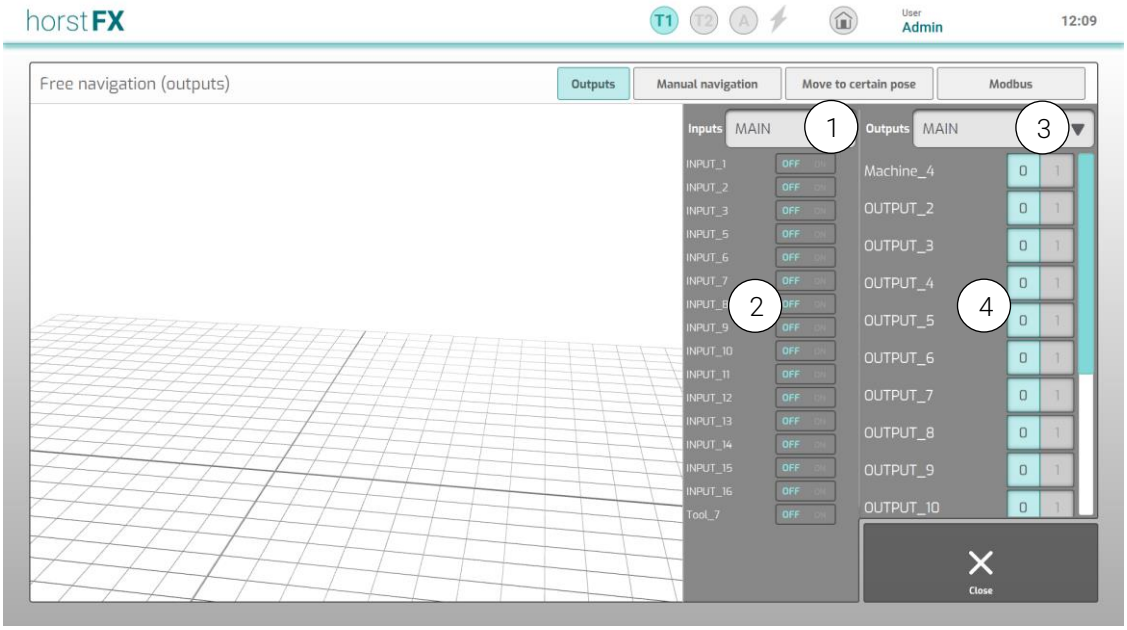


*Fig. 5-9: **Free travel (outputs)** menu*

No.	Description
1	**Inputs** selection field – for selecting the I/O map whose inputs will be displayed
2	**Inputs status** display – displays the current statuses of the inputs
3	**Outputs** selection field – for selecting the I/O map whose outputs will be displayed
4	**Outputs** display/toggle buttons – displays the current statuses of the outputs and the possibility to switch the outputs

5.4 Free travel – Move to specific pose

Pressing the **Move to specific pose** button in the **Free travel** menu selects the **Free travel (move to specific pose)** menu.

In this menu, the robot can be moved to a specific pose by either specifying the coordinates and/or orientation, or by specifying the axis values.

The pose, which is calculated using the entered values, is visualized by the wireframe model in the 3D world (see Fig. 5-12). If no wireframe is visible, the pose is not possible. Each change of one of the values in the input fields triggers an update of the wireframe model in the 3D world.

In the **Free travel (move to specific pose) – X-Y-Z** menu (1), the coordinates and orientation always refer to the TCP in the global coordinate system. The orientation in the form of Euler angle values also refers to the axes of the global coordinate system.

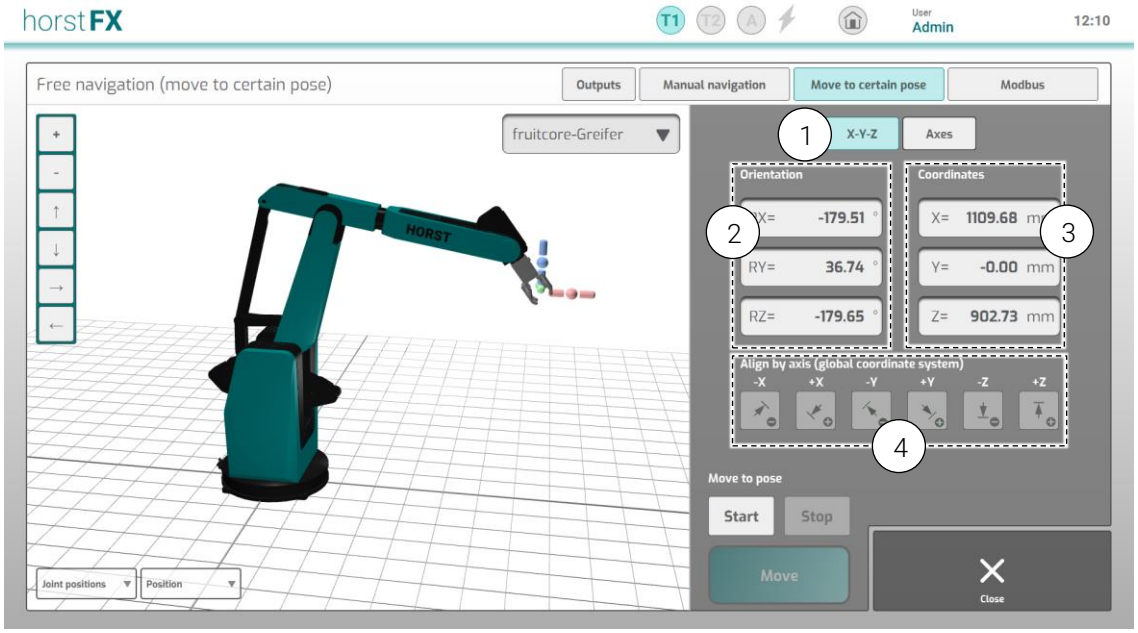


*Fig. 5-10: **Free travel (move to specific pose) – X-Y-Z** menu*

No.	Description
1	**X-Y-Z** button – displays the input fields for coordinates and orientation
2	**Orientation** input fields – for entering RX, RY, and RZ values to determine the orientation
3	**Coordinates** input fields – for entering the X, Y, and Z coordinates
4	**Align by axis** buttons – for selecting the orientation in positive or negative direction of one of the global coordinate system axes

In the **Free travel (move to specific pose) – Axes** menu (1), the pose is determined on the basis of the six axis values.

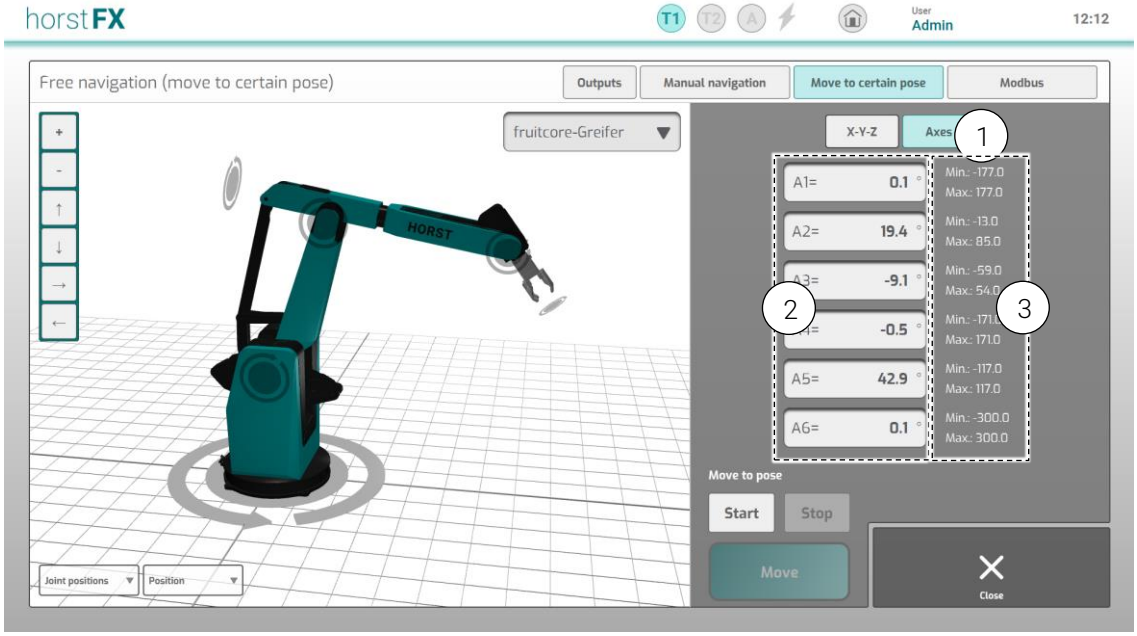


Fig. 5-11: ***Free travel (move to specific pose) – Axes*** *menu*

No.	Description
1	**Axes** button – displays the input fields for the axis values
2	**Axis values** input fields – for entering the axis values A1, A2, A3, A4, A5, and A6
3	**Axis constraints** display – displays the currently valid/set axis constraints for the respective axis

After entering the values, the pose can be moved to if it is a valid pose. If not, a corresponding message appears on the display as soon as an attempt is made to move to the pose. Likewise, a corresponding note appears as soon as the pose is reached, and the movement is completed.

A calculated path to the entered pose can be discarded by tapping the **Stop** button (2), for example, to adjust values and thus also adjust the pose.

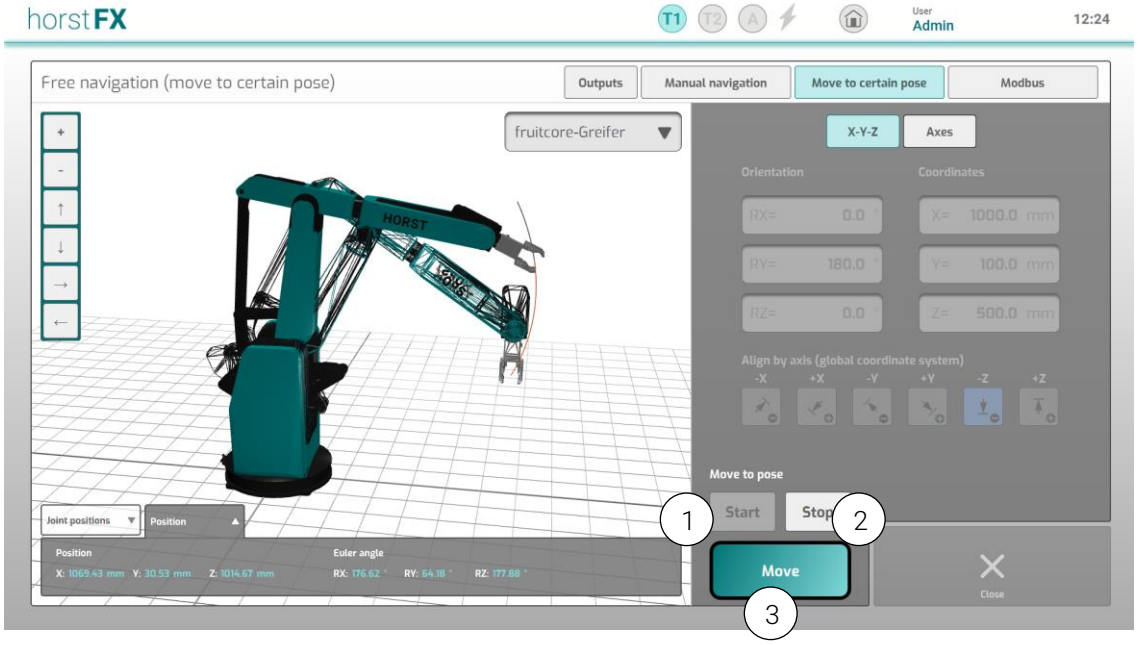


*Fig. 5-12: **Free travel (move to specific pose) – X-Y-Z** (move to pose) menu*

No.	Description
1	**Start** button – path to the target pose is calculated and visualized by orange spheres and the **Move** button (3) is also activated
2	**Stop** button – the current path to the target pose is discarded and the **Move** button (3) is also deactivated
3	**Move** button – by tapping and holding this button, the robot moves along the calculated path

5.5 Free travel – Register

Pressing the **Register** button (1) in the **Free travel** menu selects the **Free travel – Register** menu.

The entries listed here show only a certain selection of the most important registers that can be addressed via Profinet and Modbus.

The entries are only displayed in this menu and cannot be changed.

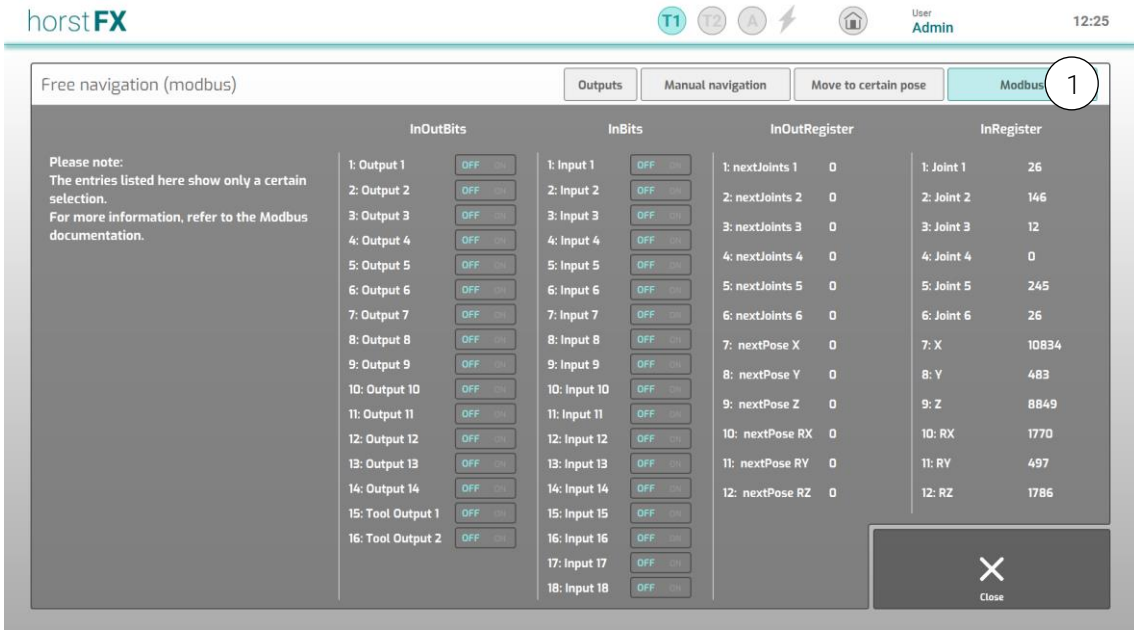


*Fig. 5-13: **Free travel (register)** menu*

The corresponding check box (2) provides the option to display additional internal registers.

For a complete overview of the register assignment and further information on the use of registers via Profinet or Modbus, please see **horstcosmos.com**. If you have any further questions, please contact the service department of fruitcore robotics GmbH.

6 Programs

The following section explains how to create programs for the automation of the robot's movements.

The robot is programmed in Teaching mode.

WARNING!
Change of the danger zone due to add-on parts and workpieces

▶ Note that the range of the robot and thus the danger zone change with add-on parts and workpieces.

Before programming the robot, installation must be carried out during initialization (see section 2).

Depending on whether Real or Simulation mode is selected, during programming the movements are executed by the robot or only by the robot model in the 3D world.

The robot can only be moved manually in two-handed operation. To move the robot, the enabling switch must always be kept pressed in the center position in operating modes T1 and T2. The desired direction of travel must also be kept pressed on the display. As soon as one of the two conditions is no longer fulfilled, the robot will brake until it comes to a standstill.

Switching to a different operating mode causes the robot to stop. A warning message appears on the display. The message must be confirmed in order to proceed. The enabling switch must be released during this time.

In Simulation mode, only the movements of the robot model are displayed in the 3D world.

In Real mode, the robot performs the movements, and the movements of the robot model are displayed in the 3D world.

6.1 New program

When the New program button in the main menu is pressed, the pop-up window for creating a new program appears.

Tapping in the input field (1) opens the on-screen keyboard for text input. The program name can be entered.

Clicking the OK button (2) creates a new program and the programming view appears on the display (see section 6.3).

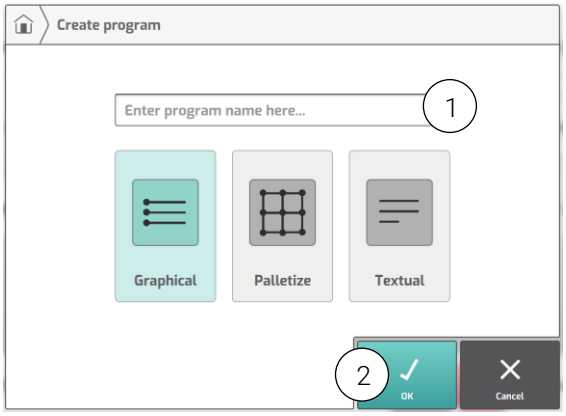


Fig. 6-1: Create program

If the option **Graphic** is selected when creating a new program, only the action **Start/configuration** (1) is visible in the program tree, together with a placeholder (2) indicating that new actions can be added at this point.

If the **Textual** option is selected, the programming view will be adapted to the textual programming. The main difference here is that the program tree is replaced by a text editor with additional programming elements (see section 6.5).

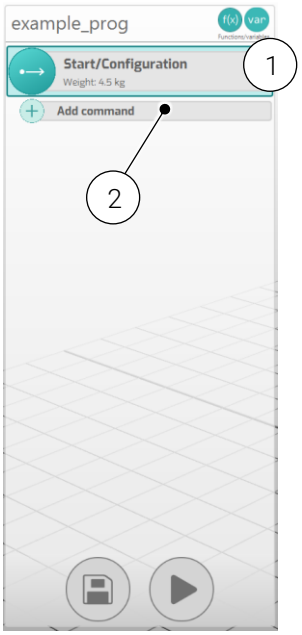


Fig. 6-2: New graphic program

6.2 Load program

Pressing the **Load program** button in the main menu displays the file manager.

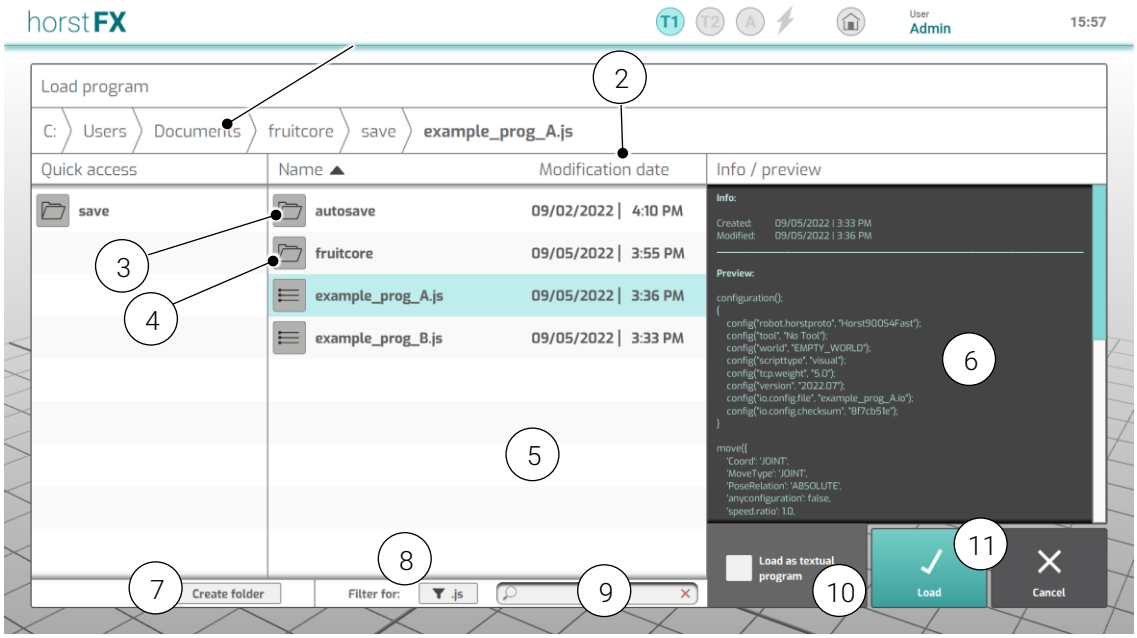


Fig. 6-3: Load program – file manager

The individual directories (1) in the path bar can be used for navigation by tapping on them. The content of a directory is listed in the middle window (5). If a folder entry (3) is tapped, its contents are listed immediately. After tapping on a program entry (4), a preview (6) of the corresponding program appears.

Tapping and briefly holding an entry (3+4) opens an editing menu. This editing menu enables the selected program/folder to be copied, cut out, deleted, or renamed.

The entries can be sorted alphabetically or chronologically (2). In addition, the **.js** button (8) can be used to filter for all programs with the .js file type. The search bar (9) can be used to search for a

program by entering text. The **Create folder** button (7) can be used to create a new folder in the currently selected directory.

If a program entry (4) is selected, the **Load** button (11) is activated. After tapping this button, the program view is changed, and the selected program is loaded and displayed in the program tree. If the selected program entry is a graphical program, the "**Load as textual program**" check box (10) is additionally displayed. If this option is activated, the program is loaded as a textual program (see Section 6.5).

6.3 Create/edit program

Creating a new program, loading a saved program, or editing a loaded program makes the programming view appear.

When creating/editing a program, the program flow is determined by the use of the predefined actions.

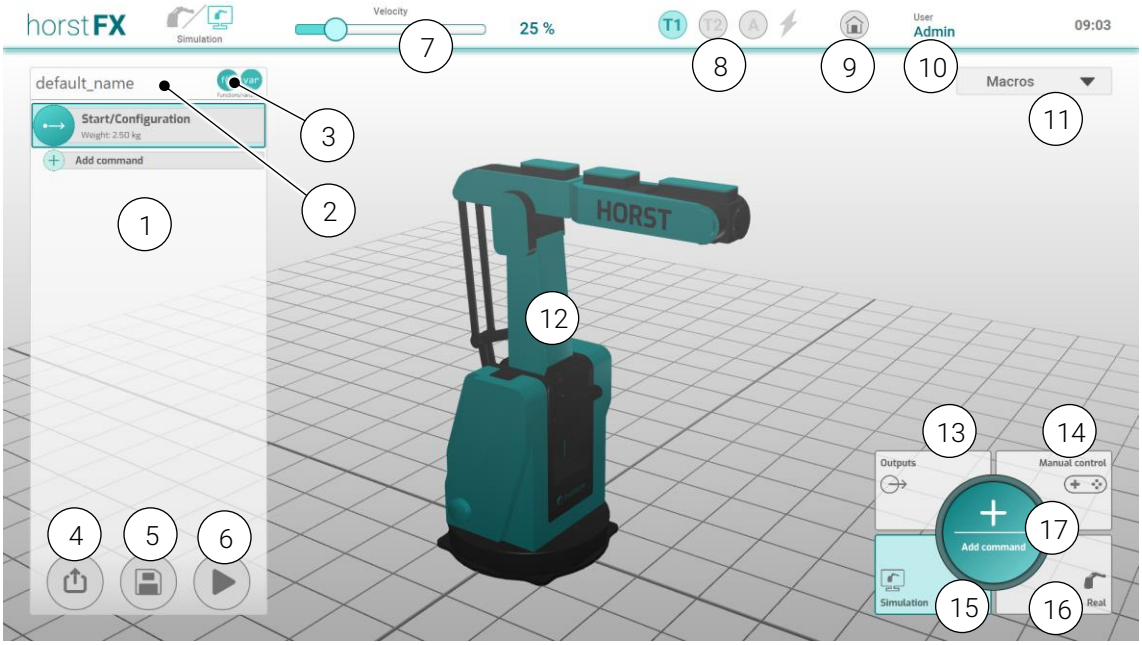


Fig. 6-4: Programming view

No.	Description
1	**Program tree** display – displays the program with all its actions/program blocks
2	**Program name** of the current program
3	**f(x) & var (functions/variables)** button – lists existing functions/variables and the possibility to create new functions/variables
4	**Export** button – via a file manager that opens, the program can be exported as a ZIP file together with all its existing configurations
5	**Save** button – the program can be saved via a pop-up window that opens (An automatic save function also saves the program every 2 minutes in an autosave file)
6	▶ (Execute) button – the **Program Execution section** appears on the screen (see 6.4 section)
7	**Speed controller** – sets the speed at which a program is executed

No.	Description
8	**Operating mode** display – displays the currently selected operating mode **T1** – Teaching mode – manual operation at reduced speed **T2** – Teaching mode – manual operation at high speed **A** – automatic mode ↯ – displays warning and error message The symbol flashes red for unacknowledged messages: emergency stop, safety stop, and system error.
9	**Main menu** button – for returning to the **main menu**
10	Display of the current (logged-in) user role
11	**Macros** button – show/hide the **Macro buttons** menu (see section 6.6.3)
12	**Robot model** view – displays the current pose of the robot
13	**Outputs** button – opens the **Manual control (outputs)** menu (see section 6.3.27)
14	**Manual control** button – opens the **Manual control** menu (see section 6.3.27)
15	**Simulation** button – selects **Simulation** mode In **Simulation** mode, only the movements of the robot model are displayed in the 3D world.
16	**Real** button – selects **Real** mode In **Real** mode, the robot performs the movements, and the movements of the robot model are displayed in the 3D world.
17	**Add command** button – opens the **action selection area**, through which an action is selected (see Fig. 6-5), which will be added to the program tree in the form of a program block

If a new program is loaded, then initially only the program block *Start/configuration* together with the placeholder *Add action*, which indicates that new actions can be added at this point, appear in the program tree.

The *Add action* placeholders are special program blocks. They are set automatically for a new program and for new grouped program blocks. As soon as an action is added to the new program or a grouped program block, the placeholder is replaced by the program block belonging to the added action. If the last program block within a grouped program block is deleted or moved, a placeholder is automatically set again.

By tapping a placeholder or the **Add command** button (see Fig. 6-4) (15) in the programming view, the **action selection area** appears. For each action that is selected, a corresponding action window opens with various configuration and selection options for the action. In addition, a corresponding program block is added to the program tree, always under the last selected program block or instead of a placeholder.

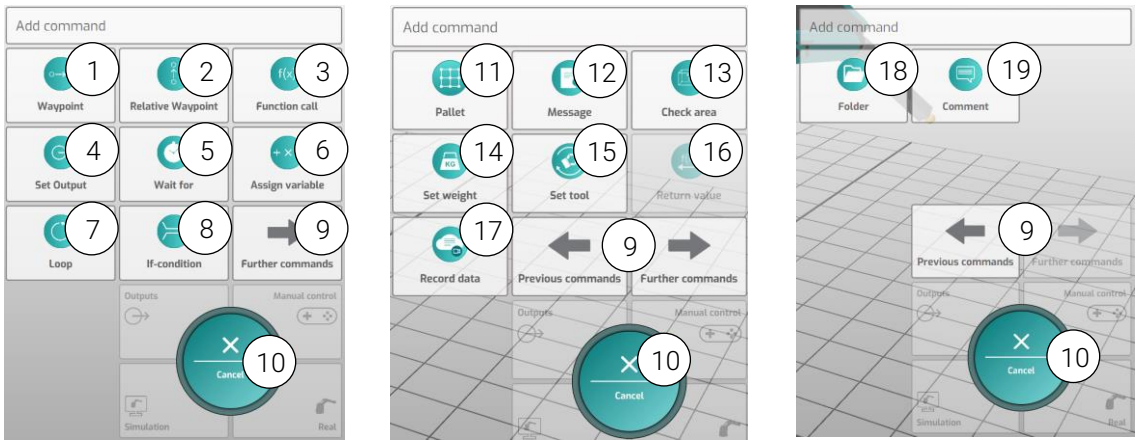


*Fig. 6-5: Programming view – **Action selection area***

No.	Description	Reference
1	**Waypoint** button	Section 6.3.1.2
2	**Relative Waypoint** button	Section 6.3.3
3	**Function call** button	Section 6.3.4
4	**Set output** button	Section 6.3.5
5	**Wait for** button	Section 6.3.6
6	**Assign variable value** button	Section 6.3.7
7	**Loop** button	Section 6.3.7.1
8	**If-condition** button	Section 6.3.9
9	**Previous/further commands** – switch to previous/next selection page	
10	**Cancel** button – **Close action selection area** button	
11	**Pallet** button	Section 6.3.10
12	**Message** button	Section 6.3.17
13	**Check area** button	Section 6.3.26
14	**Set weight** button	Section 6.3.11
15	**Set tool** button	Section 6.3.16
16	**Return value** button	Section 6.3.17
17	**Record data** button	Section 6.3.14
18	**Folder** button	Section 6.3.24
19	**Comment** button	

Each action is represented in the program tree by a program block, which displays the most important information about the respective action.

Actions already added to a program can be edited via their respective program block (see section 6.3.28).

6.3.1 Start/configuration action

The **Start/configuration action** is the first program block in every program. This is fixed in the program tree and can be neither deleted nor moved. Since no other such actions are allowed, there is no corresponding button in the **action selection area**. For this action, the corresponding action window can only be accessed via the editing menu (see section 6.3.28).

6.3.1.1 Start/configuration action (Tools & Coord.)

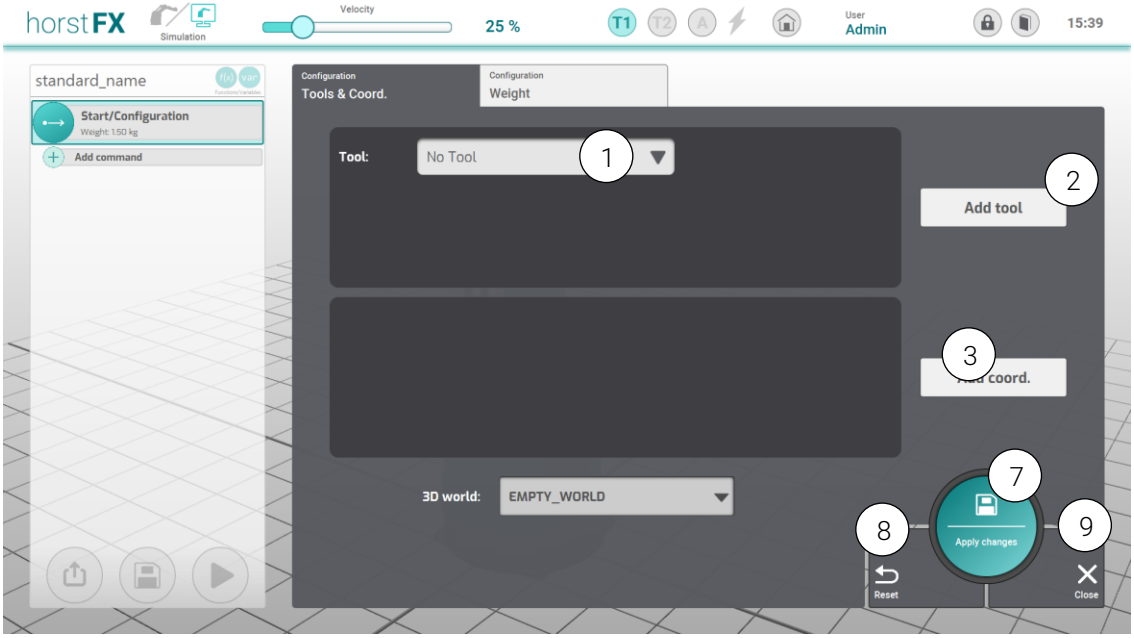


Fig. 6-6: Action window – Start/configuration (Tools & Coord.)

No.	Description
1	**Tool** selection field – for selecting/changing the tool for the current program
2	**Add tool** button – for adding more tools (see section 6.8)
3	**Info on weight** display – see important note at the end of this section
4	**Weight at TCP** input field – for entering the weight currently set at the TCP
5	**Weight change** buttons – for changing the set weight at the TCP in increments of 0.5 kg or 1.0 kg (within the valid range)
6	**3D world** selection field – for selecting the 3D world that will be displayed for the current program
7	**Apply changes** button – the changes to all configuration and selection options in the action window are applied, and the action window is closed
8	**Reset** button – resets all configuration and selection options in the action window to their default values
9	**Close** button – closes the action window

The selected tool (1) is displayed on the robot model. The TCP of this tool is relevant for all waypoints (destinations) of the program.

To prevent the robot from "jerking" during the execution of movements, the parameter for the weight present on the TCP is set ((3)-(5)). The weight configured here applies program-wide to all Waypoint and Relative waypoint actions unless it is overridden by a Set weight action (see section 6.3.15) or a separate weight is configured in a Waypoint or Relative waypoint action (see sections 6.3.1.2 and 6.3.3).

ATTENTION!

The maximum weight on the TCP should be determined as accurately as possible and the value set in the Start/configuration action window. The highest possible value is set by default.

It should be noted that if the weight is too low, the robot will not be operated within the specifications. This can lead to increased mechanical wear and possibly to step losses and thus damage to the system.

Failure to operate the robot within the specifications will void the warranty.

The load parameters apply in accordance with VDI Guideline 2861. It should be noted that the load and the associated load center of gravity must not exceed the permissible value for the nominal load, the nominal torque, and the nominal mass moment of inertia.

6.3.1.2 Start/configuration action (Weight)

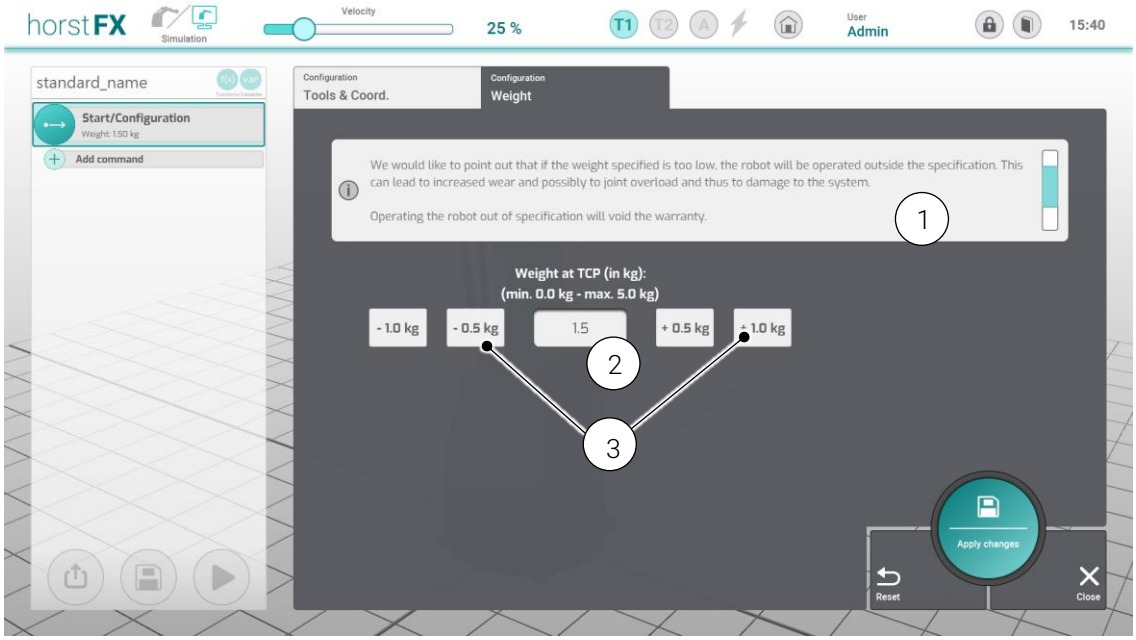


Fig. 6-7: Action window – Start/configuration (Weight)

No.	Description
1	Info on weight display – see important note at the end of this section
2	Weight at TCP input field – for entering the weight currently set at the TCP
3	Weight change buttons – for changing the set weight at the TCP in increments of 0.5 kg or 1.0 kg (within the valid range)

6.3.2 Waypoint action

Selecting the **Waypoint action** in the **action selection area** will bring up the corresponding action window. A new *Waypoint* program block is created in the program tree.

When configuring the **Waypoint action**, basic settings such as speed, type of movement, tolerances, and destination of the robot are defined. The destination defines the desired pose (position) of the robot. There are also other configuration options, which are discussed in more detail in the following subsections.

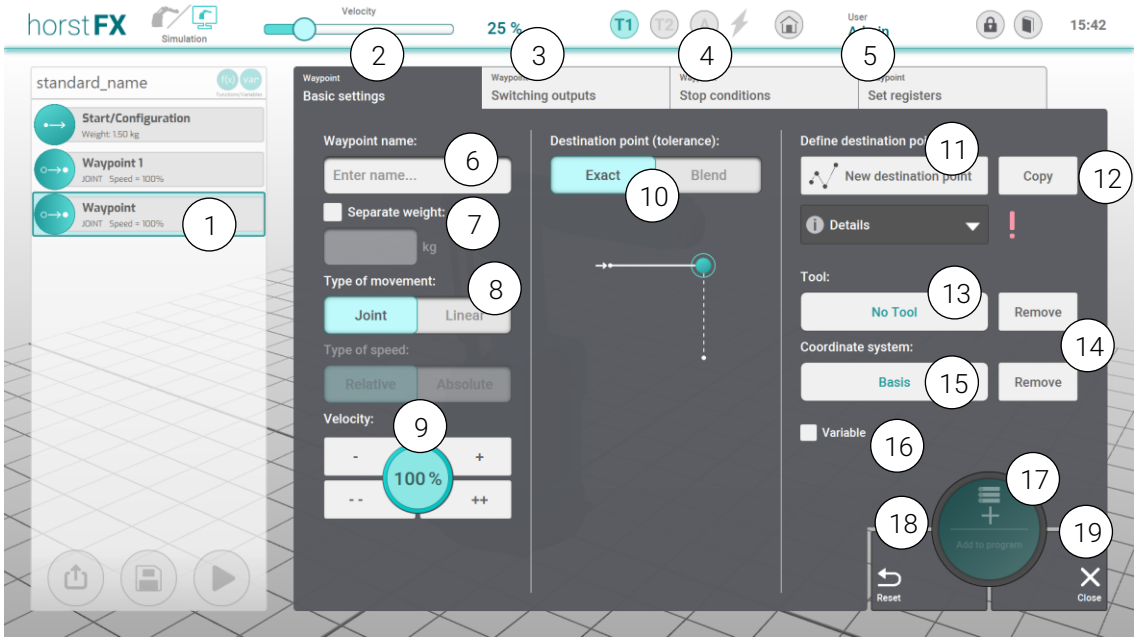


Fig. 6-8: Action window – Waypoint – Basic settings

No.	Description
1	**Program tree** display – program tree with newly created *Waypoint* program block
2	**Waypoint – basic settings** menu – see points (6) - (19)
3	**Waypoint – Switch outputs** menu – see section 6.3.2.4
4	**Waypoint – Stop conditions** menu – see section 6.3.2.5
5	**Waypoint – Set registers** menu – see section 6.3.2.6
6	**Waypoint name** input field – for naming the waypoint This name appears after the addition (12) in the program tree in the corresponding program block. If no name is entered, a default name is assigned, consisting of "Waypoint" and a number. The number is continuously increased.
7	**Separate weight** checkbox/input field – for setting a separate weight The input field is active only if the check box is selected.
8	**Joint/Linear** toggle button – type of movement with which the waypoint is approached **Linear** – the TCP travels along a straight line to the specified waypoint. **Joint** – the TCP travels along the fastest path to the specified waypoint. The movement is undefined and usually takes the shape of an arc. This is the fastest type of movement.

No.	Description
9	**Velocity** buttons – for selecting the speed at which the waypoint is approached (**-/+** in increments of 1, **- -/++** in increments of 10)
10	**Approach/Approximate positioning** toggle button – for selecting the destination approach tolerance (see section 6.3.2.1)
11	**New destination point** button – the **Define destination** menu is opened (see section 6.3.2.2)
12	**Copy** button – for selecting a destination that already exists This button is visible only if there is already a Waypoint action in the program tree. When tapped, a pop-up window appears with a selection field. There, a copy of the destination definition to be selected can be taken from all destinations of the waypoint actions that have already been added to the program tree.
13	**Tool** display – displays the tool with which this waypoint is to be approached Only relevant for programs with more than one tool (see section 6.8).
14	**Remove** button – removes the tool saved in the waypoint Only visible for programs with more than one tool (see section 6.8).
15	**Coordinate system** selection field – for selecting the coordinate system in which waypoin is defined
16	**Variable** check box – if activated: selection of a variable of type target position as destination point **Position & orientation** display – pop-up display with information on the position and orientation of the destination
17	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed The **Add to program** button is activated only after a destination is defined.
18	**Reset** button – resets all configuration and selection options in the action window to their default values
19	**Close** button – closes the action window

6.3.2.1 Approximate positioning (destination tolerances)

During program execution, the effect of approximate positioning is a harmonious blending of two or more contiguous movements (waypoints), i.e. without stopping at the defined destination(s).

This is useful, for example, in use cases where obstacles are to be avoided by defining certain waypoints, but these waypoints do not have to be reached exactly. Here, bypassing these waypoints without stopping in between can save both time and energy and thus reduce cycle times.

Approach – if Approach is selected on the toggle button, the waypoint will be approached exactly as the destination was defined. Only when the destination has been reached exactly does the program execution continue with the next action.

Approximate positioning – if Approximate positioning is selected at the toggle button, the waypoint will not be approached exactly, but will be bypassed. The bypass can be configured using two parameters, Cartesian translation (approximate positioning radius) and orientation:

- Cartesian translation (approximate positioning radius in millimeters)

The translation tolerance or the approximate positioning radius r defines the position difference between the TCP and the destination at which approximate positioning begins. As soon as the position of the TCP undershoots this difference when approaching the destination, the transition to the next movement (next waypoint) is initiated. It should be noted here that in order to avoid mutually overlapping approximate positioning radii, excessively large radii are limited.

- Orientation in degrees

  The orientation tolerance t defines the difference between the orientation of the TCP and the orientation of the destination at which approximate positioning starts. As soon as the orientation of the TCP undershoots this difference when approaching the destination, the transition to the next movement is initiated.

Use of the parameters:
Approximate positioning is initiated as soon as both of the tolerances described above are undershot when approaching the waypoint.
It is assumed that the approximate positioning should be controlled by default using translation tolerance/approximate positioning radius. The default setting is 0.0 mm. With this setting, the waypoint is approached exactly, but the next movement (next waypoint) is initiated without stopping the movement at the destination. To achieve an earlier transition to the next movement, the radius must be increased.

If the subsequent action on a Waypoint action with approximate positioning is not another Waypoint action or a Relative waypoint action, the configured approximate positioning has no effect.

6.3.2.2 Speed

Depending on the selection of the *type of movement*, different speed settings can be made. If **Joint** is selected as the type of movement, the speed can be set as a percentage using the minus and plus buttons. With the **Linear** movement type, it is also possible to enter the absolute speed and acceleration.

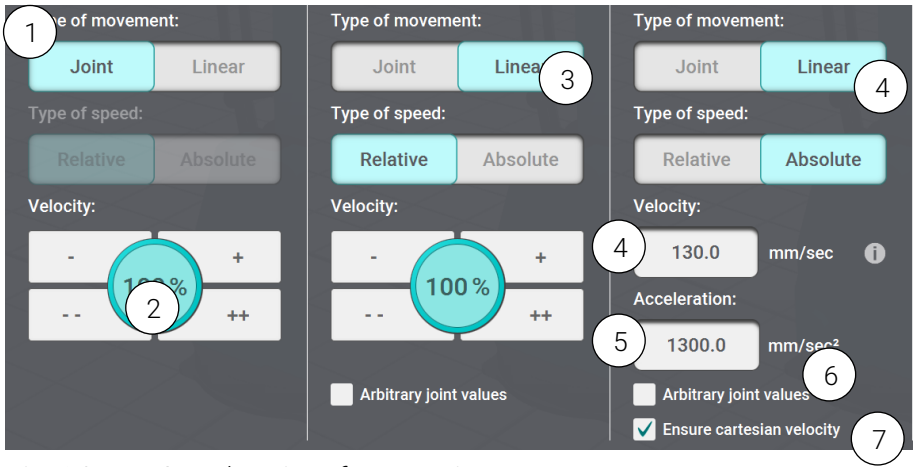


Fig. 6-9: Speed settings for waypoints

Item	Description
1	Toggle button **Joint/Linear** - sets type of movement
2	Relative speed
3	Toggle button **relative/absolute** - sets type of movement

Item	Description
4	**Speed** input
5	Input **acceleration**
6	Check box **Any axis values** - allows other target pose, at other axis values
7	Check box **Keep speed** - executes movement only if the speed can be kept.

6.3.2.3 Define destination

By pressing the button **New destination** in the action window, the **Define destination** menu appears.

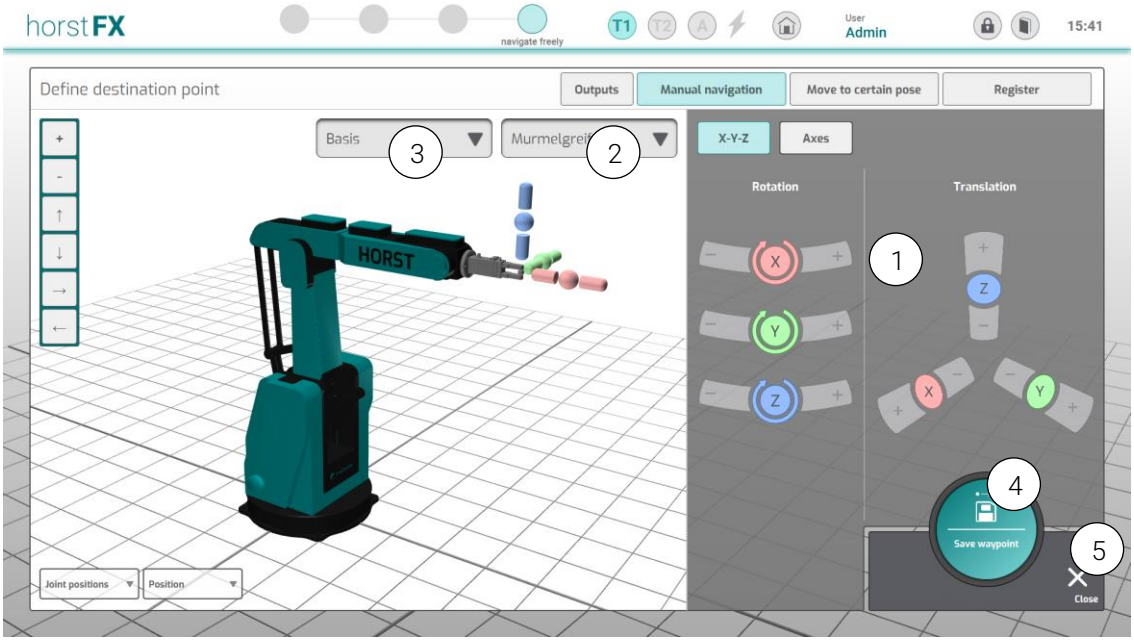


*Fig. 6-10: Waypoint – basic settings – **Define destination** menu*

No.	Description
1	Menus and controls for controlling the robot. Operation is as described in section 4.9.1.
2	**Tool** selection field – for selecting and setting the tool to which the definition of the destination or the control system in this menu should refer Only relevant for programs with more than one tool (see section 6.8).
3	**Coordinate system** selection field – for selecting and setting the coordinate system to which the definition of the destination or the control system in this menu should refer
4	**Save waypoint** button – saves the controlled pose of the robot as a destination and closes the **Define destination** menu
5	**Close** button – closes the **Define destination** menu without saving a destination

6.3.2.4 Waypoint – Switch outputs menu

In the **Waypoint – Switch outputs** menu, it is possible to switch outputs while approaching a waypoint. This means that outputs can be switched at a specific time during the movement of the robot.

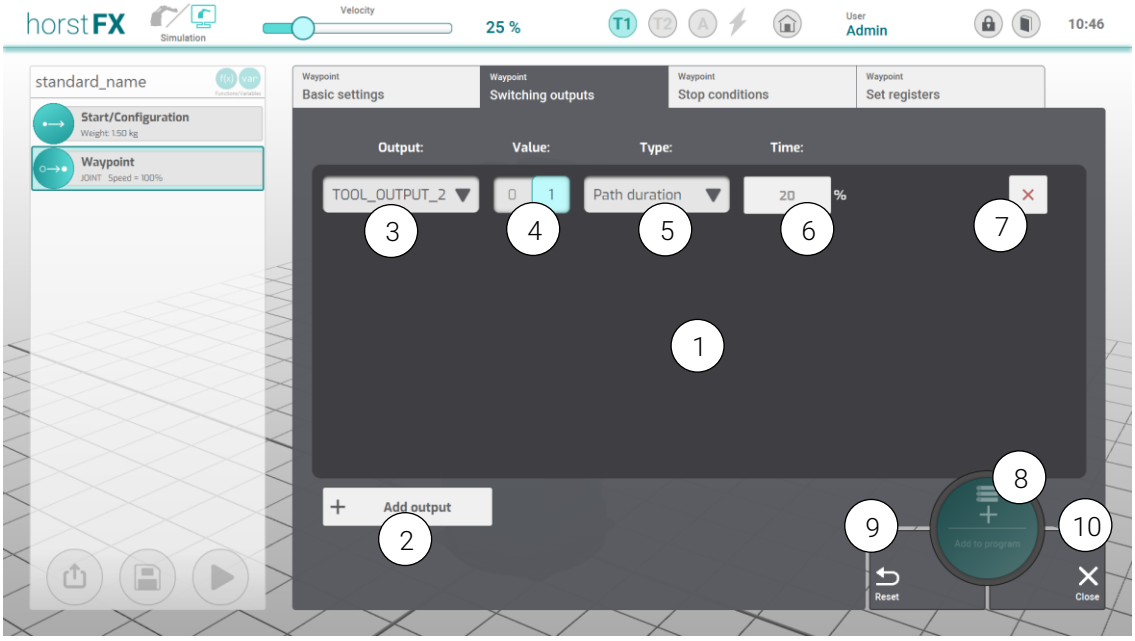


Fig. 6-11: Action window – Waypoint – Switch outputs

No.	Description
1	List – displays all entries (lines) that have been added
2	**Add output** button – adds a new entry (line) to the list (1)
3	**Output** selection field – selects the output to be switched
4	**Value** toggle button – sets the value that the output will adopt after switching
5	**Type** selection field – selects the type of the switch **Path duration** type (in percent) – defines after what percentage of time the output is switched. The time refers to the complete movement including acceleration and deceleration of the robot. **Time delay** type (in milliseconds) – defines after how many milliseconds from the beginning of the waypoint approach the output is switched. If the value is higher than the complete movement itself takes, at the end of the movement the system waits until the time value is reached, and the output is switched.
6	**Time** input field – the value in percent or milliseconds
7	**Delete** button – deletes the corresponding entry (line)
8	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed The **Add to program** button is enabled only if a destination has been defined.
9	**Reset** button – resets all configuration and selection options in the action window to their default values
10	**Close** button – closes the action window

6.3.2.5 Waypoint – Stop conditions menu

The **Waypoint – Stop conditions** menu can be used to pause or cancel movements to a waypoint when a specific condition is present.

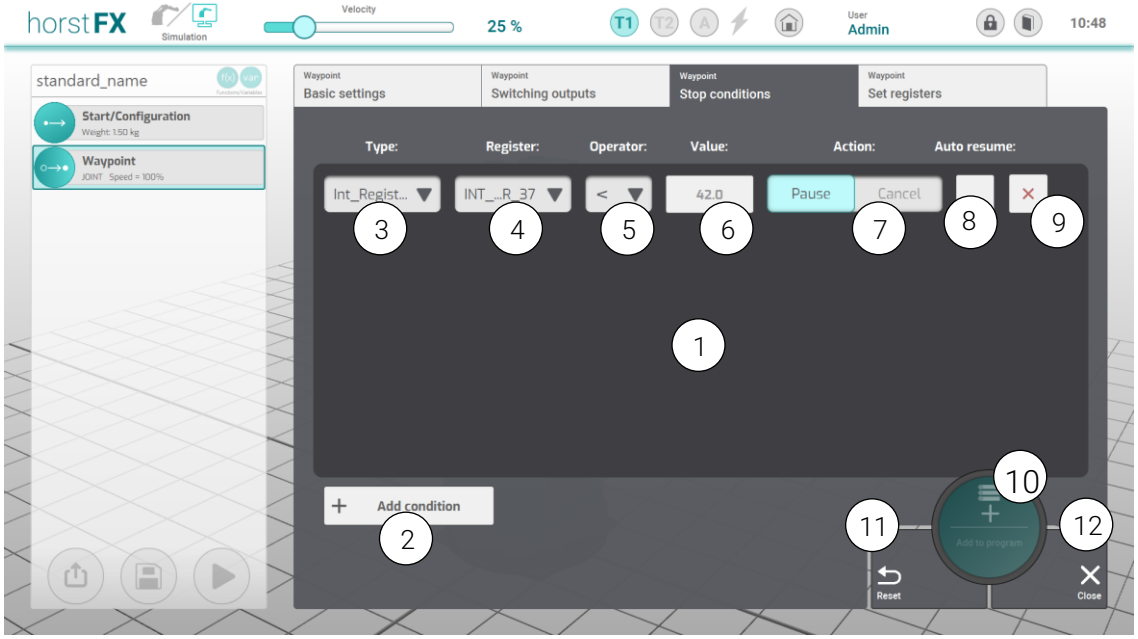


Fig. 6-12: Action window – Waypoint – Stop conditions

No.	Description
1	List – displays all entries (lines) that have been added
2	**Add condition** button – adds a new entry (line) to the list (1)
3	**Type** selection field – for selecting the type whose value is to be checked for a certain condition Input, register, and four Modbus types (InOutBits, InBits, InOutRegister, InRegister) are available for selection.
4	Depending on the selected type: For input or register: **Input/register** selection field – for selecting the input or register whose value is to be checked for a certain condition For one of the four Modbus types: **Address** input field – for entering the address whose value is to be checked for a certain condition
5	**Operator** selection field – for selecting the operator for the condition
6	**Value** input field – the value to be checked for in the condition
7	**Action** toggle button – sets the action that will be performed after the condition is met **Pause** action – if **Pause** is selected, the movement to the waypoint will be paused until the program execution is resumed using the corresponding button. **Cancel** action – if **Cancel** is selected, the movement to the waypoint is canceled and the program execution continues with the next action.
8	**Auto resume** checkbox. – automatical resume of the movement (only active, if **pause** is choses)
9	**Delete** button – deletes the corresponding entry (line)

No.	Description
10	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed The **Add to program** button is enabled only if a destination has been defined.
11	**Reset** button – resets all configuration and selection options in the action window to their default values
	Close button – closes the action window

6.3.2.6 Waypoint – Set register menu

Registers can be set while approaching a waypoint via the **Waypoint – Set register menu**. This means that values can be entered in a register at a specific time during the movement of the robot.

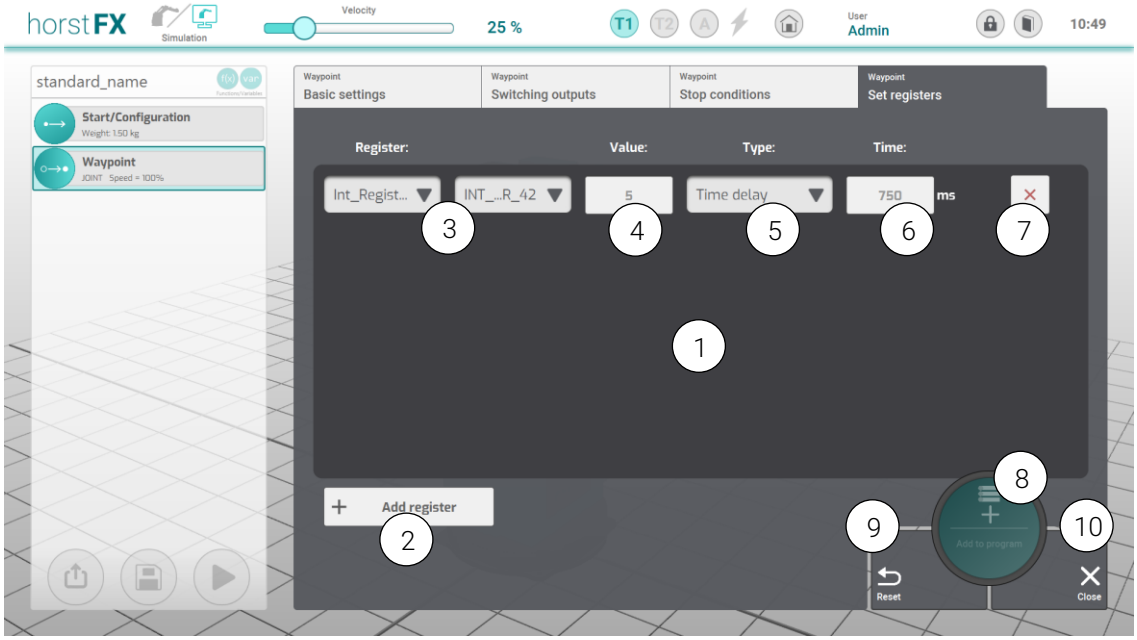


Fig. 6-13: Action window – Waypoint – Set register

No.	Description
1	List – displays all entries (lines) that have been added
2	**Add register** button – adds a new entry (line) to the list (1)
3	**Register** selection fields – for selecting the type of register (selection field on the left) and the register whose value is to be set The register types Float_Register, Int_Register, and Bool_Register can be selected. Depending on the selected register type, all corresponding registers are available for selection in the selection field on the right.
4	**Value** input field – for setting the value that the output will assume after switching

No.	Description
5	**Type** selection field – for selecting the time type **Path duration** type (in percent) – defines after what percentage of time passed the register value will be set. The time refers to the complete movement including acceleration and deceleration of the robot. **Time delay** type (in milliseconds) – defines after how many milliseconds from the beginning of the waypoint approach the register value is set. If the value is higher than the complete movement itself takes, the program waits at the end of the movement until the time value is reached, and the register value is set.
6	**Time** input field – the value in percent or milliseconds
7	**Delete** button – deletes the corresponding entry (line)
8	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed The **Add to program** button is enabled only if a destination has been defined.
9	**Reset** button – resets all configuration and selection options in the action window to their default values
10	**Close** button – closes the action window

6.3.3 Relative waypoint action

Selecting **Relative waypoint action** in the **action selection area** will bring up the corresponding action window. A new program block *Rel. waypoint* is created in the program tree.

The configuration of the **Relative waypoint action** is identical to the configuration of the **Waypoint action**.

The menus **Relative waypoint – Basic settings**, **Relative waypoint – Switch outputs**, **Relative waypoint – Stop conditions** and **Relative waypoint – Set registers** correspond to the menus (see section 6.3.1.2) **Waypoint – Basic settings**, **Waypoint – Switch outputs**, **Waypoint – Stop conditions** and **Waypoint – Set registers**.

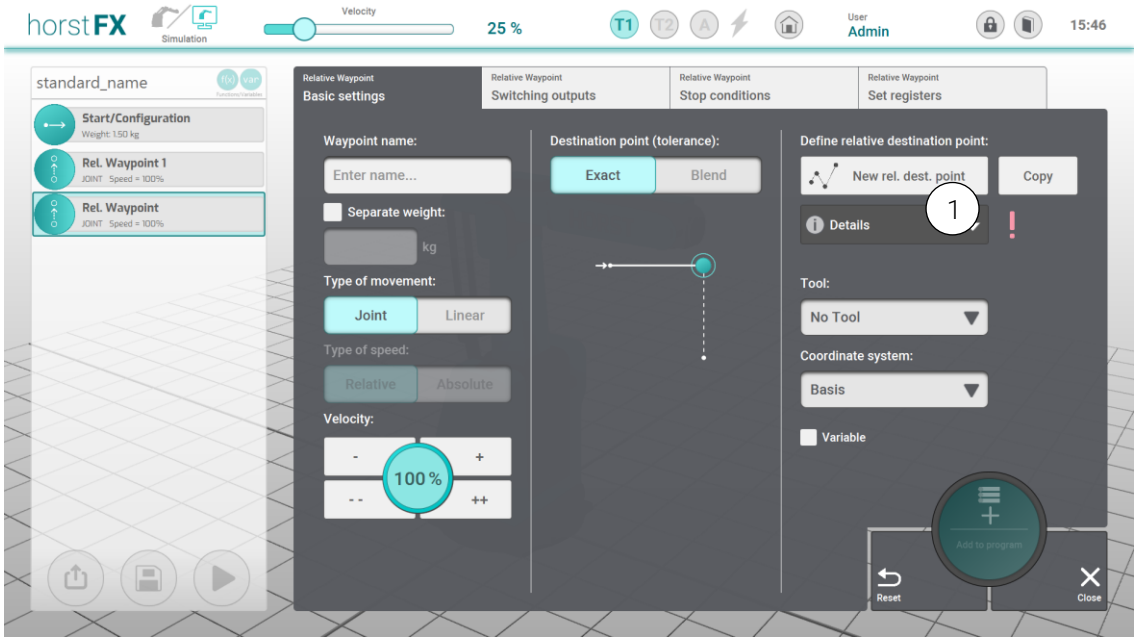


Fig. 6-14: Action window – Relative waypoint – Basic settings

The only difference between the **Relative waypoint action** and the **Waypoint action** is the **Define destination** or **Define relative destination** menu. The latter is activated by pressing the **New rel. dest. point** (1) button (compare with **New destination** button, Fig. 6-8).

By selecting **New rel. dest. point** (1) in the action window, the **Define relative destination** menu appears. Here there are several options, explained below, for defining a relative destination.

ATTENTION!
Since in a relative movement no absolute destination is known and therefore neither is the pose of the robot at the end of the relative movement, the robot may move to an invalid position or the path to the destination may not be possible. In such a case the program execution will be aborted.

6.3.3.1 Define relative destination – Robot axes

Pressing the **Axes** button (1) in the **Define relative destination** menu selects the **Define relative destination – Robot axes** menu.

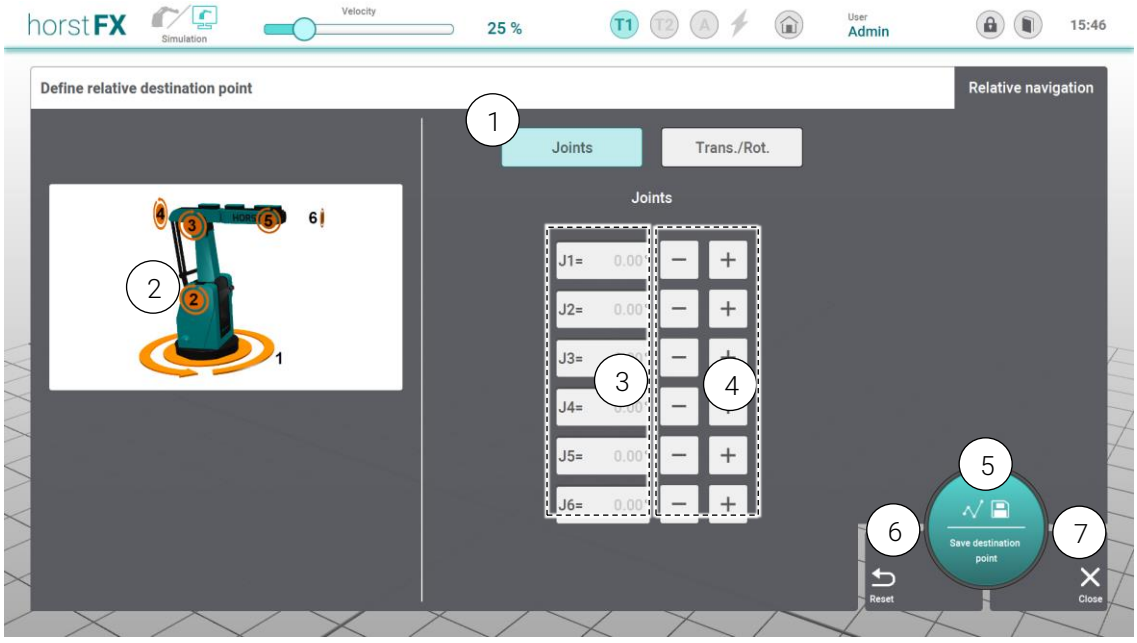


*Fig. 6-15: **Define relative destination – Robot axes** menu*

No.	Description
1	Selection buttons - depending on the selection, the corresponding elements of the menu are de-/activated
2	Displays robot model with the respective axis numbers
3	Axis value input field – for setting the value of the respective axis
4	**-/+** buttons – for changing the value of the respective axis (in 1° increments)
5	**Save destination point** button – the entered values in the menu are saved as a relative destination and the **Define relative destination** menu is closed
6	**Reset** button – resets all values entered in the menu to the last saved state

No.	Description
7	**Close** button – closes the **Define relative destination** menu without saving a relative destination

6.3.3.2 Define relative destination – translation and rotation

By selecting the button (1) **Cart. basis** or **Cart. TCP** in the **Define relative destination** menu, you can determine to which coordinate system the relative movement refers.

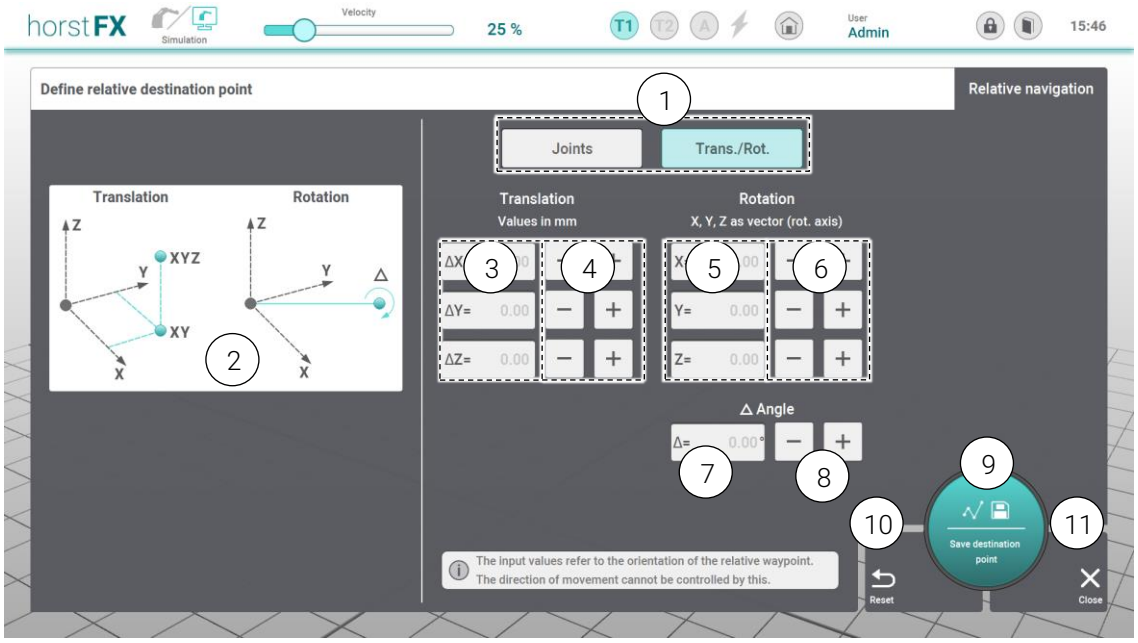


Fig. 6-16: Define relative destination menu– Translation and rotation

No.	Description
1	**Selection buttons** –depending on the selection, the corresponding elements of the menu are de-/activated
2	Display sketch of a translation and a rotation
3	**Δ values** input field – for setting the X, Y, Z value of the respective coordinate axis
4	**-/+ -** buttons – for changing the value of the respective axis (in 1 mm increments)
5	**Input values (vector) -** set the X, Y, Z value as vector (rotation axis)
6	**-/+ -** buttons – for changing the respective value (X, Y, Z in steps of 1)
7	**Δ angle input**– for setting the angle value of the rotation axis
8	**-/+ -** buttons **–** for changing the angle value (in 1° increments)
9	**Save destination point** button – the entered values in the menu are saved as a relative destination and the **Define relative destination** menu is closed
10	**Reset** button – resets all values entered in the menu to the last saved state
11	**Close** button – closes the **Define relative destination** menu without saving a relative destination

6.3.4 Function call action

By selecting the **Function call action** in the **action selection area**, the corresponding action windowappears. A new program block *function "new_function"* is created in the program tree.

For information on how to create new functions and edit existing functions, please see section 6.6.

6.3.4.1 Step 1: select function

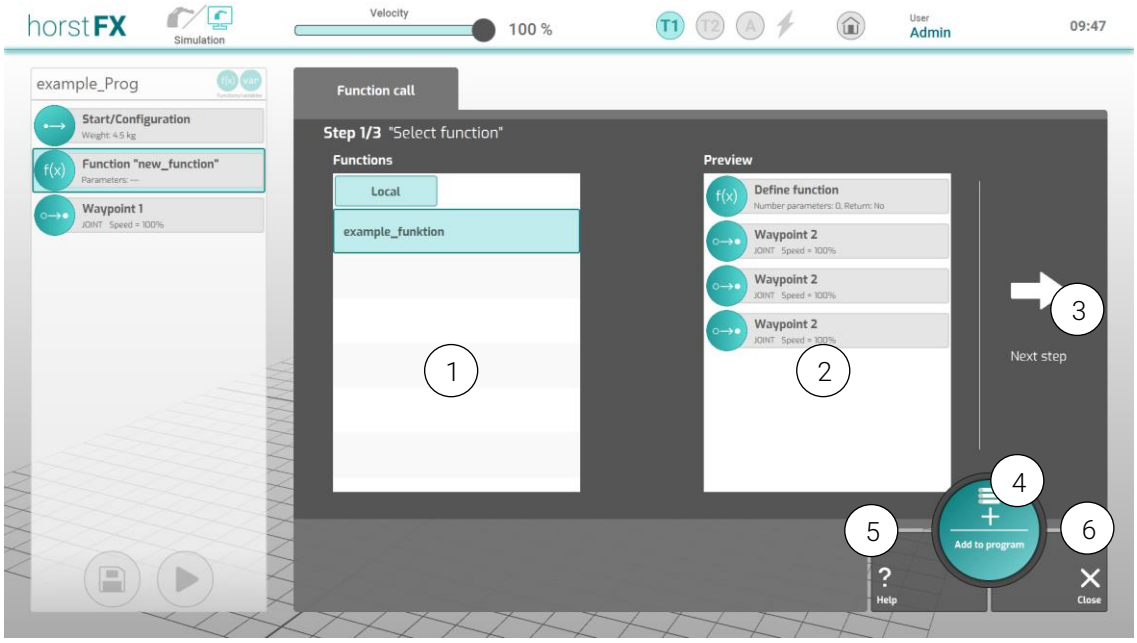


Fig. 6-17: Action window – Functions (step 1)

No.	Description
1	**Functions** display – list of all local and global functions
2	**Preview** display – preview of the content of the selected function
3	**Next step** button – switches to the next step of the action window
4	**Add to program** button – the selected function will be applied, and the action window closed
5	**Help** button – displays help in the action window
6	**Close** button – closes the action window

6.3.4.2 Step 2: function description and parameters

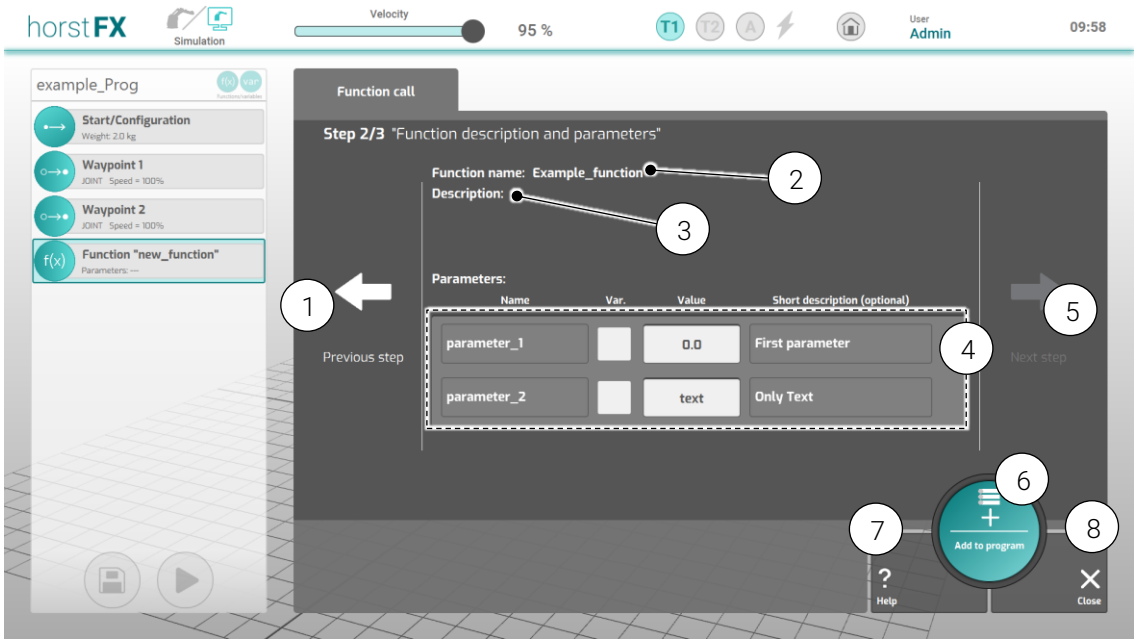


Fig. 6-18: Action window – Functions (step 2)

No.	Description
1	**Back** button – switches to the previous step of the action window
2	**Function name** display – displays the function name
3	**Description** display – displays the description of the function
4	**Transfer parameter** display / **Value** input field – the defined parameters are displayed (name, value, short description) The value of the parameters corresponds to the respective default value and can be changed here via the input field. Alternatively, the value can be assigned to a variable.
5	**Next step** button – switches to the next step of the action window
6	**Add to program** button – the selected function will be applied, and the action window closed
7	**Help** button – displays help in the action window
8	**Close** button – closes the action window

6.3.4.3 Step 3: assign return value

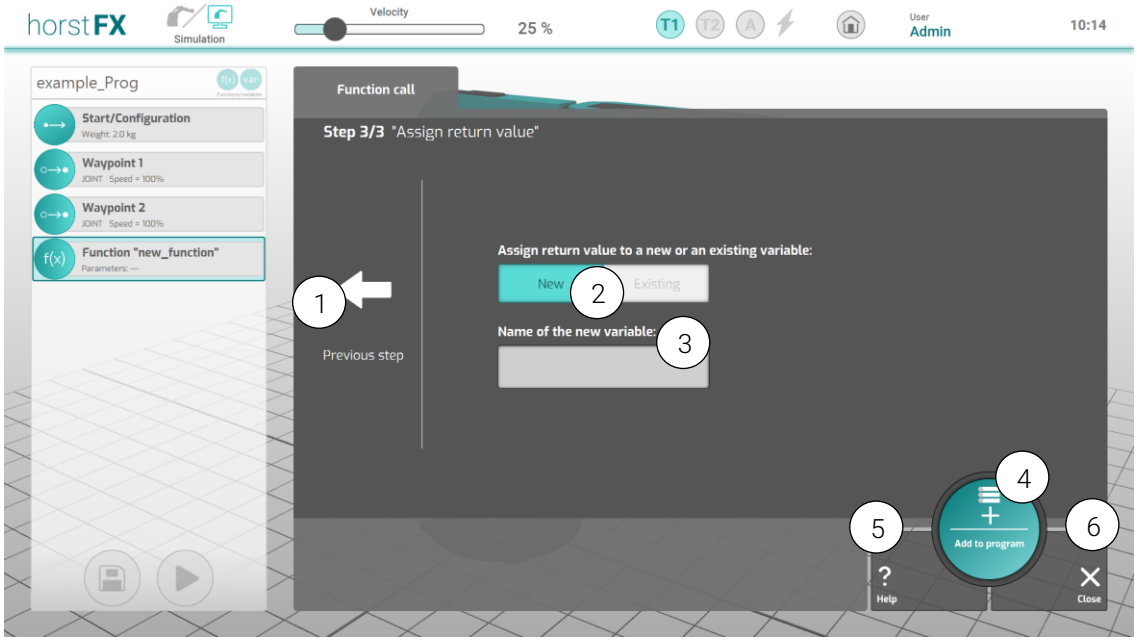


Fig. 6-19: Action window – Functions (step 3)

No.	Description
1	**Back** button – switches to the previous step of the action window
2	**Assign return value** toggle button – for selecting whether the return value should be assigned to a new or an existing variable
3	**Name** input field / **Variable** selection field – depending on the selection of (2), either the input for a new variable name or a selection field to select an existing variable is displayed here
4	**Add to program** button – the selected function will be applied, and the action window closed
5	**Help** button – displays help in the action window
6	**Close** button – closes the action window

6.3.5 Switch output action

By selecting **Switch output action** in the **action selection area**, the corresponding action window appears. A new program block *Switch output* is created in the program tree.

This action allows outputs to be switched. This can be used to trigger operations on add-on parts (e.g. grippers) or to forward information to other machines connected to the robot system.

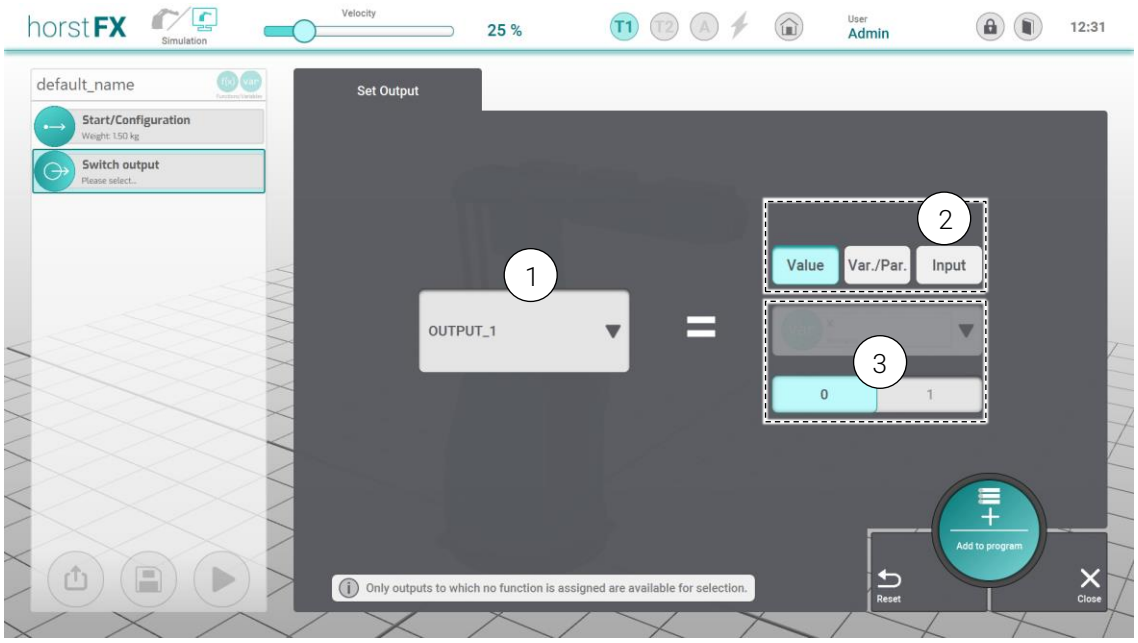


Fig. 6-20: Action window – Switch output

No.	Description
1	**Output** selection field– for selecting the output that will be switched
2	**Variable/Parameter/Input** button – activates the **Select variable/parameter/Input** selection field (5) (parameters can be selected only when editing a graphical function)
3	Button **0/1** – activates the toggle button **0/1** (6)
4	**Input** button – activates the **Select input** selection field (5)

6.3.6 Wait for action

By selecting the **Wait for action** in the **action selection area**, the corresponding action window appears. A new program block *Pause* or *Wait for* is created in the program tree.

A time period or an input signal can be selected as parameters. Program execution will be paused during this action until the set time period expires or the selected condition is met.

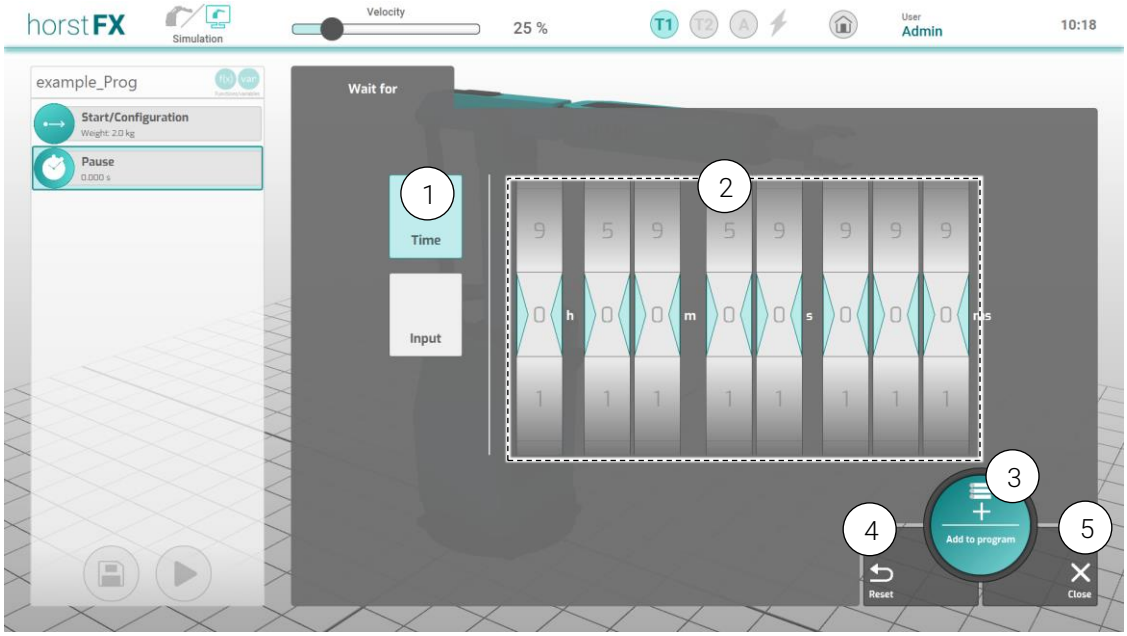


Fig. 6-21: Action window – Wait for – Time

No.	Description
1	**Time** button – for selecting the timespan of the condition
2	Timespan selection fields – time in hours, minutes, seconds, and milliseconds
3	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
4	**Reset** button – resets all configuration and selection options in the action window to their default values
5	**Close** button – closes the action window

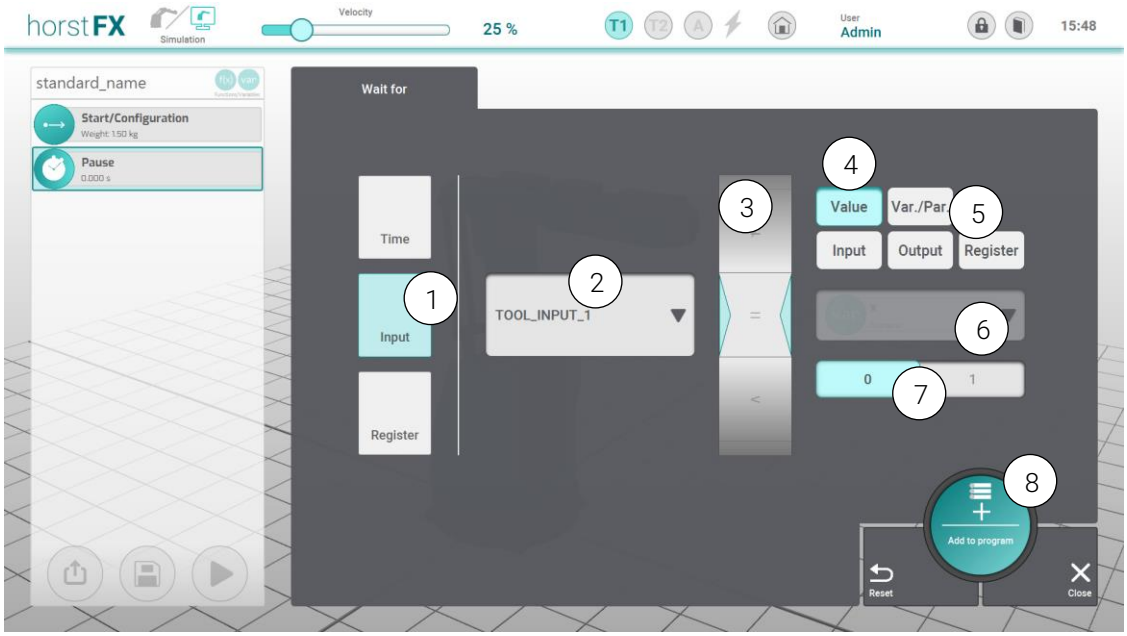


Fig. 6-22: Action window – Wait for – Input

No.	Description
1	**Input** button – for selecting the input signal condition
2	**Input** selection field – selection of the input whose value is to be checked for a certain condition
3	**Operator** selection field – for selecting the operator for the condition
4	**Value** button – activates the toggle button **0/1** (7)
5	**Var./Par./Input/Output/Register** button – activates the **Select Var./Par./Input/Output/Register** selection field (6)
6	**Select Var./Par./Input/Output/Register** selection field – for selecting the input whose value is to be checked for in the condition
7	**0/1** toggle button – for setting the value to be checked for in the condition
8	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed

6.3.7 Change variable value action

By selecting **Change variable value action** in the **action selection area**, the corresponding action window appears. A new program block *Change variable value* is created in the program tree.

This action can be used to assign a specific value to a variable or to change the previous value. For numeric variables, an incrementing and decrementing functionality is also available.

For information on how to create new variables and delete existing variables, please see section 6.6.3.

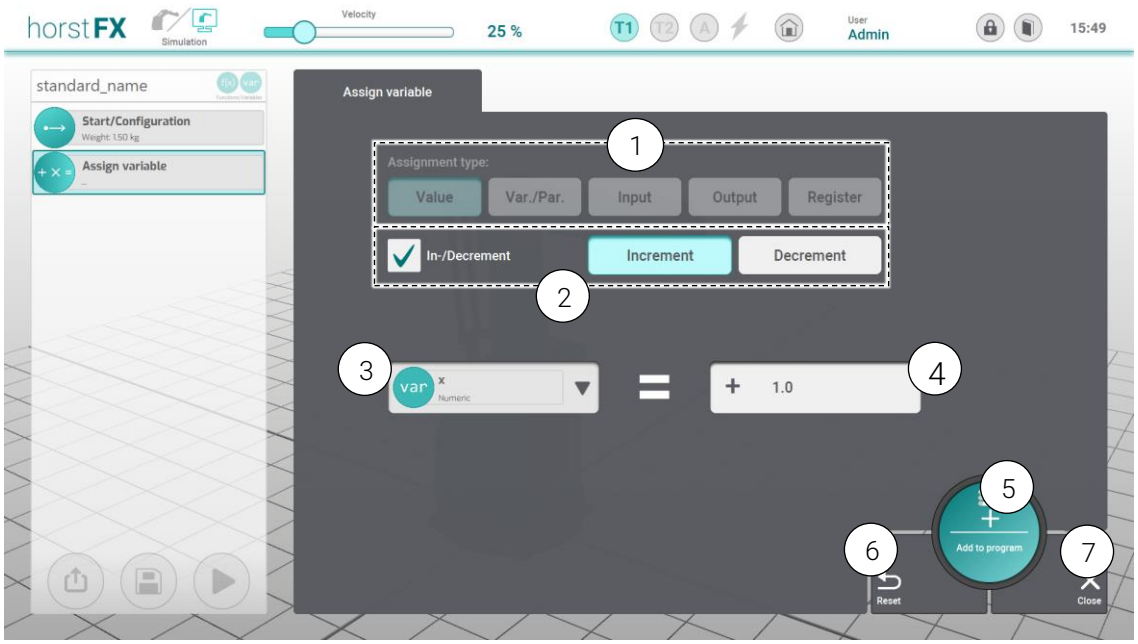


Fig. 6-23: Action window – Change variable value

No.	Description
1	**Assignment type** buttons – for selecting the assignment type (parameters can be selected only when editing a graphical function) The following types can be assigned to a variable of the *Numeric* data type: numeric value, variable/parameter of the *Numeric* data type, input, output The following types can be assigned to a variable of data type *Text*: textual value, variable/parameter of data type *Text* (parameters can be selected only when editing a graphical function)
2	**Increment/Decrement** buttons – for selecting the incrementing or decrementing functionality
3	**Select variable** selection field – for selecting the variable for which a certain value is to be set
4	**Value** input field – for entering/selecting the variable value Depending on the selection of the assignment type (3), the corresponding input or selection field appears here.
5	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed The **Add to program** button is activated only after a valid value is entered or selected.

No.	Description
6	**Reset** button – resets all configuration and selection options in the action window to their default values
7	**Close** button – closes the action window

6.3.7.1 Variables of the *target position* data type

A variable of the Target position data type is assigned a target position for a (relative) waypoint, the value of another variable or the current position of the robot as a value.

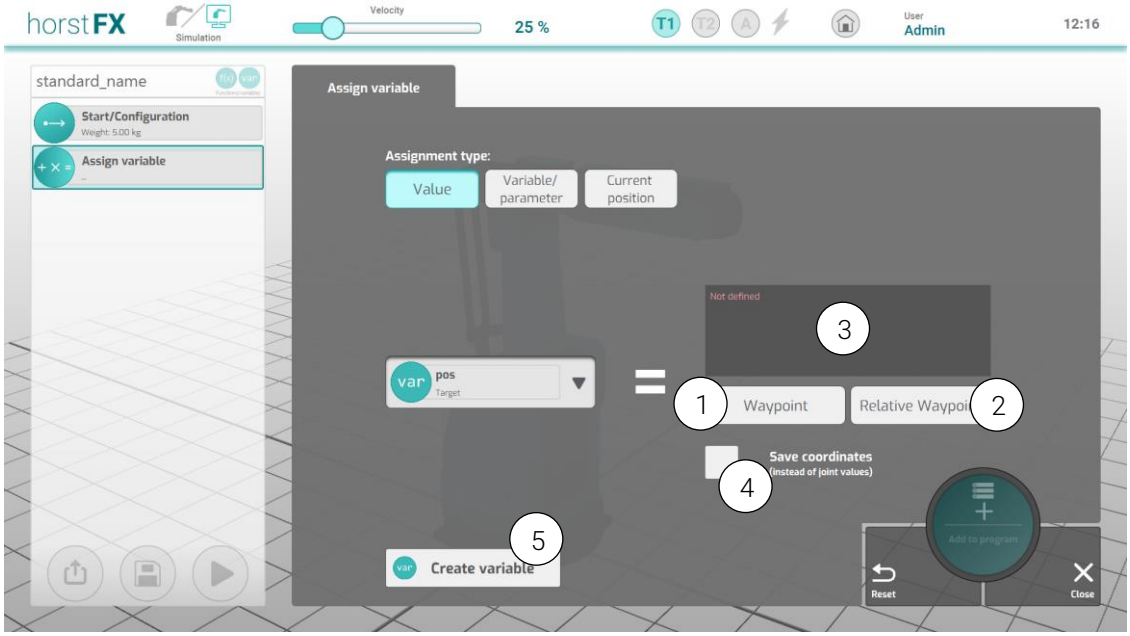


Fig. 6-24: Action window – Change variable value – Target position

No.	Description
1	**Waypoint** button – defining a destination point for a waypoint (see section 6.3.2.2)
2	**Relative waypoint** button – defining a destination point for a waypoint (see section 6.3.3.1 and 6.3.3.2)
3	**Destination point definition** display – display of the corresponding values of the defined destination point as soon as defined
4	**Save coordinates** check box – saves the values of the destination point as coordinates with orientation (by default, the axis values are saved).
5	**Create variable** button – for linking to the **Declare variable action** (see section 6.6.3) to be able to create/declare a variable directly

6.3.8 Repeat action

Selecting the **Repeat action** in the **action selection area** displays the corresponding action window. A new grouped program block *Repeat* is created in the program tree.

This action is used to create a repeat loop. Any number of actions can be added within the repeat loop. The repeat loop executes its contents until its execution condition is no longer satisfied.

If the action with the condition **Endless** (2) is selected, the repeat loop will not be exited during a program execution. As a result, the program execution can only be terminated by canceling it manually.

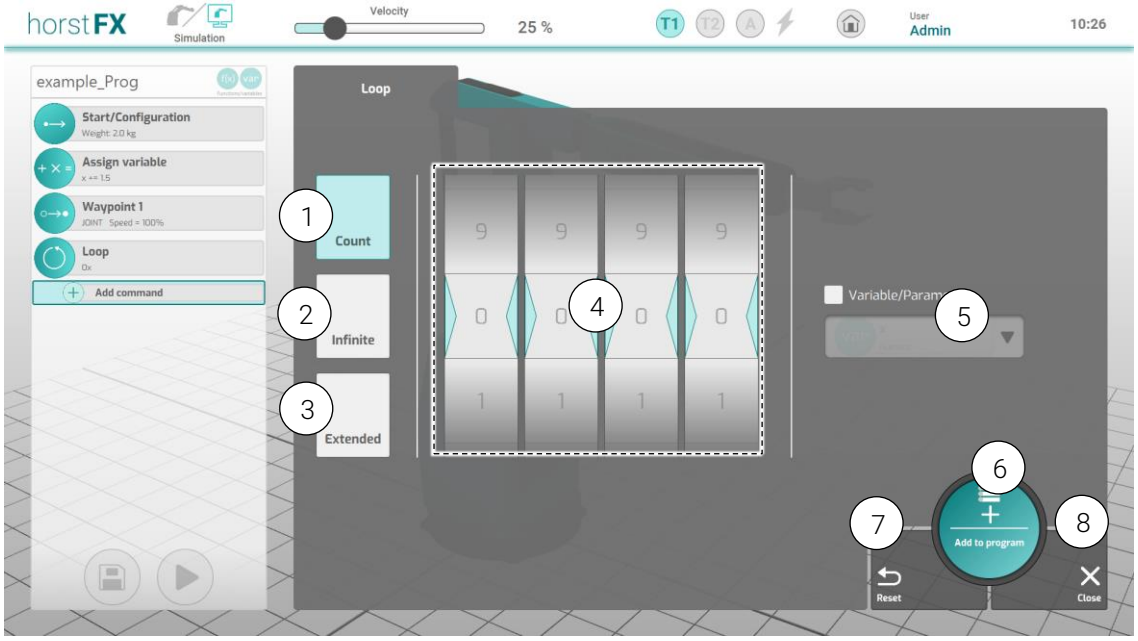


Fig. 6-25: Action window – Repeat – Number

No.	Description
1	**Count** button – for selecting the condition “Number”
2	**Infinite** button – for selecting the condition “Endless”
3	**Extended** button – for extending the selection of conditions
4	Selection fields for a certain number of repetitions (max. 9999) The repeat loop is executed until the set number of repetitions is reached (also applies to (5)).
5	**Variable/parameter** selection field – after activating the selection field, a value of a variable/parameter can be set as the number of repetitions (parameters can be selected only when editing a graphical function)
6	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
7	**Reset** button – resets all configuration and selection options in the action window to their default values
8	**Close** button – closes the action window

If the action with the condition **Extended – Input** (1), **Extended – Variable** (2) or **Extended – Output** (3) is selected, the repeat loop is executed during a program execution until the configured condition is no longer fulfilled.

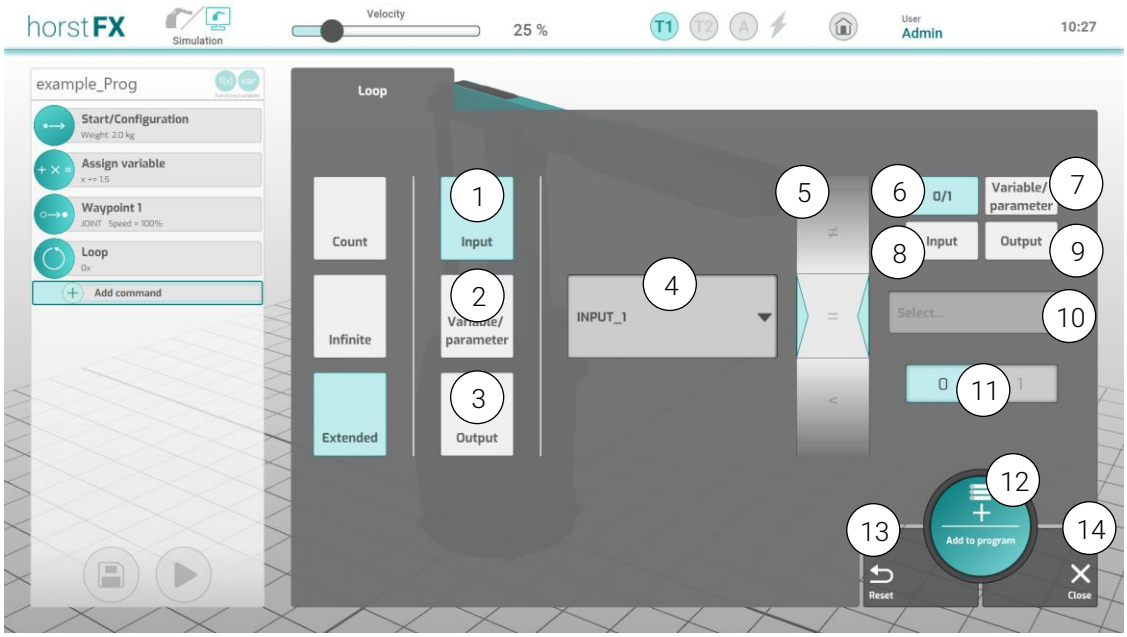


Fig. 6-26: Action window – Repeat – Extended – Input

No.	Description
1	**Input** button – for selecting the condition "Input"
2	**Variable/parameter** button – for selecting the condition "Variable/parameter"
3	**Output** button – for selecting the condition "Output"
4	**Input** selection field – selection of the input whose value is to be checked for a certain condition
5	**Operator** selection field – for selecting the operator for the condition
6	**0/1** button – activates the toggle button **0/1** (11)
7	**Variable/Parameter** button – activates the **Select variable/parameter** selection field (10) (parameters can be selected only when editing a graphical function)
8	**Input** button – activates the **Select input** selection field (10)
9	**Output** button – activates the **Select output** selection field (10)
10	**Select input** selection field – for selecting the input whose value is to be checked for in the condition
11	**0/1** toggle button – for setting the value to be checked for in the condition
12	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
13	**Reset** button – resets all configuration and selection options in the action window to their default values
14	**Close** button – closes the action window

The **Variable/Parameter** (2) and **Output** (3) conditions are configured similarly to the **Input** (1) condition.

6.3.9 If-condition action

By selecting the **If-Condition action** in the **action selection area**, the corresponding action window appears. A new grouped program block *If-condition* is created in the program tree.

The content of an if-condition is executed during a program execution only if a certain condition is met. If this does not happen, all included actions will be skipped.

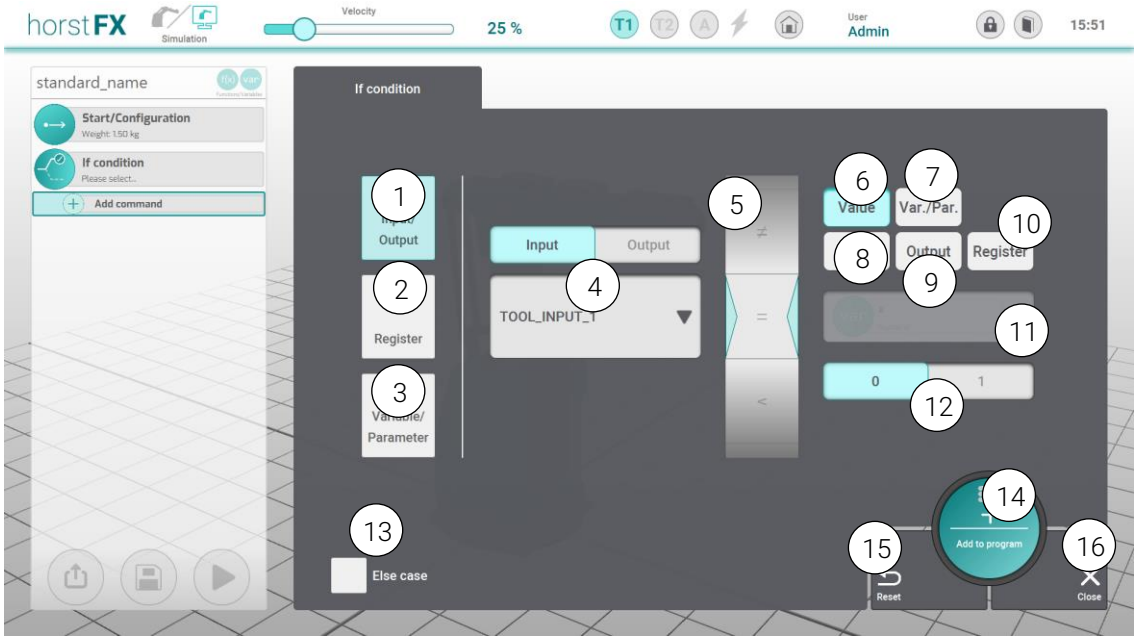


Fig. 6-27: Action window – If-condition – Input

No.	Description
1	**Input/Output** button – activates the toggle button "Input/Output"
2	**Register** button – for selecting the condition "Register"
3	**Variable/parameter** button – for selecting the condition "Variable/parameter"
4	**Input** selection field – selection of the input whose value is to be checked for a certain condition (Toggle-Button/selection field – gets activated/shown depending on the selection of (1-3))
5	**Operator** selection field – for selecting the operator for the condition
6	**Value** button – activates Value input field (11)
7	**Variable/Parameter** button – activates the **Select variable/parameter** selection field (11) (parameters can be selected only when editing a graphical function)
8	**Input** button – activates the **Select input** selection field (11)
9	**Output** button – activates the **Select output** selection field (11)
10	**Register** button – activates the **Select register** selection field (11)
11	**Select input** selection field – for selecting the input whose value is to be checked for in the condition (depends on what is selected in 6-10)
12	**0/1** toggle button – for setting the value to be checked for in the condition
13	**Add else case** check box – adds an Else-condition to the If-condition.

No.	Description
14	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
15	**Reset** button – resets all configuration and selection options in the action window to their default values
16	**Close** button – closes the action window

The **Variable/Parameter** (2) and **Output** (3) conditions are configured similarly to the **Input** (1) condition.

If the check box **Else case** (12) is activated, another grouped program block "Else-statement" is attached to the program block "If-condition" after the configuration options (13) have been accepted in the program tree. These two program blocks are permanently linked to each other.

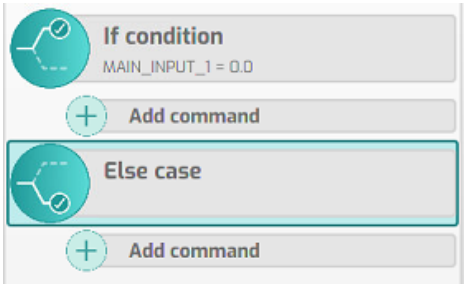


Fig. 6-28: If-condition with Else case

6.3.10 Set register action

By selecting Action **Set register** in the **action selection area**, the corresponding action window appears. A new programme component Set Register is created in the programme tree.

This action can be used to set the value of a register, which can be used in other actions or to forward information (via external interfaces) to other machines connected to the robot system.

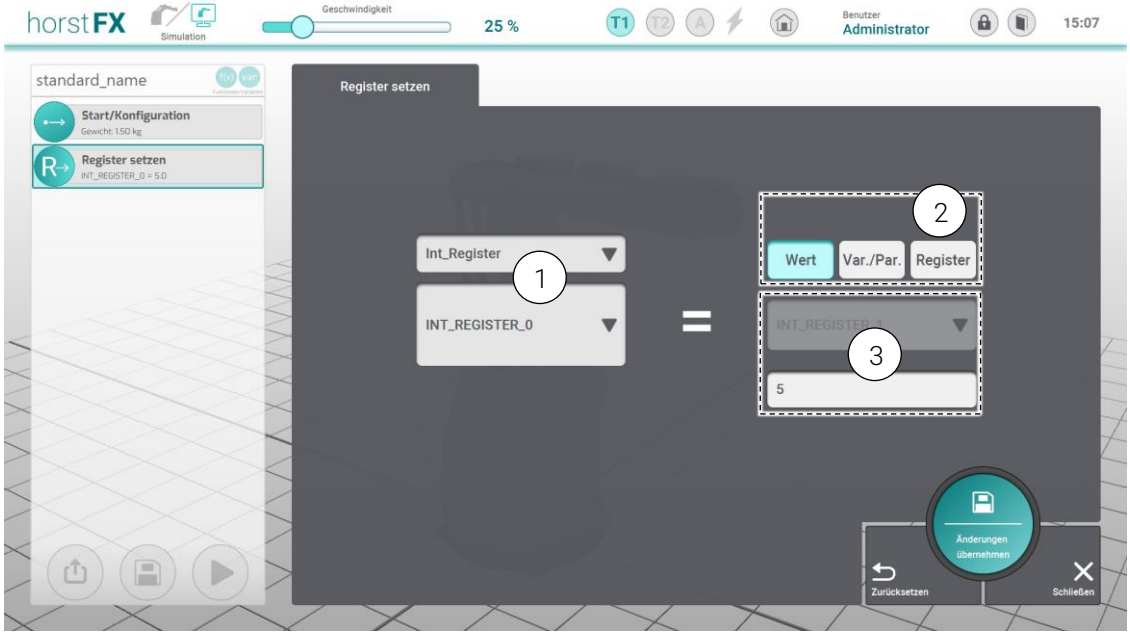


Fig. 6-29: Action window - Set registers

Item	Description
1	Selection fields **Register** - Selection of the register type and the register to which a value is assigned.
2	Buttons **Value, Var./Par., Input** - Selection of the type for the assignment

Item	Description
3	Selection field/toggle button - define the value to be assigned (is shown/activated according to the selection of the type)

6.3.11 Pallet action

By selecting the **Pallet action** in the **action selection area**, the corresponding action window appears. A new grouped program block is created in the program tree. This is a special program block for the **Pallet action**. It consists of several program blocks that are connected by a line. The special program block for the **Pallet action** begins with the *Start pallet pallet_new* program block and ends with the *End pallet pallet_new* program block ("pallet_new" is the default name of a new, as yet undefined pallet). This makes it clear which program blocks belong to a **Pallet action**. A pallet is defined in three steps.

All associated actions (program blocks within) represent **one** palletizing pass, which means that only the first palletizing point is processed during a program execution. After that, the program execution continues with the subsequent action. If all palletizing points must be processed, i.e. the complete pallet, the **Pallet action** must be set to a **Repeat action** (see section 6.3.7.1) and the number of the repeat loop must be set according to the number of defined palletizing points. If the number of the repeat loop is greater than the number of defined palletizing points, the processing starts again with the first palletizing point after the last palletizing point and continues until the number in the repeat loop is reached.

Example: To process a 4 x 6 pallet completely once, the number 24 must be entered in the repeat loop. If 28 is entered as the number, the program execution starts again with the first palletizing point after processing a complete pallet and stops after the fourth palletizing point.

To process all palletizing points or several palletizing points of a pallet, the Pallet action must be added to a Repeat action and the number of repeat loops must be set accordingly.

6.3.11.1 Step 1: create pallet grid

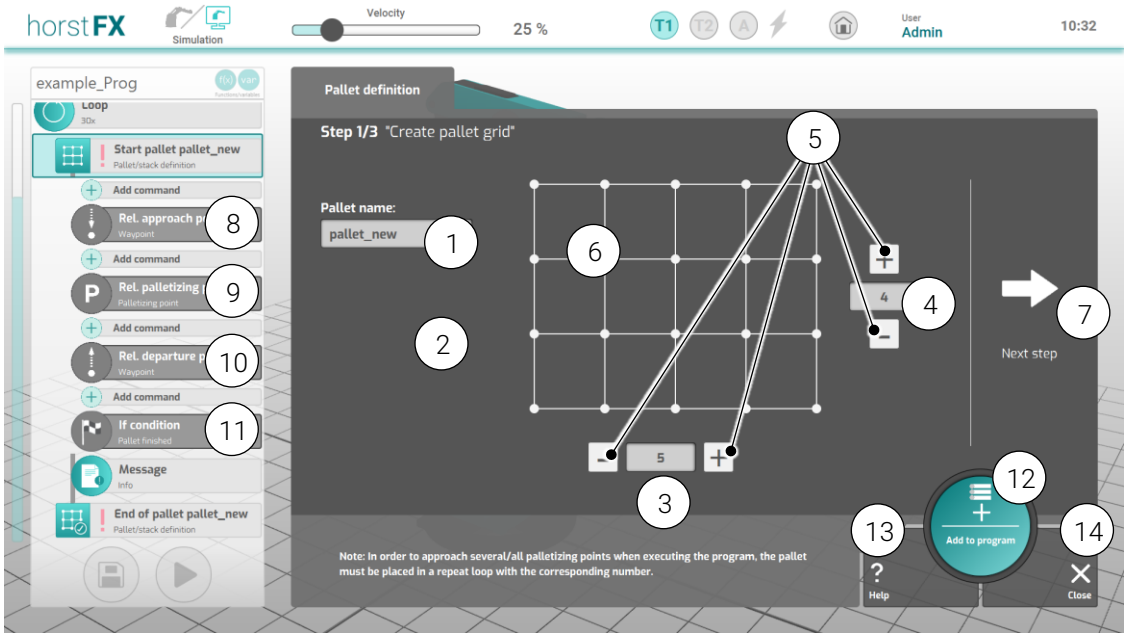


Fig. 6-30: Action window – Pallet definition (step 1)

No.	Description
1	**Pallet name** input field – for naming the pallet This name appears after the addition (11) in the program tree in the *Start pallet* and *End pallet* program blocks.
2	**Tool** selection field – for selecting the tool with which the palletizing points and the approach/departure point are to be approached Only relevant for programs with more than one tool (see section 6.8).
3	**Number** input field (columns) – number of columns in the pallet
4	**Number** input field (rows) – number of rows of the pallet
5	**-/+** buttons – change the number (in increments of 1)
6	Pallet display – visualization of the pallet, consisting of all palletizing points The display of the pallet adapts to the set number of columns and rows
7	**Next step** button – switches to the next step of the action window
8	Program block *Rel. approach point* – see section 6.3.12.4
9	Program block *Rel. palletizing point* – see section 6.3.12.4
10	Program block *Rel. departure point* – see section 6.3.12.4
11	Program block *If-condition – see* section 6.3.12.13
12	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
13	**Help** button – displays help in the action window
14	**Close** button – closes the action window

6.3.11.2 Step 2: define corner points

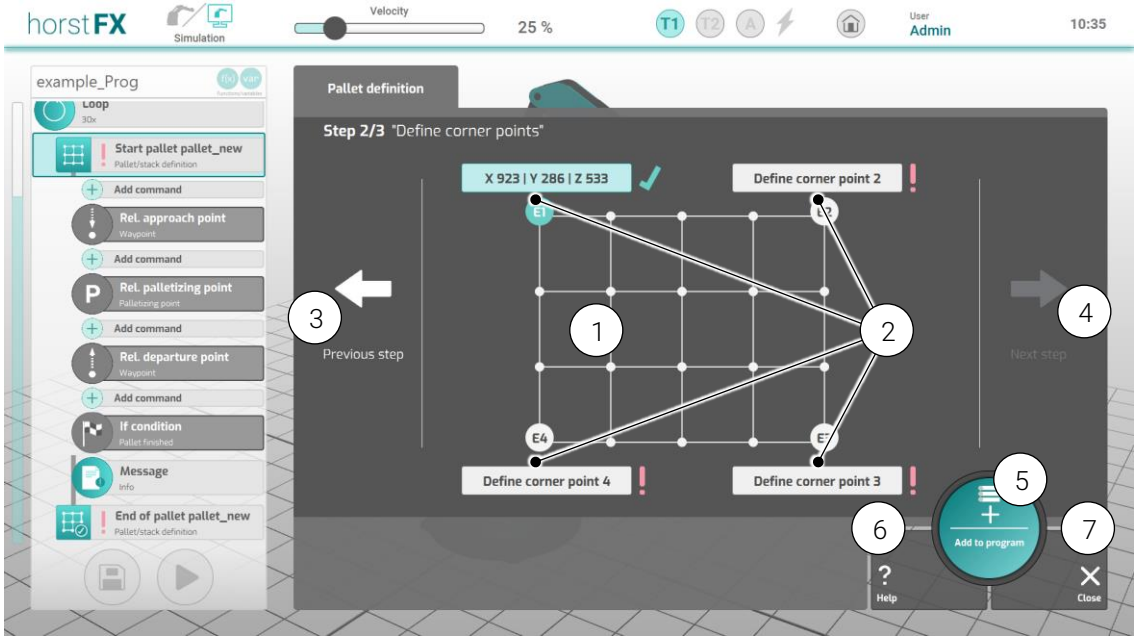


Fig. 6-31: Action window – Pallet definition (step 2)

No.	Description
1	Pallet display – visualization of the pallet consisting of all palletizing points and with marking of all corner points
2	**Define corner point 1, 2, 3, 4** button – defines a destination for the respective corner point The **Define destination** menu opens (see section 6.3.2.2).
3	**Back** button – switches to the previous step of the action window
4	**Next step** button – switches to the next step of the action window
5	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
6	**Help** button – displays help in the action window
7	**Close** button – closes the action window

6.3.11.3 Step 3: define approach and departure point

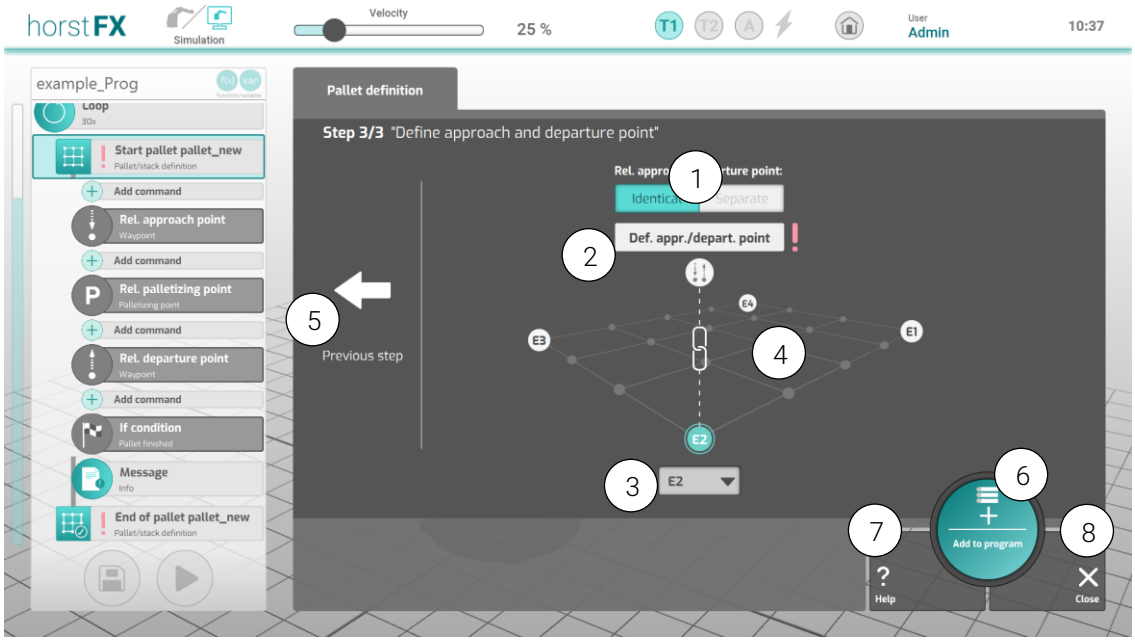


Fig. 6-32: Action window – Pallet definition (step 3)

No.	Description
1	**Identical/Separate** toggle button – for selecting whether the approach and departure points use the same destination point **Identical** – the destination point for the approach and departure points is the same. **Separate** – the approach and departure points have different destinations and thus must be defined separately.
2	**Define approach/departure point** button – defines a destination point for approach and departure. The **Define destination** menu opens (see section 6.3.2.2). If **Separate** is selected on the toggle button (1), there is a separate button for defining the approach and departure points.

No.	Description
3	**Corner point** selection field – for selecting a corner point The selected corner point serves as the starting point for the relative approach and departure point. The approach and departure points of all palletizing points are calculated relative to this selected corner point.
4	Pallet display – visualization of the pallet, consisting of all palletizing points and with marking of all corner points as well as special marking for the corner point that serves as a starting point for the relative approach and departure point of all palletizing points
5	**Back** button – switches to the previous step of the action window
6	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
7	**Help** button – displays help in the action window
8	**Close** button – closes the action window

A pallet that has not yet been completely defined can still be saved and applied to the program tree. However, as long as their status is undefined, the program blocks *Start pallet pallet_new* and *End pallet pallet_new* will be marked with a warning symbol (1). If one or more incompletely defined pallets are present in a program, no program execution can be started.

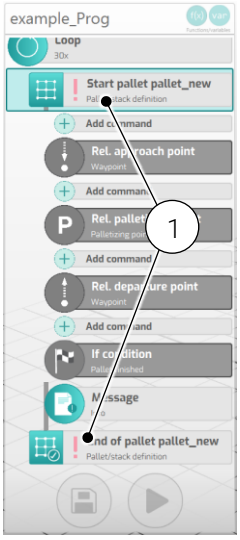


Fig. 6-33: Program blocks of the Pallet action

Once a pallet is fully defined, a simple model of the pallet is displayed in the 3D world.

6.3.11.4 Program block Rel. approach/palletizing/departure point

These three special program blocks are only available in the grouped program block of the **Pallet action**. They can be neither deleted nor moved. The corresponding action window can only be called up for this action via the editing menu (see section 6.3.28).

All three actions are similar to the **Relative waypoint action** (see section 6.3.2.6). There are simply fewer configuration and selection options (cf. Fig. 6-47), as well as no possibility to define a relative destination, since this is calculated automatically for the approach, palletizing, and departure point via the pallet definition.

Unlike the program block *Rel. palletizing point*, the two program blocks *Rel. approach point* and *Rel. departure point* have an additional option. Using the **Orientation of the point** toggle button, the *Adapted* option can be selected here, which adapts the orientation of the approach or departure point to the orientation of the respective palletizing point, which is not the case with the *Standard* option.

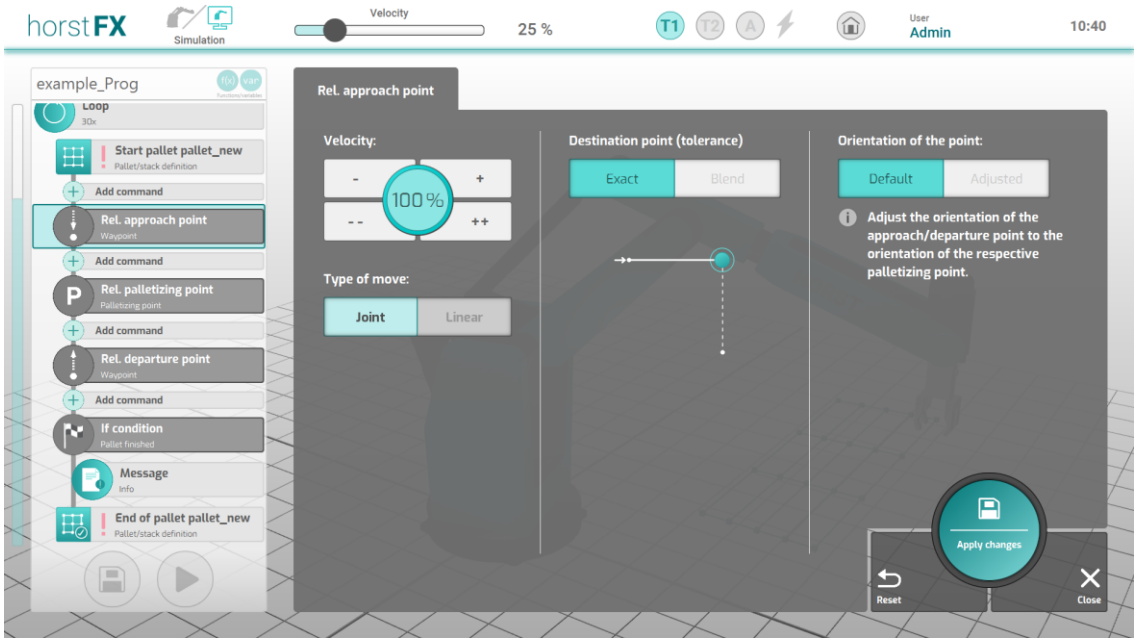


Fig. 6-34: Action window – Rel. palletizing point

6.3.11.5 If-condition program block

This special program block is only available in the grouped program block of the **Pallet action**. It cannot be deleted, edited, or moved. The corresponding action window can only be called up for this action via the editing menu (see section 6.3.28).

This action is similar to the **If-condition action** (see section 6.3.9), but the condition is fixed. It is checked whether the last palletizing point, i.e. the complete pallet, has been processed. If this is the case, a message is displayed by default indicating that the pallet has been processed.

6.3.12 Pallet action new since Version 23.04

By selecting the **Pallet action** in the **action selection area**, the corresponding action window appears. A new grouped program block is created in the program tree. This is a special program block for the **Pallet action**. It consists of several program blocks that are connected by a line. The special program block for the **Pallet action** begins with the *Start pallet pallet_new* program block and ends with the *End pallet pallet_new* program block ("pallet_new" is the default name of a new, as yet undefined pallet). This makes it clear which program blocks belong to a **Pallet action**. A pallet is defined in three steps.

All associated actions (program blocks within) represent **one** palletizing pass, which means that only the first palletizing point is processed during a program execution. After that, the program execution continues with the subsequent action. If all palletizing points must be processed, i.e. the complete pallet, the **Pallet action** must be set to a **Repeat action** (see section 6.3.7.1) and the number of the repeat loop must be set according to the number of defined palletizing points. If the number of the repeat loop is greater than the number of defined palletizing points, the processing starts again with the first palletizing point after the last palletizing point and continues until the number in the repeat loop is reached.

Example: To process a 4 x 6 pallet completely once, the number 24 must be entered in the repeat loop. If 28 is entered as the number, the program execution starts again with the first palletizing point after processing a complete pallet and stops after the fourth palletizing point.

To process all palletizing points or several palletizing points of a pallet, the Pallet action must be added to a Repeat action and the number of repeat loops must be set accordingly.

6.3.12.1 Step 1: create pallet grid

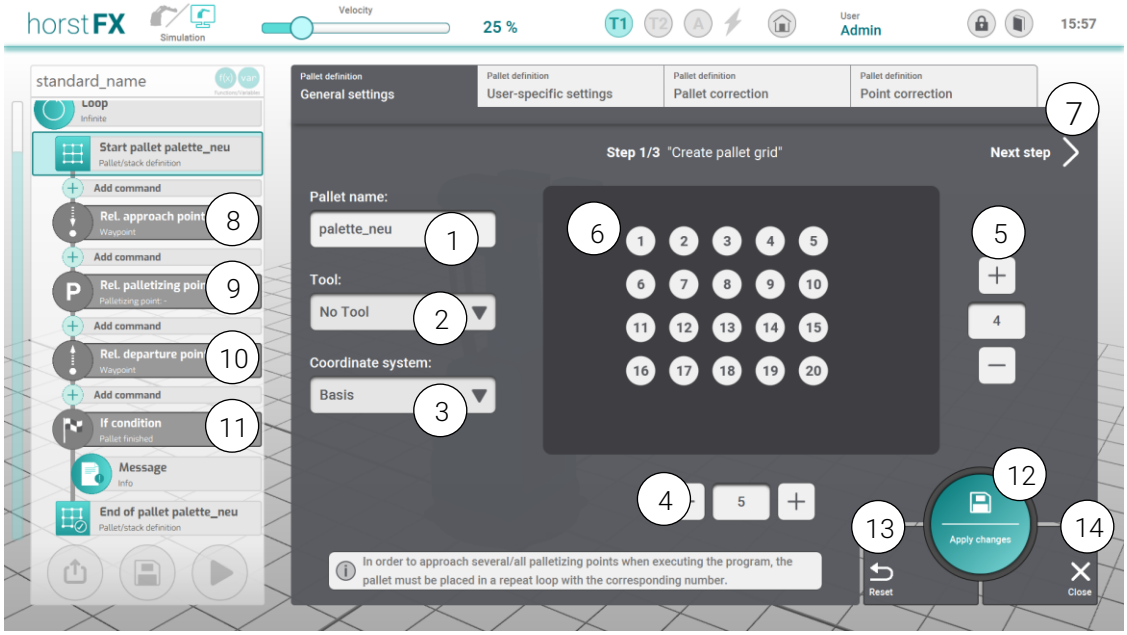


Fig. 6-35: Action window – Pallet definition (step 1)

No.	Description
1	**Pallet name** input field – for naming the pallet This name appears after the addition (11) in the program tree in the *Start pallet* and *End pallet* program blocks.
2	**Tool** selection field – for selecting the tool with which the palletizing points and the approach/departure point are to be approached Only relevant for programs with more than one tool (see section 6.8).
3	**Coordinate syst.** selection field – for selecting the coordinate system with which the palletizing points and the approach/departure point are to be approached
4	**Number** input field (columns) – number of columns in the pallet
5	**Number** input field (rows) – number of rows of the pallet
6	Pallet display – visualization of the pallet, consisting of all palletizing points The display of the pallet adapts to the set number of columns and rows
7	**Next step** button – switches to the next step of the action window
8	Program block *Rel. approach point* – see section 6.3.12.4
9	Program block *Rel. palletizing point* – see section 6.3.12.4

No.	Description
10	Program block *Rel. departure point* – see section 6.3.12.4
11	Program block *If-condition* – *see* section 6.3.12.13
12	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
13	**Help** button – displays help in the action window
14	**Close** button – closes the action window

6.3.12.2 Step 2: define corner points

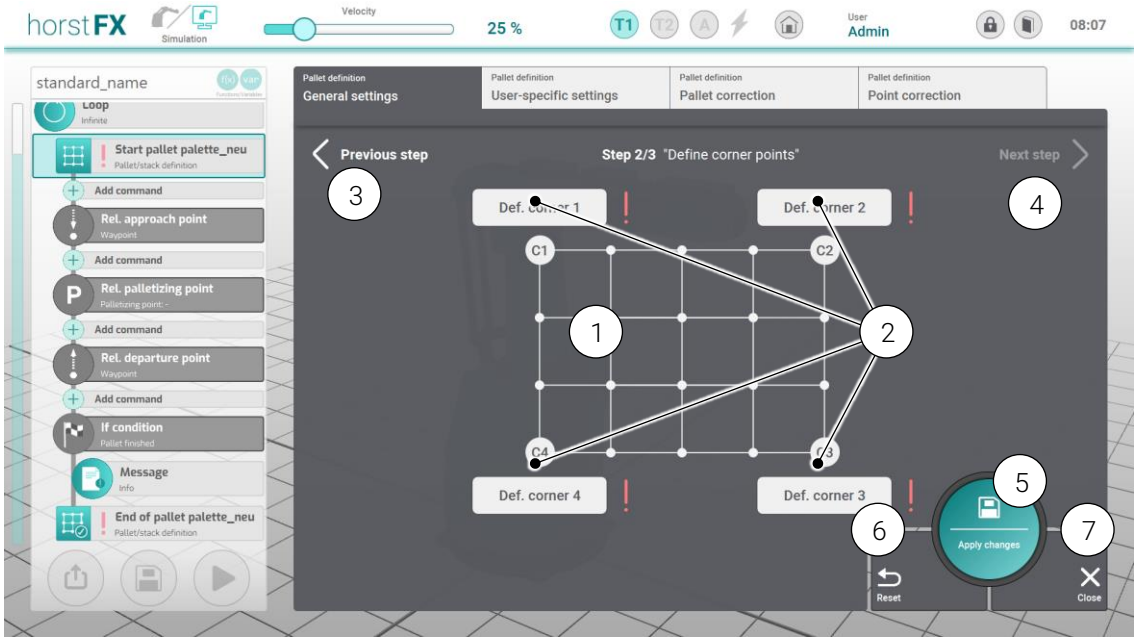


Fig. 6-36: Action window – Pallet definition (step 2)

No.	Description
1	Pallet display – visualization of the pallet consisting of all palletizing points and with marking of all corner points
2	**Define corner point 1, 2, 3, 4** button – defines a destination for the respective corner point The **Define destination** menu opens (see section 6.3.2.2).
3	**Previoues step** button – switches to the previous step of the action window
4	**Next step** button – switches to the next step of the action window
5	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
6	**Help** button – displays help in the action window
7	**Close** button – closes the action window

6.3.12.3 Step 3: define approach and departure point

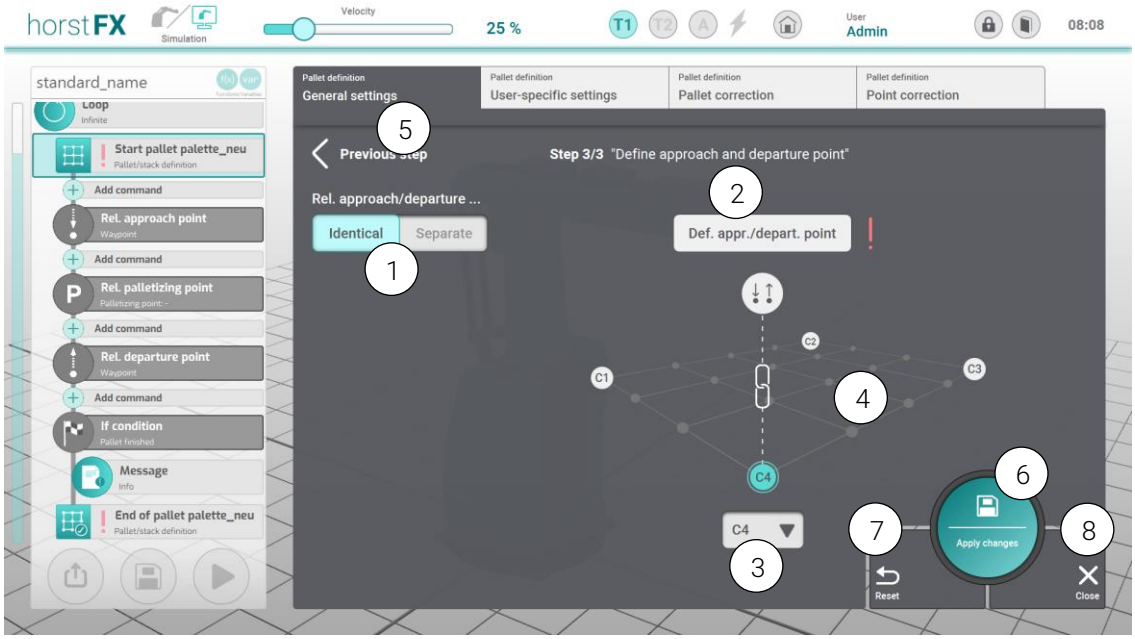


Fig. 6-37: Action window – Pallet definition (step 3)

No.	Description
1	**Identical/Separate** toggle button – for selecting whether the approach and departure points use the same destination point **Identical** – the destination point for the approach and departure points is the same. **Separate** – the approach and departure points have different destinations and thus must be defined separately.
2	**Define approach/departure point** button – defines a destination point for approach and departure. The **Define destination** menu opens (see section 6.3.2.2). If **Separate** is selected on the toggle button (1), there is a separate button for defining the approach and departure points.
3	**Corner point** selection field – for selecting a corner point The selected corner point serves as the starting point for the relative approach and departure point. The approach and departure points of all palletizing points are calculated relative to this selected corner point.
4	Pallet display – visualization of the pallet, consisting of all palletizing points and with marking of all corner points as well as special marking for the corner point that serves as a starting point for the relative approach and departure point of all palletizing points
5	**Previous step** button – switches to the previous step of the action window
6	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
7	**Help** button – displays help in the action window
8	**Close** button – closes the action window

A pallet that has not yet been completely defined can still be saved and applied to the program tree. However, as long as their status is undefined, the program blocks *Start pallet pallet_new* and *End pallet pallet_new* will be marked with a warning symbol (1). If one or more incompletely defined pallets are present in a program, no program execution can be started.

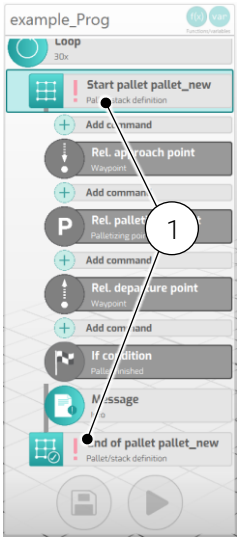


Fig. 6-38: Program blocks of the Pallet action

Once a pallet is fully defined, a simple model of the pallet is displayed in the 3D world.

6.3.12.4 User-specific settings Step 1

It is now possible to define subsections. These can be used to process parts of the pallet and also to specify any sequence of palletising points.

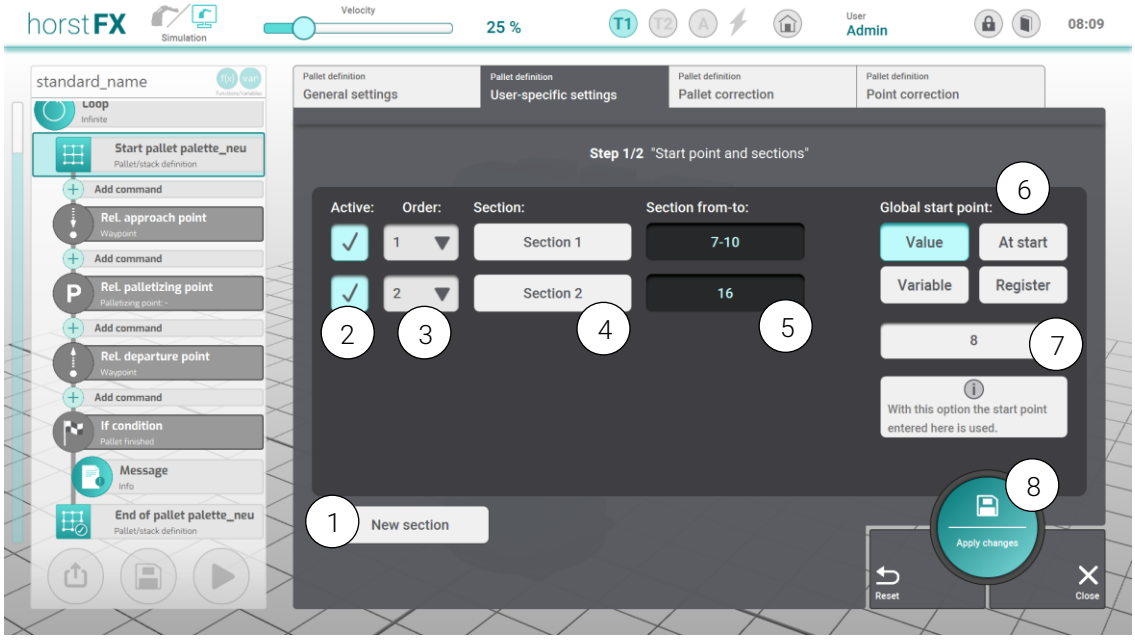


Fig. 6-39: Action window – User specific settings (step 1)

No.	Description
1	**New section** Button – adds another subsection
2	**Active** checkbox – unticked subsections are not processed by the robot
3	**Order** selection field – Determines the order of processing
4	Name of the subsection

No.	Description
5	Shows points that belong to the section
6	**Global start** buttons – **Value/At start/Variable/Register**
7	**Value** input field – **Value can be put in**
8	Apply changes button -
9	**Reset** button – resets all configuration and selection options in the action window to their default values
10	**Close** button – closes the action window

6.3.12.5 User-specific settings Step 2

After hitting the button **new section** the following screen appears

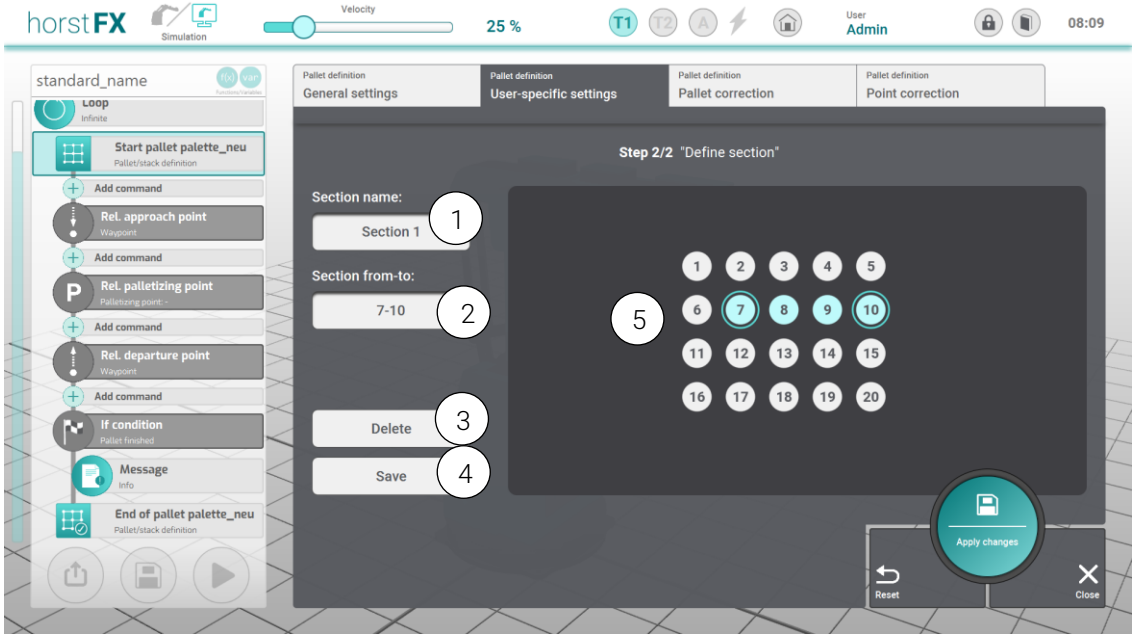


Fig. 6-40: Action window – User specific settings (step 2)

No.	Description
1	**Section name** input field – name can be given
2	**Section range** input field – range of the section can be defined
3	**Delete** button – delets the section
4	**Save** button – saves the section
5	Displays the section in the pallet

6.3.12.6 Pallet correction Step 1

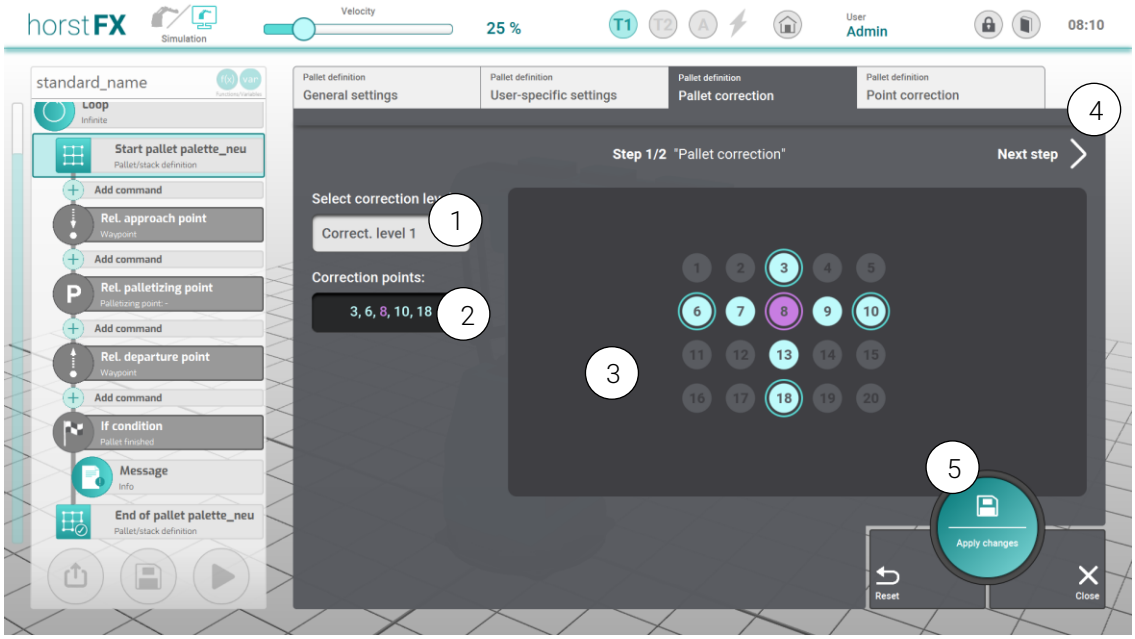


Fig. 6-41: Action window – pallet correction (step 1)

No.	Description
1	**Correction level** selection field – correction algorithm can be selected
2	**Correction point display** – correction points defined by the algorithm are shown
3	Displays the section in the pallet
4	Next step button – Pallet correction step 2 window appears, where corrections points can be adjusted individually
	Apply changes button – saves the changes

6.3.12.7 Pallet correction Step 2

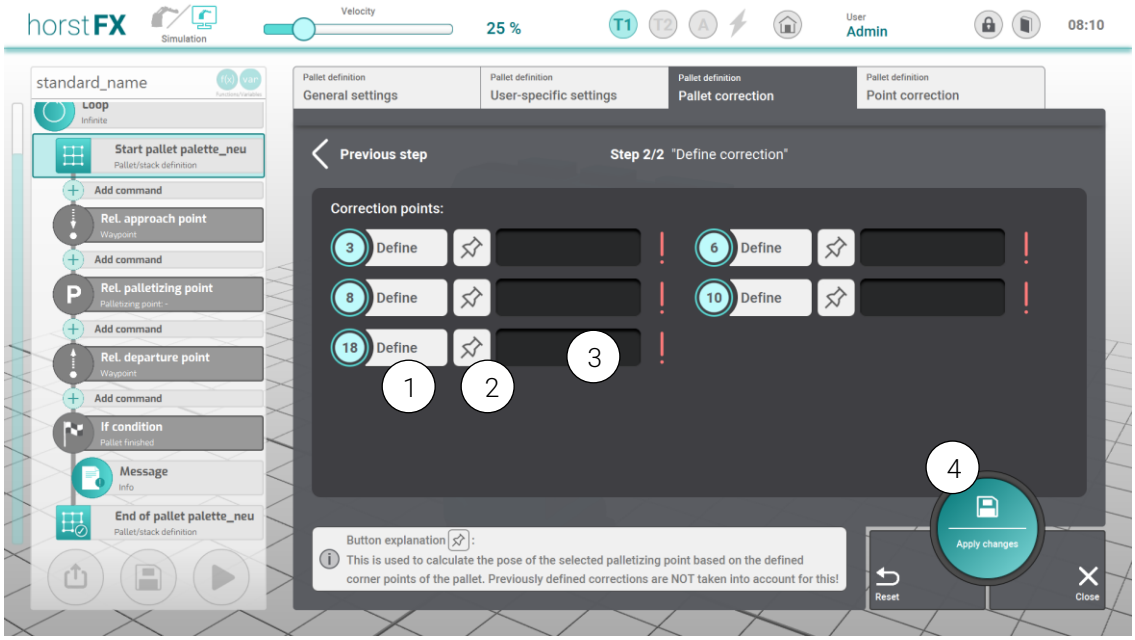


Fig. 6-42: Action window – Pallet correction (step 2)

No.	Description
1	**Define** button – **define destination** window will open (see 6.3.2.3)
2	**Pin** button – coordinates of the point of the original palette are shown
3	Displays the coordinates of the point
4	**Apply changes** button – saves the changes

6.3.12.8 Point correction step 1

Single points can be corrected with point corrections.

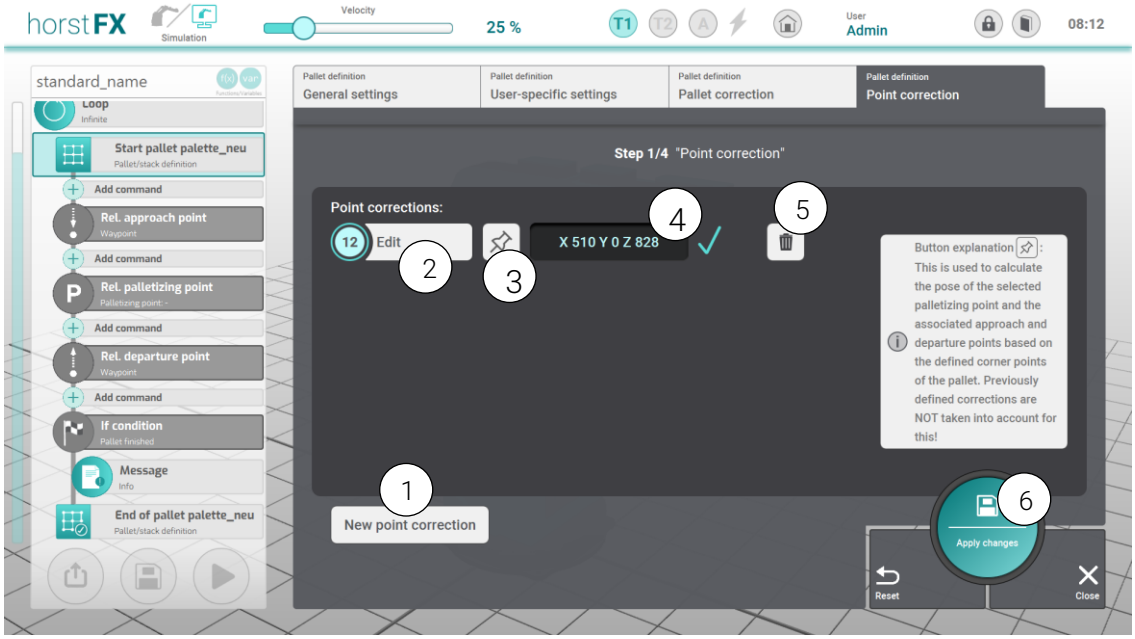


Fig. 6-43: Action window – Point correction (step 1)

No.	Description
1	**New point correction** button – adds a new (undefined) correction point
2	**Edit** button – **define destination** window will open (see 6.3.2.3)
3	**Pin** button – correction points defined by the algorithm are shown
4	Displays the coordinates of the point
5	**Delete** button – Deletes the correction point
6	**Apply changes** button – saves the changes

6.3.12.9 Point correction step 2

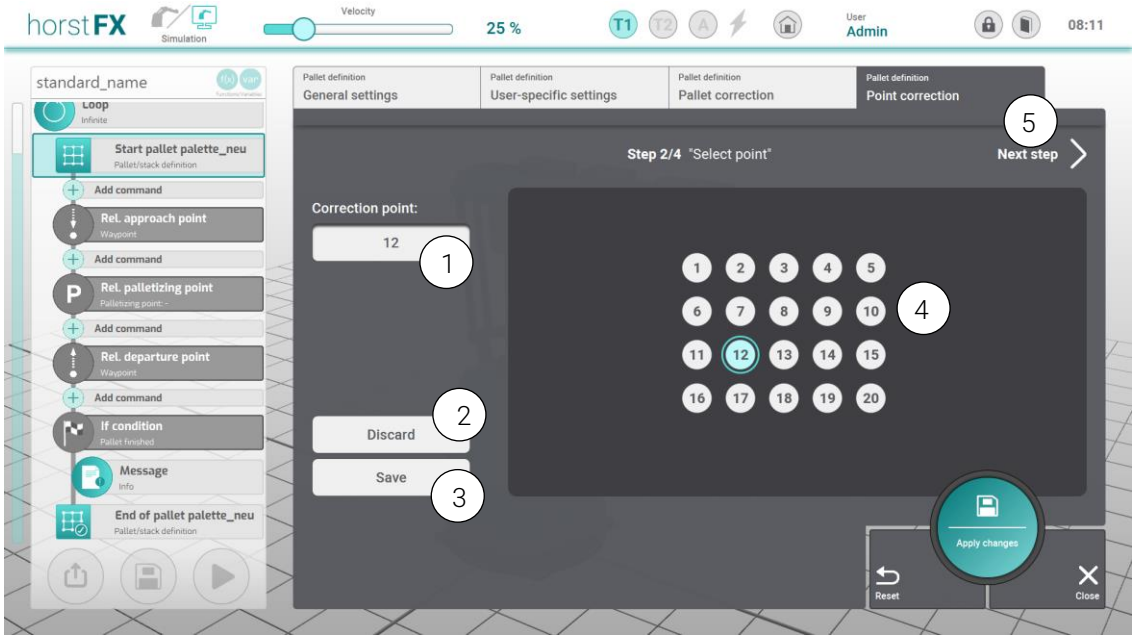


Fig. 6-44: Action window – Point correction (step 2)

No.	Description
1	**Correction point** – correction point(s) can be selected
2	**Discard** button – discards changes
3	**Save** button – saves the correection point
4	Displays the coordinates of the point
5	**Next step** button – Point correction step 3 is opened

6.3.12.10 Point correction Step 3

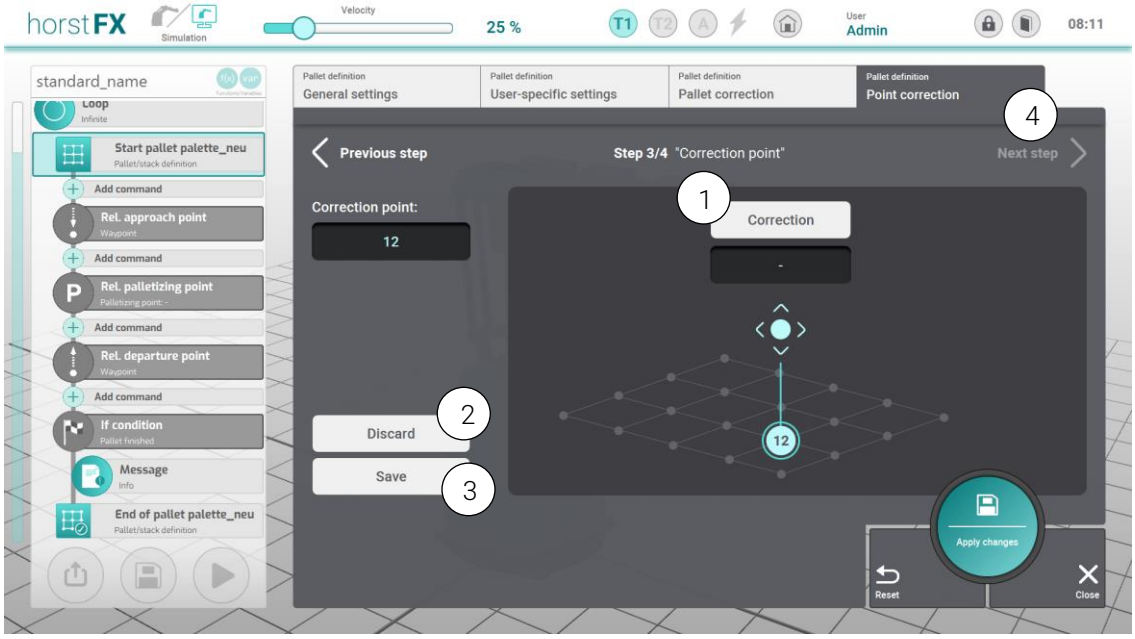


Fig. 6-45: Action window – Point correction (step 3)

No.	Description
1	**Correction** button – **define destination** window will open (see 6.3.2.3)
2	**Discard** button – discards changes
3	**Save** button – saves the correection point
4	**Next step** button – Point correction step 3 is opened

6.3.12.11 Pallet correction Step 4 Approach/Depart

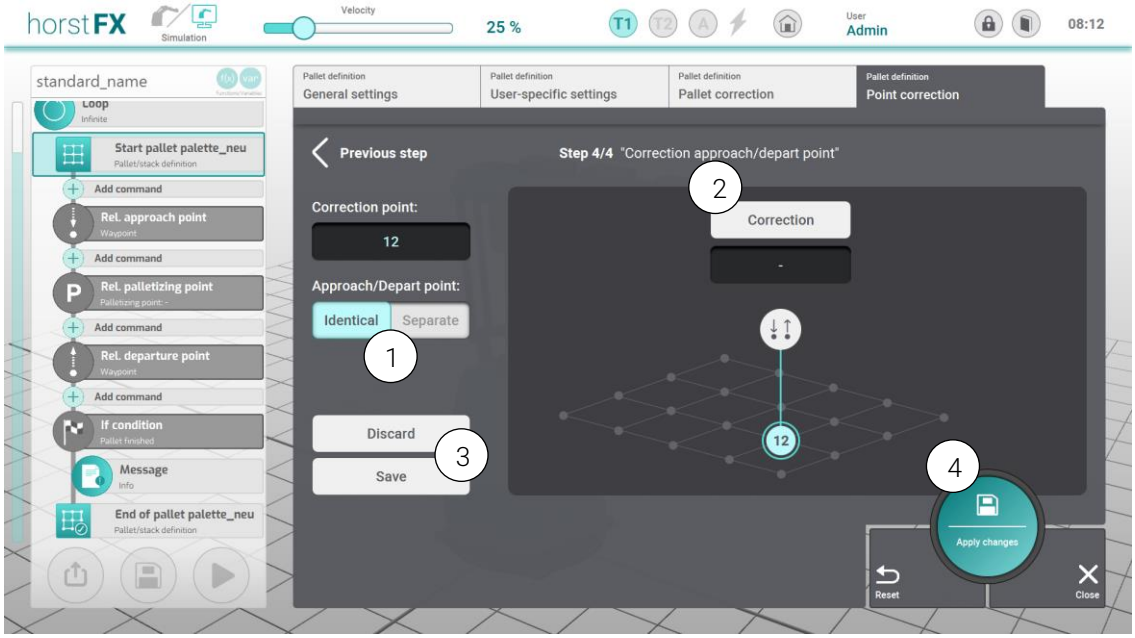


Fig. 6-46: Action window – Point correction (step 4)

No.	Description
1	**Toggle** butto identical/separate – choose if approach and depart are identical or not
2	**Correction** button – **define destination** window will open (see 6.3.2.3)
3	**Discard/Save** buttons – Discards/saves the approach/depart point correction
4	**Apply changes** button – saves the changes

6.3.12.12 Program block Rel. approach/palletizing/departure point

These three special program blocks are only available in the grouped program block of the **Pallet action**. They can be neither deleted nor moved. The corresponding action window can only be called up for this action via the editing menu (see section 6.3.28).

All three actions are similar to the **Relative waypoint action** (see section 6.3.2.6). There are simply fewer configuration and selection options (cf. Fig. 6-47), as well as no possibility to define a relative destination, since this is calculated automatically for the approach, palletizing, and departure point via the pallet definition.

Unlike the program block *Rel. palletizing point*, the two program blocks *Rel. approach point* and *Rel. departure point* have an additional option. Using the **Orientation of the point** toggle button, the *Adapted* option can be selected here, which adapts the orientation of the approach or departure point to the orientation of the respective palletizing point, which is not the case with the *Standard* option.

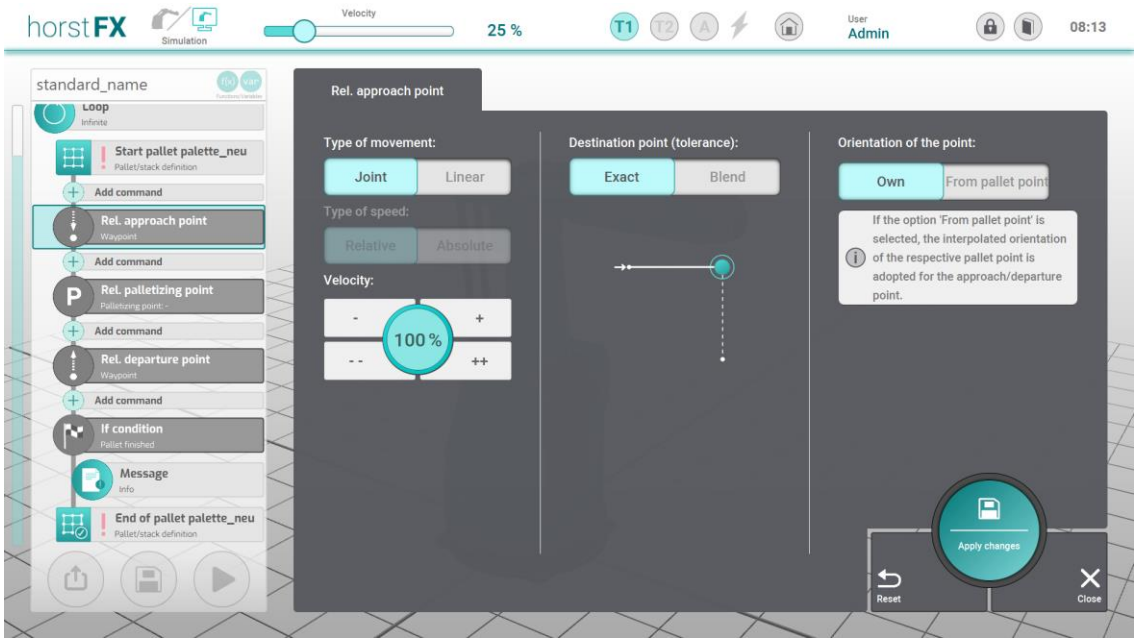


Fig. 6-47: Action window – Rel. palletizing point

6.3.12.13 If-condition program block

This special program block is only available in the grouped program block of the Pallet action. It cannot be deleted, edited, or moved. The corresponding action window can only be called up for this action via the editing menu (see section 6.3.28).

This action is similar to the If-condition action (see section 6.3.9), but the condition is fixed. It is checked whether the last palletizing point, i.e. the complete pallet, has been processed. If this is the case, a message is displayed by default indicating that the pallet has been processed.

6.3.13 Message action

By selecting the **Message action** in the **action selection area**, the corresponding action window appears. A new program block *Message* is created in the program tree.

Messages can be used to draw the operator's attention to something at certain points in the program, for example.

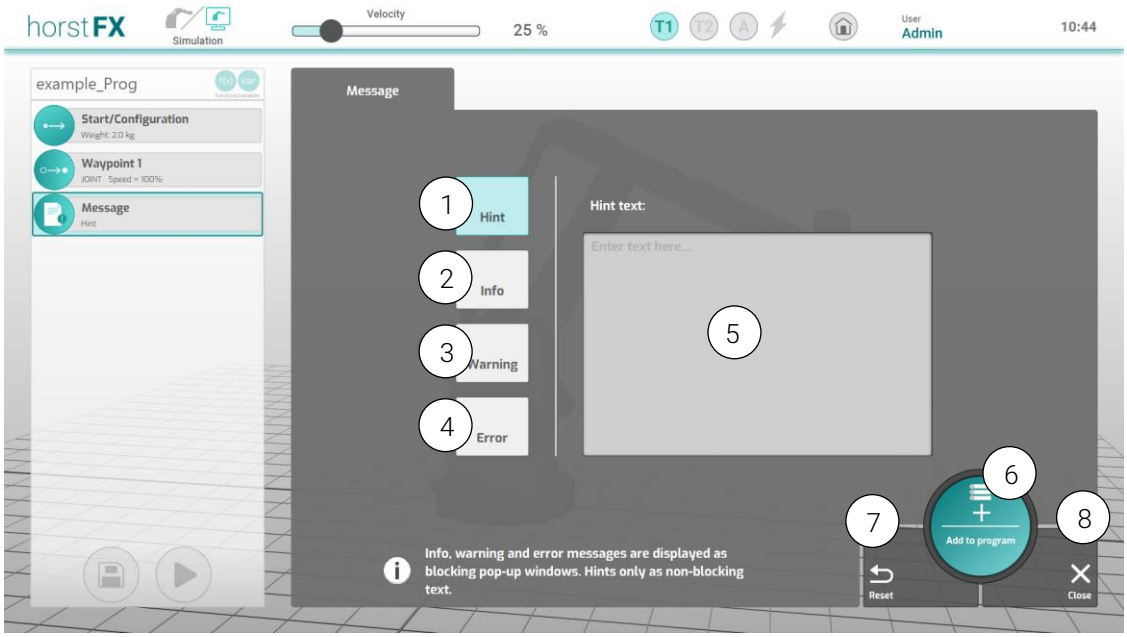


Fig. 6-48: Action window – Message

No.	Description
1	**Hint** button – selection of the message type "Note"
2	**Info** button – selection of the message type "Info"
3	**Warning** button – selection of the message type "Warning"
4	**Error** button – selection of the message type "Error"
5	**Hint text** input field – for entering the text of the note/message
6	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
7	**Reset** button – resets all configuration and selection options in the action window to their default values
8	**Close** button – closes the action window

A message appears on the display for a certain amount of time without blocking the program execution. For the other three message types, a blocking pop-up window appears, which must be confirmed before program execution can continue.

There are four different message types:

- the note message (1),
- the info message (2),
- the warning message (3) and

– the error message (4).

A note message is displayed without a pop-up window and is not blocking.

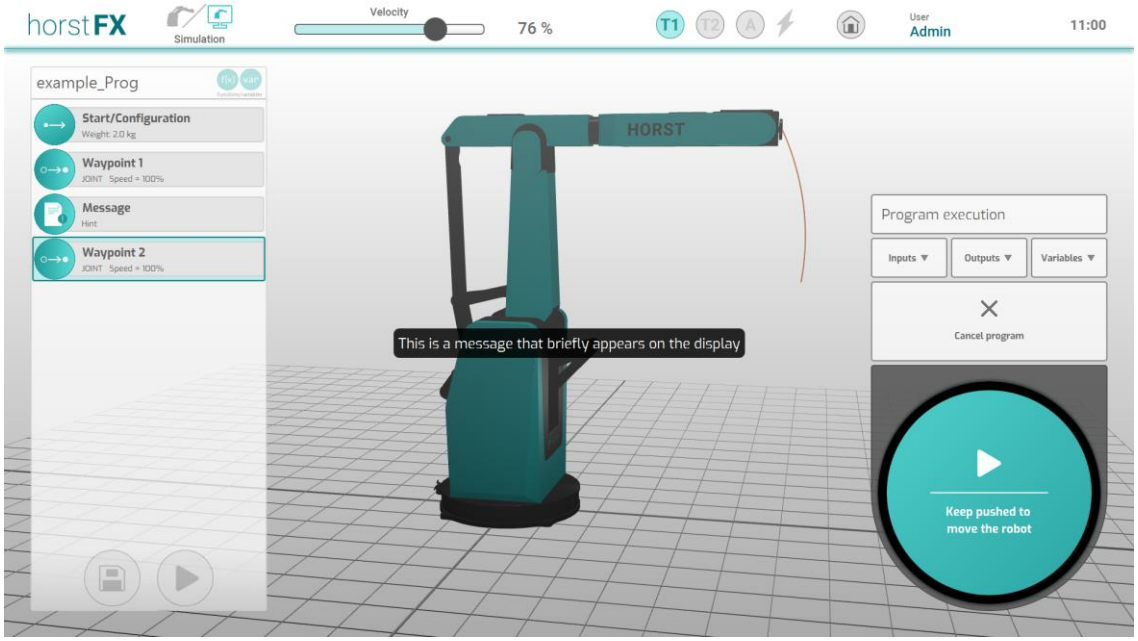


Fig. 6-49: Displaying a note message

The difference between the other three messages is only in the appearance of the pop-up window. When any of these three messages are displayed, the program execution pauses, in contrast to the note message. A pop-up window appears on the display with the entered text. Tapping the **OK (continue)** button will close the pop-up window and continue the program execution.

Fig. 6-50: Pop-up windows – message types

No.	Description
1	**Info message** pop-up window
2	**Warning message** pop-up window
3	**Error message** pop-up window

6.3.14 Check area action

By selecting the **Check area action** in the **action selection** area, the corresponding action window appears. A new grouped program block *Check area (name)* is created in the program tree.

The **Check area action** is similar in functionality to the **If-condition action** (see section 6.3.9). The content of a **Check area action** is executed during program execution only if a certain condition is

met. In this case the conditions are that the current TCP is in a defined cuboid or individual axes are in defined areas. If this does not happen, all included actions will be skipped.

There are three ways to define an area. These are in the form of a cuboid, defined either by all four corners (CUBOID_CORNERS) by one corner and distances (CUBOID_XYZ), or by axis values (JOINTS).

6.3.14.1 CUBOID_CORNERS: cuboids (corners)

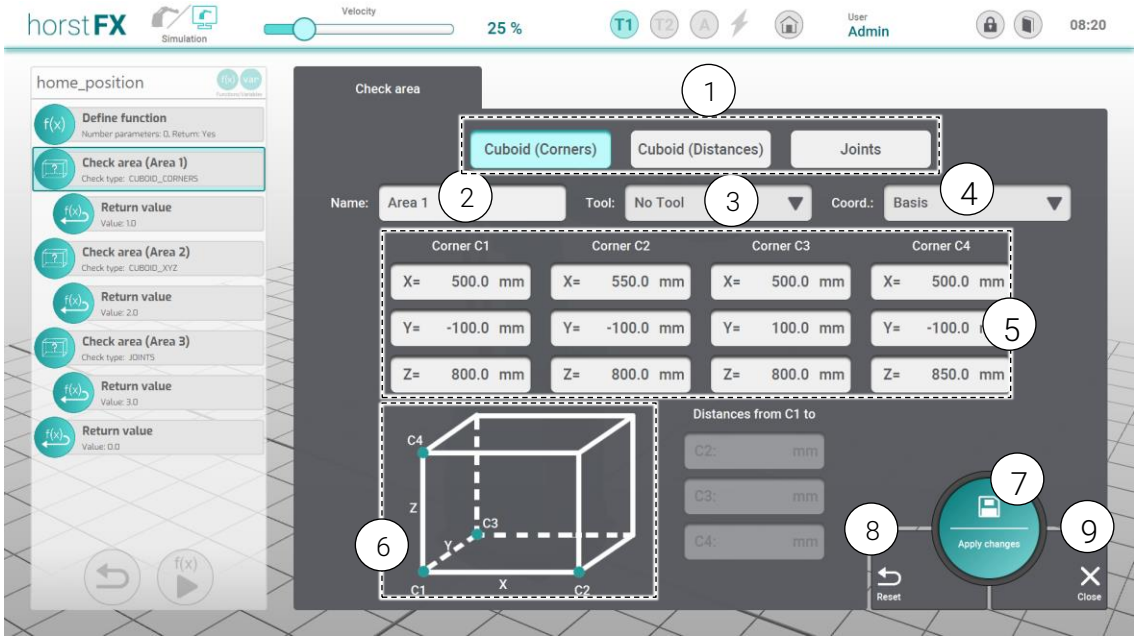


Fig. 6-51: Action window – Check area – Cuboid (corners)

No.	Description
1	**Method of area checking** selection – for selecting the method (Cuboid (corners), Cuboid (Distances), Joints Depending on the selection, the information needed at (5) changes.
2	Enter **area name** – enter the area name
3	**Tool selection** field - selection of the tool with which the test will be performed. (can only be selected if several tools are available in the program, see section 6.8).
4	**Coordinate system selection** field - selection of the coord. Syst. with which the test will be performed.
5	**Corners** input field – for entering the X, Y, and Z coordinate values of all four corner points
6	**Sketch** display – sketch illustrating how the corners and distances are laid out
7	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
8	**Reset** button – resets all configuration and selection options in the action window to their default values
9	**Close** button – closes the action window

6.3.14.2 CUBOID_XYZ: cuboids (distances)

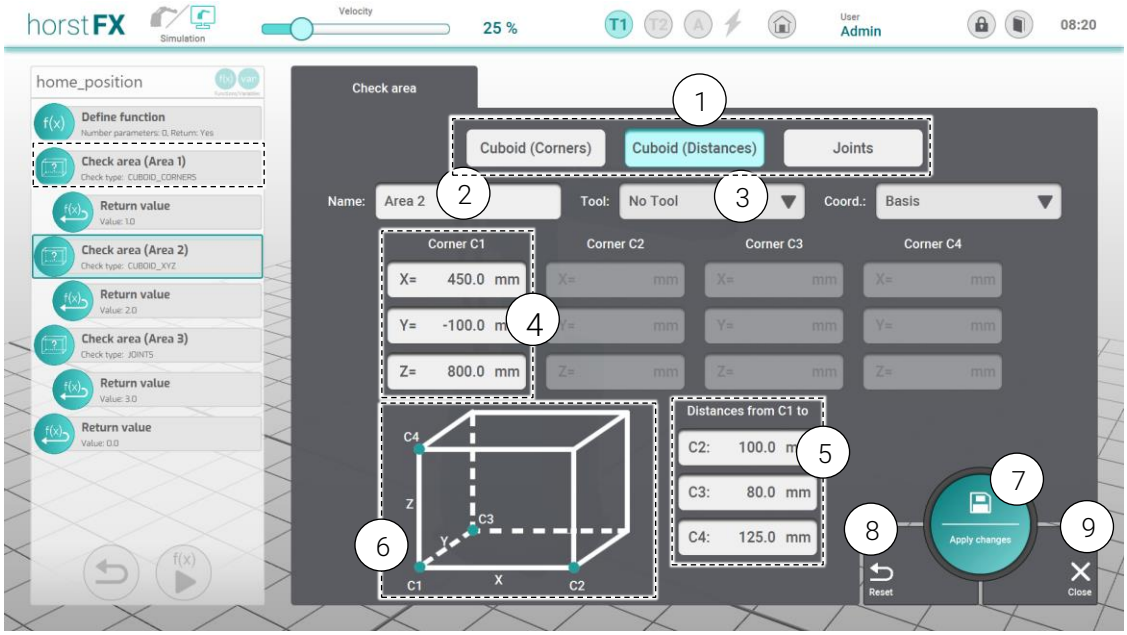


Fig. 6-52: Action window – Check area – Cuboids (distances)

No.	Description
1	Selection Buttons
2	Enter **area name** – enter the area name
3	**Tool selection** field - selection of the tool with which the test will be performed. (can only be selected if several tools are available in the program, see section 6.8).
4	**Corner C1** input field – for entering the X, Y, and Z coordinate values of the corner point
5	**Distances from C1** input field – for entering distances from corner point 1 to the other three corner points
6	**Sketch** display – sketch illustrating how the corners and distances are laid out
7	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
8	**Reset** button – resets all configuration and selection options in the action window to their default values
9	**Close** button – closes the action window

6.3.14.3 JOINTS: axes

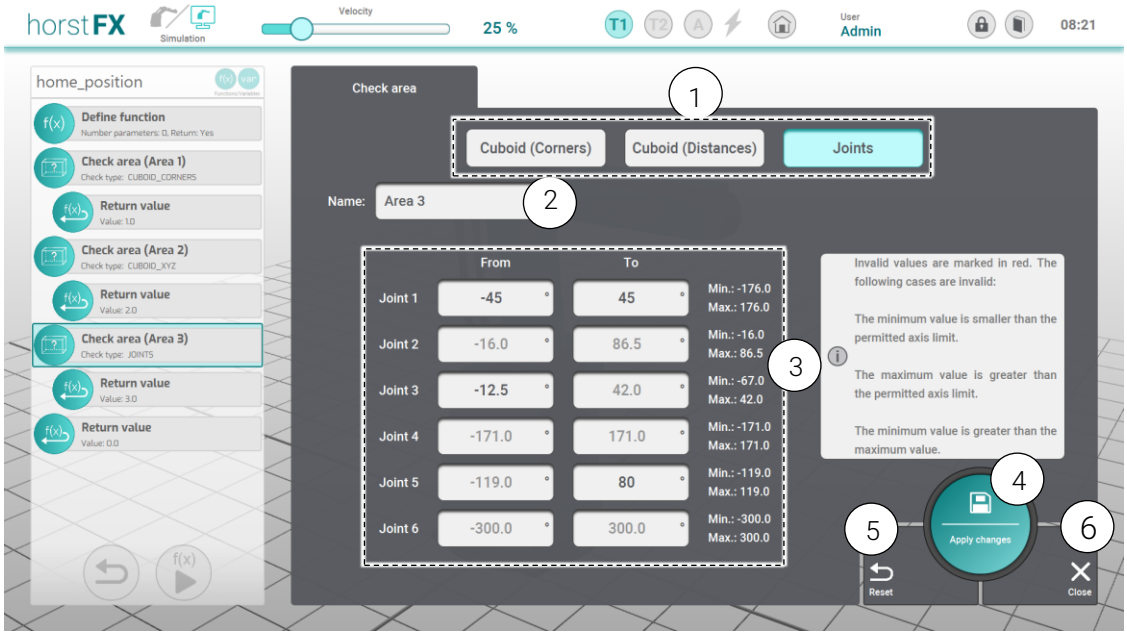


Fig. 6-53: Action window – Check area – Axes

No.	Description
1	
2	Enter **area name** – enter the area name
3	**Joint values** input field – for entering axis values (if a text field is left empty, the corresponding minimum or maximum value of the respective axis is automatically used)
4	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
5	**Reset** button – resets all configuration and selection options in the action window to their default values
6	**Close** button – closes the action window

6.3.15 Set weight action

By selecting the **Set weight action** in the **action selection area**, the corresponding action window appears. A new program block *Set weight* is created in the program tree.

All **Waypoint** and **Relative waypoint** actions performed below use the weight set in that action unless a separate weight is configured in the **Waypoint** or **Relative waypoint** action itself (see sections 6.3.1.2 and 6.3.3).

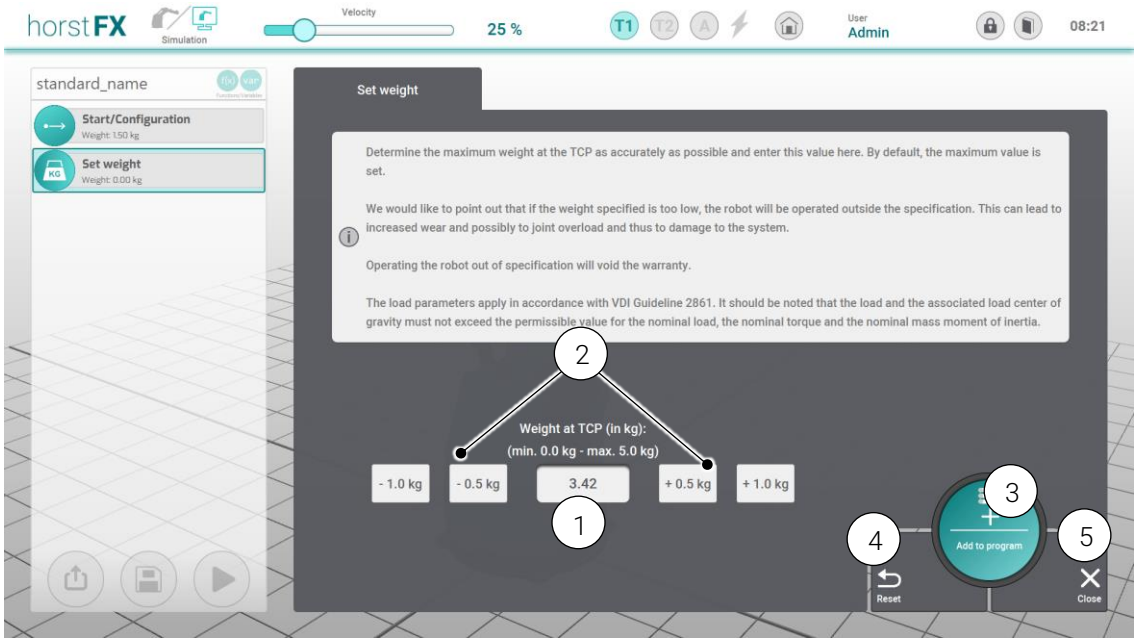


Fig. 6-54: Action window – Set weight

No.	Description
1	**Weight at TCP** input field – for entering the weight currently set at the TCP
2	**Weight change** buttons – for changing the set weight at the TCP in increments of 0.5 kg or 1.0 kg (within the valid range)
3	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
4	**Reset** button – resets all configuration and selection options in the action window to their default values
5	**Close** button – closes the action window

6.3.16 Set tool action

By selecting the **Set tool action** in the **action selection area**, the corresponding action window appears. A new *Tool change* program block is created in the program tree.

The **Set tool action** is only relevant in programs with more than one tool (see section 6.8).

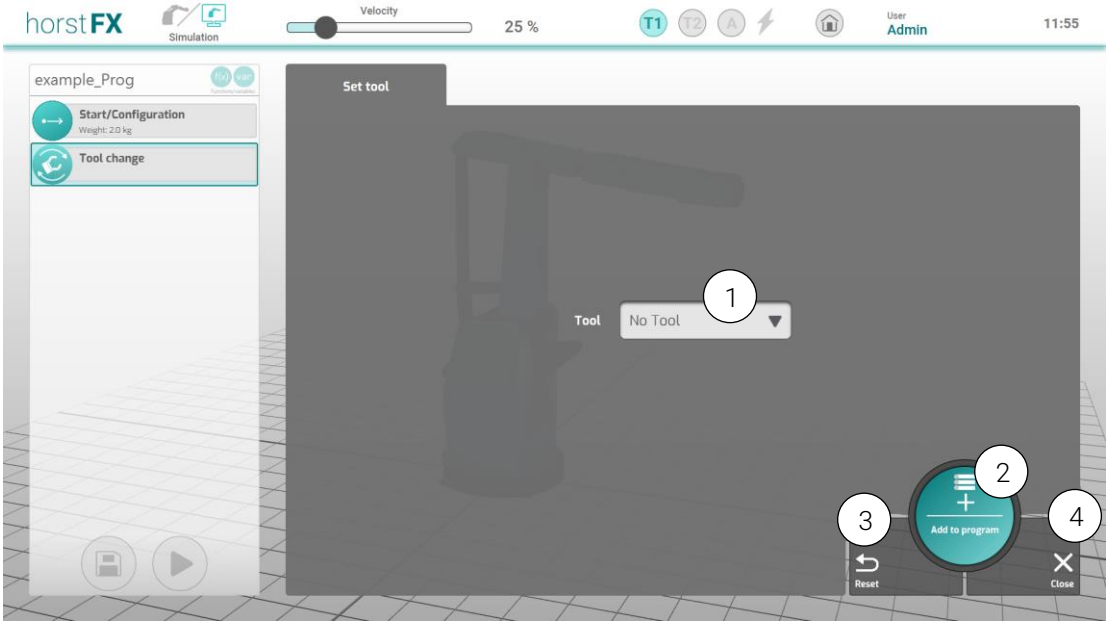


Fig. 6-55: Action window – Set tool

No.	Description
1	**Tool** selection field – for selecting the tool to be switched to
2	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
3	**Reset** button – resets all configuration and selection options in the action window to their default values
4	**Close** button – closes the action window

6.3.17 Change Coordinate system action

By selecting Action **Change Coordinate system** in the **action selection area**, the corresponding action window appears. A new programme component *Change Coordinate system* is created in the programme tree.

For use and effects of the action **Change Coordinate system** see section 6.9.

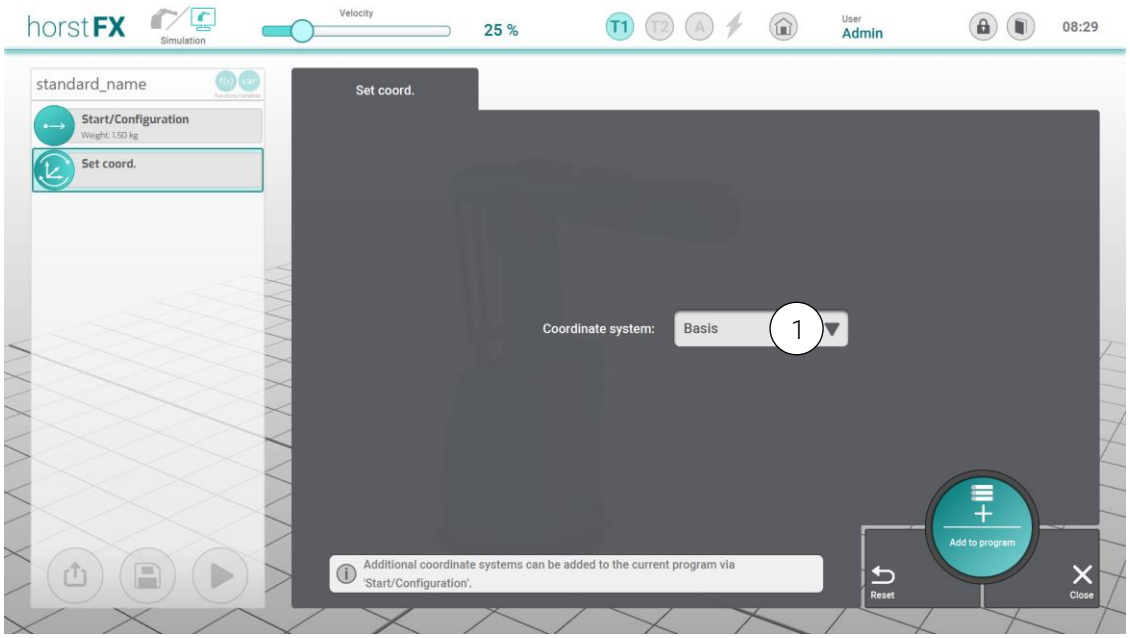


Fig. 6-56: Action window – change coordinate system

Item	Description
1	Selection field **Coordinate system** – Selection of the coordinate system to be changed to

6.3.18 Command Polygon chain

A polygon chain is a series of lines through predefined interpolation points.

The interpolation points are defined in the Settings tab.

The interpolation points can be defined in the Settings tab using the Define target point window (6.3.2.3), as can the contour radius valid for all points. The toggle button can be used to switch between the coordinates of the support points and their preview in space.

Tabs 2-4 offer further options, see 6.3.2.

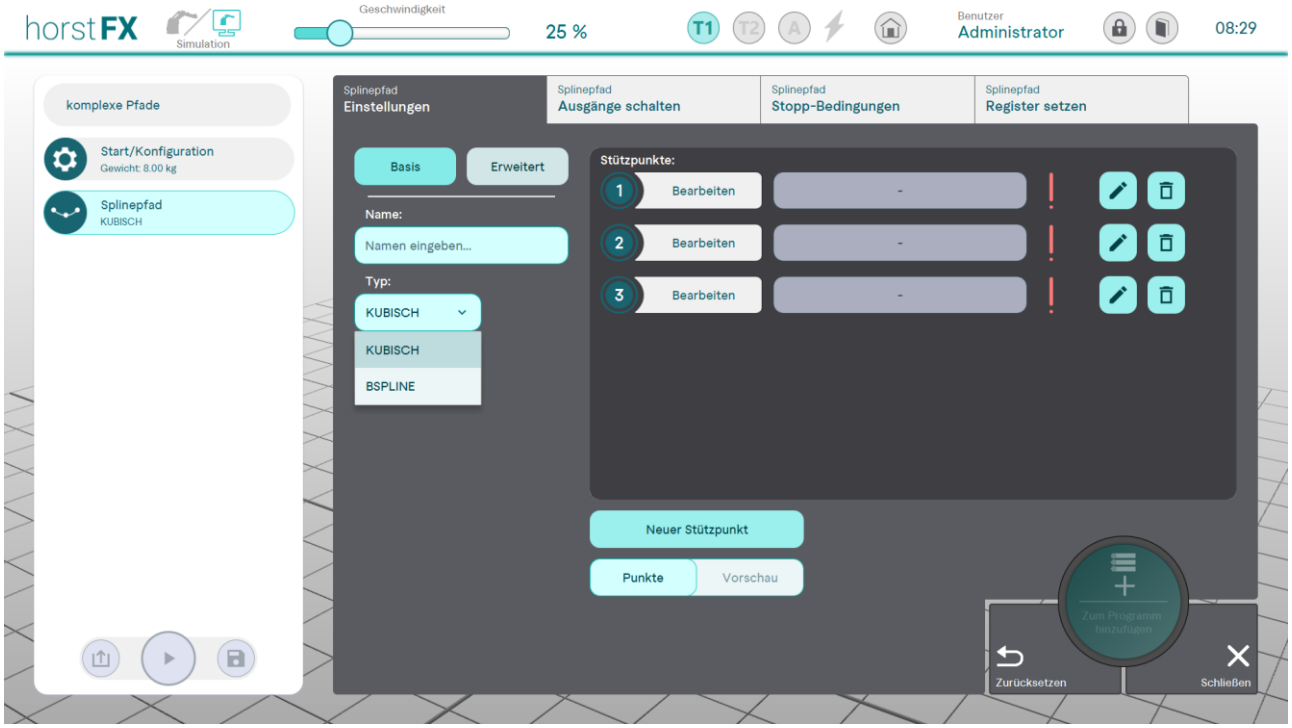
6.3.19 Command Spline path (complexer path)

With the Spline path action, it is possible to execute a movement based on a defined curve, which is advantageous for avoiding collisions or for gluing applications, for example. A normal point-to-point movement (6.3.2), on the other hand, only defines the end position.

A spline is a function that consists of polynomials (degree n) in pieces and for which certain conditions apply at the points where two pieces meet (interpolation points). Depending on the degree of the polynomials and the conditions set, different types of splines are obtained.

(Free) cubic spline: 3rd degree polynomials, 2 times continuously differentiable. Always goes exactly through the support points!

B-spline (order k): Polynomials of degree k-1. Usually does not pass through the grid points!



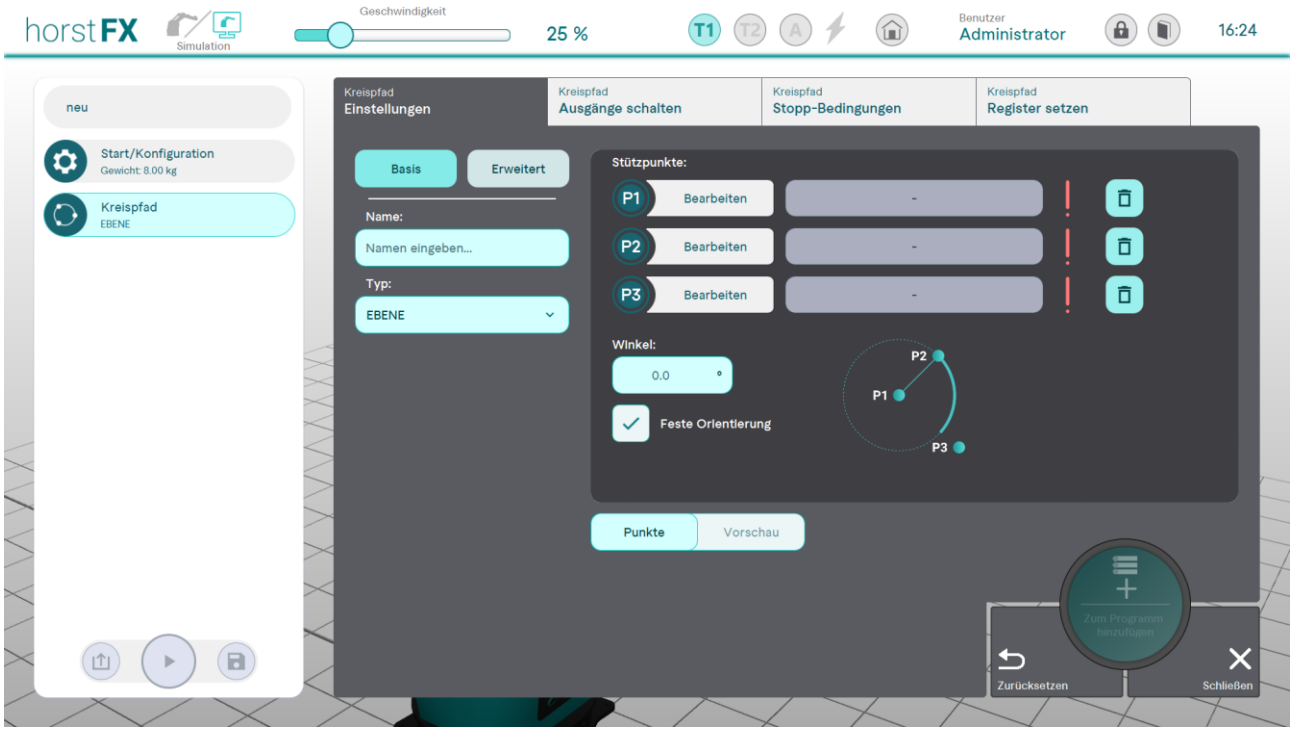
Tabs 2-4 offer further options, see 6.3.2.

6.3.20 Command Circular path (complex path)

The type of circular path can be set in the Settings tab. A circular path is always defined using 3 points.

Variant 1 - Plane:

P1 forms the centre of the circle. P2 is the starting point of the circular movement and the length of the distance P1P2 thus determines the radius of the circle. Finally, point P3 defines the plane of the circle. The orientation of the effector at the start of the circular movement is given by P2. If a "fixed orientation" is selected, the orientation remains constant during the circular movement, otherwise it can change." (The orientations of P1 and P3 are ignored.)



Variant 2 - Arc:

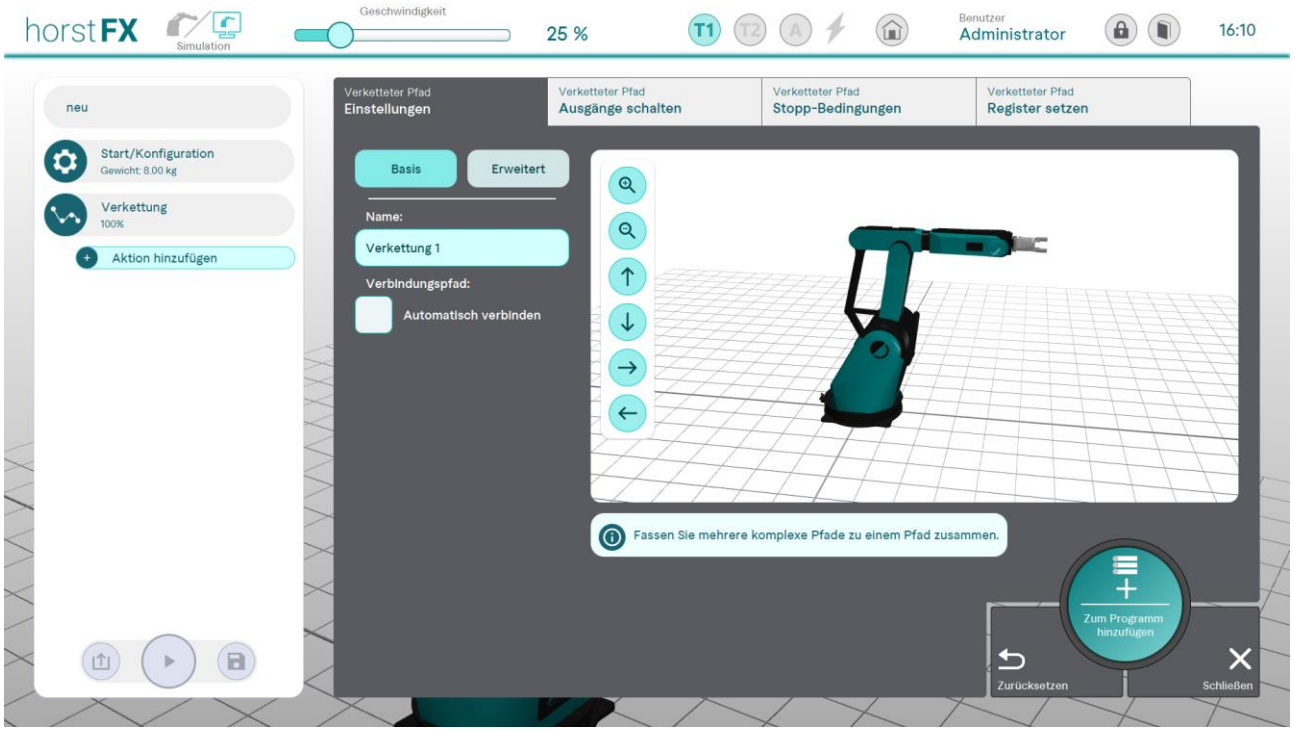
An arc is created through P1, P2 and P3. The circular movement starts in P1 and continues via P2 to P3. The start orientation of the effector is given by P1. (The orientations of P2 and P3 are ignored). The angle indicates how much of the circle is to be travelled. Accordingly, the circular path is travelled in the opposite direction if the sign is negative.

Tabs 2-4 offer further options, see 6.3.2.

6.3.21 Command concatenated path (complex path)

Linked paths connect several complex paths with each other.

Tab 1: Basic settings



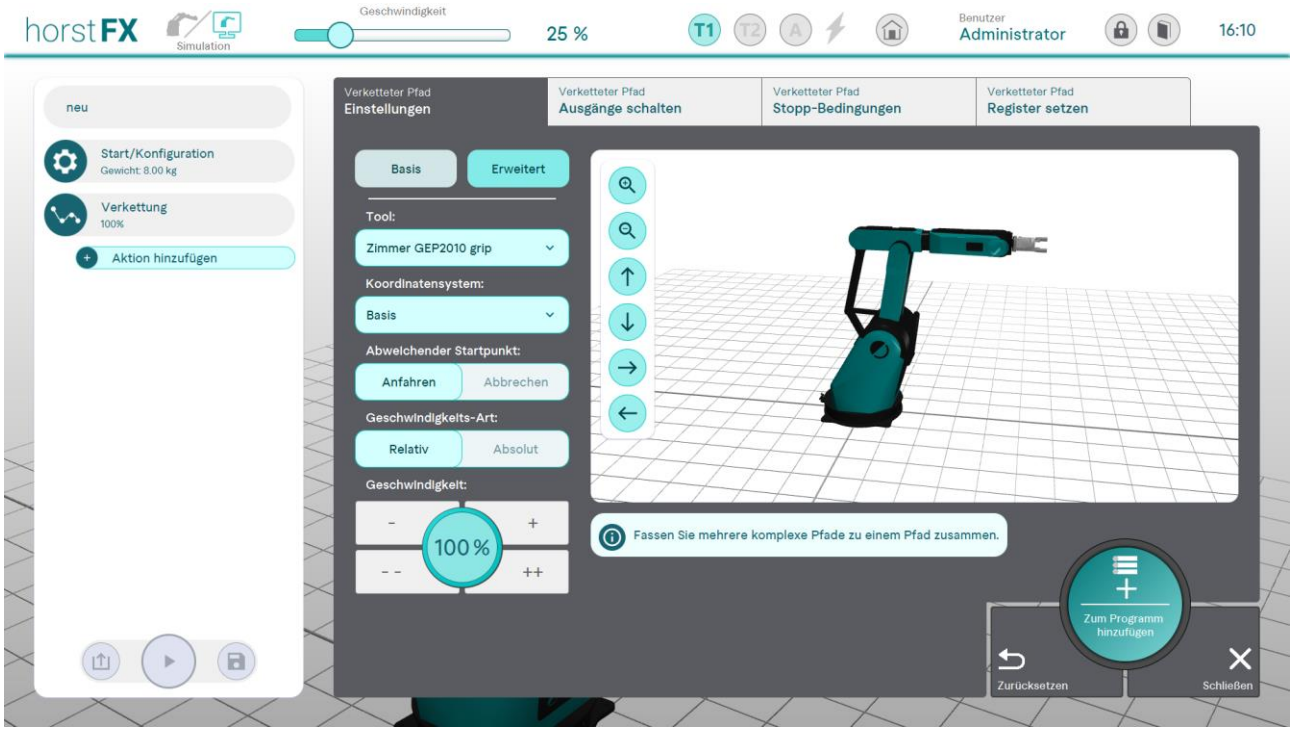
If the "Connect automatically" checkbox is not activated, the path can only be executed if the end point of a partial path is the same (except for a small tolerance) as the start point of the following partial path for all partial paths.

If "Connect automatically" is activated, those partial paths for which the end point and the following start point do not match are automatically connected by a cubic Hermite spline (if they already match, they are not connected). The automatically generated connection path is uniquely determined by the two partial paths and cannot be manipulated further.

Extended

The tool and the coordinate system can also be set here. The behaviour can also be defined in the event of a different starting point, abort or start.

Speed settings can also be made.



6.3.22 Command Try-Catch

This action is relevant for executed movements that are specified by external devices, e.g. camera or PLC.

If an error occurs, e.g. if a position specified by the camera cannot be reached, this error can be intercepted in the robot programme. All actions within the try-catch instruction are checked and only executed if they are possible.

Try

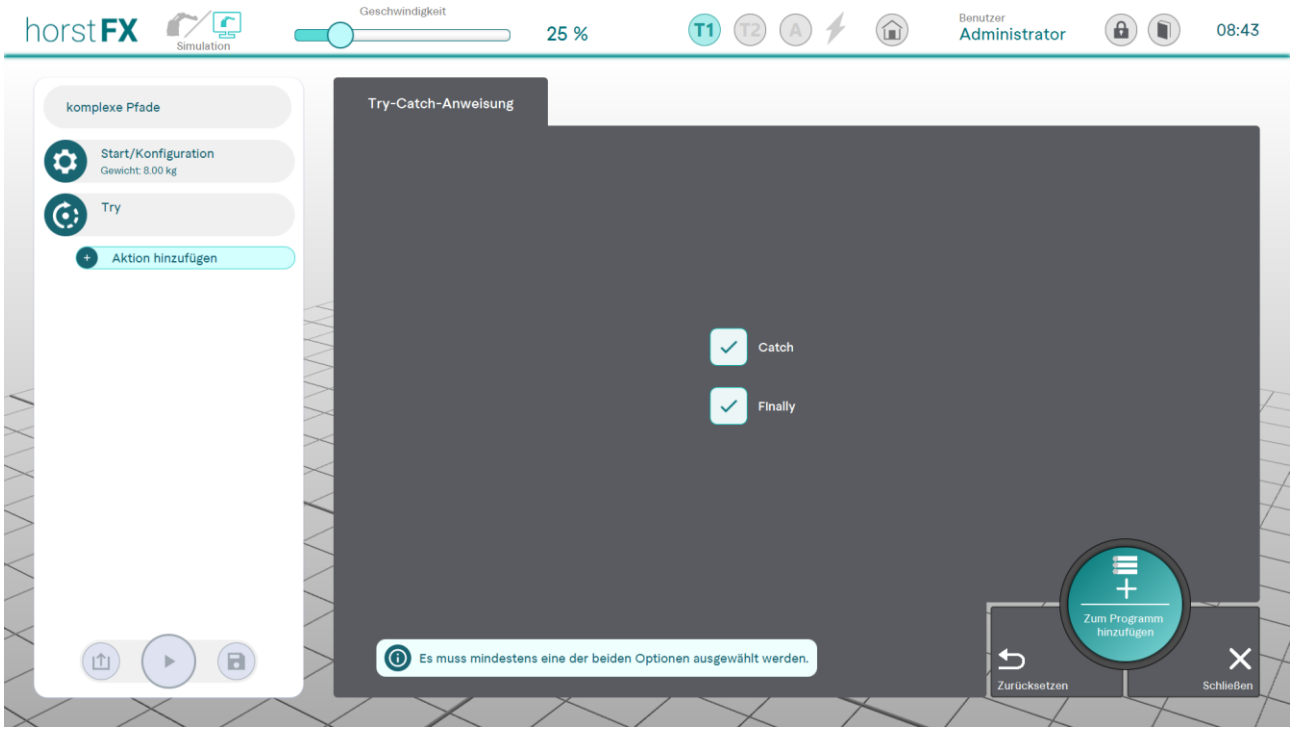
Actions within the try block are executed completely or until an error occurs. If an error occurs, the catch block is activated.

Catch

In the Catch block, the actions are only executed if an error occurs within the try.

Finally

The actions in the Finally block are always executed regardless of the events in the Try and/or Catch block. This can be important in order to end certain functions cleanly.



6.3.23 Return value action

By selecting the **Return value action** in the **action Selection area**, the corresponding action window appears. A new program block *Return value* is created in the program tree.

The **Return value action** can only be used within the processing of a function (see section 6.6) for which the *Return value* toggle button is activated (see section 6.6.1).

Return values are used to save a defined value within or at the end of a function, which is assigned to a variable. This variable can be used for the rest of the program.

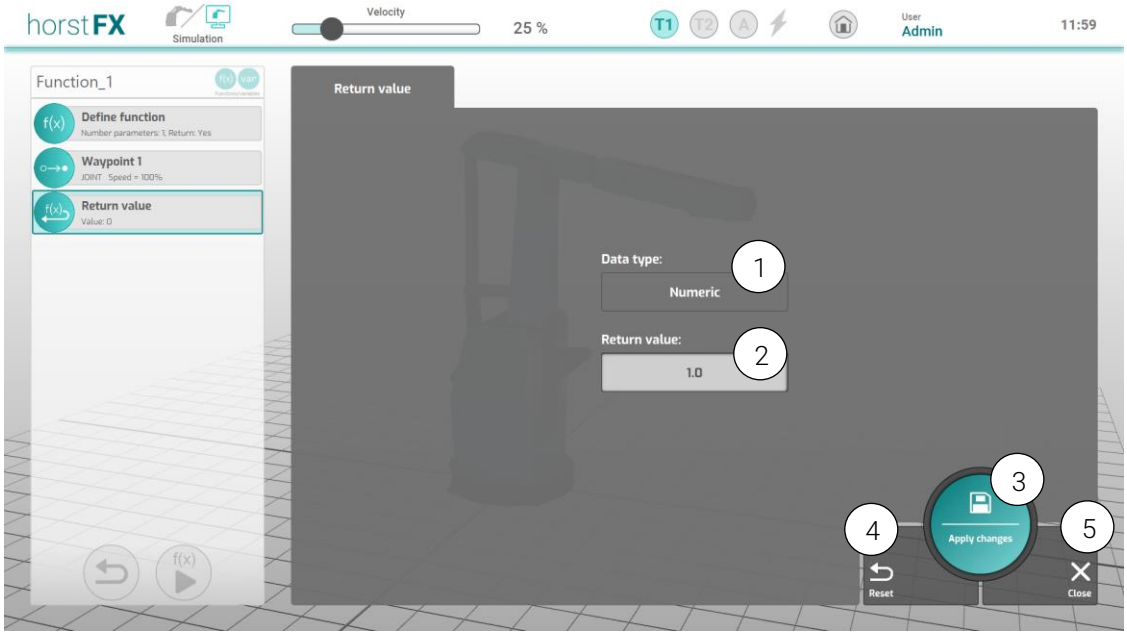


Fig. 6-57: Action window – Return value

No.	Description
1	**Data type** display – displays the data type of the return value
2	**Return value** input field – for entering the return value

No.	Description
3	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
4	**Reset** button – resets all configuration and selection options in the action window to their default values
5	**Close** button – closes the action window

6.3.24 Record data action

By selecting the **Record data action** in the **action selection area**, the corresponding action window appears. A new program block *Record data* is created in the program tree.

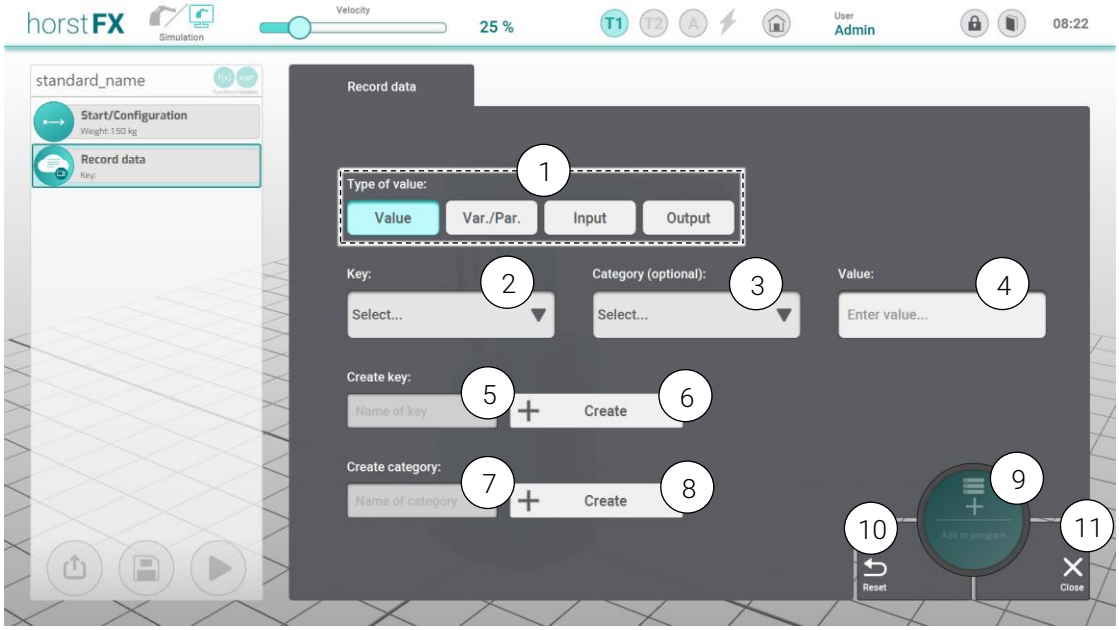


Fig. 6-58: Action window – Record data

No.	Description
1	**Type of value** selection – for selecting the value type (value, variable/parameter, input, or output) Depending on the selection, either an input or a selection field appears at (4).
2	**Key** selection field – for selecting a key for which a value is to be recorded
3	**Category** selection field – for the optional selection of a category for the key
4	**Value** input/selection field – for entering or selecting the value
5	**Key name** input field – for entering a new key name
6	**Create** button – for creating the new key name
7	**Category name** input field – for entering a new category name
8	**Create** button – for creating the new category name
9	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed

No.	Description
10	**Reset** button – resets all configuration and selection options in the action window to their default values
11	**Close** button – closes the action window

6.3.25 Create folder action

Selecting the **Folder action** in the **action selection area** will display the corresponding action window. A new grouped program block *Folder* is created in the program tree.

Folders are used to group multiple actions together. This provides better clarity for a longer sequence of actions. Furthermore, the entire folder can be moved so that not every action in it has to be moved individually.

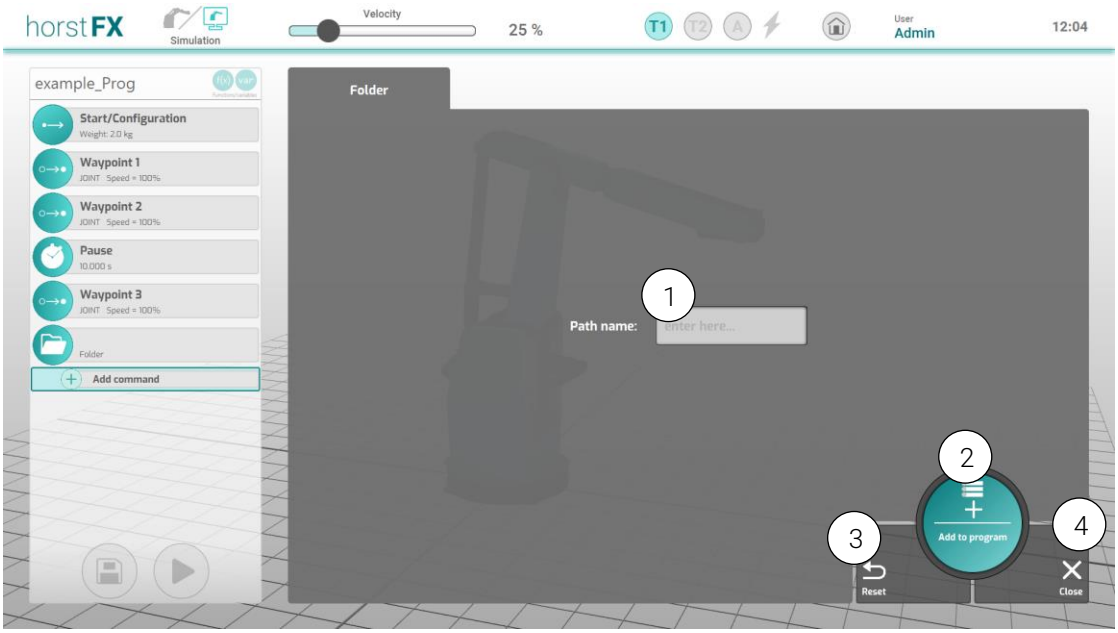


Fig. 6-59: Action window – Folder

No.	Description
1	**Path name** input field – for naming the folder
2	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
3	**Reset** button – resets all configuration and selection options in the action window to their default values
4	**Close** button – closes the action window

6.3.26 Comment action

By selecting the **Comment action** in the **action selection area**, the corresponding action window appears. A new program block *Comment* is created in the program tree.

Comments can be used to improve clarity in the program tree or to save a comment/note at certain places in the program tree.

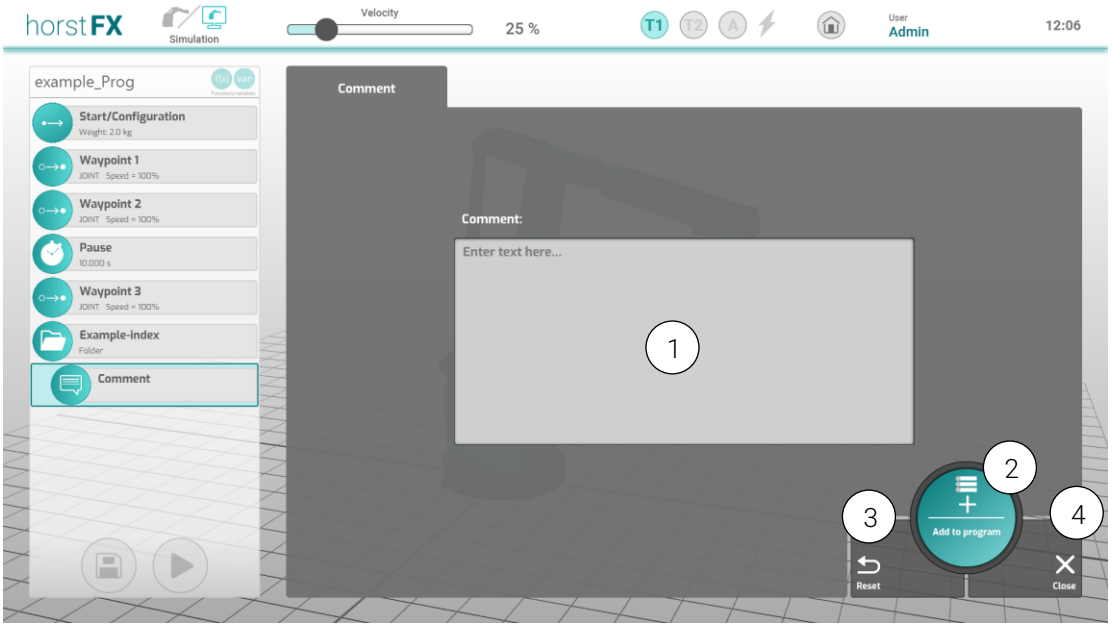


Fig. 6-60: Action window –Comment

No.	Description
1	**Comment** input field – for entering the comment text
3	**Add to program** button – the set configuration and selection options in the action window are applied, and the action window is closed
4	**Reset** button – resets all configuration and selection options in the action window to their default values
5	**Close** button – closes the action window

6.3.27 Manual control menu

By selecting **Manual control** in the programming view, the **Manual control** menu appears.
Here it is possible to move the robot freely without first selecting a **Waypoint action**.
If the controlled pose of the robot needs to be converted into a waypoint, this can be done using the **Save waypoint** button (3). Tapping this button closes the **Manual control** menu, the action window for the **Waypoint action** is displayed, the robot's pose is directly adopted as the defined destination and a corresponding *Waypoint* program block is added to the program tree.

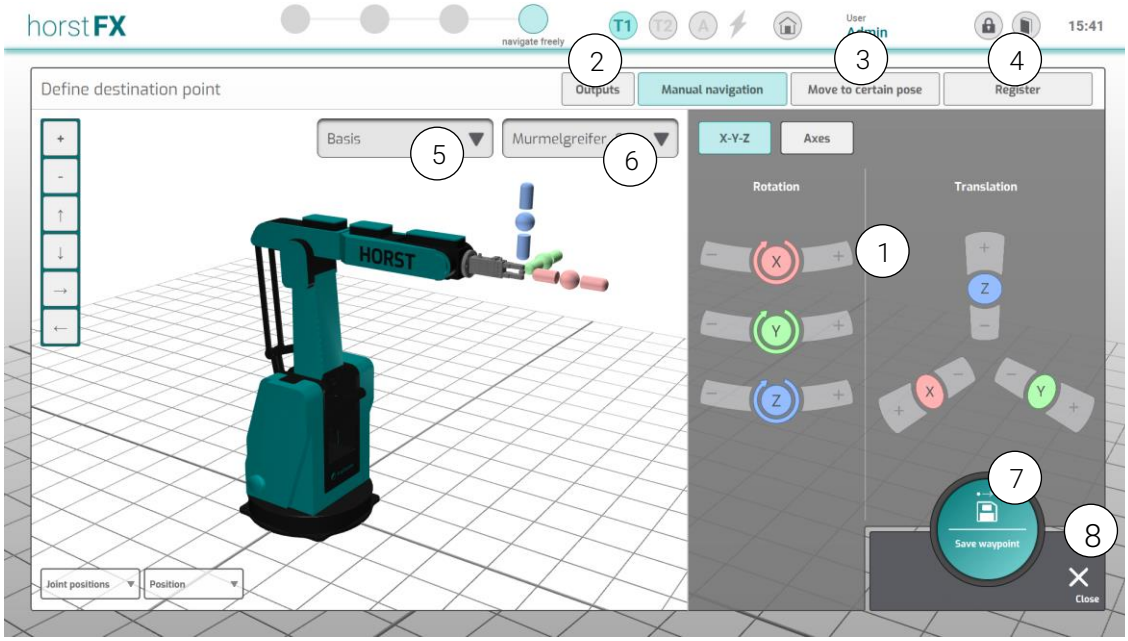


Fig. 6-61: ***Manual control*** *menu*

No.	Description
1	Menus and controls for controlling the robot. Operation is as described in sections 5.1 and 5.2.
2	**Outputs** button – displays the **Manual control (outputs)** menu Operation is as described in section 5.3 (p. 473). All outputs can be switched manually here (e.g. open/close gripper to grasp or release an object). **No** "Switch output" program block can be created via this menu.
3	**Move to certain pose** button – displays the **Manual control (move to specific pose)** menu Operation is as described in section 5.4.
4	**Register** button – displays the **Manual control (register)** menu (see section 5.5)
5	**Coordinate system** selection field – for selecting the coord. Syst. that is used for the movements
6	**Tool** selection field – for selecting the tool whose TCP will be used by the control system A selection can only be made in programs with more than one tool (see section 6.8).
7	**Save waypoint** button – applies the controlled pose of the robot as a destination point in a Waypoint action
8	**Close** button – closes the **Manual control** menu

6.3.28 Editing menu (actions)

In order for the **Editing menu** to appear, the corresponding program block must be pressed and held until the menu is displayed.

The **Editing menu** contains various options which can be applied to the respective program block and thus to the respective action. Some options are only displayed for certain program blocks/actions. As an example, the **Editing menu** of a program block *Waypoint* is used in Fig. 6-62 since all available options are displayed for it.

If an option is selected, another confirmation button ✔ appears first in the option's button. Tapping this button will apply the selected option.

In addition, a program block can be moved in the program tree within this menu.

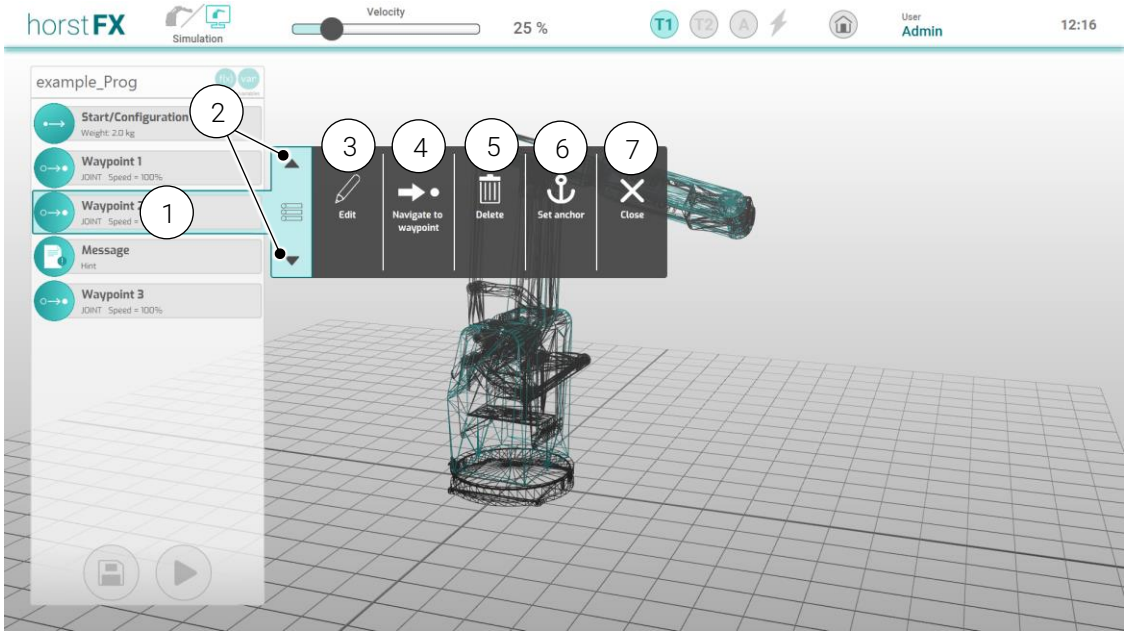


*Fig. 6-62: Selected program block Waypoint with **Editing menu***

No.	Description
1	Selected program block with **Editing menu** displayed
2	Buttons ▲▼ – move the program block in the program tree
3	**Edit** button – for editing the action The corresponding action window opens.
4	**Navigate to waypoint** button – see below: **Approach waypoint menu** section
5	**Delete** button – deletes the program block
6	**Set anchor** button – defines the program block as an anchor If a program block is defined as an anchor, this is indicated by an anchor symbol in the program block. Only one program block can be defined as an anchor in the entire program tree. The functionality of an anchor is discussed in section 6.4.
7	**Close** button – closes the context menu

Navigate to waypoint menu

In the **Navigate to waypoint** menu, you can select whether the destination of the waypoint is to be moved to automatically or manually and whether the movement is to be executed as a **joint** or **linear** movement. By default, the **Automatic** control type is selected and plans a **joint movement** to

the destination. Manual control mode can be used if the planned path cannot be traversed for some reason. It is possible to switch between the two control modes at any time. For example, if there is an obstacle in the way of the planned path, it can be manually bypassed and from this point the rest of the route to the destination point can be approached again automatically.

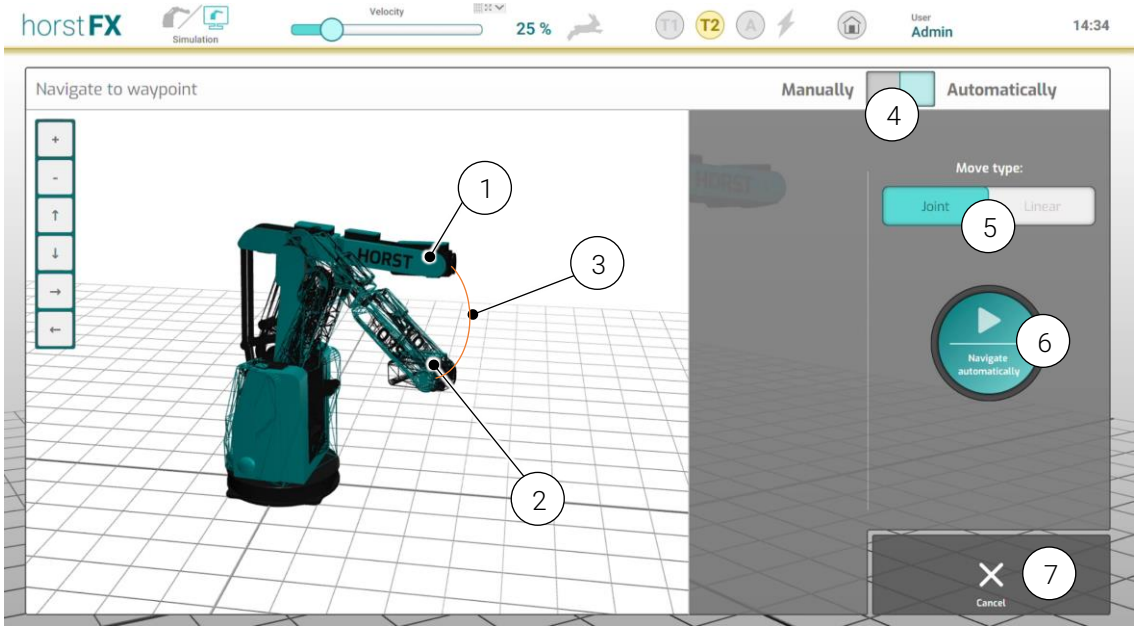


*Fig. 6-63: **Navigate to waypoint** menu*

No.	Description
1	(Standard) robot model view – displays the current pose of the robot
2	Wireframe model view – displays the robot's target pose (based on the destination defined in the waypoint)
3	Path view – displays the planned path from start to destination point
4	**Manual/Automatic** toggle button – for selecting the control mode **Manual** – the robot is controlled manually (see sections 5.1 and 5.2). **Automatic** – the robot moves automatically along the planned path.
5	**Movement type** toggle button - selection of *Joint* or *Linear* movement type
6	**Navigate automatically** button – execution of the robot movement along the planned path
7	**Cancel** button – closes the **Approach waypoint** menu

6.4 Run program

When the button ▶ (Execute) in the programming view is selected, the **Program execution** area appears on the right side of the screen.

WARNING!
Danger of impact and crushing due to robot movement

▶ The functionality of all protection devices must be restored before Automatic operating mode is selected.

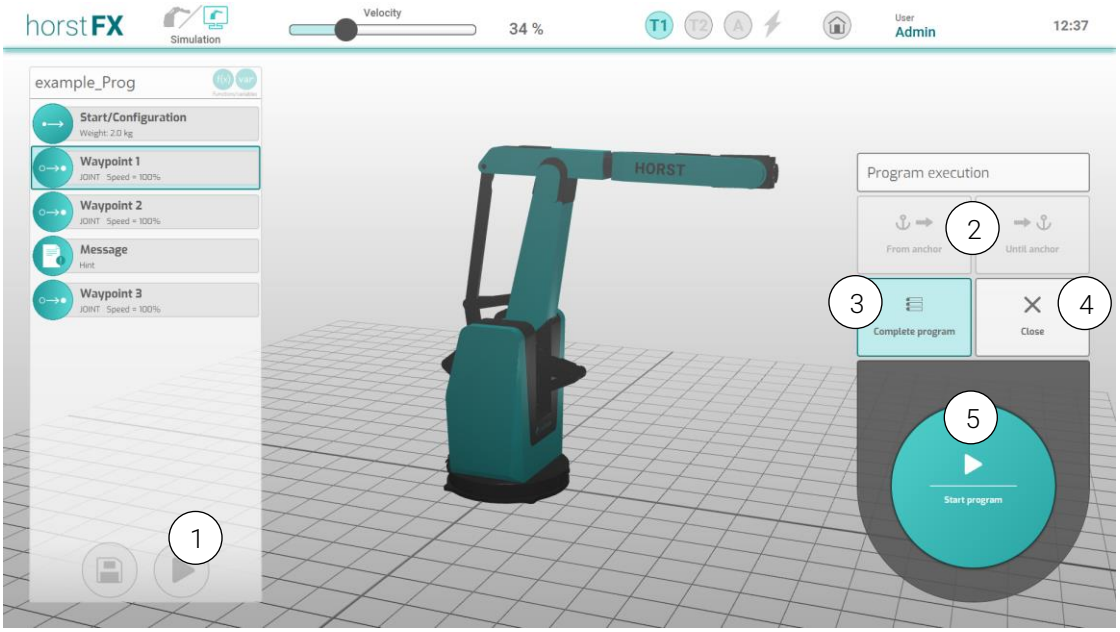


*Fig. 6-64: Programming view – **Program execution** area*

No.	Description
1	▶ Button (Execute) – shows the **program execution** area
2	**From anchor** / **To anchor** buttons – see section 6.4.1
3	**Complete program** button – selects the complete program (counterpart to **From anchor** / **To anchor** buttons)
4	**Close** button – hides the program execution area
5	**Start program** button – starts program execution **Automatic mode** – the **Program execution** area changes and adapts for program execution in **Automatic mode** (see Fig. 6-66). **Teaching mode** – the **program execution** area changes and adapts for program execution in **Teaching mode** (see Fig. 6-65).

6.4.1 From anchor / To anchor functionality

In the **Program execution** area, the **From anchor / To anchor** functionality can be used. However, a prerequisite for this is that a program block is defined as an anchor (see section 6.3.28).

If the **From anchor** functionality is selected, the program execution starts with the action belonging to the defined anchor program block and continues until the end of the program.

If the **To Anchor** functionality is selected, the program execution starts at the beginning of the program and continues up to and including the action belonging to the defined anchor program block.

In both cases, only the program blocks to be executed are shown in the program tree.

Special case: If the program block defined as an anchor is part of a grouped program block, the anchor functionalities do not refer to the entire program, but only to the contents of the grouped program block.

During program execution, the program block of the currently executed action is selected in the program tree. This graphically highlights the position where the program is being executed.

6.4.2 Program execution in Teaching mode

For more information on Teaching mode, please see section 10.2.

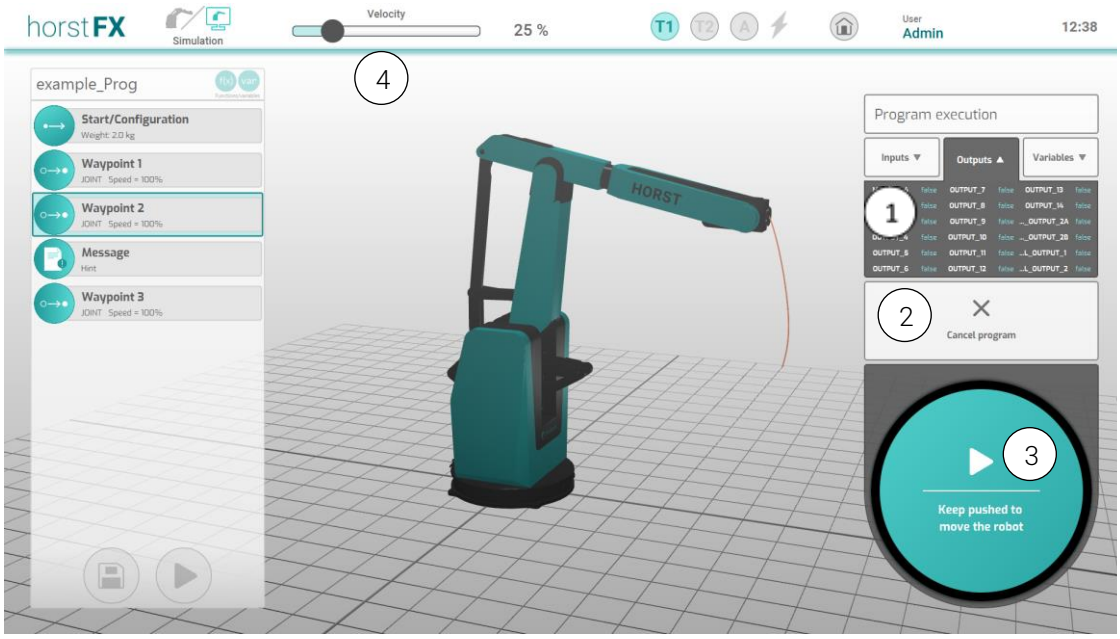


*Fig. 6-65: Program execution – **Teaching mode***

No.	Description
1	Display of the statuses of inputs, outputs, and variables
2	**Cancel program** button – stops program execution
3	Button for moving the robot In Teaching mode T1 or T2, this button must be kept pressed in order not to interrupt program execution. Releasing the button pauses the program execution.
4	**Speed controller** – sets the speed at which a program is executed

As soon as the operating mode is changed to **T2**, the speed controller (4) is set to 10% if more than 10% was previously set. For safety reasons, the speed controller can only be changed in **T2** with the enabling switch pressed. As soon as the enabling switch is released, not only does the movement end (if the robot is currently moving), but the speed controller is reduced to 10% again and a corresponding message appears on the display.

6.4.3 Program execution in Automatic mode

For more information on Automatic mode, please see section 10.3.

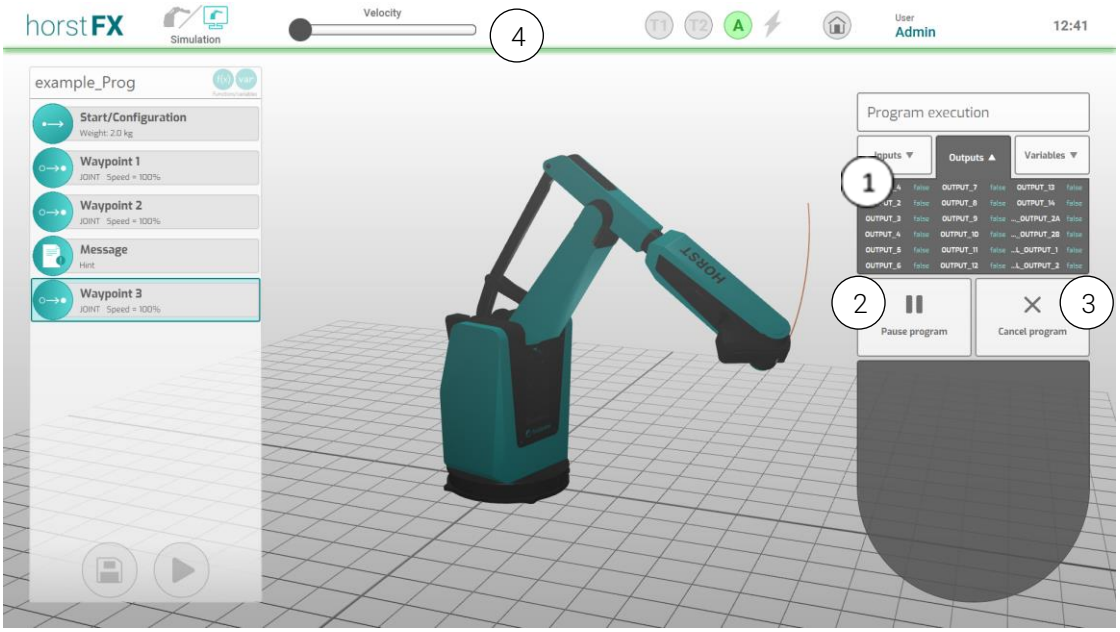


*Fig. 6-66: Program execution – **Automatic mode***

No.	Description
1	Display of the statuses of inputs, outputs, and variables
2	**Pause program** button – pauses the program execution
3	**Cancel program** button – stops program execution
4	**Speed controller** – sets the speed at which a program is executed

As soon as the operating mode is changed to **Automatic**, the speed controller (4) is set once to 10% for safety reasons if more than 10% was previously set. This is to prevent a program execution from unexpectedly executing movements at high speed.

6.5 Textual programming

In textual programming, the program tree and the actions/program blocks (see section 6.3) are replaced by a text editor. When adding actions, they are added to the text editor in textual form, exactly where the cursor is positioned.

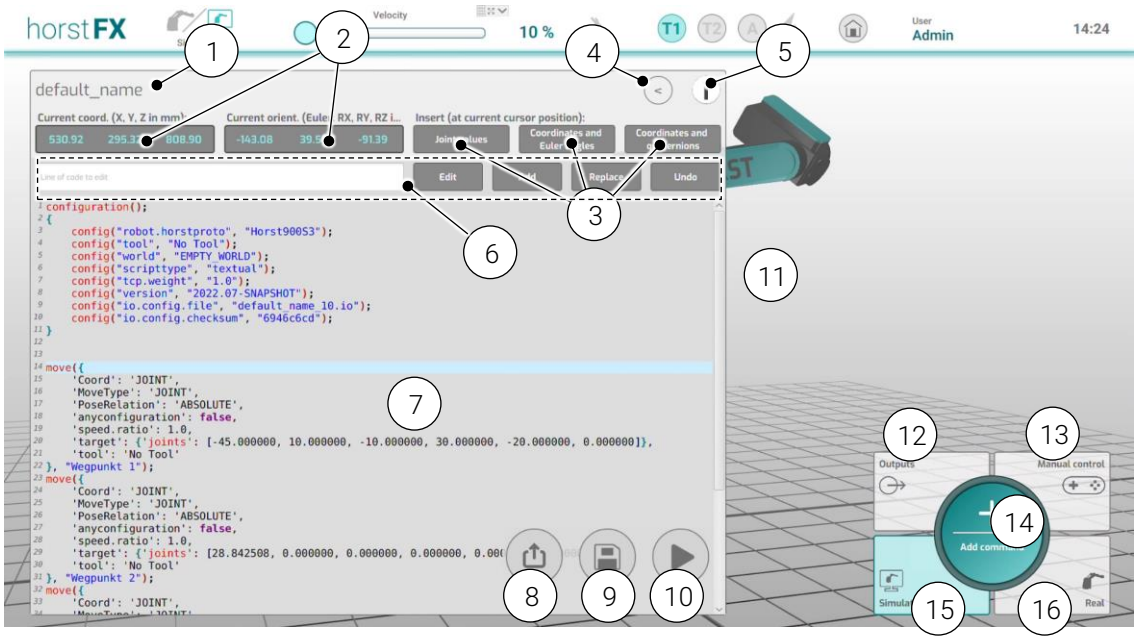


Fig. 6-67: Textual programming view

No.	Description
1	**Program name** of the current program
2	**Current coordinates** and **Current orientation** display – displays the current coordinates (X, Y, and Z in mm) and the current orientation (Euler angle in °) of the TCP By clicking on the display field, the corresponding values are saved in the temporary memory. Right-click to insert the values in array format at the current cursor position. Example of coordinates: [670.75, 0.00, 827.43];
3	**Joint values**, **Coordinates + Euler angle** and **Coordinates + Quaternions** buttons – for adding the joint value or the coordinate and orientation values at the current cursor position The corresponding values are inserted directly in the correct format for the move() command. The previous values in the move() command must be deleted accordingly.
4	**<** button (collapse/expand) – collapses/expands the area of the text editor
5	**i** button (information) – list of available commands including programming examples
6	Editing possibilities of the program code (advantageous and recommended for small changes, if no (physical) keyboard is available) Input **text editing** - adjusting/changing program code Edit **button** - inserts the currently selected line in the *text editor* into the input *text editing* **Add** button - inserts the content of the *text editing* input under the currently selected line in the *text editor* **Replace** button - replaces the currently selected line in the *text editor* with the content of the *text editing* entry **Undo** button - undoes the last change in the text editor
7	**Text editor** – shows the editable program code (commands)

No.	Description
8	Export button - via a file manager that opens, the program can be exported along with all its existing configurations as a ZIP file.
9	Save button – the program can be saved via a pop-up window that opens (An automatic save function also saves the program every 2 minutes in an autosave file)
10	▶ (Execute) button – the Program execution (textual) area will appear on the screen (see below: Program execution area (textual) section))
11	Robot model view – displays the current pose of the robot
12	Outputs button – opens the Manual control (outputs) menu (see section 6.3.27)
13	Manual control button – opens the Manual control menu (see section 6.3.27)
14	Add command button – opens the command selection area, through which an action is selected (see Fig. 6-5), which will be added to the *text editor* in textual form
15	Simulation button – selects operating mode Simulation In operating mode Simulation, only the movements of the robot model are displayed in the 3D world.
16	Real button – selects Real mode In Real mode, the robot performs the movements, and the movements of the robot model are displayed in the 3D world.

Program execution (textual) area

The Program execution (textual) area is almost identical to the Program execution area (see section 6.4). The only difference is that the functionality From anchor / To anchor is replaced by the functionality Run selected code (cf. Fig. 6-64 and Fig. 6-51).

By tapping the Run selected code button (1), only the previously selected code is executed in the text editor instead of the complete program. Any number of commands can be selected here.

Care must be taken to ensure that the selected code in the text editor contains only complete and valid commands. If this is not the case, the program execution aborts immediately with a corresponding error message.

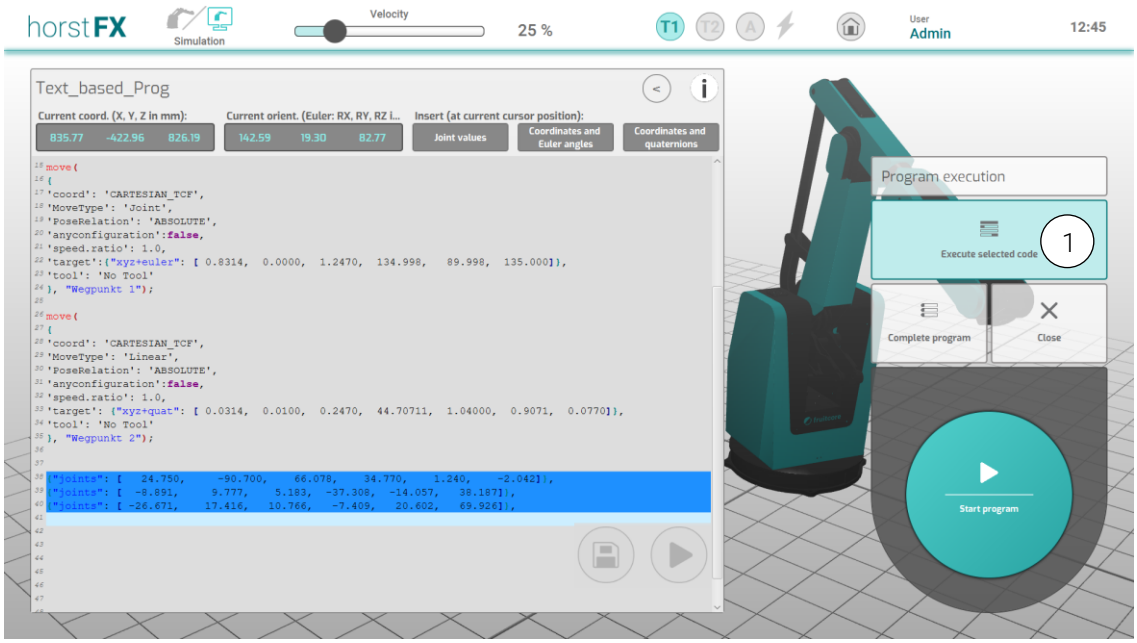


*Fig. 6-68: Programming view – **Program execution (textual)** area*

6.6 Functions

Functions can be created in any program. A function consists of various actions. Functions are used to group recurring actions and combine them in a program block. The program tree thus remains clearer, and the creation of a program becomes more convenient, if the same actions are often executed.

By tapping the **Functions/Variables** buttons (1) in the programming view, the program tree is replaced by a list of all existing functions (see Fig. 6-70).

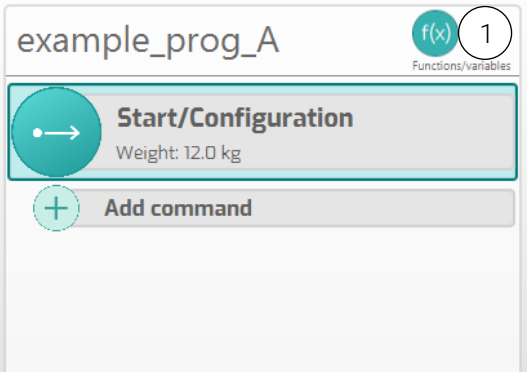


Fig. 6-69: Program tree (detail)

A function can be added at any point in the program tree using the **Function call action** (see section 6.3.4).

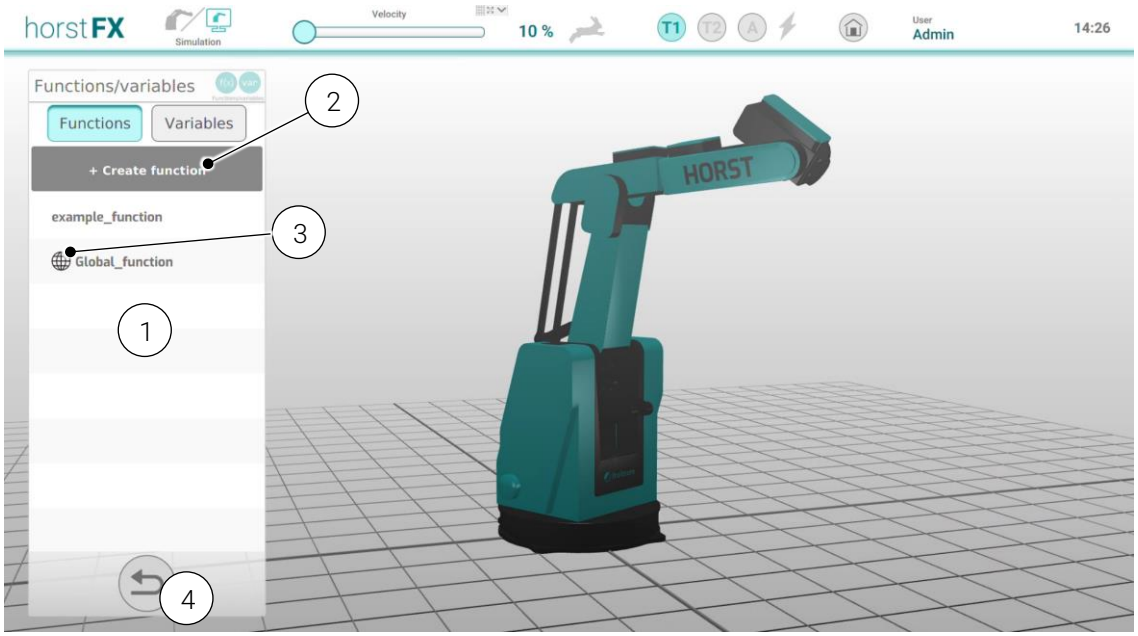


Fig. 6-70: Functions list

No.	Description
1	List of all existing functions
2	**Create function** button – for creating a new function
3	Label for global functions
4	**Back** button – hides the functions list and shows the program tree

After tapping the **New function** button, a pop-up window appears. Here the function must be named, and the selection made whether a graphical (see section 6.6.2) or a textual (see section 0) function is to be created.

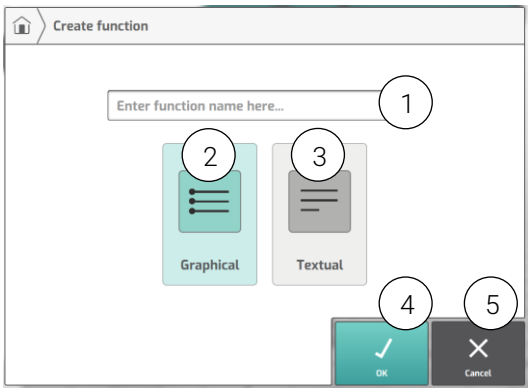


Fig. 6-71: Create new function

No.	Description
1	**Function name** input field – for entering the function name
2	**Graphical** button – for selecting a *graphical function* (see section 6.6.1)
3	**Textual** button – for selecting a *textual function* (see section 6.6.2)
4	**OK** button – opens a menu for editing the function
5	**Cancel** button – closes the pop-up window

6.6.1 Graphical functions

The **Graphical functions** menu appears when a new graphical function is created, or an existing graphical function is selected in the functions list.

If a new function is created, the **Define function** action window appears first. A description, transfer parameters, and a return value can be added here.

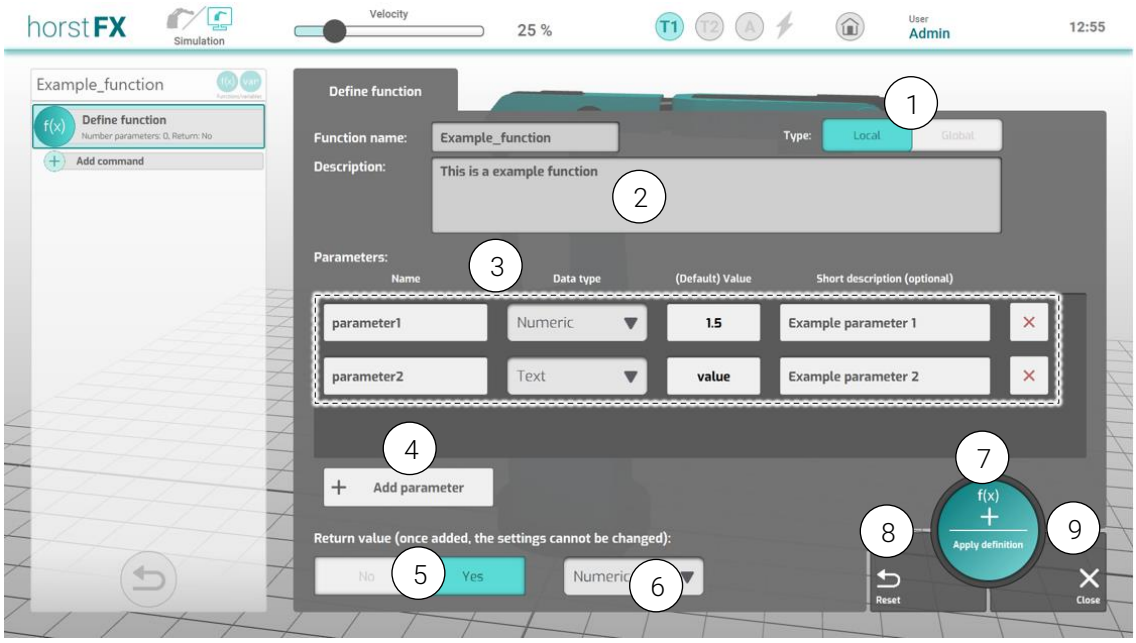


Fig. 6-72: Action window – Define function

No.	Description
1	**Type** toggle button – selection whether function of the type is *Local* or *Global*
2	**Description** input field – for entering a description of the function
3	Definition of **transfer parameters** – defines the name, data type, (default) value, and an optional description of the parameters
4	**Add parameter** button – adds a new parameter definition (2)
5	**Return value** toggle button – adds a return value to the function
6	**Return value** (data type) selection field – for selecting the data type of the return value
7	**Apply definition** button – the set configuration and selection options in the action window are applied and the action window is closed
8	**Reset** button – resets all configuration and selection options in the action window to the lastly saved state
9	**Close** button – closes the action window

Editing a graphical function is no different from editing the normal program (see section 6.3). In the **Graphical functions** menu, the program tree shows the program blocks of the function. The first program block *Define function* is the first program block in every function. This is fixed in the program tree and can be neither deleted nor moved. If this program block is being edited (see section 6.3.28), the corresponding action window opens (see Fig. 6-72).

If a return value is being added here, a corresponding **Return value** action (see section 6.3.15) with a default value is automatically appended at the end of the function. Any number of **return value** actions can be added within the function.

Functions do not have to be saved separately. If the **Graphical functions** menu is exited, a global function is automatically saved and a local function is stored in the program. Both function types can be used subsequently via the **Function call** action (see section 6.3.4).

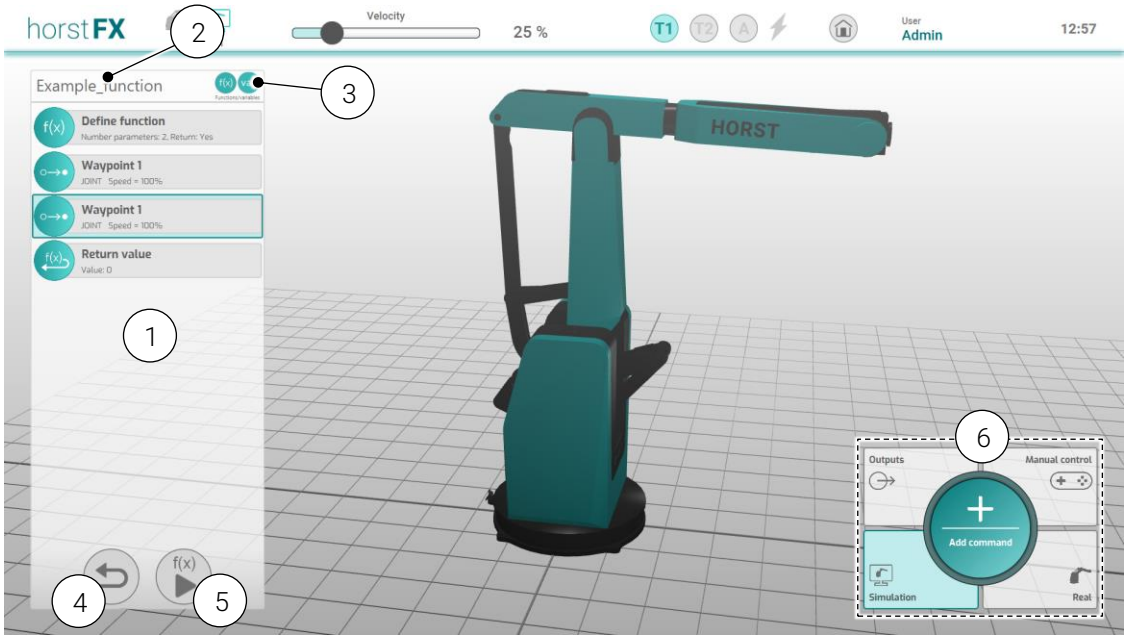


*Fig. 6-73: **Graphical function** menu*

No.	Description
1	**Program tree** display – displays the function with all its actions/program blocks
2	**Function name** of the current function
3	**Functions/Variables** button – lists existing functions/variables and the possibility to create new functions/variables
4	**Back** (to the main program) button – hides the **Graphical function** menu and shows the program tree
5	**Execute function** button – executes the current function
6	**Action area** – operation and functionalities are exactly the same as those described in section 6.3

A graphical function can be executed via the **Play function** button (see Fig. 6-73), which means that all program blocks of the function are executed, independent of the main program. If more than one tool is saved in the main program, a tool must be selected before the function execution, which will be used at the beginning of the function execution. If parameters are also defined in the function, the corresponding parameter values must be set (cf. Fig. 6-18). If nothing is changed here, the defined default values are used.

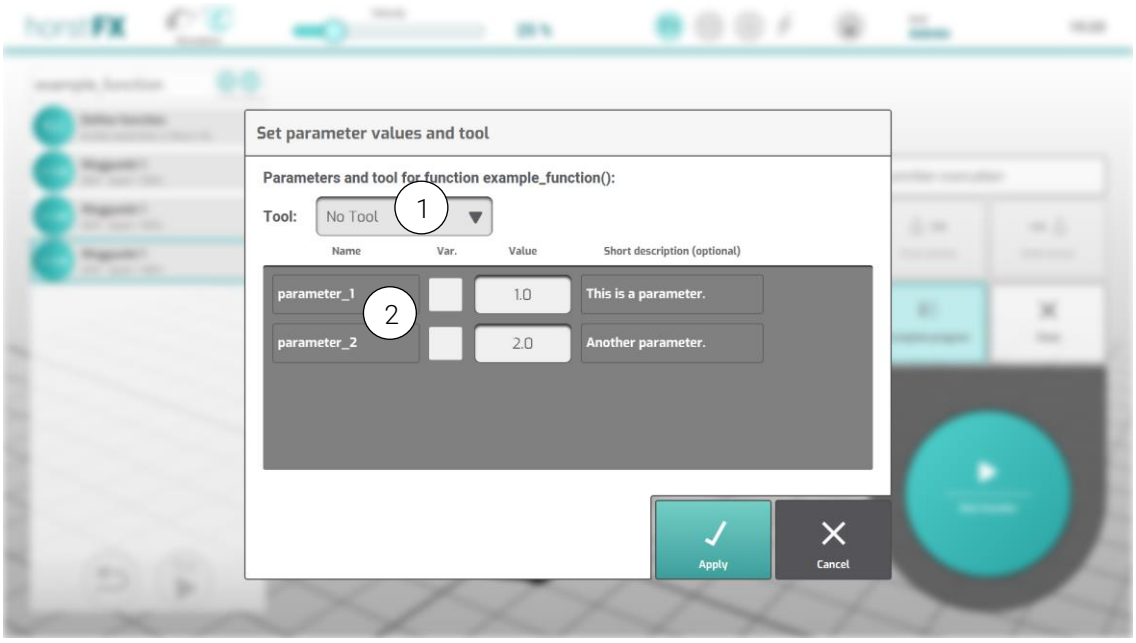


Fig. 6-74: Play function – set parameter values and tool

No.	Description
1	**Tool** selection field – for selecting/changing the tool for the current function
2	**Parameters** configuration – for setting the parameter values for the function execution

6.6.2 Textual functions

The **Textual functions** menu appears when a new textual function is created, or an existing textual function is selected in the functions list.

Editing a textual function is no different from editing the normal textual program (see section 6.5). In the **Textual functions** menu, the text editor displays all the code (commands) of the function.

Functions do not have to be saved separately. If the **Textual functions** menu is exited a global function is automatically saved and a local function is stored in the program. Both function types can be used subsequently via the **Function call** action (see section 6.3.4).

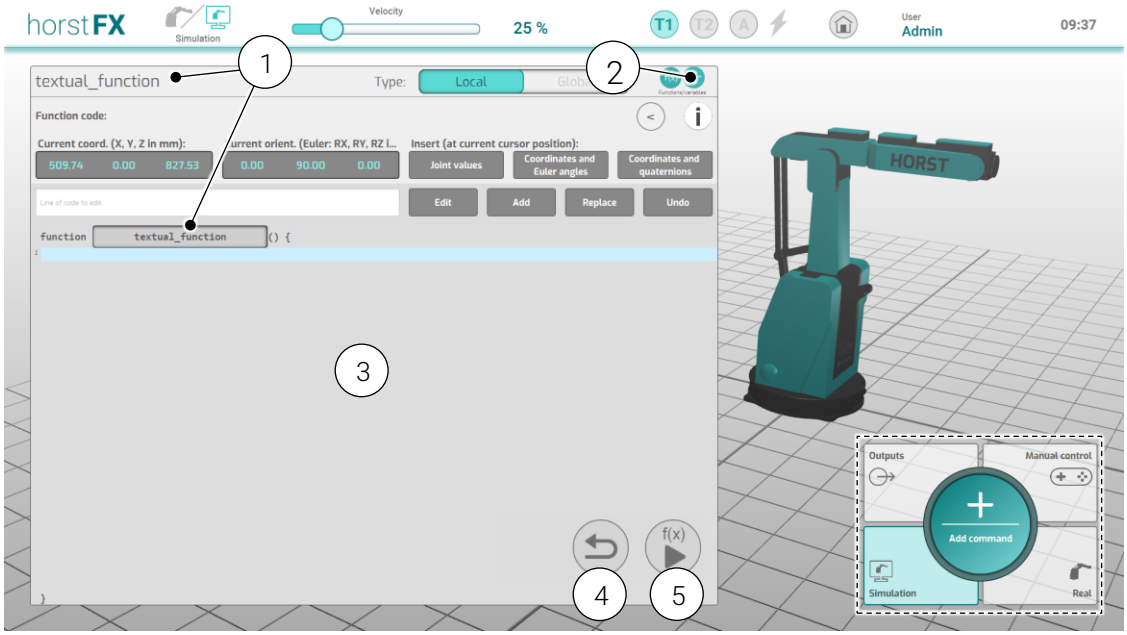


*Fig. 6-75: **Textual functions** menu*

No.	Description
1	**Function name** of the current function
2	**functions/variables** button – lists existing functions/variables and the possibility to create new functions/variables
3	**Text editor** – shows the editable program code (commands)
4	**Back** (to the main program) button – hides the **Textual functions** menu and shows the program tree
5	**Execute funktion** button – execution of the current function
	All other elements in this menu, which are not marked, are the same in operation and functionality as described in section 6.5.

6.6.3 Macro buttons

Macro buttons are quick access to execute selected functions directly. The macro buttons can be found in the normal programming view and in the **User-specific operating view** menu (see section 6.9). There are six macro buttons, each of which can be assigned a function. Both local and global functions can be used for this purpose. The assignment always applies only to the current program.

If a macro button is selected, the same menu opens as for **Execute function** command (see sections 6.6.1 and 6.6.2). However, the actions to be executed are not displayed in the **User-specific operating view** menu.

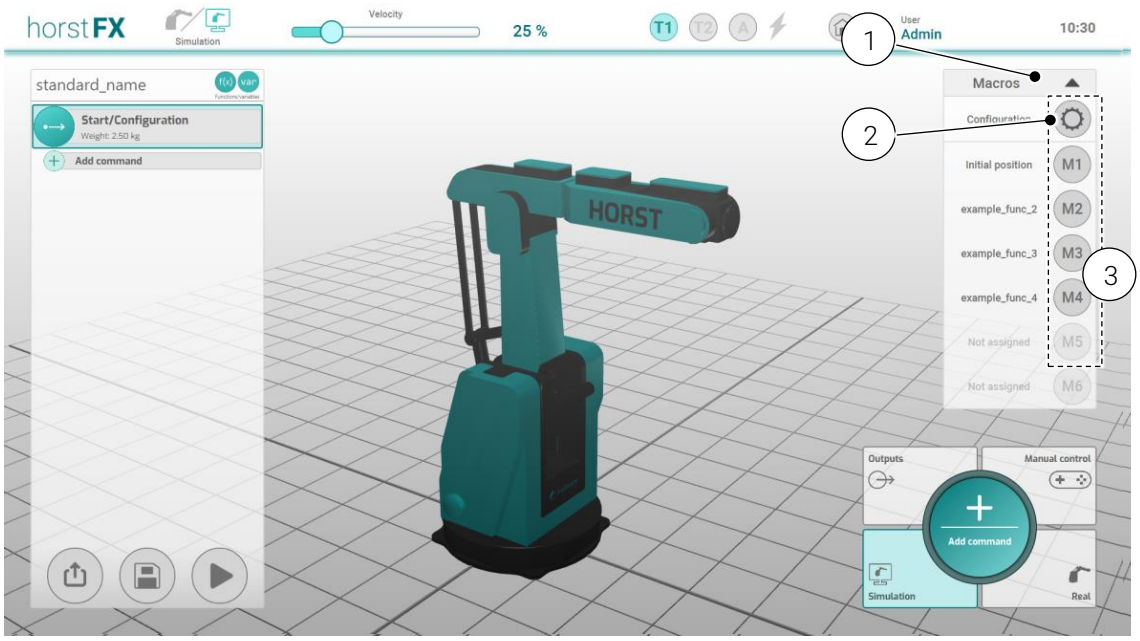


Fig. 6-76: Macro buttons

No.	Description
1	**Macros** button – collapse/expand the **macro buttons** menu
2	**Configuration** button – macro buttons assignment (see next figure)
3	**M1-M6** buttons – Macro buttons to execute the assigned functions

Configuration/Assignment of the macro buttons:

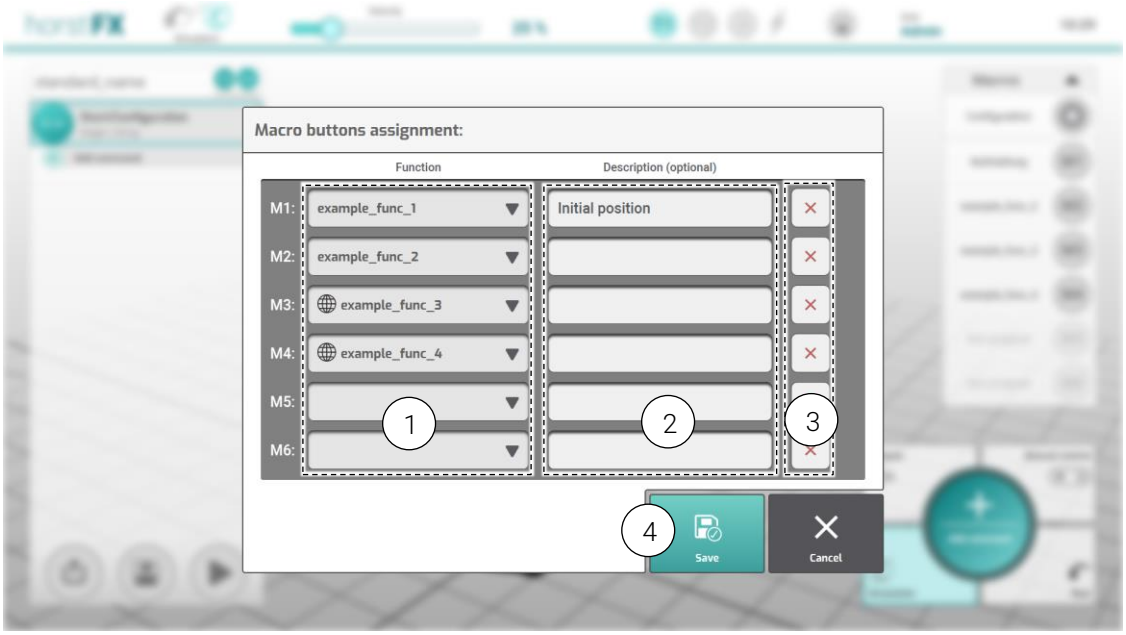


Fig. 6-77: Assignment of the macro buttons

No.	Description
1	**Function** selection field – selection of a function for assignment of the macro button

No.	Description
2	**Description** input – optional input of a description that will be displayed instead of the function name
3	**Delete** button – Deleting the selected function and description
4	**Save** button – saving the configuration/assignment

6.7 Variables

Variables can be created in any program. By tapping the **Functions/Variables** button (1) in the programming view and then tapping the **Variables** button, the program tree is replaced by a list of all existing variables (see Fig. 6-7962).

Variables are used to temporarily save certain values that are needed at a later time in the program.

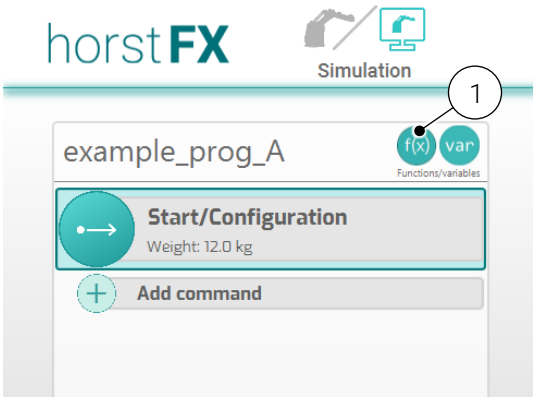


Fig. 6-78: Programming view (detail)

A variable value can be added at any point in the program tree using the **Change variable value action** (see section 6.3.7).

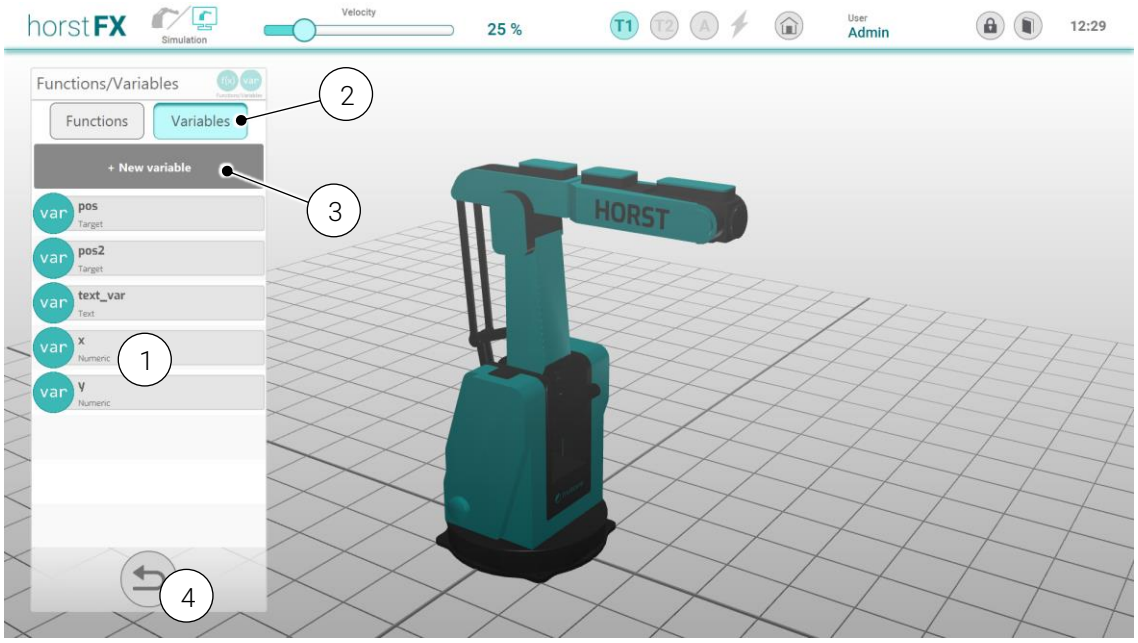


Fig. 6-79: Variables list

No.	Description
1	List of all existing variables
2	**Variables** button – shows the variables list (if not already visible)
3	**New variable** button – for creating a new variable

No.	Description
4	**Back** button – hides the variables list and shows the program tree

Tapping the **New variable** button opens the **Declare variable** action window. Here a new variable is created by assigning a valid variable name, selecting a data type and optionally assigning a value directly.

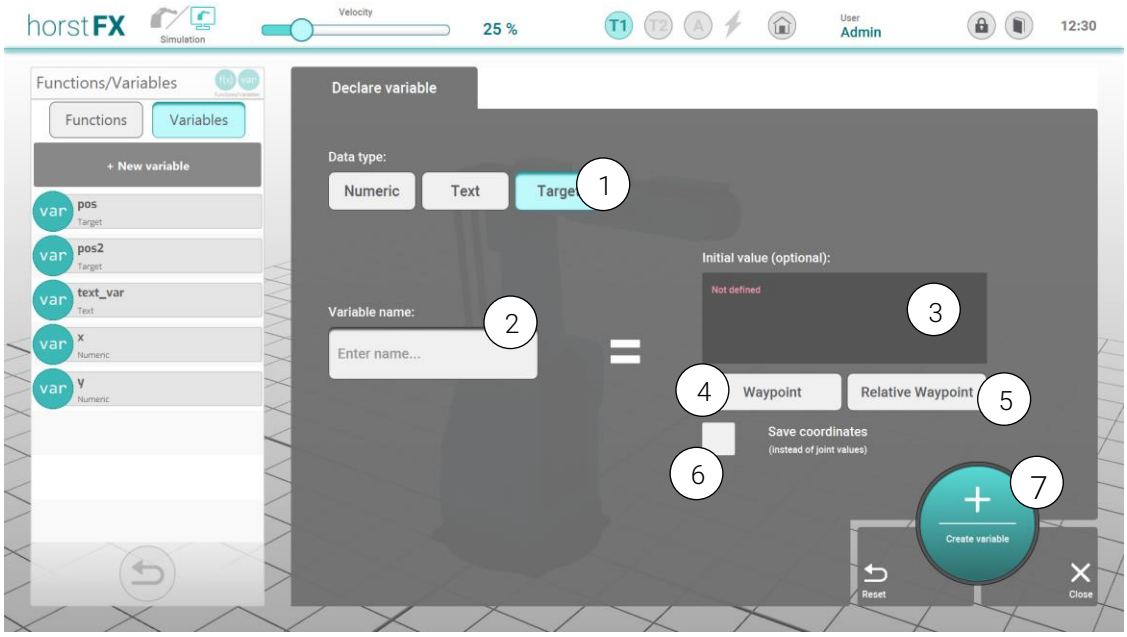


Fig. 6-80: Action window – Declare variable (new)

No.	Description
1	**Data type** buttons – selection of the data type *Numeric*, *Text* or *target position*
2	**Variable name** input field – for entering a variable name
3	Input field **initial value** – define initial value
3	**Create variable** button – the set configuration and selection options in the action window are applied, the action window is closed, and the created variable is added to the variables list
4	Button **Waypoint** – Menu **Target point** gets opened (6.3.2.2)
5	Button **Relative Waypoint** – Menu **Relative Waypoint** gets opened (6.3.3.1)
6	Checkbox **save coordinates** – saves coordinates instead of joint valuse
7	Button **create variable** – the set configuration and selection options in the action window are accepted, the action window is closed and the created variable is added to the variable listing

If an existing variable is tapped in the variables list, the **Declare variable** action window opens for editing the variable (see Fig. 6-81). Here the variable name can be changed, as well as the initial value and the variable can be deleted.

A variable can only be deleted if it is no longer used anywhere in the program, i.e. not even within other actions.

Fig. 6-81: Action window – Declare variable (edit)

6.8 Multiple tools

If more than one tool or TCP is required for a program (e.g. double gripper), it is possible to add and save additional tools in the program.

Since the **Relative waypoint action** behaves almost identically to the **Waypoint action** in terms of tools, the **Relative waypoint action** is only mentioned or described separately in this chapter if its behavior differs. Otherwise, the same functionalities apply to both actions.

6.8.1 Add more tools

Additional tools can be added to a program using the **Start/configuration action** (see section 6.3.1).

Fig. 6-82: Action window – Start/configuration with multiple tools

If more than one tool is saved, the first tool is always the default tool for the program. The first tool can therefore not be deleted since a default tool must always be saved. All other tools can be deleted (1) if they are not being used in any other action in the program.

When configuring the **Waypoint, Change pallet**, and **Tool actions**, only the tools added to the program here can be used in each case.

6.8.2 Save tool in waypoint

In the **Waypoint action** (see section 6.3.1.2) you can define which tool is to be used to approach the waypoint during program execution. For a new **Waypoint action**, the currently set tool is always applied in the **Tool** (1) display.

Fig. 6-83: Action window – waypoint

Use the **Remove** button (2) to remove the tool saved in the waypoint. In the **Tool** display (1), the text "*Mutable Tool*" is displayed after removal.

In order to save another tool in the waypoint, which is already saved in the program, it is necessary to switch to the **Define destination** menu (see section 6.8.3).

Waypoints for which "*Mutable Tool*" is saved as the tool are always approached with the tool set at this time when the program is executed. To set another tool within a program, the **Set tool action** is used (see section 6.8.6).

6.8.3 Define destination

If a waypoint is to be approached with a specific tool that does not correspond to the currently saved tool, a different tool can be selected in the **Define destination** menu (see section 6.3.2.2). For the **Relative waypoint action**, the **Define relative destination** menu applies accordingly (see sections 6.3.3.1 and 6.3.3.2). However, you can only choose from the tools that are saved in the program (see section 6.8.1).

As soon as the destination definition is saved, the selected tool is saved in the waypoint.

6.8.4 Save tool in pallet

In the Pallet action (see section 6.3.10), you can define which tool is to be used to approach the approach/ departure points and the palletizing points during program execution. When a new pallet action is created, *"Mutable Tool"* is always set in the Tool selection field (see 6.3.12.1).

If a specific tool is selected here, all approach/departure and palletizing points of the pallet are approached with the selected tool during program execution. If "*Mutable Tool"* is selected as the tool, then – as with waypoints – for each approach/departure point as well as palletizing point, the respective point is always approached with the tool set at this time during program execution. Consequently, another tool can even be used during a pallet pass using the Set tool action (see section 6.8.6).

6.8.5 Further actions with selection of the tool

Further actions to choose a specific tool or "*Variable tool*":

- Check area action (see section 6.3.14)

6.8.6 Set tool

All tools added to the program can be selected in the Set tool action (see section 6.3.16).

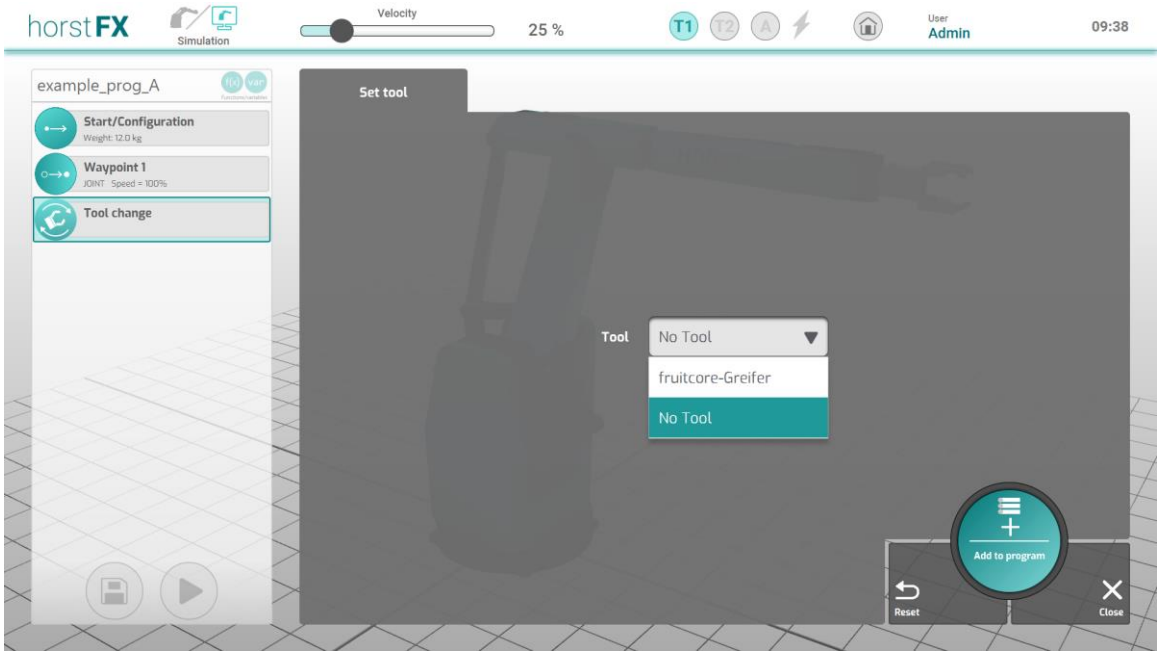


Fig. 6-84: Action window – Set tool with multiple tools

Thus, during a program execution, the tool can be changed, and another tool can be set. This action affects all other actions that have "*Mutable Tool"* set as the tool. These are executed accordingly with the last tool set or the tool set at the time of execution.

6.9 Coordinate systems (user-defined)

The creation of coordinate systems is described in section 4.5.4.
The created coordinate systems can be accessed in the actions Waypoint, Relative Waypoint, Palette, Check Area, and Change Coordinate System as well as in the control menus. However, they must first have been added to the programme via the action Start/Configuration.

7 User-specific operating view

The User-specific operating view menu is a very simplified and slimmed-down version of the normal programming view. Here, the contents of the loaded program can neither be viewed nor edited, but only the program can be executed. It is intended exclusively for use by the user role operator.

The configuration options available for the user-specific operating view menu are described in the user-specific view settings menu (see Section 4.3.5).

By pressing the load program button in the main menu, this menu appears, but only under the following conditions:

- A program must be stored in the settings menu user-specific view.
- The automatic operating mode must be selected.
- The user role operator must be logged in.

If one of these conditions is not met, the normal programming view (see Sections 6.2and 6.3) or a corresponding pop-up window is displayed instead.

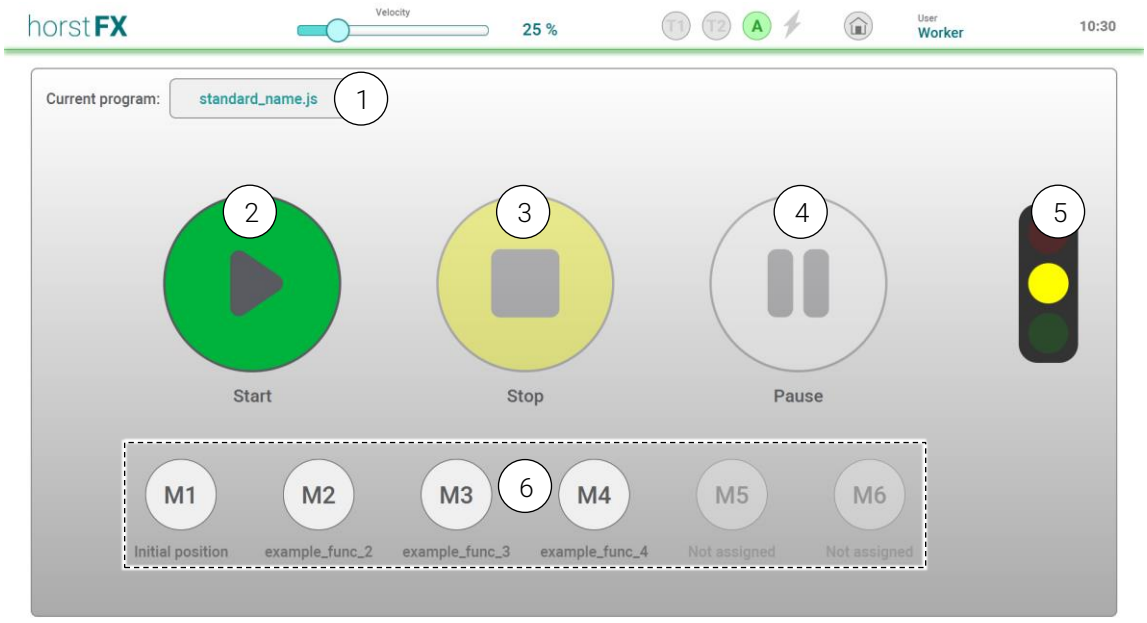


Fig. 7-1: User-specific operating view menu

Pos.	Description
1	Current program display – shows the name of the loaded program
2	Start/Continue button – starts the program execution or continues it, if it was paused before
3	Stop button – stops the executed program
4	Pause button – pauses the executed program
5	Display of the current program status - Green: program is being executed - Yellow: program paused - Red: error present
6	M1-M6 buttons - macro buttons for executing the assigned functions (see section 6.6.3)

8 Control robot externally

Pressing the **Control robot externally** button in the main menu displays the **Control robot externally** menu.

Via the primary interface of horstFX, the robot can be controlled from an external computer by means of function calls via an XML-RPC protocol (Extensible Markup Language Remote Procedure Call).

This technique allows the remote execution of methods, with data transfer via HTTP (Hypertext Transfer Protocol). The data to be transferred is in XML format. There are many XML-RPC clients in various programming languages, so integration into existing projects is easy. Several sample clients have been developed to make it easier to get started, initially in Java and in HTML/JavaScript. These clients can connect to the XML-RPC server of the horstFX application and then execute commands there.

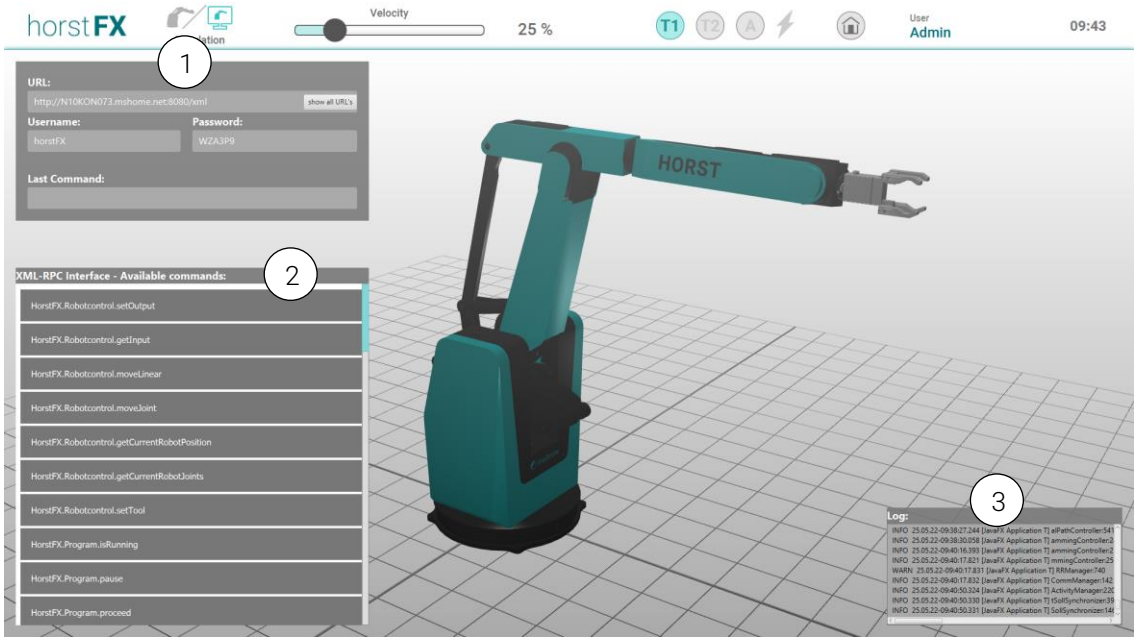


Fig. 8-1: ***Control robot externally*** *menu*

No.	Description
1	Display of data needed for the external clients to connect to the primary interface.
2	List of all available commands that can be sent to the primary interface.
3	Display of log outputs generated by incoming commands.

Further and more detailed information on the use of the external control system can be found at **horstcosmos.com**. If you have any further questions, please contact the service department of fruitcore robotics GmbH.

9 Warning and Error Messages

Warning and error messages, such as the emergency stop warning message, appear in the form of pop-up windows. If a warning or error message is ignored instead of acknowledged, a red warning symbol (lightning icon) flashes in the menu bar. By tapping this warning icon, an ignored message can be displayed again at a later time and then acknowledged/confirmed.

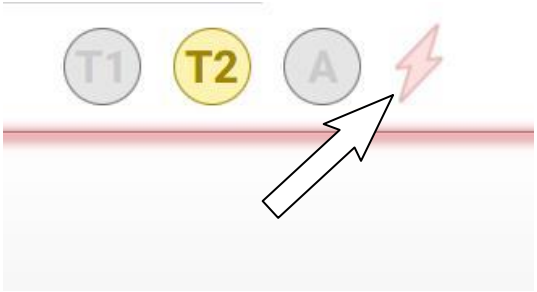


Fig. 9-1: Unacknowledged warning or error message

The following messages are possible:

- Emergency stop warning message
- Safety stop warning message
- System error message
- Operating mode change warning message

9.1 Emergency stop warning message

After triggering the emergency stop:

The **Acknowledge** button is initially disabled.

The **Acknowledge** button is activated as soon as the emergency stop button is unlocked.

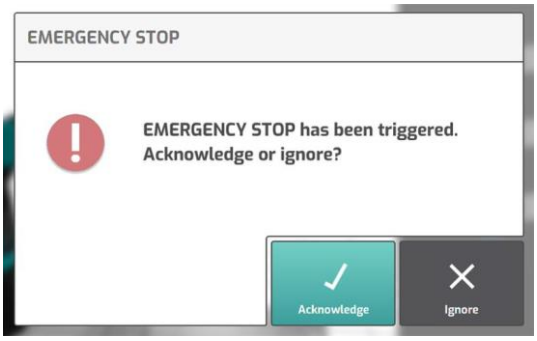


Fig. 9-2: Emergency stop warning message

Tapping the **Acknowledge** button will open a pop-up window to confirm the acknowledgement process.

Fig. 9-3: Emergency stop – confirmation

For behavior in case of emergency, see Section 10.1.

9.2 Safety stop warning message

After the safety stop has been triggered:

The **Acknowledge** button is initially disabled. The **Acknowledge** button is activated as soon as the cause of the safety stop is eliminated (e.g. the safety door of a safety cell is closed again or there are no more objects in the monitoring area of a safety scanner).

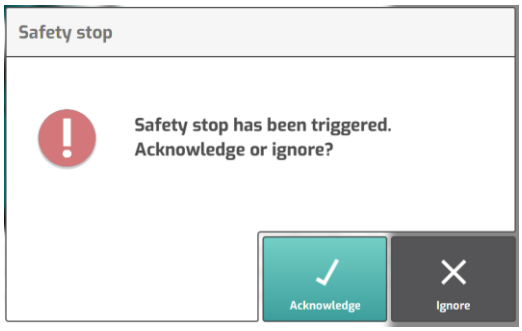


Fig. 9-4: Safety stop warning message

Tapping the **Acknowledge** button will open a pop-up window to confirm the acknowledgement process.

Fig. 9-5: Safety stop – Acknowledge

9.3 System error message

In case of malfunctions on the robot system, corresponding error messages (system errors) are displayed on the horstPANEL.

If one or more system errors occur, a corresponding pop-up window appears.

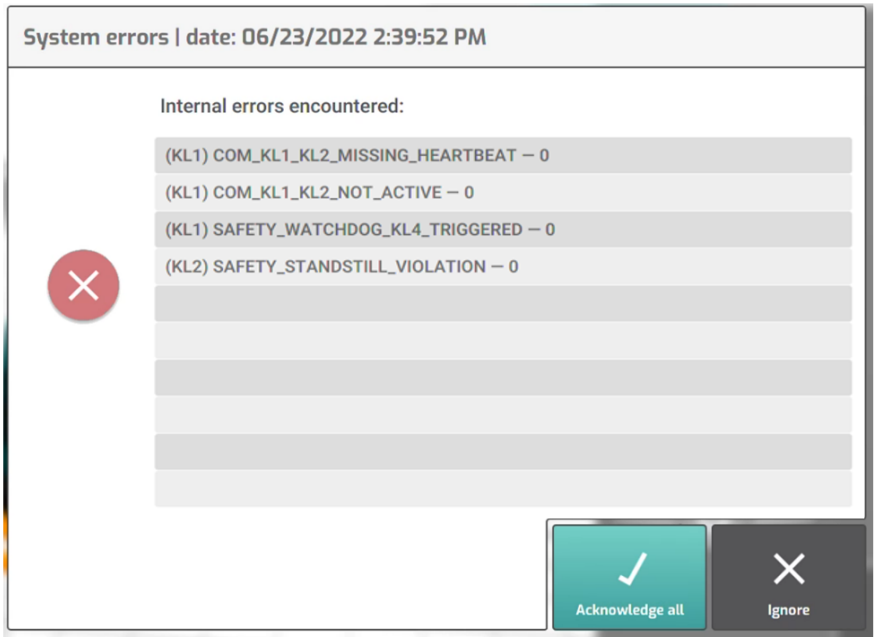


Fig. 9-6: Display system error

Tapping the **Acknowledge all** button will open a pop-up window to confirm the acknowledgement process.

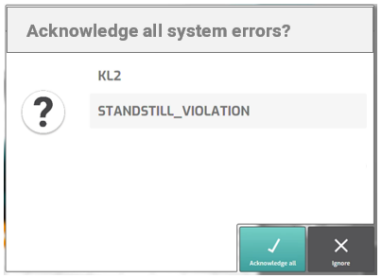


Fig. 9-7: System error – confirm acknowledgment

After all system errors have been acknowledged, normal operation can continue.

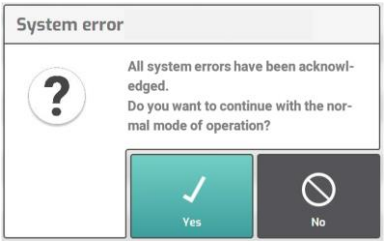


Fig. 9-8: Continue operation – query

Tapping the **Yes** button will open the pop-up window to confirm the continuation.

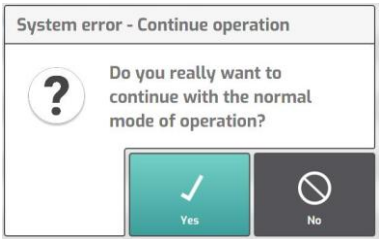


Fig. 9-9: Continue operation – confirm

9.3.1 Overload error (step loss)

If there is an overload error (step loss) among the system errors, a special pop-up window with additional information about overload (step loss) is displayed.

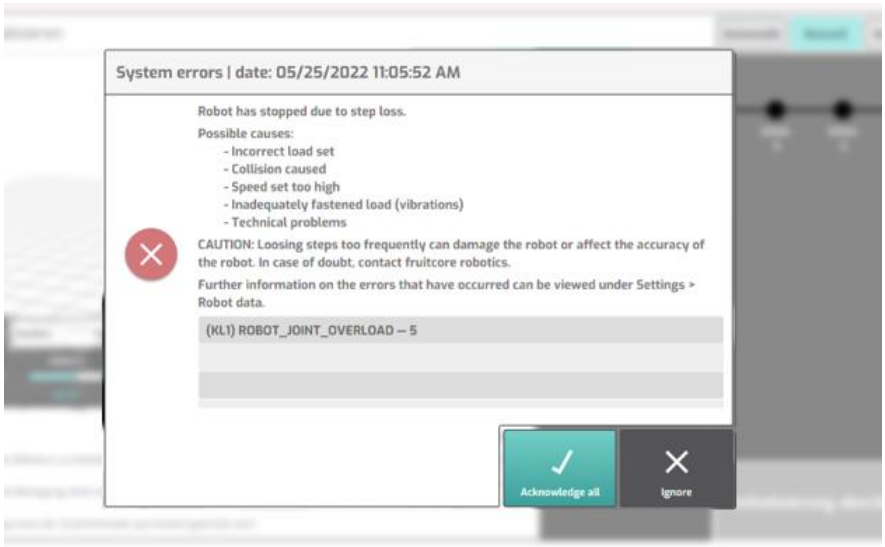


Fig. 9-10: Display system error – overload

The further procedure is identical to all other system errors (see section 9.3).

More detailed information such as the number and times of the last occurrences of the overload errors can be found in the **Settings – Robot data** menu (see section 4.4.1).

9.4 Operating mode change warning message

Switching to a different operating mode causes the robot to stop. A corresponding pop-up window appears on the display. To continue, the warning message must be acknowledged. The enabling switch must be released during this time.

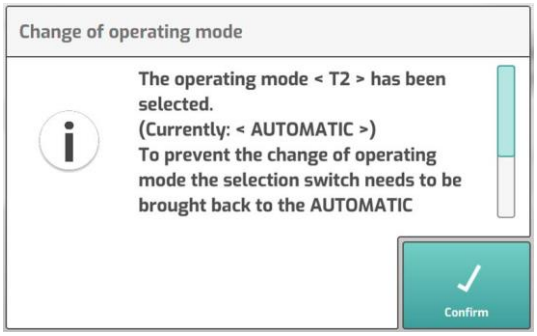


Fig. 9-11: Change operating mode – query

By tapping the Confirm button, a pop-up window will open to confirm the operating mode change.

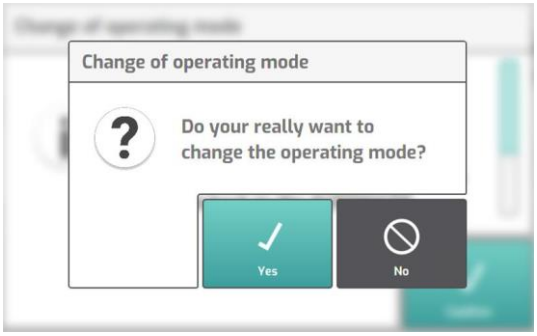


Fig. 9-12: Change operating mode – Confirm

After confirming the operating mode change, the symbol for the corresponding operating mode is highlighted in color in the menu bar.

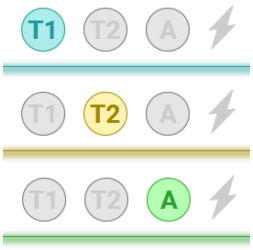


Fig. 9-13: Display of the current operating mode

10 Operation

10.1 Behavior in an emergency

WARNING!
Danger of impact and crushing due to robot movement

The safety stop function is deactivated in Teaching mode.

- While the robot is in Teaching mode, close off the area around the robot, and protect it against access by unauthorized persons. There must be no persons in the danger zone of the robot.
- In Teaching mode, secure horstPANEL and horstCONTROL against operation by unauthorized persons.

WARNING!
The robot arm may be moved by applying external force only in emergencies.

Modules of the robot system may be damaged if the robot arm was moved manually in an emergency. This may result in an uncontrolled start-up.

- Have the robot system checked by the service department of fruitcore robotics GmbH before putting it back into operation.

- In an emergency, press the emergency stop button (1).
  - ⇒ All movements of the robot are braked until it comes to a standstill. The program is paused.

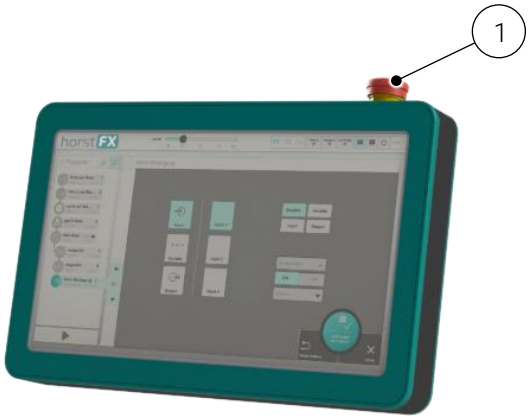


Fig. 10-1: horstPANEL

- A pop-up window with a warning message stating that an emergency stop was triggered appears on the display.

Fig. 10-2: Emergency stop warning message

- Rectify the danger situation.

Reset the emergency stop

- Check whether the danger has been rectified before resetting the emergency stop.

- Release the emergency stop button by pulling it out.
  - ⇒ The Confirm button is activated.
- Confirm the warning message on the display.
  - ⇒ If the emergency stop was reset, the program only starts up again once it is continued manually.

Fig. 10-3: Emergency stop – confirmation

10.2 Teaching mode

WARNING!
Danger of impact and crushing due to robot movement
The safety stop function is deactivated in Teaching mode.

- While the robot is in Teaching mode, close off the area around the robot, and protect it against access by unauthorized persons. There must be no persons in the danger zone of the robot.
- In Teaching mode, secure horstPANEL and horstCONTROL against operation by unauthorized persons.

The robot can only be moved manually in two-handed operation. To move the robot, the enabling switch must always be kept pressed in the center position in operating modes T1 and T2. In addition, the desired control element must be kept pressed on the display. As soon as one of the two conditions is no longer fulfilled, the robot will brake until it comes to a standstill.

- Switch on the robot system (see section 2.1).

If the power supply was interrupted previously, the robot must be reinitialized.

- If necessary, initialize the robot (see section 2.2).

Switching to a different operating mode causes the robot to stop. A warning message appears on the display. The message must be confirmed in order to proceed. The enabling switch must be released during this time.

There are two Teaching modes: T1 or T2.

T1 – programming mode

The speed of the TCP is limited to 250 mm/s. The robot can be moved only with the enabling switch.

T2 – program verification mode

The speed of the TCP (tool center point) can exceed 250 mm/s. The robot can be moved only with the enabling switch.

- ▶ Switch the key switch of horstCONTROL to T1 or T2. Pull out the key to prevent unauthorized persons from changing the operating mode.

⇒ Confirm the change of operating mode (see section 9.3.1).

⇒ Create, edit, or run a program (see section 6).

10.3 Automatic mode

In Automatic mode, the robot moves without an enabling switch and the safety stop input is active.

WARNING!
Danger of impact and crushing due to robot movement

- ▶ Ensure that suitable protection devices (e.g. separating protection device, light curtain, or safety laser scanner) have been installed.
- ▶ Check the protection devices for proper function.

ATTENTION!
Danger of damage due to incorrect or missing configurations in the software.

- ▶ Ensure that the program to be executed has been programmed and tested correctly before staring Automatic mode.

ATTENTION!
Risk of collision due to program changes during Automatic mode.

- ▶ Do not make any changes to the program during Automatic mode.
- ▶ Make sure that no unauthorized persons have access to horstPANEL.
  - ▶ Switch on the robot system (see section 2.1).

If the power supply was interrupted previously, the robot must be reinitialized.

- ▶ If necessary, initialize the robot (see section 2.2).

Switching to a different operating mode causes the robot to stop. A warning message appears on the display. The message must be confirmed in order to proceed. The enabling switch must be released during this time.

- ▶ Switch the key switch of horstCONTROL to Automatic. Pull out the key to prevent unauthorized persons from changing the operating mode.

⇒ Confirm the change of operating mode (see section 9.3.1).

⇒ Create, edit, or run a program (see section 6).

10.4 Shutdown after end of operation

At the end of operation, the robot system must be shut down.

- To cancel a running program, tap the **Cancel program** button (1).
  ⇒ The robot is braked immediately.

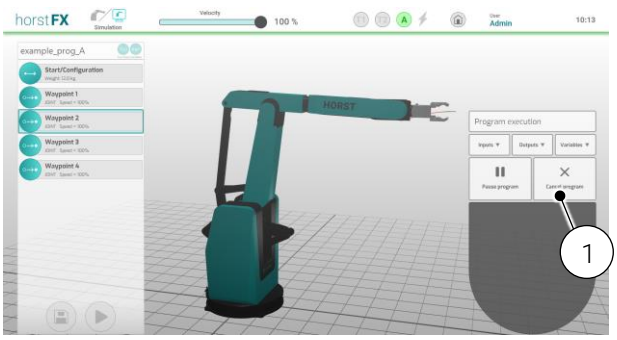


Fig. 10-4: Cancel program execution

- Check whether the robot is in a safe position (e.g. there is no workpiece in the gripper). If necessary, bring the robot into a safe position by moving it manually via the **Free travel** menu (see section 4.9.1).

ATTENTION!
Before the robot system is switched off, the proper shutdown of the computer for horstFX integrated in horstCONTROL must be ensured.

- Navigate to the main menu. Tap on the Exit horstFX button there (see Fig. 3-3). A pop-up window with two options appears. Select the option Shut down system (1) and then confirm with the OK button (2).

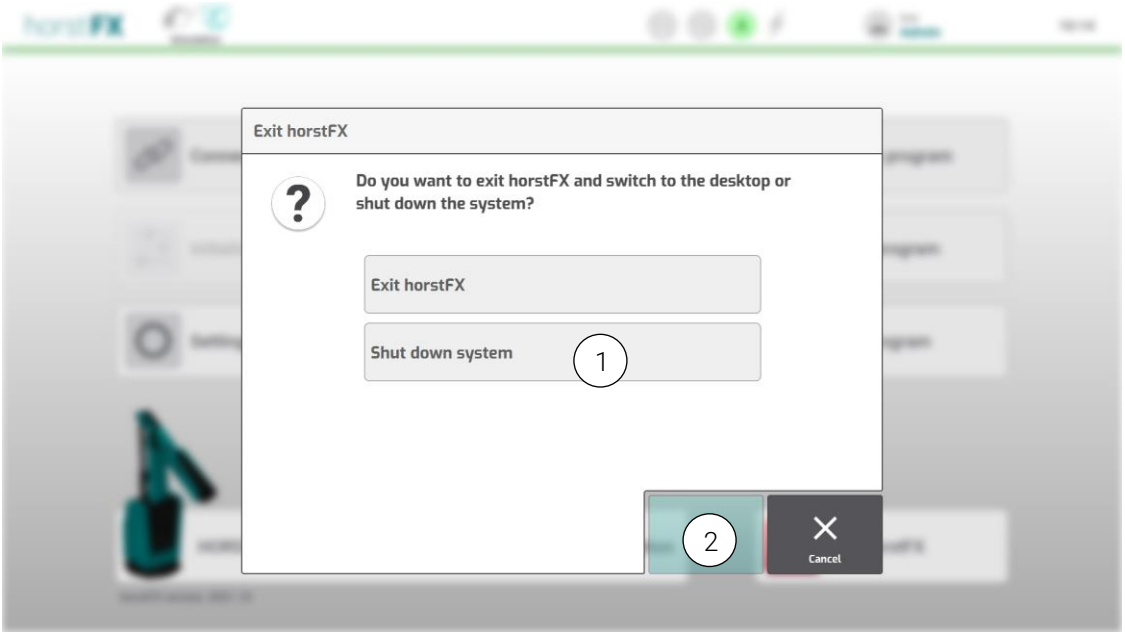


Fig. 10-5: Shutting down the computer for horstFX

Alternatively, you can shut down the computer integrated in horstCONTROL for horstFX as follows:

Press the PC ON/OFF button of horstCONTROL. A pop-up window appears. In this pop-up window, tap the **Shut Down** button (1) to shut down the computer for horstFX.

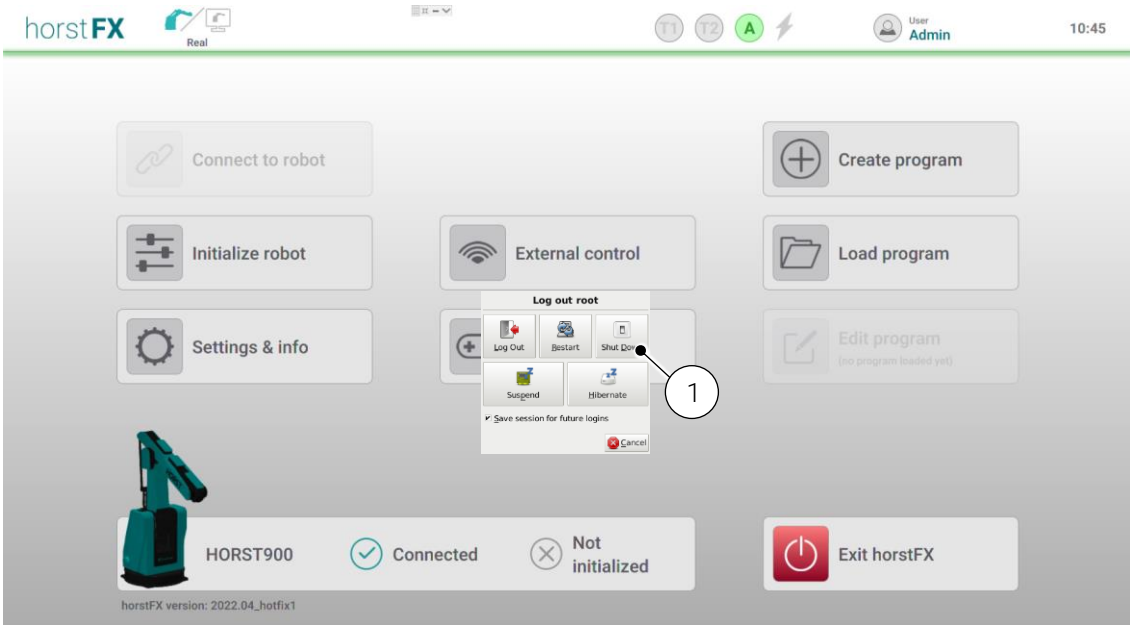


Fig. 10-6: Shutting down the computer for horstFX – alternative

- ▶ Switch the main switch of horstCONTROL to **OFF**.
- ▶ Secure the main switch with a lock.

11 Troubleshooting

If faults occur in the robot system, corresponding error messages (system errors) are displayed on the horstPANEL.

- Follow the instructions on the horstPANEL to rectify the cause of the fault.
- Confirm the fault message on the display once all causes of the fault have been removed.
- Call the fruitcore robotics GmbH service department if you cannot remove the causes of the fault yourself.

In case of software problems, switch off the robot system according to section 10.4 and switch it on again as described in section 2.