OPERATING INSTRUCTIONS

PLB 7.6

Robot Guidance Systems

SICK
Sensor Intelligence.

Described product

PLB

Manufacturer

SICK AG
Erwin-Sick-Str. 1
79183 Waldkirch
Germany

Legal information

This product includes software developed by 3ideo, Arras France.

Original document

This document is an original document of SICK AG.

Contents

1 About this document

1.1 Information on the operating instructions

These operating instructions provide important information on how to use products from SICK AG. Prerequisites for safe work are:

- Compliance with all safety notes and handling instructions supplied.
- Compliance with local work safety regulations and general safety regulations for product applications .

The operating instructions are intended to be used by qualified personnel.

NOTE
Read these operating instructions carefully before starting any work, in order to familiarize yourself with the product and its function.

The instructions constitute an integral part of the product and must be accessible to staff at all times.

These operating instructions do not provide information on operating the machine or system in which the product is integrated. For information about this, refer to the operating instructions of the specific machine.

However, you should note that the planning and use of the PLB system demands specialist technical knowledge that does not form part of these Operating Instructions.

These Operating Instructions describe the PLB system from SICK.

1.2 Explanation of symbols

Warnings and important information in this document are labeled with symbols. The warnings are introduced by signal words that indicate the extent of the danger. These warnings must be observed at all times and care must be taken to avoid accidents, personal injury, and material damage.

DANGER
... indicates a situation of imminent danger, which will lead to a fatality or serious injuries if not prevented.

WARNING
... indicates a potentially dangerous situation, which may lead to a fatality or serious injuries if not prevented.

CAUTION
... indicates a potentially dangerous situation, which may lead to minor/slight injuries if not prevented.

NOTICE
... indicates a potentially harmful situation, which may lead to material damage if not prevented.

NOTE
... highlights useful tips and recommendations as well as information for efficient and trouble-free operation.

1.3 Further information

NOTE
Further documentation for the system can be found on the online product page at:

- www.sick.com/PLB

There, additional information has been provided depending on the product, such as:

- Model-specific online data sheets for device types
- Declarations of conformity and certificates of the product family
- Accessories applicable for the system

2 Safety information

2.1 Intended use

PLB is a system for precise localization of parts stored in a defined search volume. The system may only be employed in accordance with its intended use.

The PLB system is intended to be used in industrial environments. The operator of the production facilities in which the PLB system is to be integrated must implement measures to ensure the safety of persons and equipment in accordance with statutory guidelines and regulations.

The product must only be used within the limits of the prescribed and specified technical specifications and operating conditions at all times.

Incorrect use, improper modification or manipulation of the product will invalidate any warranty from SICK; in addition, any responsibility and liability of SICK for damage and secondary damage caused by this is excluded.

In the case of any non-specified usage or in the event of any modifications to system components (e.g. by opening the camera housing or during the course of assembly and electrical installation) or to the SICK software, any warranty claims against SICK AG shall be null and void.

2.2 Incorrect use

Any use outside of the stated areas, in particular use outside of the technical specifications and the requirements for intended use, will be deemed to be incorrect use.

The robot guidance system is NOT a safety component according to the respective applicable safety standards for machinery. The position data of the system must NOT be used for safety-critical control of the robot or the system. The system must NOT be used for functional safety.

The system must NOT be used in explosion-hazardous areas, in corrosive environments, or under extreme environmental conditions.

Any use of accessories not specifically approved by SICK AG is at your own risk.

If the system is to be used under other conditions or in different environments, then the manufacturing service may issue an operating license in consultation with the customer and in exceptional cases.

2.3 Cybersecurity

To protect against cybersecurity threats, it is necessary to continuously monitor and maintain a comprehensive cybersecurity concept. A suitable concept consists of organizational, technical, procedural, electronic, and physical levels of defense and considers suitable measures for different types of risks. The measures implemented in this product can only support protection against cybersecurity threats if the product is used as part of such a concept.

You will find further information at www.sick.com/psirt, e.g.:

- General information on cybersecurity
- Contact option for reporting vulnerabilities
- Information on known vulnerabilities (security advisories)

NOTE
SICK uses standard IP technology in its products. The emphasis is placed on availability of products and services. SICK always assumes that the integrity and confidentiality of the data and rights affected by the use of the aforementioned products will be ensured by the customer. In all cases, appropriate security measures, such as network separation, firewalls, virus protection, and patch management, must be taken by the customer on the basis of the situation in question.

2.4 Limitation of liability

Applicable standards and regulations, the latest state of technological development, and our many years of knowledge and experience have all been taken into account when assembling the data and information contained in these operating instructions. The manufacturer accepts no liability for damage caused by:

- Failing to observe the operating instructions
- Incorrect use
- Use by untrained personnel
- Unauthorized conversions
- Technical modifications
- Use of unauthorized spare parts, consumables, and accessories

With special variants, where optional extras have been ordered, or owing to the latest technical changes, the actual scope of delivery may vary from the features and illustrations shown here.

2.5 Modifications and conversions

NOTICE
Modifications and conversions to the device may result in unforeseeable dangers.

Interrupting or modifying the device or SICK software will invalidate any warranty claims against SICK AG. This applies in particular to opening the housing, even as part of mounting and electrical installation.

2.6 Qualification requirements for personnel

WARNING
Risk of injury due to insufficient training.
Improper handling may result in considerable personal injury and material damage.

- All work must only ever be carried out by the stipulated persons.

These operating instructions list the training requirements for the various fields of activity, as follows:

- **Instructed personnel** have been given a briefing by the operator about the tasks assigned to them and about potential dangers arising from improper action.
- **Skilled personnel** have the specialist training, skills, and experience, as well as knowledge of the relevant regulations, to be able to perform tasks delegated to them and to detect any potential dangers independently.
- **Electricians** have the specialist training, skills, and experience, as well as knowledge of the relevant standards and provisions to be able to carry out work on electrical systems and to detect any potential dangers independently. Relevant applicable national regulations must be observed.

2.7 Operational safety and particular hazards

Please observe the safety notes and the warnings listed here and in other sections of this production documentation to reduce the possibility of risks to health and avoid dangerous situations.

Some operational safety and particular hazards depend on camera variant, and some are applicable to all variants.

If the product is operated in conjunction with external illumination systems, the risks described here may be exceeded. This must be taken into consideration by users on a case-by-case basis. Please note the accompanying production documentation.

Visionary-S and Visionary-T Mini (PLB510 + PLB540)

CAUTION
Optical radiation: Class 1 Laser Product

The accessible radiation does not pose a danger when viewed directly for up to 100 seconds. It may pose a danger to the eyes and skin in the event of incorrect use.

- Do not open the housing. Opening the housing may increase the level of risk.
- Current national regulations regarding laser protection must be observed.

Zivid 2+ (PLB536)

CAUTION
Warning! Optical radiation: LED risk group 2, visible radiation, 400 nm to 780 nm

Potentially dangerous optical radiation. Can be damaging to the eyes.

- Do not look into the light source for extended periods of time.
- Never point the light source at people.
- Avoid any reflections on people from reflective surfaces. Be particularly careful during mounting and alignment work.
- Do not open the housing. Opening the housing will not switch off the light source. Opening the housing may increase the level of risk.
- Comply with the current national regulations on photobiological security of lamps and lamp systems.

Ensenso X36 and C57 (PLB520 5 MP + PLB521+ PLB522)

CAUTION
Optical radiation: LED risk group 1, visible radiation, 400 nm to 780 nm

The LEDs may pose a danger to the eyes in the event of incorrect use.

- Do not look into the light source intentionally.
- Do not open the housing. Opening the housing will not switch off the light source. Opening the housing may increase the level of risk.
- Comply with the current national regulations on photobiological security of lamps and lamp systems.

Ensenso N36 (PLB515)

The product is fitted with LEDs in risk group 0. The accessible radiation from these LEDs does not pose a danger to the eyes or skin.

For both laser and LED radiation:

It is not possible to entirely rule out temporary disorienting optical effects, particularly in conditions of dim lighting. Disorienting optical effects may come in the form of dazzle, flash blindness, afterimages, photosensitive epilepsy, or impairment of color vision, for example.

For all PLB variants:

WARNING
Electrical voltage!

Electrical voltage can cause severe injury or death.

- Work on electrical systems must only be performed by qualified electricians.
- The power supply must be disconnected when attaching and detaching electrical connections.
- The product must only be connected to a voltage supply as set out in the requirements in the operating instructions.
- National and regional regulations must be complied with.
- Safety requirements relating to work on electrical systems must be complied with.

WARNING
Risk of injury and damage caused by potential equalization currents!

Improper grounding can lead to dangerous equipotential bonding currents, which may in turn lead to dangerous voltages on metallic surfaces, such as the housing. Electrical voltage can cause severe injury or death.

- Work on electrical systems must only be performed by qualified electricians.
- Follow the notes in the operating instructions.
- Install the grounding for the product and the system in accordance with national and regional regulations.

3 Product description

3.1 Operating principle and basic characteristics

The PLB system is a flexible robot guidance system. It is used for three-dimensional localization of randomly oriented parts or features in a defined search volume. The search volume can also be adapted to a bin with walls for collision detection. The PLB system locates the parts and determines their precise position in the search volume. The system uses this information to calculate the coordinates of a located part and transfers these to the robot controller. The robot controller is then able to guide the robot precisely to the part in the search volume.

In more detail, and using a bin picking example, a 3D camera acquires a precise image of the search volume's contents. The system uses the image to calculate a 3D point cloud.

The system compares the shape of previously defined parts with the 3D point cloud. Once it has determined the precise position of a part, the system uses the defined CAD model of the robot gripper to determine whether the part can be picked up without this causing the gripper to collide with other parts or obstacles.

The system outputs the position of the part in the robot coordinate system together with a list of validated pick poses in which the robot can pick up the part without collision.

When the robot has successfully picked up the part, the robot controller sends a command to the system instructing it to acquire a new image of the search volume's content and determine the position of the next part.

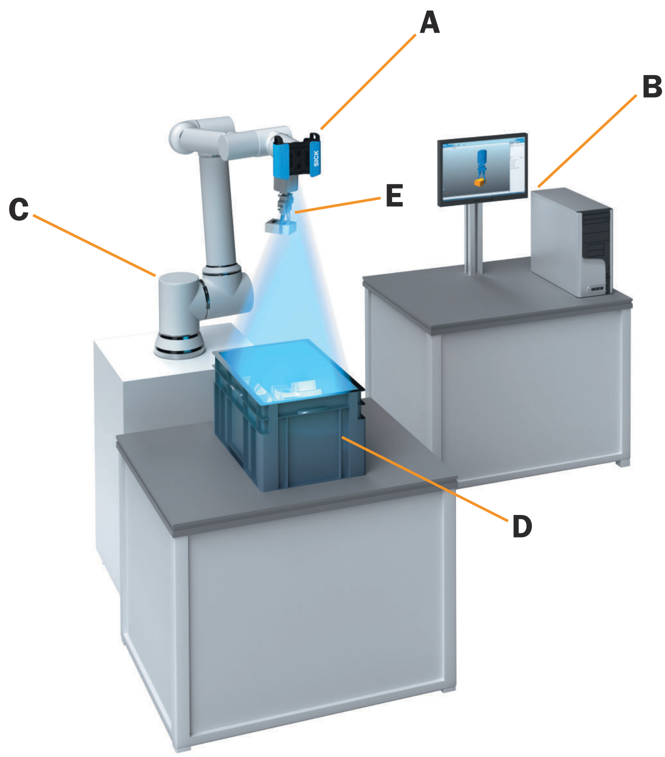


Figure 1: The PLB system in a bin picking robot cell

A	Camera (sensor for data acquisition)
B	Computer with PLB software
C	Robot
D	Search volume (bin)
E	Robot gripper

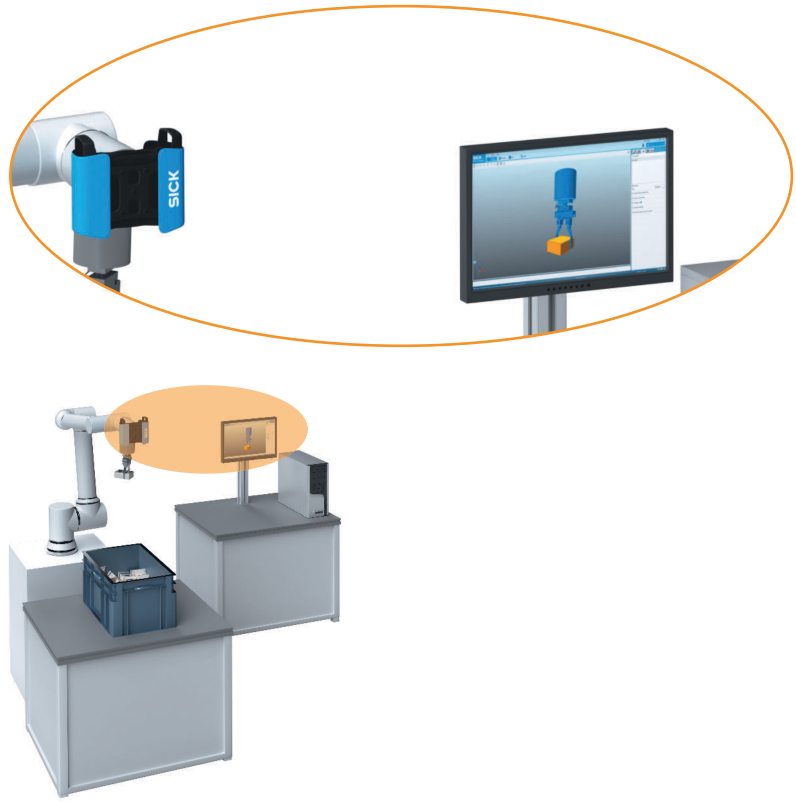


Figure 2: Detailed view: PLB system consisting of camera and software

3.2 Scope of delivery

The system consists of the following components:

- 3D high-end camera: The camera acts as a data streamer which sends the image data to the PC via a Gigabit Ethernet interface
- PLB Engine: Runtime environment which performs the calculations and analyzes the data
- PLB Studio: Application program for job configuration and data viewing

The following accessories are included in the system's scope of delivery:

- Quickstart for the camera setup
- Software license key

The following accessories and components are required and not included:

- Accessories, see www.sick.com/PLB.
    - Plug connectors and cables
    - Power supply units and cables
    - Mounting device
    - Alignment target
- PC with high performance graphics and a network interface card or USB connection, see "Recommended PC specifications", page 110 for details on the specific PLB system type.

3.3 Mounting concepts

In this section the two mounting concepts are described, including their variations regarding hand-eye alignment and robot commands.

This PLB system can be mounted in one of two ways:

- Mounted in a fixed position overlooking the robot cell. We call this a **stationary** camera.
- Mounted on the robot arm. We call this a **robot mounted** camera.

The PLB system default setting is that of the camera being stationary. There are variations between the two mounting concepts regarding the hand-eye alignment, and in some of the commands from the robot to the system.

The mounting concept of a device is set when adding the device to **PLB Studio**, see "Adding a camera", page 45.

Hand-eye alignment

In the hand-eye alignment process, the PLB system calulates a transform between the camera's and the robot's coordinate systems, to be able to pick the located parts. Since there is a variation of how the camera is mounted, there are two seperate processes on how to do the hand-eye alignment. For more information, see "Hand-eye alignment", page 53.

Robot commands

The robot commands affected by the robot mounted cameras are commands for hand-eye alignment and commands that include triggering an image. A triggered image by a robot mounted camera needs to have the positioning of the robot arm at that moment to be able to calculate where to pick the located part, which is not the case with a stationary camera.

! **NOTICE**

Robot mounted camera

All robot mounted commands for triggering an image acquisition must always include the position and rotation of the robot's flange from the default origin of the robot (usually the robot base).

If triggering an image acquisition from PLB Studio, PLB will use the latest position and rotation sent in a command from the robot.

The commands that differ for a robot mounted camera are named with "RM" to the end of the command. For more information, see "Robot commands", page 152.

3.4 Product characteristics

3.4.1 Dimensional drawings

3.4.1.1 Ensenso X36

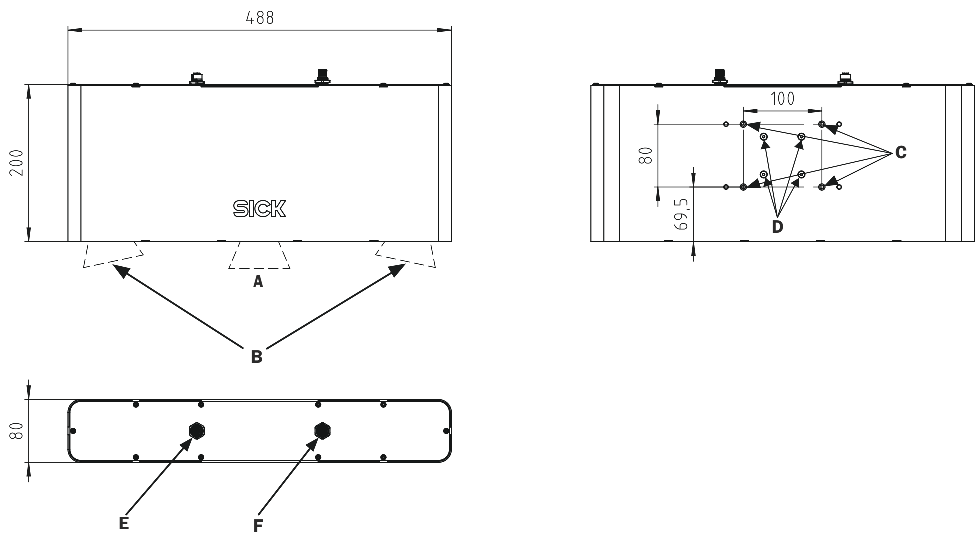


Figure 3: Ensenso X36 camera dimensional drawing (not true-to-scale)

A	LED projection module
B	Image sensors
C	Fastening threads (M8 x 6 mm)
D	M8x1.25
E	Ethernet (M12, 8 pin, x-coded)
F	Power (M12, 5 pin, L-coded)

3.4.1.2 Ensenso N36

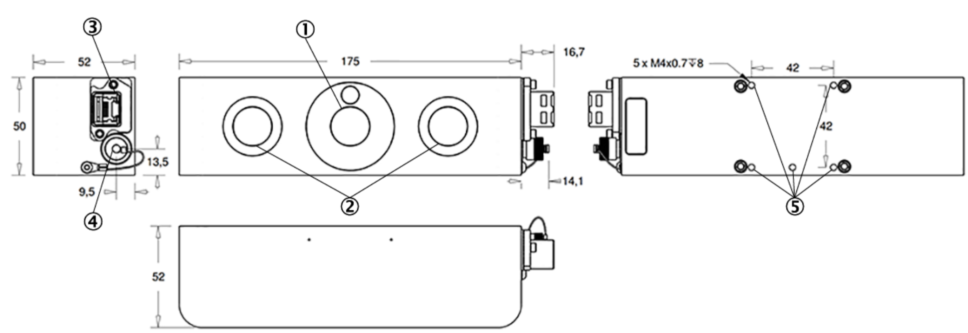


Figure 4: Ensenso N36 camera dimensional drawing (not true-to-scale)

①	LED projection module
②	Image sensors
③	Ethernet (RJ45)
④	Power and I/O (Binder 712, 7-pin)
⑤	Fastening threads (5x M4x0.7)

3.4.1.3 Ensenso C57

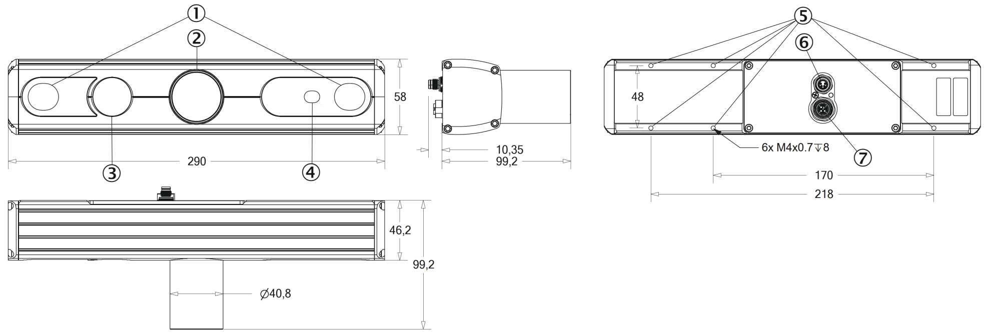


Figure 5: Ensenso C57 (PLB522) camera dimensional drawing (not true-to-scale)

①	3D sensors
②	Projector
③	LED projection module
④	2D sensor
⑤	Fasthening threads (6x M4x0,7)
⑥	Power (M8, 4-pin)
⑦	Ethernet (M12, 8 pin, x-coded)

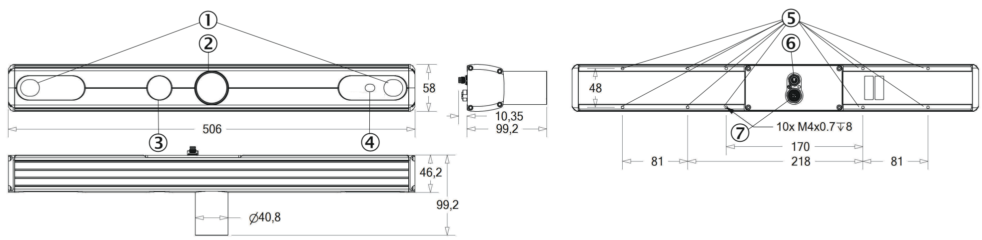


Figure 6: Ensenso C57 (PLB521) camera dimensional drawing (not true-to-scale)

①	3D sensors
②	Projector
③	LED projection module
④	2D sensor
⑤	Fasthening threads (10x M4x0,7)
⑥	Power (M8, 4-pin)
⑦	Ethernet (M12, 8 pin, x-coded)

3.4.1.4 Visionary-S

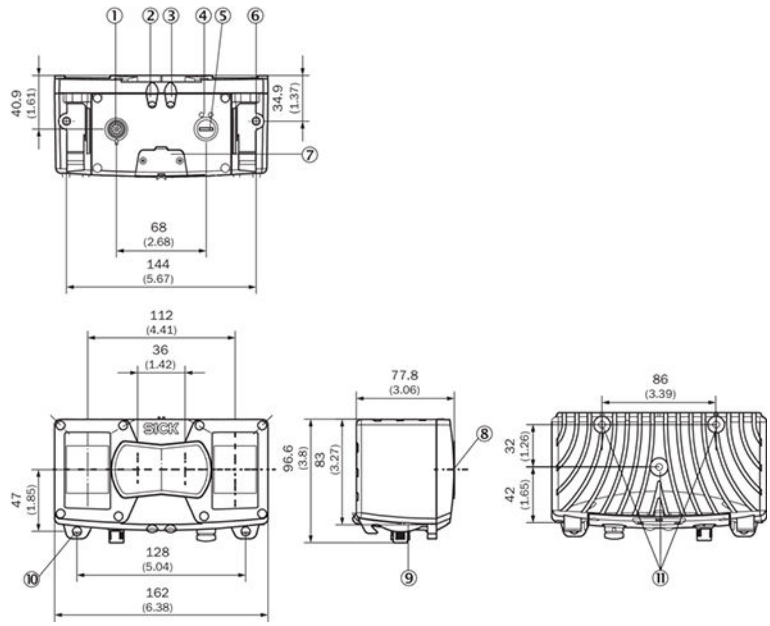


Figure 7: Visionary-S camera dimensional drawing (not true-to-scale)

① Power and I/O (M12, 17-pin)

② Device display

③ Application display

④ Ethernet status displays

⑤ Ethernet (M12, 8-pin, x-coded)

⑥ Fastening threads (2x M6, 7 mm deep)

⑦ Service interface

⑧ Optical axis

⑨ Bracket interface

⑩ Bracket attachment

⑪ Fastening threads (3x M6, 10 mm deep)

3.4.1.5 Visionary-T Mini

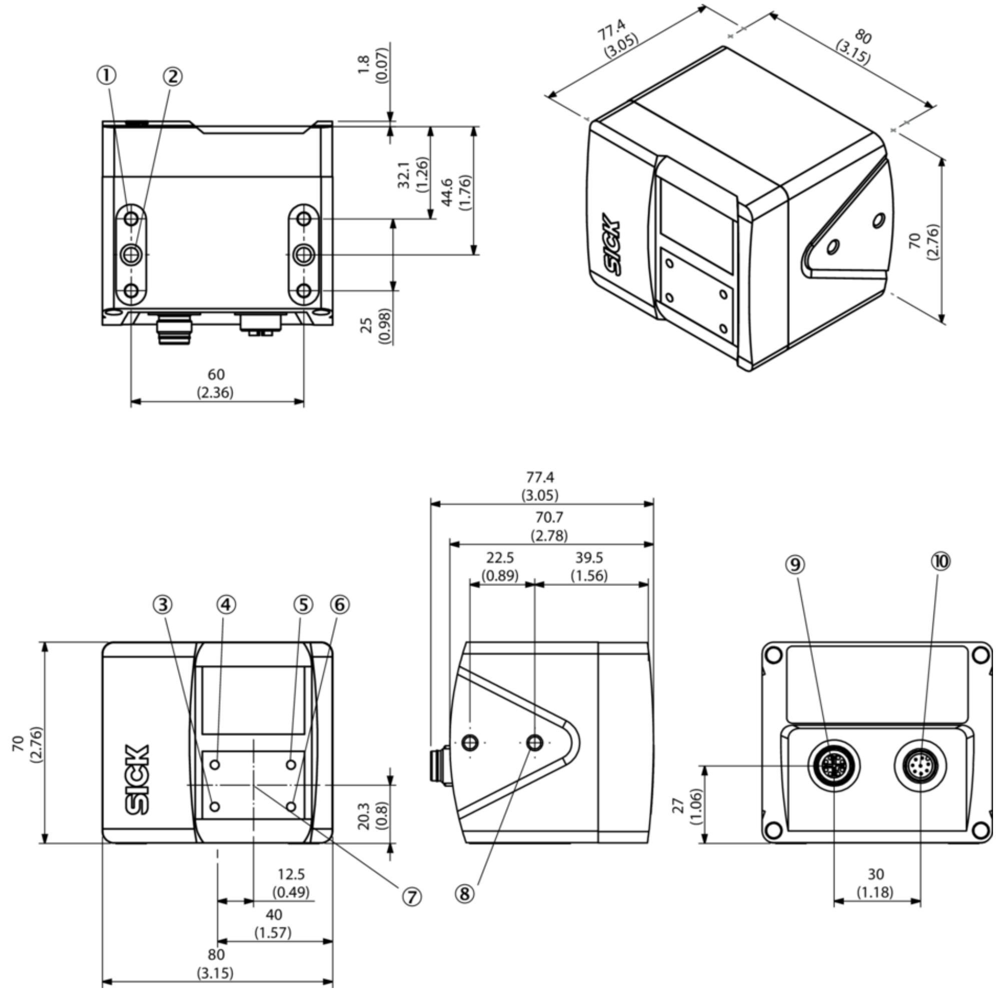


Figure 8: Visionary-T Mini camera dimensional drawing (not true-to-scale)

①	Threaded mounting hole (4x M5, 7.5 mm depth)
②	Fit (2x ø 5H7, 7 mm depth)
③	Device status display
④	Application status display
⑤	Ethernet status display
⑥	Application status display
⑦	Sensor coordinate origin
⑧	Threaded mounting hole (4x M5, 5.5 mm depth)
⑨	Ethernet (8-pin, M12, X-coded)
⑩	Power and I/O (8-pin, M12, A-coded)

3.4.1.6 Zivid 2+

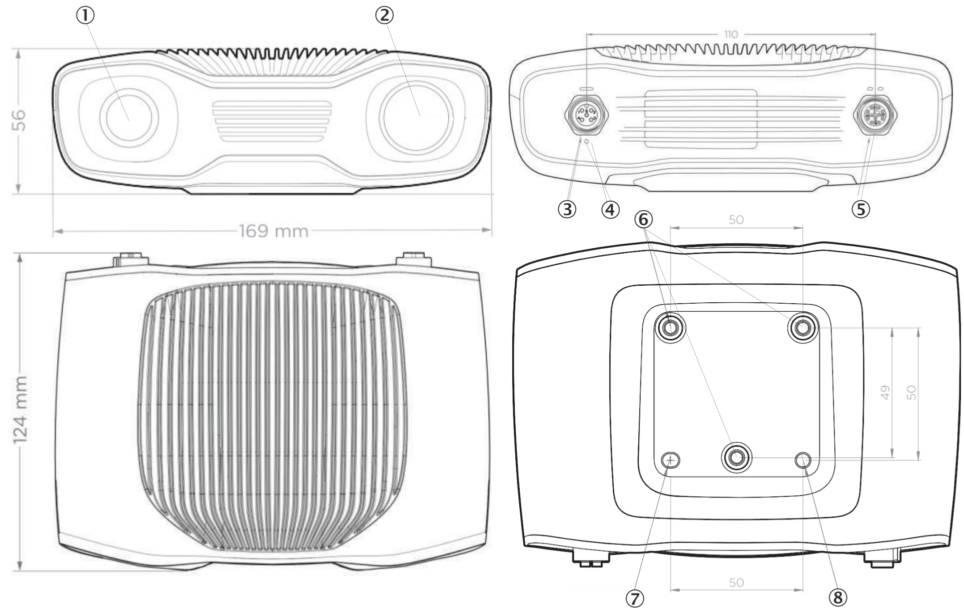


Figure 9: Zivid 2+ camera dimensional drawing (not true-to-scale)

① Projector
② 3D Camera
③ Power and I/O (M12, 5 pin)
④ Factory reset button
⑤ Ethernet (M12, 8-pin, x-coded)
⑥ Fastening threads (3x M5x0.8)
⑦ Alignment hole (Ø 5x1 H11)
⑧ Alignment hole (Ø 5 H11)

3.4.2 Volume of view

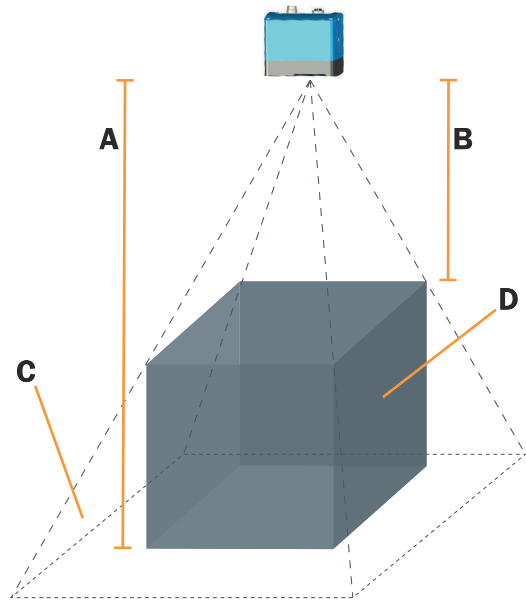


Figure 10: Example image Volume of View

A Maximum distance
B Minimum distance
C Volume of view
D Search volume

Table 1: PLB520 (X36 5MP)

Parameter	PLB520-S 5MP	PLB520-M 5MP	PLB520-L 5MP
Min. distance	800 mm	1400 mm	1500 mm
Max. distance	1200 mm	2200 mm	2500 mm
Field of view at min. distance[1]	700x600 mm	800x700 mm	1400x1200 mm
Field of view at max. distance[1]	800x900 mm	1300x1200 mm	2200x2000 mm
Example search volume[2]	400x300x200 mm	800x600x400 mm	Euro-pallet: 1200x800x1000 mm US-pallet: 1016x1219x1000 mm

Table 2: PLB515 (N36)

Parameter	PLB515-S	PLB515-M	PLB515-L
Min. distance	300 mm	600 mm	1000 mm
Max. distance	400 mm	1000 mm	1500 mm
Field of view at min. distance[1]	300x200 mm	400x300 mm	600x500 mm
Field of view at max. distance[1]	400x250 mm	1100x700 mm	1200x750 mm
Example search volume[2]	300x200x100 mm	400x300x400 mm	600x500x500 mm

Table 3: PLB521 (C57)

Parameter	PLB521-S	PLB521-M	PLB521-L
Min. distance	1600 mm	1500 mm	1200 mm
Max. distance	2150 mm	2500 mm	5000 mm
Focus distance	1850 mm	1850 mm	1850 mm
Field of view at min. distance[1]	810x700 mm	1160x980 mm	1180x1040 mm
Field of view at max. distance[1]	1010x940 mm	1690x1640 mm	4020x4330 mm
Field of view at focus distance[1]	940x810 mm	1370x1210 mm	1770x1600 mm
Example search volume[2]	650x700x400 mm	950x1000x600 mm	900x800x800 mm

Table 4: PLB522 (C57)

Parameter	PLB522-S	PLB522-M	PLB522-L
Min. distance	910 mm	860 mm	750 mm
Max. distance	1060 mm	1140 mm	1450 mm
Focus distance	970 mm	970 mm	970 mm
Field of view at min. distance[1]	470x400 mm	660x560 mm	770x650 mm
Field of view at max. distance[1]	510x460 mm	800x740 mm	1270x1250 mm
Field of view at focus distance[1]	490x420 mm	720x630 mm	930x840 mm
Example search volume[2]	200x450x100 mm	600x550x400 mm	700x600x600 mm

Table 5: PLB540 (Visionary-T Mini)

Parameter	PLB540
Min. distance	500 mm
Max. distance	2000 mm
Field of view at min. distance[1]	700x600 mm
Field of view at max. distance[1]	2800x2300 mm
Example search volume[2]	700x600x1500 mm

Table 6: PLB510 (Visionary-S)

Parameter	PLB510
Min. distance	1000 mm
Max. distance	2000 mm
Field of view at min. distance[1]	1000x900 mm
Field of view at max. distance[1]	2200x1800 mm
Example search volume[2]	1000x900x1000 mm

Table 7: PLB536 (2+)

Parameter	PLB536
Min. distance	1300 mm
Max. distance	2000 mm
Field of view at min. distance[1]	750x650 mm
Field of view at max. distance[1]	2000x1300 mm

Parameter	PLB536
Example search volume[2]	750x440x400 mm

1 Length x Width

2 Length x Width x Height

4 Transport and storage

4.1 Transport

NOTICE
Damage due to improper transport!

- The product must be packaged with protection against shock and damp.
- Recommendation: Use the original packaging.
- Note the symbols on the packaging.
- Do not remove packaging until immediately before you start mounting.

4.2 Unpacking

- To protect the device against condensation, allow it to equilibrate with the ambient temperature before unpacking if necessary.
- Handle the device with care and protect it from mechanical damage.
- To avoid ingress of dust and water, only remove the protective elements, e.g. protective caps of the electrical connections just before attaching the connecting cable.

4.3 Transport inspection

Immediately upon receipt in Goods-in, check the delivery for completeness and for any damage that may have occurred in transit. In the case of transit damage that is visible externally, proceed as follows:

- Do not accept the delivery or only do so conditionally.
- Note the scope of damage on the transport documents or on the transport company's delivery note.
- File a complaint.

NOTE
Complaints regarding defects should be filed as soon as these are detected. Damage claims are only valid before the applicable complaint deadlines.

4.4 Storage

Store the device under the following conditions:

- Recommendation: Use the original packaging.
- Do not store outdoors.
- Store in a dry area that is protected from dust.
- So that any residual damp can evaporate, do not package in airtight containers.
- Do not expose to any aggressive substances.
- Protect from sunlight.
- Avoid mechanical shocks.
- Storage temperature: see "Ambient data", page 109.
- For storage periods of longer than 3 months, check the general condition of all components and packaging on a regular basis.

5 Mounting

5.1 Mounting instructions

The correct installation of the device is a crucial factor determining the measurement accuracy of the system.

- The device is fixed properly in position and that all the screws are tightened.
- Typical space requirement: See "Product characteristics", page 17 for type-specific field of view diagrams and dimensional drawings.
- Comply with technical data, such as the permitted ambient conditions for operation, see "Technical data", page 107.
- To prevent condensation, avoid exposing the device to rapid changes in temperature
- Protect from direct sunlight
- Try to minimize risk of reflections on the objects to be detected
- Ensure that there is good heat transfer from the device, in particular at high ambient temperatures (e.g., via the bracket to the mounting base or ensure that the back of the device is a sufficient distance from the wall of a housing)
- Only to be mounted using the threaded mounting holes provided for this purpose.
- Clear view of the objects to be detected
- Shock and vibration-free mounting. Vibrations and shocks impair the accuracy of measurement. The mounting frame used for device installation should be stable enough to withstand both permanent and intermittent vibrations.

5.2 Required parts

You need the following parts to mount and connect the PLB system:

- PLB device
- Mounting device (bracket) with sufficient load-bearing capacity and suitable dimensions, supplied by the customer.
- Screws for mounting on the mounting device. Screw length is dependent on the mounting base (wall thickness of the bracket).
  - When using an optional SICK mounting device, the screws for mounting are included with delivery.
- When proceeding with connection, see "Electrical installation", page 27. Technical type-specific requirements (see "Technical data", page 107):
  - PC
  - Power supply
  - Power cable
  - Ethernet cable

6 Electrical installation

6.1 Notes on electrical installation

NOTICE
Equipment damage due to incorrect supply voltage!

An incorrect supply voltage may result in damage to the equipment.

The device may only be powered using a voltage source that meets the following requirements:

- SELV: (EN / IEC 61010-1 or IEC 60950-1) or ES-1 (EN IEC 62368-1)

NOTICE
Equipment damage or unpredictable operation due to working with live parts!

Working with live parts may result in unpredictable operation.

- Only carry out wiring work when the power is off.
- Only connect and disconnect electrical connections when the power is off.

- **The electrical installation must only be performed by electrically qualified personnel.**
- **Standard safety requirements must be met when working on electrical systems!**
- Only switch on the supply voltage for the device when the connection tasks have been completed and the wiring has been thoroughly checked.
- When using extension cables with open ends, ensure that bare wire ends do not come into contact with each other (risk of short-circuiting when the supply voltage is switched on). Wires must be appropriately insulated from each other.
- Wire cross-sections in the supply cable from the customer's power system must be designed in accordance with the applicable standards. When this is being done in Germany, observe the following standards: DIN VDE 0100 (Part 430) and DIN VDE 0298 (Part 4), and/or DIN VDE 0891 (Part 1).
- Circuits connected to the device must be designed as SELV circuits (SELV = Safety Extra Low Voltage).
- Protect the device with a separate fuse at the start of the supply circuit.

NOTE
Layout of data cables

- Use screened data cables with twisted-pair wires.
- Implement the screening design correctly and completely.
- To avoid interference, e.g. from switching power supplies, motors, clocked drives, and contactors, always use cables and layouts that are suitable for EMC.
- Do not lay cables over long distances in parallel with power supply cables and motor cables in cable channels.

The enclosure rating for the device is only achieved under the following conditions:

- The cables plugged into the connectors are screwed tight.
- Any electrical connections that are not being used must be fitted with protective caps/plugs that are screwed tight (as in the delivery condition).

6.2 Prerequisites for the safe operation of the device in a system

WARNING
Risk of injury and damage caused by electrical current!
As a result of equipotential bonding currents between the SICK device and other grounded devices in the system, faulty grounding of the SICK device can give rise to the following dangers and faults:

- Dangerous voltages are applied to the metal housings
- Devices will behave incorrectly or be destroyed
- Cable shielding will be damaged by overheating and cause cable fires

Remedial measures

- ► Only skilled electricians should be permitted to carry out work on the electrical wiring.
- ► Ensure that the ground potential is the same at all grounding points.
- ► If the cable insulation is damaged, disconnect the voltage supply immediately and have the damage repaired.
- ► Where local conditions are unfavorable and therefore do not meet conditions for a safe grounding method (same ground potential at all grounding points), take measures in accordance with the following formats.

The device is connected to the peripheral devices (voltage supply, any local trigger sensor(s), PLC) via shielded cables. The cable shield – for the data cable, for example – rests against the metal housing of the SICK device. The device can either be grounded through the cable shield or through one of the threaded mounting holes.

If the peripheral devices have metal housings and if the cable shields also lie on their housings, it is assumed that all devices involved in the installation have the **same ground potential**.

This is achieved by complying with the following conditions:

- Mounting the devices on conductive metal surfaces
- Correct grounding of the devices/metal surfaces in the system.
- If necessary: low-impedance and current-carrying equipotential bonding between areas with different ground potentials

If these conditions are not fulfilled, equipotential bonding currents can flow along with the cable shielding between the devices due to differing ground potentials; this can be dangerous. This is, for example, possible in cases where there are devices within a widely distributed system covering several buildings.

Remedial measures

The most common solution to prevent equipotential bonding currents on cable shields is to ensure low-impedance and current-carrying equipotential bonding. If this is not possible, the following solution approaches serve as a suggestion.

NOTICE
We expressly advise against opening up the cable shields. This would mean that the EMC limit values can no longer be complied with and that the safe operation of the device data interfaces can no longer be guaranteed.

Measures for widely distributed system installations

On widely distributed system installations with correspondingly large potential differences, the setting up of local islands and connecting them using commercially available **electro-optical signal isolators** is recommended. This measure achieves a high degree of resistance to electromagnetic interference.

The use of electro-optical signal isolators between the islands isolates the ground loop. Within the islands, a stable equipotential bonding prevents equalizing currents on the cable shields.

Measures for small system installations

For smaller installations with only slight potential differences, insulated mounting of the SICK device and of peripheral devices may be a sufficient solution.

Even in the event of large differences in the ground potential, ground loops are effectively prevented. As a result, equalizing currents can no longer flow via the cable shields and metal housing.

NOTICE
The power supply for the SICK device and the connected peripheral devices must also guarantee the required level of insulation.
Under certain circumstances, a tangible potential can develop between the insulated metal housings and the local ground potential.

6.3 Connectors and pin assignment

6.3.1 Ensenso X36 specifications

The X36 camera uses a M12, 5-pin female connector, L-coded, for power, and a M12, 8-pin connector, x-coded, for ethernet.

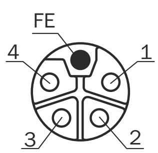


Figure 11: Ensenso X36, power connector

Table 8: Pin assignments for the Ensenso X36 power connector

Pin	Signal
1	Not connected
2	Ground (GND)
3	Not connected
4	Power Supply Input 24V DC, 5A
FE	Functional earth

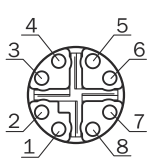


Figure 12: Ensenso X36, ethernet connector

Table 9: Pin assignments for the Ensenso X36 ethernet connector

Pin	Signal
1	TRD0_P
2	TRD0_N
3	TRD1_P
4	TRD1_N

Pin	Signal
5	TRD3_P
6	TRD3_N
7	TRD2_P
8	TRD2_N

6.3.2 Ensenso N36 specifications

The N36 camera uses a Binder 712, 7-pin male connector, for power and a RJ45 connector for ethernet.

Alternatively, connect the camera to a PoE injector or PoE compatible switch for power supply (see accessories on www.sick.com/PLB).

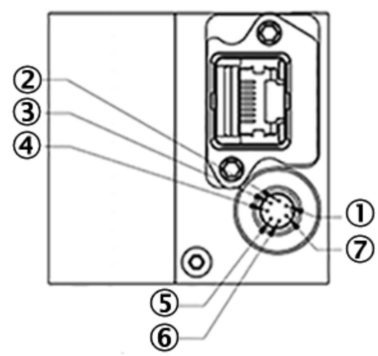


Figure 13: Ensenso N36, power connector

Table 10: Pin assignments for the Ensenso N36 power connector

Pin	Signal
1	Input +
2	Power (12 ... 24 V)
3	Shield
4	Ground (GND)
5	Output +
6	Output -
7	Input -

6.3.3 Ensenso C57 specifications

The PLB521 and PLB522 cameras uses a 4-pin male M8 connector for power, and a 8-pin male M12 connector, x-coded, for ethernet.

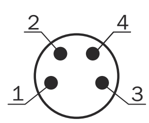


Figure 14: PLB521, PLB522 power connector

Table 11: Pin assignments for the PLB521, PLB522 power connector

Pin	Signal
1	Power Supply Input 24V DC
2	Not connected

Pin	Signal
3	Ground (GND)
4	Not connected

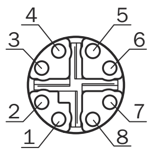


Figure 15: PLB521, PLB522 ethernet connector

Table 12: Pin assignments for the PLB521, PLB522 ethernet connector

Pin	Signal
1	TRD0_P
2	TRD0_N
3	TRD1_P
4	TRD1_N
5	TRD3_P
6	TRD3_N
7	TRD2_P
8	TRD2_N

6.3.4 Visionary-S specifications

The Visionary-S camera uses a M12, 17-pin male connector, for power, and a M12, 8-pin connector, X-coded, for ethernet.

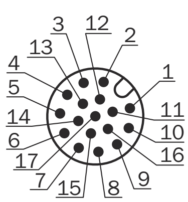


Figure 16: Visionary-S, power connector

Table 13: Pin assignments for the Visionary-S power connector

Pin	Signal
1	Ground (GND)
2	24 V DC ± 15 % – supply voltage
3	Not connected
4	Not connected
5	Not connected
6	Not connected
7	TxD (RS-232), Aux – service only
8	RxD (RS-232), Aux – service only
9	SENS GND – GND for electrically decoupled inputs
10	SENS IN1 – switch input, electrically decoupled
11	Not connected
12	Not connected
13	INOUT 1

Pin	Signal
14	INOUT 2
15	SENS IN2 – switch input, electrically decoupled
16	INOUT 3
17	INOUT 4

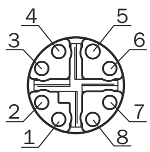


Figure 17: Visionary-S, ethernet connector

Table 14: Pin assignments for the Visionary-S ethernet connector

Pin	Signal
1	TRD0_P
2	TRD0_N
3	TRD1_P
4	TRD1_N
5	TRD3_P
6	TRD3_N
7	TRD2_P
8	TRD2_N

Status LEDs

Device	Application	Description
Any	Blue	Data transmission: API channel deactivated and diagnostic channel active
Any	Blue - flashing slowly	Data transmission deactivated
Any	Green	API channel, data transmission active
Blue	Off	Illumination off
Blue - flashing slowly	Off	System start
Green	Blue - flashing slowly	Trigger mode active; waiting for trigger
Orange - flashing slowly	Any	Device warning, e.g., temperature exceeds warning level
Red - flashing slowly	Red	Max. operating temperature exceeded

6.3.5 Visionary-T Mini specifications

The Visionary-T mini camera uses a M12, 8-pin, A-coded connector, for power, and a M12, 8-pin connector, X-coded, for ethernet.

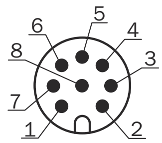


Figure 18: Visionary-T Mini, power connector

Table 15: Pin assignments for the Visionary-T Mini power connector

Pin	Signal
1	Supply voltage (24 V DC -30% ... +25%)
2	INOUT 3
3	GND – reference mass
4	INOUT 4
5	INOUT 1
6	INOUT 5
7	INOUT 6
8	INOUT 2

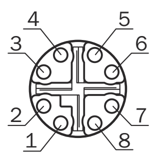


Figure 19: Visionary-T Mini, ethernet connector

Table 16: Pin assignments for the Visionary-T Mini ethernet connector

Pin	Signal
1	TRD0_P
2	TRD0_N
3	TRD1_P
4	TRD1_N
5	TRD3_P
6	TRD3_N
7	TRD2_P
8	TRD2_N

Status LEDs

Square symbol	Circle symbol	Description
Green	Off	System start
Green	Off	Data transmission: API channel deactivated and ready for data transmission
Green	Blue	Data transmission: API channel deactivated - SOPAS active
Green	Green	Data transmission: API channel active
Orange - flashing slowly	All	Device warning, e.g., temperature exceeds warning limit
Red - flashing slowly	Red	Max. operating temperature exceeded or short-circuit

Square symbol	Circle symbol	Description
Red - flashing quickly	Red	Illumination fault

6.3.6 Zivid 2+ specifications

The Zivid 2+ camera use a M12, 5-pin connector for power and a M12, 8-pin, x-coded, connector for ethernet.

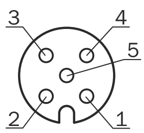


Figure 20: Zivid 2+, power connector

Table 17: Pin assignments for the Zivid 2+ power connector

Pin	Signal
1	24 V DC ± 20 % (max 5A)
2	24 V DC ± 20 % (max 5A)
3	Ground (GND)
4	Ground (GND)
5	Not connected

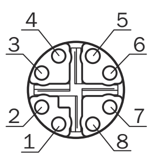


Figure 21: Zivid 2+, ethernet connector

Table 18: Pin assignments for the Zivid 2+ ethernet connector

Pin	RJ45 (color)
1	1 (WH/OG)
2	2 (OG)
3	3 (WH/GN)
4	6 (GN)
5	7 (WH/BN)
6	8 (BN)
7	5 (WH/BU)
8	4 (BU)

Status LED specification

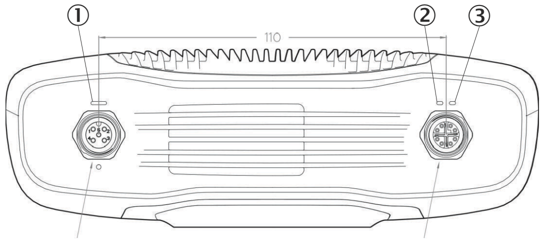


Figure 22: Zivid 2+, control LED

① Power light
② Ethernet speed light
③ Ethernet link light

Table 19: Status LED specification for the Zivid 2+

Light source	Color	Description
Power	Blue	Power OK
	Red	Error
Ethernet speed	Blue	10 Gigabit/s
	Yellow	5, 2.5 Gigabit/s
	Green	1 Gigabit/s
	Red	Error
Ethernet link	Blue	Activity
	Red	Error

6.4 Electrical installation and network integration

WARNING

- Before commencing installation, make sure that the system is powered down.
- Make sure that the camera is turned off during installation.

First of all, you must connect the camera to the computer on which the PLB software will be used (see "Recommended PC specifications", page 110). The following instructions are a general example, please see "Connectors and pin assignment", page 29 for any PLB type specific instructions regarding Ethernet and power.

1. Connect the camera's Ethernet port directly to the computer's network port using a Ethernet cable.
2. Switch off the power supply and connect it to the camera’s power socket.
3. Switch the power supply on again.

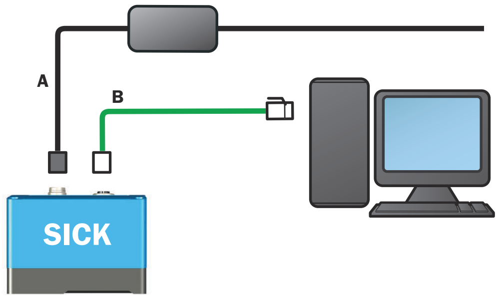


Figure 23: Connecting the camera (general example shown)

A Power supply incl. power cord

B Ethernet cable

NOTE

If connecting the camera to a network: Due to the large data volumes generated by the camera, you must set up a separate network for the system. Only by doing this can you prevent deterioration of communications due to other data traffic.

7 Configuration

7.1 PC configuration

7.1.1 Installing the PC software

You can download the PLB software from the SICK support website supportportal.sick.com. The first time you use the support website you need to register a user account.

NOTE

- Deactivate standby and idle mode on your computer.
- Deactivate all firewalls.
- Deactivate virus protection.
- Make sure that you possess administrator permissions.

Proceed as follows:

1. Establish a connection to the Internet and log into the site supportportal.sick.com.
2. Hover the **Systems** tab and click **Robot Guidance**.
3. Choose **PLB**.
4. In the list **Download** select the most recent available program version (see figure 24).
   - Download:
     - **PLB X.X** (runtime environment and configuration program)
     - **PLB X.X Xxx** (drivers and/or programs for the specific camera type)
5. Double-click the required file and follow the installation instructions.

✓ The file **PLB X.X** installs PLB Engine, PLB Studio, and all the necessary add-on programs.

6. See "Activating network and camera drivers", page 38 for network interface card configuration and installation regarding the specific camera type.

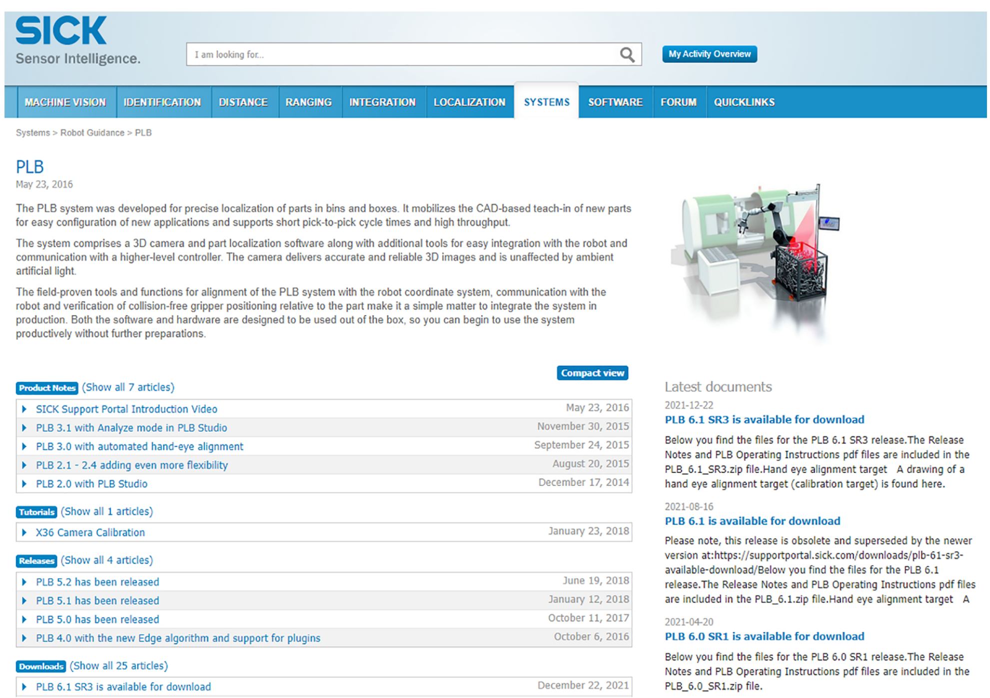


Figure 24: SICK support website, downloading the most recent PLB software

7.1.2 Activating network and camera drivers

7.1.2.1 PC-to-camera network interface card

We recommend using an Intel network interface card (NIC) for the communication between the PC running PLB and the camera, see "Recommended PC specifications", page 110.

For best performance, edit the NIC settings as described in this section. These settings are generally applicable for all PLB variants, the options can vary depending on the NIC being used.

NOTE
Install the driver from the manufacturer for the PC-to-camera network interface card. Do not use the default generic driver provided by Windows.

Enable jumbo frames

Make sure that jumbo frames are enabled for the NIC.

1. Open the **Control Panel**.
2. Click **Network and Sharing Center**.
3. In the menu to the left, click **Change adapter settings**.
4. Right-click **Local Area Connection** and click **Properties**.
5. In the **Local Area Connection** dialog, click **Configure**.

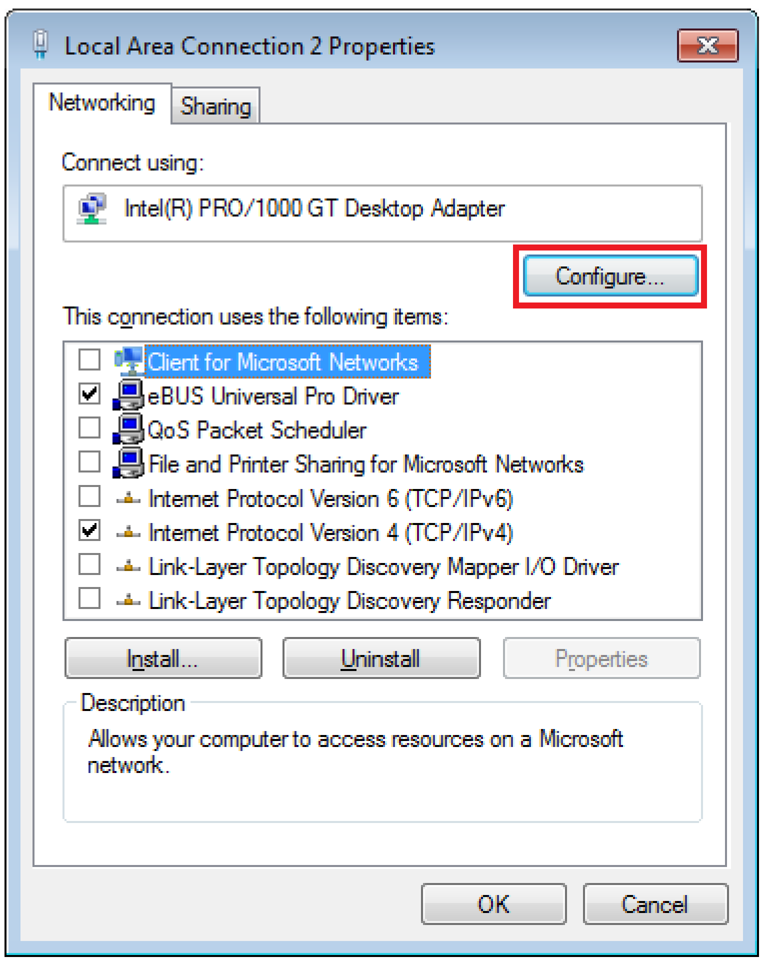


Figure 25: Local Area Connection Properties

6. Select the **Advanced** tab.

7. Select the **Jumbo packet** row and set the value to 4088 Bytes, or higher.

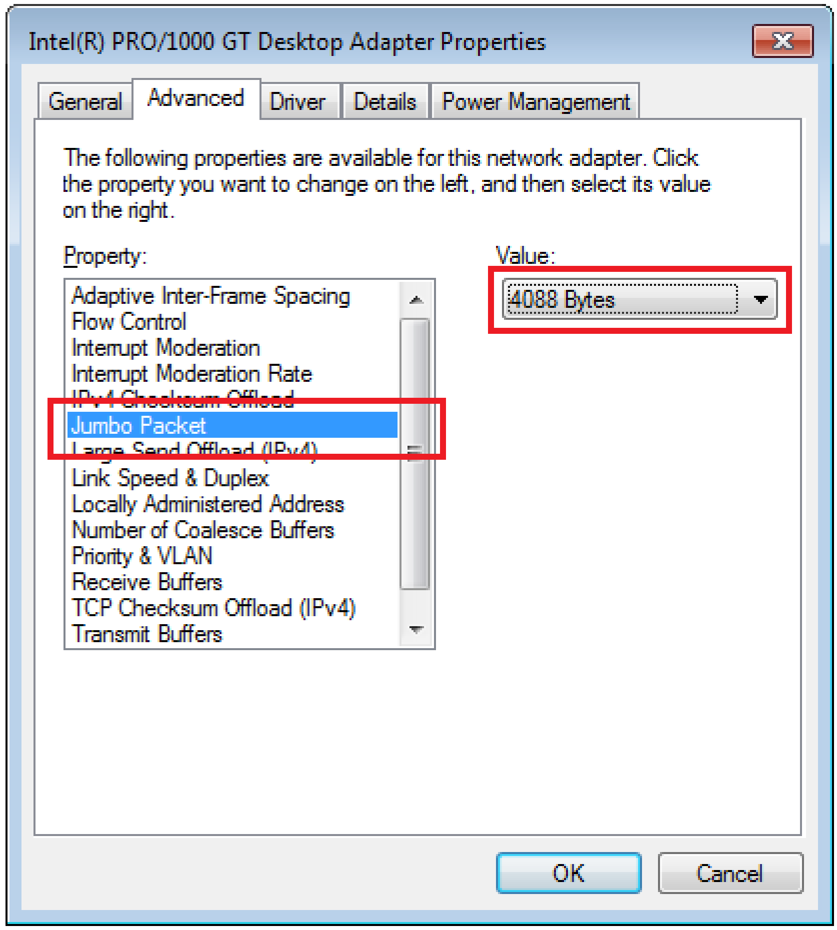


Figure 26: Jumbo frames settings

Maximize receive buffers

We recommend that you set the size of the NICs receive buffers to their maximum value.

1. Open the **Advanced** tab of the **Desktop Adapter Settings** dialog (see Enable jumbo frames)
2. Select **Receive buffers** and set the value to its maximum value.

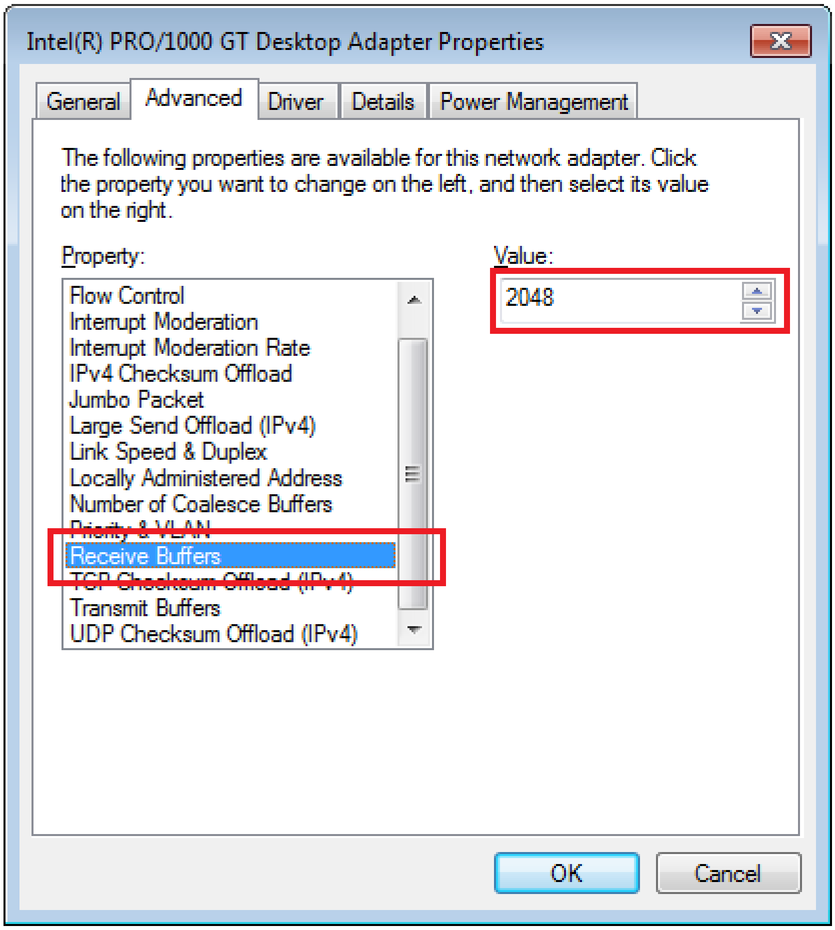


Figure 27: Receive buffers settings

7.1.2.2 Ensenso cameras

In order to use Ensenso cameras, the uEye network driver and the Ensenso SDK must be installed.

Proceed as follows:

1. Download the uEye and Ensenso SDK programs from the SICK Support Portal, as described in "Installing the PC software", page 37.
2. Double-click the uEye installer (.exe) to start the uEye installation.
3. When prompted to select a setup type, select **Complete** and click **Next**.
4. Follow the instructions on the screen to finish the installation. Once the software has been installed, the uEye network service is automatically bound to all local network adapters.
5. Double-click the EnsensoSDK installer (.exe) to start the installation.
6. Follow the instructions on the screen to finish the installation.
7. After finishing the installation, re-start the computer.

7.1.2.3 Visionary cameras

In order to use the Visionary cameras, the SOPAS Engineering Tool (ET) must be installed.

Proceed as follows:

1. Download the Visionary installation files from the SICK Support Portal, as described in "Installing the PC software", page 37.

2. Double-click the SopasET installer (.exe) to start the SOPAS ET installation.
3. Follow the instructions on the screen to finish the installation.

7.1.2.4 Zivid camera

In order to use Zivid camera, the Zivid SDK must be installed.

Proceed as follows:

1. Download the Zivid SDK programs from the SICK Support Portal, as described in "Installing the PC software", page 37.
2. Double-click the Zivid SDK installer (.exe) to start the installation.
3. Follow the instructions on the screen to finish the installation.

To change the IP addess for Zivid 2+, go to support.zivid.com and search for "Zivid 2+ Network Configuration" for more information.

7.1.3 Running PLB Studio

Before we examine the PLB Studio configuration function in detail, let's take a brief look at the user interface.

1. Click **Start** to open the **Start** menu.
2. Enter `PLB Engine`.
3. Click the **PLB Engine** entry to start the application.

PLB Studio is automatically started with PLB Engine. PLB Engine can be running while PLB Studio is closed.

To open PLB Studio while PLB Engine is running, right-click the PLB icon in the taskbar and click **Open PLB Studio**. see figure 28

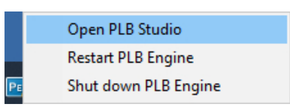


Figure 28: Open PLB Studio, restart and close PLB Engine.

At the top of the PLB Studio window (see figure 29) is the menu bar (**A**), where the current mode is displayed. You can select **Configure**, **Analyze**, or **Run** mode. It's only possible to select **Align** mode via the robot communication interface. The name of the active job is displayed in the list to the right of the menu bar. You select an active job and manage jobs in the list.

The visualization area (**B**) displays the image and results. At the top of this area, you find the image handling controls, the view controls, and the actions toolbar. In the lower-left corner, you find the 3D navigation control.

To the right of the visualization area, you find the parameter editing area (**C**). The tabs at the top of this area let you select what to configure; part, gripper, pick pose, bin, or system settings. The available parameters are displayed below.

The system log (**D**) and the status bar (**E**) are located at the bottom of the user interface. The system log contains warning and error messages.

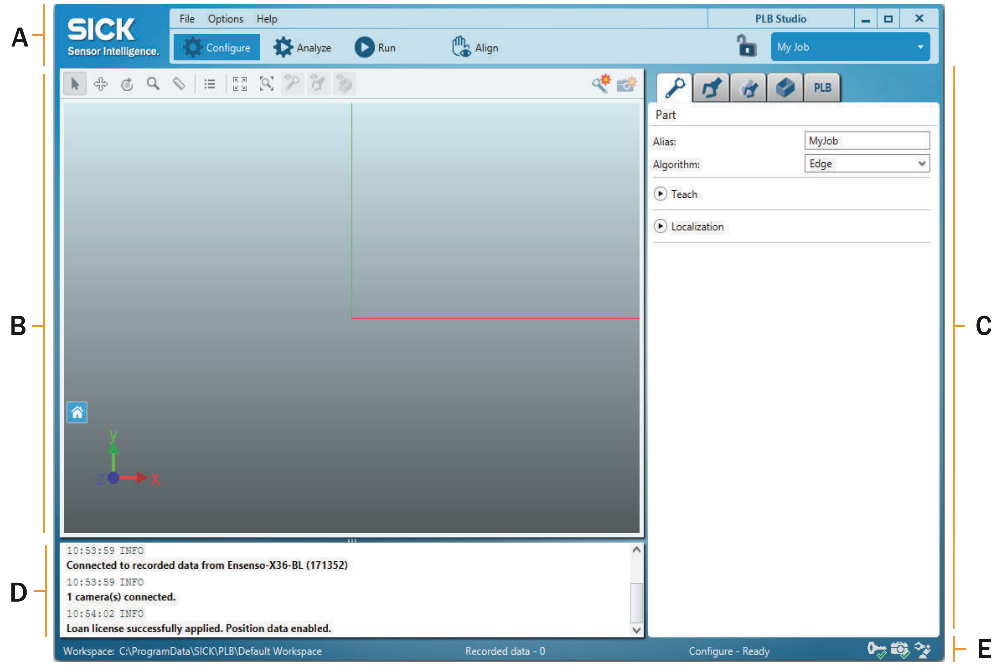


Figure 29: Overview of PLB Studio

A	Menu bar
B	Visualization area
C	Parameter editing area
D	System log
E	Status bar

Status bar

The status bar contains the following information:

- Path to the workspace containing all the system settings and the job library
- Mode: Configure, Analyze, Run, Align
- System status: Ready, Processing, Capturing
- License status, camera connection status, and robot controller connection status

Workspace folder

You access the program information and the Workspace folder via the menu bar. The Workspace folder contains the system and job configuration settings. By default, it is located under **C:\ProgramData\SICK\PLB\Default Workspace**. However, you can modify the location of the Workspace folder by choosing **File > Workspace folder** and specifying an alternative storage location. After modifying the path of the Workspace folder, you must restart the program manually in order for the changes to take effect. The program then automatically creates a system folder containing the default system settings and an empty job library.

7.1.3.1 Image handling and 3D navigation control

Image handling controls

Use the image handling controls to manipulate regions and perspective when viewing images. As an alternative to the buttons, you can use a mouse with a scroll wheel, as described below.

Button	Name	Description
	Select	Click to show the region marker tool. Only used for edge algorithm jobs in Configure mode.
	Move	Click and drag to move the image. **Shift** + click and hold the wheel button.
	Rotate	Click and drag to rotate the image. **Ctrl** + click and hold the wheel button.
	Zoom	Click and drag upwards to zoom in and downwards to zoom out. Rotate the mouse wheel.
	Distance measure-ment	Click on start point and then click on end point to draw a measure-ment line. Alternatively, click on start point and drag to end point. Next click adds a new measurement line from previous end point. Right click clears all measurement lines. Only available in point cloud views.

3D navigation control

Use the 3D navigation control in the lower-left corner of the image viewer to switch between different viewing angles:

- Click an arrowhead (X, Y or Z) to view a 2D projection of the object.
- Click the same arrowhead twice to flip the 2D projection (for example, to switch between the top and bottom view for the Z-axis).
- Click **Home** to restore the original viewing position.

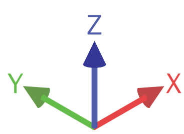


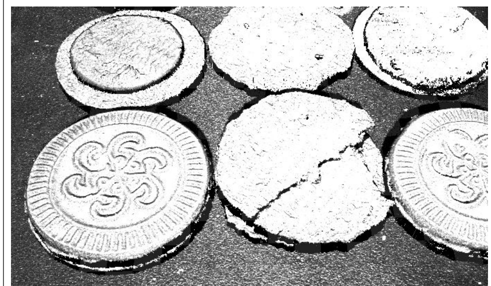
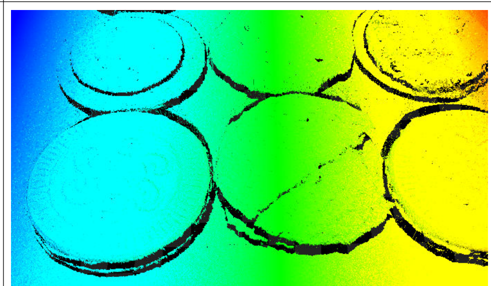
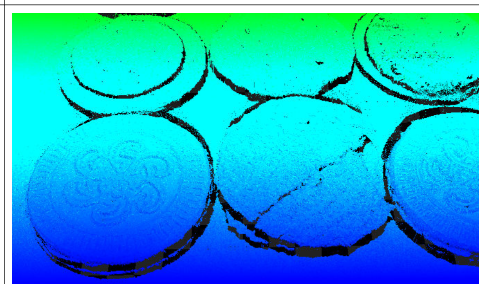
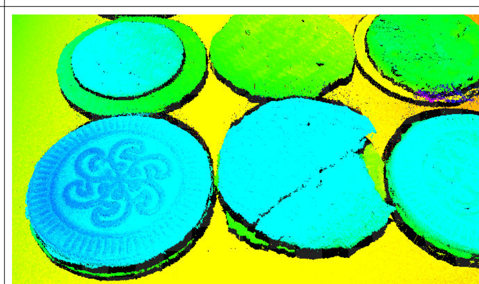
Figure 30: 3D navigation control

7.1.3.2 Image view options

Button	Name	Description
View	View	Select a view mode in the list, see "View modes", page 43.
	Color	View image in color (various options) or grayscale.
	Options	Options for **Color Range**, **Surface**, and **Points**, as described in this manual.
	Fullscreen	Set the Visualization window to fullscreen.
	Exit full-screen	Exit the fullscreen Visualization window.
	Zoom to extents	Zoom to display the maximum extents of all objects.

7.1.3.3 View modes

When you view a 3D image, you can select different ways to color it. The following view modes are available:

View mode	Description	Example
Intensity	Color is proportional to the reflectance values along the laser line. Suitable to show the surface details of an object, such as a print.	
X	Color is proportional to x-coordinate.	
Y	Color is proportional to y-coordinate.	
Z	Color is proportional to depth (z-coordinate). Suitable to show large variations in depth.	

7.2 Camera configuration

7.2.1 Calibration correction

NOTE
This section is only applicable for Ensenso and Zivid cameras.

The Ensenso and Zivid cameras have a method to verify and correct the calibration if anything has changed during transport. This is not a new calibration of the camera, only a correction of the factory calibration.

It is recommended to perform a calibration correction after the device is mounted, before the device is added to the PLB system. It can also be performed for maintenance purposes.

7.2.1.1 Ensenso dynamic recalibration

To proceed with the Ensenso dynamic recalibration, it is required to have a calibration target from Ensenso. It can be purchased as an accessory on www.sick.com/PLB.

1. Place the calibration target at the maximum distance of the intended working volume (depending on PLB type, see "Volume of view", page 22)
2. Open NxView
3. Select the correct camera
4. Click the **Open...** button

> **!** **NOTICE**
> Do NOT click the **Calibrate...** button. This calibration option will lead to the factory calibration being overwritten.

✓ A new window of NxView opens for the selected camera

5. Change the operation mode (to the right) by clicking the **Workplace calibration** tab, see figure 31

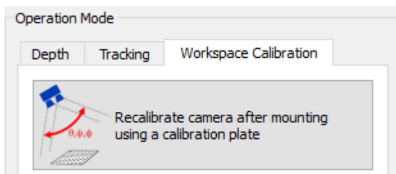


Figure 31: Operation mode: Workspace calibration, in NxView

6. Click the **Recalibrate camera after mounting using a calibration plate** option, see figure 31

✓ The image of the target in NxView will show lines and arrows in green and red. The red lines in the circles can vary in length, see figure 32.

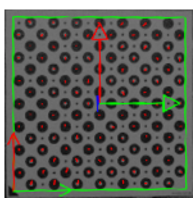


Figure 32: Recalibration target in NxView

7. Click **Start**

✓ The recalibration is performed

8. When finished, exit NxView

7.2.1.2 Zivid infield correction

To proceed with the Zivid infield correction, it is required to have a calibration target from Zivid. It can be purchased as an accessory on www.sick.com/PLB.

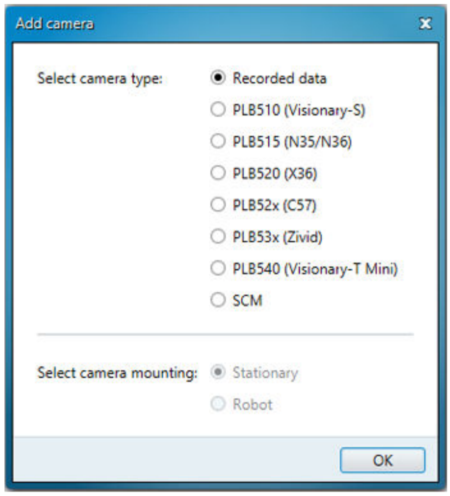
The guidelines and how to run the infield correction is found on support.zivid.com. Search for "Infield correction" and select the search result with the same name to get started.

7.2.2 Adding a camera

Before you establish the camera connection, you must add the camera to the PLB system. If the system uses multiple cameras, repeat the last three steps for all cameras.

1. Open **PLB Studio**.
2. In the menu bar, click **Configure** .
3. In the parameter editing area, click **System settings** .
4. Go to the settings for **Cameras, Type - Camera alignment** (see figure 33).
5. To add a camera, click **Create new camera** .

✓ The **Add camera** dialog box appears, see figure 34.
6. In the dialog box, select the camera type and mounting (see "Mounting concepts", page 15).
7. Click **OK**.
✓ The camera is added to the list with the mounting in parenthesis, and a camera alignment alias is automatically assigned to it. If it is the first camera that is added, the alias is 0 (zero).



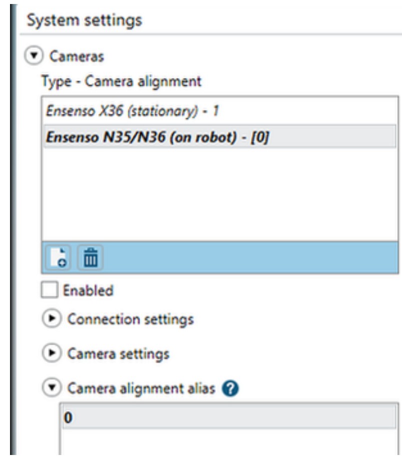


Figure 34: Add camera dialog box

Figure 33: Add camera and Type - Camera alignment

7.2.3 Adding and editing a camera alignment alias

When a camera is added to the system, it is automatically given a camera alignment alias. If the camera is going to be used at multiple positions, you need to assign a camera alignment alias to each position.

1. In the menu bar, click **Configure**.
2. In the parameter editing area, click **System settings**.
3. Go to the settings for **Camera, Type - Camera alignment**.
4. Select the camera.
5. Go to the settings for **Camera alignment alias**.
6. To add a new alias, click **Create new camera alignment**.
✓ A new alias is added to the list of the selected camera. As a default, it is set to the smallest available integer.
7. To edit an alias, click the edit icon and type the new name. The name must be unique.

7.2.4 Using multiple cameras or camera positions

It is possible to use multiple cameras in the PLB system. It is also possible to move a camera and use it in different positions. In those cases, you do the hand-eye alignment process for each camera and each camera position. Each camera or camera position is identified by a camera alignment alias.

The **Cameras,Type - Camera alignment** list shows all cameras in the system and their corresponding camera alignment aliases (see figure 33). Enabled cameras are shown in plain text and disabled cameras are shown in italics. Bold text means that the camera is selected. The selected camera alignment alias of that camera is shown in square brackets.

7.2.5 Connecting to Ensenso cameras

When using an Ensenso camera you need a camera parameter file. The camera parameter file contains settings to optimize the image quality for specific imaging distances and dimensions. You can either use the default parameter file supplied with the PLB system or create a parameter file using the Ensenso application NxView.

In order to use an Ensenso camera in PLB Studio, proceed as follows:

1. Open PLB Studio.
2. In the menu bar, click **Configure**.
3. In the parameter editing area, click **System settings** .
4. If there are multiple cameras in the system, in the **Cameras** section, select the camera in the **Type - Camera alignment** list.
5. Enable the camera.

✓ PLB will search for the camera, and add the serial number and a default parameter file if the camera is available on the network.

Optional:

6. To make further changes to the settings, disable the camera.
7. To import a custom parameter file from NxView, in the **Camera settings** section, under **Camera parameter file**, click and navigate to the destination folder where the file is located (see figure 35). The default parameter file supplied with the PLB system is located in the folder:
   `C:\ProgramData\SICK\PLB\Default Workspace\System`
8. If the C57 camera is used: Select the checkbox **2D color** to enable the color camera.
9. Enable the camera.

Figure 35: Importing a camera parameter file

7.2.6 Connecting to Visionary cameras

Before connecting the Visionary camera to PLB Studio, the camera firmware needs to be updated in SOPAS ET. If using a Visionary-S (PLB510), an additional parameter needs to be adjusted.

1. Start SOPAS ET.
2. Click on **Search devices**.
3. Double-click on the correct camera in the list.

✓ The camera will be added to the project workspace.

4. If incorrect IP address:
   - Click **Edit IP**, then click **Auto** or change the IP address manually. Return with **OK**.
5. Click **Offline** to connect the camera, if not already **Online**.
   - In the pop-up window, select **Read parameters**, then **OK**.
   - Login to the camera (Userlevel: **Service**, Password: CUST_SERV).
6. Click on the options for the device and **Download firmware**.
7. Click **Browse**, and select the correct firmware file (.spk) included in the Visionary software package.
8. Click **Start update**.
9. Login to the camera (Userlevel: **Service**, Password: CUST_SERV).

✓ Download started, pop-up showed when finished.

10. Click **OK**.
11. Click **Close**.

12. Ensure **Update device object in SopasET project** option is enabled, click **Finish**.
✓ Device is updated.
13. If needed, click **Install device driver**, select **Device upload**, and **OK**.
✓ Firmware updated.

For Visionary-S camera (PLB510):
14. Double-click on the device from the project workspace.
15. Go to **Visual settings** > **Stereo settings**.
16. For the setting **Priority RGB/3D**, select **RGB image quality**.
17. Click **Write parameters to device**.
18. Click **Save the parameters permanently**.
✓ The setting is adjusted for the camera.

NOTE
See the SOPAS ET software and SICK Support Portal for advice on how to set up the camera and where to find camera documentation.

In SOPAS ET it is possible to configure the image quality (menu option **Visual settings**, use **Save parameters permanently** for the parameters to be used in the PLB system).

Connect to PLB Studio

In order to use a Visionary camera in PLB Studio, proceed as follows:

1. Open PLB Studio.
2. In the menu bar, click **Configure**.
3. In the parameter editing area, click **System settings**.
4. If there are multiple cameras in the system, select the correct one in the **Camera, Type - Camera alignment** list.
5. Go to **Connection settings**.
6. Enter the IP address of the camera (default: 192.168.1.10) in the **IP address** box.
7. Enable the camera.

7.2.7 Connecting to Zivid camera

When using a Zivid camera you need a camera parameter file. The camera parameter file contains settings to optimize the image quality for specific imaging distances and dimensions. You can use the default parameter file supplied with the PLB system or create a parameter file using the application Zivid Studio (included in the Zivid SDK).

In order to use a Zivid camera in PLB Studio, proceed as follows:

1. Open PLB Studio.
2. In the menu bar, click **Configure**.
3. In the parameter editing area, click **System settings**.
4. If there are multiple cameras in the system, in the **Cameras** section, select the camera in the **Type - Camera alignment** list.
5. Enable the camera.

✓ PLB will search for the camera, and add the serial number and a default parameter file if the camera is available on the network.

Optional:

6. To make further changes to the settings, disable the camera.
7. To import a custom parameter file from Zivid Studio, in the **Camera settings** section, under **Camera parameter file**, click and navigate to the destination folder where the file is located.
   - The default parameter file supplied with the PLB system is located in the folder `C:\ProgramData\SICK\PLB\Default Workspace\System`
   - More information regarding Zivid Studio can be found at support.zivid.com.

8. To use a separate **2D exposure** with the possibility to configure specific **2D exposure** settings, enable **2D exposure**. . Under **2D parameter file**, click and navigate to the destination folder where the Zivid 2D configuration .yml file is located.
9. Enable the camera.

7.2.8 Testing the installation

To test whether the system has been installed correctly, acquire an image using the camera. If there are multiple cameras or camera positions in the system, repeat the last two steps for all camera alignment aliases. Proceed as follows:

1. Click **Start** to open the **Start** menu.
2. Enter **PLB Studio** and then click the PLB Studio search result. (See "Running PLB Studio", page 41 for more information about PLB Studio.)

✓ PLB Studio opens

3. In the menu bar, click **Configure**.
4. In the parameter editing area, click **System settings**.
5. Applicable only if there are multiple cameras or camera positions in the system: In the 3D visualization area, right-click **Trigger camera acquisition** and select a camera alignment alias.
6. In the 3D visualization area, click **Trigger camera acquisition**. The image is displayed in PLB Studio, see figure 36.

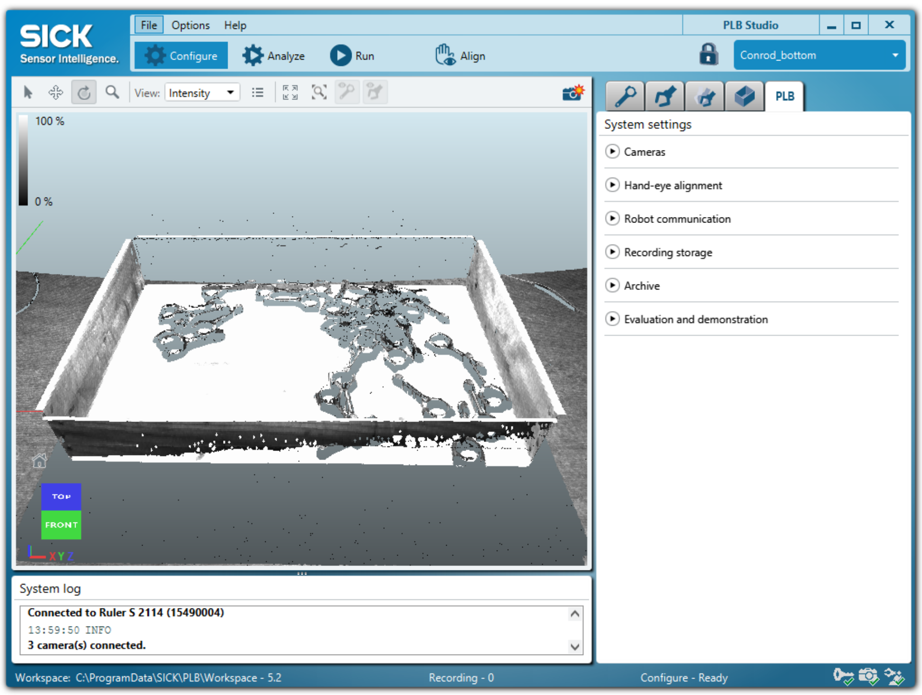


Figure 36: Displaying the result of test image acquisition in PLB Studio

7.2.9 Activating the software license key

The scope of delivery of the camera includes a license key. The license key is supplied printed in the camera box, and on a label on the camera. If you replace the camera, you must also replace the license key.

If the system uses multiple cameras, each camera has its own license key. You must activate the license key for each camera. The slave license keys of the additional cameras are shorter than the master license key of the first camera. Proceed as follows to activate the license key:

1. Click the desktop icon or use the **Start** menu (see Testing the installation) to open PLB Engine and PLB Studio.
2. In the menu bar, click **Configure** .
3. In the parameter editing area, click **System settings** .
4. If there are multiple cameras in the system, select the camera in the **Camera, Type - Camera alignment** list.
5. Make sure that the camera is connected (see also "Camera configuration", page 44). The camera is connected if this icon is displayed in the status bar
6. Go to the **Connection settings** and click (see figure 37).

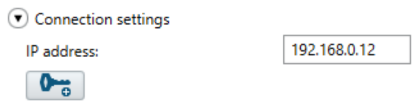


Figure 37: Connection settings

✓ The **License key** dialog box opens.

7. Enter the license key and click **OK**.

7.3 Robot integration

NOTE
We recommend to complete this section before job configuration, but it is mandatory to complete before running jobs.

7.3.1 Robot program

The communication between PLB and the robot controller is based on a command from the controller and a corresponding response message from PLB. A robot program is created to combine the commands to a sequence that can be executed depending on the response messages. Both XML-based and CSV-based formatting are supported.

The commands and messages must be defined in a robot communication file imported to the PLB system. A template file is in default used for the commands and messages. The template files can be copied and adapted to suit requirements, see "Configuring messages and commands", page 150.

NOTE
Robot mounted camera
The robot communication file must include the commands that are specific for robot mounted cameras. For more information see "Mounting concepts", page 15.

If using an upgraded PLB system, create a new temporary workspace folder ("Work-space folder", page 42) for the system to fetch updated template files. The new files will include a robot mounted section of commands.

For all commands and messages, see:

- "Robot commands", page 152
- "PLB response messages", page 161
- State values: "State values", page 167
- Result codes: "Result codes", page 167

NOTICE
Robot mounted camera

All robot mounted commands for triggering an image acquisition must always include the position and rotation of the robot's flange from the default origin of the robot (usually the robot base).

If triggering an image acquisition from PLB Studio, PLB will use the latest position and rotation sent in a command from the robot.

An example of how the robot program of a basic localization can be constructed is presented in figure 38. For more examples, see "Robot program sequence", page 169.

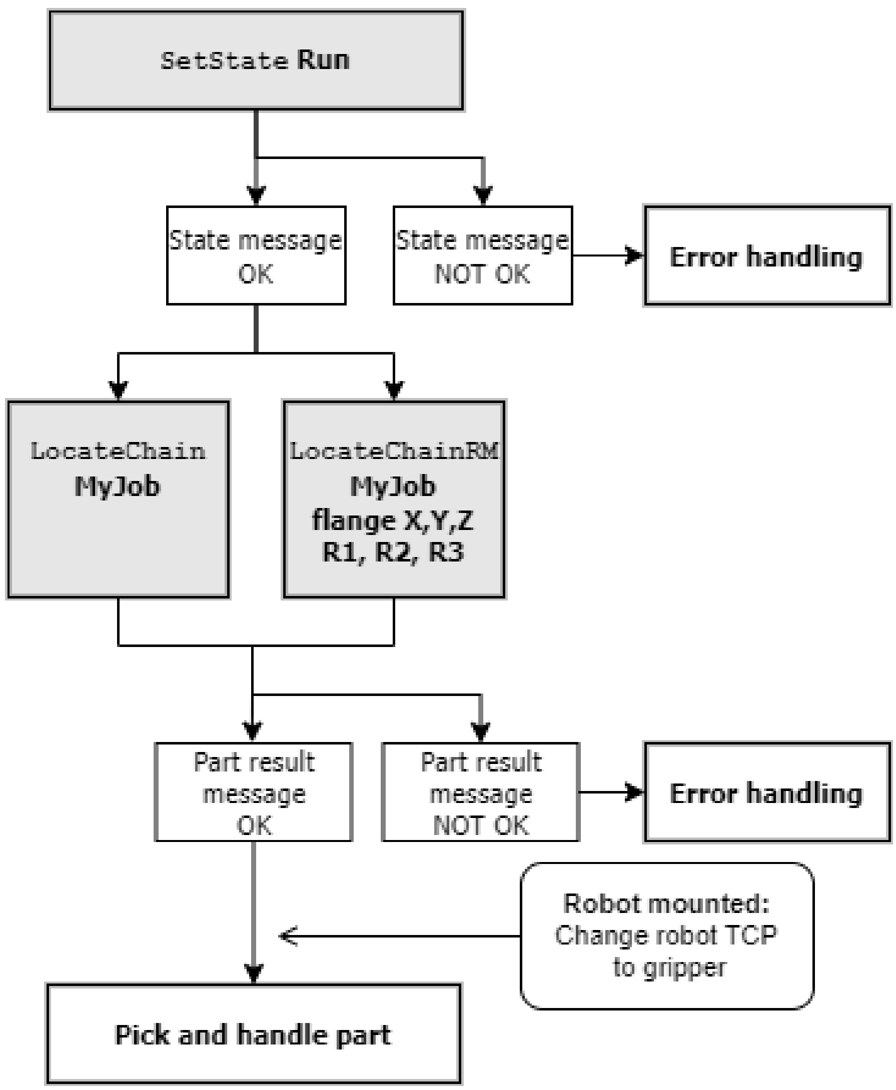


Figure 38: Example of a communication workflow

7.3.2 Setting up communications with the robot controller

To set up communications between the system and the robot controller, you must specify the following in PLB Studio:

- The port number that is to be used by the PLB TCP/IP server
- The template file containing the communication protocol; for detailed information, see "Command interface", page 150.
- The rotation convention; for detailed information, see "Rotation conventions", page 53.

Proceed as follows:

1. Open PLB Studio.
2. In the menu bar, click **Configure**.
3. In the parameter editing area, click **System settings**.
4. Go to the settings for robot communication (see figure 39, page 52).
5. Enter the required port number in the corresponding text box.
6. Expand the Advanced section.
7. Specify the path to the template file and the rotation convention to use.
8. If you want all values in the robot command interface to be rounded to a specific number of fractional figures, select **Use fixed number of decimals** and enter **Number of decimals.**

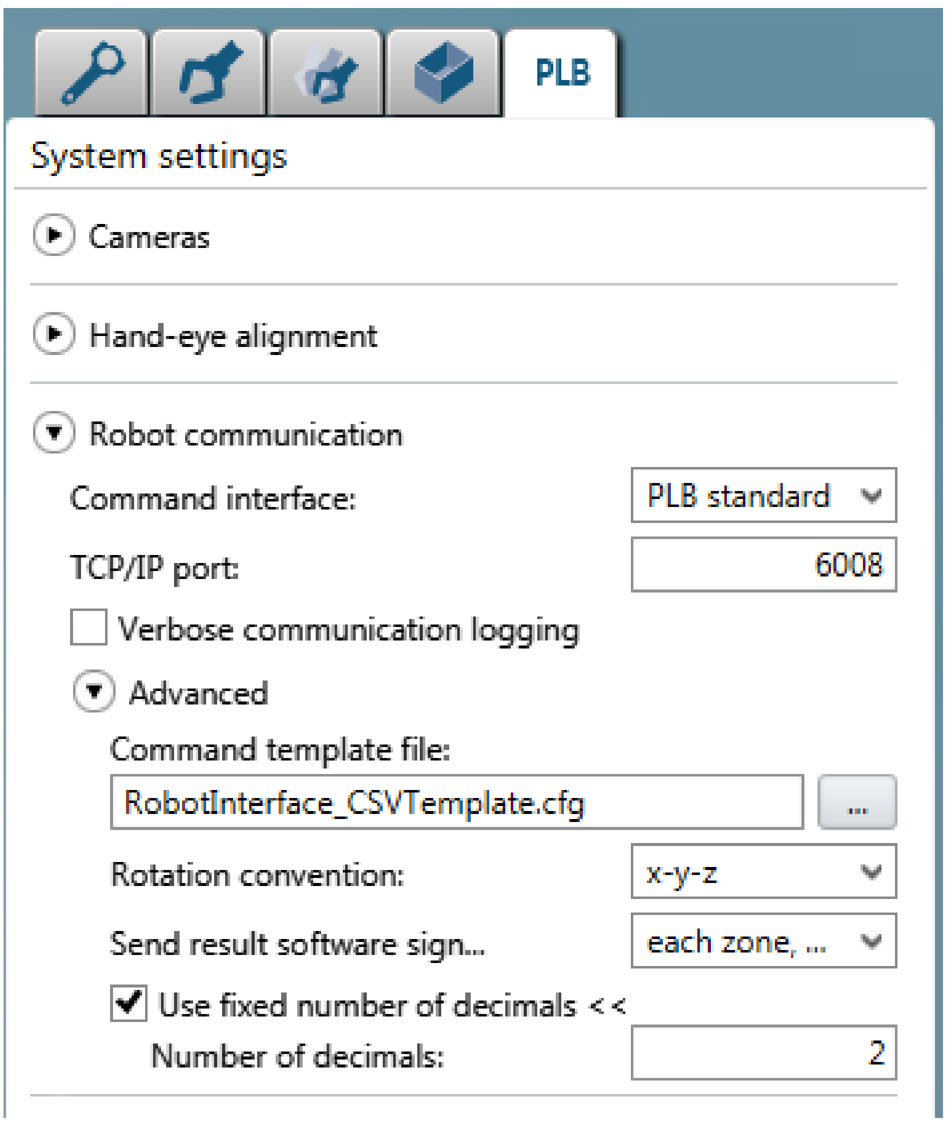


Figure 39: Specifying the TCP/IP port number, the path to the template file, and the rotation convention

7.3.3 Rotation conventions

Angle rotation in robot interface responses

PLB supports output of all proper Euler and Tait-Bryan angle rotation orders of the X, Y and Z angle rotations, and also both intrinsic (local) and extrinsic (global) rotation conventions. PLB supports 24 angle rotation variants in total.

PLB naming convention examples

The naming convention x-y-z is used for extrinsic rotations around the global (static) X, Y and Z axes in said order. The naming convention z-y'-z'' is used for intrinsic rotations around the local (moving) Z, Y and Z axes in said order.

Configuration in PLB GUI

Rotation convention is set in Configure mode in the robot communication section in the PLB tab. Setting is available when using the PLB standard robot interface protocol, but other robot interface protocols can have fixed rotation convention. Setting will not be available in these cases.

7.3.4 Hand-eye alignment

7.3.4.1 Overview hand-eye alignment

In order to be able to locate parts accurately and determine pick poses, the measurement values output by the camera must be transformed from the camera frame into the robot frame. We refer to this process as hand-eye alignment.

When referring to a frame, we mean both the coordinates and the rotation within that frame. For example, the robot frame is defined as x, y, z and R1, R2, R3 (default Euler/Tait-Bryan angles) from the origin (usually the robot base).

For information on the hand-eye alignment process depending on mounting concept, see "Stationary camera", page 53 or "Robot mounted camera", page 55.

The alignment target

For the hand-eye alignment process an alignment target is used. The target should be shaped like a cone. If the image data quality is low, then it is preferable to place an additional sphere with a conical hole on the cone.

The alignment target can be purchased as an accessory on **www.sick.com/PLB**. You can also create the alignment target yourself. Dimensional drawings of the alignment targets can be downloaded from **SICK Support Portal**.

The alignment cone should have an apex angle of approximately 84° and no more than 90°. It should have a diameter of between 80 and 120 mm (larger diameters improve accuracy). The additional alignment sphere should have a diameter of 150 mm with a conical hole that fits the alignment cone.

The surface of the target should be matte, non-reflective and gray. Avoid very dark or shiny surfaces.

7.3.4.2 Stationary camera

The alignment target is mounted to the robot arm, see figure 40. Align the robot arm in such a way that the target's apex points upward.

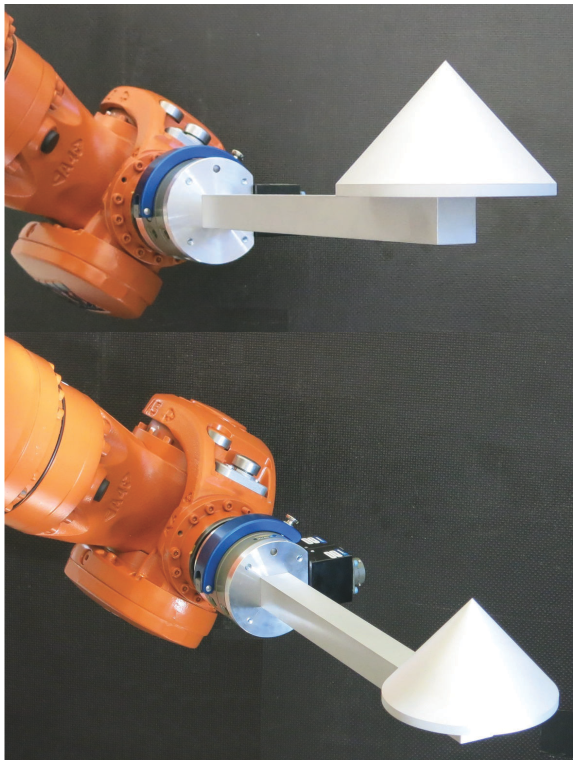


Figure 40: Alignment target fixed for stationary camera, alignment cone on the robot arm

Procedure

A robot tool frame is defined with its TCP on the top of the cone, regardless if an additional sphere is used or not. The target is placed in the camera field-of-view and an image is acquired. PLB locates the target in the camera image. The information about the location of the target in the camera frame and in the robot frame (supplied by the robot) is saved as an alignment point. PLB can calculate the transform between the two frames when this has been repeated for several positions in the camera field-of-view.

Choose 15[1)] 3D measurement points that are evenly distributed within the bin volume. You can, for example, choose one 3D measurement point in each of the corners and one centered between them at three different depths of the bin. It is recommended to position the target with a varying amount of tilt throughout the alignment process (see figure 41).

To complete the alignment, repeat one measurement point. The alignment target should be located at the same point in the volume, but the robot needs to be repositioned, see figure 42. The second robot position must be rotated as far as possible around the center point of the alignment target, recommended 180 degrees rotation from its first position.

1) Recommended number = 15, minimum number = 4.

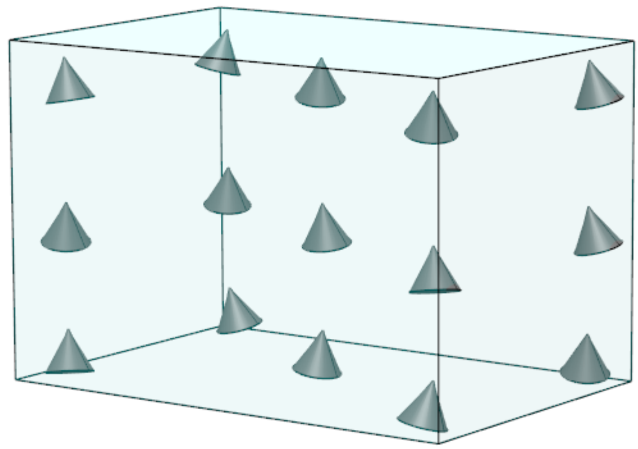


Figure 41: Selecting 3D measurement points in the bin volume

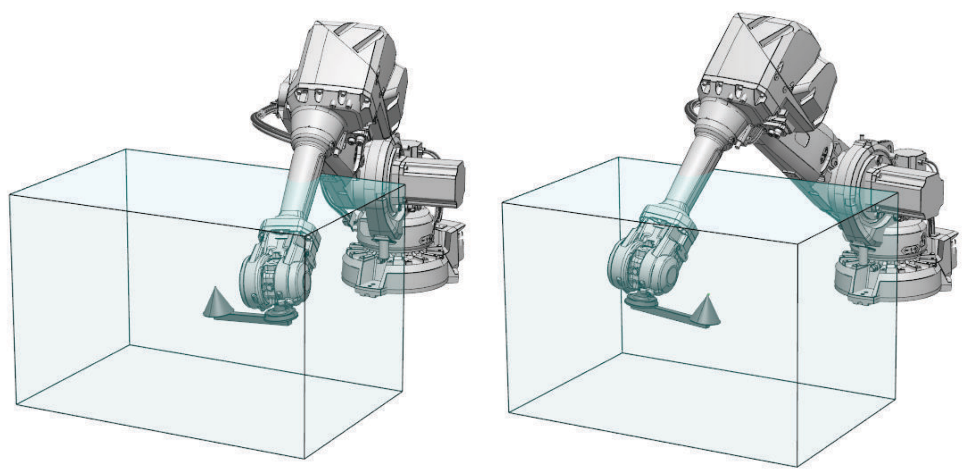


Figure 42: Repeated measurement point with robot rotation

7.3.4.3 Robot mounted camera

The alignment target is mounted within the robot cell, see figure 43. The target must remain in the same place during the hand-eye alignment process. The robot arm with the camera must be able to approach and view the target with a variation of position, distance, and rotation.

NOTE

The target must remain fixed in the same place throughout the alignment process.

It is recommended to keep the alignment target indefinitely mounted in the same place in the robot cell, for easier verification of the alignment at a later stage.

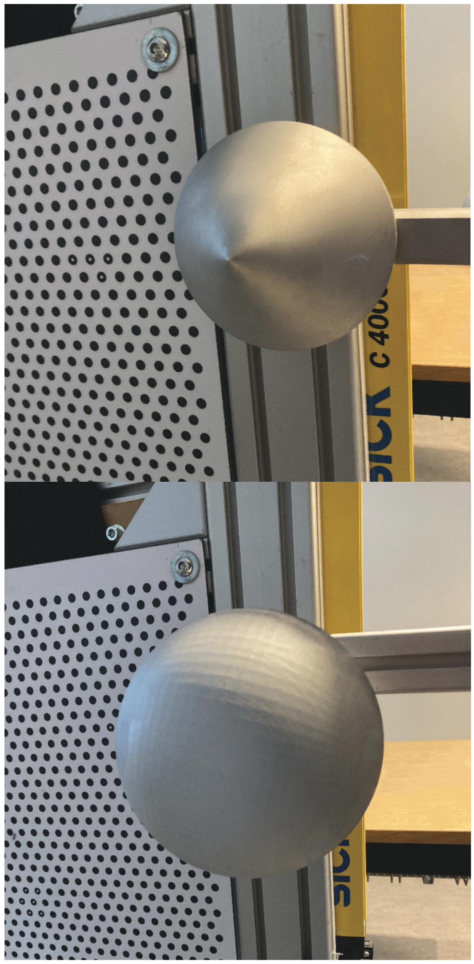


Figure 43: Alignment target fixed for robot mounted camera, top: alignment cone, bottom: alignment sphere

Procedure

When aligning a robot mounted camera, the alignment target is in a fixed position. The robot arm with the camera is moved within the intended working volume. The goal is to achieve a transform between the camera frame and a known robot point's frame (the flange on the robot arm).

NOTICE
Always use the flange's position and rotation from the default origin of the robot (usually the robot base) when running the PLB system.

The robot provides the flange's position in the robot frame when a new image is acquired. The alignment target is located by PLB in the acquired image, and the target's position is saved in the camera frame. The positions in the robot frame and the camera frame are saved as an alignment point in the system. By acquiring several alignment points, a transform is calculated by PLB between the camera and robot frames.

Instructions

Use 15 different alignment points, defined by manually jogging the robot with the camera to new positions within the entire intended working volume.

NOTE
To get a better overview of the alignment in PLB Studio, position the robot for the last alignment point to be centered and at the maximum distance of the intended working area.

The alignment points must achieve enough variation in the robot frame (x, y, and z-axis, and their rotations). Also, the target must be visible in a variation of places within the intended working area in the camera's field-of-view.

NOTE
If using a robot without rotation around all axes, see "Robot without all-axes rotation", page 59 for more details.

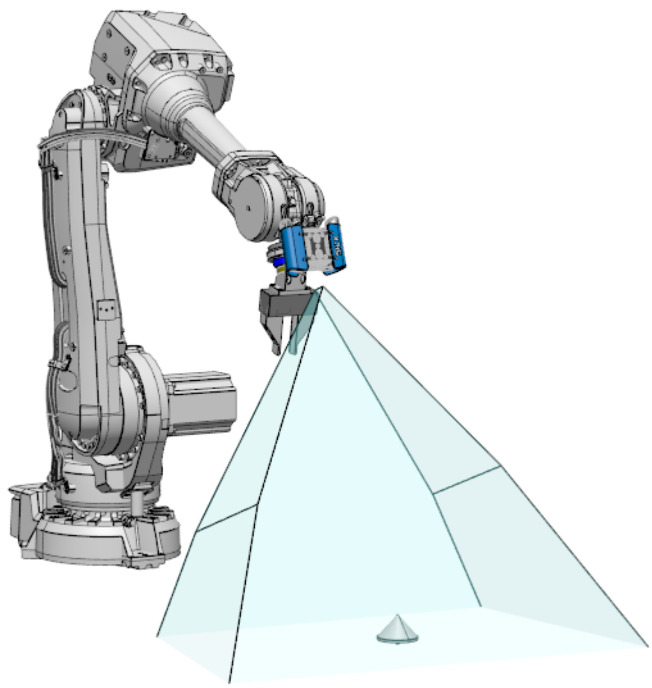


Figure 44: Robot mounted alignment point example

For all alignment points, alter:

- Position on the z-axis to the alignment target (figure 45, repositioned from figure 44)
- Position on the x-axis and y-axis to the alignment target
- Rotation on the z-axis to the alignment target (figure 46, rotated from figure 44)
- Rotation on the x-axis and y-axis (tilt angles) to the alignment target (figure 47)

It is important that the alignment points are of a sufficent variation in position, rotation, and tilt for the robot, as well as different placements in the camera's field-of-view. The PLB system will send an error if the points are not sufficiently varied for the hand-eye alignment to be accurate, see the **System log** in PLB Studio and "Result codes", page 167.

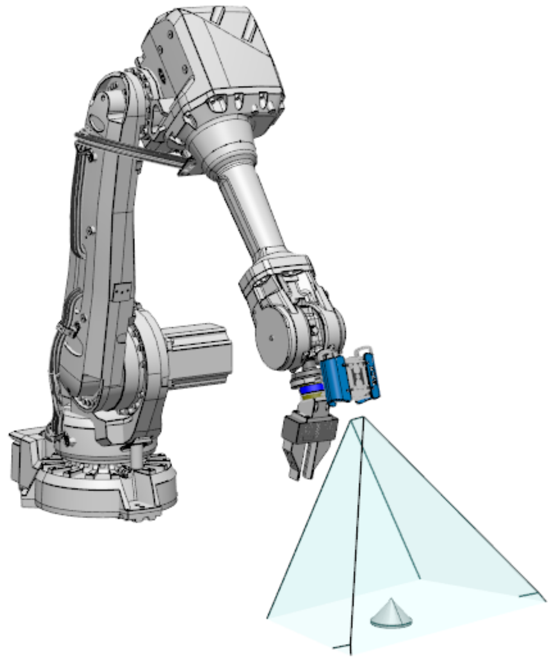


Figure 45: Robot mounted alignment, z-axis position

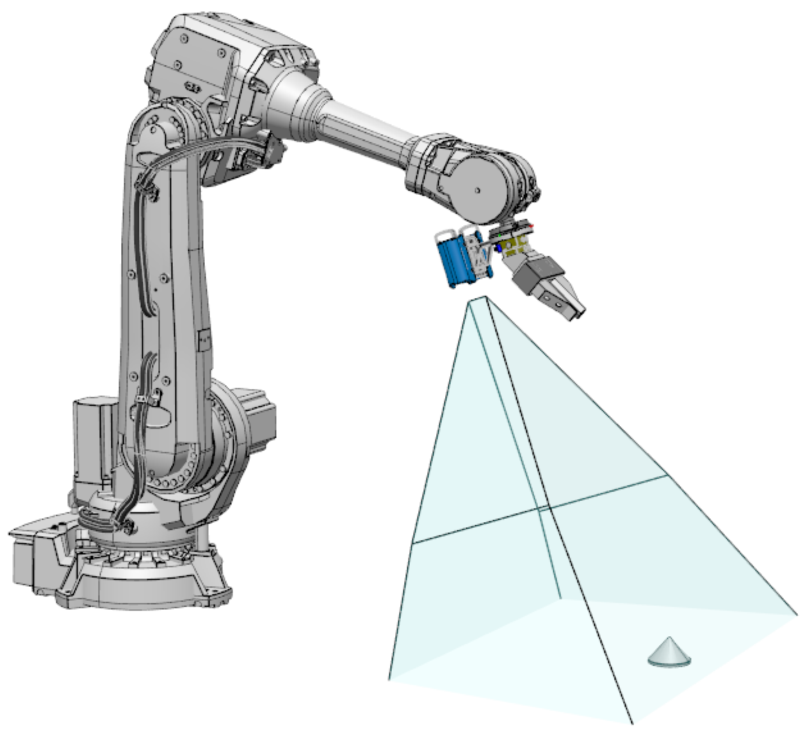


Figure 46: Robot mounted alignment, z-axis rotation

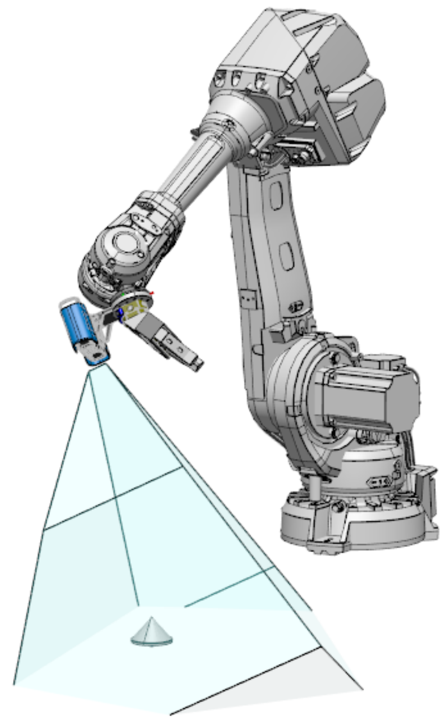


Figure 47: Robot mounted alignment, x-axis and y-axis rotation

Robot without all-axes rotation

The hand-eye alignment procedure usually requires rotation on all axes. If this is not possible because of robot limitations, then an additional step is required after the alignment points are defined.

1. Mount and configure a pointer tool as a TCP on the robot.
2. Jog the robot so the TCP is on the top of the alignment target cone.
3. Send a `AddAlignmentTargetPositionRM` command with the robot TCP position.

7.3.4.4 Configuring hand-eye alignment

Hand-eye alignment configuration is used to:

- Define the target dimensions; diameter and apex angle.
- Set the threshold for maximum error allowed in the hand-eye alignment.

NOTE
All actions regarding hand-eye alignment are performed via the robot communication interface, including switching to alignment mode.

To configure hand-eye alignment:

1. Open PLB Studio.
2. In the menu bar, click **Configure** .
3. In the parameter editing area, click **System settings** PLB.
4. Go to the settings for **Hand-eye alignment** (see figure 48, page 60).
5. Select **Cone** or **Sphere** in the **Alignment target type** list.

6. Specify the diameter and top angle for an alignment cone, or the diameter for an alignment sphere.

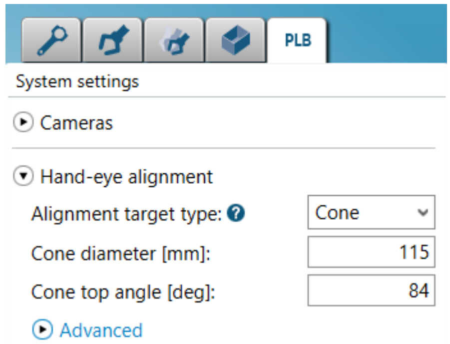


Figure 48: Specifying the target type and specifications.

You will find additional information about hand-eye alignment in the following sections:

- Commands and messages are described in "Robot commands", page 152 and "PLB response messages", page 161 respectively.
- For information about robot program sequences, see "Robot program sequence", page 169.
- For a description of parameters, see "Hand-eye alignment", page 133.

Example of a successful alignment

See figure 49, for the result of a successful alignment.

Note that:

- The result code sent to the PLB controller is 0. For a list of result codes, see "Result codes", page 167. Result codes related to hand-eye alignment starts at code 131.
- **Max deviation** is less than 2 mm.
- **Scale deviation** is in the range -2...2 [%].

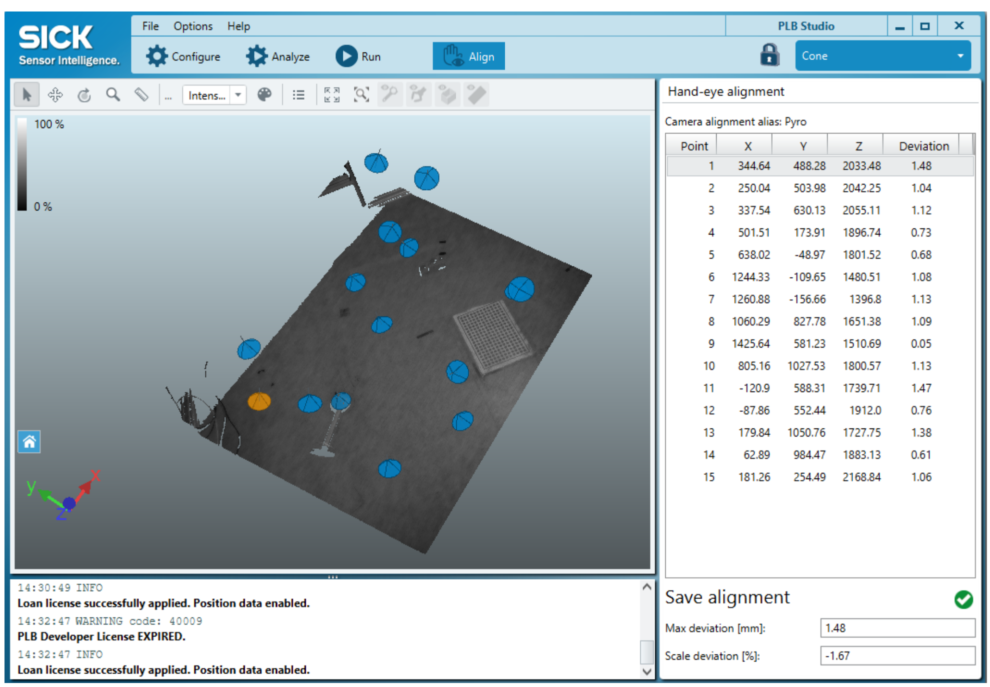


Figure 49: Example of a successful alignment for a robot mounted camera

7.4 Job configuration

7.4.1 Requirements and workflow

NOTE
Please note that "Underlying concepts", page 113 provides information on the various concepts that are essential for understanding system configuration. We strongly recommend to familiarize yourself with these concepts before you configure your first job.

Job configuration is used to

- Specify the parts to be located
- Specify the pick poses so that the system is able to determine whether a located part can be picked up without collision.

Requirements

We recommend to have completed the robot integration (including hand-eye alignment) before proceeding with job configuration. It is although required to make an accurate job configuration of **Bin and search volume** (see "Robot integration", page 50).

To configure a job using the Surface algorithm, you need solid CAD models (not surface models) of the parts and the grippers. We recommend using the STEP format. The IGES format is also supported.

The Edge and plug-in algorithms do not use CAD models of the parts, but CAD models of the grippers are still needed.

NOTE
Please note that you can speed up the calculation by reducing the size of the gripper CAD file. Since only the volume of the gripper is required for collision detection, you can remove details such as screws or holes from the model.

Job configuration step by step

Job configuration comprises the following steps (see figure 135):

1. Configuring the part
2. Configuring the gripper and adapting the gripper coordinate system to the robot's TCP
3. Defining the potential pick poses for the part
4. Defining the bin
5. Testing and modifying the parameters
6. Locking the job

7.4.2 Creating a new job

Proceed as follows to create a new job (see figure 50).

1. Open PLB Studio via the taskbar (see figure 28).
2. In the menu bar, click **Configure**.
3. Click the job drop-down list to manage jobs.
4. In the bottom line, click **Create new job** to add a new job.
5. Optional: Enter a name for the job and press **Enter**.
6. Optional: To use the edge algorithm, select **Edge** as the algorithm in the **Parts** section of the parameter editing area. To use a plug-in algorithm, select the name of that algorithm.

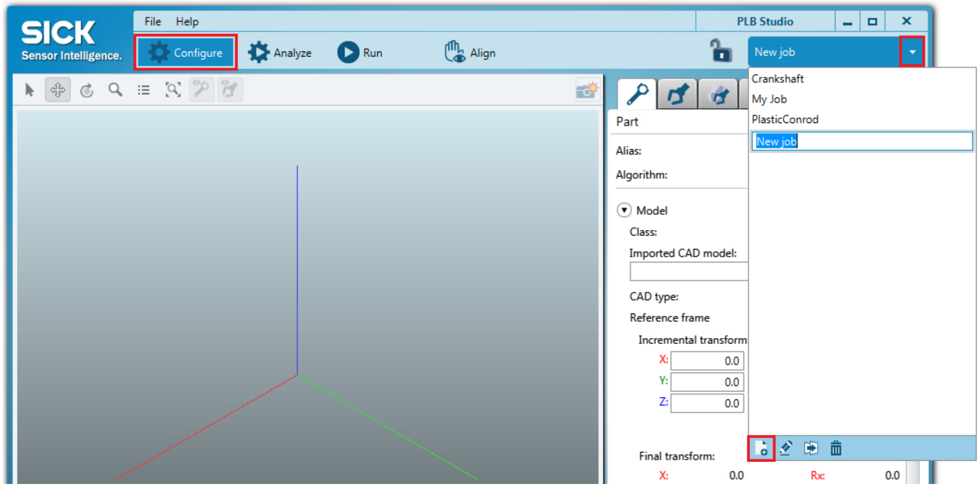


Figure 50: PLB Studio, creating a new job

7.4.3 Configuring parts

7.4.3.1 Configuring parts for the Surface algorithm

You will find detailed information on the concept of parts classes in "General concepts", page 114.

After you have created a new job, proceed as follows:

1. Select the required job from the drop-down list at the top right.
2. In the parameter editing area, click **Part** to edit the part configuration.
3. Use the drop-down list to assign the part to a class (see figure 51, page 62).
4. Enter the path of the corresponding CAD model (see figure 51, page 62).

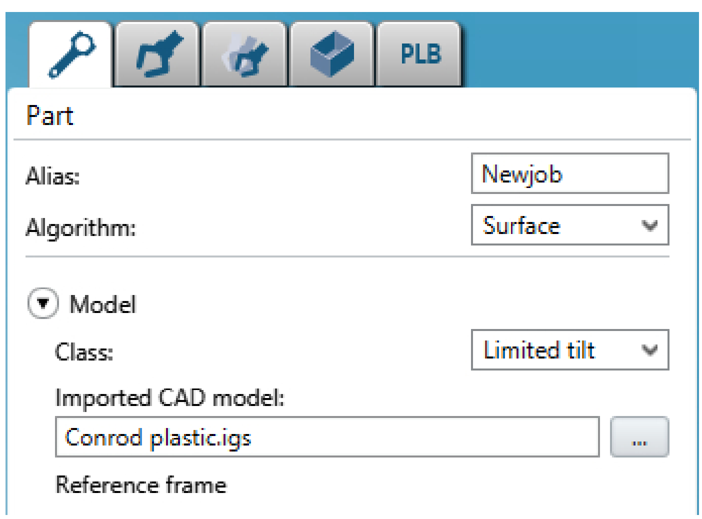


Figure 51: Configuring a new part

5. Adapt the part's reference frame by entering the required values under **Incremental transform** (see figure 52, page 63):
   - Select the **Local transform** checkbox to make the modifications in the part's local coordinate system.
   - Click the + and − buttons to modify the final transform for all axes.
   - Click the plus and minus signs next to each text box to modify the final transform for one axis at a time.

The final transform (in global coordinates) is described by the part's translation (X, Y, Z) in mm and its rotation (Rx, Ry, Rz) in degrees. Click 0 to set all the values to zero.
The display in the visualization area is updated accordingly (see figure 53, page 63).

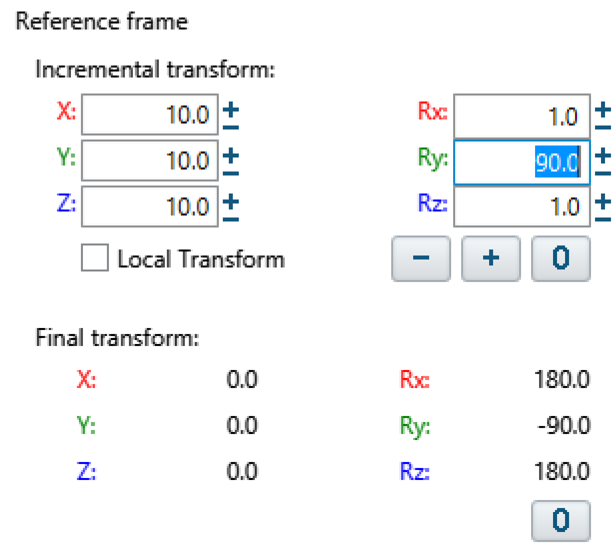


Figure 52: Adapting the part's coordinate system

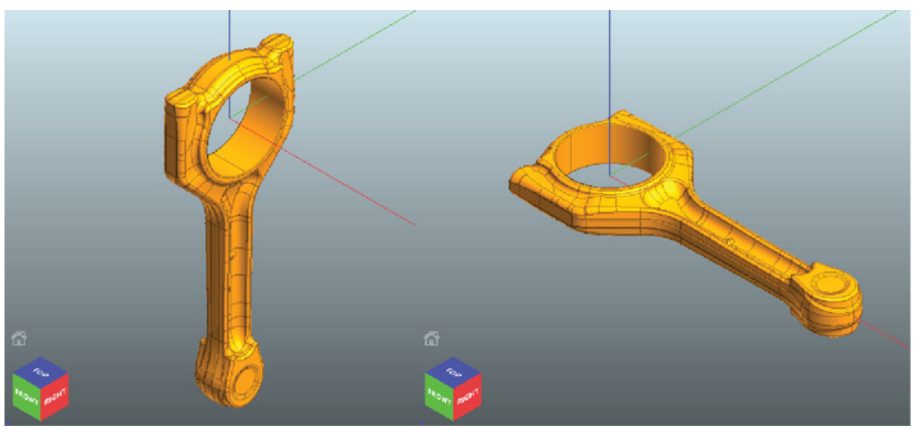


Figure 53: Original and adapted part reference frame

6. Detailed information on the other parameters available to you for parts configuration can be found in "Parameters for job configuration", page 135.
7. During part configuration, it is also essential to specify any symmetries or pseudo-symmetries that the part may possess.
   Symmetries and pseudo-symmetries are only taken into account during part localization, not during the calculation of the pick poses. "Symmetry and pseudo-symmetry", page 125 contains more detailed information on the concepts of symmetry and pseudo-symmetry.
   Any symmetries that are present must be taken into account during the alignment of the part coordinate system. You must subsequently adapt the corresponding configuration settings for the part.
   This does not apply to:
   - Parts that have been assigned to the "Revolving" class. The parameter "Symmetry around X-axis" is automatically set for such parts. However, if the part also possesses other symmetries then you must add these later.

In the case of pseudo-symmetrical parts, you should mark the areas where there is a deviation by defining an inclusion or exclusion (see "Inclusion and exclusion boxes", page 126).
For detailed information on the other parameters available to you when specifying symmetries or pseudo-symmetries as well as when configuring inclusion and exclusion boxes, see "Geometry", page 136 and "Overlap", page 139.

8. If you do not want to use the whole CAD model as a reference during the localization of the part, select the **Partial model** checkbox, and define inclusion and exclusion boxes. Boxes defined as inclusions are taken into account when the part is localized and boxes defined as exclusions are not.

7.4.3.2 Configuring parts for the edge algorithm

After you have created a new job, proceed as follows:

1. Select the required job from the drop-down list at the top right.
2. In the parameter editing area, click **Part** .
3. Put the part in the camera's field of view. The part should be placed separately in its default tilt.
4. Click **Trigger reference image acquisition** to acquire a reference image.
5. In the parameter editing area, use the **Z plane** slider to adjust the Z-plane level so that the plane is below the part.
6. Optional. Use the teach rotation increment controls to rotate the point cloud until the part is parallel with the Z-plane. This is useful if the camera is facing the scene at an angle.
7. Select the top view in the 3D navigation control in the lower left part of the visualization area.
8. Click **Select** to access the region marking toolbox to mark interesting part features. Select the shape of your marker (square or circular) and adjust the size using the slider. If needed, use the eraser to revise your markings or use the trash can to erase all markings, see figure 54.

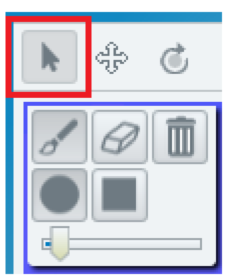


Figure 54: The region marking toolbox

Marked regions are displayed in green on the blue Z-plane, see figure 55. All volume above the green region marking is included in the region of interest used to teach the edge algorithm.

9. Use the region marker to mark relevant edges and flat surfaces on the part. Detected edges and surfaces are visualized with indicators, as shown in figure 55. Small circles indicate detected edges, while large circles with a line indicate detected flat surfaces.
   - Edges are necessary to detect the part's X and Y position as well as rotation. There should only be edge indicators on the actual part edges, not in the middle of a plane.
   - Flat surfaces are necessary to detect the part's Z position and tilt. There should only be surface indicators on the part, not on the underlying surfaces.

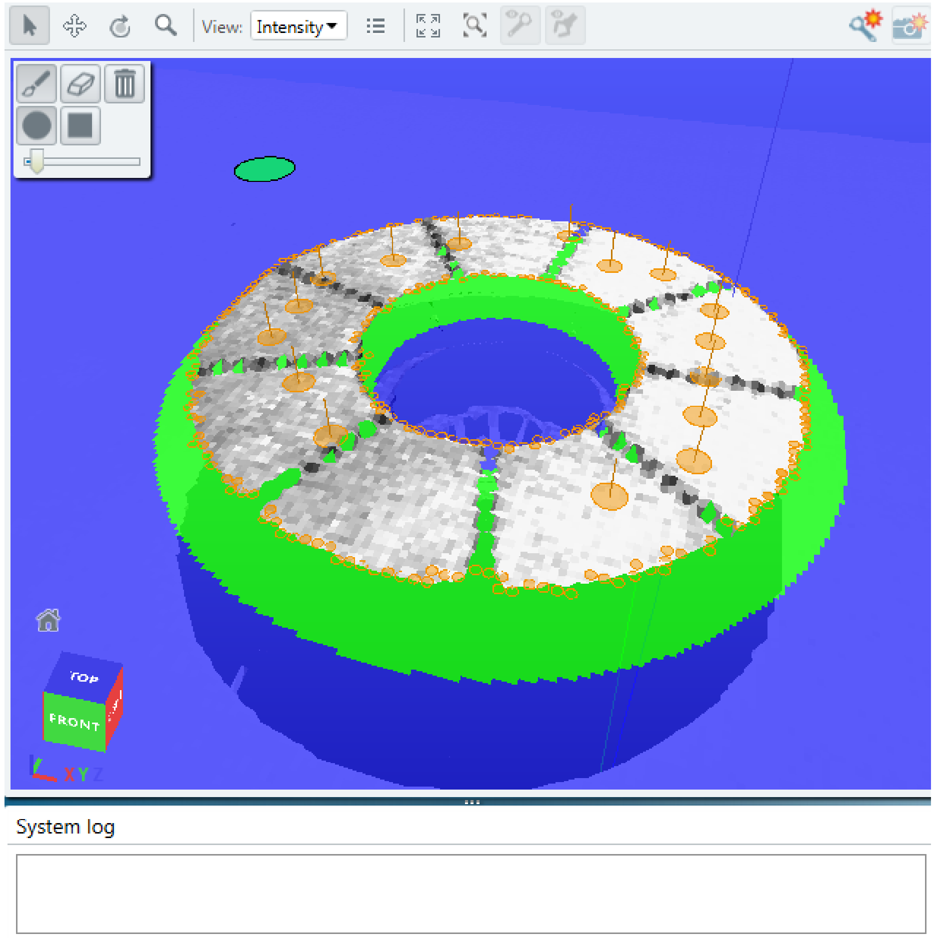


Figure 55: Marking an object

10. If needed, readjust the height of the Z-plane. A too low Z-plane may result in surface and edge indicators on the bin bottom, as shown in figure 56.

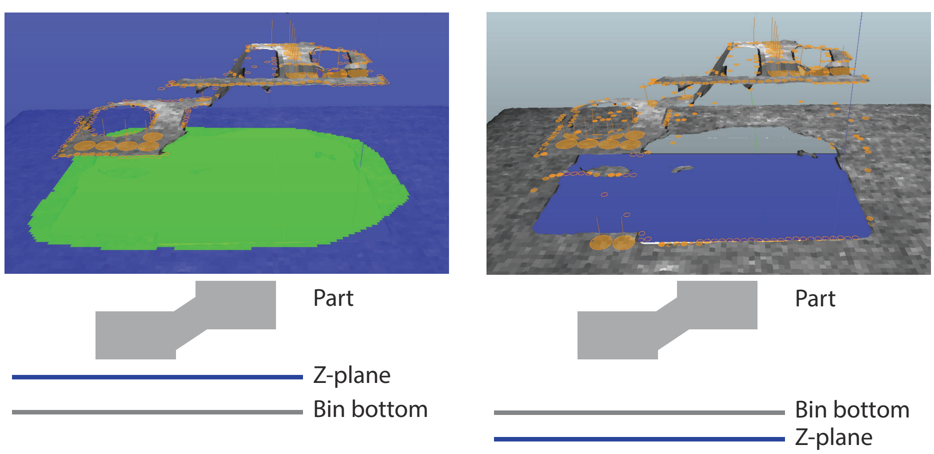


Figure 56: Left: Good teach. Right: Surface and edge indicators on the underlying surface.

11. If needed, adjust the **Edge threshold** to exclude edge indicators located on the part surfaces. Any height differences in the image above the threshold will be detected as an edge.
12. Adjust the **Maximum number of edges.** Too many edge indicators will increase the calculation time in run mode. A large part requires a higher threshold value than a small part. It is recommended to start with the default value and increase the threshold for robustness or decrease the threshold for speed.

13. Repeat steps 9. to 12. until the result is satisfactory.

NOTE
If the object is not perfectly symmetric, two separate edge jobs are needed to be able to locate the object when turned on either side, see figure 57.

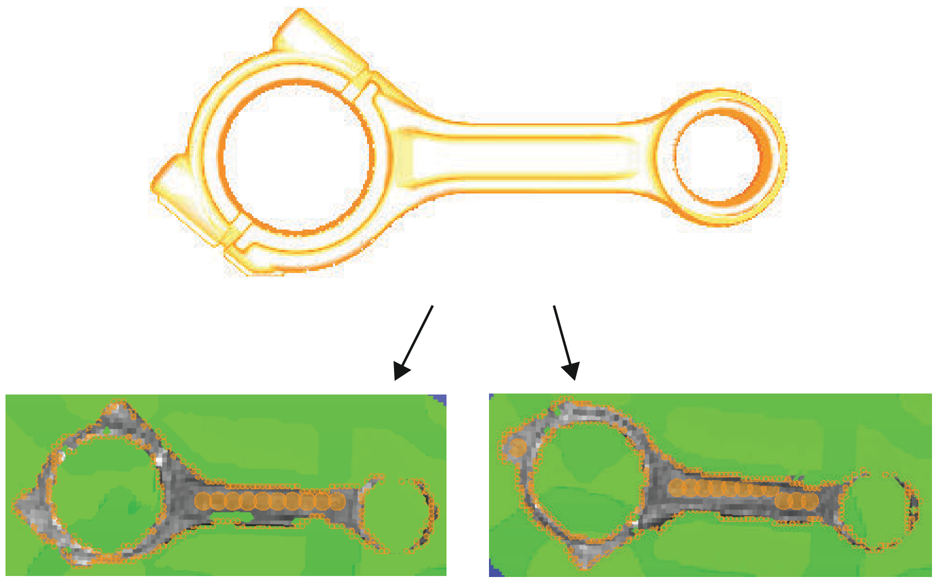


Figure 57: Two separate edge jobs are needed for non-symmetric parts

Lock part reference frame

Part reference frames are defined relative to a reference point. The reference point is at the center of the current shape of edge indicators, at the height of the closest surface indicator.

Pick poses are defined relative this reference point, and can become incorrect if the region marking is changed or if a new reference image is acquired.

- Update region marking: Enable **Lock part reference frame** to prevent defined pick poses from moving if the shape of the edge indicators is changed.
- Update reference image: Disable **Lock part reference frame** and configure the algorithm again, see "Configuring parts for the edge algorithm", page 64, and then continue with the configuration.

7.4.3.3 Configuring parts for plug-in algorithms

The following part settings are available in the **Localization** section in the **Part** tab in PLB studio:

- **Score threshold:** The plug-in algorithm will only return results with a score above this threshold. The default value is 70.
- **Downsampling distance:** PLB sends an image to the plug-in algorithm. The distance between points of the point cloud, or the distance between height map samples in this image can be changed. Downsampling distance is disabled by default.
- **Blacklist:** see "Blacklist", page 115.

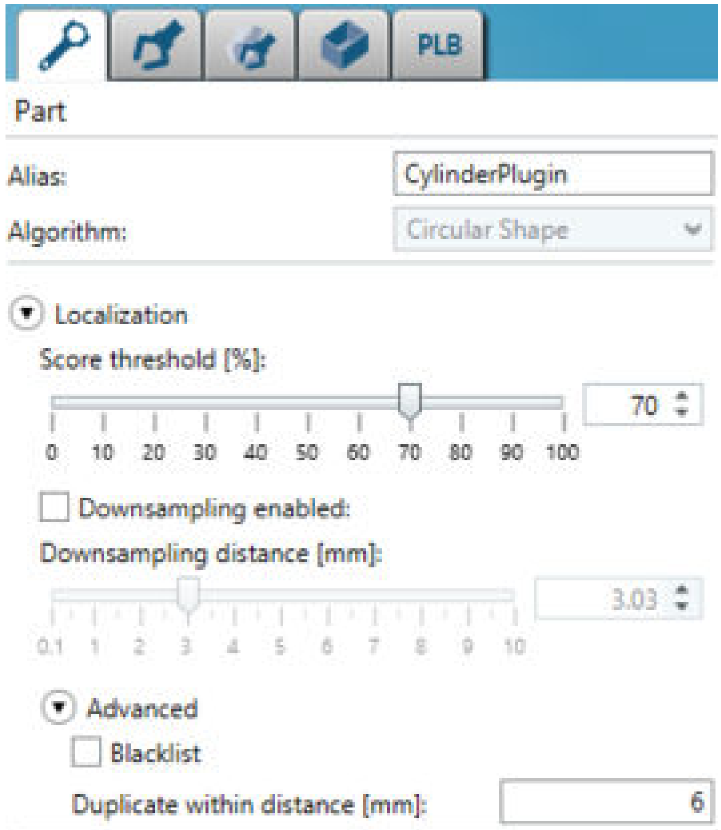


Figure 58: Localization settings for plug-in algorithms.

The plug-in specific settings are available in a separate section in the **Part** tab:

- Click **?** (Help) to get more information about specific plug-in settings. The button is not available for all plug-in algorithms.

Note that **Result sort order** and **Limit pick rotation** are configured in the **Bin and search volume** tab, see "Configuring result sort order for edge and plug-in algorithms ", page 77 and "Configuring the reference point for limited pick rotation", page 79.

7.4.4 Configuring a gripper

In "Gripper design for bin picking applications", page 180 you will find useful information on gripper design.

When you have configured the part, you must configure one or more grippers. To do this, proceed as follows:

1. Select the required job from the job drop-down list.
2. In the parameter editing area, click **Gripper** to edit the gripper configuration.
3. Click **Create new gripper** and enter a name for the gripper (see figure 59).
4. Choose **Model** > **Imported CAD model** to import the corresponding CAD model.
5. Adapt the gripper reference frame to the robot's TCP by entering the required values under **Incremental transform** (see figure 60):
    - Select the **Local transform** checkbox to make the modifications in the gripper’s local coordinate system.
    - Click the + and - buttons to modify the final transform for all axes.
    - Click the plus and minus signs next to each text box to modify the final transform for one axis at a time.

- Recommended settings:
  - Place the origin of the robot's gripper TCP centered in the gripping position, since all gripper related features in PLB are based on the gripper TCP.
  - Place the gripper facing positive Z, since this is normally the robot direction outward from the robot flange (see figure 61).

The final transform (in global coordinates) is described by the gripper's translation (X, Y, Z) in mm and its rotation (Rx, Ry Rz) in degrees. Click 0 to set all the values to zero.

The display in the visualization area is updated accordingly (see figure 61).

6. You can find further information and parameters for gripper configuration in "Gripper", page 142.

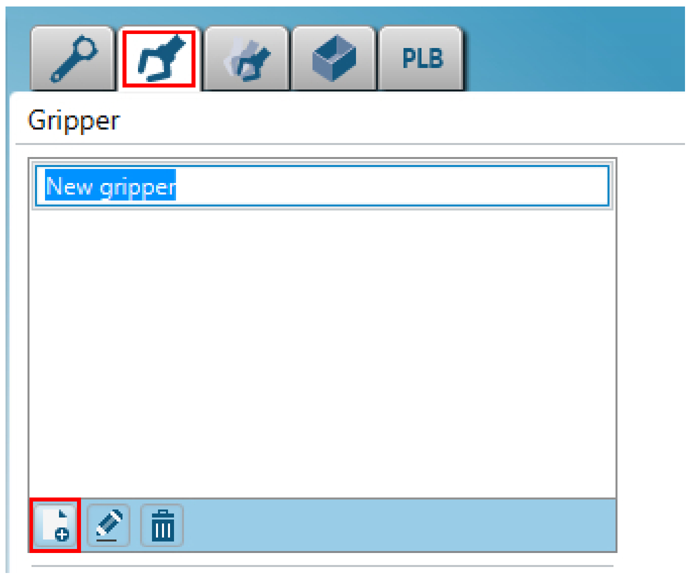


Figure 59: Configuring a new gripper

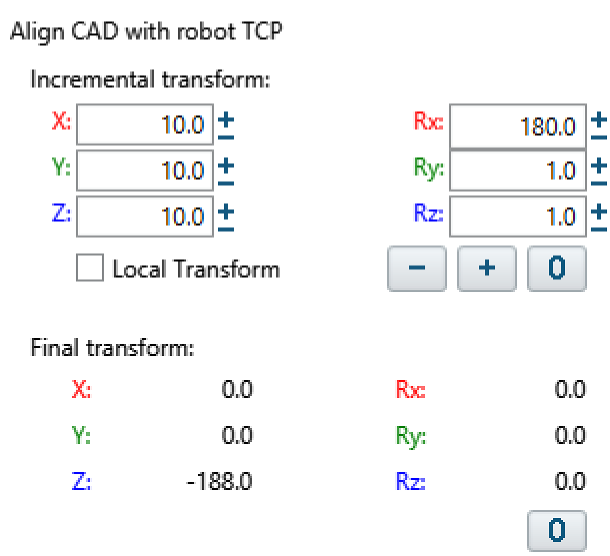


Figure 60: Adapting the gripper's reference frame to the TCP

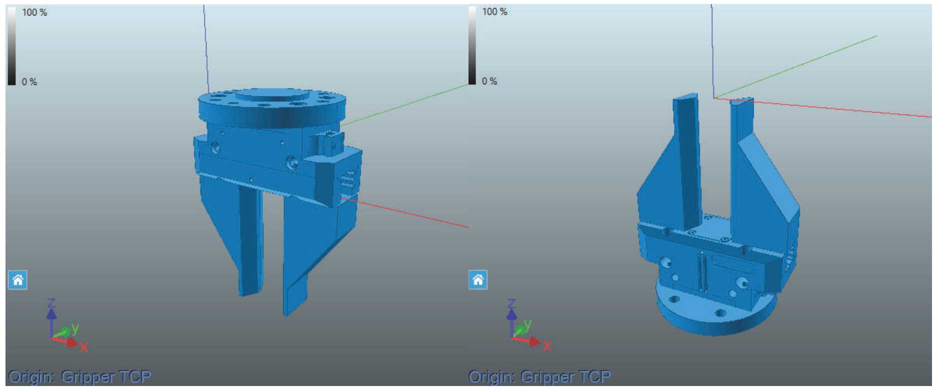


Figure 61: Original (left) and adapted gripper (right) reference frame

7.4.5 Configuring pick poses

When you have configured the part and the gripper, you can define the corresponding pick poses. The pick poses are used for collision detection and to get pick coordinates to the robot. You will find additional information on determining the pick pose in "Vertical correction", page 124.

You can define up to 128 pick poses for a part. You can also use different grippers for different pick poses on the same part (see figure 62). The system can output up to eight pick poses for each located part.

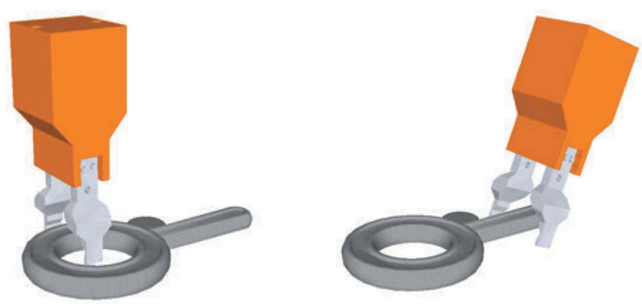


Figure 62: Example of two different pick poses for the same part

Take account of the following points when determining the pick pose:

- If you want to perform vertical correction of the pick pose then the X-axis must be the axis of rotation. (For detailed information on vertical correction, see "Vertical correction", page 124.)
- So-called bounding boxes are defined around the part and gripper. Make sure that the Z-coordinate of the midpoint of the box surrounding the gripper is higher than the midpoint of the box surrounding the part.
- N-fold rotational symmetries are not taken into account. This means, for example, that for a 6-fold rotational symmetry, you need to configure six different pick poses even though the appearance and shape of the part remain strictly identical each time it is rotated through 60° (see figure 64 and figure 64).
  (For detailed information on rotational symmetries, see "Symmetry and pseudo-symmetry", page 125)

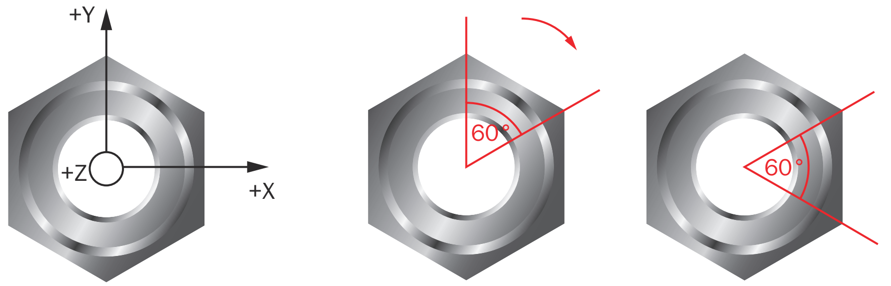


Figure 63: Part with n-fold rotational symmetry (here: 6-fold), coordinate system, and 60° rotation around the Z-axis

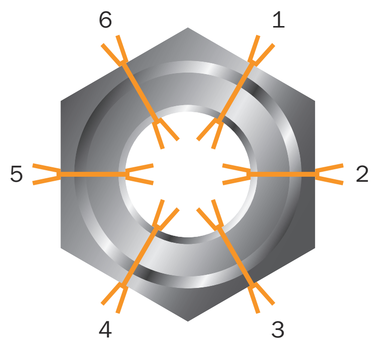


Figure 64: Part with n-fold rotational symmetry (here: 6-fold), pick poses to be configured

Take account of the following points when configuring the pick pose for plug-in jobs:

- For plug-in jobs, there is no visualization of the part when configuring pick poses.
- The origin of the coordinate system in the pick pose view is the part locate result. Pick poses are defined relative to this origin.

WARNING
For the Edge algorithm, do not define pick poses below the lowest surface of the part.

- Pick poses below the part could evade collision tests (especially a pick pose that is straight down from the middle of the part where we are more likely to have missing data). See figure 65.
- Since the part will never be detected upside down (limited tilt of 15 degrees for the Edge algorithm), setting a pick pose below the part would send the robot's gripper below the part, which could result in a dangerous collision.

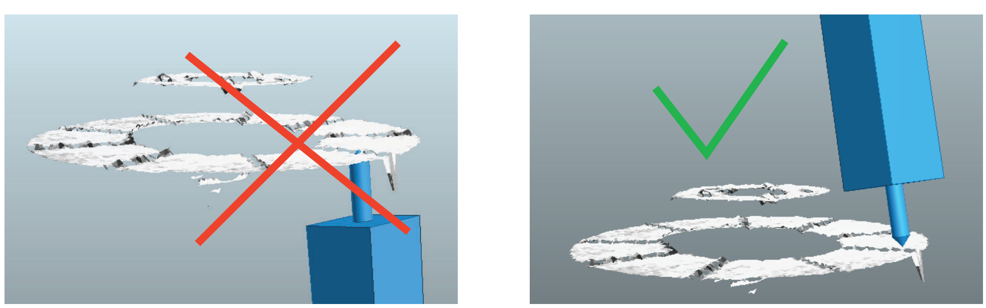


Figure 65: For Edge jobs, use pick poses where the gripper is above the part, as shown in the right image

There are two ways of configuring the pick pose:

- using the PLB system
- using the robot controller

The instructions for the respective ways of configuring the pick pose are presented in the sections below.

Configure the pick pose using the PLB system

1. Select the required job from the job drop-down list.
2. In the parameter editing area, click **Pick poses** .
3. Click **Create new pick pose** and assign a name for the new pick pose, then choose one of the previously configured grippers from the drop-down list (see figure 66).
4. Adapt the pick pose as required under **Incremental transform:**
   - Select the **Local transform** checkbox to make the modifications in the pick pose's local coordinate system.
   - Click the and buttons to modify the final transform for all axes.
   - Click the plus and minus signs next to each text box to modify the final transform for one axis at a time.

   The final transform (in global coordinates) is described by the pick pose's translation (X, Y, Z) in mm and its rotation (Rx, Ry Rz) in degrees. Click to set all the values to zero.
   The display in the visualization area is updated accordingly (see figure 67).
   A distance of approximately 3 mm should be observed between the part and the gripper. Otherwise, there is a risk that measurement noise will cause collisions to be detected.
5. Repeat the above steps for each desired pick pose.
   - To create a new pick pose with the same configuration as the selected pick pose, click **Clone selected pick pose** .
   - Detailed information on the other parameters available to you for pick pose configuration can be found in "Pick poses", page 143.

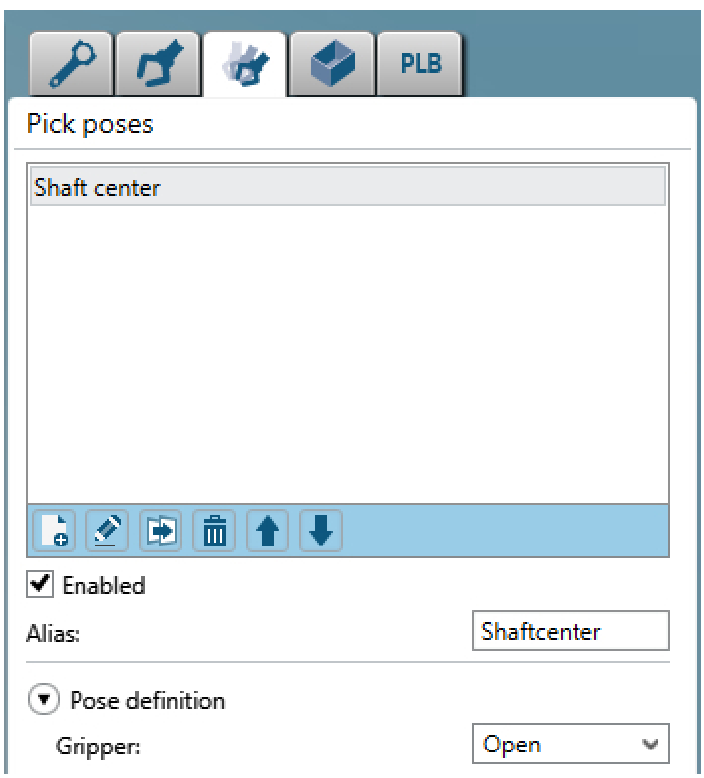


Figure 66: Configuring pick poses

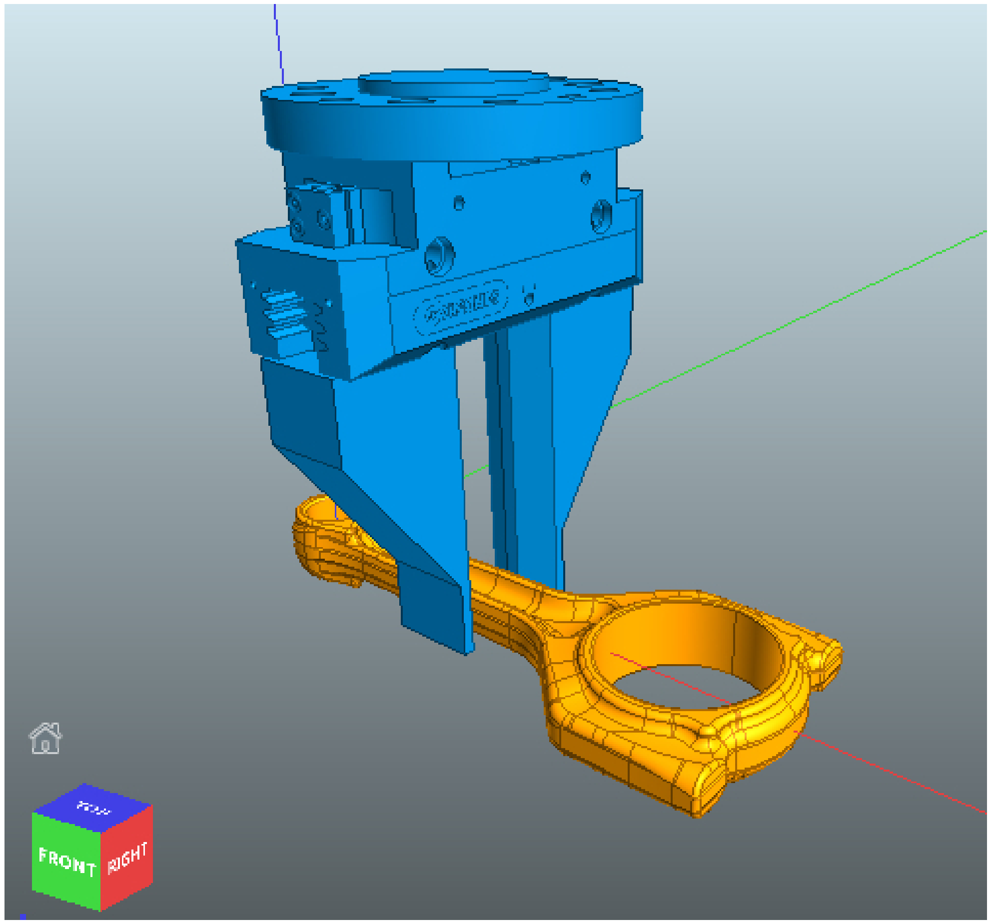


Figure 67: Display of the pick pose in the visualization area

Configure the pick pose using the robot controller

1. Ensure that the following steps of the job configuration are completed:
   - Configuring parts
   - Configuring a gripper
   - Configuring the bin and search volume

NOTE
The instructions for configuring the bin and search volume can be found in the section after this one, see "Configuring the bin and search volume", page 75.

2. Send a `SetState` command from the robot controller to the PLB system to start **Run** mode.

   Alternatively, open **PLB Studio** and click 

.
3. Trigger an image.
   - Stationary: Send a `Trigger` command.
   - Robot mounted: Send a `TriggerRM` command.
4. Send a `LocatePart` command.
5. Send a `SetState` command from the robot controller to the PLB system to start **Configure** mode.

   Alternatively, open **PLB Studio** and click 

.
6. Move the gripper to the desired picking position for the part.
   - Note: The part must remain still while doing so.
7. Send a `DefinePickPose` command with the gripper TCP coordinates.

✓ The pick pose is saved in the PLB system.

✓ The job is unlocked, if it uses the Surface algorithm.

8. Repeat the above steps for each desired pick pose.
   - Detailed information on the robot commands can be found in "Robot commands", page 152.

7.4.5.1 Configuring pick poses for limited pick rotation

This section is applicable to edge and plug-in algorithms.

If the **Limit pick rotation** feature is selected (see "Configuring the reference point for limited pick rotation", page 79), a reference point with an axis is visible when you configure pick poses. Follow the procedure in "Configuring pick poses", page 69 and make sure that the pick pose x-axis is parallel to the reference point axis, see figure 68.

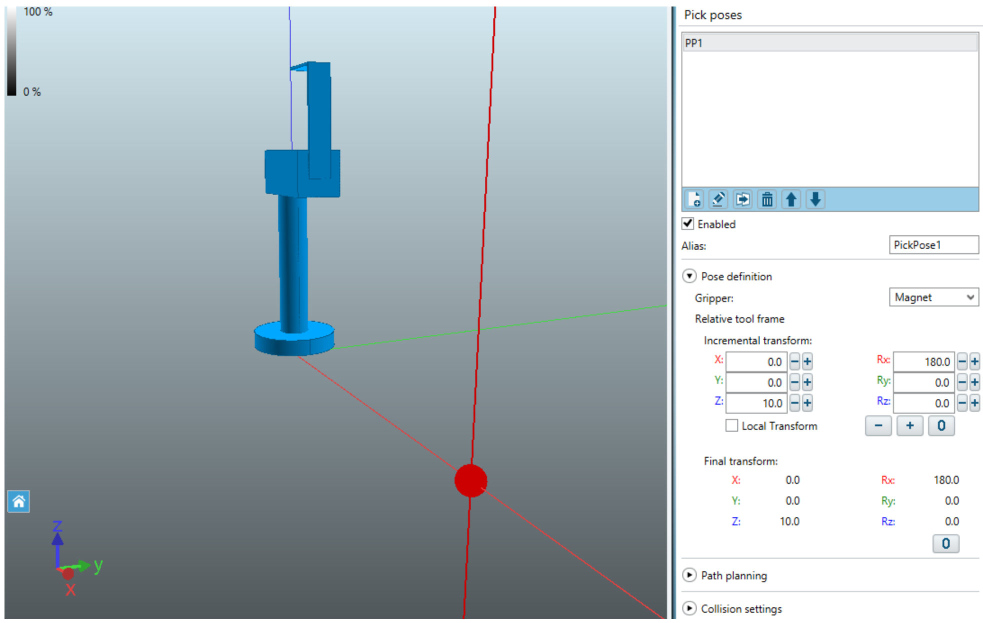


Figure 68: Gripper parallel to the reference point axis

7.4.5.2 Configuring rotational pick poses

For Surface algorithm jobs, it is possible to define pick poses in an arc around a rotation point, see figure 69. The start- and endpoints of the arc are visualized as transparent grippers. It is possible to adjust the position and tilt of the rotation axis.

1. Follow the procedure in "Configuring pick poses", page 69 to configure a pick pose.
2. Under **Dynamic pick poses**, select **Rotational pick pose**.
3. Enter a value (0...180) in the **Rotation start** and **Rotation end** boxes or drag the sliders, to define an arc around the rotation point.
4. Configure the rotation point and rotational axis. The controls are the same as when configuring the pick pose. See "Configuring pick poses", page 69, step 4..

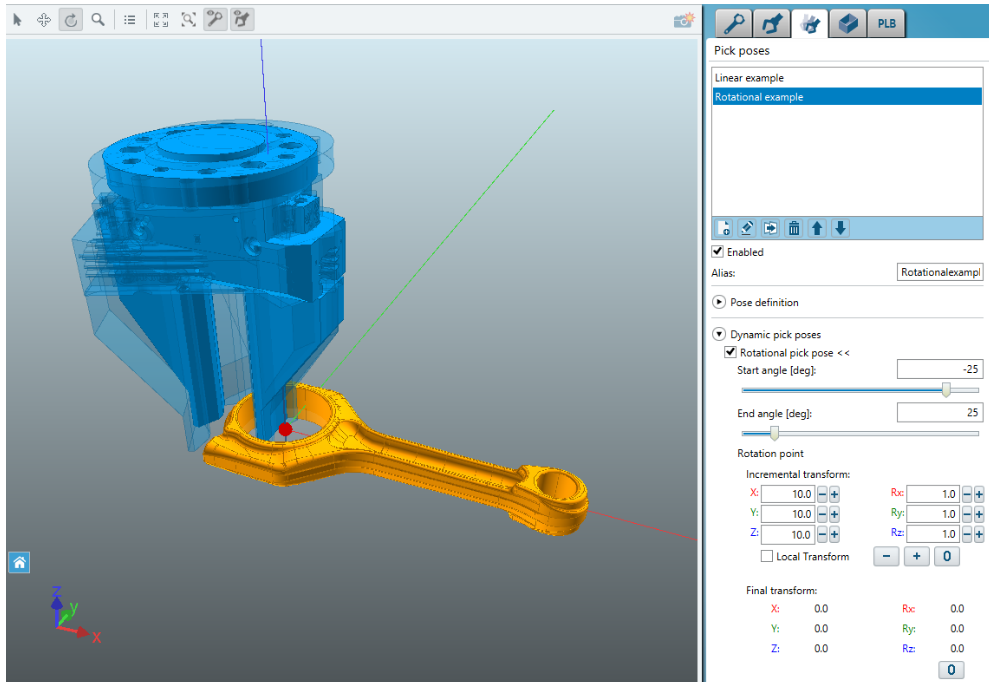


Figure 69: Configure a rotational pick pose

7.4.5.3 Configuring linear pick poses

For Surface algorithm jobs, it is possible to define linear pick poses along a line. see figure 70.

1. Follow the procedure in "Configuring pick poses", page 69 to configure a pick pose start point.
2. Under **Dynamic pick poses**, select **Linear pick pose**.
3. Set the pick pose stop point. The controls are the same as when configuring the pick pose. There are no controls for rotating the gripper. See "Configuring pick poses", page 69, step 4..

✓ The start- and endpoints are visualized as transparent grippers.

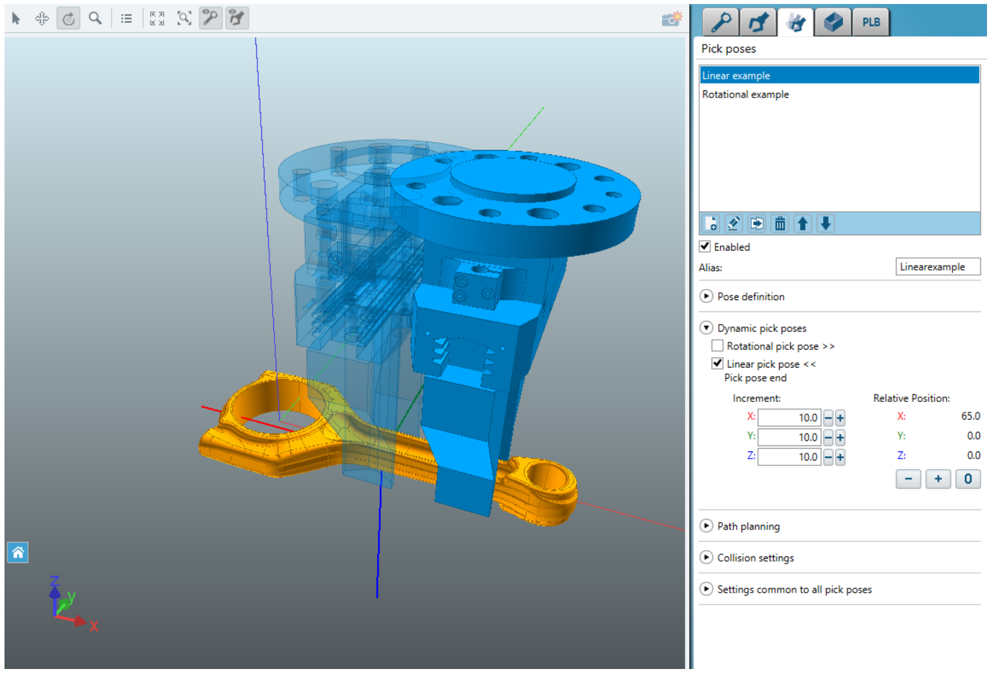


Figure 70: Configure a linear pick pose

7.4.6 Configuring the bin and search volume

To delimit the volume within which the system searches for parts, you must configure a search volume or a bin. A bin is defined with walls, and a search volume is defined without walls. (See also "Volume of view and search volume", page 114).

In this instruction, bin and search volume are used interchangeably.

All points located outside the search volume are ignored as possible starting points for the localization algorithm but are nevertheless taken into account for the purposes of collision detection.

To configure a bin or search volume:

1. Select the required job from the job drop-down list.
2. In the parameter editing area, click **Bin and search volume** to edit bin configuration.
3. Click to create a new bin or search volume and then give it a name.
4. In the visualization area, action toolbar, click to start an image acquisition.
5. In the **Volume type** list, select **Bin**, or **Search volume**.
6. In the **Volume position** list, select **Fixed in robot cell**, or **Follow camera**.
7. Under **Definition**, enter the dimensions of the bin or search volume (see figure 71).
8. Change the values under **Incremental transform** so that visualization of the bin or search volume is aligned to the image (see figure 72):
   - If **Volume type** is **Bin**, click **Auto adjust bin pose** to let PLB automatically detect the bin pose.
   - Select **Local transform** to make the modifications in the bin's local coordinate system.
   - Click and to modify the final transform for all axes.
   - Click the plus and minus signs next to each text box to modify the final transform for one axis at a time.
     The final transform (in global coordinates) is described by the bin's translation (X, Y, Z) in mm and its rotation (Rx, Ry Rz) in degrees. Click to set all the values to zero.
9. Repeat the above steps for each desired bin. Click to clone a bin.

10. Detailed information on the other parameters available for bin configuration is found in "Bin and search volume", page 145.

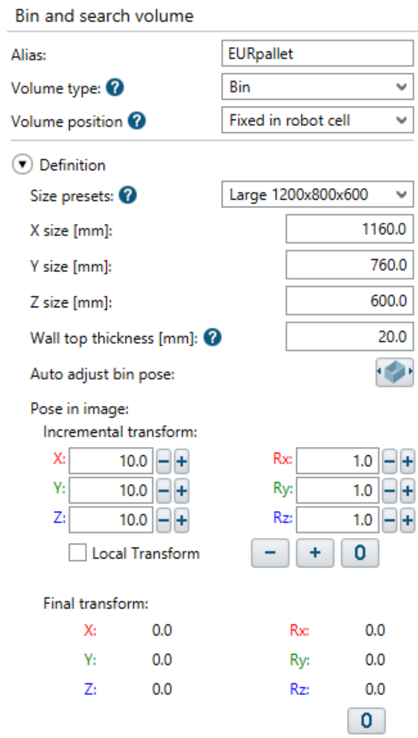


Figure 71: Configuring the bin and search volume

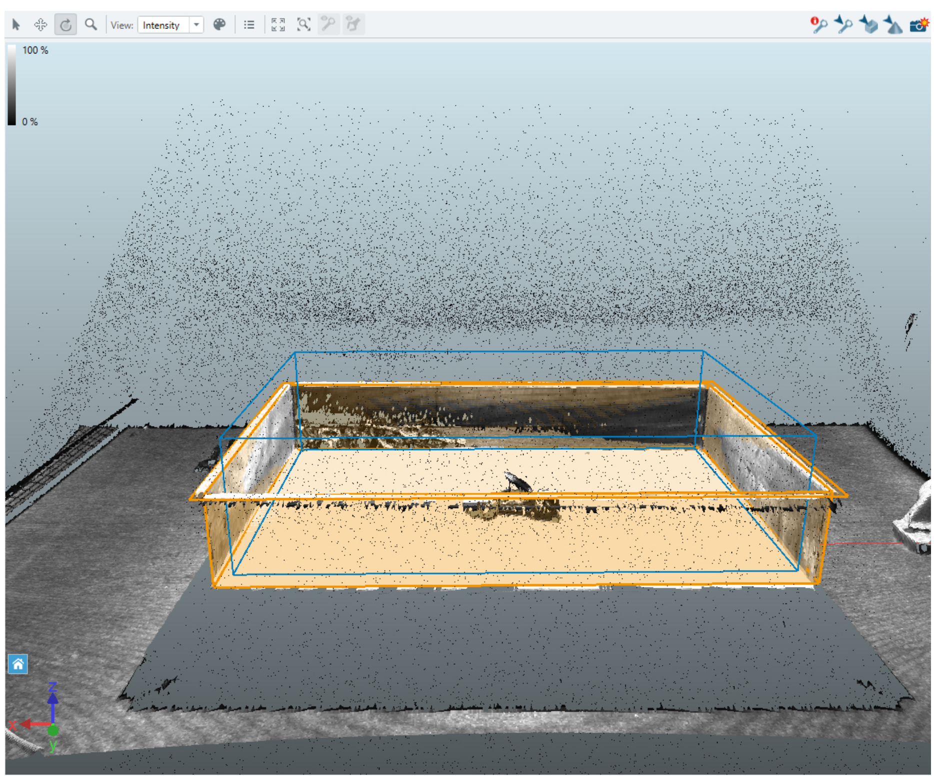


Figure 72: Display of the bin and search volume in the visualization area

7.4.6.1 Configuring result sort order for edge and plug-in algorithms

Configure result sort order

This section is applicable to edge and plug-in algorithms.

The bin coordinate system axes (x, y, z) are displayed in the center of the defined bin. The lanes or layers defined in the following steps are defined in this coordinate system.

1. In the parameter editing area, click **Bin and search volume** .
2. Expand the **Result sort order** section.
3. Select **Result sort order** to define the order of the valid locate results:
   - **Highest z value**
   - **Lanes**
   - **Layers**
   - **Highest score**
   - **Algorithm priority**
   - **Least tilted**
   - **Most tilted**
4. If you selected:
   - **Lanes**, continue in Configure lanes.
   - **Layers**, continue in Configure layers.

Configure lanes

1. Select **Lane order** to define from which bin edge the lanes start:
   - **Highest x value**
   - **Lowest x value**
   - **Highest y value**
   - **Lowest y value**
2. Select **Sorting within lanes** to define how to sort the results within each lane:
   - **Highest score**
   - **Highest y value** or **Highest x value**
   - **Lowest y value** or **Lowest x value**
   - **Highest z value***
   - **Least tilted**
   - **Most tilted**

   Note that **Lane order** and **Sorting within lanes** must be perpendicular to each other. This means, for example, that if you select **Highest x value** or **Lowest x value** in step 1., neither of them are available in step 2.
3. Set **Lane width** to define the width of each lane. A lane is specified from the center of the first (according to **Lane order**) valid part found, with half of the **Lane width** on either side. Parts with its center within this lane are considered to belong to the lane. The next lane starts from the center of the next valid part not part of the previous lane.

✓ The defined lane is visualized as shown in figure 73.

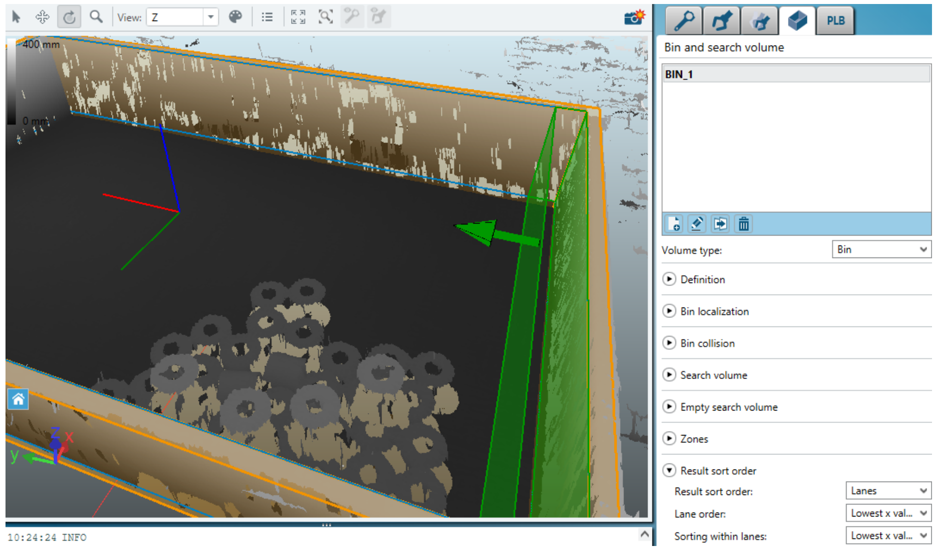


Figure 73: Configuring lanes

*The **Highest z value** option is not available when **Lanes** are selected in **Sorting within layers**.

Configure layers

1. Select **Sorting within layers** to define how to sort the results within each layer:
   - **Highest score**
   - **Highest x value**
   - **Lowest x value**
   - **Highest y value**
   - **Lowest y value**
   - **Least tilted**
   - **Most tilted**
   - **Lanes**

   If you selected **Lanes**: follow the steps in figure 73, then continue with step 2..
2. Set **Layer depth** to define the depth of each layer. A layer is specified from the center of the topmost valid part found, with an equal amount of space above and below. Parts with their center within this layer is considered to belong to the layer. The next layer starts from the center of the next valid part not part of the previous layer.

✓ The defined layer is visualized as shown in figure 74.

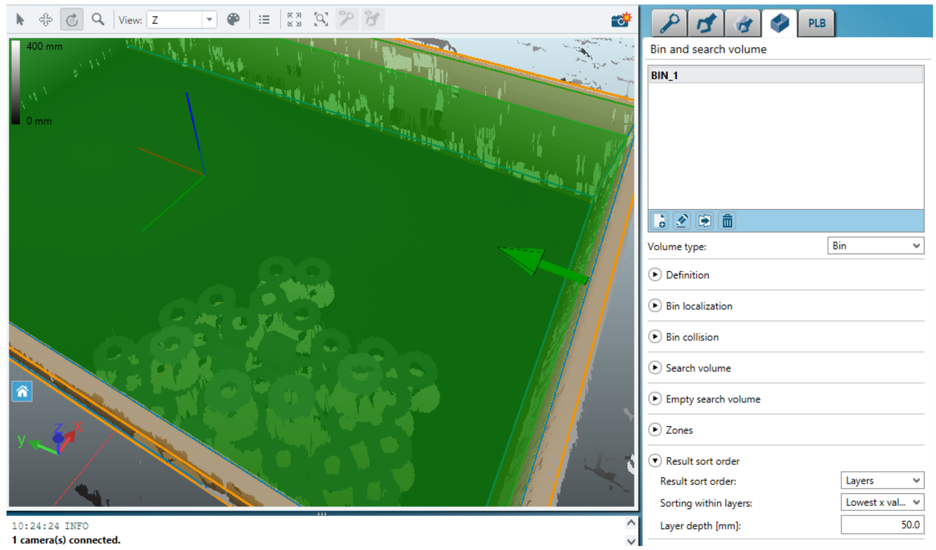


Figure 74: Configuring layers

7.4.6.2 Configuring the reference point for limited pick rotation

This section is applicable to edge and plug-in algorithms.

The purpose of the **Limit pick rotation** option is to adjust pick poses so the robot does not pick from angles that are hard to reach. **Limit pick rotation** is dependent on part and gripper characteristics. The following modes are available, with their respective part and gripper requirements.

Rotation mode (selected in step 4.)

- **Rotate gripper**: The gripper is rotated around the Z-axis to the limit pick rotation reference point in all pick positions. Pick poses are adapted to the located part pose and the limit pick rotation reference point. This option is normally used for single point suction or magnetic grippers, since asymmetric grippers will grip differently if rotated at the pick position.
- **Rotate part**: The located part is artificially rotated around the Z-axis to the limit pick rotation reference point. Pick poses are based on the new artificial part pose instead of the original located part pose. This option is normally used for parts that are symmetric around the Z-axis, since there otherwise will be a difference between the real part pose and the artificial part pose.

To configure the reference point for the pick pose, do as follows:

1. In the parameter editing area, click **Bin and search volume** .
2. Expand the **Limit pick rotation** section.
3. Select **Rotate pick pose to reference point**.
4. Select **Rotation mode**.
5. Set **X**, **Y**, and **Z** to define the position of the reference point:
   - Click the and buttons to modify the final position for all axes.
   - Click the plus and minus signs next to each text box to modify the final position for one axis at a time.
   - Click to set the reference point to the center of the bin.
   - Select **Lock reference point to search volume** to lock the reference point to the bin position. That is, the reference point will change if the bin is changed or if a locate bin command changes the bin position.
   - When **Lock reference point to search volume** is selected, bin position changes using the increment transform control is in the local bin coordinate system.

✓ The reference point is visualized as a red sphere with an axis parallel to the world

Z-axis or the local bin Z-axis (if **Lock reference point to search volume** is selected). See figure 75. It is also visualized in the **Pick poses** and **Analyze** tabs.

✓ The reference point changes color from red to orange when **Lock reference point to search volume** is selected. This indicates that the reference point will move if the bin position is changed.

✓ The origin of the local bin coordinate system is visible when **Lock reference point to search volume** is selected.

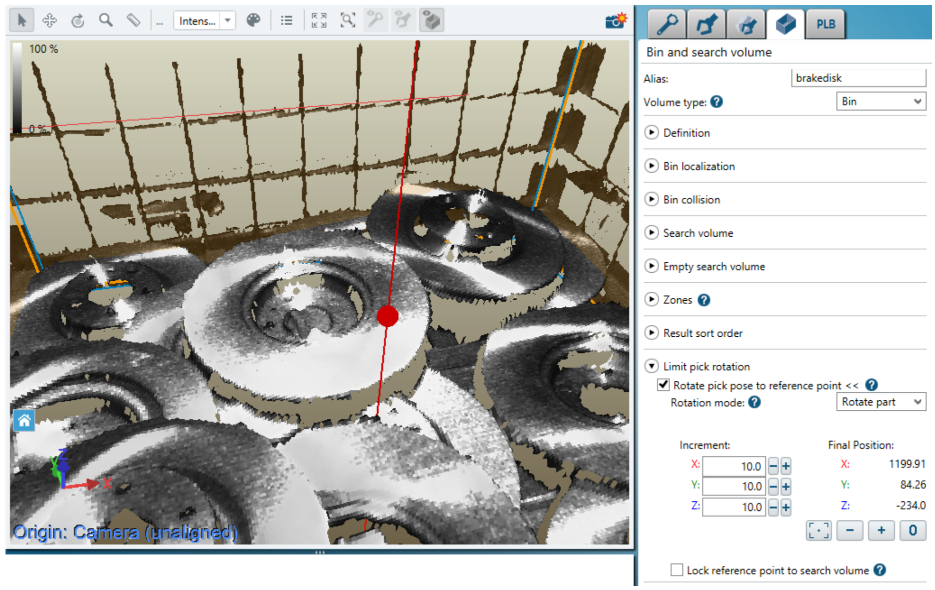


Figure 75: Defining the reference point

7.4.7 Locking the job

Lock the entire job (part, grippers, and pick poses) by clicking the in the menu bar. The job needs to be locked to be able to run it.

The calculation may take a certain time to complete. During this time, the icon becomes a . The icon is displayed to indicate that the calculation is complete.

8 Operation

8.1 Running PLB Watchdog

PLB Watchdog is part of the PLB software installation.

PLB Watchdog is an application that watches that PLB Engine is running and responsive. If the PLB Engine process is not found or seems to be stuck in a bad state, the application will restart PLB Engine.

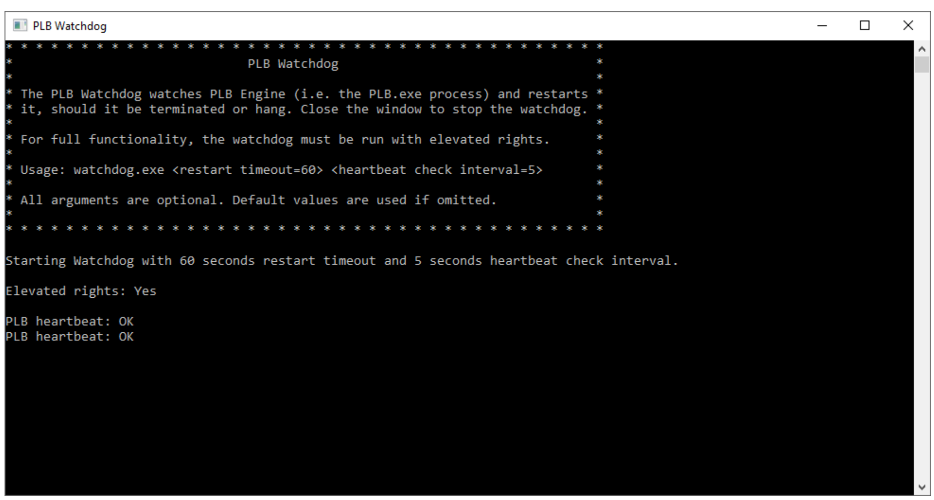
NOTE
PLB Watchdog will restart PLB Engine also if PLB Engine is shut down from the System Tray taskbar. To properly shut down PLB Engine, close the PLB Watchdog terminal window before shutting down PLB Engine.

NOTE
PLB is operational without PLB Studio. Running PLB Engine without PLB Studio is recommended for maximum performance and stability if PLB is used non-stop for more than a week without a restart. To disable the automatic start of PLB Studio when starting PLB Engine, see "To disable automatic start of PLB Studio", page 81.

To start and stop PLB Watchdog

1. Click **Start** to open the **Start** menu.
2. Enter `PLB Watchdog`.
3. Click the **PLB Watchdog** entry to start the application.

✓ PLB Watchdog opens in a terminal window.



- To quit the application, close the terminal window.

To disable automatic start of PLB Studio

By default, PLB Studio is automatically started when PLB Engine starts. To stop PLB Studio from starting when PLB Watchdog restarts PLB Engine, disable automatic start of PLB Studio.

- In PLB Studio, click **Options**, then click **Start PLB Studio when PLB Engine starts** to clear the checkmark.
- To enable automatic start, click **Start PLB Studio when PLB Engine starts** to add the checkmark.

8.2 Run jobs

8.2.1 Starting the image acquisition

WARNING
Pay attention to the safety notes in "Operational safety and particular hazards", page 11 when using the PLB system.

Proceed as follows to start image acquisition:

1. Deactivate your computer's standby and idle mode.
2. Open **PLB Studio**.
3. Click Run to start **Run** mode.
   Alternatively, you can send a `SetState` command from the robot controller to the system (see "SetState", page 153).
4. Click to acquire an image. Alternatively, you can send a `Trigger/TriggerRM` command from the robot controller to the system (see "Trigger", page 153 or "TriggerRM", page 153).

NOTICE
Robot mounted camera
All robot mounted commands for triggering an image acquisition must always include the position and rotation of the robot's flange from the default origin of the robot (usually the robot base).
If triggering an image acquisition from PLB Studio, PLB will use the latest position and rotation sent in a command from the robot.

5. Wait while the image data is acquired. The image is then displayed in PLB Studio.
6. Click to acquire a new image.

8.2.2 Viewing data in PLB Studio

NOTE
If your computer is running at high load then there may be some delay when displaying the measurement results, i.e. the display in PLB Studio is not necessarily synchronous with the transmission of the data by the robot controller. Delays of this sort only affect the display. The transmission of data to the robot controller always has a higher priority.

In **Run** mode, the following information is displayed at runtime (see figure 76, page 83):

- 3D image of the bin, the located part and the calculated pick pose (visualization area, **A**)
- Toolbar with action buttons (**B**); **Locate part**, **Locate bin**, **Verify alignment**, and **Trigger image acquisition**
- Result log (**C**), see description below figure 76 for further details
- Latest result (**D**), shows numbers on the latest action:
  - the coordinates of where a part was found, or
  - the size of a detected bin, or
  - the coordinates of a detected alignment target

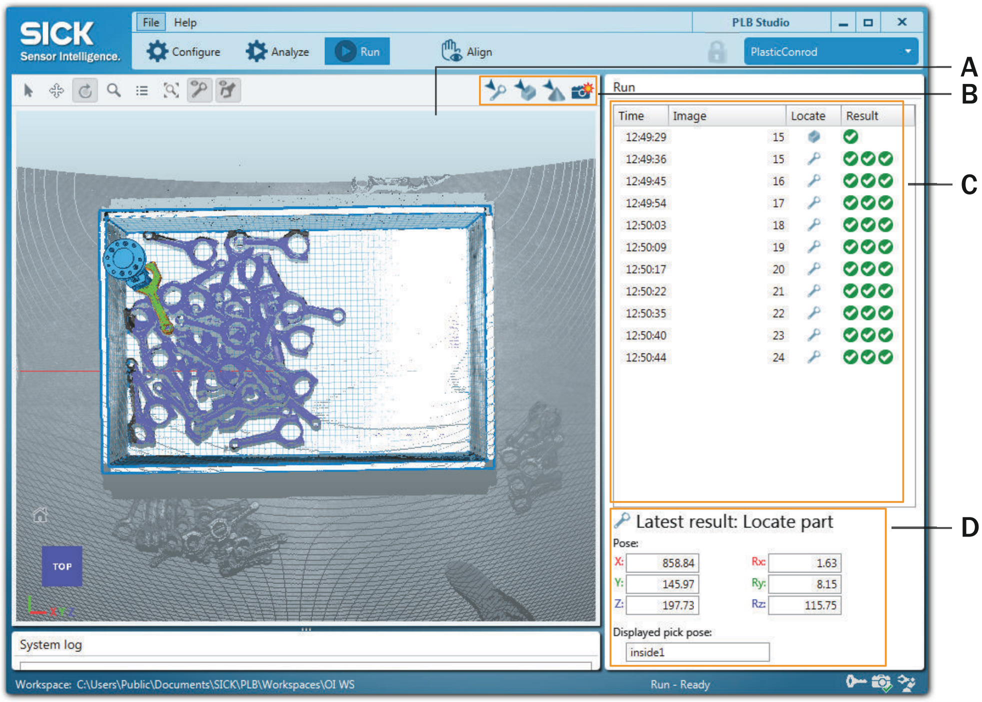


Figure 76: PLB Studio, Run mode

The result log (**C**) shows a list of the latest results of the `LocatePart`, `LocateBin` and `VerifyAlignment/VerifyAlignmentRM` actions. The columns are:

- Time; the time of the response
- Image; the ID of the image that was used for analysis
- Locate; what was located: part, bin or alignment target
- Result; the result of the action

The result column contains 1, 2 or 3 icons, where means successful and means failed. `LocatePart` result has three icons, the first shows if the part was located, the second if the part was overlapped and the third if collision was detected. `LocateBin` result has one icon, showing if the bin was located or not. `VerifyAlignment/VerifyAlignmentRM` result has two icons, the first show if a target was located or not and the second shows if the located target was within limits. A means that the partial result was not calculated, due to an earlier partial result failed.

8.2.3 Running the job configuration

Run mode allows you to run your job configuration and also acquire and view data.

Before locating the first part, the bin needs to be located to set the search area. This is also important to do if the bin has been changed or moved in the robot cell.

You should now proceed as follows:

1. Click to start image acquisition.
   Alternatively, you can send a `Trigger/ TriggerRM` command from the robot controller to the system (see "Trigger", page 153 or see "TriggerRM", page 153).
2. Click to determine the position of the bin in the acquired image (point cloud). This operation is performed on the basis of the search volume defined by you in the job configuration (see also "Configuring the bin and search volume", page 75).

Alternatively, you can send a `LocateBin` command from the robot controller to the system (see "LocateBin", page 156).

3. Click to determine the position of the next reachable part in the acquired image (point cloud).
   Alternatively, you can send a `LocatePart` command from the robot controller to the system (see "LocatePart", page 154).
4. If required, choose another job.
5. Click to determine the position of the newly selected job in the same image (the same point cloud).

When doing so, please note the following points:

- The status of part and bin localization are both indicated by means of the icons = Successful and = Failed.
- If you have accurately configured the search volume, i.e the configured and actual bin positions match, then bin localization can be performed successfully.
- The `LocateBin` result is the position of the bin identified in the point cloud. This is stored temporarily and is used for the calculation of the position of the part when you click . If you modify the parameters for the search volume during configuration then the result is discarded.
- If the attempt to locate the bin using fails, then the search volume created by you in the job configuration is used for part localization.

8.2.4 Managing recorded data

NOTE
We recommend using an HDD drive to store the recording files. The automatic function for cleaning up the recording folder can take a long time to run if an SSD drive is used. It also results in premature wear to SSD drives.

In **Run** mode and **Align** mode the system automatically saves the recorded data. The data is primarily used for analysis and troubleshooting. Images acquired in **Configure** mode are not saved. If you acquire new images in **Analyze** mode, they are saved if they are analyzed with the `LocatePart`, `LocateBin`, or `VerifyAlignment`/ `VerifyAlignmentRM` actions. If they are only analyzed using breakpoint they will not be saved.

Recording folders

During the installation of the PLB software package, the base folder for recordings is created under `C:\ProgramData\SICK\PLB`.

Every time you go to the **Analyze**, **Run** or **Align** mode, a new recording folder is created in the base folder. The recording folder is named after the recording time (format: `Recording_ YYYY-MM-DDThh.mm.ss.xxxxxx`). The recording folder contains the images and configuration files.

The following files are written to the recording folder:

- A grayscale image for providing an overview of the scenario.
- The report files, containing all the results that have been output during system operation.
- The .DAT and .XML files, containing the image data used as input to PLB.
- The internal job configuration file for the current job.

Each recording folder can contain four numbered subfolders. Every numbered folder can contain up to 30 images. When the numbered folders are full a new recording folder is created automatically.

Configure the storage location

Settings for managing recordings are available in the parameter editing area, on the **System settings** tab .

You can change the storage location for the base folder under **Recording storage>Recordings base folder** (see figure 77, page 85).

You can configure the number of stored recording folders by changing the parameter **Recording storage** > **Maximum number of sub folders.** When the maximum possible number of recording folders has been reached, the oldest folder is deleted. Write-protected folders are ignored here.

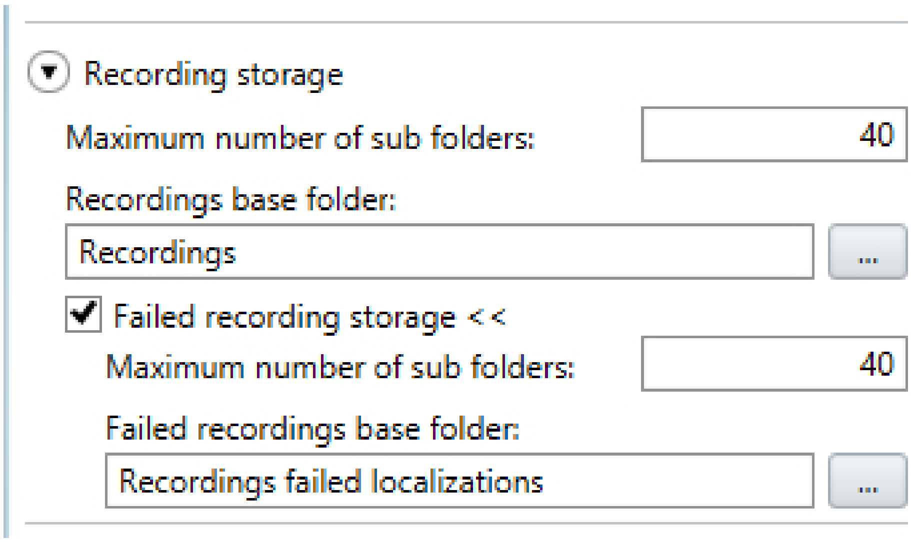


Figure 77: Base folder for recordings

Store failed localization recordings

Recordings from failed bin or part localizations can be stored in a separate base folder. A failed recording folder is only created when localization fails.

Only the images that cause failures and the failed result files (there may be successful localizations made on the same image) are stored in the failed recording folder.

Select **Failed recording storage** to automatically store failed localizations. See figure 77.

Archive folder

Images in the Recordings folder will eventually be overwritten. It is recommended to manually move important recordings to the Archive folder.

To select the path to the Archive folder, proceed as follows:

1. Click to configure **System settings.**
2. Go to the settings for **Archive**, see figure 78, page 86.
3. Specify the path to the **Archive base folder.**

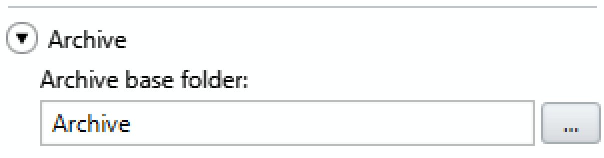


Figure 78: Base folder for Archive

The **Recordings** folders that are put in the **Archive** folder can be renamed, but the **Recordings** subfolders and file names need to remain unchanged.

8.3 Analyze jobs

8.3.1 Viewing data and parameters in Analyze mode

Analyze mode is a combination of the **Configure** mode and the **Run** mode. Use the **Analyze** mode to analyze images and test configuration settings.

Proceed as follows to start **Analyze** mode:

1. Open **PLB Studio**.
2. Click  to start **Analyze** mode.

The following information is displayed in **Analyze** mode, see figure 79.

- The 3D visualization area with the action buttons toolbar. The action buttons are the same as in Run mode, with the addition of a toggle button (**A**) to display the **Locate part analysis** control.
- The parameter editing area is the same as in **Configure** mode, with the addition of the **Analyze** tab. The selected tab (**B**) determines the content of the (**C**) panel.

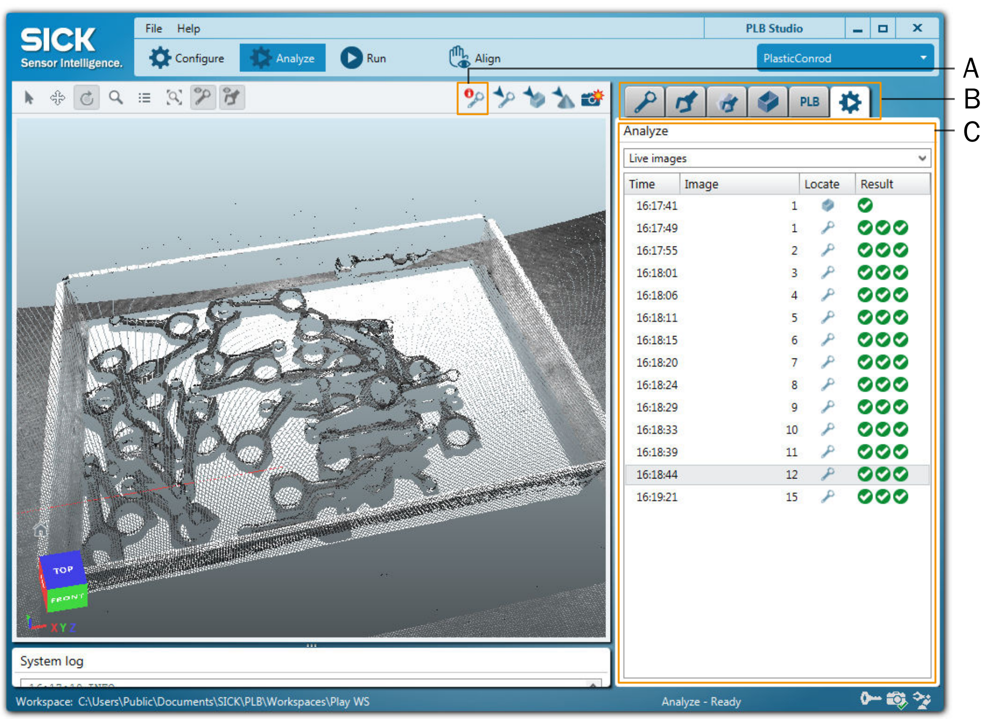


Figure 79: PLB Studio, Analyze mode

Analyze tab

When the **Analyze** tab (**B**) is selected a list of images (with results) is displayed in the (**C**) panel. The content of the list depends on the selection in the drop-down control at the top of the panel. Use the drop-down (see figure 80, page 87) to select images that should be available for analysis:

- Live images, containing the latest 100 images that have been analyzed, using `LocatePart`, `LocateBin`, or `VerifyAlignment`/`VerifyAlignmentRM`, since **PLB Engine** was started.
- Folders in the **Recordings** folder (see "Managing recorded data", page 84)
- Folders in the **Archive** folder, containing images from previously recorded data (see "Managing recorded data", page 84)

Figure 80: Select images available to analyze

Locate part analysis control

When starting **Analyze** mode a **Locate part analysis** window is available, see figure 81, page 88. If the window is closed it can be displayed by clicking the  button. For further information on how to use it, see "Analyzing measurement results using breakpoints", page 91.

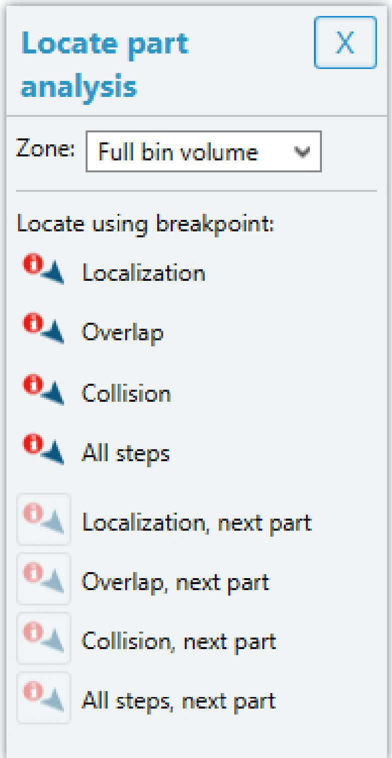


Figure 81: Locate part analysis (surface part job)

Result details

When locating a part, bin, or alignment target in **Analyze** mode, a window with detailed information will be displayed, see figure 82, figure 83 and figure 84.

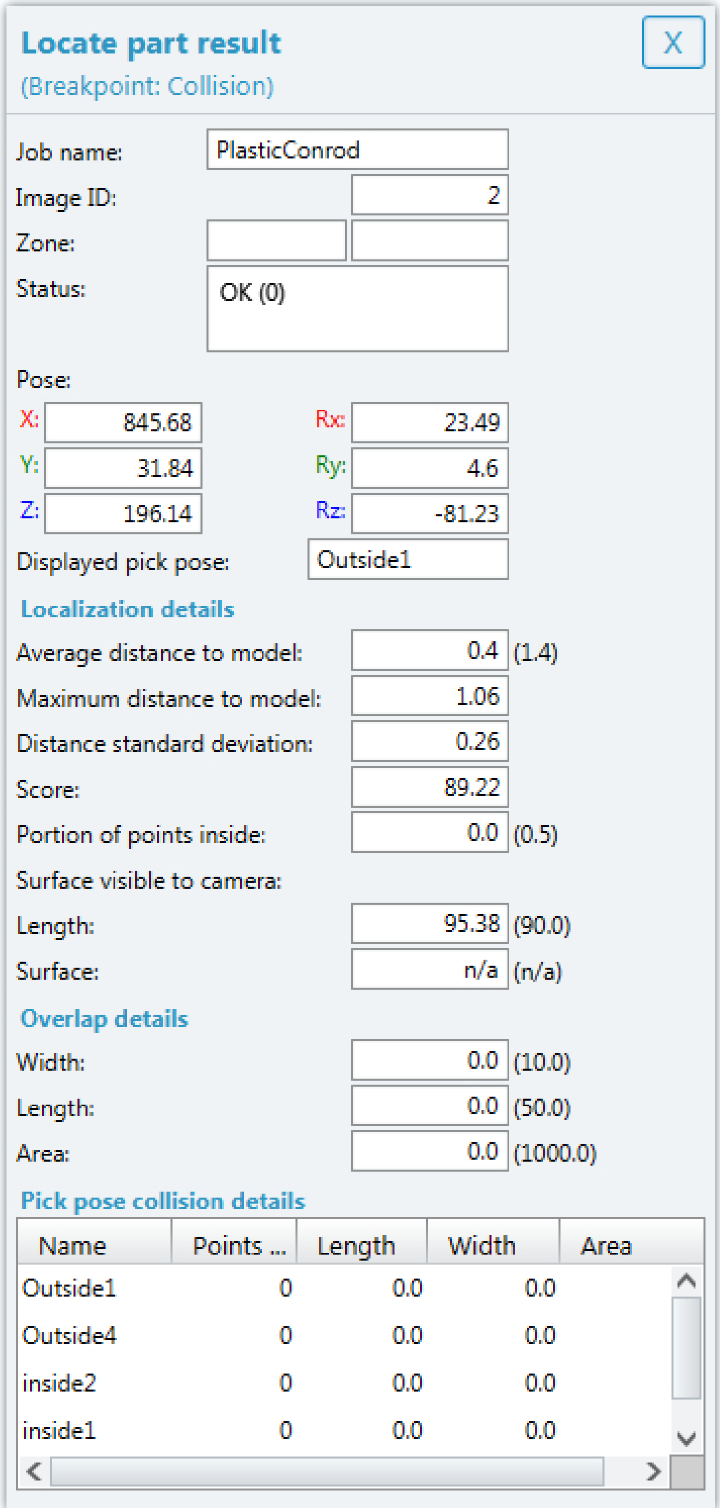


Figure 82: Locate part result (surface part job)

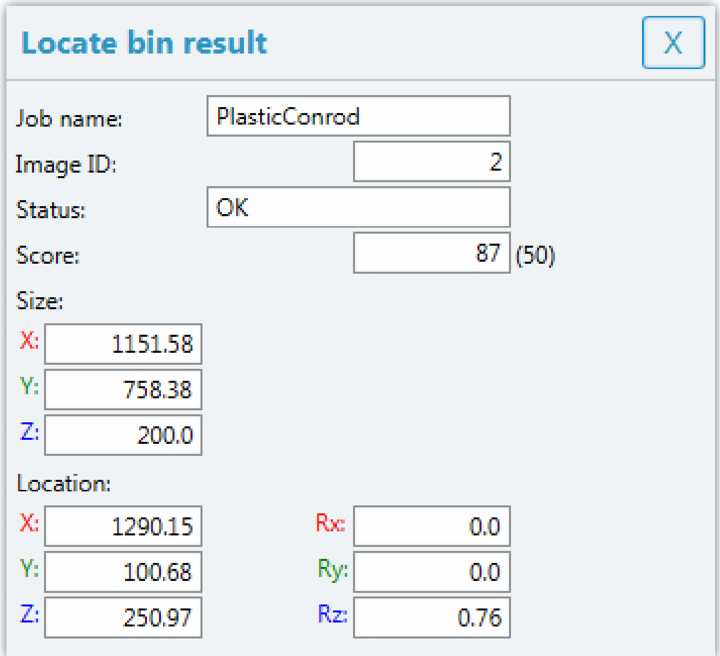


Figure 83: Locate bin result

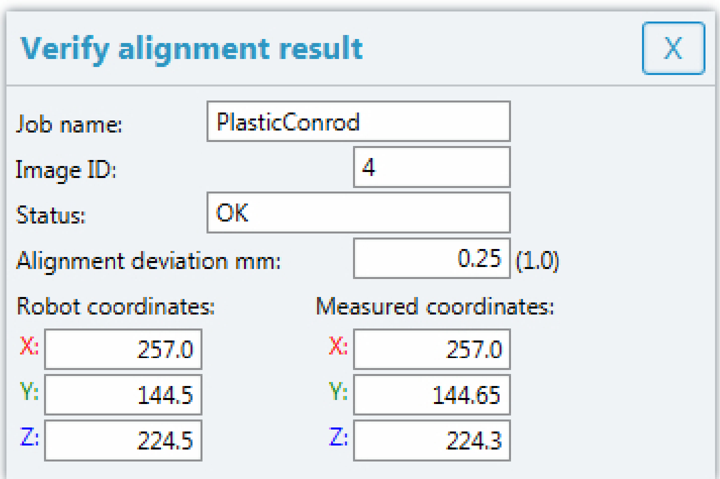


Figure 84: Verify alignment result

8.3.2 Optimizing the job configuration

You can use the **Analyze** mode to analyze images and test modified configuration settings.

8.3.2.1 Reproducing measurement results

To reproduce a measurement result, proceed as follows:

1. Start **Analyze** mode.
2. Select available images in the drop-down list (see figure 80, page 87).
3. Select the image to analyze.

✓ The point cloud displayed in the 3D visualization area is updated.

4. Click the  button to reproduce the result.

The result is now visualized in the 3D visualization area and the result details are displayed in the **Locate part result** window (see figure 82, page 89).

NOTE
Please note that in **Analyze** mode the **Blacklist** functionality is discarded.

8.3.2.1.1 Result visualization

The **Pick pose collision details** list in the **Locate part result** window views all possible pick poses for the part. For the selected pick pose in the list, an animation of the gripper's path to the pick pose is shown.

- Select a pick pose in the list to show an animation of its path. Double-click the selected pick pose to repeat the animation.
- To change the animation settings, right-click on a pick pose in the list and select one of the options below. The selected option applies to all pick poses.
    - **Approach animation** animates the gripper's path between the Approach pose and the pick pose. Note: This option is not supported by the Surface algorithm.
    - **Path animation** (default) animates the gripper's path to the pick pose, including the defined Entry, Approach, Clear and Exit poses.
    - **No animation** turns the animations off.

8.3.2.2 Analyzing measurement results using breakpoints

When performing the calculations the software runs through the software modules **Localization**, **Overlap**, and **Collision**. To get a deeper understanding of the results it is possible to stop the analysis of the image at breakpoints. The following breakpoints are available:

- **Localization:** The calculation is stopped in the localization software module; the located part is displayed, the localization thresholds are not applied, overlap and collision are not calculated.
- **Overlap:** The calculation is stopped in the overlap software module; the located part is displayed, with overlapping points in red. Thresholds configured for overlaps are not applied. No collision detection is performed.
- **Collision:** The calculation is stopped in the collision software module. Thresholds configured for collisions are not applied. Collision points are visualized in red. see "Collision results", page 93 for more information about the collision results.
- **All steps:** All the steps are run and the corresponding result is displayed.

When stepping through the analysis, it is possible that the part that is displayed with breakpoint **Overlap** is not the same as the part that was displayed with breakpoint **Localization**. If this is the case then the first part did not pass the **Localization** limits. Analogously this can happen with the other breakpoints as well.

NOTE
Overlap calculations are only applicable to the Surface algorithm.

To analyze measurement results using breakpoints proceed as follows:

1. Start **Analyze** mode.
2. Select available images in the drop-down list (see figure 80, page 87).
3. Select the image to analyze.
4. Click a  button to start calculations to the breakpoint.

The result is visualized in the 3D visualization area and the calculated numbers are displayed in a **Locate part result** window (see figure 82, page 89).

The next steps depend on the results received:

- If the located part is the one you are interested in, it is likely you want to change some parameter settings and then perform the same calculations with the new settings.
  a) Select the tab (**Part**, **Pick poses**, or **Bin and search volume**) with the settings you want to change.
  b) Change the settings.
  c) Click a button to start the calculations.
- If the located part is not the part you are interested in, it is likely you want to get the next part.
  a) Click a button to start the calculations to a breakpoint, **Next part.**
  b) If needed, click the button again to get the next '**Next part**'.

NOTE
If "Depth layer filtering" is enabled, then only the first layer will be analyzed when using breakpoints.

Next part

When the breakpoint is **Next part** the calculations will continue from where they stopped in the previous calculation. There has to be a step to continue from, either a "first" calculation or a "next" calculation.

NOTE
When it is possible to continue with a "next" calculation, the buttons are enabled. If it is not possible to continue the "next" calculation, the buttons are disabled and you need to start over with a "first" calculation.

For plug-in or Edge algorithm jobs, the "next part" button adapts depending on the "first" calculation. To change the type of breakpoint calculation, you need to start over with a "first" calculation.

Surface algorithm examples:

- If the first step is breakpoint **Localization** and you select to calculate **Localization, next part**, you will get a new part.
- If the first step is breakpoint **Localization** and you select to calculate **Overlap, next part**, you will get overlap calculations for a new part. If you want to see overlap for the first part you need to select **Overlap**, provided that the first part passed the localization limits.
- If the first calculation is **All steps** (you will get the first part that passes all tests) and you select to calculate **Localization, next part**, you will get a new part.
- If the first calculation is **All steps** (you will get the first part that passes all tests) and you calculate **Localization, next part**, you will get a new (second) part. If you continue and select to calculate **Overlap, next part**, you will get a new (third) part. If you (in the third step) want to see overlap for the second part in this sequence you need to start over with **All steps** and then **Overlap, next part.**

NOTE
Please note that during the calculations it is possible that several attempts were made and discarded to find a part that passed all tests.

8.3.2.2.1 Collision results

When performing an analysis with **Collision** as breakpoint, all enabled pick poses and their collision results are shown in the **Pick pose collision details** list in the **Locate part result** window.

For the Surface algorithm:

- **Points** lists the number of collision points for each pick pose. If no collision was detected, the **Points** value is set to zero.
- **Length**, **Width** and **Area** specify the dimensions and the bounding box size of the collision area for each pick pose.

If a collision was detected, the collision points for the selected pick pose are visualized as red dots in the image area.

For the Edge algorithm:

- The **Result** symbol shows if the collision check was OK or not for each pick pose. A gray symbol means that no collision check was performed.
- **Points** lists the number of collision points for each pick pose. If no collision was detected, the **Points** value is set to zero. If a collision with the bin was detected, the **Points** value is set to 'bin' (see figure 86, page 94).
- **Approach** lists the distance with free space along the approach vector. If a collision was detected at the pick pose, the **Approach** value is set to zero.

If a collision was detected, the collision points for the selected pick pose are visualized as red dots in the visualization area. If there was a collision with the bin, red dots are displayed where the collision occurred.

Right-click a pick pose in the list and select **Approach animation** to view an animation of the gripper's path along the approach vector (from the **Approach** pose to the pick pose):

- If a collision at the pick pose was detected, the gripper turns red, see figure 85, page 93.

Pick pose collision details

Name	Result	Points	Approach
Collision with approach	✗	Bin	95.0
Approach disabled	✓	0 pt	n/a
Collision disabled	●	n/a	n/a
Collision with approach ...	✗	11 pt	0.0

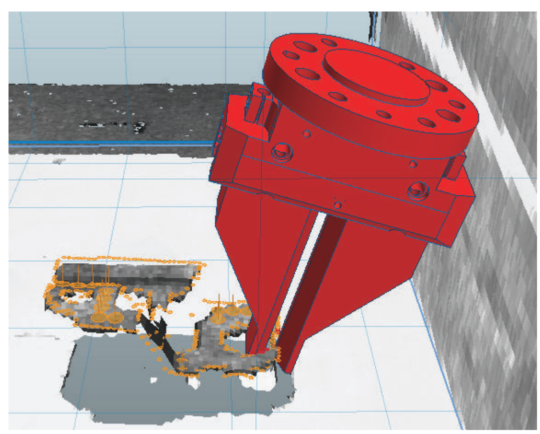


Figure 85: Collision at pick pose

- If there was no collision at the pick pose, the gripper's path to the pick pose is animated. If there is a collision along the path, the gripper model turns red and stops, see figure 86, page 94.

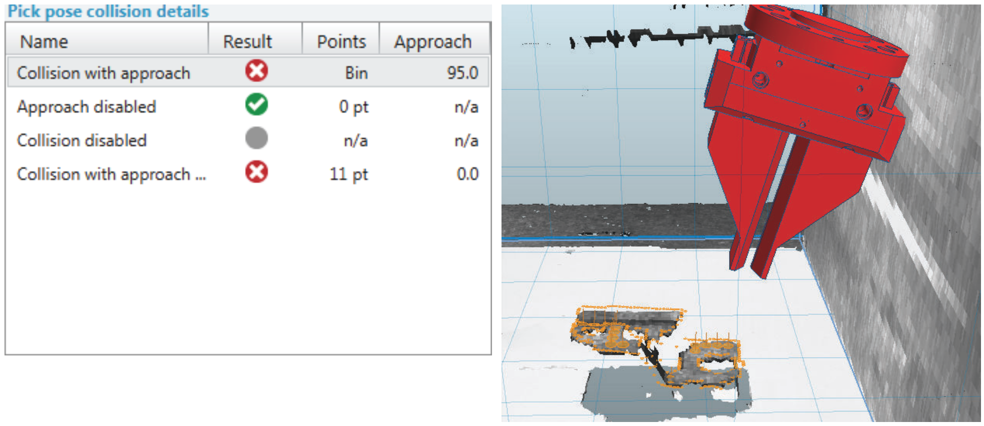


Figure 86: Collision with bin wall, 95 mm from pick pose

NOTE
The **Approach animation** is not supported by the Surface algorithm.

8.3.2.3 Analyzing measurement results in a specific zone

In **Run** mode the order of the zone being analyzed is based on the height of the contents of the zone. In **Analyze** mode you can select to analyze the full bin or a specific zone.

To analyze measurement results in a specific zone proceed as follows:

1. Start **Analyze** mode.
2. Select available images in the drop-down (see figure 80, page 87).
3. Select the image to analyze.
4. Select the desired zone in the **Zone** drop-down in top of the **Locate part analysis** window, see figure 87, page 94.
5. Click the  button to start the calculations.

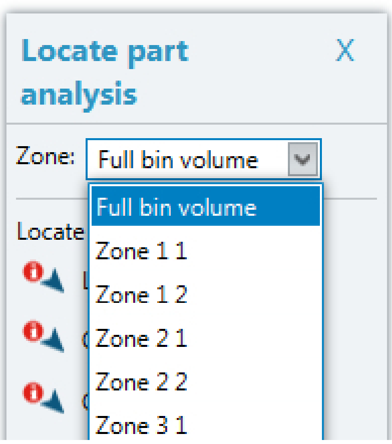


Figure 87: Zone selection

The result is visualized in the 3D visualization area and the calculated numbers are displayed in a **Locate part result** window (see figure 82, page 89).

8.3.2.4 Analyzing measurements in a specific search volume

If you want to analyze a fraction of the full search volume you can define a new search volume specifying that volume.

To analyze measurement results in a specific search volume proceed as follows:

1. Start **Analyze** mode.
2. Click to configure **Bin and search volume** settings.
3. Click to add a new search volume and edit the name.
4. Go to the settings for **Definition**.
5. Adjust the dimensions and position to your preference.
6. Click .
7. Select available images in the drop-down list (see figure 80, page 87).
8. Select the image to analyze.
9. Click a button to start the calculations.

The result is visualized in the 3D visualization area and the calculated numbers are displayed in a **Locate part result** window (see figure 82, page 89).

NOTE
Remember to activate the full search volume before going to **Run** mode again.

8.3.2.5 Suggestions on parameter changes depending on the status

In the **Locate part result** window a status message and a code are displayed, see figure 88.

Descriptions and suggested actions for all general status messages are summarized in table 20. Status messages applicable for the Surface algorithm only are summarized in table 21, and status messages applicable for the Edge algorithm only are summarized in table 22.

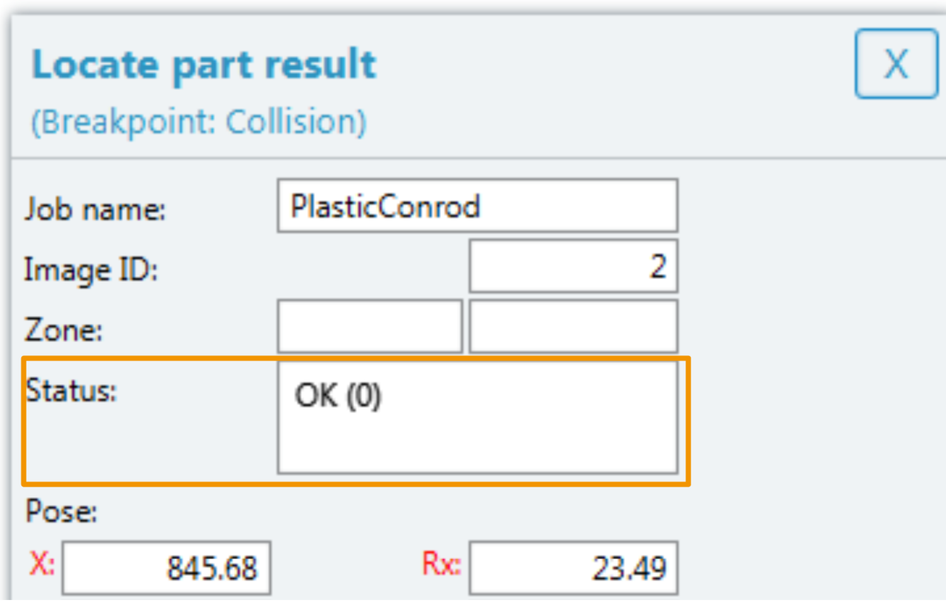


Figure 88: Status of a locate part result

Table 20: General status messages in Analyze mode

Status message	Code	Description, suggestions
OK	0	Not an error
Localization module		

Status message	Code	Description, suggestions
Not enough points in the used search volume	11	The number of measurement points is too small. This may be caused by an incorrectly positioned search volume or an incorrectly positioned (virtual) lower plane. Adjust the settings under **Bin and search volume** in such a way that all the relevant 3D measurement points are located within the search volume.
No possible part is found	12	It is not possible to identify any part candidate. Make sure that the used image contains parts from the selected job, and that the parts are inside the used search volume. **For the Surface algorithm** Make sure the CAD model is correct in comparison with the parts in the scene. Adapting the parameter **Image data quality**, located in **Part** > **Localization**, can help in some cases. **For the Edge algorithm** Make sure that relevant edge and surface indicators are visible in the teach image by changing the **Edge threshold** or **Z plane** parameters located in **Part** > **Teach.**
Collision module Collision problems can generally be eliminated by configuring additional pick poses or adapting already configured pick poses.		
Pick poses are in a collision with 3D data	33	All the configured and enabled pick poses cause collisions. Check whether the thresholds for collisions can be increased. Adapt the parameter under **Pick poses** > **Collision settings.** Configure additional pick poses or modify the existing positions.

Table 21: Status messages for the Surface algorithm in Analyze mode

Status message	Code	Description, suggestions
Localization module		
Diverging part localization	14	The system cannot find a stable localization result. This may be due to measurement noise. Change the parameter in **Part** > **Localization** > **Filter ratio standard deviation.** Another possible cause is that the part is flat and has few edges. Such parts are more difficult to identify. Reduce the values under **Part** > **Part orientation in bin** so that the part can be located in fewer orientations. You can also adapt the parameter **Average distance to model** (under **Part** > **Localization**) and the parameter **Pseudo flat** (under **Part** > **Geometry**).
The part orientation in the bin is not OK	16	The located part is outside of the values specified in **Part** > **Part orientation in bin**. Adapt the values accordingly.

Status message	Code	Description, suggestions
A portion of points inside the model is too high	17	The ratio between points found within the part and the points found on the surface of the part is too high. This may be due to reflections that make it appear as if there are points inside the part. In this case, increase the value under **Part** > **Localization** > **Portion of points inside.** However, it is also possible that localization is too imprecise, e.g. the part is located in an incorrect rotation. In this case, you should not increase the **Portion of points inside** value. Instead check parameters that affect localization accuracy, such as **Average distance to model.**
Not enough of the part surface is visible to the camera	18	Too much of the part surface is obscured from the camera. If necessary, adapt the parameters under **Part** > **Localization** > **Surface visible to the camera.**
Overlap module		
The located part is overlapped	21	The located part is overlapped. Check whether the threshold for overlaps can be increased.
Collision module Collision problems can generally be eliminated by configuring additional pick poses or adapting already configured pick poses.		
All pick poses are located below the part	31	All configured and enabled pick poses are located below the part in its current orientation. Configure the pick poses for all possible orientations.
Pick poses are in a collision with the lower plane	32	All configured and enabled pick poses cause collisions with the (virtual) lower plane. If possible, move the lower plane or adapt the pick poses.

Table 22: Status messages for the Edge algorithm in Analyze mode

Status message	Code	Description, suggestions
Localization module		
Fine adjustment failed	1	No suitable candidate was found. Not enough data points/edges/surfaces in the volume of view.
Height/tilt adjustment failed	2	Make sure that there are parts in the bin, that they are identical to the reference part, and that the correct face is upwards. The algorithm works for parts with limited tilt (up to 15 degrees). If a part has asymmetric mirrored faces, make sure to use the correct job for localization.
Surface test failed (too many invalid surface probes)	3	The number of surface probes is too small, or there are too many incorrectly positioned surface probes. Make a new teach by adjusting the z-plane level and/or the region marking.
Edge test failed (unstable convergence).	4	There are incorrectly positioned edge probes, so the algorithm is unable to find a match. Try adjusting the edge threshold value.
No part is within thresholds	13	It is not possible to detect a part that is within the defined thresholds. Change the threshold by adapting the **Score** value.

9 Maintenance

9.1 Cleaning and maintenance

At regular intervals, check the inspection window and the housing of the device for contamination (see table 23). This is especially relevant in harsh operating environments (dust, abrasion, dampness, fingerprints, etc.).

The inspection window must be kept clean and dry during operation.

Table 23: Maintenance schedule

Maintenance work	Interval	To be carried out by
Clean housing and front screen	Cleaning interval depends on ambient conditions and climate	Specialist
Check the screw connections and plug connections	Every 6 months	Specialist

NOTE

- The camera housing does not contain any user-serviceable parts. If the device is opened, the warranty will be deemed null and void.
- Exercise caution when cleaning the device. If the device position is changed during cleaning, the hand-eye alignment must be redone.
- See specific instructions for the particular device in the following sections.

NOTICE
Device damage due to improper cleaning!

Improper cleaning may result in damage to the device.

- Only use suitable cleaning agents.
- Never use sharp objects for cleaning.

Cleaning procedure

CAUTION
Laser and LED safety

See "Operational safety and particular hazards", page 11 for information regarding laser and LED safety.

► Switch off the device for the duration of the cleaning operation. If this is not possible, use suitable laser protection goggles. These must absorb radiation of the device's wavelength effectively.

Cleaning the inspection window

NOTICE
Damage to the inspection window!

Reduced analysis performance due to scratches or streaks on the inspection window!

► Use a mild cleaning agent that does not contain powder additives. Do not use aggressive cleaning agents, such as acetone, etc.
► Avoid any movements that could cause scratches or abrasions on the inspection window.
► Only use cleaning agents suitable for the screen material.

NOTE
If the inspection window is scratched or damaged (cracked or broken), the window must be replaced. Contact SICK Service to arrange this.

Cleaning the housing

To ensure that heat is adequately dissipated from the device, the housing surface must be kept clean.

► Clear the build-up of dust on the housing with a soft brush.

9.1.1 Ensenso cameras

Lenses, housings and glass windows should be cleaned with water, iso-propanol or ethyl alcohol under normal operating temperature conditions of the respective device.

Make sure to not use abrasive cleaning equipment or substances. Use only optical quality tissue/cloth for glass windows and lenses.

9.1.2 Visionary cameras

Clean the housing using a soft cloth. Either use a dry cloth, or dampen it with lukewarm water and a small amount of mild cleaning agent.

Check the screw connections and connectors regularly.

- Visionary-S: Clean the area between the cooling ribs regularly.
- Visionary-T Mini: Ensure adequate heat dissipation to guarantee the availability of the device in continuous operation.

9.1.3 Zivid camera

- Check screw connections and connectors at regular intervals.
- Do not block the air opening on the front and rear sides of the product.
- The unit uses active and passive cooling, allow some space around the device for airflow.
- Use a small vacuum cleaner or a small canister of compressed air to remove dust or other accumulated particles from the glasses or in between the ribs of the heat sink.
- Regularly clean optical glass parts of the device.

9.2 Manual data backup and restore

We recommend that you make backup copies of the job configuration each time you modify the system configuration.

All user-defined settings are saved in the Workspace folder. You can adapt the storage location for this folder in PLB Studio by choosing **File** > **Set workspace folder**. We recommend that you regularly backup the Workspace folder.

To backup your Workspace folder, proceed as follows:

1. Start PLB Engine and PLB Studio by clicking the desktop icon or by choosing **Start** > **Programs** > **SICK** > **PLB** > **PLB Engine**.
2. Select **File** > **Backup workspace folder**.
3. Choose the desired storage location.

✓ Workspace folder backed up.

Proceed as follows to restore the Workspace folder:

1. Start PLB Engine and PLB Studio by clicking the desktop icon or by choosing **Start** > **Programs** > **SICK** > **PLB** > **PLB Engine**.
2. Select **File** > **Restore workspace folder**.
3. Select which backup file to restore from.
4. Enter a name to the workspace folder.
5. Shut down PLB Engine via the Windows taskbar.
6. Restart PLB Engine and PLB Studio so that the changes take effect.

✓ Workspace folder restored.

You can also create backup copies of **selected** jobs. To do this, proceed as follows:

1. Go to your Workspace folder.
2. Open the job library here.
3. Zip the required job folder.

To restore the backed up job again, unpack the backed up job folder to the job library.

10 Troubleshooting

10.1 Identifying and correcting communication problems

You can use a suitable utility program to identify errors during communications between the robot and the system.

The following description relates to the freely available Hercules SETUP program which you can download from http://www.hw-group.com/products/hercules/index_en.html.

In the following, we assume that Hercules SETUP and the PLB system are running on one and the same computer.

10.1.1 Testing robot-to-system communication

To test robot-to-system communication, allow the utility program to act as robot or TCP client.

To do this, proceed as follows:

1. Make sure that PLB Studio is using the template file **RobotInterface_CSVTemplate.cfg** to communicate with the robot controller. You will find the corresponding setting under **System settings** > **Robot communication** > **Template file** (see figure 89, page 101).

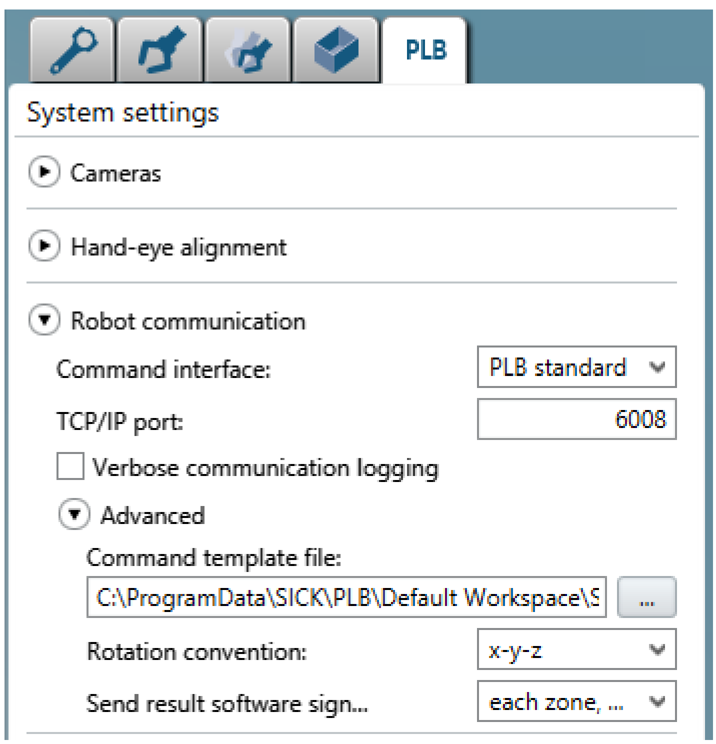


Figure 89: Specifying the template file for communication with the robot controller

2. Check the **Verbose communication logging** box so that all the commands and messages sent between the system and the robot controller are recorded in the system log. This simplifies troubleshooting if communication problems occur.
3. Start Hercules SETUP.
4. Open the **TCP Client** tab (see figure 90, page 102).
5. Enter IP address 127.0.0.1 (localhost) and port number 6008.
6. Click **Connect** and wait until the connection is established.
7. In one of the three Send fields, enter the command `SetState,2, ;`.
8. Click **Send**.

9. If the system has been configured correctly, it acknowledges the request with the message **State,0,2;**.

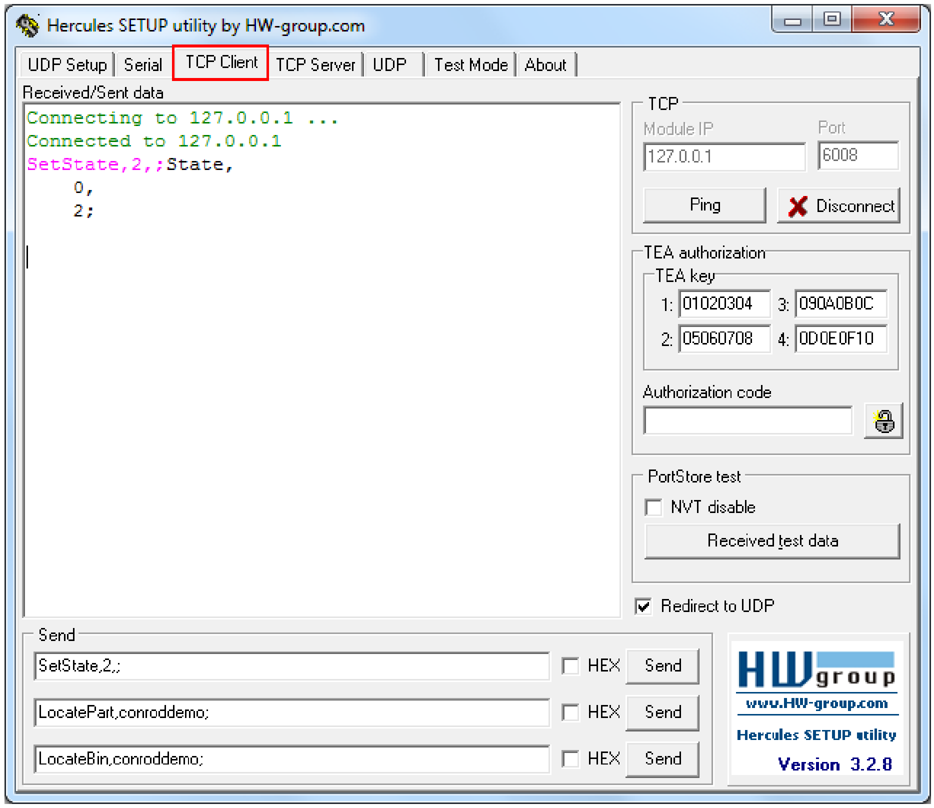


Figure 90: Testing robot-to-system communication

10.1.2 Testing system-to-robot communication

To test system-to-robot communication, allow the utility program to act as the PLB system or TCP server. Proceed as follows:

1. Make sure that the robot has no active connection on port 6008.
2. Start Hercules SETUP.
3. Open the **TCP-Server** tab (see figure 91, page 103).
4. Enter port number 6008.
5. Click **Listen**.
6. Connect the robot to the TCP server. Hercules SETUP reports that a connection has been established.
7. Send a message of your choice from the robot to the communication interface. Hercules SETUP receives and displays the command.
8. The message that is displayed should be identical to the message that was sent by the robot.

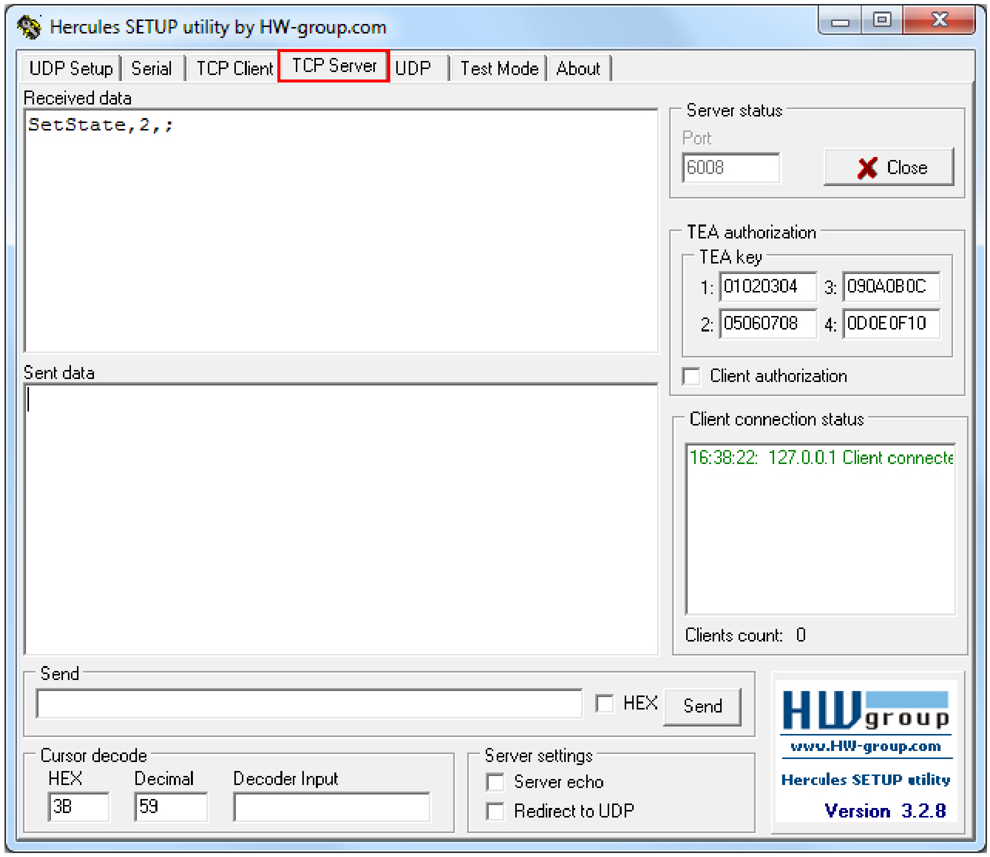


Figure 91: Testing system-to-robot communication

10.2 Error description

- For detailed information on error messages, see "Result codes", page 167.
- For detailed information on status messages, see "Suggestions on parameter changes depending on the status", page 95.

10.3 Causes of deadlock situations and countermeasures

The most frequent causes of deadlock situations and recommended countermeasures are listed in table 24.

A deadlock situation occurs in the following cases:

- The system is not able to locate parts.
- The part cannot be picked, for example, due to a collision with a bin wall or the presence of other overlapping parts.
- The robot arm or gripper collide with obstacles.
- The part is located outside of the robot's reach due to axis limitations or singularities.

NOTE

You should note that more restrictive configuration settings may make it impossible to locate parts while settings that permit greater tolerances increase the risk of incorrect localizations. As a result, you should preferably implement some of the remedies listed here.

If the measures listed here are not able to solve your problem and ensure trouble-free operation of your equipment, please contact the SICK subsidiary responsible for your country (see also "Service and support", page 105).

Table 24: Causes of deadlock situations and countermeasures

Cause of error	Recommended remedy(ies)
The system cannot locate a part even though it could be picked up without difficulty.	• Configure additional pick poses. • Use a second job for the same part, with different settings, allowing you to locate a part with less accuracy.
The system locates a number of parts but the robot is not able to pick them up.	• Modify the gripper. • Use additional grippers.
The system locates a number of parts but the robot is not able to reach them due to axis limitations.	If parts cannot be reached, it may help to place the robot on a platform in order to increase its reach.
Shiny parts, e.g. the flat side of a cylinder, cause interference due to secondary reflections.	Surface jobs: Increase the value **Portion of points inside** under **Part** > **Localization**.
Poor localization of parts whose surfaces cannot be exactly identified by the camera. This results in false overlap detection. This applies, for example, to relatively flat or steplike parts.	Surface jobs: • Try using the Edge algorithm instead. • If possible, increase the thresholds for overlaps. • You can increase localization accuracy by increasing the value **Length % (slice)** under **Part** > **Localization** > **Surface visible to camera**. This also reduces the risk of overlaps being falsely detected.
If the CAD model of the part differs greatly from its actual shape then this can cause localization or overlap detection to fail.	If the part cannot be located: • Increase the value **Average distance to model** under **Part** > **Localization**. • Increase the value **Portion of points inside** under **Part** > **Localization**. • Reduce the value under **FilterRatioStdv** under **Part** > **Localization**. If false overlaps are detected, you should reduce the thresholds for overlaps if possible. If only parts of the CAD model differ, you can avoid having to perform the overlap test by defining so-called exclusion boxes for the relevant areas.
Steep surfaces (as viewed by the camera) and extremely shiny material may be incorrectly interpreted as overlaps.	Minimize the extent of the overlap test by defining so-called exclusion boxes for the relevant areas.
The parts are too small.	Move the parts closer to the camera if your application scenario permits this.
Thin flat parts are not found by the edge algorithm.	Decrease the value **Edge threshold** under **Part** > **Teach**.

10.4 Measures to restore measurement accuracy

Temperature fluctuations and changes in air humidity may impair measurement accuracy.

Changes in air humidity may result in misting on the camera window. In this case, clean the camera window with a soft, dry cloth.

In the following cases, you should perform hand-eye alignment again. For detailed information, see "Overview hand-eye alignment", page 53.

- If the camera has been exposed to significant fluctuations in temperature since the last time hand-eye alignment was performed. A temperature difference of approximately 10 °C corresponds to an imprecision of 1 mm.
- If the position of the camera has changed, for example after a collision, you should align the unit again in order to make sure that the bin is still in the camera's field of view.
- If the position of the robot has been moved.

You can test if a new alignment is needed by performing a verify alignment procedure. For a robot program flow chart see figure 127, page 176.

If you have reason to believe that the camera is damaged, contact us directly. You will find contact information in "Service and support", page 105.

10.5 Service and support

SICK possesses a global network of subsidiaries and service engineers that is able to provide you with the support you need quickly and efficiently. Our experts are available to help and advise you.

10.5.1 SICK support portal

Generate a ZIP file with information about the system, the active job, and a captured image from PLB Studio. Create a support ticket with information about the problem and a SICK representative will contact you.

For help and support:

1. In **Analyze** mode, in the **Analyze** tab, select a captured image.
2. Click **Export active job** .
3. Save the generated ZIP file.
4. Go to **supportportal.sick.com**.
5. Log in or register an account.
6. Create a support ticket, and append the generated ZIP file.

10.5.2 SICK's PLB optimization service

You can use PLB Studio for analysis of given application scenarios and find suitable parameters.

For long-term operation, however, you need stable parameter settings that cover all potential changes to the application scenario (e.g. bin filling level, bin position) as well the environmental conditions.

SICK offers a comprehensive optimization service for your PLB system. We need approximately 200 recorded images that are representative of your application scenarios to help optimize your system.

Acquire images from a full and an almost empty bin as well as of various filling levels in-between. Acquire images in which the parts are in the middle and at the edges of the bin.

If the parts are to be evaluated for the edge algorithm then there needs to be some images with an isolated part in default tilt. This is needed to be able to teach the part reference.

SICK possesses a worldwide network of subsidiaries and service engineers. If you want to make use of the PLB optimization service, go to **supportportal.sick.com** and register an account.

10.6 Repairs

Repair work on the device may only be performed by qualified and authorized personnel from SICK AG. Interruptions or modifications to the device by the customer will invalidate any warranty claims against SICK AG.

10.7 Returns

- ► Only send in devices after consulting with SICK Service.
- ► The device must be sent in the original packaging or an equivalent padded packaging.

NOTE
Optional memory card

- Check whether there is a memory card in the card slot of the device. If yes, remove the memory card from the faulty device in **de-energized state**.
- Do not send in the memory card!

NOTE
To enable efficient processing and allow us to determine the cause quickly, please include the following when making a return:

- Details of the contact person
- Description of the application
- Description of the fault that occurred

10.8 Disposal

CAUTION
Risk of injury due to hot device surface!
The surface of the device can become hot during operation.

- ► Before commencing disassembly, switch off the device and allow it to cool down as necessary.

Any device which can no longer be used at the end of the product life cycle must be disposed of in an environmentally friendly manner in accordance with the respective applicable country-specific waste disposal regulations. As they are categorized as electronic waste, the device must never be disposed of with household waste.

NOTICE
Danger to the environment due to improper disposal of the device.
Disposing of devices improperly may cause damage to the environment.
Therefore, observe the following information:

- Always observe the national regulations on environmental protection.
- Separate the recyclable materials by type and place them in recycling containers.

---

11 Technical data

11.1 System characteristics

Table 25: System characteristics

System type	Robot guidance
Applications	Localization of parts in bin and search volume
System features	3D camera and software package for part localization
Camera	3D snapshot
Volume of view	See "Volume of view", page 22.
Working distance	See "Volume of view", page 22.
Light source	• Ensenso X36, N36, C57 (PLB521/522): LED, blue light, 465 nm • Visionary-S: Laser, infrared, invisible, 808 nm • Visionary-T Mini: Laser, infrared, invisible, 855 nm, ± 5 nm • Zivid 2+: LED, white light
Laser class	• Visionary-S, Visionary-T Mini: 1
LED risk group	• Ensenso X36, C57 (PLB521/522): Risk group 1 • Ensenso N36: Exempt group (no risk) • Zivid 2+: Risk group 2
3D CAD format	• STEP • IGES
Localization principle	• Surface algorithm: Matching of 3D shape between CAD model and 3D image • Edge algorithm: Matching of 3D shape between taught reference and 3D image. • Plug-in algorithm
Output data	x, y, z (mm or inch), roll, pitch, yaw (degrees)

11.2 Interfaces

Table 26: Interfaces

Communication	2 Ethernet ports (camera to PC and PC to robot controller)
Data transfer rate (Ethernet)	• 1 Gbit/s (camera to PC) • 10/100 Mbit/s (PC to robot controller)
Protocol (Ethernet)	• UDP/IP (camera) • TCP/IP (robot controller)

11.3 Electrical and mechanical components

Ensenso X36 5 MP

Table 27: Mechanical and electrical components of Ensenso X36 5 MP

Connections	• Power: M12, 4-pin male connector • Ethernet: M12, 8-pin connector, X-coded
Supply voltage	24 V DC, internally fused[1]
Housing material	Aluminum
Housing color	Gray
Dimensions (L x W x H)	488 mm x 200 mm x 80 mm

Weight	2500 g
Enclosure rating	IP54 (when assembled in the SICK housing)

1) Internal slow-blow fuse 3.5 A

Ensenso N36

Table 28: Mechanical and electrical components of Ensenso N36

Connections	• Power: Binder 712, 7-pin, male connector ○ Alternatively, use a PoE injector for power supply • Ethernet: RJ45
Supply voltage	12 ... 24 V DC
Housing color	Blue
Dimensions (L x W x H)	175 mm x 52 mm x 50 mm
Weight	650 g
Enclosure rating	IP65/IP67

Ensenso C57 (PLB521)

Table 29: Mechanical and electrical components of Ensenso C57 (PLB521)

Connections	• Power: M8, 4-pin male connector • Ethernet: M12, 8-pin connector, X-coded
Supply voltage	24 V DC
Housing material	Aluminum
Housing color	Black
Dimensions (L x W x H)	506 x 58 x 100 mm
Weight	1660 g
Enclosure rating	IP65/IP67

Ensenso C57 (PLB522)

Table 30: Mechanical and electrical components of Ensenso C57 (PLB522)

Connections	• Power: M8, 4-pin male connector • Ethernet: M12, 8-pin connector, X-coded
Supply voltage	24 V DC
Housing material	Aluminum
Housing color	Black
Dimensions (L x W x H)	290 x 58 x 100 mm
Weight	1160 g
Enclosure rating	IP65/IP67

Visionary-S

Table 31: Mechanical and electrical components for Visionary-S

Connection	• Power: M12, 17-pin male connector • Ethernet: M12, 8-pin connector, X-coded
Supply voltage	24 V DC ± 15 %
Housing color	Blue, black
Dimensions (L x W x H)	162 mm x 93 mm x 78 mm
Weight	1700 g
Enclosure rating	IP67

Visionary-T Mini

Table 32: Mechanical and electrical components for Visionary-T Mini

Connection	• Power: M12, 8-pin connector, A-coded • Ethernet: M12, 8-pin connector, X-coded
Supply voltage	24 V DC (–30% ... +25%)
Housing color	Blue, black
Dimensions (L x W x H)	80 mm x 70 mm x 77 mm
Weight	520 g
Enclosure rating	IP65/IP67

Zivid 2+

Table 33: Mechanical and electrical components of Zivid 2+

Connections	• Power: M12, 5-pin connector • Ethernet: M12, 8-pin connector, X-coded
Supply voltage	24 V DC (+10% / -20%)
Housing color	Black
Dimensions (L x W x H)	169 mm x 124 mm x 56 mm
Weight	1000 g
Enclosure rating	IP65

11.4 Ambient data

Table 34: Ambient data

Ambient temperature	Operation • 0 ... 40 °C, non-condensing • **Visionary-T Mini:**–10 ... +50 °C[1)] Storage • **Ensenso X36, N36:** 0 ... 50 °C, non-condensing • **Ensenso C57 (PLB521/522):** -10 ... +50 °C, non-condensing • **Visionary-S**: –20 ... +70 °C, non-condensing • **Visionary-T Mini**: –20 ... +80 °C non-condensing • **Zivid 2+**: -20 ... 60 °C, non-condensing
Shock load[2)]	Ensenso X36, N36 • 80 g/1.9 ms • 25 g/6 ms Ensenso C57 (PLB521/522) • 60 g/1.9 ms • 25 g/6 ms Visionary-S, Zivid 2+ • 15 g/11 ms Visionary-T Mini • 30 g/11 ms

Vibration load[2]	**Ensenso X36, N36** • 10 g, 30 ... 500 Hz **Ensenso C57 (PLB521/522)** • 5 g, 30 ... 500 Hz **Visionary-S, Zivid 2+** • 5 g, 10 ... 150 Hz **Visionary-T Mini** • 5 g, 10 ... 500 Hz

1) Housing operating temperature –10 °C ... +65 °C
2) Not during scanning

11.5 Recommended PC specifications

General information

PLB must run on a high-performance desktop or industrial PC.

Do not run PLB on a laptop or a processor optimized for low energy consumption. Make sure that the CPU is specified for at least 65W TDP. Energy-saving functions will cause degraded performance (i.e. longer processing times, etc.) and may also cause other issues, such as image capture failures, etc.

In general, PLB performance will benefit from higher CPU frequency as well as higher CPU core count.

Table 35: Recommended PC specification

Operating system	Windows 10
CPU	• Intel Core i7/i9 or Intel Xeon • ≥ 3 GHz base frequency • ≥ 6 cores E.g. Intel Core-i7 10700
RAM	≥ 16 GB DDR4
Graphics card	**Ensenso X36 5 MP[1]** • NVIDIA GTX 1660 Super (or better NVIDIA GPU, with CUDA and Compute Capability 3.0 support) • ≥ 4 GB memory **Zivid 2+** • NVIDIA GeForce GTX 1060 or better • ≥ 3 GB memory • Integrated GPU is supported (Intel i7 with HD630 or better), but a dedicated GPU gives the best performance **All other cameras[1]** • NVIDIA GeForce GT610 1 GB or higher • OpenGL ES 2.0 support
Screen resolution	≥ 1920x1080 pixels
Hard disk[2]	• SSD system disk ≥ 512 GB (for OS, PLB installation, recording, and data storage)
Network interface card for PC-to-robot communication	Network interface card with Gigabit Ethernet interface. **Ensenso X36 5 MP** • ≥ 1.5 Gigabit network interface card

Network interface card for PC-to-camera communication	Make sure to replace the automatically installed driver with a driver from Intel that supports increasing the number of receive buffers, etc. See "Activating network and camera drivers", page 38. **Ensenso X36 5 MP** • Intel Ethernet Server Adapter I350-T2 **Zivid 2+** • Intel 10 GB Ethernet Converged Network Adapter X550-T2 **All other cameras** • Intel Gigabit CT Desktop Adapter

1) Ensenso cameras can enable GPU acceleration in the PLB Studio menu **Options**, for a more efficient 3D stereo image calculation. GPU acceleration requires a NVIDIA GPU with CUDA and Compute Capability 3.0 support, with at least 8 GB memory.

2) You can store the PLB program data, the workspace files, and the recording files on different drives.

12 Accessories

All accessories for the product can be found at www.sick.com/PLB.

13 Appendix

13.1 Underlying concepts

This chapter explains the various concepts that are essential for understanding system configuration. This chapter consists of three main sections detailing general concepts, as well as concepts specific to the Surface algorithm or Edge algorithm.

The corresponding configuration settings make it possible to optimize the system's parts identification and localization performance.

13.1.1 Algorithms

There are two algorithms available for the PLB system: the Surface algorithm and the Edge algorithm. In addition to this, it is possible to use plug-in algorithms.

When to choose which algorithm depends on the part and job. In many situations, one algorithm is clearly preferable. In other situations, it is beneficial to use different algorithms for different stages of a task.

13.1.1.1 Surface algorithm

The Surface algorithm:

- uses a CAD model as a reference and locates parts by looking for surfaces
- works best with voluminous parts that have visible surfaces facing different directions
- finds parts regardless of the orientation of the parts

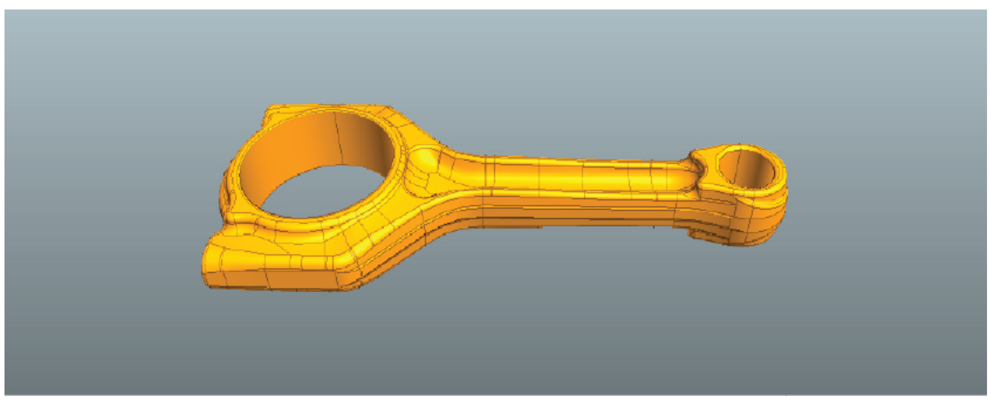


Figure 92: Example part suitable for the surface algorithm

13.1.1.2 Edge algorithm

The edge algorithm uses an image of the part as reference, and locates parts by looking for edges, preferably on a part with several distinct and visible edges at different angles.

The edge algorithm performs best when parts in the bin have a limited tilt compared to the part in the reference image.

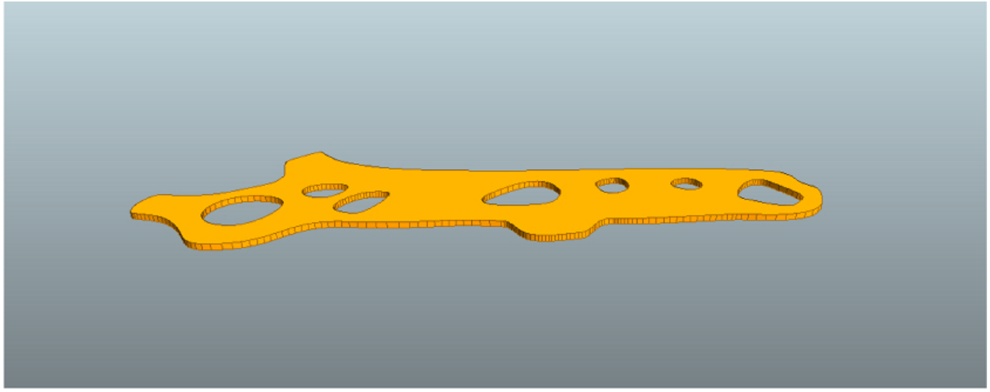


Figure 93: Example part suitable for the edge algorithm

13.1.1.3 Plug-in algorithms

It is possible to install custom-made algorithms that are configured and used similar to the surface and edge algorithms.

13.1.2 General concepts

This section explains concepts that are relevant for all algorithms.

13.1.2.1 Volume of view and search volume

Volume of view designates the camera's three-dimensional area of vision. The search volume is a volume within the volume of view that contains the measurement data that is actually analyzed by the system.

Specifying the search volume restricts the quantity of measurement data that is to be analyzed and consequently speeds up localization. The search volume is defined in the job configuration.

You will find detailed information on defining the search volume in "Configuring the bin and search volume", page 75 and "Running the job configuration", page 83.

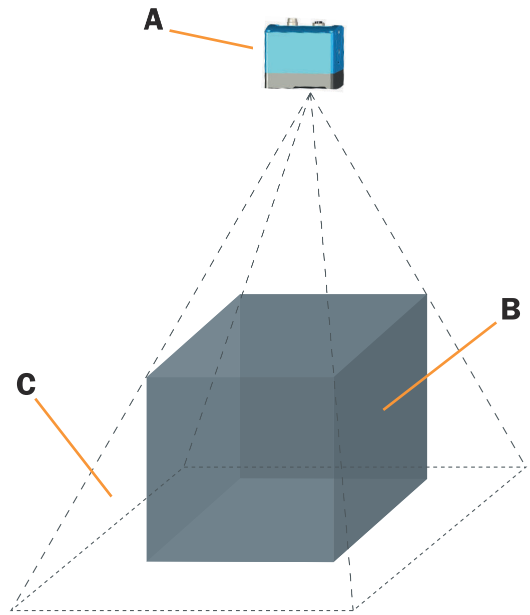


Figure 94: Volume of view and search volume

A Camera
B Search volume (as defined in the job configuration)
C Volume of view

13.1.2.2 Blacklist

When the blacklist function is activated, located parts are added to the blacklist. When a part which is already in the blacklist is located, it will be discarded by PLB until the blacklist is cleared. It is recommended to clear the blacklist when switching bin if the blacklist is activated.

Use the blacklist if there are constraints unknown to PLB, which prevent the robot from picking located parts, for example robot reach problems.

For more information about the blacklist, see "Localization", page 137.

13.1.2.3 Zones

You can subdivide the bin from which the robot removes the parts into zones. The advantage of this approach is that the system can determine the position of several different parts from one image acquisition.

In order to ensure that the fill level of the bin remains relatively even, the system internally determines the sequence in which it processes the zones.

You should note that although this can increase the relative processing time, it does not do so in direct proportion to the number of defined zones. The following examples are intended to illustrate this behavior:

- If you create four zones then the calculation of the results from all four zones takes longer time than if you were only using one zone and there was therefore only one result to calculate.
- If you create four zones then the calculation of the results from all four zones takes less time than if you were to use a single zone for four different images because, in this case, it would be necessary to acquire four images.

In order to reduce the overall cycle time, we recommend that you work with zones in the following cases:

- To avoid system deadlocks: To locate a reachable part, the system uses exactly the same computational power for each zone as for the entire search volume. If the system is not able to locate a reachable part in the search volume as a whole then it is more likely that localization will be successful for at least one of the defined zones.
- For the localization of multiple potentially reachable parts: If your robot controller performs additional calculations in order to determine whether or not a part is reachable (e.g. during the path planning or collision detection process) then it is possible to output a list of potentially reachable parts.

To reduce the cycle time: If it is possible to ensure that the parts are not moving then all the located parts can be picked up one after the other before a new image acquisition is started.

13.1.2.4 Reference frames

Reference frames are used when reporting part position and pick poses.

13.1.2.4.1 Reference frames of parts and grippers

The system outputs the position of the part reference frame, and the gripper position relative to the part reference frame.

When CAD files are used to set the reference (for grippers and for surface parts), you can align the reference frame. When you do this, the transformation from the original to the newly aligned reference frame is calculated.

It is recommended to align the part reference frame in the following situations:

- If this makes it easier to reach a specific position on the part.
- If the selected parts class (see "Parts classes", page 120) requires the part's coordinate axes to have a particular orientation.
- If the part has symmetries/pseudo-symmetries that have to be taken into account.

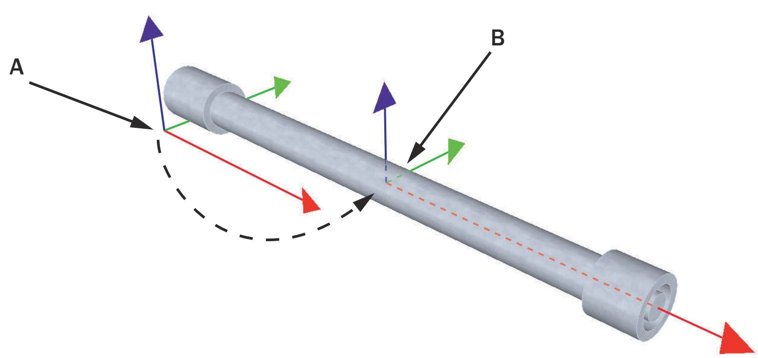


Figure 95: Alignment of the part reference frame (example)

A Part reference frame defined in the CAD model

B Aligned part reference frame

It is necessary to adapt the gripper reference frame in order to harmonize the position of the gripper with the TCP (Tool Center Point) defined at the robot (see figure 96, page 117).

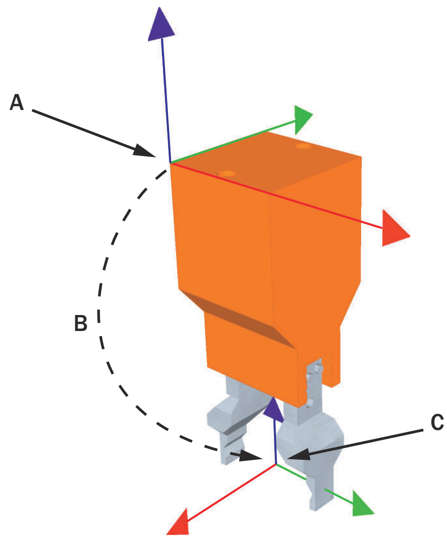


Figure 96: Alignment of the robot gripper's reference frame (example)

A Gripper reference frame defined in the CAD model

B Transformation used to align the gripper reference frame

C Aligned gripper reference frame (= position of the robot's TCP)

13.1.2.4.2 Pick pose

The pick pose is the position of the gripper in the part reference frame or the position and orientation that the gripper must assume in order to pick up the part without collision. You will find detailed information on configuring the pick pose in "Configuring pick poses", page 69.

The pick pose is determined on the basis of the aligned part reference frame as well as of the aligned gripper reference frame, see figure 97, page 117.

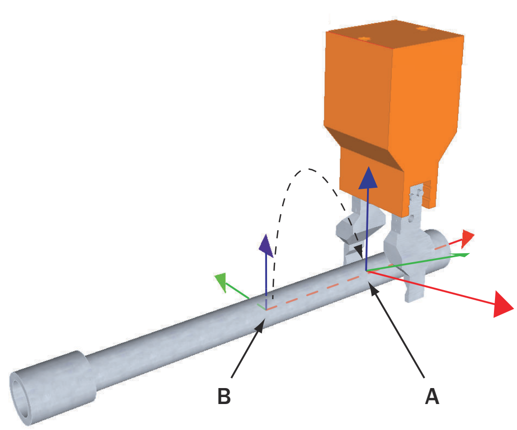


Figure 97: Determining the pick pose

A Aligned gripper reference frame

B Aligned part reference frame

For each pick pose, PLB can return four additional absolute positions for the robot to use when picking parts. They are typically used in the following order:

1. The Entry pose places the gripper in the middle and at a certain distance above the bin top. The gripper is neutrally aligned with the bin.
2. The Approach pose places the gripper in the same orientation as the pick pose, but at a certain distance away from the part (the direction of the distance is specified in the gripper configuration). Along with the Pick pose, collision tests with 3D data are performed for this pose.
3. The Clear pose places the gripper in the same orientation as the pick pose, but at a certain distance in the positive Z direction of the bin. This pose extracts parts along the bin’s orientation.
4. The Exit pose places the gripper in the middle and at a certain distance above the bin top. The gripper is neutrally aligned with the bin.

WARNING
For the Entry, Clear and Exit poses it cannot be guaranteed that the gripper does not collide with 3D data.

13.1.2.4.3 Pick pose results in relativeToolFrame or absoluteToolFrame

The system can output the results to the robot controller in two ways:

- **For a less complex robot program**, use `absoluteToolFrame`.
  - Pick pose of the part in the robot reference frame (`absoluteToolFrame`). The pick pose is calculated as follows:

    `absoluteToolFrame` = `partFrame` × `relativeToolFrame`

- **If flexibility is needed in the robot program**, use `relativeToolFrame`.
  - Position of the part in the robot reference frame (`partFrame`).
  - Pick pose of the part in the part reference frame (`relativeToolFrame`).

NOTE
Tip to integrators on the use of the `relativeToolFrame`.

To pick parts, send the robot to the `absoluteToolFrame` position using the tool defined in PLB. To pile the parts after picking them, a multitude of pick poses created in PLB will generate the need for the same multitude of drop poses in the robot program. Using the `relativeToolFrame`, we know how the part is gripped and can use this information to let the robot automatically calculate the drop position based on the pick pose.

13.1.2.4.4 Waypoints

The gripping motion to and from the pick pose can be controlled by waypoints.

Entry pose

The entry pose is the starting position before moving into the bin. The gripper is centered above the top of the search volume, with rotation and tilt adapted to the located part and the pick pose.

The default setting is to start the gripping motion with the gripper's tool center point positioned 100 mm above the top of the search volume.

Approach pose

The approach pose is the position when closing in for the pick. The gripper is positioned at a short distance from the part in the approach direction of the pick pose, with rotation and tilt adapted to the located part and the pick pose.

The default settings are:

- To position the gripper's tool center point 100 mm from the located part in the approach direction.
- To check if the gripper can move safely from the approach pose to the pick pose without collision.

Clear pose

The clear pose is the position after lifting up the part and it is positioned directly above the pick pose. The clear pose is often different from the approach pose. The part is lifted upward in the search volume Z direction from the pick pose to the clear pose. The gripper holds the part in the same rotation and tilt as in the pick pose.

The default settings are:

- To move the gripper's tool center point 100 mm upward after gripping the part.
- To check if the part is overlapped and can move safely from the pick pose to the clear pose without collision (only applies to some algorithms, such as the Surface algorithm).

Exit pose

The exit pose is the position after moving out of the bin. The gripper is centered above the top of the search volume, holding the part in the same rotation and tilt as in the pick pose.

The default setting is to end the pick with the gripper's tool center point 100 mm above the top of the search volume.

Robot frame

The system can output the waypoints in the robot reference frame via the robot interface:

- Entry pose: `absoluteEntryToolFrame`
- Approach pose: `absoluteApproachToolFrame`
- Clear pose: `absoluteClearToolFrame`
- Exit pose: `absoluteExitToolFrame`

13.1.2.5 Collision

Collision detection is performed to prevent the gripper from colliding with obstacles. Collision detection is based on the CAD model of the robot gripper in combination with the specified pick pose.

NOTE
Please note that only collisions between the CAD model of the gripper and 3D measurement points (e.g. of other parts, the bin walls) can be detected. It is not possible to perform collision detection for objects that were not specified during job configuration (e.g. robot arm, cabling).

The following limitations apply to collision detection for surface, edge and plug-in algorithms:

- Collisions can only be detected for gripper movement into the bin (pick-up movement into the bin) but not for the movement of the gripper out of the bin (removal of the part from the bin).
- Collisions can only be detected from the approach pose waypoint to the pick-up point.
- Collision detection along the approach direction is only performed for 3D measurement points that are located within the gripper's CAD model.

The following limitations apply to collision detection for the surface algorithm only:

- A check for overlaps by 3D measurement points located above the gripper is performed only at the pick-up point itself.
- Even if you disable collision detection, collisions are still detected for pick poses that are located below the part.

13.1.3 Surface algorithm concepts

This section explains concepts that are relevant for the Surface algorithm.

13.1.3.1 Parts classes

To make it possible to identify and locate the parts that are to be picked up, they must first be assigned to a given class.

To assign parts to classes, you must possess corresponding CAD models of the parts and import these into the configuration program. For detailed information, see "Creating a new job", page 61.

You can choose between the following parts classes:

- Limited tilt
- Revolving
- Revolving pick pose
- Beam
- No constraints

The crucial element when classifying a part is its shape. However, it is also necessary to take account of the part's orientation or position in the bin (roughly ordered or completely random).

13.1.3.1.1 Limited tilt

Parts that are comparatively well ordered and lie flat in the bin should be assigned to the "Limited tilt" class. They should not be tilted from the horizontal by more than 30°.

The parts themselves do not have to be flat but should possess characteristic surface features that can be easily recognized from above.

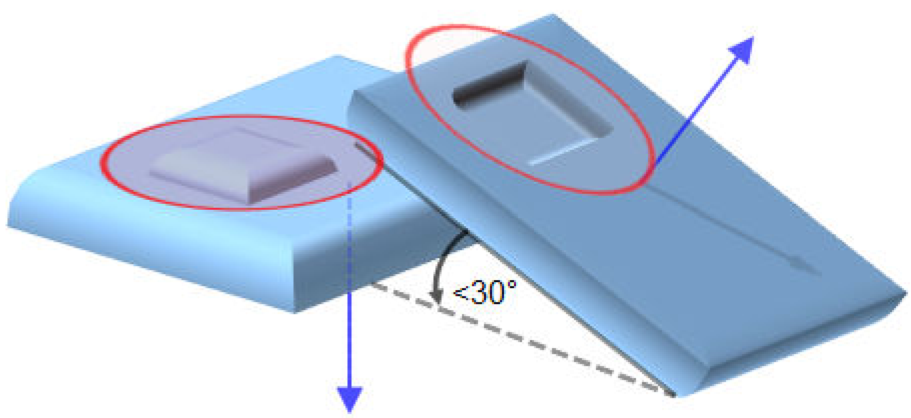


Figure 98: Requirement for assignment to the "Limited tilt" class

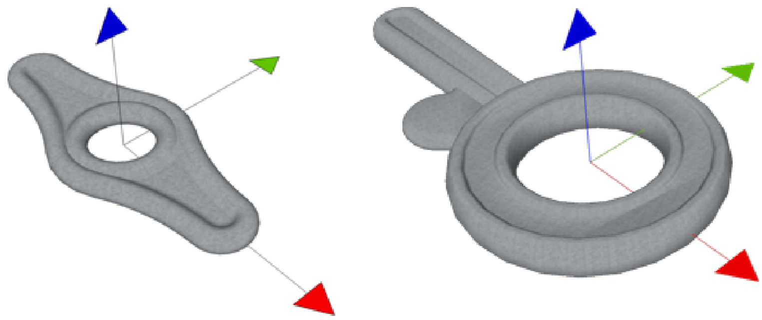


Figure 99: Typical example of the "Limited tilt" class

13.1.3.1.2 Revolving

Use the "Revolving" class for parts with precise rotational symmetry around an axis, e.g. lathed parts.

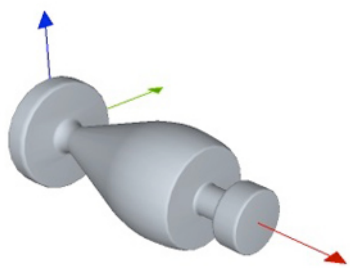


Figure 100: Example of the "Revolving" class, rotational symmetry around the X-axis

Requirements

- The length of the part must be at least twice its diameter (L > 2D).
- The part must be rotationally symmetrical around the X-axis.
- The part must lie relatively flat in the bin. It should not be tilted from the horizontal by more than 45°.

13.1.3.1.3 Revolving pick pose

Use the "Revolving pick pose" class for parts with a cylindrical section suitable for pickup, e.g. crankshafts.

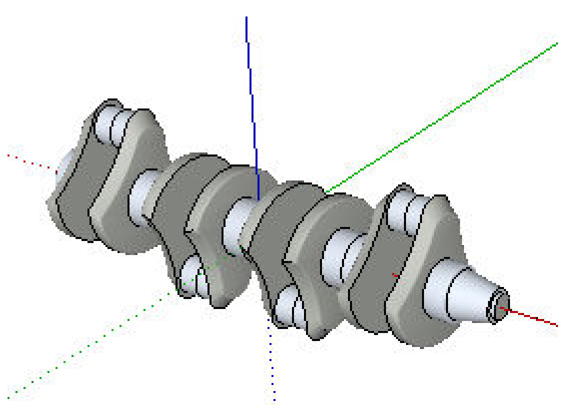


Figure 101: Typical example of the "Revolving pick pose" class, pick pose at a cylindrical section

Requirements

- The part's X-axis must coincide with the cylindrical section at which the part has to be picked up.
- The part must lie relatively flat in the bin. It should not be tilted from the horizontal by more than 20°.

NOTE
If a part is to be picked up at a cylindrical section then you can perform a so-called vertical correction for the corresponding pick pose. For detailed information, see "Vertical correction", page 124.

13.1.3.1.4 Beam

Use the "Beam" class for long parts that are not rotational symmetrical about the x-axis.

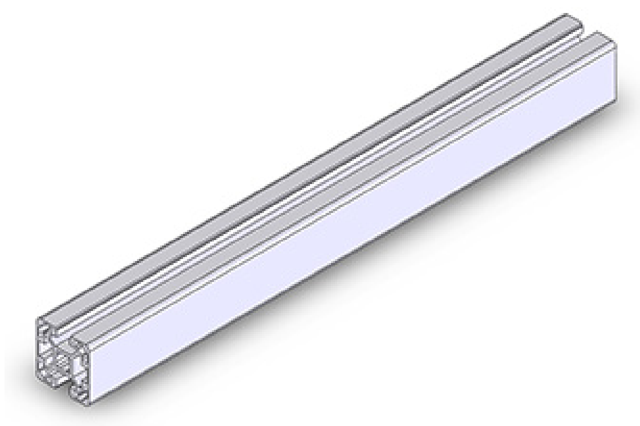


Figure 102: Typical example of the "Beam" class

Requirements

- The length of the part must be at least twice its diameter (L > 2D).
- The CAD file of the part must be positioned with the long side along the x-axis.
- The part can be rotated about x and z, but must not stand up in the bin. It should not be rotated about the y axis by more than 20°.

13.1.3.1.5 No constraints

There are no restrictions placed on the shape or orientation of parts that are assigned to the "No Constraints" class. This class is suitable for parts that are completely randomly oriented in the bin.

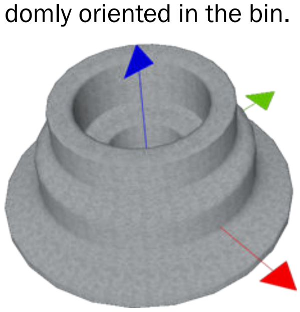


Figure 103: Typical example of the "No Constraints" class

13.1.3.1.6 Recommended procedure when classifying parts

The diagram below will help you when assigning parts to a given class.

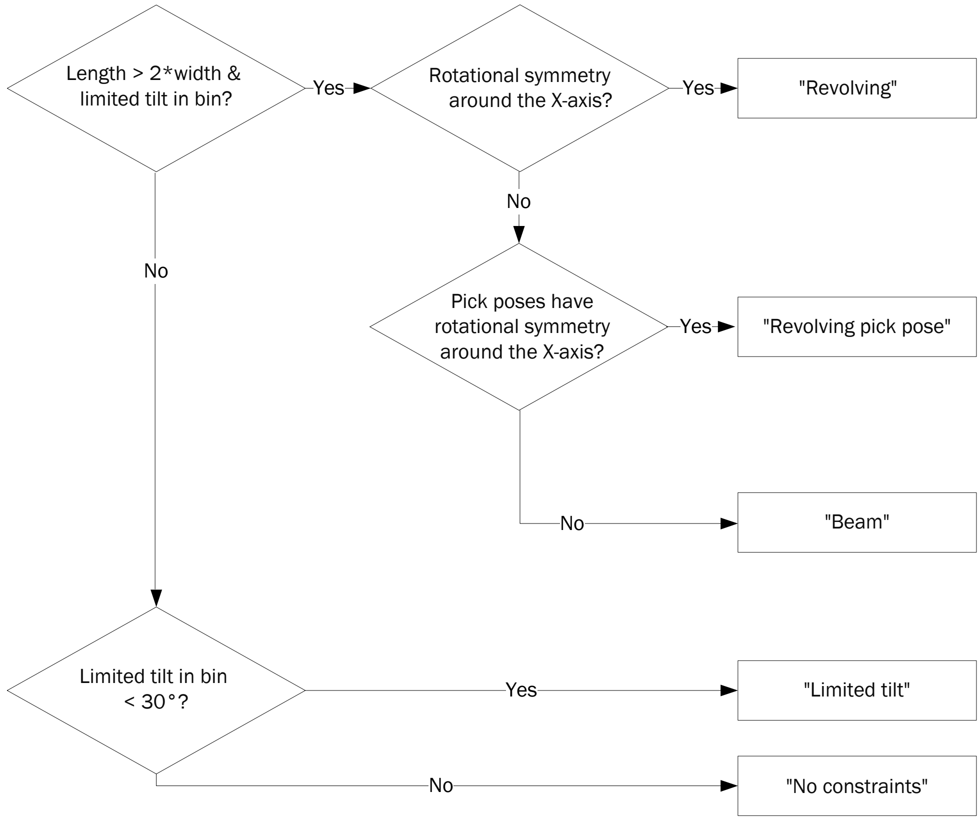


Figure 104: Decision schema

13.1.3.2 Overlap

The term overlap refers to the presence of 3D measurement points above the located part. The measurement points may belong to another part or to the walls of the bin, for example.

On the basis of the top view of the part's CAD model, the system checks whether the part is overlapped. Any holes that may be present in a part are disregarded in connection with overlapping.

figure 105 and figure 106 illustrate the area of the partial surface that is taken into account when calculating the overlap in two different situations. In both cases, the part on the left is overlapped by the part on the right.

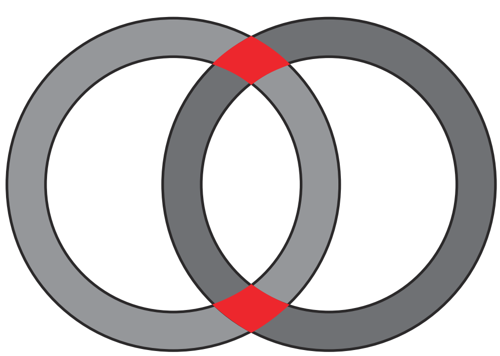


Figure 105: Overlapping of a hollow part

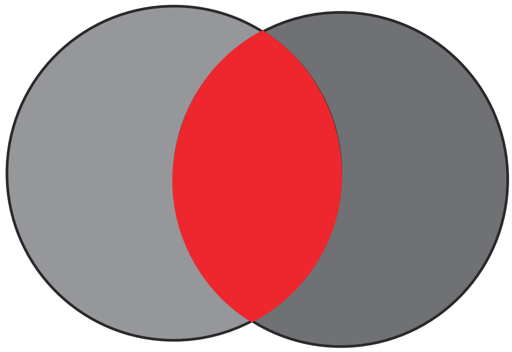


Figure 106: Overlapping of a solid part

You can specify tolerance values during job configuration (see "Configuring pick poses", page 69).

13.1.3.3 Vertical correction

If the part has pick areas with rotational symmetry around the X-axis, you can perform a so-called vertical correction of the pick pose.

Vertical correction is expressed as a rotation around the X-axis in the part reference frame: When the part has been located, its rotation around the X-axis is automatically set to 0°. A check is then performed to determine whether the part can be picked up without collision in the defined pick pose.

To use vertical correction for individual pick poses, you must enable the **Dynamic pick poses** option **Cylindrical around x-axis at pick pose.**

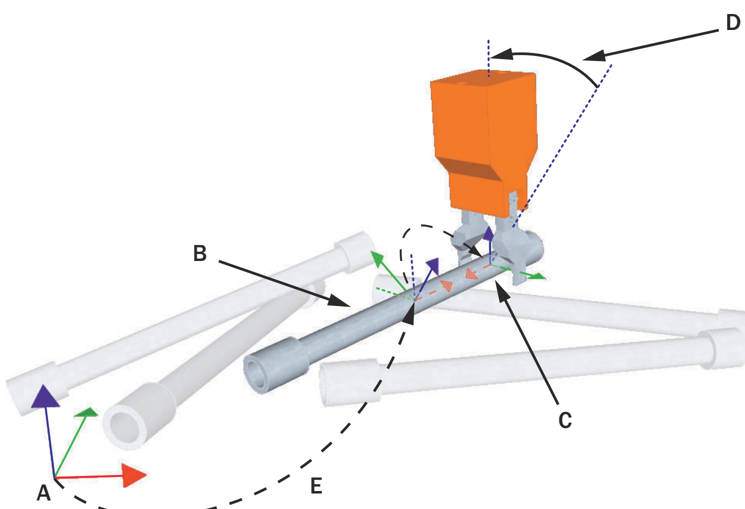


Figure 107: Vertical correction of pick pose

A	Robot reference frame
B	Part reference frame as defined in the PLB system
C	Position of the robot's TCP in the pick pose
D	Vertical correction of pick pose
E	Position of the part in the robot reference frame

However, performing a vertical correction does not mean that the part will only be picked up from above. The following example makes this clear (see figure 108, page 125):

Assume that the pick pose is defined in such a way that the gripper picks up the part from an angle of 30° relative to the part's X-axis. Assume also that the part is rotated through 45° around its X-axis in the bin. In this case, although the calculated pick pose compensates for the 45° rotation of the part, the desired 30° angle of the gripper relative to the X-axis of the part is nevertheless retained.

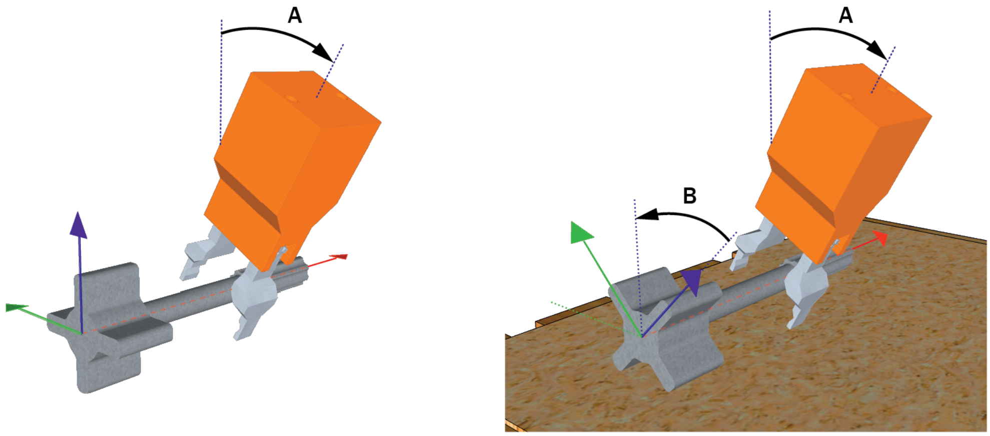


Figure 108: Left: Configuration, right: Operation

A Angle of inclination of gripper (here: 30°)

B Vertical correction of pick pose

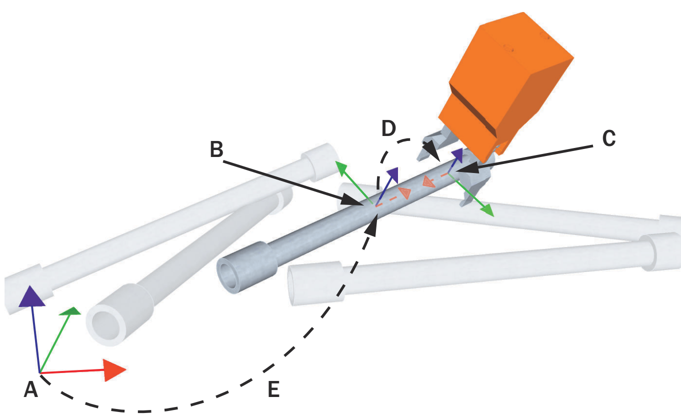


Figure 109: Relationship between the robot reference frame and the reference frames that are output by the system

A Robot reference frame

B Part reference frame as defined in the PLB system

C Position of the robot's TCP in the pick pose

D Pick pose in the part reference frame

E Position of the part in the robot reference frame

13.1.3.4 Symmetry and pseudo-symmetry

System performance can be optimized as follows by taking account of any symmetries or pseudo-symmetries that may be present:

- Specifying a part's symmetry accelerates computation because the system does not also have to inspect the part for symmetries after it has been located.
- Specifying a part's pseudo-symmetry optimizes localization accuracy because the system makes sure that the part has the correct orientation before it is picked up.

A part is symmetrical if rotating it around one of its axes causes it to remain identical in at least two positions (e.g. following a 360° rotation around the X-axis, a 180° rotation around the Y axis, etc.). (see figure 110, page 126)

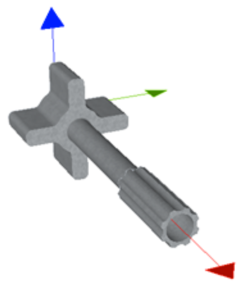


Figure 110: Example of a symmetrical part (4-fold symmetry around the X-axis)

A part is pseudo-symmetrical if it appears to be rotationally symmetrical around a given axis but has to be picked up at a specific position due to a deviation in its shape. (see figure 111, page 126)

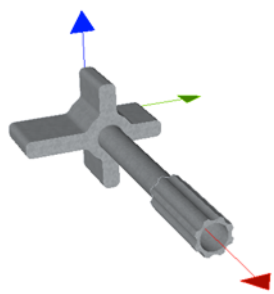


Figure 111: Example of a pseudo-symmetrical part (4-fold pseudo-symmetry around the X-axis)

13.1.3.5 Inclusion and exclusion boxes

To accelerate the calculation process, you can mark the areas of deviation in pseudo-symmetric parts by defining a volume, or box, around them. Here, we distinguish between so-called inclusions and exclusion boxes.

Boxes defined as inclusions are taken into account when determining pseudo-symmetry whereas boxes defined as exclusions are not.

It may therefore be useful to define a deviation in the surface of a part caused by productionrelated inaccuracies (e.g. casting errors) as an exclusion box.

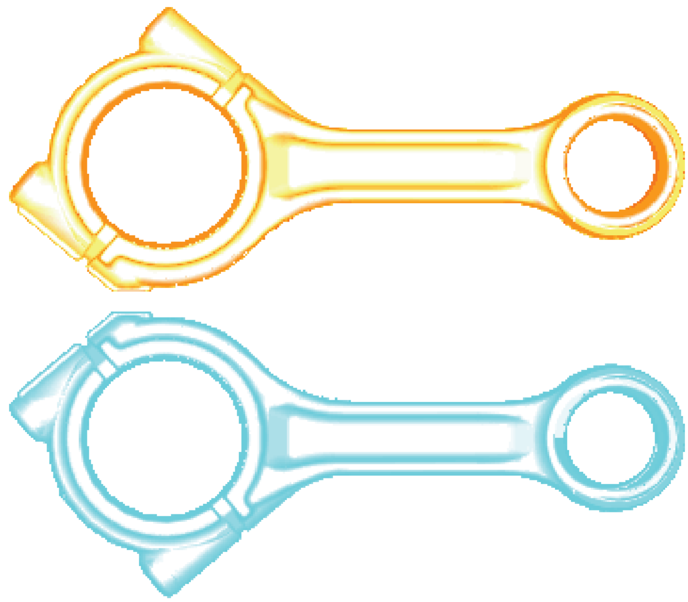


Figure 112: Pseudo-symmetric part (top) rotated around the axis of rotation (bottom)

A box has already been defined as an inclusion for the part depicted in figure 113 (highlighted in purple).

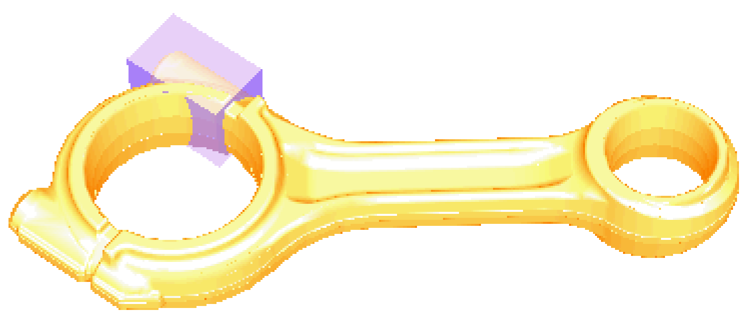


Figure 113: Box defined as an inclusion

figure 114 and figure 115 illustrate examples of pseudo-symmetric parts each of which has a defined inclusion box.

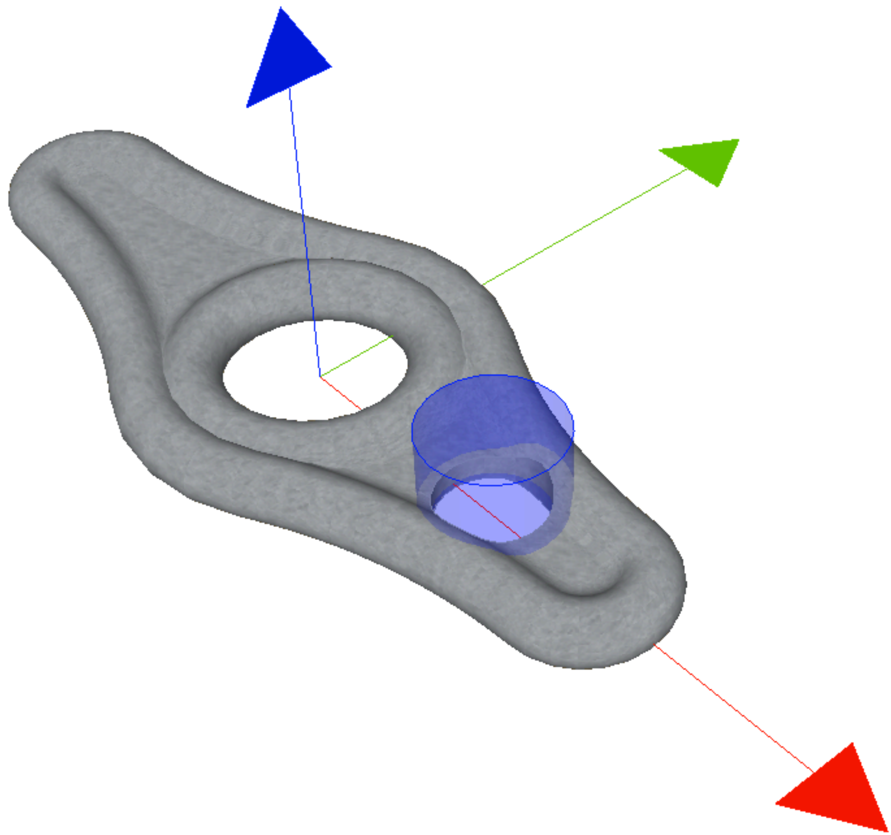


Figure 114: Pseudo-symmetric part (symmetry around X-axis, pseudo-symmetry around Y-axis) with inclusion box

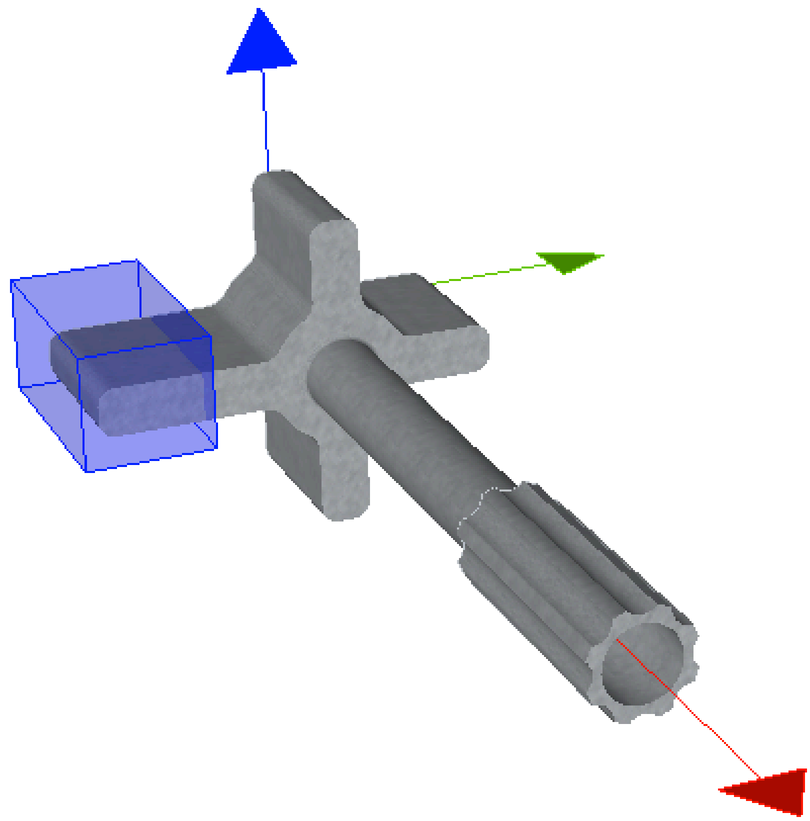


Figure 115: Pseudo-symmetric part (4-fold pseudo-symmetry around X-axis) with inclusion box

13.1.3.6 Partial model

It is possible to use only some sections of the CAD model as reference during localization. To specify which sections to use, you can define inclusion and exclusion boxes.

The partial model can be used to focus the localization on important details of the part, for instance if it is important that a small detail on a large part has correct position.

13.1.4 Edge algorithm concepts

This section explains concepts that are relevant for the edge algorithm.

13.1.4.1 Part model

The edge algorithm uses two types of features: edges and surfaces. Edges appear on distinctive structural features and hard edges. Surfaces appear on flat sections of the part.

The detected edges and surfaces are visualized with edge indicators and surface indicators in the reference image. Edge indicators are visualized as small circles, while surface indicators are visualized as large circles with a line representing the normal vector of the surface. see figure 116, page 130.

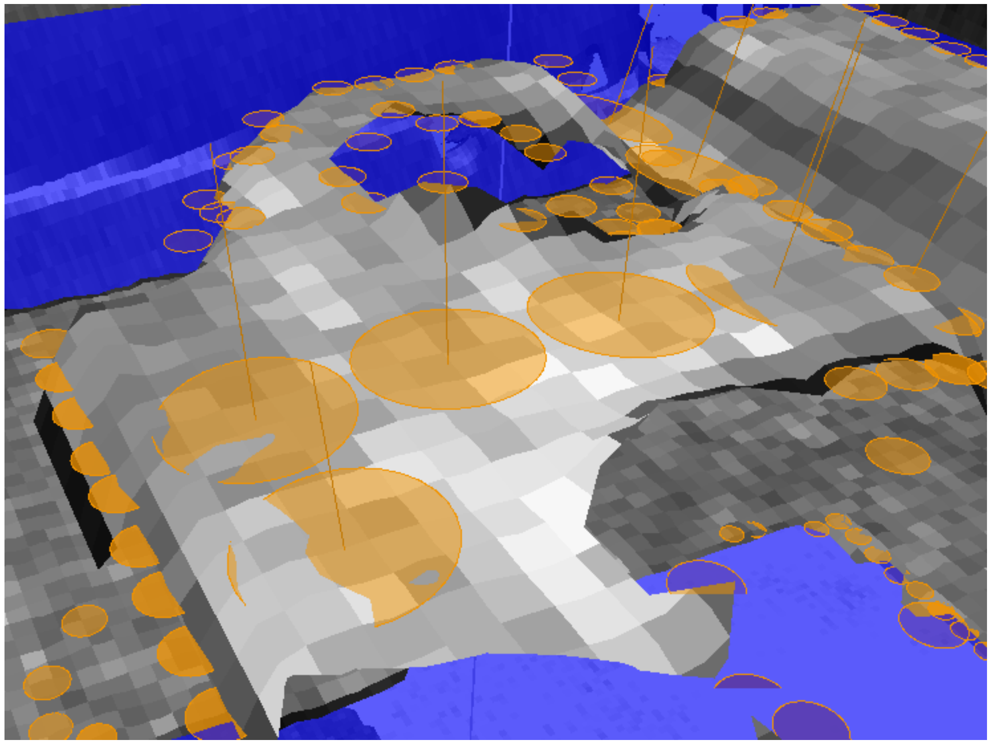


Figure 116: Surface indicators (large circles) and edge indicators (small circles)

A number of things can be done in order to acquire a good reference image with clearly visible edges:

- Use one isolated part in the expected tilt (default orientation) when acquiring a reference image.
- Set the Z-plane level so that the plane is below relevant edges.
- Use the region marking tool to highlight part contours in order to get a sufficient amount of edge indicators.
- Use the region marking tool to fill flat surfaces in order to get a sufficient amount of surface indicators.
- If unwanted indicators appear (e.g. if a surface tilts so much that it is identified as an edge) mask off the area by using the region marking tool eraser.

13.1.4.2 Score

Parts detected with the edge algorithm get a score value, telling how well the part matches the reference. PLB will only report parts that are above a score threshold.

If the score value is low this could be caused by:

- a badly located part,
- a part located with good accuracy but partially overlapped by another part,
- a part with an edge outside the search volume,
- a part with an edge against the edge of another part (no height difference).

13.1.4.3 Tilt

The edge algorithm works best when parts are not tilted too much compared to the reference image of the part. For jobs where parts occur with widely varying tilts, several jobs can be configured to locate all parts necessary.

13.1.5 Plug-in algorithms

It is possible to use a custom plug-in algorithm to locate parts. Plug-in algorithm parameters are added to the **Part** tab in PLB Studio. See the respective plug-in instructions for more information about the intended use of the algorithm, installation, etc.

13.1.5.1 Limited pick rotation

Plug-in jobs can be configured to adjust the pick pose of the located part, to make it easier for the robot to pick. The result is adjusted so the pick pose x-axis, as defined in the **Pick poses** tab (see "Configuring pick poses for limited pick rotation", page 73), is directed towards a reference point.

The reference point is configured in the **Bin and search volume** tab (see "Configuring the reference point for limited pick rotation", page 79) in relation to the robot frame origin. The reference point is typically set to the robot position when picking from a platform or conveyor belt. When picking from a bin with walls, it is typically set inside the bin on the robot side.

An example of gripper position before and after the adjustment is shown in figure 117 and figure 118.

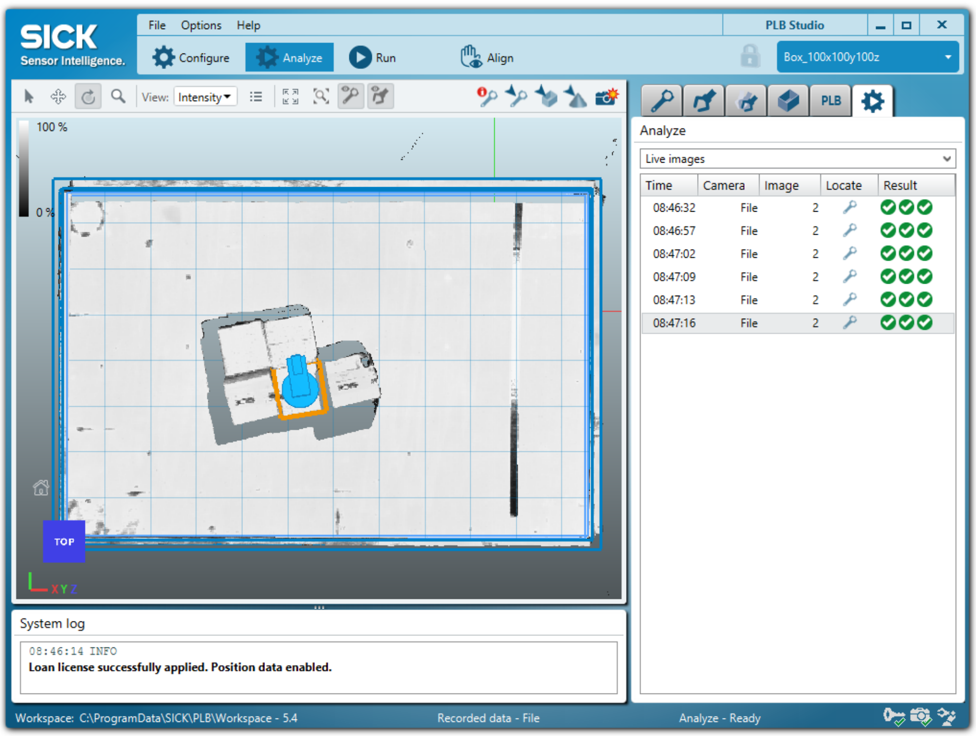


Figure 117: Gripper position without limit pick pose rotation

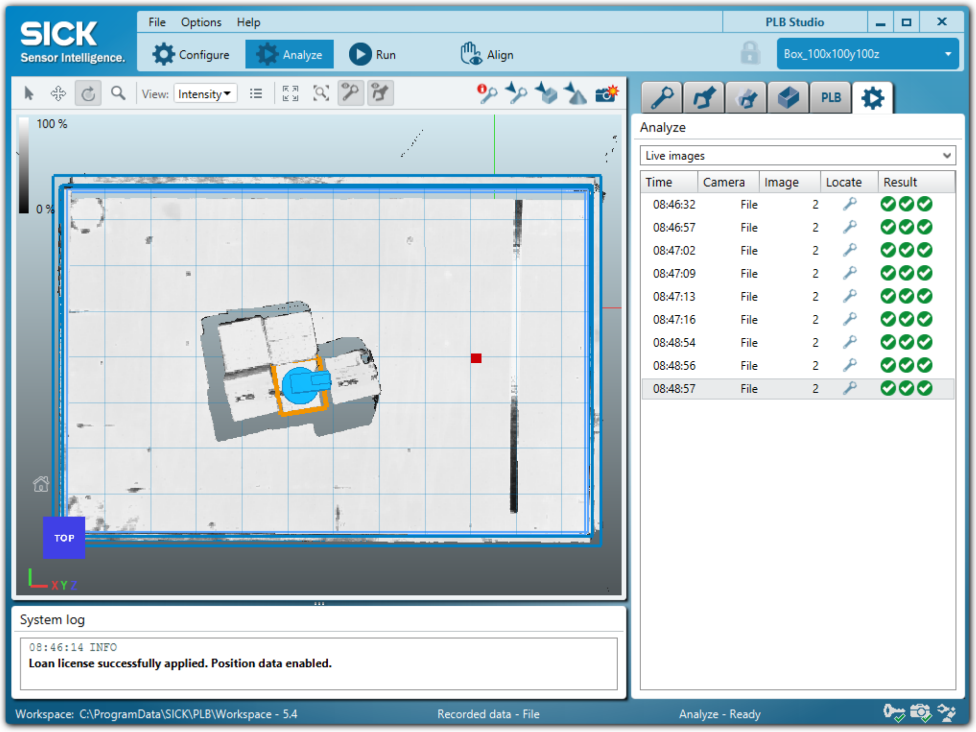


Figure 118: Gripper position adjusted to reference point

This feature can be enabled for the following two types of plug-in jobs:

- The pick poses are centered on the found parts.
- Symmetrical parts, where a fixed pick pose works no matter how the part is rotated about the z-axis.

13.2 Parameter settings in PLB Studio

This appendix contains a description of all the settings that you can make in PLB Studio.

13.2.1 Parameters for system configuration

These parameters apply globally throughout the system. You will find them under **Configure** and **System settings** . Default values are indicated in bold.

13.2.1.1 Cameras

Parameter	Permitted values	Description
Select camera type	**Recorded data**, X36, N36, C57, Visionary-S, Visionary-T Mini, Zivid 2+	A new camera is created and added to the camera list with type and camera alignment alias names.
Select camera mounting	**Stationary**, Robot	Select how the new camera is mounted (see "Mounting concepts", page 15).
Enabled	Enabled cameras are shown in a non-slanted font in the camera list. Note that the camera selected in the camera list must be enabled before switching to the **Analyze** or **Run** page.	

Parameter	Permitted values	Description
Connection settings/ IP address	0.0.0.0... ...255.255.255.255 Visionary cameras: **192.168.1.10**	The IP address of the camera. The IP address is only specified once and must be updated if you replace the camera or if the IP address of the camera changes (see also "Adding and editing a camera alignment alias", page 46).
Connection settings/ Serial number	Enter the serial number of the Ensenso or Zivid camera. The Ensenso and Zivid cameras use a serial number instead of the IP address for the connection.	
Connection settings/ 	Master license: 9 blocks consisting of 5 characters each xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx Slave license: 2 blocks consisting of 5 characters each xxxxx-xxxxx	Click and then enter the software license key. The button is only available if a camera is connected.
Connection settings/ Status	Read-only	Information about the camera connection, license, and additional camera information.
Camera settings/ Camera parameter file	The path for the parameter file needed to operate the camera. Use the default file or a created parameter file, see "Connecting to Ensenso cameras", page 47 or "Connecting to Zivid camera", page 48.	
Data selection/ Recorded data path	The path of the parent folder for a given recording folder with the format Recording_YYYYMM-DDTHH.MM.SS (see "Managing recorded data", page 84).	
Camera alignment alias	Camera alignment alias of the selected camera. For multiple hand-eye alignments for a camera, the different camera alignment alias names are shown in a list.	
Camera alignment alias/ Aligned with robot	Alignment status of the selected camera alignment alias.	

13.2.1.2 Hand-eye alignment

Parameter	Permitted values	Description
Alignment target type	**Cone** Sphere	The type of alignment target being used.
Cone diameter	80...120 (**115**) [mm]	The diameter of the alignment target cone.
Cone top angle	75...90 (**84**) [deg]	The top angle of the alignment target cone.
Sphere diameter	120...170 (**150**) [mm]	The diameter of the alignment target sphere.
Advanced \Max target deviation	0...3 (**2.0**) [mm]	The maximum allowed deviation between the located target position and the target position given by the robot. This threshold is used to determine if a result is OK or failed: • Save alignment fails if any of the added points has a deviation larger than this limit. • Verify alignment fails if the deviation is larger than this limit.

Parameter	Permitted values	Description
Advanced \Max scaling deviation	0...3 (**2.0**) [%]	The maximum allowed deviation in scale of an alignment matrix. A scaling error means that there is a difference in the scale of the camera world and the robot world. This threshold is used to determine if a save or verify alignment result is OK or failed.

13.2.1.3 Robot communication

Parameter	Permitted values	Description
Command interface	**PLB Standard**, VASS	Protocol used by PLB robot communication interface. The protocol is defined with a command template file when using PLB Standard. Other command interface options can be more restricted.
TCP\IP port	1024...65535 (**6008**)	Port number of the TCP/IP server. The robot client must also use this port number.
Verbose communication logging	Enabled or **disabled**	Enable this function if you want to generate a comprehensive log file. The log file contains all the commands and messages that are exchanged between the robot controller and the system and are listed in the system log. This function is particularly useful when creating the robot program and during troubleshooting in the event of communication problems.
Advanced\Robot communication file	**\RobotInterface_CSVTemplate.cfg**	Path of the file in which communication to the robot controller is defined.
Advanced\Rotation convention	x-y-**z**	Rotation convention used for PLB results via the robot communication interface.
Advanced\Send result software signal for	**Each zone with pick candidate**, Each zone with or without pick candidate.	Determines whether the system outputs a measurement result for each analyzed zone or for each analyzed zone with a reachable part. After the last zone has been analyzed, a message is output irrespective of the result of the analysis. It indicates that the analysis has been completed.
Advanced\Fixed number of decimals	Enabled or **disabled** 0...9 (**2**)	Enable to limit the number of decimals in locate results, e.g. to shorten the message length in the response to the robot controller.

13.2.1.4 Recordings storage

Parameter	Permitted values	Description
Maximum number of sub folders	0...1000 (**40**)	Maximum number of stored recording folders (format: Recording_YYYY-MMDDThh.mm.ss.xxxxxx). When the number of recording folders exceeds this value, the oldest folder is deleted. Set the value to 0 to disable automatic deletion.
Recordings base folder	**Recordings**	Path to the base folder for recordings.
Saved recordings base folder	**Recordings saved localizations**	Path to the base folder for recordings saved by the ExportSupportData robot command.
Failed recording storage	Enabled or **disabled**	Enable to store the recordings of failed localizations.

Parameter	Permitted values	Description
\Failed recordings base folder	**Recordings failed localizations**	Path to the base folder for recordings of failed localizations.

13.2.1.5 Archive

Parameter	Permitted values	Description
Archive base folder	**Archive**	Path to the base folder for the archive.

13.2.1.6 Evaluation and demonstration

Parameter	Permitted values	Description
Enable Surface algorithm	**Enabled** or disabled	Evaluate and demonstrate PLB using the Surface algorithm.
\Captcha dialog for full functionality	**Enabled** or disabled	PLB Surface algorithm can be evaluated and demonstrated without a license. A captcha dialog will pop up when going to run mode. When the captcha code is entered the part frame and the visualization overlay becomes available, for a limited amount of part localizations. This makes it possible to evaluate and demonstrate the full functionality of PLB. If disabled, the part frame is not available and the part and gripper model overlays will not be displayed in run mode.

13.2.2 Parameters for job configuration

These parameters are used for configuring jobs. You will find them under **Configure**. The specifications given in square brackets indicate higher-level sections of the PLB graphical user interface which contain the parameters.

13.2.2.1 Part - all algorithms

These parameters are used to configure parts. You will find them in the **Part** tab.

Parameter	Permitted values	Description
Alias		The name of the job that is used in the communication protocol.
Algorithm	**Surface** Edge The name of a plug-in algorithm	Selection of localization algorithm. Choose • **Surface** to locate the part using the surface-based algorithm with a part CAD model as a reference. • **Edge** to locate the part using the edge-based algorithm with an image as a reference. • The name of the algorithm to locate the part using a plug-in algorithm (if applicable).

13.2.2.2 Part - Surface algorithm

These parameters are used to configure parts using the Surface algorithm. You will find them in the **Part** tab if you have selected **Surface algorithm.**

13.2.2.2.1 Model

These parameters relate to the CAD model of the part.

Parameter	Permitted values	Description
Class	**No constraints** Limited tilt Revolving Revolving pick pose Beam	Classifying parts improves the reliability of localization and consequently helps improve system performance in terms of computing speed and stability. You will find further information on the concept of parts classes in "General concepts", page 114.
\Imported CAD model		Path to the imported CAD file for the part.
\Reference frame\Final transform		Transformation between the part reference frame defined in the CAD model and the adapted part reference frame.

13.2.2.2.2 Geometry

These parameters are used to specify symmetries.

Parameter	Permitted values	Description
Revolution axis	Enabled or **disabled:** x-axis y-axis z-axis	Use this parameter for rotationally symmetrical parts. Please note that the setting “Revolution: z-axis” is not compatible with the setting "Part\Localization\Surface visible to camera\Length".
Symmetry axis	Enabled or **disabled:** x-axis y-axis z-axis	Use this parameter for parts that look identical if flipped about the X, Y, or Z-axis.
Pseudo flat	Enabled or **disabled**	Use this parameter for parts that have flat, parallel surfaces.
No variation along x-axis	Enabled or **disabled**	Use this parameter for parts that have identical dimensions along the X-axis, e.g. cylinders, pipes, uniform beams.
Pseudo symmetry	Enabled or **disabled**	Use this parameter for parts that look almost identical if flipped about the X, Y, or Z-axis.
\Axis of symmetry	x-axis y-axis z-axis	Specify the axis of flip symmetry.
\Inclusion exclusion box		Use an inclusion box to define a section that is not identical if the part is flipped, or use an exclusion box to define the section which is identical if the part is flipped.
Cyclic repetition	Enabled or **disabled**	Use this parameter for parts that possess an n-fold symmetry (cyclic repetition) around the X, Y, or Z-axis, such as sprockets, for example.
\Axis of symmetry	x-axis y-axis z-axis	Specify the axis around which the part is rotated.
\Repetitions	3...20	The number of times a part is rotated before reaching its starting position again. Example: The number of rotations of a sprocket corresponds to the number of teeth.
Pseudo cyclic repetition	Enabled or **disabled**	Use this parameter for parts that possess an n-fold symmetry (cyclic repetition) around the X, Y or Z-axis but with a slight deviation.
\Axis of symmetry	x-axis y-axis z-axis	Specify the axis around which the part is rotated.

Parameter	Permitted values	Description
\Repetitions	3...20	The number of times a part is rotated before reaching its starting position again.
\Inclusion exclusion box		Inclusion and exclusion boxes should exactly surround the corresponding area of the part while remaining as small as possible.
Partial model	Enabled or **disabled**	Use this parameter when you do not want to use the whole CAD model as a reference when the part is located.
\Inclusion exclusion box		Inclusion and exclusion boxes should exactly surround the corresponding area of the part while remaining as small as possible.

13.2.2.2.3 Part orientation in bin

These parameters can be used to define constraints that are to apply to localization. If a part is found in a position that is not defined here then it cannot be picked up.

Parameter	Permitted values	Description
Maximum rotation around axis\X, Y, Z	Enabled or **disabled**; Default settings depend on the classification of the part	Set this parameter if • rotation of the part around an axis is restricted when it is lying in a bin, or • the part can only be picked up if it has a certain rotation
Maximum tilt of plane when seen from:\Top, Bottom, Front, Right, Back, Left	Enabled or **disabled**; Default setting depends on the classification of the part	Set this parameter if • rotation of the part around a plane is restricted when it is lying in a bin, or • the part can only be picked up if it has a certain oblique angle This parameter is of particular relevance for parts which typically always lie with a given side facing upward, i.e. if the camera always sees a certain side of the part. Use the 3D navigation aid to identify the front, back, top and bottom sides of the part.

13.2.2.2.4 Localization

Test the parameters listed here thoroughly before using them in production.

The following parameters influence localization accuracy:

Parameter	Permitted values	Description
Image data quality	Good, **Medium**, Poor	Lower this setting if the parts' dimensions and appearance can differ from the part CAD, or if the 3D image of the parts is of poor image data quality. This setting can be increased if the 3D image of the parts is of very high image data quality.

Parameter	Permitted values	Description
Maximum number of results	4...16 (**4**)	Increase this value to increase the maximum number of possible localization results reported from the same image. Normally a new image is acquired after each robot pick, but if parts can be picked without moving other parts, this value can be increased to pick more parts from the same image. Increasing this value can lead to longer localization times.
Filter ratio standard deviation	1.9...3.6 (**2.6**)	This parameter refers to the internal iterations of the localization algorithm and determines the number of image points that are transferred or held back from one iteration to the next. To obtain a suitable setting, we recommend that you reduce or increase the value in steps of 0.2 (e.g. 2.2, 2.4, 2.6).
Average distance to model	0...50 [mm] (**1.5**)	Points located within this distance to the CAD model are located on the surface of the part. Set this parameter in order to permit small deviations between the CAD model and the physical part or to compensate for measurement tolerances (such as measurement noise).
Portion of points inside	0...100 [%] (**0.5**)	This parameter can be used to limit the number of points that may be located within the CAD model. The value is calculated from the number of measurement points inside the model divided by the number of measurement points on the surface of the model. Ideally, there should be no measurement points inside the model. In practice, however, these occur due to surface reflections and measurement inaccuracies.

The following parameters define what proportion of a part surface must contain 3D measurement points:

Parameter	Permitted values	Description
Surface visible to camera	**Enabled** or disabled	This parameter determines the size of the part that the camera must recognize. To make sure that parts are recognized between camera configurations, only points belonging to a surface whose normal is angled 45 degrees or less to the vertical positive Z-axis can be characterized as visible to the camera.
\Length	**Enabled** or disabled 0...100 (**80**) [%]	The box surrounding the part is subdivided into slices along its length. The width of each slice is determined by this parameter. At least X % of the slices must contain at least one 3D measurement point.
\Surface	Enabled or **disabled** 0...100 (**70**) [%]	The box surrounding the part is subdivided into slices along its two longest sides, i.e the box is subdivided into squares. At least X % of the squares must contain 3D measurement points.
\Resolution	1...50 (**3**) [mm]	The resolution of the slices.

Parameter	Permitted values	Description
Algorithm time out	0.1...30.0 (**30.0**) [s]	Maximum permitted time in seconds that is available in order to locate a part. If it is not possible to locate a part within this period then the calculation is aborted.
Result sort order	**Highest z value** Highest x value Lowest x value Highest y value Lowest y value Scanned surface Horizontal parts)	This parameter determines the measurement points at which the localization algorithm is started. X, Y, and Z refer to the coordinates of the image data that were acquired during hand-eye alignment. If you select the value “Highest z value” then the algorithm starts at the highest points. If you select the value “Highest x value” then the algorithm starts at the leftmost points, etc. If you select the value “Scanned surface” then the algorithm starts at the parts which have the most measurement points. If you select the value “Horizontal parts” then the algorithm starts at the parts which have the smallest oblique angle.
Blacklist	Enabled or **disabled**	The blacklist functionality is useful if the robot for some reason cannot pick a located part. When the blacklist is enabled, the position of located parts is added to the blacklist. When PLB has located a part it will check if the result is already on the blacklist; if so it will not send the result but instead try to locate a new part. Blacklist is only active in Run mode, not in Analyze mode. It is recommended to clear the blacklist when replacing the bin. The blacklist is cleared when: • sending the ClearBlacklist command, see "ClearBlacklistMsg", page 165 for further information. • going to Configure mode.
Duplicate within distance	0 ... 1000 (**6**) [mm]	A new result that is within this distance from a blacklisted result will not be reported. Even if the scene has not changed, the image data will change within a small tolerance and a located part will not have the exact same position.

13.2.2.2.5 Overlap

You use these parameters to define the tolerance for overlaps. To achieve optimum system performance, only specify the corresponding values after you have run subsequent iterations on the parameters in order to adapt them.

Parameter	Permitted values	Description
Enable overlap tests	**Enabled** or disabled	Enables overlap detection
\Vertical offset	Enabled or **disabled** 0...9999 (**100**)	Measurement points that lie outside the offset or which have a value greater than the value specified here are not considered to be overlaps. In this way, it is possible to ensure that the gripper can pick up parts even when there are large gaps.

Parameter	Permitted values	Description
\Vertical tolerance	0...9999 (**10**)	Points with a value smaller than the value specified here are not considered to be overlaps. It is assumed that measurement points that are located very close to the part are not overlaps but instead represent measurement noise.
\Width	0...9999 (**10**)	If the box surrounding an overlap area is at least as wide and as long as is specified by the values entered here then an overlap is detected.
\Length	0...9999 (**50**)	
\Area	0...9999 (**1000**)	If an overlap area is greater than the specified value then an overlap is detected.
\Inclusion and exclusion boxes		Only 3D measurement points that lie inside the inclusion box are taken into account during the overlap test. 3D measurement points that lie outside the inclusion box are not taken into account during the overlap test. You can define up to eight inclusion and eight exclusion boxes.
\Selected box properties \X, Y, Z	0...9999	Size of the inclusion or exclusion box (length, width, height).
\Selected box properties \Center pose X, Y, Z, Rx, Ry, Rz	X, Y, Z: -9999...9999, Rx, Ry, Rz: -180...180	Position and orientation of the inclusion or exclusion box.

13.2.2.3 Part - edge algorithm

These parameters are used to configure parts using the edge algorithm. You will find them in the **Part** tab if you have selected the edge algorithm.

13.2.2.3.1 Teach

Parameter	Permitted values	Description
Edge threshold	1.0...2000.0 (**6.0**) [mm]	Height differences in the reference image which is above this threshold will be considered as edges.
Resample resolution	0.1 ... 10 (**3.5**) [mm]	Resampling resolution (pixel size) of the 3D image. This value should be near the point cloud density of the camera used. Increasing this value increases performance but decreases the quality of the edge/plane markers. Decreasing this value increases quality but also increases the risk of noise in the pixel sampling.
Maximum number of edges	50...**1000**	The more edges that are detected in the reference image, the longer will the localization calculation time be. If calculation time is too long for your application then the maximum number of edges can be reduced. This can reduce the localization performance. (Robustness vs speed)

Parameter	Permitted values	Description
Z plane	Permitted values depend on image data [mm]	All image data points that are above this Z plane can be used for detecting edges and surfaces. Click the **Select** button in the PLB Studio visualization area to choose a tool to paint the area (in X and Y direction) which defines the teach volume.
Lock part reference frame	Enabled or **disabled**	Enable to prevent defined pick poses from moving when changing the part shape.

13.2.2.3.2 Localization

Parameter	Permitted values	Description
Score threshold	0...100 (**70**) [%]	Located parts that have a score value below this threshold will not be reported.
Rotation range +/-	0...**180** [degrees]	If the rotation of the parts in the bin is limited, then this parameter can be set to decrease calculation time and increase robustness.
Limit tilt of part	Enabled or **disabled**	Do not report results tilted above a set limit.
\Tilt limit	0...90 [degrees]	Located parts that are tilted above this limit are not reported. The tilt angle is calculated between the tilt (relative to the defined bin) in the reference image and tilt of the located part.
Advanced\Search depth	2...10 (**10**)	The edge algorithm is an iterative process. The search depth parameter determines how long the algorithm should try to find a part before aborting calculations.
Advanced\Blacklist	Enabled or **disabled**	The blacklist functionality is useful if the robot for some reason cannot pick a located part. When the blacklist is enabled, the position of located parts is added to the blacklist. When PLB has located a part it will check if the result is already on the blacklist; if so it will not send the result but instead try to locate a new part. Blacklist is only active in Run mode, not in Analyze mode. The blacklist is cleared when: • sending the ClearBlacklist command, see "ClearBlacklist", page 158 for further information. • going to Configure mode. It is recommended to clear the blacklist when replacing the bin.
Advanced\Duplicate within distance	0 ... 1000 (**6**) [mm]	A new result that is within this distance from a blacklisted result will not be reported. Even if the scene has not changed, the image data will change within a small tolerance and a located part will not have the exact same position.
Advanced\Require converging match	**Enabled** or disabled	Disable if there is a problem with non-converging localization results.

13.2.2.4 Part - plug-in algorithm

These parameters are used to configure parts using a plug-in algorithm. You will find them in the **Part** tab if you have selected a plug-in algorithm.

Parameter	Permitted values	Description
Score threshold	0...100 (**70**) [%]	Located parts that have a score value below this threshold will not be reported.
Downsampling distance	Enabled or **disabled**	PLB sends an image to the plug-in algorithm. The distance between points of the point cloud, or the distance between height map samples in this image can be changed.
Advanced\Blacklist	Enabled or **disabled**	see "Localization", page 141.

13.2.2.4.1 Localization

Parameter	Permitted values	Description
Score threshold	0...100 (**70**) [%]	Located parts that have a score value below this threshold will not be reported.
Downsampling distance	Enabled or **disabled**	PLB sends an image to the plug-in algorithm. The distance between points of the point cloud, or the distance between height map samples in this image can be changed.
Advanced\Blacklist	Enabled or **disabled**	see "Localization", page 141.

13.2.2.4.2 Other plug-in settings

Available parameters vary depending on the plug-in algorithm. Click **?** (Help) to get more information about specific plug-in settings. The button is not available for all plug-in algorithms.

13.2.2.5 Gripper

These parameters are used to configure grippers. You will find them in the **Gripper** tab.

Parameter	Permitted values	Description
Imported CAD model		Path to the imported CAD file for the gripper
Align CAD with robot TCP\Incremental transform		Incremental transformation which results in the final transformation between the CAD model of the gripper and the TCP
Align CAD with robot TCP\Final transform		Final transformation which describes the alignment of the CAD model of the gripper with the TCP
Approach direction\Approach along	+x axis -x axis +y axis -y axis **+z axis** -z axis	Specifies the axis along which the gripper approaches the part.

13.2.2.5.1 Advanced - edge and plug-in algorithms

These parameters are only applicable for the edge algorithm and plug-in algorithms.

Parameter	Permitted values	Description
Advanced\Collision accuracy level	**Default** Custom	Sets the resolution of the gripper model to be used for collision checks.
Advanced\Accuracy value	50 000... 100 000 000 (**10 000 000**)	Number of elements in the gripper model to be used for collision tests. This parameter is only visible and editable when the **Collision accuracy level** parameter is set to **Custom**.

Parameter	Permitted values	Description
Advanced\Show collision volume	Enabled or **disabled**	Displays the representation of the gripper during collision checks.

13.2.2.6 Pick poses

These parameters are used to configure pick poses. You will find them in the **Pick poses** tab.

Parameter	Permitted values	Description
Enabled	**Enabled** or disabled	Enables or disables the pick pose.
Alias	Alphanumeric characters a...z, A...Z, 0...9	The name of the pick pose that is used in the communication protocol.

13.2.2.6.1 Pose definition

Parameter	Permitted values	Description
Gripper		Select the gripper that is to be used.
Relative tool frame\Final transform		Final transformation which describes the conversion of the gripper reference frame to the desired pick pose

13.2.2.6.2 Dynamic pick poses - Surface algorithm

These parameters are only applicable for the Surface algorithm.

Parameter	Permitted values	Description
Cylindrical around x-axis at pick pose	Enabled or **disabled**	Enable to correct pick poses at cylindrical sections of the part (see "Vertical correction", page 124). Cannot be combined with the **Rotational pick pose** option.
Rotational pick pose	Enabled or **disabled**	Enable to define multiple pick poses in an arc around a rotation point (see "Configuring rotational pick poses", page 73). Cannot be combined with the **Cylindrical around x-axis at pick pose** or the **Linear pick pose** options.
\Start angle	-180...**0** [degrees]	Set the start angle to define how far along the arc pick poses will be defined.
\End angle	**0**...180 [degrees]	Set the end angle to define how far along the arc pick poses will be defined.
\Rotation point		Use the incremental transform tools to define the rotation point and the rotational axis.
Linear pick pose	Enabled or **disabled**	Enable to define multiple pick poses in a line (see "Configuring linear pick poses", page 74). Cannot be combined with the **Rotational pick pose** option.
\Pick pose end		Use the increment tools to define where the linear pick pose will end. The linear pick pose starts as defined in "Pose definition", page 143.

13.2.2.6.3 Waypoints

Parameter	Permitted values	Description
Entry pose\Height above bin	0-1000 (**100**)	The distance from the gripper's tool center point to the top of the bin. The gripper will assume an orientation aligned with the pick position of the located part.
Approach pose\Distance	0-1000 (**100**)	Distance in the approach direction of the gripper at which the Approach pose should be placed.
Clear pose\Height above part	0-1000 (**100**)	The distance from the gripper's tool center point to part, in the positive Z direction of the bin.
Exit pose\Height above bin	0-1000 (**100**)	The distance from the gripper's tool center point to the top of the bin. The gripper will assume an orientation aligned with the pick position of the located part.

13.2.2.6.4 Pose collision settings - All algorithms

These parameters are applicable for both the surface and the edge algorithm.

Parameter	Permitted values	Description
Collision test\Points	**Enabled** or disabled	Point collision detection can be disabled separately for each pick pose.
	0...999999 (**10**)	Number of 3D measurement points that may lie within the gripper without a collision being detected.
Collision test\Approach test	**Enabled** or disabled	Enables collision detection for the approach direction. Used to detect whether the gripper collides with 3D measurement points as it approaches the pick pose. Overlaps are not detected along the approach direction but only at the pickup point.

13.2.2.6.5 Pose collision settings - Surface algorithm

These parameters are only applicable for the Surface algorithm.

Parameter	Permitted values	Description
Collision test\Area	Enabled or **disabled**	Area collision detection can be enabled separately for each pick pose.
	0...999999 (**1000**)	If an overlap area is greater than the specified value then a collision is detected. This parameter is only available for the Surface algorithm.
Collision test\Rectangle	Enabled or **disabled**	Rectangle collision detection can be enabled separately for each pick pose.
\Width	0...999999 (**20**)	If the box surrounding an overlap area is at least as wide and as long as is specified by the values entered here then a collision is detected. These parameters are only available for the Surface algorithm.
\Length	0...999999 (**50**)	

13.2.2.6.6 Settings common to all pick poses - All algorithms

These parameters are applicable for edge, surface and plug-in algorithms.

Parameter	Permitted values	Description
Inclination angle	Enabled or **disabled** 5...90 (**45**)	Describes the gripper's maximum angle of inclination. The system only suggests pick poses in which the gripper's angle of inclination relative to the robot's reference frame Zaxis is smaller than this value.

13.2.2.6.7 Settings common to all pick poses - Surface algorithm

These parameters are only applicable for the Surface algorithm.

Parameter	Permitted values	Description
Order of priority	**List order** Part center of gravity Topmost pick pose Highest x Lowest x Highest y Lowest y	If more than one validated pick pose is output then the pick poses are sorted in the sequence specified here. “List order”: The sequence of pick poses in the list is adhered to. “Part center of gravity”: The pick pose closest to the part's center of gravity is preferred. “Topmost pick pose”: Topmost pose first (in the Z direction). “Highest x”: The leftmost pose, etc.
Collision check	**Only inside** Inside and overlap	The collision tests can be enabled and disabled for each pick pose. This setting is common to all pick poses (where collision test is enabled). A collision can be detected for the following cases, if: • the gripper model collides with 3D measurement points (inside) or • the gripper model is overlapped by 3D measurement points (overlap) This parameter determines if both cases are calculated or if only the first case is calculated. If you use the setting “Only inside” then you must also enable the setting “Virtual walls”.

13.2.2.7 Bin and search volume

These parameters are used to configure the bin and search volume. You will find them in the **Bin and search volume** tab.

Parameter	Permitted values	Description
Alias		Bin or search volume alias used as `$binAlias` parameter value in robot commands.
Volume type	**Bin** or Search volume	A bin is defined with bin walls. It is possible to fine-tune the bin position using the `LocateBin` robot interface command, and PLB checks for collisions with the walls. Search volume is used if there are no bin walls.

Parameter	Permitted values	Description
Volume position	**Fixed in robot cell** or Follow camera	**Fixed in robot cell:** The search volume is defined in absolute robot world coordinates, and will remain in a fixed spot in the robot cell even if the camera position during locate is changed. **Follow camera:** The search volume is defined relative to the camera position, and will move if the camera moves. This option can be useful if the robot cell has many search volumes. Note: The **Follow camera** option requires a completed hand-eye alignment.

13.2.2.7.1 Definition

Parameter	Permitted values	Description
Size presets	**Large 1200x800x600** Medium 800x600x400 Small 400x300x200 Custom size	Preset internal dimensions of the search volume.
X size, Y size, Z size	10...9999 [mm]	Only 3D measurement points that lie inside this volume are taken into account during localization. Define the search volume in such a way that it lies within the actual bin volume or so that the bin walls are masked out.
Wall top thickness	3...100 (**20**) [mm]	Describes the thickness of the bin walls. This information is required for bin localization.
Center pose/Final transform X, Y, Z, Rx, Ry, Rz	X, Y, Z: -9999...9999, Rx, Ry, Rz: -180...180	Position and orientation of the search volume.

13.2.2.7.2 Bin localization

Parameter	Permitted values	Description
Score threshold	10...100 (**50**)	If you use the bin localization function then the result of the calculation must be higher than this threshold value in order to be categorized as successful.
Allowed bin offset	0...2000 [mm] (**200**)	This parameter defines how much found bin pose is allowed to deviate from pre-defined bin pose. Lower this parameter to prevent the wrong bin to be found if multiple bins are visible in the scene.
Max length/width ratio dev.	0...10 [%] (**2**)	This parameter defines how much found bin shape is allowed to deviate from pre-defined bin shape. Increase this parameter if bin length and width relation vary between individual bins.

13.2.2.7.3 Bin collision

These parameters are used to configure virtual bin walls in order to optimize collision detection.

Parameter	Permitted values	Description
Grid resolution	1...100 (**10**)	Describes the grid resolution of the virtual bin walls. This should be high enough to permit the detection of collisions with the bin walls. At the same time, the value should be as small as possible to ensure that the processing time does not increase due to the presence of an excessive number of measurement points.
Virtual walls	**Enabled** or disabled	"Virtual walls" make it possible to fill the bin walls with additional measurement points. This is useful if there are only a small number of measurement points on the bin walls. Because images are acquired from the top of the bin, it is difficult to obtain measurement points on the bin walls. However, this information is needed for collision detection. You can remedy this problem by defining artificial bin walls that are added to the 3D point cloud. If “Virtual walls” are disabled, then the “Pose collision settings/Settings common to all pick poses/Collision check” should be set to “Inside and overlap”.
\Virtual wall position	**Inside wall** Outside wall	The inside or outside bin walls are filled with additional points. Use the setting “Inside wall” if the bin walls are exactly as thick as the top edge of the bin. This prevents the gripper from colliding with the inside of the bin. Use the setting “Outside wall” if the bin walls are thinner than the top edge of the bin. Otherwise false collisions may be detected.
Virtual bottom	Enabled or **disabled**	"Virtual bottom" makes it possible to fill the bottom of the bin with additional measurement points. If your bin has an open base, you can define a plane that is flush with the base. The system then only considers points that are located above this plane. This makes it possible to prevent possible interferences or measurement noise. Please note that points that lie directly above this plane (approx. 2 mm) are automatically excluded from the calculation.
Virtual walls above top	0...1000 (**0**)	Specifies the height of the virtual bin walls that extend beyond the top edge of the bin. Use this parameter if the bin has handles.
Safety walls	**Enabled** or disabled	"Safety walls" make it possible to create additional safety walls that surround the bin. Use the parameter to prevent the gripper from performing pickup through the bin walls.
\Distance	1...200 (**50**)	Describes the distance between the bin walls and the additional safety walls.

13.2.2.7.4 Search volume

Parameter	Permitted values	Description
Height above top	-500...2000 (**100**)	This parameter allows top parts in overfilled bins to be included in the search volume. Lower this parameter to prevent an overhanging bin wall top to affect overlap and collision checks.

Parameter	Permitted values	Description
Distance from wall	-100...100 (**10**)	This parameter reduces the search volume by the corresponding value in the X and Y directions to compensate for measurement discrepancies.
Height above bottom	0...200 (**0**)	This parameter reduces the search volume by the corresponding value in the Z direction to compensate for measurement discrepancies.

13.2.2.7.5 Lower plane

Parameter	Permitted values	Description
Dynamic lower plane	**Enabled** or disabled	If this parameter is set then the program searches the current image for the virtual lower plane. Only 3D measurement points that lie above this plane are taken into account during localization.
\Bottom flatness tolerance [mm]	1...50 (**3**)	If the lower plane is dynamically adapted then points that lie within this tolerance range are considered to belong to the plane.
\Plane offset [mm]	-100...1000 (**2**)	Offset (margin) from the best-fit plane located by the algorithm. The search volume starts from this plane offset. If set to 0, the search volume starts from the best-fit plane. A positive value means that the search volume starts above the best-fit plane. If a dynamic lower plane is found, the plane offset is visualized in a semi-transparent blue color in the visualization area.
\Collision plane offset [mm]	-200...1000 (**0**)	Offset (margin) from the search volume plane, defined by the plane offset. If the gripper collides with the collision plane, the pick pose is rejected. If set to 0, the collision plane is defined at the same plane where the search volume starts. A negative value lowers the collision plane below the plane where the search volume starts. If set to a value that differs from 0, the collision plane is visualized in a semi-transparent red color in the visualization area.

13.2.2.7.6 Empty search volume

Parameter	Permitted values	Description
Empty threshold	0...10000 (**100**) [points]	If it is not possible to locate any part then the number of points in the search volume is determined. If the corresponding value is smaller than this threshold value then the search volume is considered to be empty.

13.2.2.7.7 Zones

Parameter	Permitted values	Description
Divide search volume in zones	1...4 x 1...4 (**1x1**)	Number of zones into which the search volume is subdivided.

Parameter	Permitted values	Description
Zone margin	0...1000 [mm] (**0**)	Margin added around the zones so that they overlap slightly. This is necessary in order to locate parts that are positioned between two zones.

13.2.2.7.8 Result sort order

These parameters are applicable to edge and plug-in algorithms.

Parameter	Permitted values	Description
Result sort order	**Highest z value** Lanes Layers Highest score Algorithm priority Least tilted Most tilted	Defines the order of the valid locate results.
If **Result sort order:** Lanes or **Sorting within layers:** Lanes		
Lane order	**Highest x value** Lowest x value Highest y value Lowest y value	Defines from which bin edge the lanes start. Note that the x or y direction is based on the local bin coordinate system.
Sorting within lanes	**Highest score** Highest x value Lowest x value Highest y value Lowest y value Highest z value* Least tilted Most tilted	How to sort the results within each lane. Note that **Lane order** and **Sorting within lanes** must be perpendicular. *The **Highest z value** option is not available when **Lanes** are selected in **Sorting within layers.**
Lane width	0...2000 [mm] (**50**)	How wide each lane is.
If **Result sort order:** Layers		
Sorting within layers	**Highest score** Highest x value Lowest x value Highest y value Lowest y value Least tilted Most tilted Lanes	How to sort the results within each layer.
Layer depth	0...2000 [mm] (**50**)	How deep each layer is.

13.2.2.7.9 Limit pick rotation

These parameters are applicable to edge and plug-in algorithms.

Parameter	Permitted values	Description
Rotate pick pose to reference point	Enabled or **disabled**	Enable to adjust pick poses to make it easier for the robot to pick the part.
Rotation mode	**Rotate gripper**, Rotate part	Select if the gripper should be rotated at each pick pose, or if the part should be rotated before using the normal pick poses (see "Configuring the reference point for limited pick rotation", page 79).

Parameter	Permitted values	Description
\Reference point position X, Y, Z	-10000...10000 (**0**)	Coordinates for the rotate pick pose reference point.
\Lock reference point to search volume	Enabled or **disabled**	If enabled, the reference point is defined relative to the search volume and will be adjusted when the search volume is changed, e.g. by a LocateBin command.

13.3 Command interface

Messages and commands that are exchanged between the system and the robot controller are configured in a special template file. This appendix describes the format of the template file. It also contains a list of the messages and commands.

13.3.1 Configuring messages and commands

The syntax used is flexible and makes it possible for you to configure messages and commands to meet your needs. Both XML-based and CSV-based formatting are supported. There are reference files with all available robot command choices in the **RobotInterface_CommandReference** folder that is located in your workspace's System folder.

The corresponding template files are named **RobotInterface_XMLTemplate.cfg** and **RobotInterface_ CSVTemplate.cfg**.

When you start the program for the first time, the template files are copied to the default Workspace folder **C:\ProgramData\SICK\PLB\Default Workspace\System**. If you create a new workspace then the template files are copied to this workspace's system folder. If you then select the new workspace, it is used on the next system start.

Proceed as follows to adapt the template files:

1. Open the robot interface template file in a text editor of your choice.
2. Edit it as required by positioning the parameter names (starting with $) at the required location in the template file.
3. Save the file.
4. Open PLB Studio.
5. Click **System settings** , **Robot communication** > **Advanced**, and choose the template file (see figure 119, page 151).
6. Click ... and select the required template file.
   Please note that after making any changes to the template file after selecting it, you must always de-select and re-select the file to reload the contents to PLB or restart PLB Engine.

During operation, the system searches the template file for the parameter names and replaces them with the corresponding values.

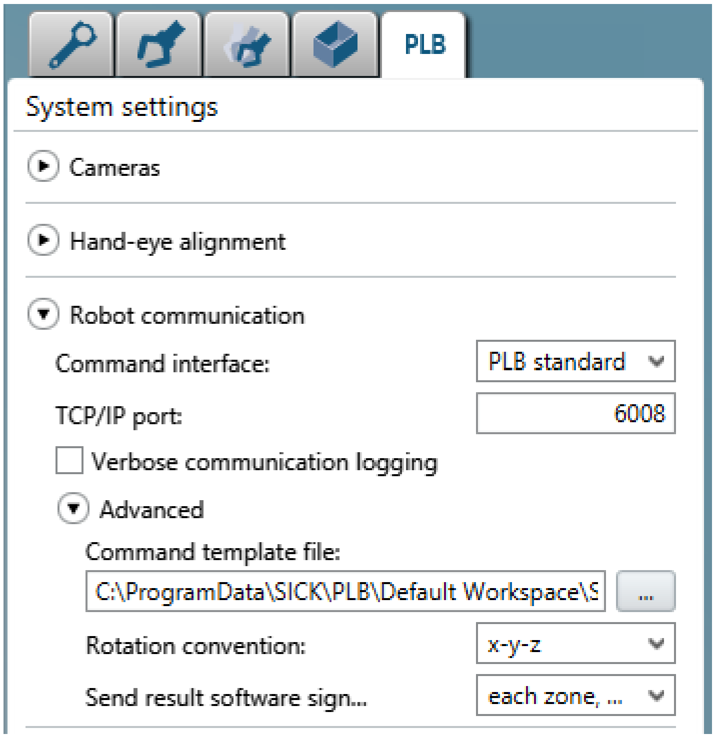


Figure 119: PLB Studio, selecting the template file

13.3.2 Format of the template file

A full list of command template options is found in the CSV and XML command reference files, see `C:\ProgramData\SICK\PLB\Default Workspace\System\RobotInterface_CommandReference`.

NOTE
Note that whitespaces in messages and commands are ignored in order to make communications more resistant to formatting errors and any reformatting that the robot controller may apply to XML commands.

Take account of the following points when formatting the template file:

- Messages or commands consist of a header in square brackets, followed by a command template.
- Headers, e.g. `[LocatePart]`, and $-parameters, e.g. `$jobAlias`, must be spelled as in the robot interface command reference files.
- Other text strings can be changed, but the start of the command must be unique within the template to make it possible for PLB to separate between different commands.
- All $-parameters are optional. The currently active setting will be used if a command is sent with a missing $-parameter.
- Parameter names must be separated from one another by at least one character. This must not be a whitespace character or an alphanumeric character.
- The character that immediately follows a parameter name must not be an alphanumeric character.
- The messages and commands in CSV format must end with a separate character. In the examples the semi-colon (;) is used.

Example

This is an example of a `SetState` (set state to Run) command from the robot controller and a `StateMsg` response (result is OK, set state is Run) from PLB.

Table 36: Template

Format	[SetState]	[StateMsg]
CSV	SetState,$state;	State,$result,$state;
XML	<SetState><State>$state</State></SetState>	<StateMsg><Result>$result</Result><State>$state</State>/StateMsg>

Table 37: Example

Format	Command	Response
CSV	SetState,1;	State,0,1;
XML	<SetState><State>1</State></SetState>	<StateMsg><Result>0</Result><State>1</State></StateMsg>

13.3.3 Robot commands

The PLB system can receive the following commands from the robot controller:

- Heartbeat
- SetState
- Trigger
- TriggerRM
- LocatePart
- LocateChain
- LocateChainRM
- LocateBin
- AddAlignmentPoint
- AddAlignmentPointRM
- AddAlignmentTargetPositionRM
- SaveAlignment
- VerifyAlignment
- VerifyAlignmentRM
- ClearBlacklist
- Log
- SetJobParameter
- SetCameraParameter
- GetJobs
- GetCameraAlignments
- ExportSupportData
- DefinePickPose

Each command is acknowledged with a message. You should always wait until you receive the message before sending a new command to the system.

If the system receives a command but is not yet able to process it then it issues a message indicating that it is not yet ready to process the new command. This only occurs if the system receives a command before it has been able to respond to the preceding command.

13.3.3.1 Heartbeat

Description: Use this command to verify that the robot has a working connection to PLB.
This command can be sent in any state, i.e. a response is sent even if PLB is busy with an ongoing operation.

Response: The response to the command is a `HeartbeatMsg` message.

13.3.3.2 SetState

Description

You use the `SetState` command to define the required system state. You can choose between:

- **Run**: To start production mode (see "Starting the image acquisition", page 82). The system is ready to receive commands of the type `Trigger/TriggerRM`, `LocatePart`, `LocateBin` and `VerifyAlignment/VerifyAlignmentRM`.
- **Configure**: For the initiation of the job configuration. For error handling purposes, the `SetState` command can also be used to abort an ongoing calculation in Run mode.
- **Align**: For starting the alignment procedure. The system is ready to receive commands of the type `AddAlignmentPoint/AddAlignmentPointRM` and `SaveAlignment`.
- **Analyze**: To test parameter values and analyze results from specific images. The actions can only be done from PLB Studio; it's only allowed to change to a different state from the robot communication interface.

Response

The `SetState` command is always responded to with a `StateMsg` message.

Parameters

`$state`
This parameter contains the current system state. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

`$jobAlias`
This parameter is used to inform the system of the job that is active.

`$cameraAlignmentAlias`
This parameter is used to tell the system what camera alignment alias that is active.

`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command. This may be useful, for example, if the PLB system is integrated into a pre-existing robot cell with an interface that cannot be fully adapted. Or if the content of one and the same command differs depending on the time at which it is sent.
If characters other than alphanumeric characters are to be ignored in the command then you must specify these explicitly in the template file.

13.3.3.3 Trigger

Description

The `Trigger` command starts the image acquisition of stationary cameras.
See TriggerRM for robot mounted cameras.

Response

The command is responded to with a `TriggerMsg` message.

Parameters

`$imageId`
This parameter is used to assign a corresponding ID to an image. The image ID can be inserted in the `PartResultsMsg` message in order to track which results belong to which images.

`$cameraAlignmentAlias`
This parameter is used to tell the system which camera alignment alias that is to be used at image acquisition.

`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.4 TriggerRM

Description

The `TriggerRM` command starts the image acquisition of robot mounted cameras.
See Trigger for stationary cameras.

Response

The command is responded to with a `TriggerMsg` message.

Parameters

`$imageId`
This parameter is used to assign a corresponding ID to an image. The image ID can be inserted in the `PartResultsMsg` message in order to track which results belong to which images.
`$cameraAlignmentAlias`
This parameter is used to tell the system which camera alignment alias that is to be used at image acquisition.
`$flangeX, $flangeY, $flangeZ`
These parameters specify the robot flange position at the time of the image acquisition.
`$flangeR1, $flangeR2, $flangeR3`
These parameters specify the robot flange rotation at the time of the image acquisition. Rotations are described in selected Euler or Tait-Bryan rotation convention (see "Rotation conventions", page 53). It is possible to instead describe the robot flange rotation in rotation quaternions, see below.
`$flangeQX, $flangeQY, $flangeQZ, $flangeQW`
These parameters specify the robot flange rotation at the time of the image acquisition in rotation quaternions. It is possible to instead describe the robot flange rotation in Euler or Tait-Bryan angles, see above.
`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.5 LocatePart

Description

The `LocatePart` command starts the localization (and following calculations) of the part that is to be picked.

Response

The command is responded to with a `PartResultMsg` message.

Parameters

`$jobAlias`
This parameter is used to inform the system of the job that is to be used for the localization of the part.
`$gripperAlias`
Only enabled pick poses using the gripper with this alias is evaluated during localization of the part. All enabled pick poses will be evaluated if this parameter is missing.
`$binAlias`
This parameter is used to inform the system of which bin or search volume to use when locating the part.
`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.6 LocateChain

Description

You use the `LocateChain` command to trigger image acquisition and execute a chain of jobs. The chain stops when a job returns a valid pick pose for a part, or all jobs have been executed and no valid pick pose has been found.
If the command is used to pick multiple parts from the same image, the blacklist function must be enabled. Otherwise, a previously reported result may be reported again.
This command is used for stationary cameras. See LocateChainRM for robot mounted cameras.

Response

The command is responded to with a `PartResultMsg` message.

Parameters

`$triggerImage`

If `$triggerImage` is set to a value > 0, a new image acquisition is triggered. If the parameter is not set, it defaults to 1, which will also trigger new image acquisition.

`$triggerImage` = 0 will execute a chain of jobs on the previous image.

`$imageID`

This parameter is used to assign a corresponding ID to an image. The image ID can be inserted in the `PartResultsMsg` message in order to track which results belong to which images.

`$cameraAlignmentAlias`

This parameter is used to tell the system which camera alignment alias that is to be used at image acquisition.

`$locateBin`

If `$locateBin` = 1, a `LocateBin` command will be executed before each new job is located. If `$locateBin` = 0, the pre-configured bin or search volume will be used for each job in the chain. If the parameter is not set, it defaults to 0.

`$gripperAlias`

Only enabled pick poses using the gripper with this alias are evaluated during localization of the part. If this parameter is missing, all enabled pick poses are evaluated.

`$jobAliasChain`

Define the chain of jobs on the format:

`<JobAlias1>-<JobAlias2>-...-<JobAliasN>`

Any number of jobs can be used. If this parameter is empty, the currently selected job will be used as the only job in the job chain.

13.3.3.7 LocateChainRM

Description

You use the `LocateChainRM` command to trigger image acquisition and execute a chain of jobs. The chain stops when a job returns a valid pick pose for a part, or all jobs have been executed and no valid pick pose has been found.

If the command is used to pick multiple parts from the same image, the blacklist function must be enabled. Otherwise, a previously reported result may be reported again.

This command is used for robot mounted cameras. See LocateChain for stationary cameras.

Response

The command is responded to with a `PartResultMsg` message.

Parameters

`$triggerImage`
If `$triggerImage` is set to a value > 0, a new image acquisition is triggered. If the parameter is not set, it defaults to 1, which will also trigger new image acquisition.
`$triggerImage` = 0 will execute a chain of jobs on the previous image.
`$imageID`
This parameter is used to assign a corresponding ID to an image. The image ID can be inserted in the `PartResultsMsg` message in order to track which results belong to which images.
`$cameraAlignmentAlias`
This parameter is used to tell the system which camera alignment alias that is to be used at image acquisition.
`$flangeX, $flangeY, $flangeZ`
These parameters specify the robot flange position at the time of the image acquisition.
`$flangeR1, $flangeR2, $flangeR3`
These parameters specify the robot flange rotation at the time of the image acquisition. Rotations are described in selected Euler or Tait-Bryan rotation convention (see "Rotation conventions", page 53). It is possible to instead describe the robot flange rotation in rotation quaternions, see below.
`$flangeQX, $flangeQY, $flangeQZ, $flangeQW`
These parameters specify the robot flange rotation at the time of the image acquisition in rotation quaternions. It is possible to instead describe the robot flange rotation in Euler or Tait-Bryan angles, see above.
`$locateBin`
If `$locateBin` = 1, a `LocateBin` command will be executed before each new job is located. If `$locateBin` = 0, the pre-configured bin or search volume will be used for each job in the chain. If the parameter is not set, it defaults to 0.
`$gripperAlias`
Only enabled pick poses using the gripper with this alias are evaluated during localization of the part. If this parameter is missing, all enabled pick poses are evaluated.
`$jobAliasChain`
Define the chain of jobs on the format:
`<JobAlias1>-<JobAlias2>-...-<JobAliasN>`
Any number of jobs can be used. If this parameter is empty, the currently selected job will be used as the only job in the job chain.

13.3.3.8 LocateBin

Description

Use this command to locate the bin.

Response

The command is responded to with a `BinResultMsg` message.

Parameters

`$jobAlias`
This parameter is used to inform the system of the job that is to be used for the localization of the bin.
`$binAlias`
This parameter is used to inform the system of which bin or search volume to locate.
`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.9 AddAlignmentPoint

Description

The `AddAlignmentPoint` command is used to add another alignment target feature point in Align mode.
This command is used for stationary cameras. See AddAlignmentPointRM for robot mounted cameras.

Response The command is responded to with an `AddAlignmentPointMsg` message.

Parameters `$targetX,$targetY,$targetZ`
These parameters are used to inform the system of the position of the alignment target in the robot coordinate system.
`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.10 AddAlignmentPointRM

Description The `AddAlignmentPointRM` command is used to add another alignment target feature point in Align mode. Note that the alignment target must remain in the same position for all added alignment target feature points.
This command is used for robot mounted cameras. See AddAlignmentPoint for stationary cameras.

Response The command is responded to with an `AddAlignmentPointMsg` message.

Parameters `$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.11 AddAlignmentTargetPositionRM

Description The `AddAlignmentTargetPositionRM` command is used to add an alignment target position in Align mode. Note that the alignment target must remain in the same position for all added alignment target feature points.
This command is used for robots with limited rotation around all axes in conjunction with robot mounted cameras.

Response The command is responded to with a `AddAlignmentTargetPositionMsg`.

Parameters `$targetX,$targetY,$targetZ`
These parameters are used to inform the system of the position of the alignment target in the robot coordinate system.

13.3.3.12 SaveAlignment

Description The `SaveAlignment` command is used to save the alignment transform.

Response The command is responded to with a `SaveAlignmentMsg`.

Parameters `$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.13 VerifyAlignment

Description The `VerifyAlignment` command is used to verify the current alignment. It will calculate the distance between a located alignment target and the robot coordinates that are supplied as input to the command.
This command is used for stationary cameras. See VerifyAlignmentRM for robot mounted cameras.

Response The command is responded to with a `VerifyAlignmentMsg`.

Parameters

`$targetX,$targetY,$targetZ`
These parameters are used to inform the system of the position of the alignment target in the robot coordinate system.
`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.14 VerifyAlignmentRM

Description

The `VerifyAlignmentRM` command is used to verify the current alignment. It will locate the alignment target, and calculate the difference from the target position during hand-eye alignment.
Note that the alignment target must remain in the same position as during alignment.
This command is used for robot mounted cameras. See VerifyAlignment for stationary cameras.

Response

The command is responded to with a `VerifyAlignmentMsg`.

Parameters

`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.15 ClearBlacklist

Description

This command is used to clear a job blacklist. If the blacklist is enabled, it is recommended to clear it when replacing the bin.

Response

The command is responded to with a `ClearBlacklistMsg` message.

Parameters

`$jobAlias`
This parameter is used to inform the system of the job which blacklist is to be cleared.
`$ignore`
This parameter is used to ignore any desired alphanumeric characters in a command.

13.3.3.16 Log

Description

This command logs a message to the PLB log file `/ProgramData/SICK/PLB/System.log`. This may be useful for debugging purposes.

Response

The command is responded with a `LogMsg` message.

Parameters

`$logLevel`

Log level	$logLevel
Debug	0
Warning	1
Error	2
Fatal	3
Info	4

All messages will be written to the `System.log` file. Messages with log level 1 or above will also be displayed in the PLB Studio log window.
`$logMsg`
Free text message. The string must be ended with a # character.

13.3.3.17 SetJobParameter

Description PLB algorithm parameters can be configured via the robot interface with the `SetJobParameter` command. All parameters configurable in PLB Studio in Analyze mode are also configurable in Run mode using this command.

Response The command is responded to with a `SetJobParameterMsg` message.

Parameters `$jobAlias`
This parameter is used to inform the system of the job whose parameter is to be configured.
`$parameterName`
Location path (XPath) of the parameter within the job to be configured.

- Found in the job directory for the job to be configured, in the file Job.xml, within the PLB Workspace folder
- Using plug-in: Found in the job directory for the job to be configured, in the file /Plugins/<pluginName>.xml, within the PLB Workspace folder
- The path must include all applicable module names to the parameter and the specific parameter name, separated with /

`$parameterValue`
Set the parameter to this value.

Example **Using Edge or Surface algorithm:**
Configure the algorithm to stop calculation after 10 seconds for job `MySurfaceJob`.

$jobAlias	$parameterName	$parameter-Value
MySurfaceJob	/Algorithm/LOCALIZATION_TIMEOUT	10

Using plug-in:
Configure the plug-in `PluginExamplePeakPointCloud` for job `MyJob` to write result to file.

$jobAlias	$parameterName	$parameter-Value
MyJob	/PluginExamplePeakPointCloud/ResultFileModule/RESULT_LOG	true

13.3.3.18 SetCameraParameter

Description Some camera parameters can be configured via the robot interface with the `SetCameraParameter` command.

Response The command is responded to with a `SetCameraParameterMsg` message.

Parameters `$cameraAlignmentAlias`
This parameter is used to inform the system of the camera whose parameter is to be configured.
`$parameterName`
Location path (XPath) of the camera parameter to be configured.

- Found in the device directory for the camera to be configured, in the file \System\Devices\<Camera number>\Camera.xml, in the PLB Workspace folder
- The path must include all applicable module names to the parameter and the specific parameter name, separated with /

`$parameterValue`
Set the parameter to this value.

Example Set `ENABLED` to false, in order to disconnect the camera with camera alignment alias `MyCamera`.

$cameraAlignmentAlias	$parameterName	$parameter-Value
MyCamera	/CameraCommon/ENABLED	false

13.3.3.19 GetJobs

Description The `GetJobs` command returns a list of all job aliases of jobs that can be used in Run mode (valid jobs), or of jobs with incomplete configuration (non-valid jobs).

Response The `GetJobs` command is responded to with a `GetJobsMsg` message.

Parameters `$status`

Jobs status	`$status`
All valid jobs	1
All non-valid jobs	2

13.3.3.20 GetCameraAlignments

Description The `GetCameraAlignments` command returns a list of camera alignment aliases of all enabled and aligned cameras (valid cameras), or of all cameras with incomplete configuration (non-valid cameras).

Response The `GetCameraAlignments` command is responded to with a `GetCameraAlignmentsMsg` message.

Parameters `$status`

Camera status	`$status`
All valid cameras	1
All non-valid cameras	2

13.3.3.21 ExportSupportData

Description The ExportSupportData command is used to save the latest recording to the save recordings base folder.

Response The command is responded to with a `ExportSupportDataMsg`.

Parameters `$label`
This parameter is used to add additional information to the recording name. Label must be a single word without special characters.

13.3.3.22 DefinePickPose

Description A pick pose can be defined by locating a part, moving the robot gripper to the intended pick position, and sending the DefinePickPose command with the gripper TCP coordinates.

Response The command is responded to with a `DefinePickPoseMsg`.

Parameters

`$pickPoseAlias`
The alias of the new or updated pick pose.
`$gripperAlias`
The alias of the gripper that will be used by the pick pose.
`$absoluteToolFrameX, $absoluteToolFrameY, $absoluteToolFrameZ`
The gripper TCP position for the new or updated pick pose.
`$absoluteToolFrameR1, $absoluteToolFrameR2, $absoluteToolFrameR3`
The gripper TCP rotation for the pick pose. Rotations are described in selected Euler or Tait-Bryan rotation conventions (see see "Rotation conventions", page 53).
`$absoluteToolFrameQX, $absoluteToolFrameQY, $absoluteToolFrameQZ, $absoluteToolFrameQW`
The gripper TCP rotation for the pick pose. Rotations are described in rotation quaternions.

NOTE
It is possible to send the gripper TCP rotation for the pick pose in either Euler or Tait-Bryan angles or rotation quaternions.

13.3.4 PLB response messages

The PLB system can output the following messages to the robot controller:

- HeartbeatMsg
- StateMsg
- TriggerMsg
- PartResultMsg
- BinResultMsg
- AddAlignmentPointMsg
- AddAlignmentTargetPositionMsg
- SaveAlignmentMsg
- VerifyAlignmentMsg
- ClearBlacklistMsg
- LogMsg
- SetJobParameterMsg
- SetCameraParameterMsg
- GetJobsMsg
- GetCameraAlignmentsMsg
- ExportSupportDataMsg
- DefinePickPoseMsg

13.3.4.1 HeartbeatMsg

Description

This message is sent in response to a `Heartbeat` command and contains information about the current state.

Parameters

`$result`
This parameter is used to inform the robot controller of the result of the command. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.
`$state`
The current state.
`$rotationConvention`
This parameter contains the rotation convention currently set by the user in the PLB system. The parameter is in the format eXYZ (extrinsic XYZ rotation) or iXYZ (intrinsic X Y’ Z’’ rotation).

13.3.4.2 StateMsg

Description

This message is sent in response to a `SetState` command and contains information about the current system state. If the status has been modified successfully then the result reported in the message is 0 (OK).

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not the system state could be set successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

`$state`

This parameter is used to inform the robot controller of the current system state. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.4.3 TriggerMsg

Description

This message is sent in response to `Trigger` and `TriggerRM` commands.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not image acquisition could be performed successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

`$imageId`

This parameter contains the ID assigned to the acquired image.

`$cameraAlignmentAlias`

This parameter is used to tell the robot controller what camera alignment alias that was used at image acquisition.

13.3.4.4 PartResultMsg

Description

The message is sent in response to `LocatePart`, `LocateChain`, and `LocateChainRM` commands.

NOTE

For the position information parameters, rotations are given on the format R1, R2, R3. The corresponding rotation axis depends on the rotation convention set in "Setting up communications with the robot controller", page 51. For example:

- If the user has set x-y'-z'' (iXYZ) as rotation convention, R1 is the rotation around the global/local x-axis, R2 is the rotation around the new local y-axis, and R3 is the rotation around the new local z-axis.
- If the user has set z-y-x (eZYX) as rotation convention, R1 is the rotation around the global z-axis, R2 is the rotation around the global y-axis, and R3 is the rotation around the global x-axis.

The parameter `$rotationConvention` contains the current rotation convention, as described below.

Parameters

`$result`
This parameter is used to inform the robot controller of whether or not the part could be located successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

`$imageId`
This parameter contains the ID of the image used during localization.

`$jobAlias`
This parameter is used to inform the robot controller of the part that was located.

`$binAlias`
This parameter informs the robot controller of which bin or search volume was used when locating the part.

`$cameraAlignmentAlias`
This parameter is used to tell the robot controller which camera alignment alias was used at image acquisition.

`$partFrameX, $partFrameY, $partFrameZ, $partFrameR1, $partFrameR2, $partFrameR3`
These parameters contain the position information (reference frame) for the located part.

`$boundingBoxSizeX, $boundingBoxSizeY, $boundingBoxSizeZ`
These parameters are only valid for plug-in part localization results.

`$numValidPickPoses`
This parameter contains the number of available pick poses.

`$pickPoseAliasN`
This parameter contains the name of the validated pick poses. N is a number between 0 and 7.

`$rotationConvention`
This parameter contains the rotation convention currently set by the user in the PLB system. The parameter is in the format eXYZ (extrinsic XYZ rotation) or iXYZ (intrinsic X Y’ Z’’ rotation).

`$absoluteToolFrameNX, $absoluteToolFrameNY, $absoluteToolFrameNZ, $absoluteToolFrameNR1, $absoluteToolFrameNR2, $absoluteToolFrameNR3`
These parameters contain the absolute tool position information for the validated pick poses (combination of partFrame and relativeToolFrame). N is a number between 0 and 7.

`$absoluteExitToolFrameNX, $absoluteExitToolFrameNY, $absoluteExitToolFrameNZ, $absoluteExitToolFrameNR1, $absoluteExitToolFrameNR2, $absoluteExitToolFrameNR3`
These parameters contain the absolute position information for the Exit pose of the validated pick poses. N is a number between 0 and 7.

`$absoluteClearToolFrameNX, $absoluteClearToolFrameNY, $absoluteClearToolFrameNZ, $absoluteClearToolFrameNR1, $absoluteClearToolFrameNR2, $absoluteClearToolFrameNR3`
These parameters contain the absolute position information for the Clear pose of the validated pick poses. N is a number between 0 and 7.

`$absoluteEntryToolFrameNX, $absoluteEntryToolFrameNY, $absoluteEntryToolFrameNZ, $absoluteEntryToolFrameNR1, $absoluteEntryToolFrameNR2, $absoluteEntryToolFrameNR3`
These parameters contain the absolute position information for the Entry pose of the validated pick poses. N is a number between 0 and 7.

`$absoluteApproachToolFrameNX, $absoluteApproachToolFrameNY, $absoluteApproachToolFrameNZ, $absoluteApproachToolFrameNR1, $absoluteApproachToolFrameNR2, $absoluteApproachToolFrameNR3`
These parameters contain the absolute position information for the Approach pose of the validated pick poses. N is a number between 0 and 7.

`$relativeToolFrameNX, $relativeToolFrameNY, $relativeToolFrameNZ, $relativeToolFrameNR1, $relativeToolFrameNR2, $relativeToolFrameNR3`
These parameters contain the position information (reference frame) for the validated pick poses. N is a number between 0 and 7.
`$highestPointX, $highestPointY, $highestPointZ`
These parameters contain the coordinates of the highest point of the highest cluster of points in the bin.
`$binFillingRatio`
This parameter shows an estimate of how much of the bin is filled with parts.
`$moreToAnalyze`
This parameter contains information about zones that are available for analysis. The value 0 means that no further data is available for analysis.

13.3.4.5 BinResultMsg

Description
The message is sent in response to a `LocateBin` command.

Parameters
`$result`
This parameter is used to inform the robot controller of whether or not the bin could be located successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.
`$imageId`
This parameter is used to inform the robot controller of the image on which the analysis is based.
`$jobAlias`
This parameter is used to inform the robot controller of the active job.
`$binAlias`
This parameter informs the robot controller of which bin was used when locating the bin.
`$cameraAlignmentAlias`
This parameter is used to tell the robot controller which camera alignment alias that was used at image acquisition.
`$rotationConvention`
This parameter contains the rotation convention currently set by the user in the PLB system. The parameter is in the format eXYZ (extrinsic XYZ rotation) or iXYZ (intrinsic X Y' Z'' rotation).
`$binLength, $binWidth, $binHeight`
These parameters define the size of the located bin.
`$binX, $binY, $binZ, $binR1, $binR2, $binR3`
These parameters define the position and orientation of the located bin.
`$binScore`
This parameter specifies how accurately the bin could be located. It is dependent on the level of precision with which the search volume (bin) was configured, in particular in terms of the size of the bin and the thickness of the bin walls.
You have the option of setting a threshold value for bin localization in order to display whether or not bin localization was successful. The default values for the configuration of the search volume are only replaced by the current measurement values if the bin was located successfully.
`$binFillingRatio`
This parameter shows an estimate of how much of the bin is filled with parts.

13.3.4.6 AddAlignmentPointMsg

Description
The message is sent in response to `AddAlignmentPoint` and `AddAlignmentPointRM` commands.

Parameters

`$result`
This parameter is used to inform the robot controller of whether or not an alignment point could be added successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.
`$imageID`
This parameter contains the ID of the image used.
`$cameraAlignmentAlias`
This parameter is used to tell the robot controller which camera alignment alias that was used at image acquisition.

13.3.4.7 AddAlignmentTargetPositionMsg

Description

The message is sent in response to `AddAlignmentTargetPositionRM` command.

Parameters

`$result`
This parameter is used to inform the robot controller of whether or not the alignment target's position could be added successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.4.8 SaveAlignmentMsg

Description

The message is sent in response to a `SaveAlignment` command.

Parameters

`$result`
This parameter is used to inform the robot controller of whether or not alignment could be saved successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.
`cameraAlignmentAlias`
This parameter is used to tell the robot controller which camera alignment alias that was used at image acquisition.

13.3.4.9 VerifyAlignmentMsg

Description

The message is sent in response to `VerifyAlignment` and `VerifyAlignmentRM` commands.

Parameters

`$result`
This parameter is used to inform the robot controller of whether or not alignment could be verified successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.
`$imageID`
This parameter contains the ID of the image used.
`$cameraAlignmentAlias`
This parameter is used to tell the robot controller which camera alignment alias that was used at image acquisition.
`$targetX,$targetY,$targetZ`
These parameters are used to inform the system of the position of the located alignment target in the robot coordinate system. (Not the same as the robot coordinates supplied in the `VerifyAlignment` command.)
`$posError`
This parameter contains the distance between the located alignment target and the coordinates supplied as input in the `VerifyAlignment` command.

13.3.4.10 ClearBlacklistMsg

Description

The message is sent in response to a `ClearBlacklist` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not a blacklist could be cleared successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

`$jobAlias`

This parameter is used to inform the robot controller of the job in which the blacklist was cleared.

13.3.4.11 LogMsg

Description

This message is sent in response to a `Log` command.

Parameters

`$result`

This parameter is used to inform the robot controller of the result of the command. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.4.12 SetJobParameterMsg

Description

The message is sent in response to a `SetJobParameter` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not a parameter could be set successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.4.13 SetCameraParameterMsg

Description

The message is sent in response to a `SetCameraParameter` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not a parameter could be set successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.4.14 GetJobMsg

Description

The `GetJobMsg` message is sent in response to a `GetJobs` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not any jobs where found.

`$jobAliasList`

This parameter includes a list of job aliases of valid or non-valid jobs.

13.3.4.15 GetCameraAlignmentsMsg

Description

The `GetCameraAlignmentsMsg` message is sent in response to a `GetCameraAlignments` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not any camera alignments where found.

`$cameraAlignmentList`

This parameter includes a list of camera alignment aliases of valid or non-valid cameras.

13.3.4.16 ExportSupportDataMsg

Description

The message is sent in response to a `ExportSupportData` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not the recording could be saved successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.4.17 DefinePickPoseMsg

Description

The message is sent in response to a `DefinePickPose` command.

Parameters

`$result`

This parameter is used to inform the robot controller of whether or not the pick pose could be defined successfully. For information on the corresponding values, see "State values", page 167 and "Result codes", page 167.

13.3.5 State values

The following values are available to describe the system state:

Table 38: State values

Name	Number
Run	1
Configure	2
Align	3
Analyze	4

13.3.6 Result codes

The following codes are used in the result messages:

Table 39: Result codes

Code	Name	Description
0	OK	The command was executed successfully.
-1	Unexpected error	An unexpected error has occurred.
License error		
3	License error	A license error occurs if there is no enabled camera with a valid master license key, or if there is an enabled additional camera without a valid slave license key.
File error		
4	Save file error	A save file error occurs if the destination folder is invalid, or if PLB is missing correct access rights to write in the folder.
Job error		
21	Unknown job	The job (`$jobAlias` parameter) was not recognized.
22	Invalid job	The job configuration is invalid. For details, refer to the system log file.
Error on switching the system status		
31	Set state failed	The system could not be switched to the desired state. For details, refer to the system log file.
32	PLB busy	The system could not be switched to the desired state because it is still executing a trigger, LocateBin or LocatePart command.
Trigger error		

Code	Name	Description
41	No sensor connected	Image acquisition was not possible because the communication with the camera has not been established or the camera is not connected.
42	Invalid state	The command is invalid in the current state.
43	Overtriggered	A trigger command was received but cannot be processed because the system is already acquiring data.
44	Acquisition failed	Data acquisition failed.
Camera error		
51	Unknown camera	The camera alignment alias (`$cameraAlignmentAlias` parameter) was not recognized.
52	Invalid camera	The camera alignment configuration is invalid. For details, refer to the system log file.
53	Wrong camera mounting	The received command is not compatible with the mounting (stationary or robot) of the selected camera.
Error, failing to process command		
101	No image	No image is present for analysis.
102	Invalid state	The system has received a command but is unable to process it in the current state.
103	PLB busy	The system has received a command but is unable to process it because the system is still processing a preceding command.
104	Robot plugin locate failed	The plugin failed due to an exception or reported any other code than 0.
Result of the localization of the part		
111	No part located	It was not possible to locate a part.
112	Part overlapped	A part was located successfully but cannot be picked up due to overlaps.
113	Gripper collision	A part was found and the overlaps are located within the permitted tolerances. Despite this, the part cannot be picked up without collision.
114	Bin empty	The system could not locate any part because the number of measurement points is insufficient. This happens if the number of points is below the `Empty bin threshold` parameter or if the points are not clustered (to be considered as noise).
115	Unknown gripper	There is no gripper with this `$gripperAlias` for this job.
116	No enabled pick poses	There are no enabled pick poses using this `$gripperAlias` for this job.
117	Incompatible plugin data	Current image/point cloud does not contain all the data required by the selected plugin algorithm.
118	Result outside search volume	The valid result(s) are located outside the search volume.
Bin error		
121	Bin not found	The bin could not be located. The value of `$binScore` is below the defined threshold.

Code	Name	Description
122	Bin not defined	A configured bin with correct dimensions and wall top thickness is needed to locate a bin. It is not possible to locate search volumes.
123	Unknown bin	The `$binAlias` parameter is not defined for the current job.
Result of the hand-eye alignment		
131	Alignment target not found	The alignment target could not be located.
132	Failed to add alignment point	The alignment point could not be added; for example, if the current image has already been used to add an alignment point.
133	Too few points in alignment	There are too few points added to the alignment. Add more points and try again.
134	Alignment calculation failed	The alignment calculation failed; for example, if the added points are too similar to each other.
135	Threshold exceeded	The limit for re-projection error/position deviation was exceeded.
136	Too much scaling	The alignment results in a scaling mismatch between the detected cone positions and reported robot coordinates. The accepted scaling deviation is 2%. Make sure that the robot tool center point definition is accurate. If a non-standard cone is used, make sure the cone top angle setting is correct.
137	Insufficient variation of points	The variation in distance and tilt of the alignment target feature points are not enough for an accurate hand-eye alignment.
Result of parameter change		
141	Invalid parameter name	The parameter name path is invalid.
142	Invalid parameter value	The parameter could not be changed to the desired value.
143	Invalid parameter change occasion	It is not allowed to change this parameter in Run mode.
144	Read-only parameter	The parameter can not be changed.
Result of get configuration		
151	Invalid configuration option	An invalid configuration option is used, e.g., wrong `$status`.
Result of define pick pose		
161	No previous part localization found	A `DefinePickPose` command was received, but a previous `LocatePart` command cannot be found.
General error		
301	Invalid robot coordinates	The robot coordinates are invalid.
302	Unknown gripper	There is no gripper with this `$gripperAlias` for this job.
303	Unknown pick pose	There is no pick pose with this $pickPoseAlias for this job.

13.3.7 Robot program sequence

This section is applicable for stationary cameras, see "Mounting concepts", page 15.

Main robot program sequence

The following figures illustrate the possible appearance of a simple robot program that meets the various requirements. This can be used as the starting point for programming.

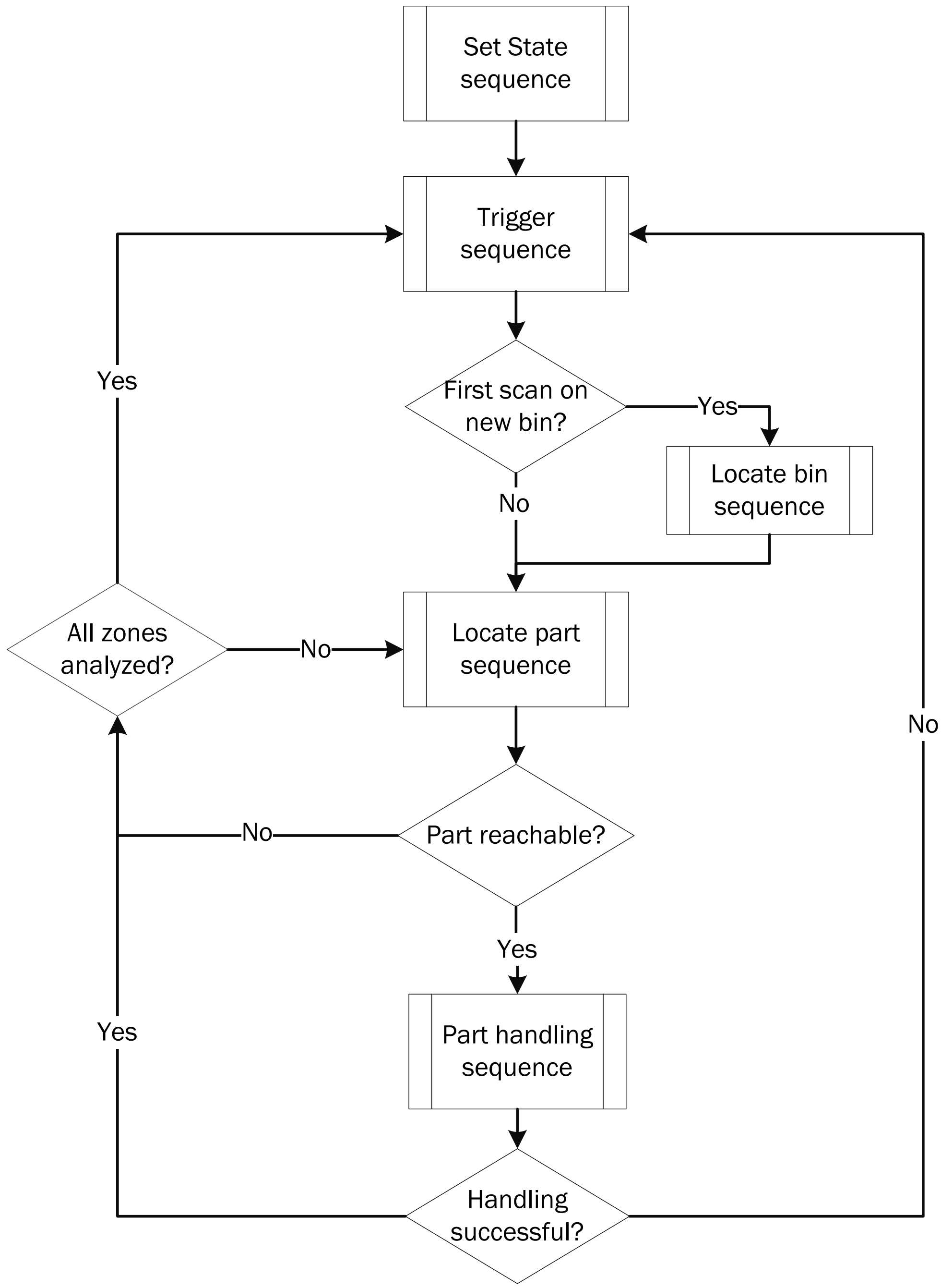


Figure 120: Simple robot program (overview)

After the initialization phase, PLB Engine is started and the connection to the TCP/IP server is established. The robot controller sends a `Start` command to the system. The system is now ready.

The main loop starts with the sending of the `Trigger` command to the system. This starts the image acquisition. After a command has been sent, the robot controller should wait for the response from the system.

If it is the first acquired image of a bin, the robot controller should send a `LocateBin` command to the system.

When the bin has been successfully located, the robot controller should send a `LocatePart` command to the system.

After the part has been picked up and transported, the robot controller sends a new `LocatePart` command if further data is still available for analysis. Otherwise, the robot controller sends a new `Trigger` command to the system.

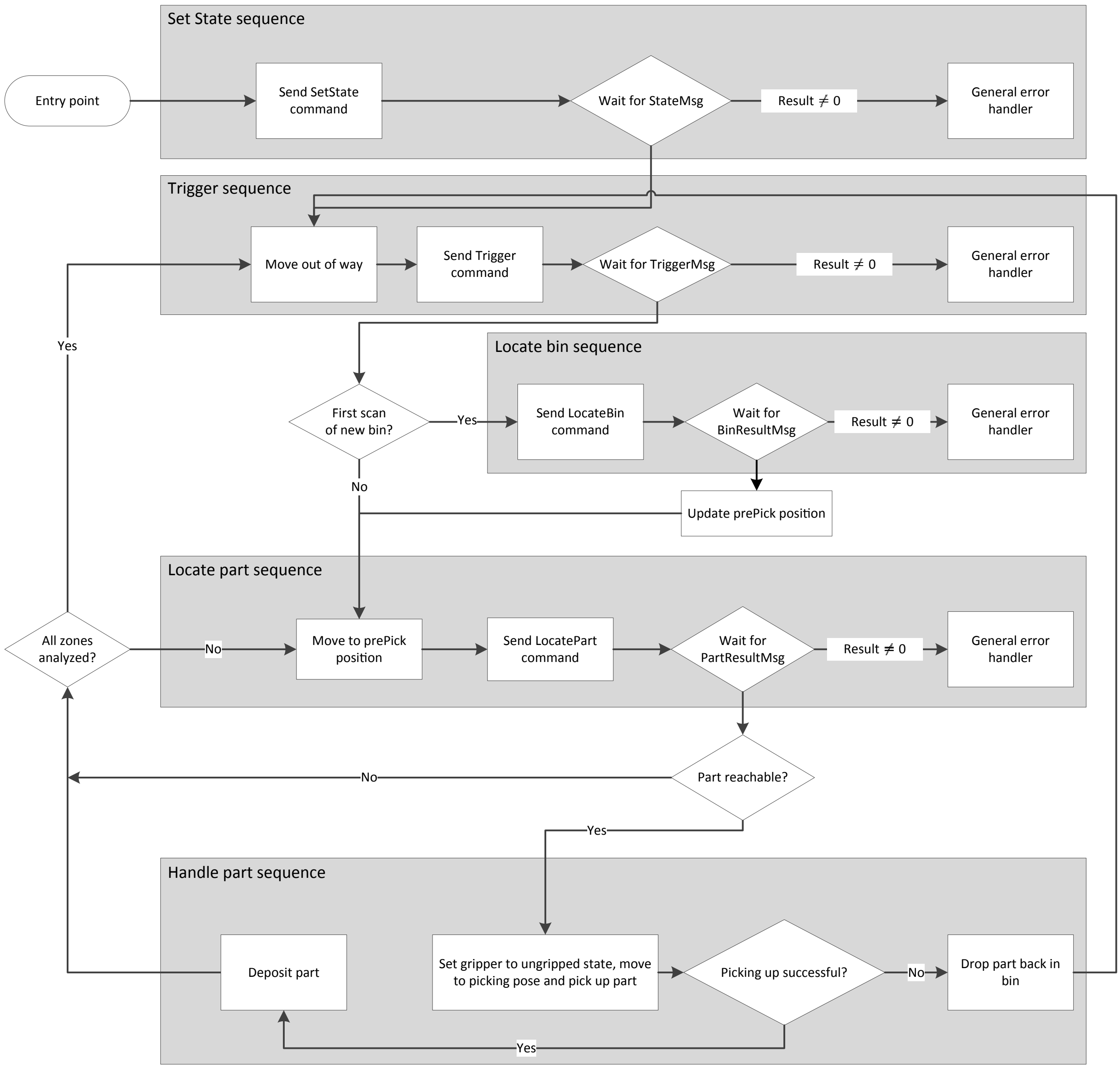


Figure 121: Robot program (detailed view)

You can configure the system to output a corresponding message as soon as an error occurs.

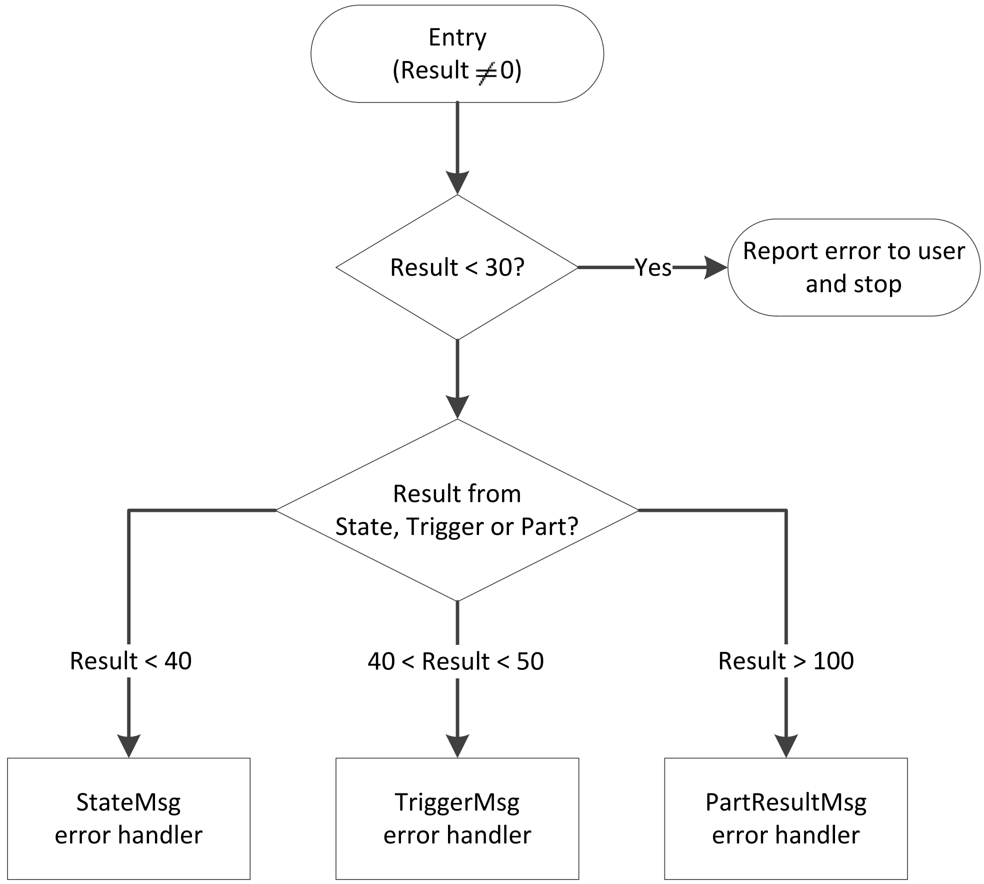


Figure 122: General error handling

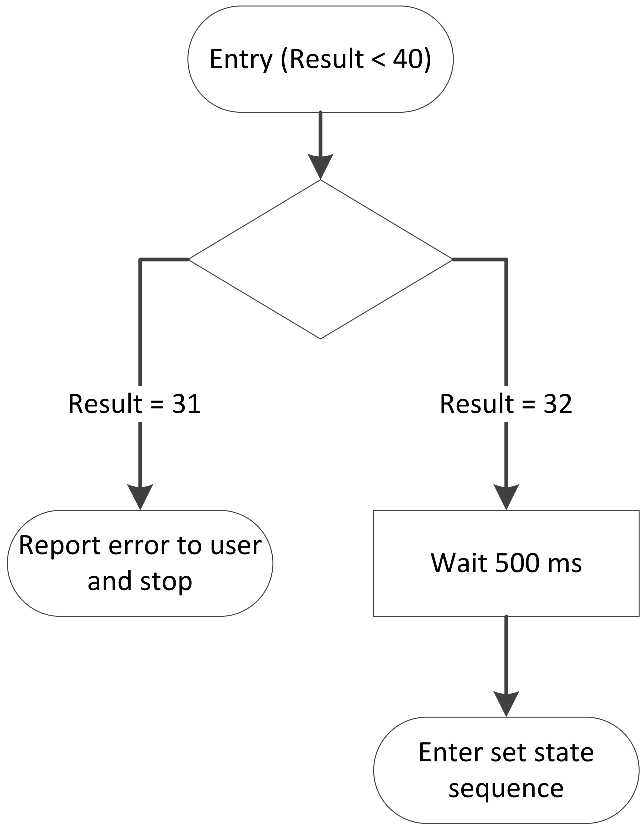


Figure 123: Procedure in the case of errors in StateMsg

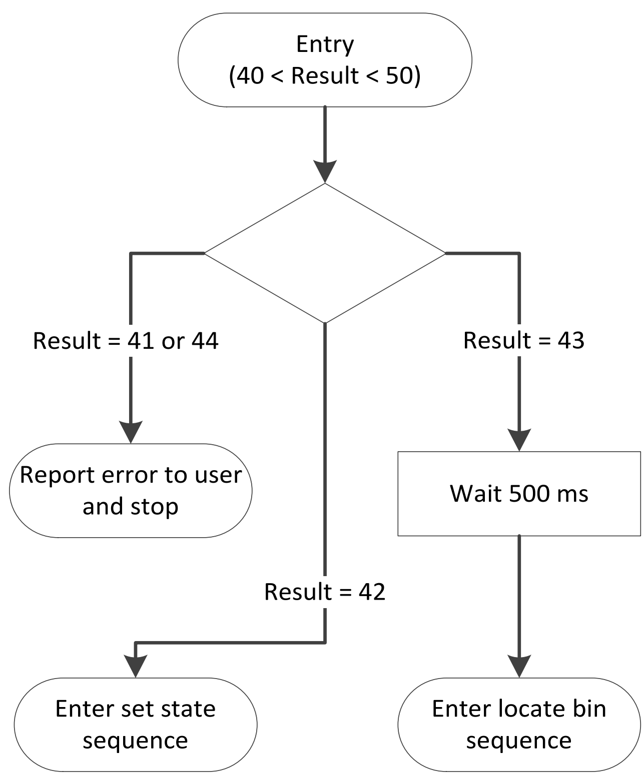


Figure 124: Procedure in the case of errors in TriggerMsg

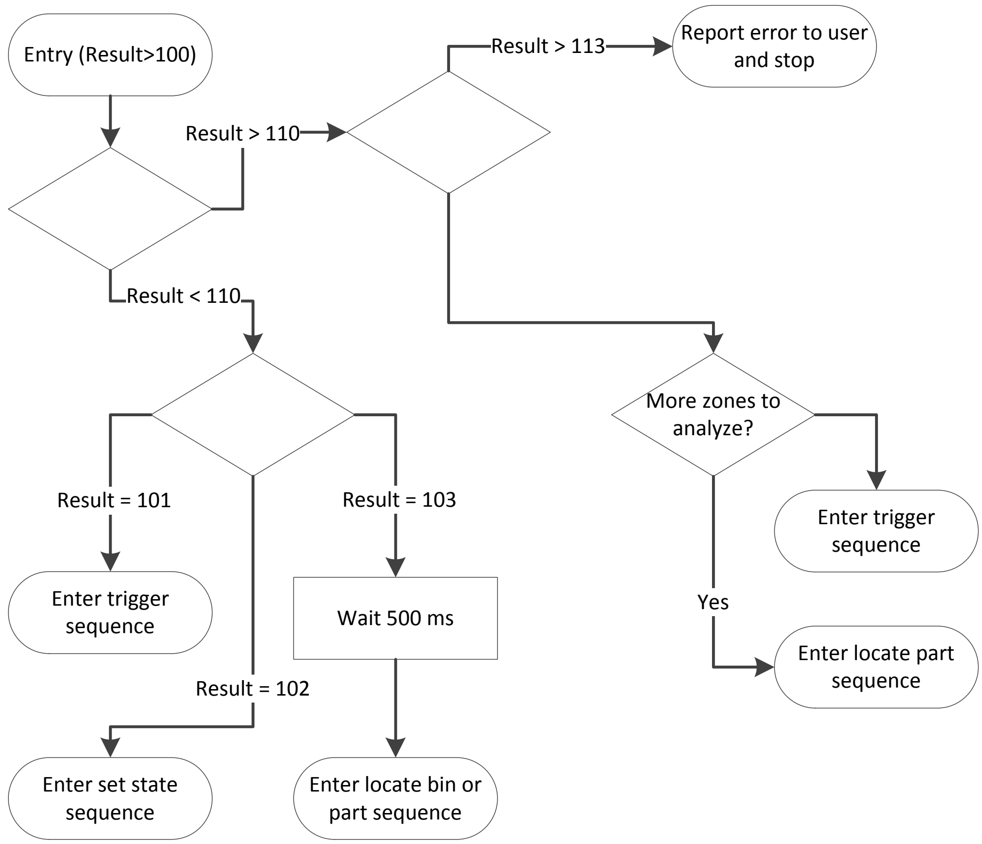


Figure 125: Procedure in the case of errors in PartResultMsg

Hand-eye alignment sequence

The following figures illustrate the possible appearance of a simple robot program that performs hand-eye alignment. This can be used as the starting point for programming.

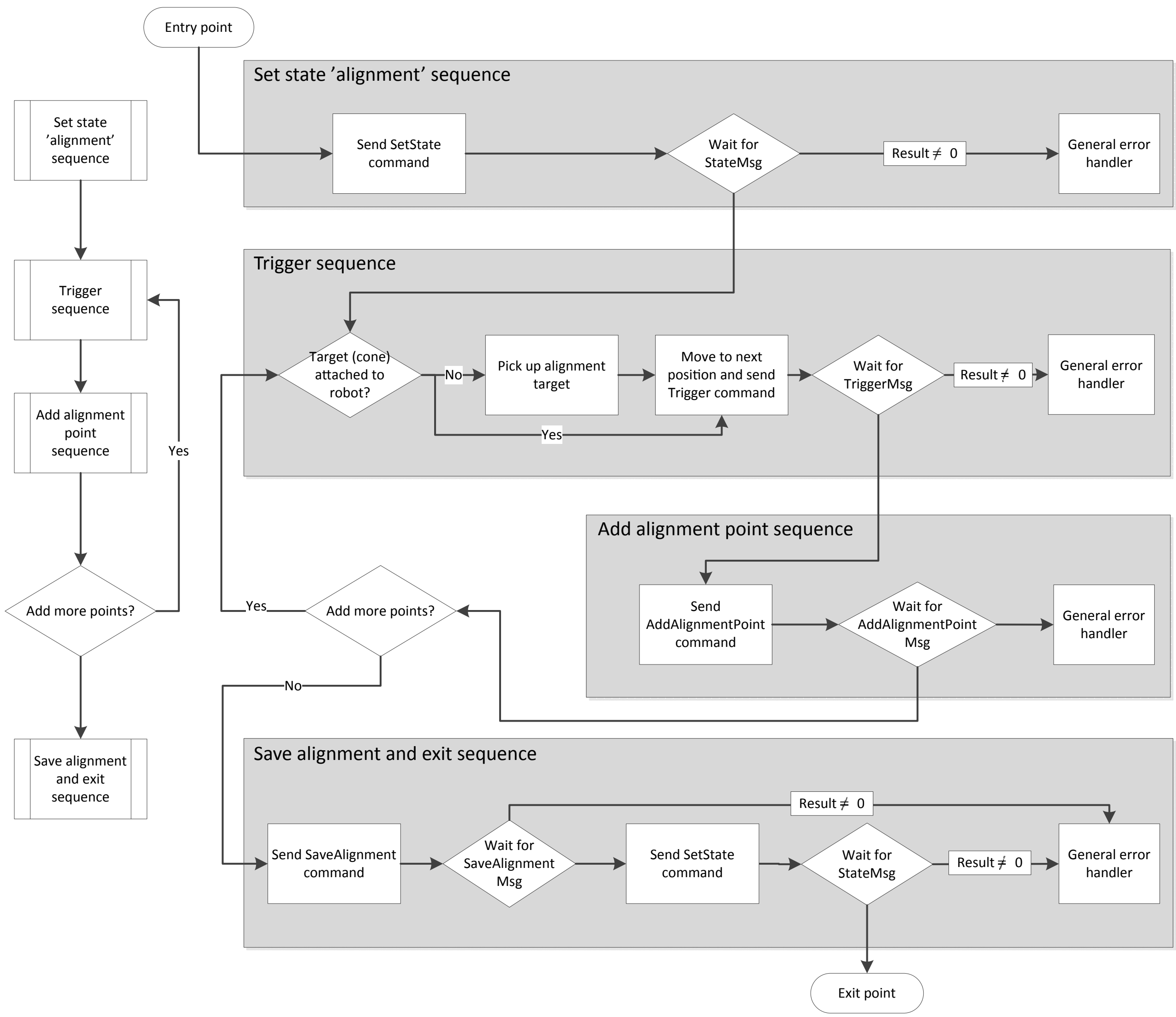


Figure 126: Hand-eye alignment sequence (overview and detailed view)

The robot controller sends a `SetState` command to the system, to set it in **Align** state. After a command has been sent, the robot controller should wait for a response from the system.

Make sure that the alignment target is attached to the robot and the correct tool is selected in the robot controller.

The loop starts with the sending of the `Trigger` command to the system. This starts the image acquisition. When the system has responded add the current point to the alignment calculations by sending an `AddAlignmentPoint` command.

Continue moving the target to new positions and adding alignment points until enough points are added. At least four points are needed to be able to calculate the alignment transform. The alignment points should fill out the search volume.

When all wanted points are added send a `SaveAlignment` command.

Verify alignment sequence

The accuracy of the alignment can, at any time, be verified in a verify alignment sequence. The following figure illustrates a possible appearance.

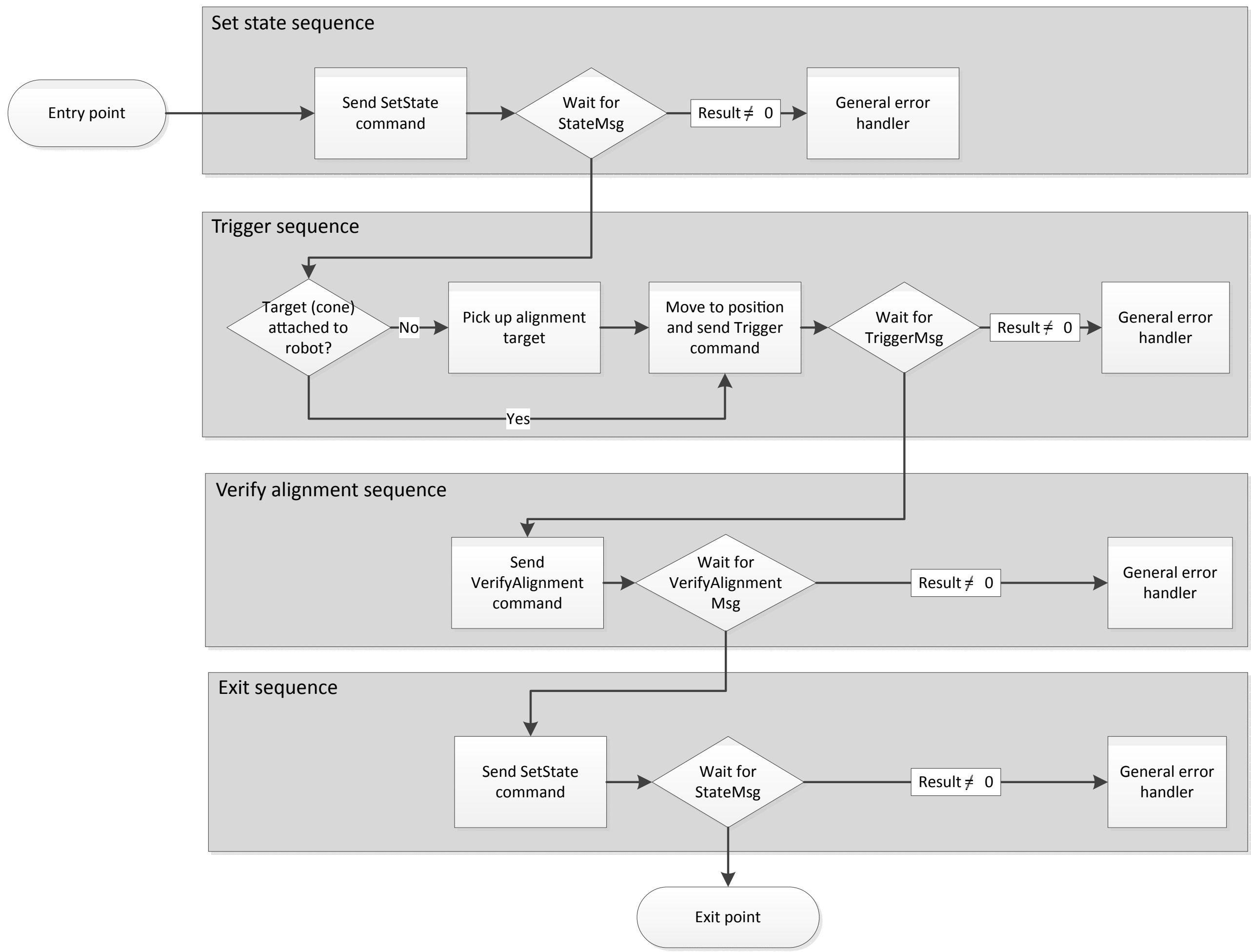


Figure 127: Verify alignment sequence

The robot controller sends a `SetState` command to the system, to set it in **Run** state. After a command has been sent, the robot controller should wait for a response from the system.

Make sure that the alignment target is attached to the robot and the correct tool is selected in the robot controller.

The loop starts with the sending of the `Trigger` command to the system. This starts the image acquisition. When the system has responded, send an `VerifyAlignment` command. The system will now calculate the distance between the input coordinates and the target that was located in the image. This deviation should normally be within the threshold set for the hand-eye alignment. If many points that were used during the alignment were close to the threshold it is likely that the result for `VerifyAlignment` is close to or above the threshold.

13.3.8 VASS command interface

This section is applicable for stationary cameras, see "Mounting concepts", page 15.

The PLB VASS interface provides functionality for the following mode IDs:

- 02 = Send life bit
- 20 = Calibration
- 21 = Calibration end
- 22 = Validation
- 23 = Validation end

- 40 = Measure bin
- 41 = Queue locate part
- 42 = Get locate part result sequence
- 53 = Block camera
- 54 = Unblock camera

Robot control via the VASS interface is only possible in Run mode.

To set up communications between the system and the robot controller, proceed as follows:

1. Open PLB Studio.
2. In the menu bar, click **Configure**.
3. In the parameter editing area, click **System settings**.
4. Go to the settings for robot communication (see figure 128, page 177).
5. Select VASS under the robot communication menu.
6. Enter the required port number in the corresponding text box.

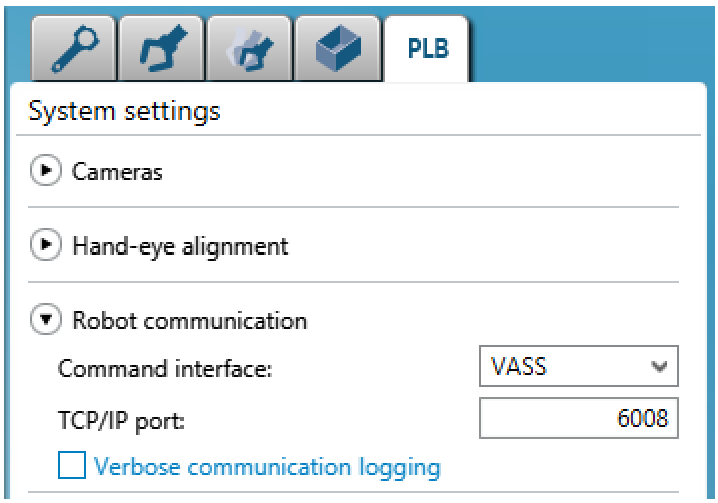


Figure 128: Specifying the VASS interface for communication with the robot controller

13.3.8.1 Calibration procedure

The calibration procedure is called Hand-Eye Alignment in PLB terminology, see "Overview hand-eye alignment", page 53 for a detailed description of the calibration procedure. To perform the calibration procedure with the VASS interface:

1. Mount a calibration cone with a known TCP on the robot.
2. The robot visits a number of preconfigured waypoints (typically 3 layers with 5 points in each layer). For each point, the robot sends mode ID 20 with the cone position defined in a fixed coordinate system in the <POS> tag. The system will trigger an image acquisition and locate the cone in each position.
3. When all positions have been located the robot sends mode ID 21. The calibration is saved and the PLB system will only report waypoints in the fixed coordinate system used by the robot.

13.3.8.2 Validation procedure

The calibration validation procedure works similarly to the calibration procedure.

1. Mode ID 22 is sent with the cone in different positions, with the same requirements as mode ID 20.
2. When the procedure is finished the robot sends a mode ID 23.

The PLB system then compares the maximum deviation of the points collected with a threshold value (the max alignment deviation value) and responds with an error if any of the points deviate more than the threshold value.

13.3.8.3 Locate bin procedure

The locate bin command is sent when a new bin is presented to the robot. The robot sends a mode ID 40 and the PLB system triggers an image acquisition and locates the bin.

In the response from PLB, one of the following status IDs is returned to indicate the status of the bin:

- 0 = Error (no bin located)
- 40 = Bin is OK
- 41 = Bin level is under 20%
- 51 = The bin is empty

For response messages with status ID 40, 41 or 51, the frame of the bin is returned in the <POS> tag. This frame can be used to calculate a suitable approach position for the robot above the centre of the bin.

13.3.8.4 Locate part procedure

The locate part procedure consists of two commands, mode ID 41 and 42.

Mode ID 41

When the robot sends mode ID 41, the PLB system triggers an image, checks the bin level, queues a part localization request and reports one of the following status IDs:

- 0 = Error
- 1 =Part localization request successfully added to queue

Mode ID 42

When the robot sends mode ID 42, the PLB system will start the locate part procedure. If no previous mode ID 41 has been sent, this command will send a mode ID 41 before starting the locate part procedure.

The robot controls the locate part procedure by sending several mode ID 42 commands. see table 40, page 178 below.

Table 40: Mode ID 42 commands

Command sequence	Response ID	Information
0	0, 40, 41, 51	Status of the bin, see "Locate bin procedure", page 178 for details.
1	61	Pick position number in the Y-axis attribute of the <POS> tag, and the gripper angle adjustment number in the Z-axis attribute of the <POS> tag. Note that the gripper angle adjustment number is just an ID, not the actual angle to be used.
2	60	Entry pose (above center of bin)
3	60	Approach pose (approach direction and distance from part to be picked)
4	62	Pick pose
5	60	Clear pose (vertical above pick pose)
6	63	Exit pose (above center of bin)

As soon as the robot is outside the field of view, a new mode ID 41 can be sent (in order to speed up cycle time).

13.4 Robot reach and work range

It is essential to make sure that the robot arm can reach the entire volume of the bin and that the gripper can pick up parts at all levels and in the corners.

During planning, you should note that the camera is fixed above the bin and may represent a static obstacle which must be avoided by the robot arm.

We recommend that you fill the bin with dummy parts. Then jog the robot gripper manually to five different positions at three levels in the bin (see figure 129, page 179).

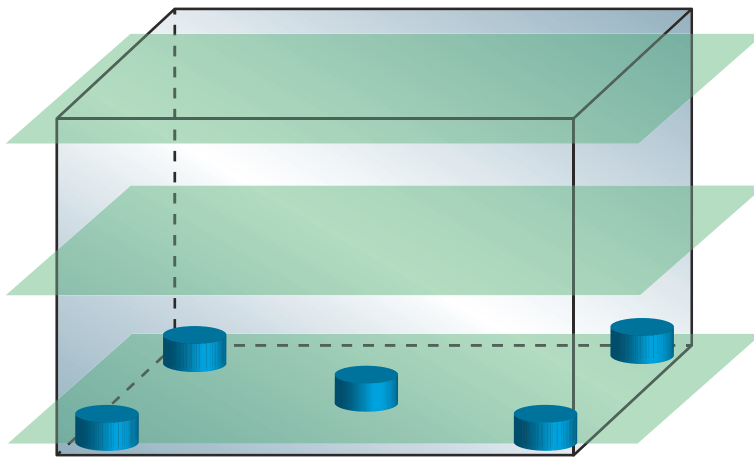


Figure 129: Testing the reach of the robot

If the gripper is not able to reach all the parts,

- Use a different robot model.
- Change the position of the robot in the cell. You can modify the robot's reach by placing it on a platform.
- Possibly consider a solution in which the robot gripper does not necessarily have to reach the entire bin volume, i.e. in which the bin is moved from time to time between image acquisitions.

13.5 Mathematical model of hand-eye alignment

During hand-eye alignment, each feature point is transformed as follows

$$\begin{pmatrix} x_{Robot} \\ y_{Robot} \\ z_{Robot} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_{Camera} \\ y_{Camera} \\ z_{Camera} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

The 3x3 matrix A describes any rotation with scaling between two coordinate systems. The translation vector t describes a translation.

If matrix A is a pure rotation matrix then the equation system describes a rigid transformation achieved by means of rotation and translation. Otherwise the equation system describes an affine transformation achieved by means of rotation, translation, scaling and shearing. Although shears do not usually occur, they nevertheless make it possible to partially compensate for small robot or camera measurement inaccuracies.

The transform is defined by determining the camera and robot coordinates for a number of feature points in the application scenario.

The equation system has 12 degrees of freedom if the affine transformation matrix is used and only 6 if the rigid transformation matrix is used.

Each feature point generates three equations. This means that the measurement of five feature points provides enough information to solve the least-squares problem and calculate optimum values for A and t.

If more than four feature points are measured then the calculation is more precise and stable against measurement noise and errors.

To determine the transform you need at least four feature points. However, we recommend that you do not use fewer than the 15 points recommended in "Overview hand-eye alignment", page 53.

13.6 Gripper design for bin picking applications

The design of the gripper plays an important role in determining the functional capability of the robot cell. Unlike in conventional applications, the position of the parts is not known in advance because they are randomly oriented in the bin.

To illustrate the requirements placed on the gripper, we recommend that you fill the bin with parts and then remove them one by one manually as if in production operation.

Consider how the gripper must be constructed in order to pick up the parts in all the possible positions and orientations without collision.

The basic gripper characteristics are as follows:

- It must be able to reach and pick up all the parts irrespective of their location in the bin.
- It must be able to pick up one and the same part in different ways.
- It must be large and robust enough to lift the part.
- Its base plate must be as small as possible to prevent it colliding with parts if a part is picked up at an oblique angle.

NOTE
The longer the fingers of the gripper are, the further the gripper base plate is from the parts that are to be picked up. As a result, parts are picked up at more oblique angles and this makes it easier to avoid collisions with other parts.

13.7 Collision-free path planning

Since the precise robot path is also not known in advance, path planning constitutes an additional challenge in bin picking applications. On the one hand, singularities must be avoided. On the other, the robot arm and gripper must have enough space to reach into the bin and then travel back out of the bin with the part without colliding with other parts, the bin walls or other obstacles.

When writing the robot program, position a reference point over the middle of the bin. The reference point acts as the start and end point for the movement into and out of the bin.

Choose the distance between the reference point and the bin in such a way that the robot arm, gripper and part can be moved into or out of the bin without this resulting in any collisions.

Plan a static path to approach and then withdraw from the reference point (see figure 130, page 181).

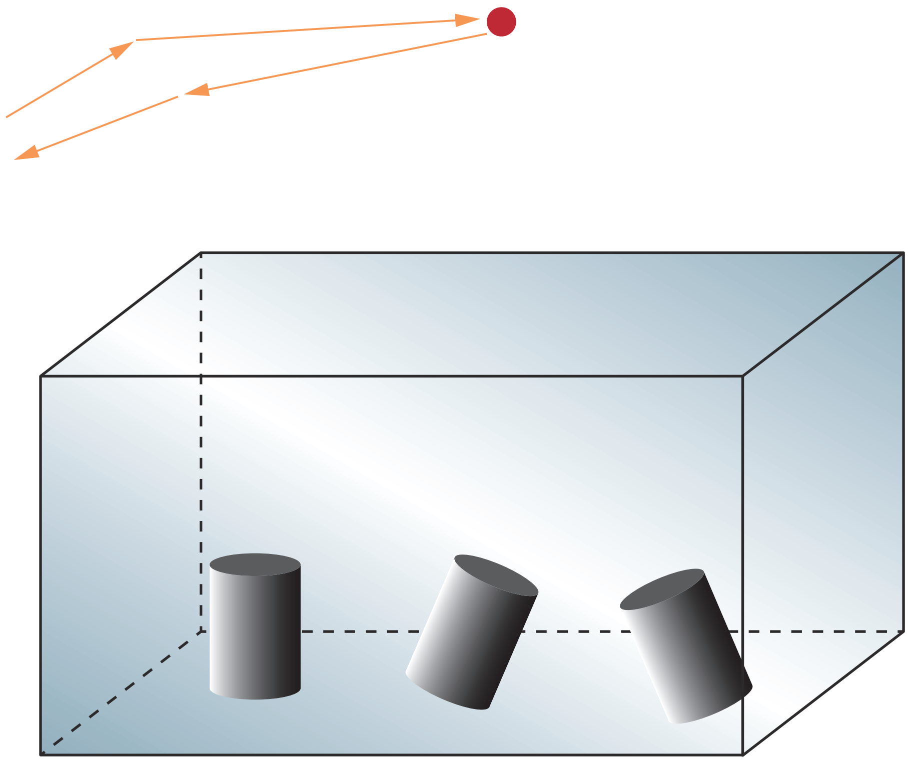


Figure 130: Movement over the bin (static path)

Plan a dynamic path to move from the reference point to the pick pose (see figure 131, page 181). The path trajectory depends on the position of the part in the bin.

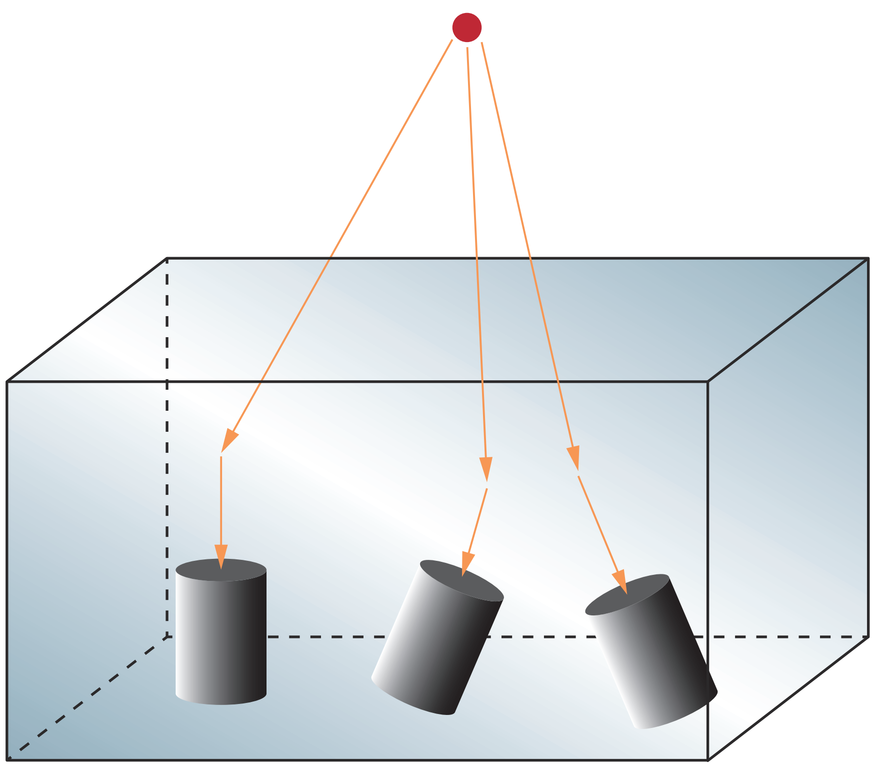


Figure 131: Movement into the bin (dynamic path)

The direction of movement when leaving the bin depends on the shape of the part. Depending on the shape, it may be better to use the part coordinate system or the world coordinate system (see figure 132, page 182). After the part has been lifted, move the gripper to the central point and then travel away from the bin (see figure 130, page 181).

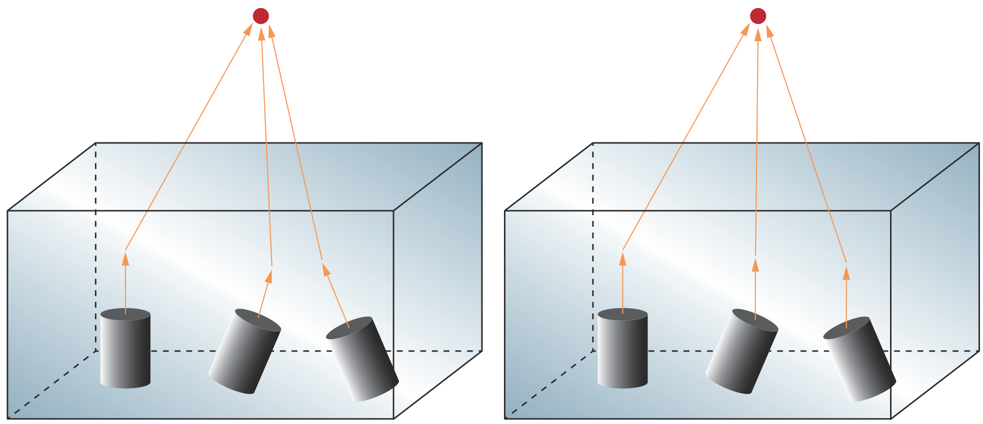


Figure 132: Movement out of the bin. Left: in the part coordinate system; right: in the world coordinate system

13.8 Flow charts

figure 133 shows the steps that are necessary in order to set up the system for production operation.

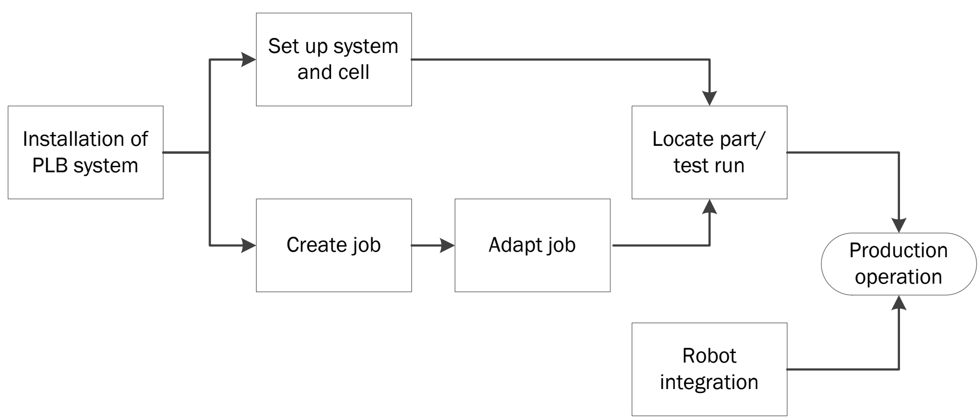


Figure 133: Workflow for production operation

figure 134 shows the procedure for testing and adapting parameters.

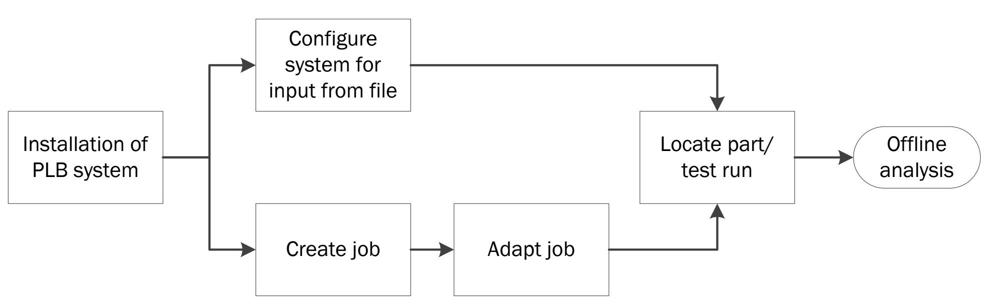


Figure 134: Workflow for analyze mode

figure 135 shows the configuration steps applicable to parts, grippers and pick poses.

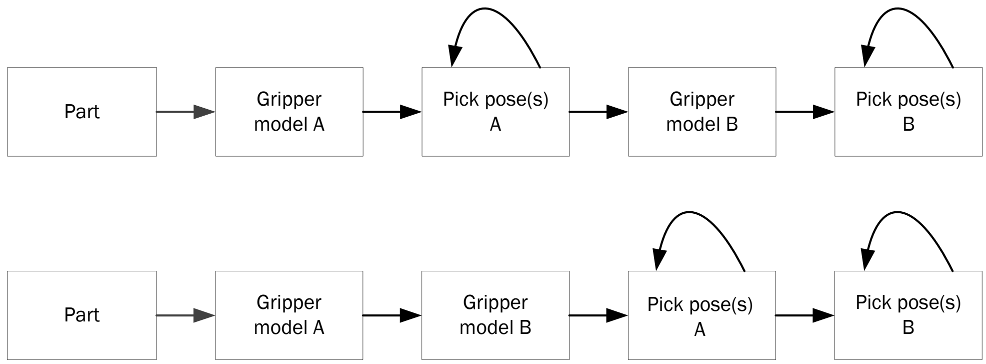


Figure 135: Workflow for job configuration (variants)

figure 136 shows the computation loop that is worked through before the system outputs the position of the part and the corresponding pick poses to the robot controller. If, after a specified number of attempts, the system is not able to locate a part then localization is aborted with an error message.

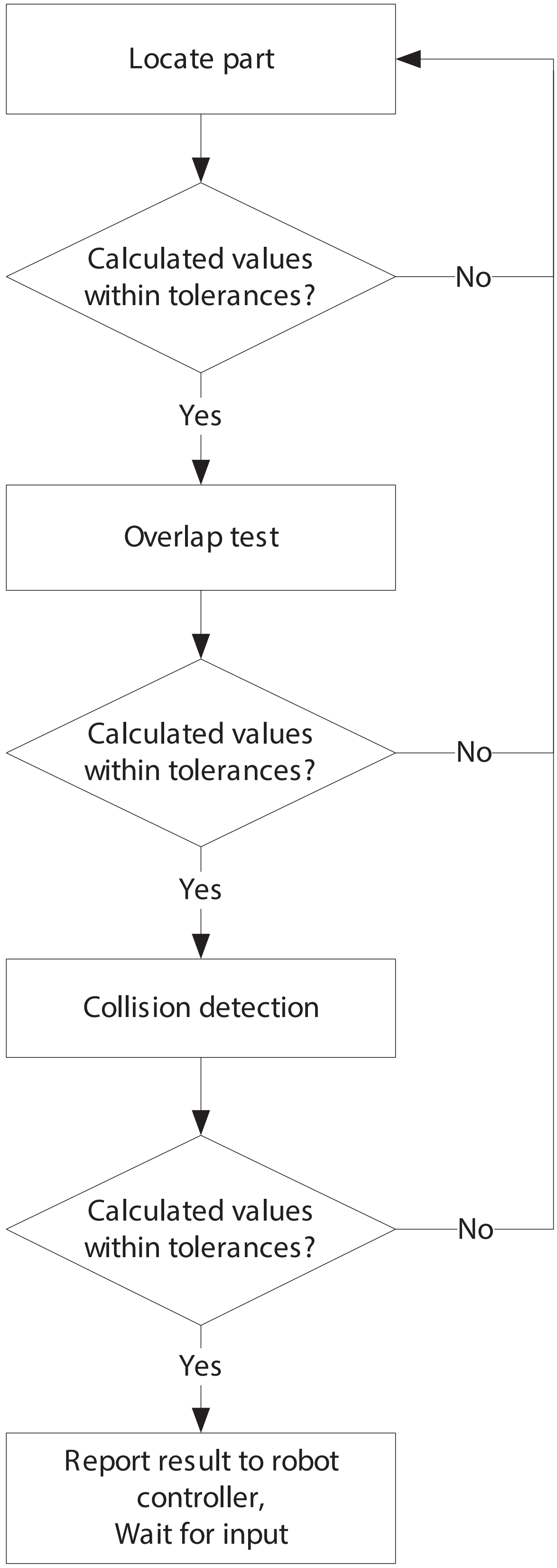


Figure 136: Sequence of calculations performed for localization, overlap testing and collision detection

8014916/2024-05/en

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 01
E-Mail info@sick.de

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972-4-6881000
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mario.garcia@sick.com

Netherlands
Phone +31 (0) 30 229 25 44
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 (0)11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321
E-Mail info@sickkorea.net

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail info@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com

Sensor Intelligence.