

Robotics and automation

Shayan Moradkhani, Fernando Ubis

Session 4/6 Notes

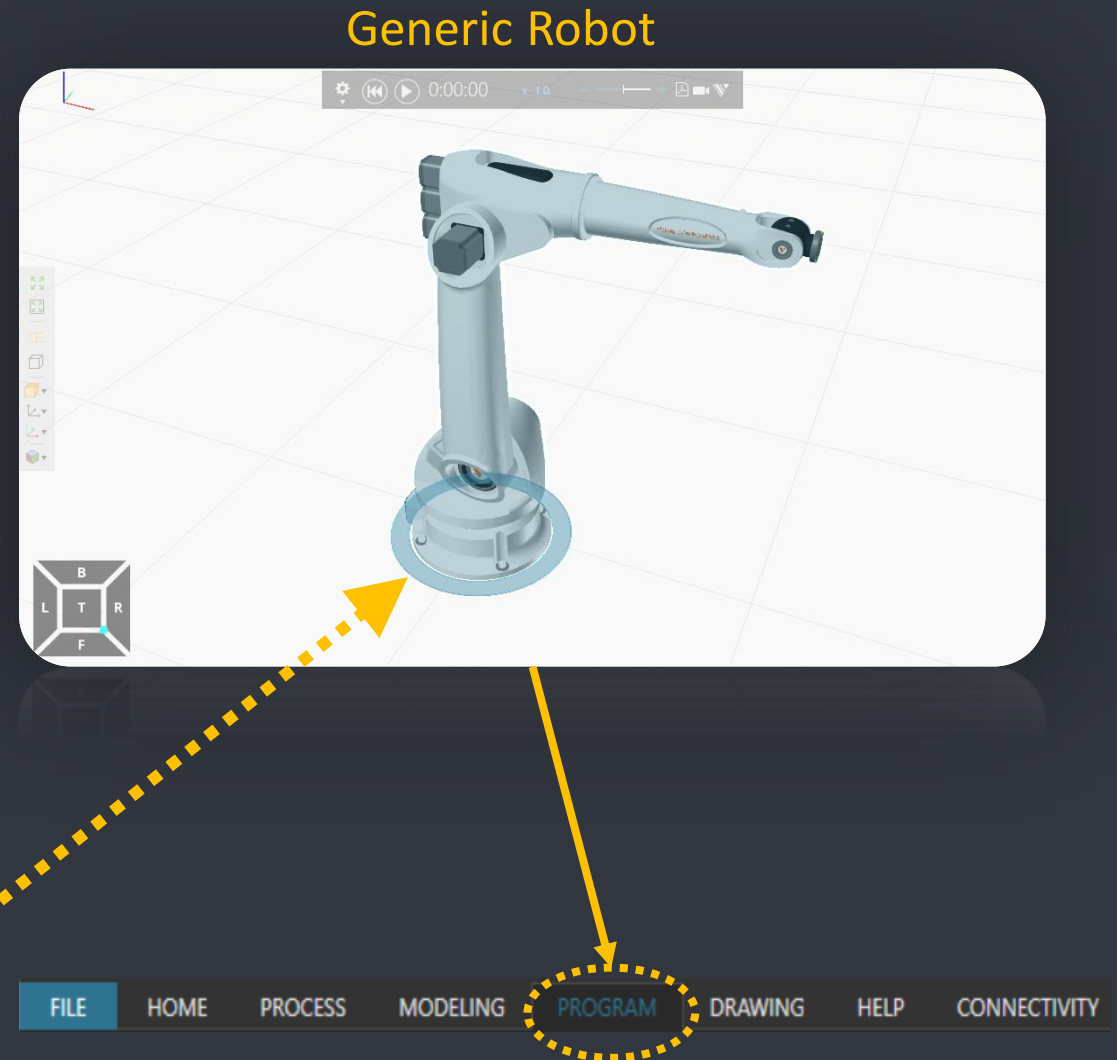
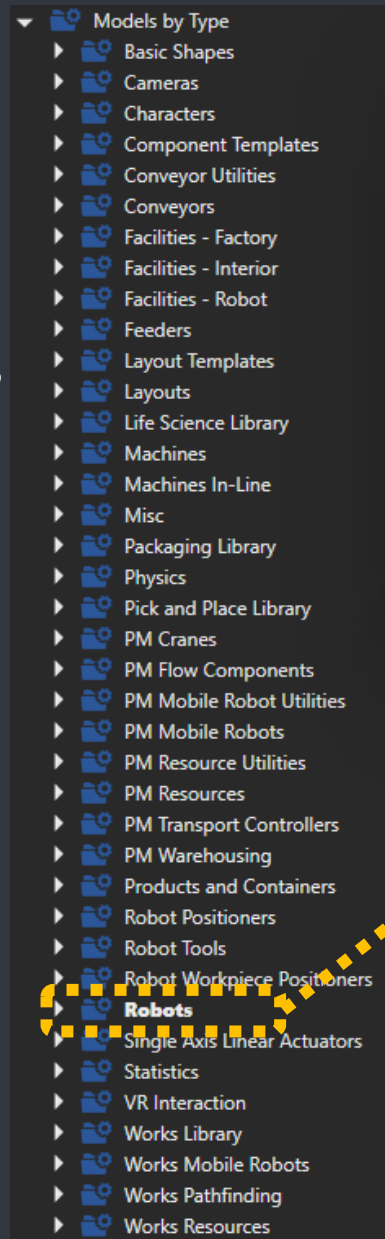
Optimai 10-03-2022



4.4

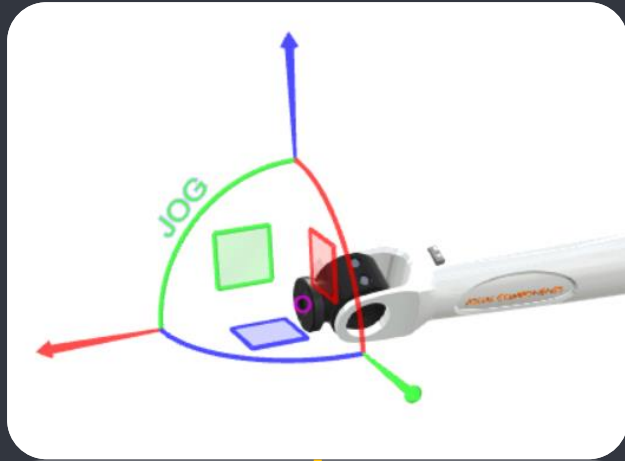
PROGRAM TAB

- 1) TEACHING TARGET POINTS
- 2) PTP VS LIN
- 3) MOVE VS JOG
- 4) EDITING TARGET POINTS
- 5) TOOL CENTER POINT AND BASE
- 6) MAPPING SIGNALS TO ROBOT ACTIONS
- 7) GRASPING
- 8) RELEASING
- 9) MULTI GRASPING AND RELEASING
- 10) POINTS REFERENCE
- 11) SIGNALS
- 12) SEQUENCES
- 13) ROUTINE PROPERTIES AND VARIABLES
- 14) STATEMENT PROPERTIES
- 15) SAVING A ROBOT PROGRAM



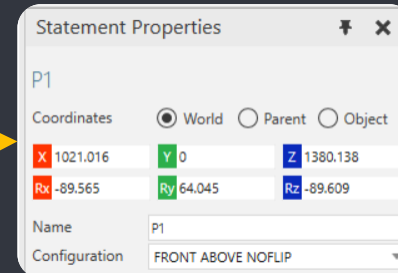
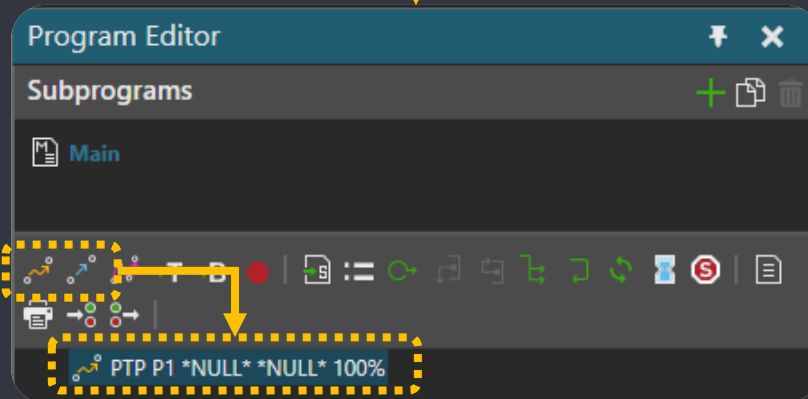
TEACHING TARGET POINTS

1



1. Use **Jog** tool to jog robot to a posture
 - Drag the TCP
 - Interact the joints
 - Use snap and align commands if needed
2. Add a motion statement (target point)
 - Point-To-Point (**PTP**) or
 - Linear (**LIN**) motion type

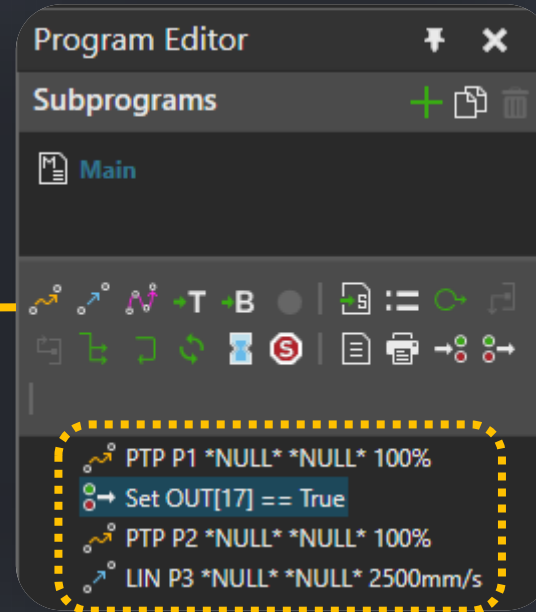
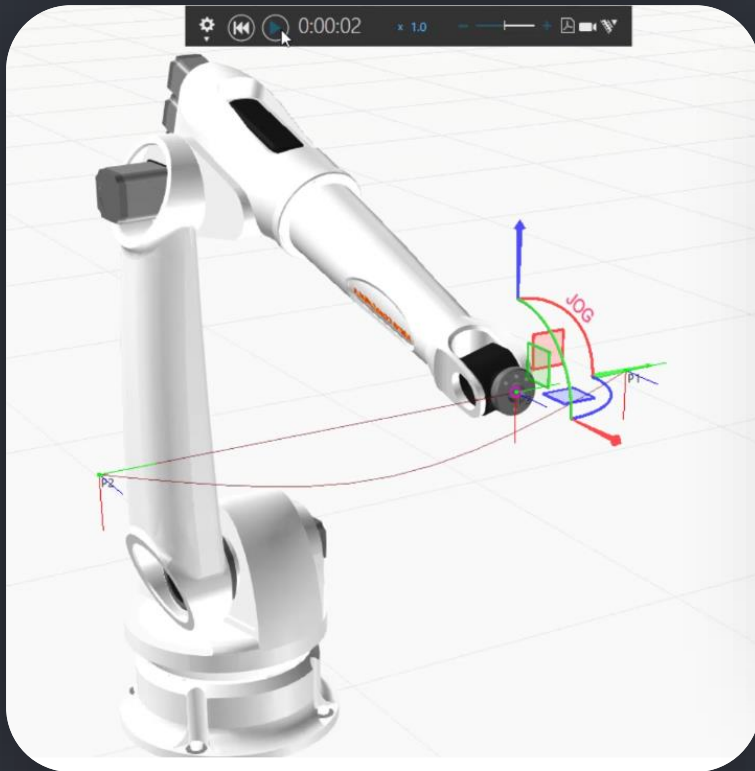
2



3

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

PTP VS LIN



PTP



- Joint interpolated motion where all the joints move to the target point values at the same time
- Quick movements
- Possible change of configuration

LIN



- Linear interpolated motion
- Tool Center Point (TCP) follows the straightest path to the target point
- Used in approach movements

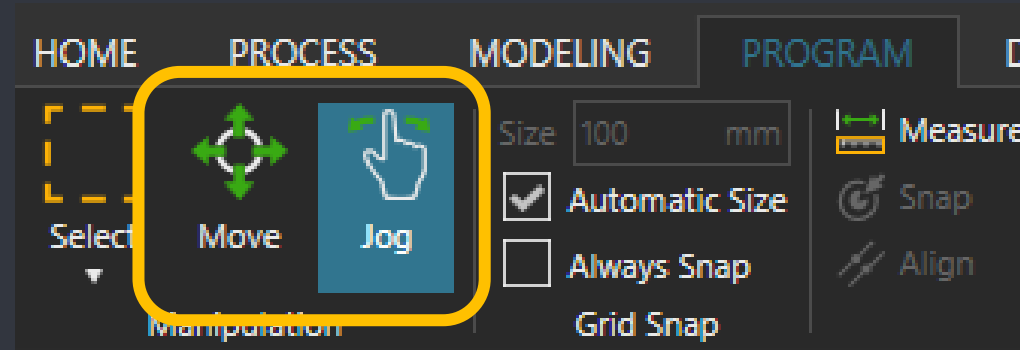
- 1) Teaching Target Points
- 2) **PTP vs LIN**
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

Hint: Activate to show selected frame types



Program Tab

MOVE VS JOG



Jog

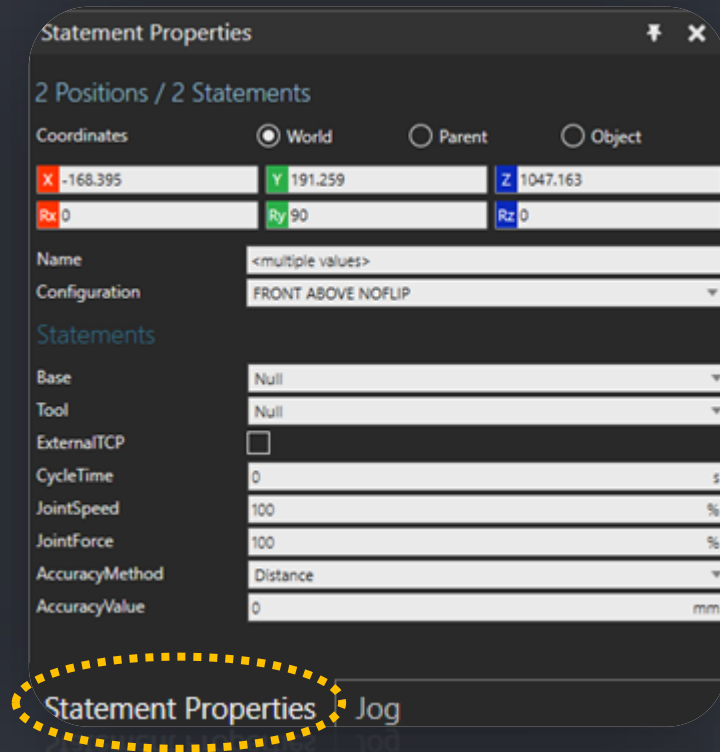
- Manipulate robot and other devices
- 3D handle shown at the active TCP
- Touch up required to update program points

Move

- Move selected item(s)
- Point, Component, Base, Tool
- Moving a program point edits the point immediately

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) **MOVE vs JOG**
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

EDITING TARGET POINTS

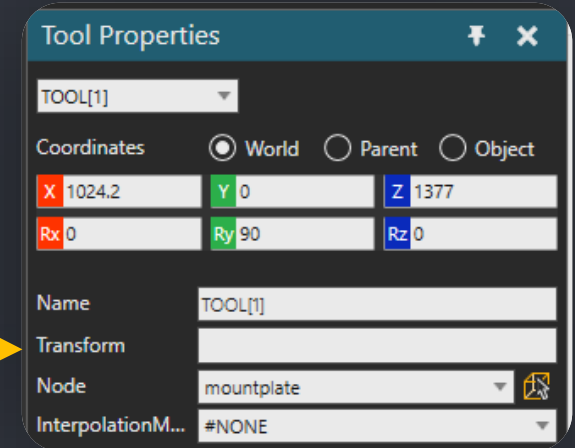
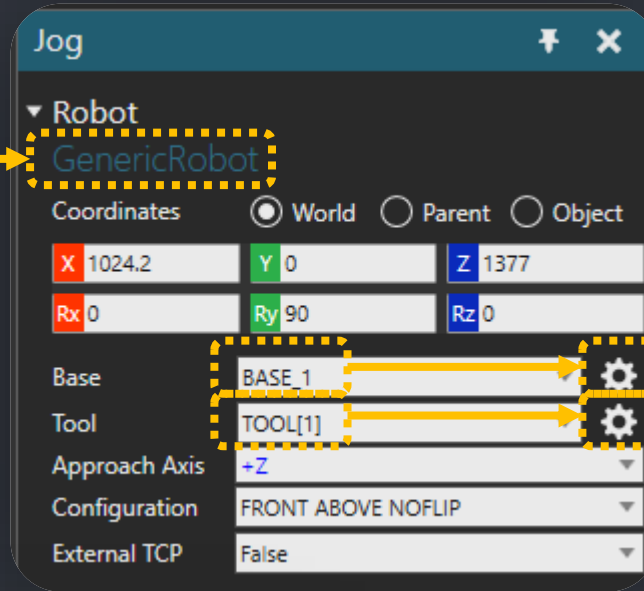


- Use **Move** tool to relocate single or multiple points location
- Select single or multiple points to edit their properties like
 - Base
 - Tool
 - Speed

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) **Editing Target Points**
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

TOOL CENTER POINT AND BASE

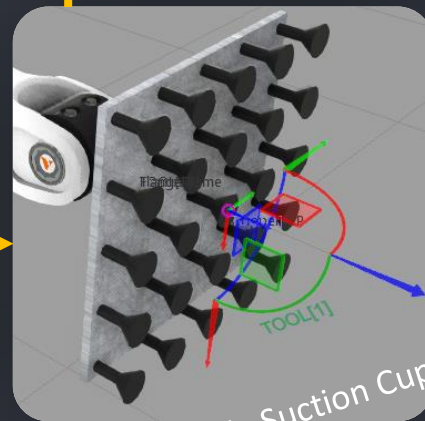
- Use **Move** tool to define tool location
 - E.g. While dragging an arrow handle of manipulator for a selected object in 3D world, hold down CTRL, and then point to geometry to snap axis to that position
- Attach to a **Node**



Models by Manufacturer

Visual Components

Robot Tools

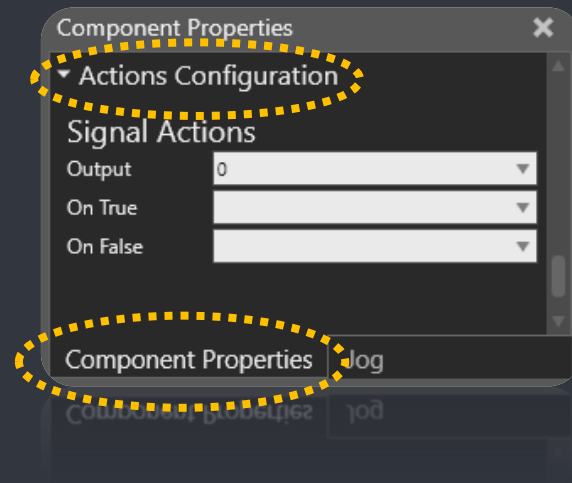


Parametric Suction Cup Gripper

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) **Tool Center Point and BASE**
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

MAPPING SIGNALS TO ROBOT ACTIONS

Click on *Generic Robot*:



- Map output signals to actions in **Signal Actions configurator**
- Default robot output signal ports
 - Most of the robots components have an Action Script behavior which automatically maps signals 1 to 48 to tool frames and signals 49 to 80 to base frames.

RULE OF THUMB:
First 100 signals are reserved for standard actions

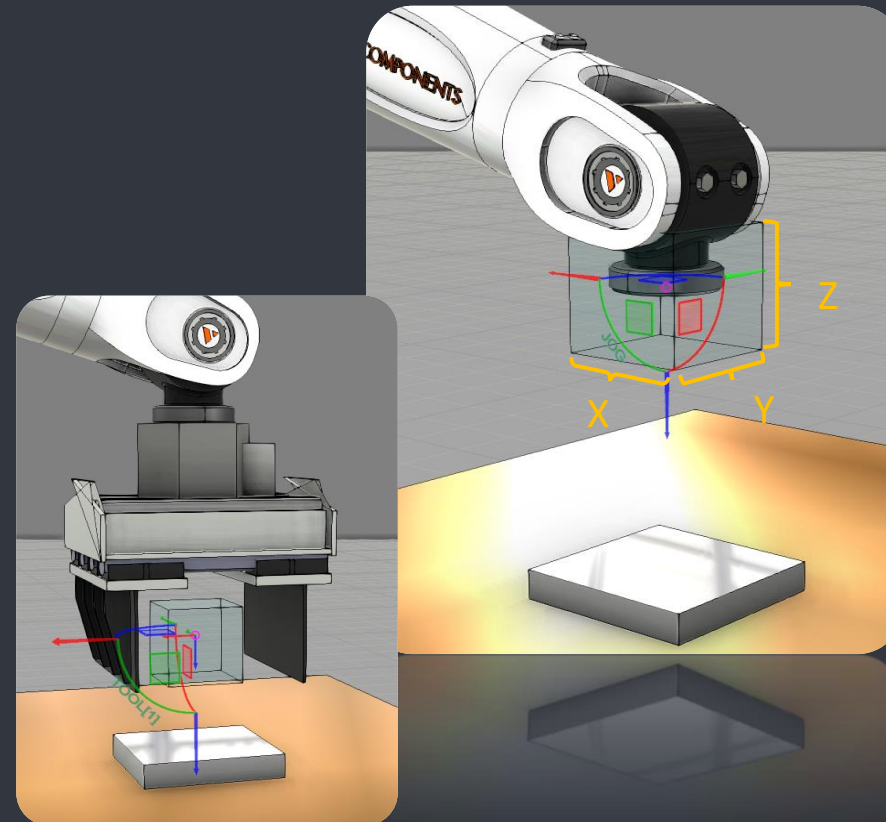
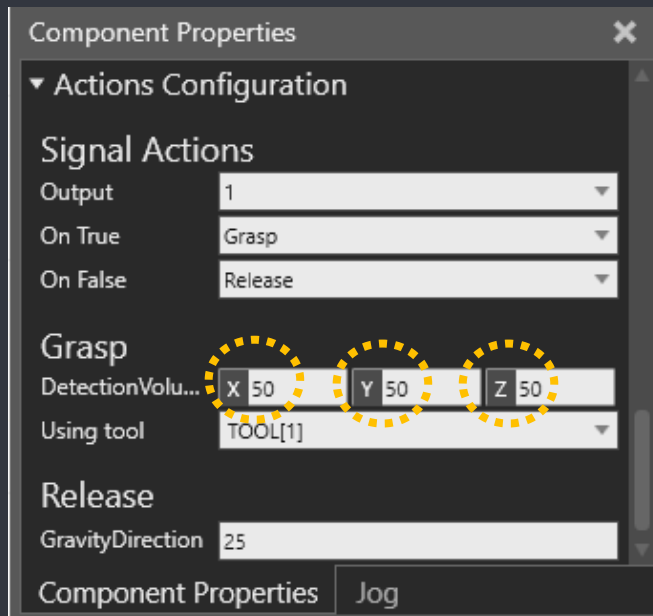
| Output # | Action On True | Action On False |
|----------|------------------------|-------------------------|
| 1...16 | Grasp | Release |
| 17...32 | Trace Tool On | Trace Tool Off |
| 33...48 | Mount Tool | Unmount Tool |
| 49...80 | Trace External Tool On | Trace External Tool Off |
| 81 | Start Swept Volume | Stop Swept Volume |

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions**
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

GRASPING

➤ "Imaginary" detection volume is used to detect nodes within the given 3D space for grasping

- The given volume is **symmetric** to the used **TCP** point



- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping**
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

GRASPING

➤ Detection volume can be visualized in SignalActions tab

➤ No affect for Actions Configurations
=> Only for visualization!

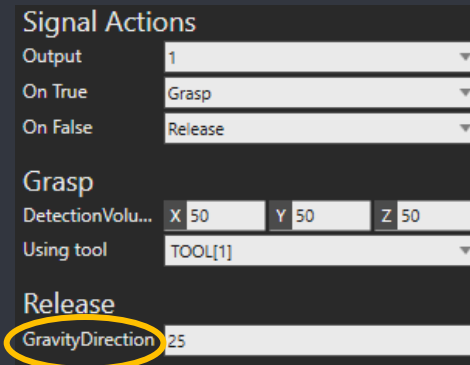
➤ You can list components that robot **must not pick** in ExcludeGrasping

| Default | Executor | WorkSpace | SignalActions |
|----------------------|-------------------------------------|-----------|---------------|
| DisplayMessages | <input type="checkbox"/> | | |
| ReleaseToWorld | <input type="checkbox"/> | | |
| ReleaseToPhysics | Never | | |
| MultiGrasp | <input type="checkbox"/> | | |
| TraceWidth | 2 | | |
| TraceZOffset | 0 | | |
| UpdateWorldOnRel... | <input type="checkbox"/> | | |
| DetectLockedCom... | <input type="checkbox"/> | | |
| DetectHiddenCom... | <input type="checkbox"/> | | |
| BundleReleasedCo... | <input type="checkbox"/> | | |
| ShowDetectionVolu... | <input checked="" type="checkbox"/> | | |
| DetectionVolumeG... | Box | | |
| DetectionVolumeTo... | TOOL[1] | | |
| DetectionVolume | X 50 | Y 50 | Z 50 |
| GraspIncludeEmpty... | No -Delete | | |
| ExcludeReleasingTo | | | |
| ExcludeGrasping | | | |



- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) **Grasping**
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

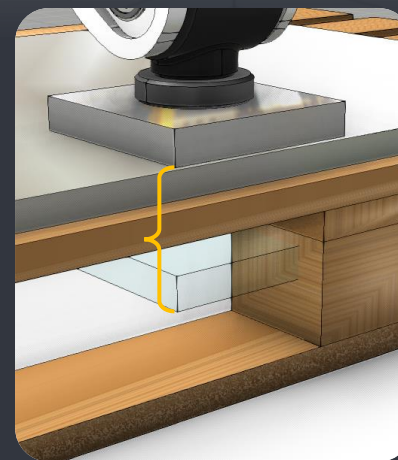
RELEASING



Note: DetectionVolume and GravityDirection can be set for each output (action) individually



- The grasped **component's bounding box** is used to detect a node (below) to release to
- Gravity direction is used to **translate the component** (used for detection) along **world's Z axis**



The grasped component will be attached to the **pallet**

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) **Releasing**
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

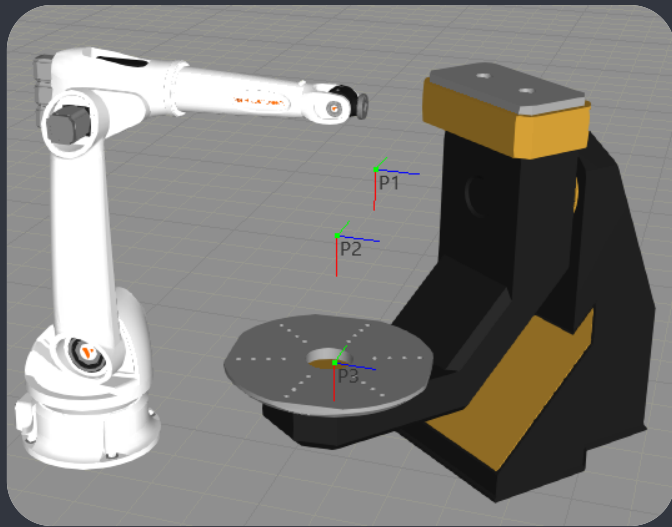
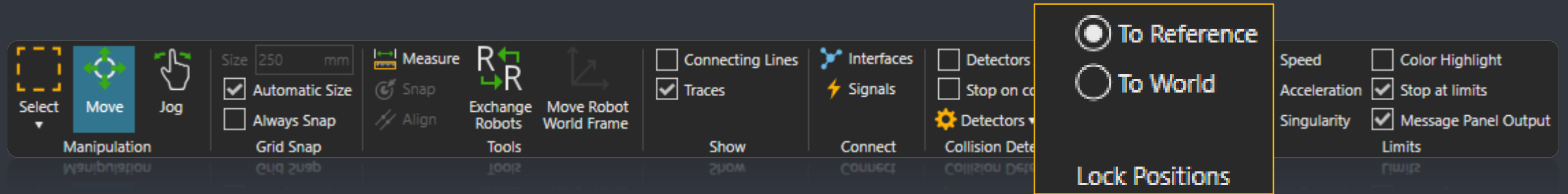
MULTI GRASPING AND RELEASING

- Enable **MultiGrasp** from component properties
- **DetectionVolume** for the selected output is used to grasp **every detected component** within the volume
- All contained components are **released simultaneously**

| WorkSpace | SignalActions |
|-------------------|-------------------------------------|
| DisplayMessages | <input type="checkbox"/> |
| ReleaseToWorld | <input type="checkbox"/> |
| ReleaseToPhysics | Never |
| MultiGrasp | <input checked="" type="checkbox"/> |
| TraceZOffset | 0 |
| UpdateWorldIO... | <input type="checkbox"/> |
| DetectLockedCo... | <input type="checkbox"/> |

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing**
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

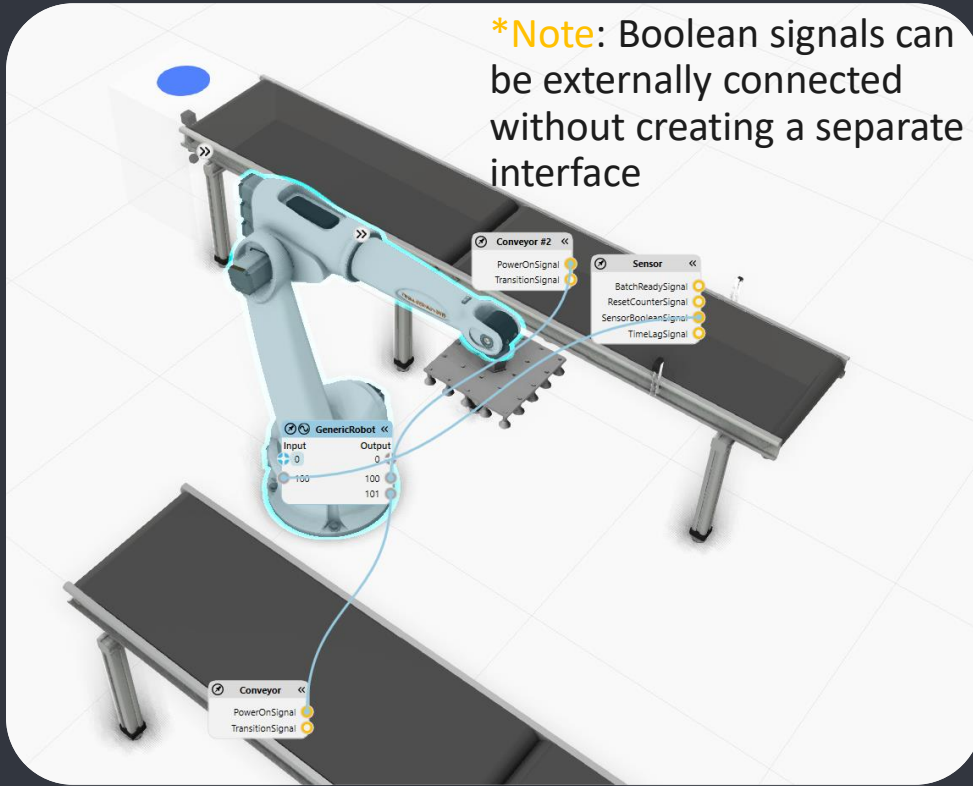
POINTS REFERENCE



- Taught points can be **locked** to the **world** or to the **reference** (base) coordinate system
- **Null** base referenced points are based on the **robot world frame** by default
- Use Case Examples:
 - a. Lock **To World**: Moving robot while keeping taught points in place
 - b. Lock **To World**: To keep points in place when rebaseing
 - c. Lock **To Reference**: Move the robot and points along with it
 - d. Lock **To Reference**: Move robot and keep points in place that are referenced/based to another component e.g. work piece positioner

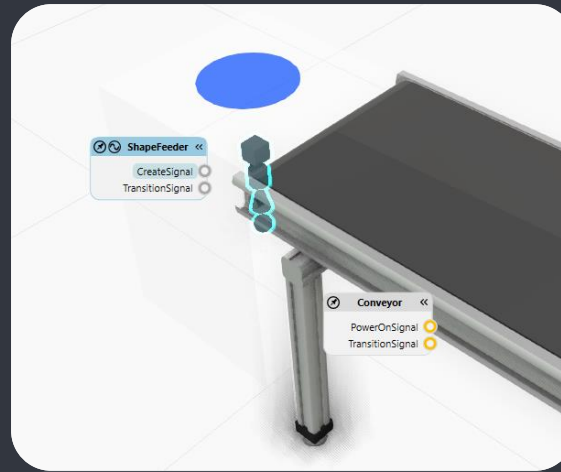
- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference**
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

SIGNALS



*Note: Boolean signals can be externally connected without creating a separate interface

- a. Internal to component or external via assigned interface*
- b. Signal handling done within the component e.g. a connected sensor behavior or a python script



⚡ Signals



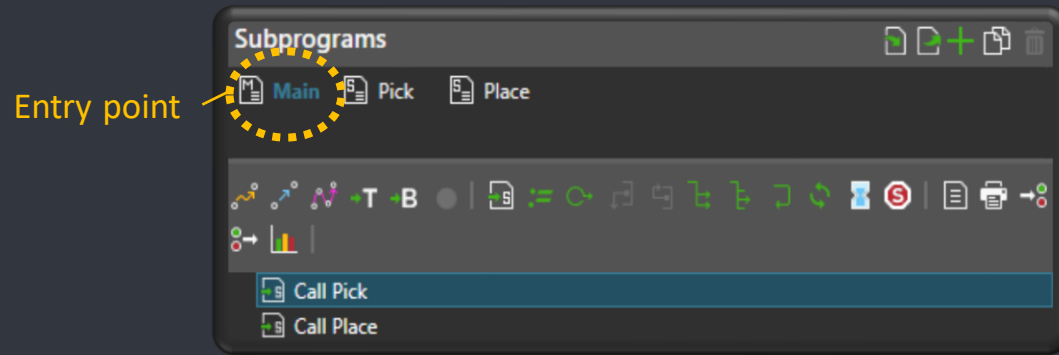
Tip: Access boolean signals from Signals button or quick access menu after selecting component (shown only if component has interfaces)

■ Several signal types

- ⚡_B
Boolean
- ⚡_I
Integer
- ⚡_R
Real
- ⚡_S
String
- ⚡_M
Matrix
- ⚡_C
Component
- ⚡_{Map}
Boolean Map

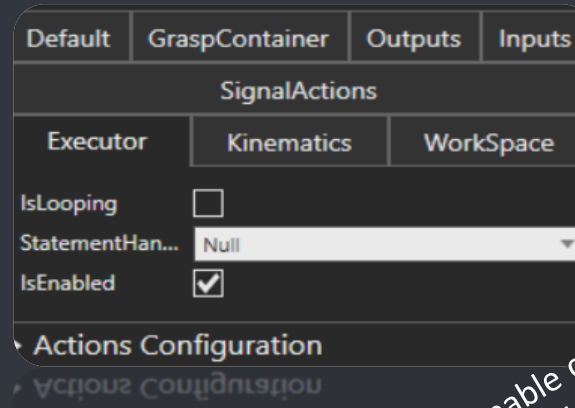
- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) **Signals**
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

SEQUENCES



Entry point

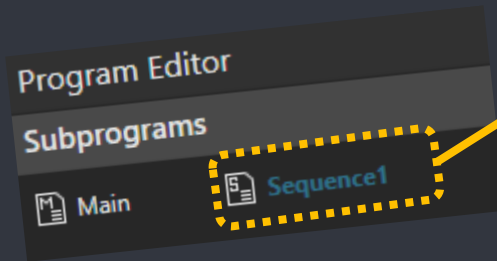
- **Main** sequence/routine is **called** always on simulation start,
 - if **Executor** is enabled
- Subroutines can be called from the Main routine



Hint: Enable continuous main program looping with **IsLooping** property

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

ROUTINE PROPERTIES AND VARIABLES



Routine Properties

Sequence1

Name: Sequence1

Variables

| Type | Name | Value |
|----------|-----------|-------|
| ✓ | Boolean_1 | False |
| I 123 | counter | 0 |

Add new variables

- Add new variables
- Boolean variable
- Integer variable
- Real variable
- String variable

```
Assign counter = 0
PTP P1 *NULL* *NULL* 100%
While counter < 2
  LIN P2 *NULL* *NULL* 2500mm/s
  LIN P3 *NULL* *NULL* 2500mm/s
Assign counter = counter + 1
PTP P4 *NULL* *NULL* 100%
```

➤ Each routine can have **variables** that can be used within the routine



➤ Not possible to share variables between routines

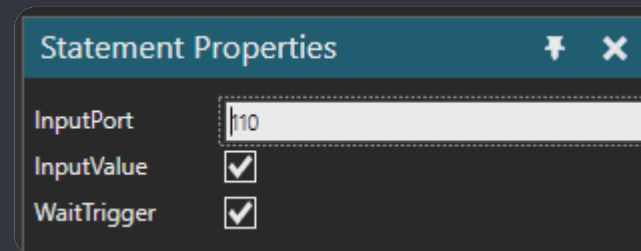
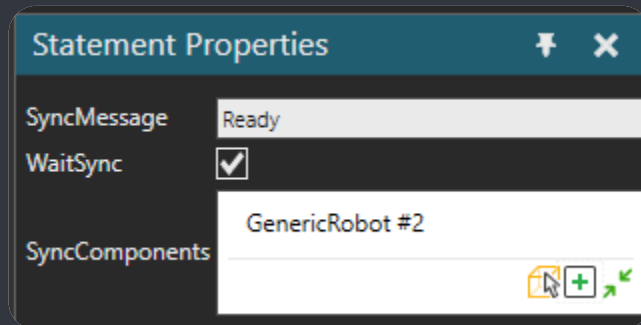
➤ Typical use case is a loop counter

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) Saving a Robot Program

STATEMENT PROPERTIES

All statements have properties. Most are self-explanatory, just a few tips:

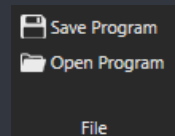
- **Sync** 
 - *WaitForSync* = Wait for reply
- **Wait** 
 - Input: *WaitTrigger* = On rising edge



- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) **Statement Properties**
- 15) Saving a Robot Program

SAVING A ROBOT PROGRAM

- Programs are saved **along with the component**
- Components are saved along with the **layout**
- Saving a robot with a **premade program** is accomplished by saving to a reusable separate file via **modeling tab** with **Save As**



Hint: For exporting and importing VC robot program check this add-on in the forum: <https://forum.visualcomponents.com/forums/topic/save-robot-program/>

- 1) Teaching Target Points
- 2) PTP vs LIN
- 3) MOVE vs JOG
- 4) Editing Target Points
- 5) Tool Center Point and BASE
- 6) Mapping Signals to Robot Actions
- 7) Grasping
- 8) Releasing
- 9) Multi Grasping and Releasing
- 10) Points Reference
- 11) Signals
- 12) Sequences
- 13) Routine Properties and variables
- 14) Statement Properties
- 15) **Saving a Robot Program**