

Introduction

- Each robotic arm is designed and manufactured by considering several key parameters, including the number of axes and corresponding degrees of freedom, the working envelope and the area the robot covers, payload, speed and acceleration, repeatability etc.
- The KUKA Lego robot is a scaled down poly-articulated type robotic arm with 6 axes.

Key Terminologies

- Degree of Freedom – The number of independent motions of a robot.
- Tool Reference Frame (TRF) – The coordinate system fixed with the robot's end effector.
- Tool Center Point (TCP) – The point of the tool where the action is performed.

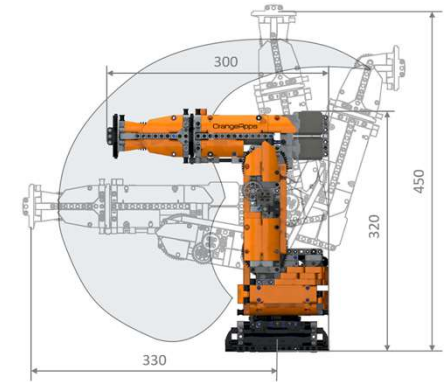


Fig. KUKA Lego robot

Specifications:

- Degrees of freedom : 6
- Controller : built-in LEGO® SPIKE™ controller
- Reach : approx.. 330 mm

Robot Programming Methods

There are two main programming methods, namely online programming and offline programming.

Online Programming

- Teach pendant programming

A teach pendant (SmartPAD) is utilized to guide the robot to the desired coordinate or positions.

- Lead-through programming

It is very similar to teach pendant, but here the robot is physically guided to the desired position.

Offline Programming

- KRL programming

KUKA Robot Language (KRL) is a textual, structured programming language used to program robots.

- Simulation Software

Simulation software can be used to design, test and optimize robot program in a 3D virtual environment.

Types of Motion

- PTP (Point-to-point) motions

The robot guides the TCP from one point to another along the fastest path.

- LIN (Linear) motions

The robot moves the TCP along a straight line at a defined velocity from one point to another.

- CIRC (Circular) motions

The robot moves along a circular path from one point to another.

- SPLINE motions

The robot moves through multiple points describing a smooth curve.

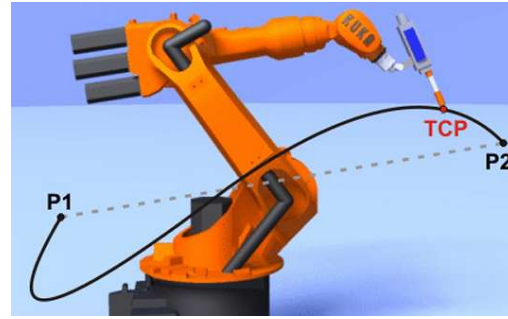


Fig. PTP motion

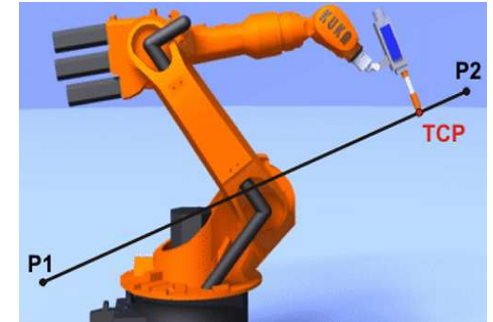


Fig. LIN motion

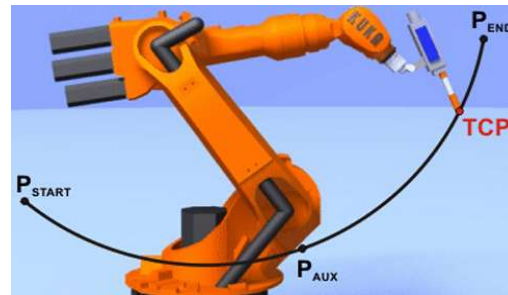


Fig. CIRC motion

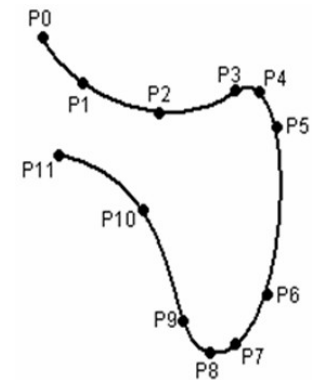


Fig. SPLINE motion

KUKA SmartHMI User Interface

Function: The KUKA SmartHMI has all the control and display functions required to operate and program the robot

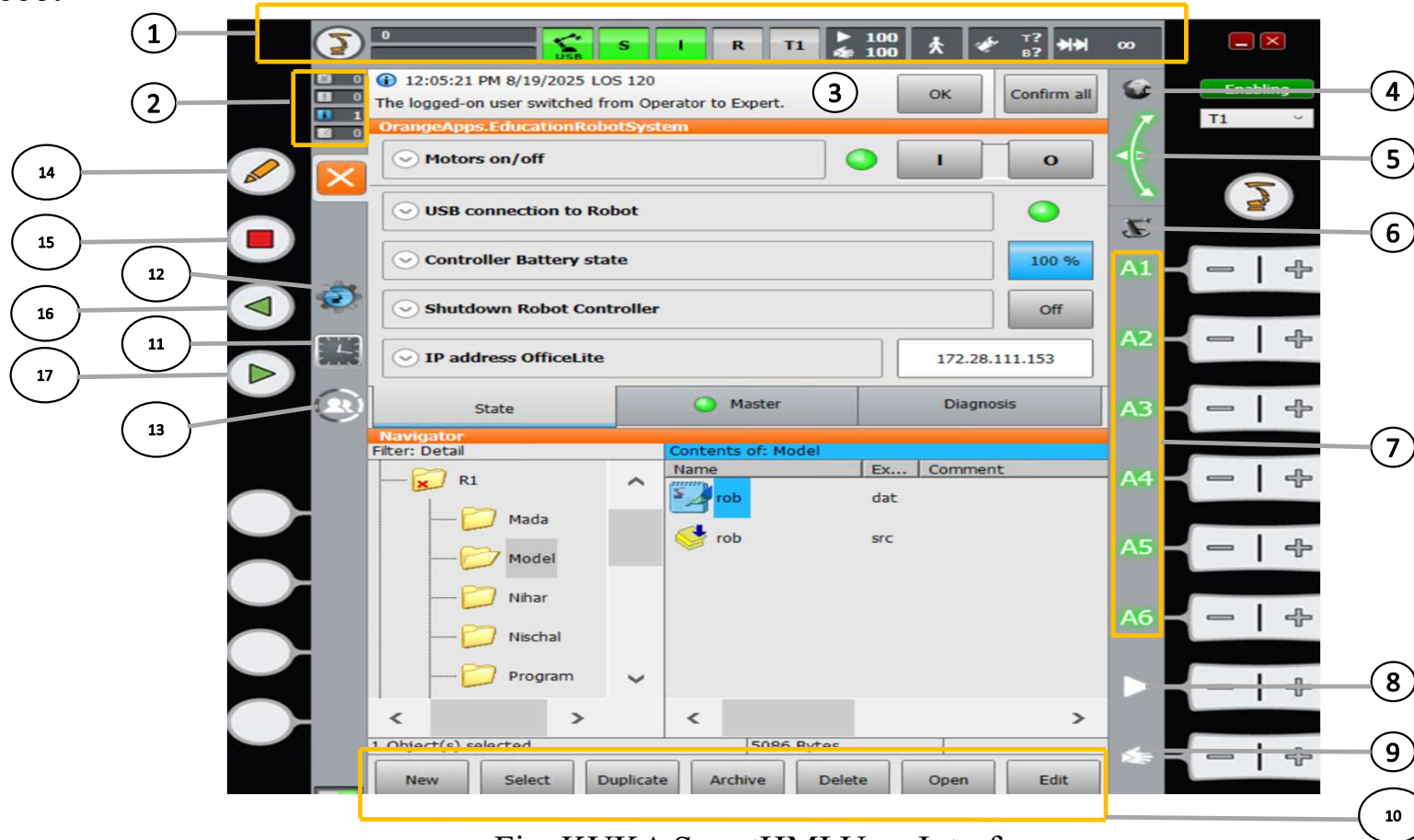


Fig. KUKA SmartHMI User Interface

| Sl.No | Description |
|-------|-------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Status Bar |
| 2 | Message Counter- displays how many of each message types are active |
| 3 | Message window- All the active messages can be expanded by touching the message window. |
| 4 | Space Mouse status indicator - The current coordinate system which is used as reference for jogging. By default, it is set to world |
| 5 | Space Mouse alignment indicator- displays the current alignment of the Space Mouse, which can be modified |
| 6 | Jog Keys status indicator- the current coordinate system for jogging |
| 7 | Jog Key labels – for axis specific jogging (from A1 to A6) |
| 8 | Program override |
| 9 | Jog override |
| 10 | Button bar – the activation of these buttons depends on the current active window in the SmartHMI. |
| 11 | Clock- the clock displays the system time |
| 12 | Work Visual icon- to access the Project management window |
| 13 | Configuration- to set the user group. |
| 14 | Keyboard key- to display the keyboard. |
| 15 | Stop key- to stop the program |
| 16 | Start backwards key- to start the program backwards. |
| 17 | Start Key- to start the program. |

Status Bar – The status bar displays the status or modes of various critical settings of the robot.

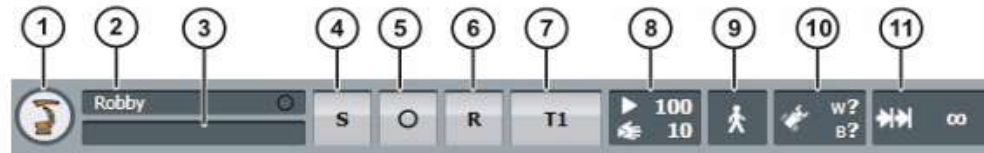


Fig. Status bar of KUKA SmartHMI

| Sl.No | Description |
|-------|-------------------------------------------------------------------------------------------------------------------------|
| 1 | Main menu key |
| 2 | Robot name |
| 3 | Program name – the name of the selected program is displayed here. |
| 4 | Submit interpreter status indicator |
| 5 | Drive status indicator- displays the mode of the drives, either Switched on or off (the symbol O for OFF and I for ON). |
| 6 | Robot interpreter status indicator- indicates whether the program is selected, executing or stopped |
| 7 | Current Operating mode – usually T1 for testing and teaching operations |
| 8 | POV/HOV status indicator- indicates the current program override and the current jog override |
| 9 | Program run mode status indicator- indicates the current program run mode |
| 10 | Tool/Base status indicator |
| 11 | Incremental jogging status indicator |

Jogging the robot

In axis specific jogging, each axis of the robot can be moved in the positive and negative direction using the jog keys.

Precondition

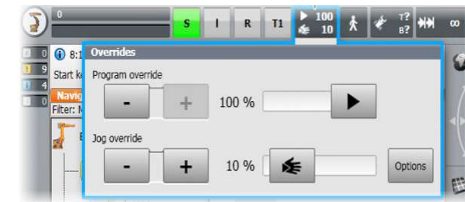
- The Operating mode is set to T1.
- Drives are switched ON.

Procedure

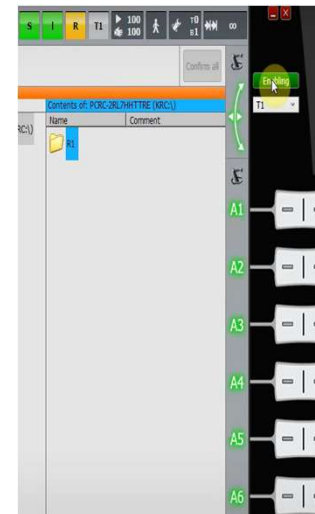
1. Select axes as the coordinate system for the jog keys
2. Set Jog override (specified in percentage of the velocity of robot during jogging)
3. Select the enabling switch, and thus the jog key axes A1 to A6 are highlighted.
4. Press + or – joy keys to move respective axes in the positive and negative direction.



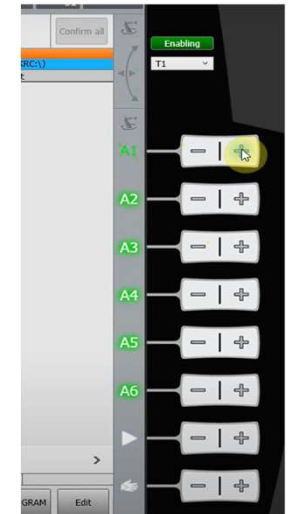
(1)



(2)



(3)



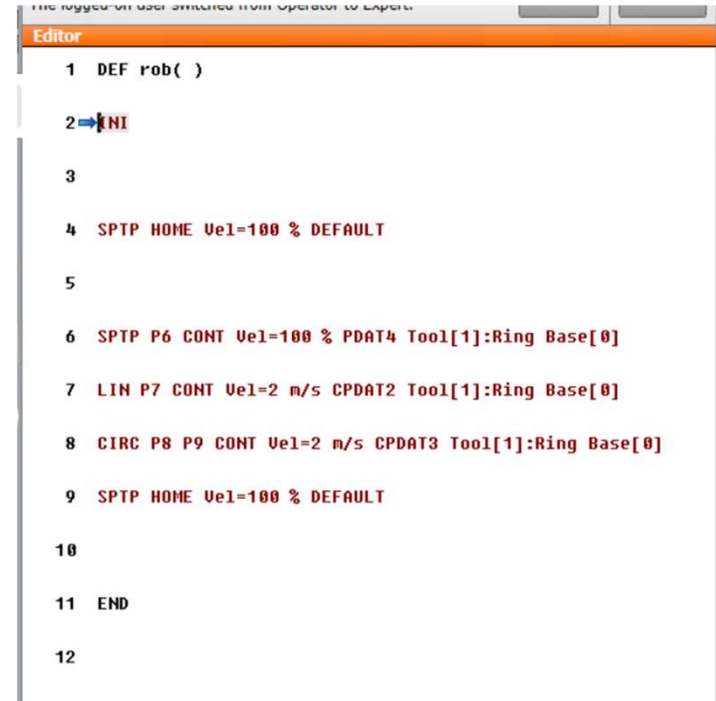
(4)

Fig. Jogging the robot

Sample KRL Code

Open the .Src file to write the program code.

| Line No. | Description |
|----------|------------------------------------------------------------------------------|
| 1 | The DEF line specifies the name of the program. The line can also be hidden. |
| 2 | The INI is the initialization line. |
| 4 | HOME position command |
| 6 | This line contains the Point-to-Point motion command. |
| 7 | This line contains the Linear motion command. |
| 8 | This line contains the Circular motion command. |
| 9 | HOME position command |
| 11 | The END is the last line of the program. |



The screenshot shows a software window titled "Editor" with a list of line numbers on the left and KRL code on the right. The code is as follows:

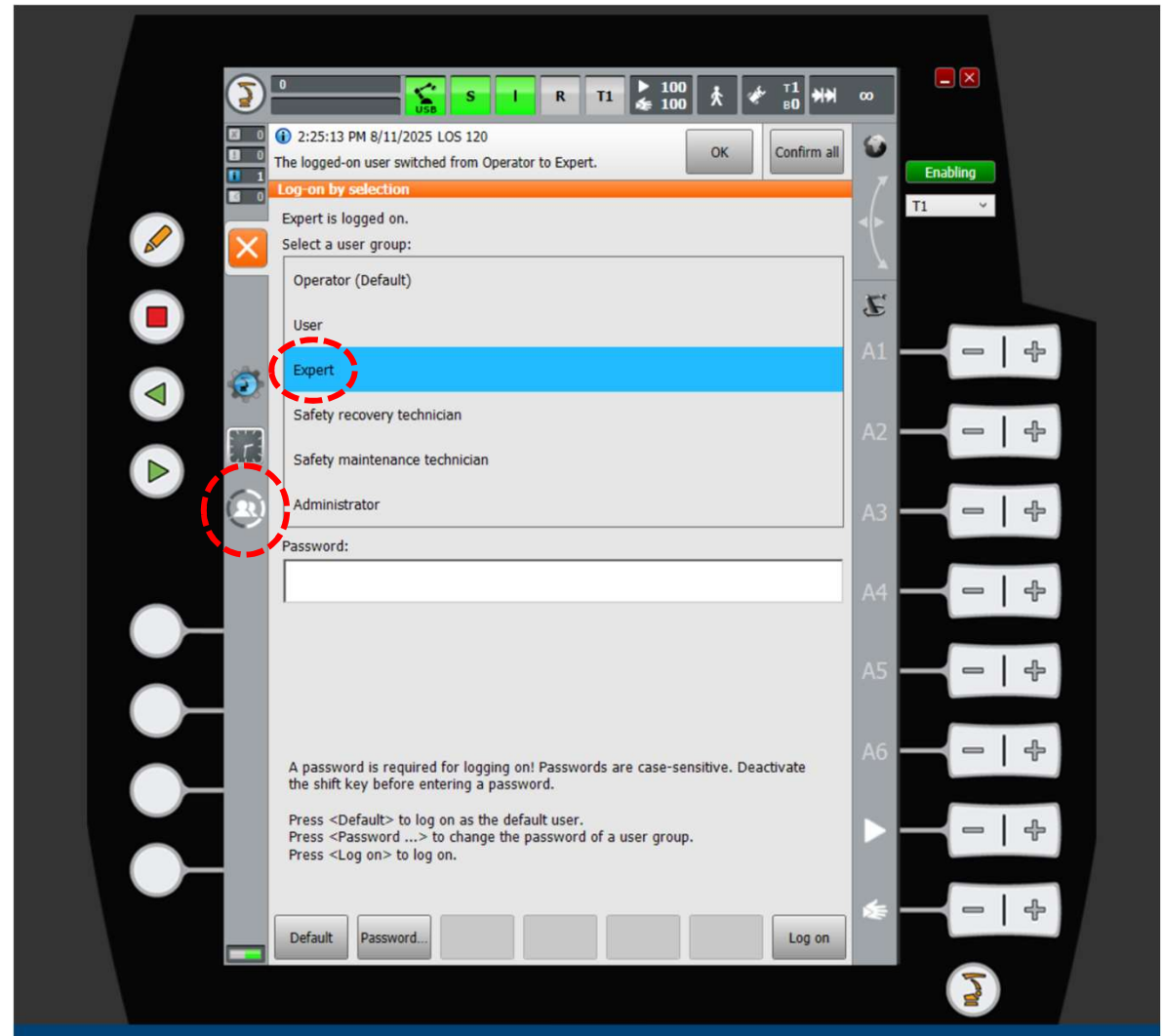
```
1 DEF rob( )
2 →INI
3
4 SPTP HOME Vel=100 % DEFAULT
5
6 SPTP P6 CONT Vel=100 % PDAT4 Tool[1]:Ring Base[0]
7 LIN P7 CONT Vel=2 m/s CPDAT2 Tool[1]:Ring Base[0]
8 CIRC P8 P9 CONT Vel=2 m/s CPDAT3 Tool[1]:Ring Base[0]
9 SPTP HOME Vel=100 % DEFAULT
10
11 END
12
```

Fig. Sample Program code

Set Expert Mode

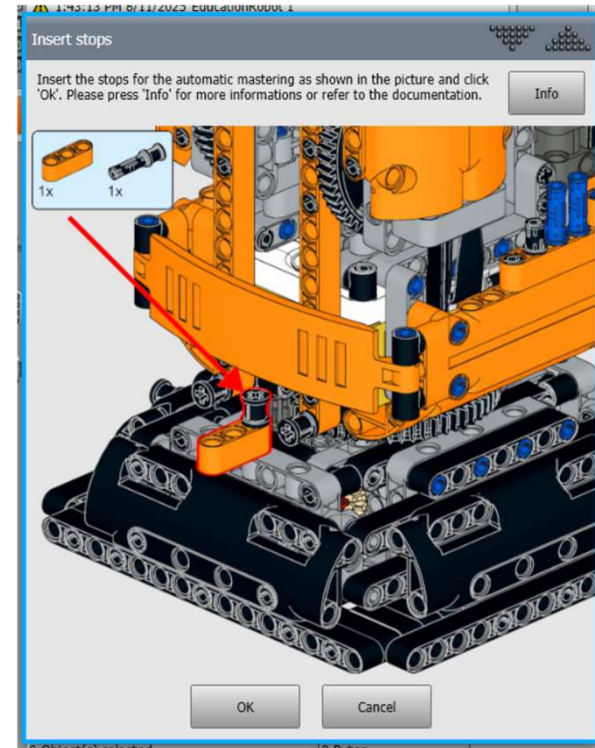
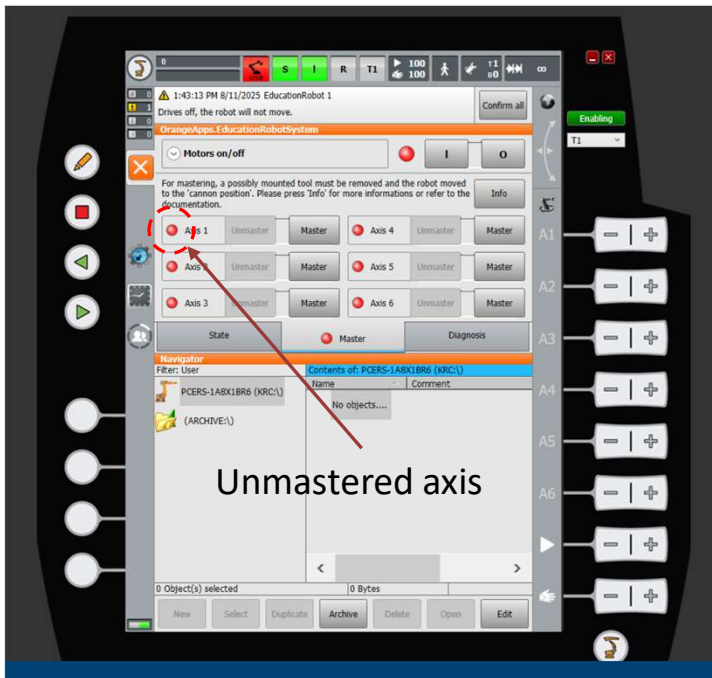
- The Kuka file types (.Dat file and .Src file) can only be accessed through the Expert user group.
- In the user selection panel select expert mode.

Password – kuka



Master the Robot

- Each axis has to be mastered individually before proceeding with the pick and place task.
 - Click the robot icon and select the master tab.
 - Select axis 1 and follow the on screen instructions to master it.
 - Continue with the other axes till all are mastered



Mastered axis

- Once mastered, the red indicator turns to green. When all the axes are indicated green you can proceed with the task.



How to create a .src file

- Switch to expert mode.
- In the navigator menu, ensure that the specified folder is selected and the cursor is on the right half of the menu.
- Click New>Module>Enter file name.
- A new .src file and the corresponding .dat file are created.
- Click the select option to open and edit the program.














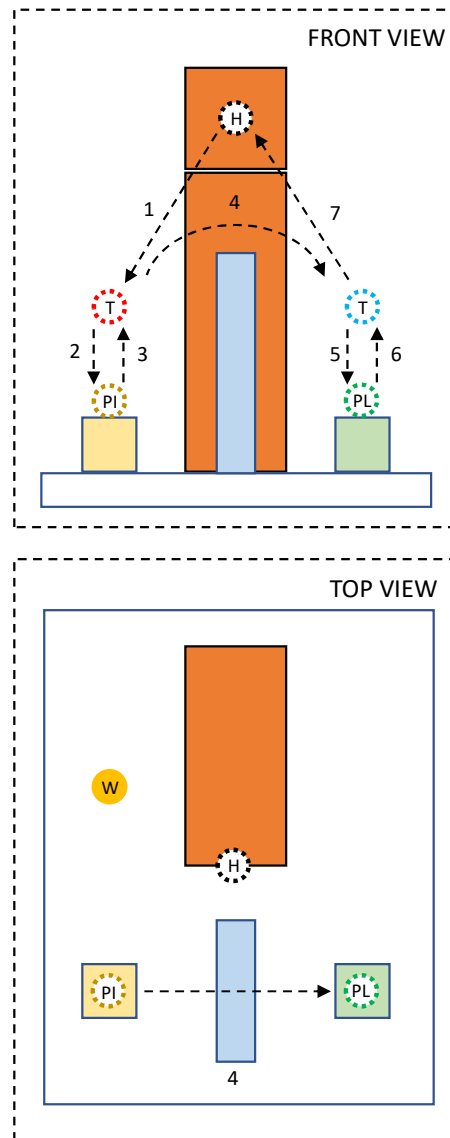
Description of the task

- The primary objective of the task is to pick the work piece (Ping Pong ball) from the fixed pick up position, traverse over or around the obstacle and place it at the designated position.
- The TCP has to follow the positions indicated in the diagram.
- The coordinates of the said points and the type of motion can be chosen by the user.

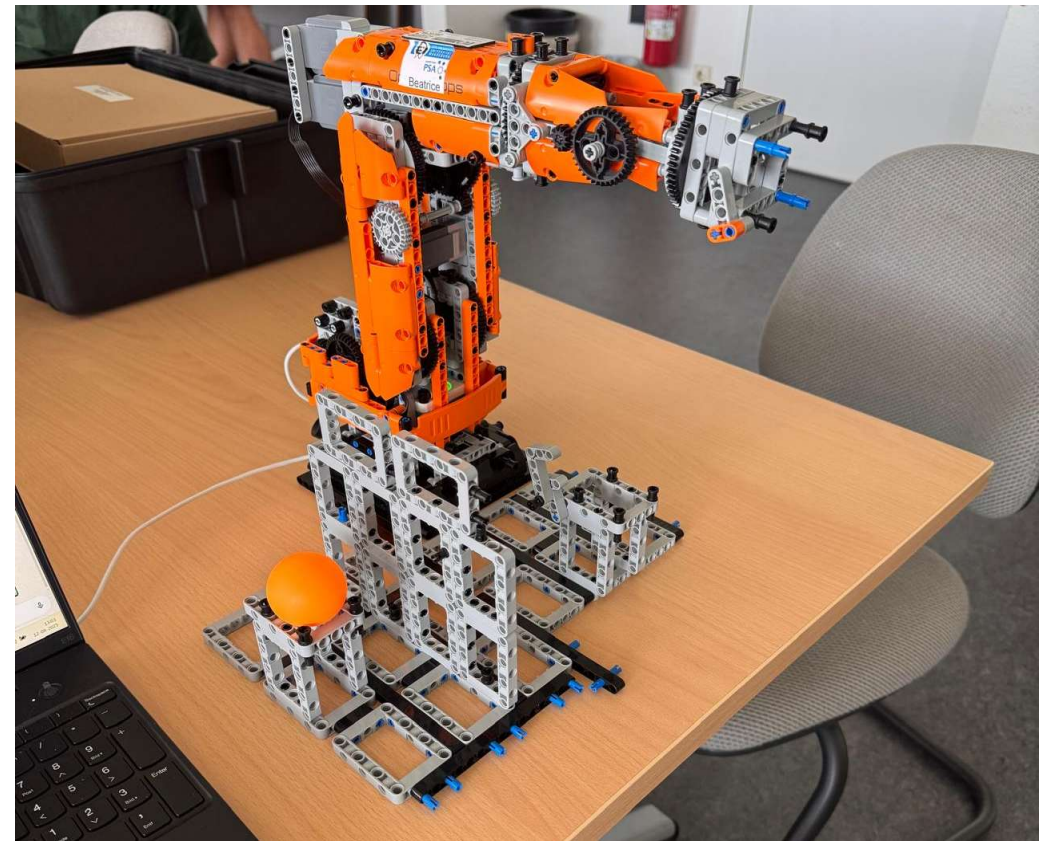
Steps:

1. Ensure that the robot is mastered and at home position.
2. Place the work piece on the pick position.
3. The TCP moves from home position(H) to target position 1, then down to the pick position.
4. Once the work piece is secure in the gripper, the TCP moves to the target position 2 avoiding the obstacle.
5. The work piece is placed at the place position. The TCP returns to home position. This completes one work cycle.

-  TCP Home
-  TCP Target Position 1
-  TCP Pick position
-  TCP Target Position 2
-  TCP Place Position
-  Work Piece
-  Work Piece Pick Position
-  Work Piece Place Position
-  Obstacle
-  KUKA ERS 3.0
-  Work Table



Pick and Place Operation using KUKA robot



Commands to play, reverse and stop the program

The screenshot shows the KUKA robot programming interface. At the top, there's a status bar with icons for USB, Stop (S), Run (R), and T1. Below it, a log shows the time 2:23:09 PM 8/12/2025 and the message 'The logged-on user switched from O'. The main editor window displays a program with the following code:

```
1 DEF spractice( )  
2 →INI|  
3  
4 SPTP HOME Vel=100 % DEFAULT  
5  
6 SPTP HOME Vel=100 % DEFAULT  
7  
8 END  
9
```

On the left side, a vertical toolbar contains several icons. A red box highlights three icons: a red square (stop), a green left-pointing triangle (reverse), and a green right-pointing triangle (play). A callout points to these icons with the text 'Commands to play, reverse and stop the program'.

At the top center, a red box highlights a circular icon with a double arrow. A callout points to it with the text 'To cancel and reset the program'.

On the right side, a vertical toolbar contains several icons. A red box highlights a series of six buttons labeled A1 through A6. Each button has a minus sign, a vertical line, and a plus sign. A callout points to these buttons with the text 'Used to create movement around the axes of the robot'.

At the bottom, a status bar shows 'BP 2' and 'KRC:\R1\pratheek2\src'. Below this, a row of buttons includes 'Change', 'Commands', 'Motion', 'Open/close fold', 'Block selection', 'Touch-up', and 'Edit'. A callout points to the 'Motion' button with the text 'To create a point and define the movement'. Another callout points to the 'Block selection' button with the text 'To define the position of the created point'.

Used to create movement around the axes of the robot

Set Target Position

The screenshot displays a robotic control interface with a code editor and a sequence of movement commands. The code editor on the left contains the following sequence:

```
1 DEF pick( )  
2 →INI  
3 SPTP HOME Vel=100 % DEFAULT  
4 SPTP P1 Vel=100 % PDAT1 Tool[1]:Ring Base[0]  
5 SPTP P2 Vel=100 % PDAT2 Tool[1]:Ring Base[0]  
6 SPTP P3 Vel=100 % PDAT3 Tool[1]:Ring Base[0]  
7 SPTP P4 Vel=100 % PDAT4 Tool[1]:Ring Base[0]  
8 SPTP P5 Vel=100 % PDAT5 Tool[1]:Ring Base[0]  
9 SPTP HOME Vel=100 % DEFAULT  
10 END  
11
```

On the right side of the interface, a vertical panel shows a sequence of movement commands, each with a green label (A1 through A6) and a button with minus and plus signs. Arrows point from the code editor to these buttons:

- Line 3 points to the Home Position button (A1).
- Line 4 points to the Home to Position 1 button (A2).
- Line 5 points to the Position 1 to Position 2 (Picking Position) button (A3).
- Line 6 points to the Position 2 to Position 3 button (A4).
- Line 7 points to the Position 3 to Position 4 button (A5).
- Line 8 points to the Position 4 to Position 5 (Placing Point) button (A6).
- Line 9 points to the Position 5 to Home button (A7).

The interface also includes a top status bar with various icons and a bottom toolbar with additional controls.

