Introduction to the KUKA Workstation

Mike Mehramiz, George Owen, Bruno Korst
Revised by Manfredi Maggiore

Abstract
In this experiment you will be introduced to the basic operation of the `KUKA KR6 R700 SIXX` workstation. You will learn details about the robotic arm, the cabinet, the pendant and its screen, as well as the safety and emergency features. You will learn how to move the robot, create your own coordinate system and will move the robot using a program.

Keywords
KUKA — Robots — Reference — Pendant — Safety

Contents

Introduction

The purpose of this lab is to introduce you to an industrial robot and its safety rules. During the lab and from the lab instruction, you will be provided with enough information about our robot KUKA KR6 R700 sixx [1] to learn how to safely move an individual robot axis through the use of KUKA's smartPAD. You will also learn how to assign a coordinate system for this activity and learn how to use Inline Form for PTP (Point To Point) motions. Be sure to read this entire document prior to coming to the lab, as you should be prepared to operate the robot safely.

1. The Hardware

An industrial robot as defined by ISO 8373:2012 as an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications. Figure 1 below shows the parts of a typical industrial robot.

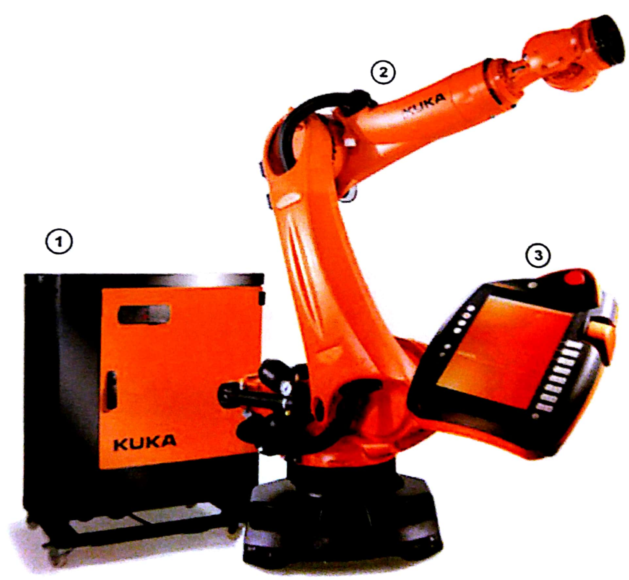


Figure 1. A typical industrial robot

Three main parts of an industrial robot:

1. Controller (KR C4 control cabinet);
2. Manipulator (the robot arm itself);
3. Teaching pendant (KUKA smartPAD)

The industrial robot to be used in this experiment moves in six axes. Figure 2 below shows on a generic industrial robot how the axes are named and how they move.

[1] The double-x in sixx is not a typo!

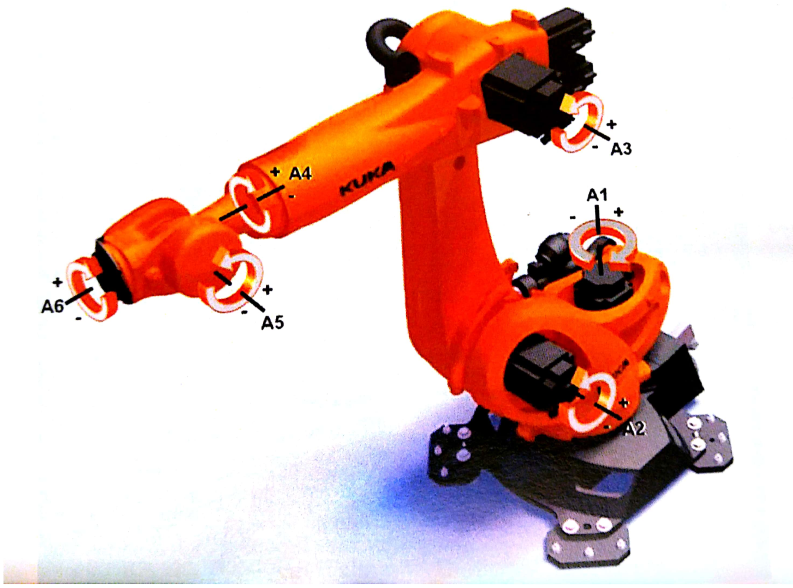


Figure 2. Six axes and their movement

1.1 The KUKA KR6 R700 Sixx

The KUKA KR6 R700 robot has six axes and is categorized as a KUKA *small robot* because its maximum payload is 6kg. KUKA small robots can be used in many application as their lager relatives but with less payload. KR 6 R700 sixx is rated as a robot with high working speeds and high precision. Table1 shows the range of motion of each KR 6 R700 sixx axes. In case you have noticed that there are way too many sixes written in this document, note also that we have six of these robot arms in the lab.

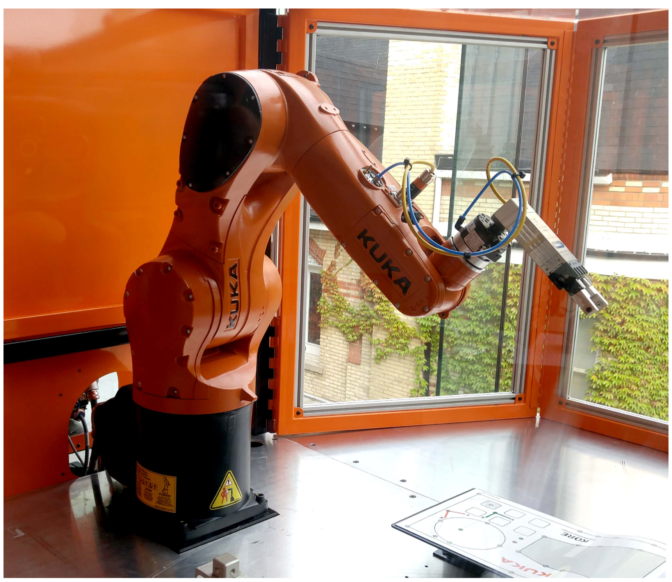


Figure 3. The KUKA KR6 R700 Sixx in BA3114

Axis	Movement
Axis 1	$\pm 170°$
Axis 2	$+45°/-190°$
Axis 3	$+156°/-120°$
Axis 4	$\pm 185°$
Axis 5	$\pm 120°$
Axis 6	$\pm 350°$

Table 1. Range of motion

1.2 smartPAD

The smartPAD, shown in Figure 4 is the teaching pendant for the industrial robot. It is used to control and program the robot. The robot operator – for this case, you – can see the messages from the controller (KR C4) on its screen. The smartPAD has a touch screen and it does not need any extra mouse or keyboard to work with the robot. The parts show in 3 are described in the table following below.

Figure 4. Front view of the smartPAD

Item	Description
1	Button for disconnecting the smartPAD
2	Keyswitch for connection manager. The key must be in.
3	**Emergency Stop Button**. The button locks when pressed.
4	Space mouse. Used for moving the robot manually.
5	Jog keys. Also used for moving the robot manually.
6	Key for setting the program override.
7	Key for setting the jog override.
8	Main menu key. Shows the menu items on the screen.
9	Status keys. Sets technology package parameters.
10	Start key. This key is used to start a program.
11	Start *backwards* key. Executes backwards step by step.
12	Stop key. This key stops a program that is running.
13	Keyboard key. Displays the keyboard on the screen.

Table 2. Table of Functions

The back of the smartPAD also contains some control features, as Figure 5 below presents. Note that the enabling switch has 3 positions: released (i.e. not pressed), centre position and panic position. The enabling switch must be held in the centre position during operating mode T1, in order for the operator to jog the manipulator (the arm). In addition, the USB connector is available to save/archive and restore data, but only to and from FAT32 formatted USB keys.

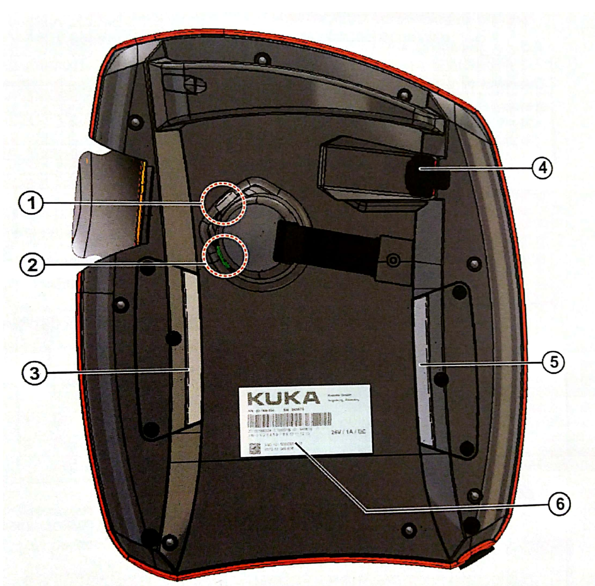


Figure 5. Back view of the smartPAD

Item	Description
1,3,5	Enabling switch (3 positions)
2	Start key (green) - start a program
4	USB connection
6	Identification plate

Table 3. Table of functions

2. Safety

The first priority in our laboratory is to work in a safe environment. Understanding and following the safety rules of the equipment in use is extremely important for those who operate any industrial robot. In the process of working with an industrial robot, one is bound to have close contact with it, but such contact should be limited to primarily three instances:

- a) the *commissioning phase*, when the robot is put in place and made to work for the first time,
- b) *in teaching*, when all characteristics and features of the robot are explored, and
- c) *during maintenance*, when parts are replaced and adjusted.

However, when an industrial robot is put in its place in a production line, there should not be absolutely any possible physical connection between the user and the robot. Failure to observe this may result in severe injury or death to either the operator or persons in close contact with an operating robot.

In addition, considerable damage to property may also result. Finally, never dismantle or deactivate any of the safety measures designed for the robot. They are in there for the safety of the operator and other workers, as well as for the protection of potential material damage.

2.1 Robot Safety Features

In our laboratory, each station is equipped with a number of safety features. These features are identified in Figure 6 below. It is very important that you to follow safety rules when operating this station. If one removes or disables the safeguards put in place, personal injury or material damage will result.

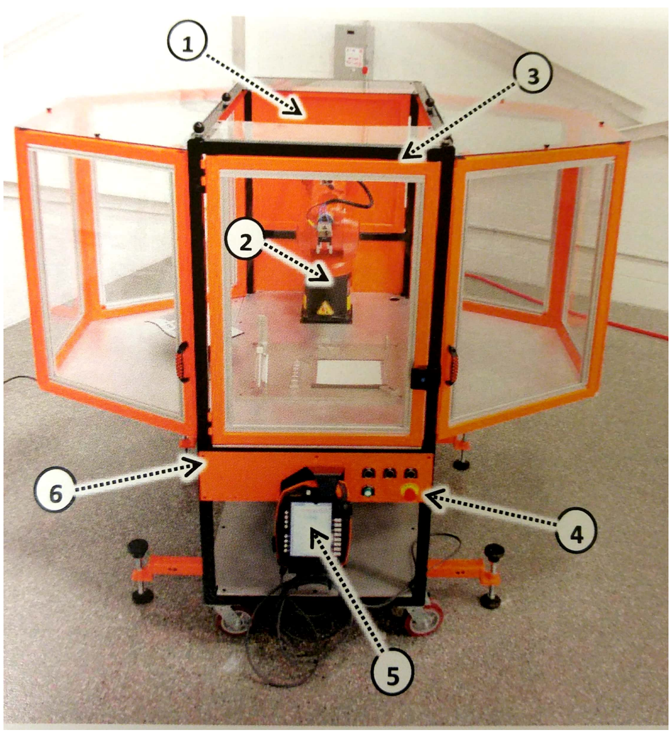


Figure 6. Workstation Safety Features

Item	Description
1	Safeguard
2	Mechanical end stop or axis range limitation for axis 1(A1)
3	Safety gate with contact sensor
4	EMERGENCY STOP button (external)
5	EMERGENCY STOP button enabling switch with key
6	Integrated KR C4 safety controller

Table 4. Table of functions

3. Operating Modes

There are two operating modes that you can choose to work in our laboratory: T1 (manual reduced velocity) and AUT (automatic). T1 is used for test operation, programming and teaching and its velocity in program mode and jog mode is a maximum of 250mm/s. AUT is chosen for industrial robots without higher-level controllers (PLC) and its velocity in program mode corresponds to the user-programmed velocity. Jogging with the jog keys and Space Mouse is not possible in AUT mode.

Users should follow three steps to change the operating mode.

1. On the smartPAD, turn the switch for the connection manager. At this point, the connection manager is displayed.
2. Select the operating mode
3. Return the connection manager switch to the original/default position. The selected operating mode is displayed in the status bar of the smartPAD.

Figure 7. Turning the switch

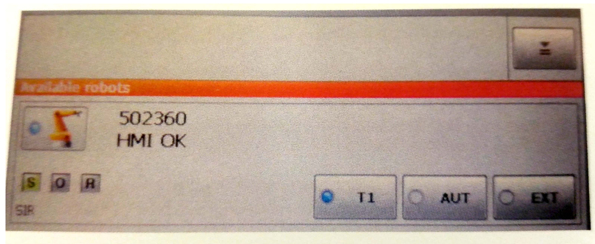


Figure 8. Selecting operating mode

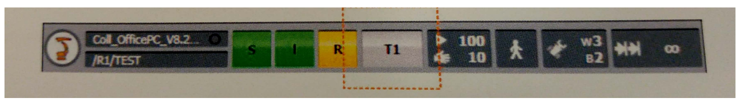


Figure 9. Operating mode is displayed

3.1 Operator Safety

The operator safety mechanism is used for interlocking the enclosure, or physical safeguards of the workstation. Automatic operation does not work without it. In case there is a disruption of this mechanism during automatic operation, the manipulator (that is, the robot arm) stops. This safeguard is not active in T1 mode (Manual Reduce Velocity). Note the position of the interlocking mechanism on Figure 10 below.

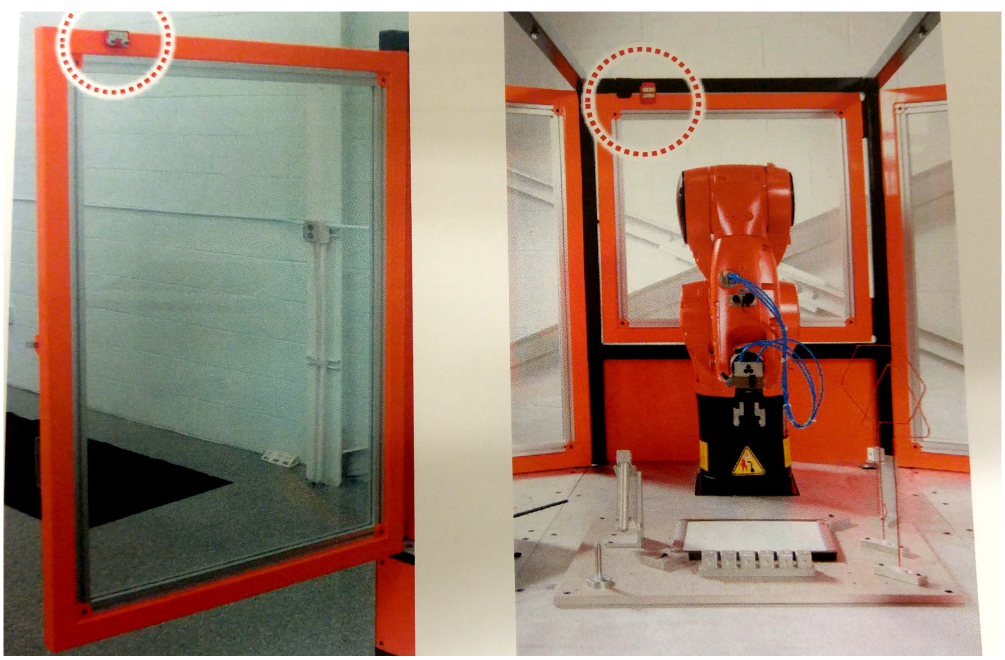
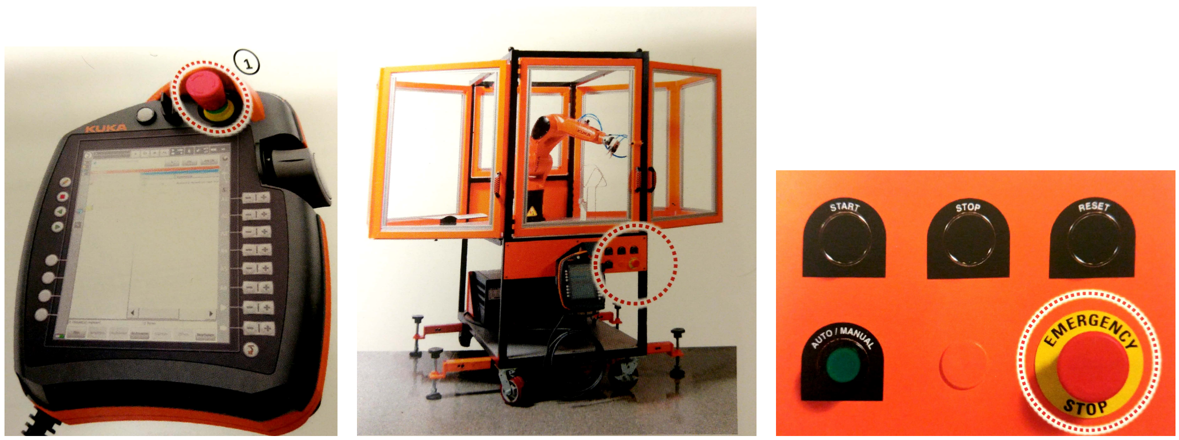


Figure 10. Gate switches – front and rear door

3.2 Emergency Stop Device

An emergency stop switch can be found on the smartPAD as well as on the external (front) panel of the workstation. Their locations are shown in Figure 11 shown below. In case of a potentially hazardous situation or an emergency, the button should be pressed. This will cause the robot arm to stop. The emergency stop button must be turned to the ”release” position at the beginning of every session in order for the robotic arm to become operable.

Note also that the emergency stop button does not work when the SmartPAD is disconnected from the operator panel. As a safety precaution, users should always be at a close distance to the workstation in order to have access to one of the two emergency stop buttons. The emergency stop button placed on the external frame works in identical fashion to that on the SmartPAD.



(a) Emergency Stop on smartPAD (b) Emergency Stop on chassis (c) Panel view

Figure 11. Emergency Button

3.3 Enabling Switches

There are three enabling switches on the back of the smartPAD as Figure 4 showed above. These switches are located roughly where the operator’s hands would hold the smartPAD, in order to make it easier for the operator to press them. As the operator presses on the switches, there are three possible positions: released (default), in which there is no operation; central position, in which the robot moves; and panic mode, in which the switch is pressed down to its full extent (it would make intuitive sense that a panicked operator would hold on strongly to the smartPAD). For the robotic arm to be moved in test mode (T1), the user should hold one of the enabling switches in the central position.

3.4 Mechanical Axis Range Limitation

The robot in our lab has been fitted with a mechanical axis range limitation on axis A1. This range limitation is adjustable to restrict the working range of the robot arm to the required minimum. One should keep in mind that the robot should be always restricted – if possible physically or mechanically – to its minimal working range. Figure 12 below shows the angles at which it is mechanically possible to limit the range.

3.5 Safety Labels

Labelling on the robot system consists of: Rating Plates, Warning labels, Safety symbols, Designation labels, Cable markings and Identification plates. All of these labels, symbols and marks are safety-relevant parts of the robot. Figure 13 shows some of the labels found on the workstation.

4. Hands-on Activity I

4.1 Moving Individual Robot Axes via smartPAD

In this activity, you will move the robot along individual axes by *jogging* it in the plus and minus directions as shown on Figure 14. Jogging the robot is only possible while in T1 operating mode and with the enable switch pressed.

After you follow the procedure below step by step and understand how to move the robot, you will be able to create your own reference, trajectory, etc.

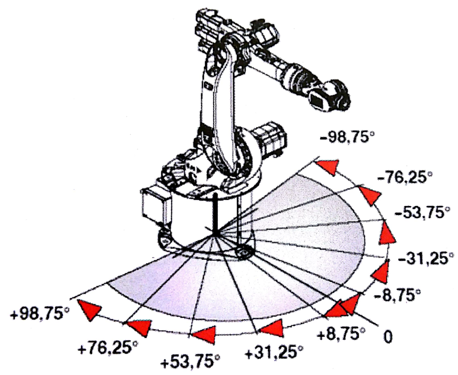


Figure 12. Mechanical Range Limitation

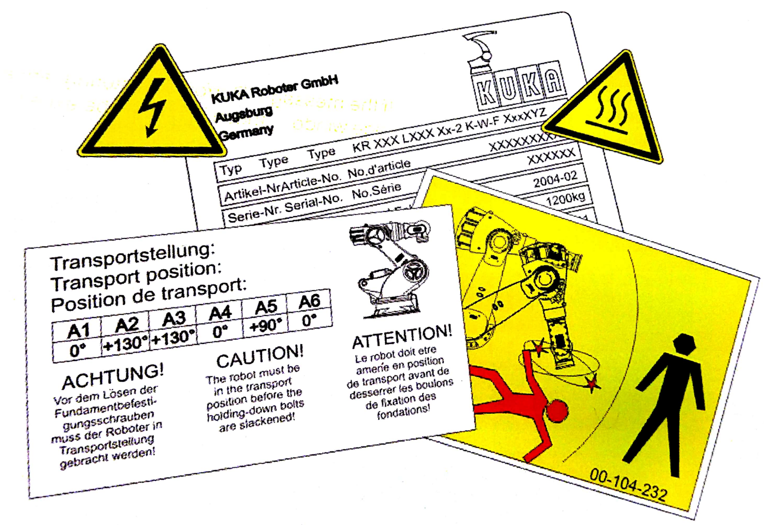


Figure 13. Labels on the Robot

4.1.1 Procedure

Follow the procedure below, keeping in mind the safety instructions previously explained in this document. The pictures show parts of the screen on the pendant.

1. Set the operating mode to T1 as explained previously;
2. Select *Axis* as the option for the jog keys, as shown below (soft button close to the jog knob);

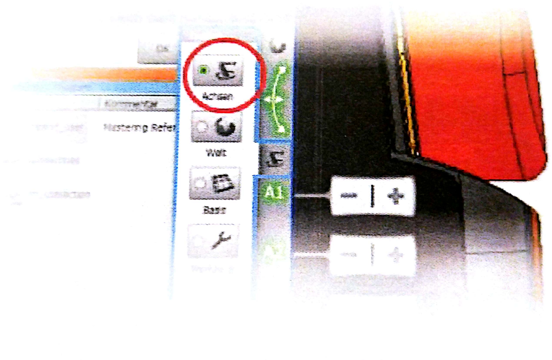


Figure 14. Jog along axis

3. Set jog to override;

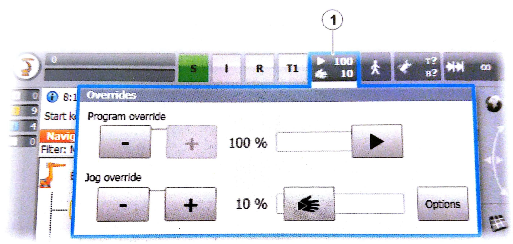


Figure 15. Jog Override

4. Press the enabling switch into the centre position and hold it down as explained previously. Axes A1 to A6 will be lit *green* next to the jog keys;
5. Press the *Plus* or *Minus* jog key to move along an axis in the positive or negative direction.

Figure 16. Plus and Minus Toggles on the Pendant

You may see a message such as *Software limit switch A5* displayed on the SmartPAD. If this is the case, you have moved the robot to a software-preset limit along the axis indicated (this case: A5). All you need to do is to move the robot in the opposite direction to fix the problem.

5. Coordinate Systems

Coordinate systems are points of reference that the user chooses in order to set all other points for the robot to move. Users can define five coordinate systems in the robot controller. They are: `ROBROOT` (Robot Base), `WORLD`, `BASE`, `FLANGE` and `TOOL`. In this lab, you will use the `BASE` coordinate system. Figure 17 shows each of the coordinate systems.

Note that the `ROBROOT` takes the base of the robot itself as the global reference. `WORLD` takes some point external to the robot as a global reference. Then `BASE` uses a point on the base of the target working area as reference. Finally `FLANGE` will

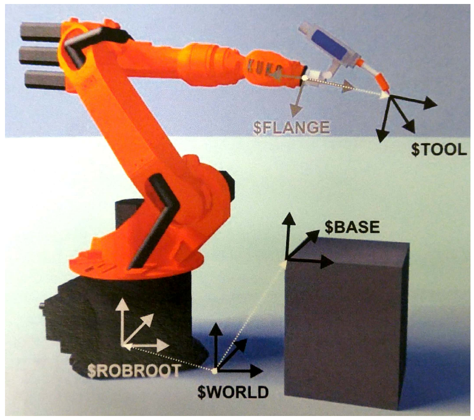


Figure 17. Five Coordinate Systems

use the extremity of the robot without the tool as reference, whereas `TOOL` will use the tip of the tool (a pen, a welding tip) as the global reference. Depending of the type of work to be performed, the operator much choose the appropriate reference, so that the robot performs most efficiently in terms of speed and precision.

5.1 Moving the Robot in Base Coordinate System

After the operator chooses a base coordinate system, the robot will move having as a reference the "zero" point of that coordinate system. The robot controller can accept up to 32 base coordinate systems and they are often oriented *along the edges* of work pieces, as shown on Figure 18.

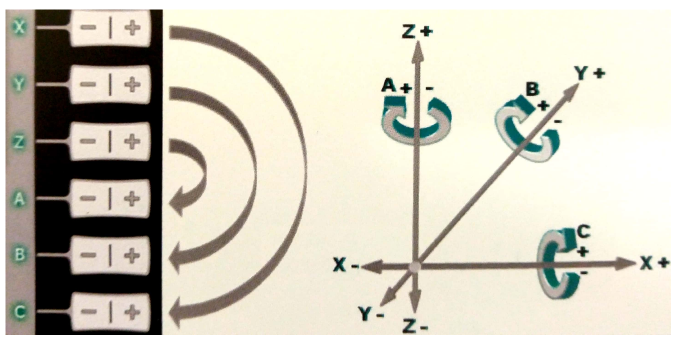


Figure 18. Cartesian Coordinate System

The robot can be moved via the jog keys or space mouse on the smartPAD. Its velocity can also be changed via the jog override (HOV). In this experiment we are using only the jog key. As you recall, users can jog the robot when it is on T1 mode and the enabling switch is pressed to its central position. The robot is moved with the jog keys in a straight line (X,Y,Z) or rotational (A,B,C), as shown on Figure 18.

5.2 Base Calibration

Base calibration is the creation of a coordinate system at a specific point in the robot environment, relative to the `WORLD` coordinate system. There are three methods to assign a robot a base coordinate system: the *3-point* method, the *Indirect* method and the *Numeric* method. We will use for our next exercise the 3-point method to calibrate the robot.

6. Hands-on Activity II

In this activity, you will first calibrate the robot using the 3-point calibration method, in order for you to make the robot move with the newly defined reference.

6.1 3-Point Method Calibration

Follow the steps below to calibrate the robot.

1. In the main menu, select `Start-up -> Calibrate -> Base -> 3-point`;
2. Assign a number and a name for the base you have created. Confirm by pressing `Next`;
3. Enter the number of the tool (choose one), whose Tool Centre Point (TCP) is to be used for the new base calibration. Confirm by pressing `Next`;
4. Move the TCP to the origin of the new base as shown in Figure 19 below. Press the `Calibrate` soft key and confirm the position by pressing `Yes`;

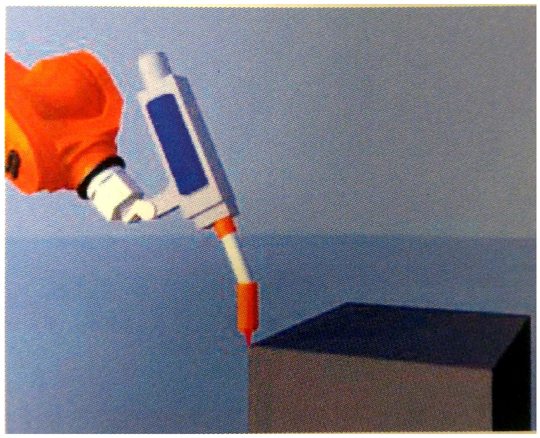


Figure 19. First Point Defined (Origin)

5. Move the TCP to a point on the positive X axis of the new base, as shown in Figure 20 Press the `calibrate` soft key and confirm the position by pressing `Yes`;

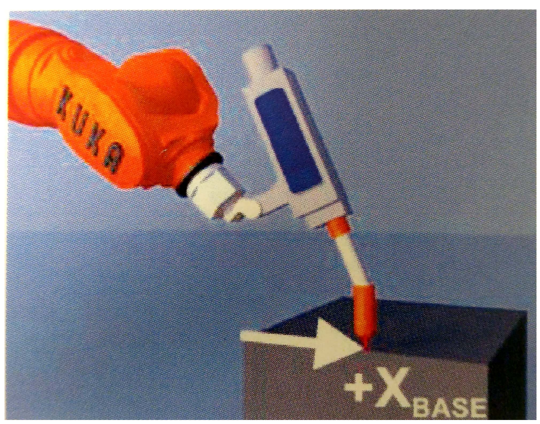


Figure 20. Second Point Defined (X axis)

6. Move the TCP to a point in the XY plane with a positive Y value as shown in Figure 21. Press `calibrate` and confirm with `Yes`;
7. Press `Save`
8. Close the menu.

6.2 Moving the Robot in the (New) Base Coordinate System

Now you will move the robot in the new coordinate system, which you have just used to calibrate the robot. Follow the procedure below:

1. Select `Base` as the option for the jog keys;

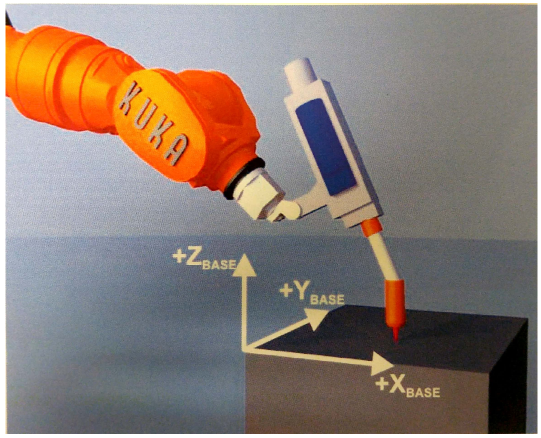


Figure 21. Third Point (XY plane)

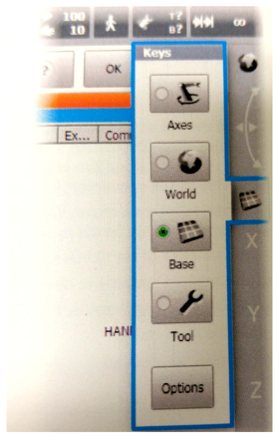


Figure 22. Select (new) Base Coordinate

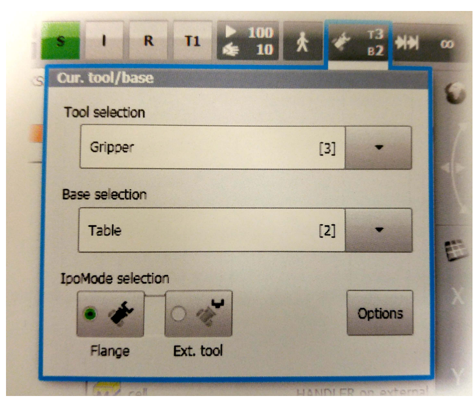


Figure 23. Select Tool and Base Type

2. Select the `tool` and the `base`
3. Select jog override velocity
4. Press the enabling switch into centre position and hold it down;
5. Use the jog keys to move the robot.

Having done all this, you are now ready to use your knowledge of how to calibrate a robot and to choose base and tool type

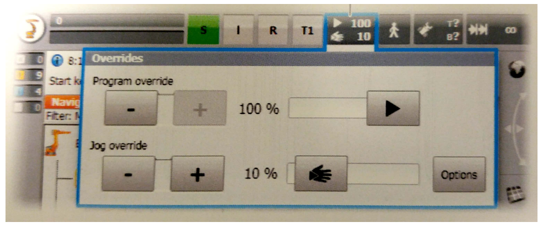


Figure 24. Override velocity

to operate on a given surface. This surface will have a point at which you will place your reference. In other words, you will calibrate the robot, using the 3-point method, by placing the reference on that point.

Your tasks are:

- Calibrate the *green* base on the robot table using the 3-point method. Assign the base number 1 and the name `My_Base`. Use `My_Marker` which has already been defined as `Tool Number 1` as the calibration tool. Below is a picture of your working surface containing the point at which you will place your reference.

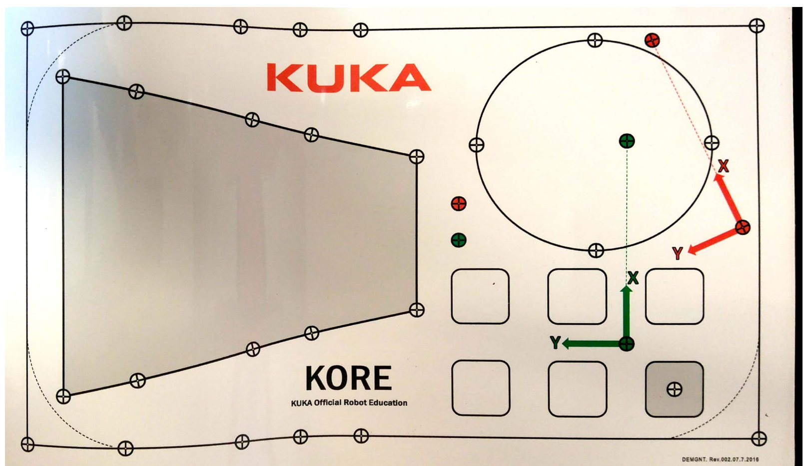


Figure 25. Your Working Surface

- Save the data of the calibrated base;
- Test jogging along the edges of the wave table in the base coordinate system using `My_Base` with the marker in the gripper.

7. Hands-on Activity III: Commanding the KUKA robot arm through Matlab

Below are the Matlab commands you will use to control the Kuka robot arm.

- `startConnection` establishes connection between Matlab and KUKA.
- `stopConnection` terminates connection between Matlab and KUKA.
- `getAngles()` returns the vector of current joint angles $q = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$.
- `stop()` stops KUKA motion.
- `moveAxis(axis,vel)` moves a single KUKA axis. `axis` takes a value between 1 and 6 that corresponds to the joint angle to be commanded. `vel` is a signed value that determines the commanded angular speed of the joint. In this lab `vel` should be set to no greater than 0.01.

- `setAngles(q,vel)` sets KUKA arm angles to those defined in `q` to within a small tolerance. `vel` corresponds to the speed of motion. For this lab, set `vel` to 0.04. To cancel the command during execution press `ctrl c` and run `stop()` in the Matlab command window.
- `setHome(vel)` sets KUKA arm to the HOME configuration. `vel` corresponds to the speed of motion. For this lab, set `vel` to 0.04. To cancel the command during execution press `ctrl c` and run `stop()` in the Matlab command window.
- `setGripper(state)` sets the gripper state. `setGripper(0)` closes the gripper; `setgripper(1)` opens the gripper.

Warning: Never use the `clear all` command in Matlab. Instead use `clearvars -except udpObj` to avoid errors from occurring.

The steps to **connect KUKA to the external PC**:

- Run `startConnection` in Matlab.
- On the KUKA SmartPad run `RSI_Ethernet.src` until the line `RSI_MOVECORR()`.
- To run the program, hold down half-way one of the enable buttons on the back of the KUKA SmartPAD. The enable buttons are labelled 3 and 5 in Figure 26. If you press too hard, an emergency stop will be activated. While holding down the enable button, press and hold the run button labelled 10 in Figure 27. Note that when running the line `RSI_ON` a warning will appear saying 'Caution - sensor correction is activated". Simply confirm this warning to continue.

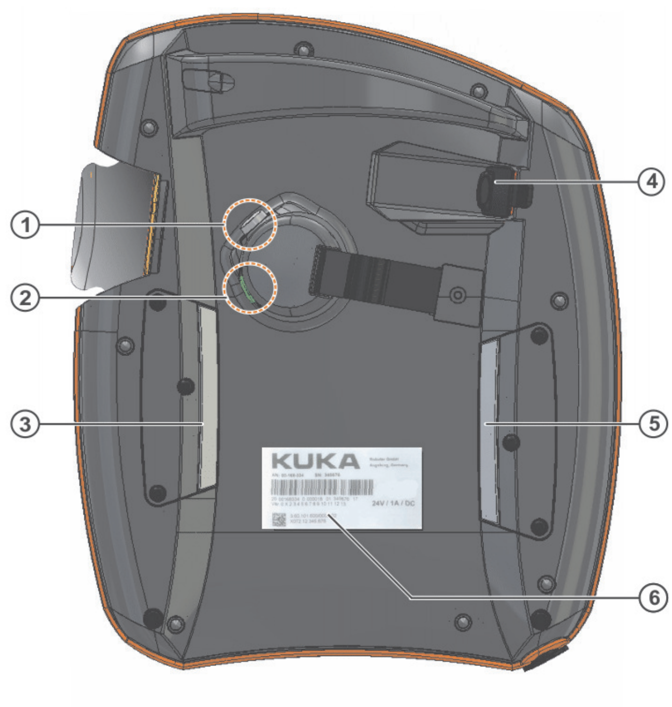


Figure 26. 3 and 5 are Enable buttons on the KUKA SmartPAD.

The steps to **command the KUKA arm from Matlab** are as follows:

- Run the desired command in the Matlab command window. The robot will not yet move.
- Hold down half-way one of the enable buttons on the back of the KUKA SmartPAD.
- While holding down the enable button, press and hold the run button. Now the robot will carry out the desired command as long as both the enable and run buttons are being held down.
- To make the robot stop moving, simply release the enable button. To cancel the current command, press `ctrl c` and run `stop()`.

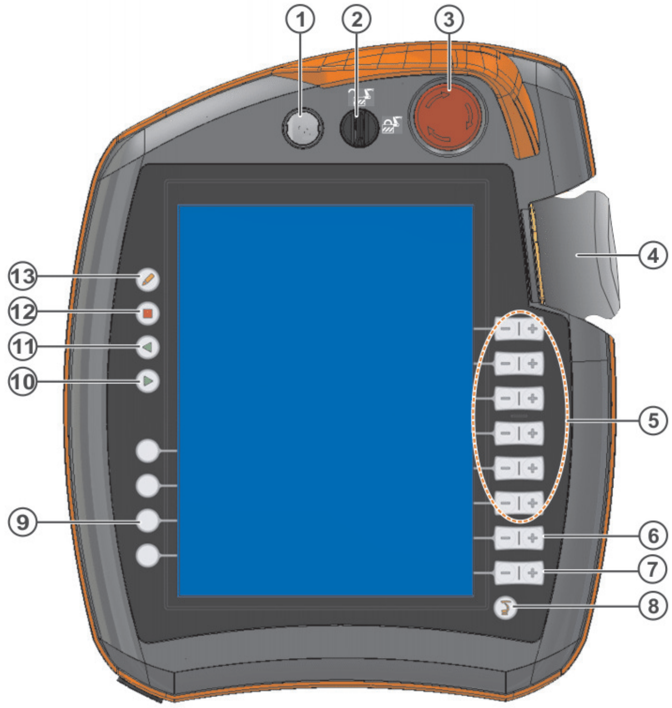


Figure 27. 10 is the Run button on the KUKA SmartPAD.

- If the robot ever hits an object while moving (such as the ground), immediately release the enable button and the robot will stop. Then press `ctrl c` in the Matlab control window to cancel the command and run `stop()`. Immediately call a TA to resolve the collision.
- If the robot does not move when commanded, it means an error has occurred. Call a TA to resolve the issue.
- For safety, students must never open the safety gate enclosing the robot unless instructed to do so. The enable and start button must never be pressed while the safety gate is open.

Warning: Make sure to FIRST connect KUKA to Matlab using `startConnection.m`**. THEN run the RSI from the Kuka SmartPad**

The steps to **disconnect KUKA from the external PC**:

- When finished running the lab, return to the program `RSI_Ethernet.src` on the SmartPAD, click on line 32 and then click `Block selection` to set the program cursor to the command `ret=RSI_OFF()`. Then continue running the program to the end. The program status indicator will turn black to indicate the program is complete.
- Run `stopConnection` in Matlab.

The steps to **deal with an RSI connection problem:**

- Type `CTRL+C` within the Matlab environment.
- There are two options available on the top portion of the SmartPAD screen (see Figure 30): **cancel** or **reset**. Use **reset**, not cancel, to reinitialize the RSI program from the Kuka SmartPAD.
- After resetting the RSI program, push the play button multiple times until the program reaches the **RSI MOVECORR()** line.
- In Matlab, run `getAngles()` to verify that the RSI connection has been correctly re-established.

Perform a few trials as follows. Command joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = (0, \pi/2, 0, 0, \pi/2, 0)$. This corresponds to the robot HOME configuration. Try commanding a few other angles within $\pm\pi/4$ of the nominal ones above.

Position the robot's end effector somewhere on the ruled area of the floor using the `moveAxis(axis,vel)` command or using the Jog keys on the KUKA SmartPad. Once on the floor, read the joint angles. Save these angles, move the robot to the HOME configuration, and then command the saved angles. You should verify that the robot's end effector returns to the same position you chose earlier.

8. Robot Programmed Motions

In this section you will use preset programs to move the robot. You will use the smartPAD to accomplish this task. In order for you to choose a program and execute it, follow the procedure below.

- Select the program from the smartPAD as shown in Figure 28

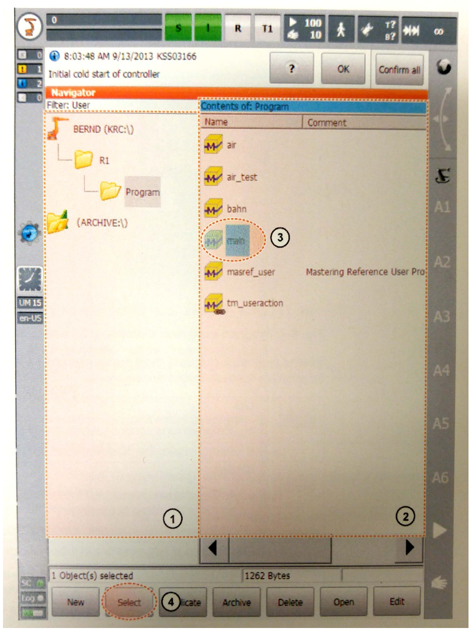


Figure 28. smartPAD Navigator

Item	Description
1	Navigator - directory and driver structure
2	Navigator - directory and data list
3	Selected program
4	Soft button for selecting a program

Table 5. Table of functions

- Option 1: Use the `Start Forward` or `Start Backward` keys on the smartPAD to run the program.
- Option 2: Set program velocity (program override, POV);

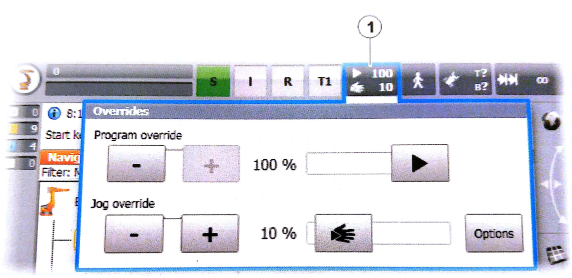


Figure 29. Setting Program Velocity

- Reset the program. A selected program can be cancelled (closed) or reset (that is, started from the top) by pressing the program state icon.
- Press the enable switch;

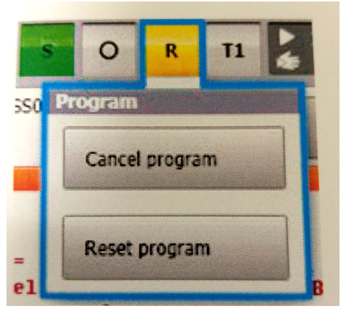


Figure 30. Cancel or Reset a Program

- Press and hold down the start key
- Once the end position has been reached, the motion is stopped. The colour of the program state icon changes to *red*.
- Continued sequence (depending on what operating mode is set):
  T1: continue with the program by pressing the Start key;
  AUT: activate drives then start the program by pressing the Start key.

8.1 Creating a New Program Module

As the operator, you are allowed to use the Program directory to create and save a new program module. This module can be placed at any location in the directory tree. You can create sub directories and store your program modules there as well. Program modules are identified by the letter `M`. You can add a comment that uniquely identifies or describes the module. Follow these steps to create a program module:

1. In the directory structure, select the folder in which the program is to be created (Figure 28)
2. Press the `New` soft key (Figure 28)
3. Enter a name for the program, and a comment if desired, and confirm by pressing `OK`.

8.2 Creating Point-to-Point (PTP) Motion

Follow the procedure below to create point to point motion using the robot.

1. Set the mode to `T1`;
2. Select a program;
3. Move the TCP to the desired end-point position;
4. Place the cursor in the line immediately after that in which the motion instruction will be placed;
5. From the menu, navigate `Command --> Motion --> PTP`. At this point, an inline form appears, which looks like Figure 31 below.

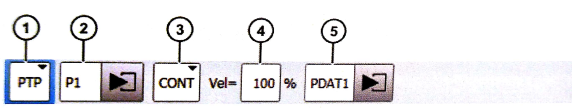


Figure 31. Inline form for PTP motion

6. Enter the parameters according to the table below:
7. Enter the desired data on the option window labelled `Frames`. The window is presented on Figure 32, and the below explains each option.
8. On the motion parameter window, enter the acceleration and approximate distance (`CONT` option).
9. Save instruction with `CMD OK`. The current position of the TCP is considered the end point.

Item	Description
1	Motion type PTP;
2	Name of the end point is set automatically;
3	`CONT`: end point is approximated (empty of motion stops at endpoint);
4	Velocity (1 to 100%);
5	Motion dat set: acceleration, approximate distance;

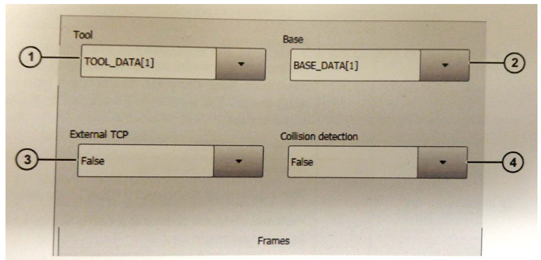


Figure 32. Option Window: `Frames`

Item	Description
1	Tool selection. External TCP if **True**. Values 1 - 16;
2	Base selection. External TCP if **True**. Values 1 - 32;
3	**True** if tool is fixed, **False** if tool is mounted on flange;
4	**True** if controller calculates torque, **False** otherwise;

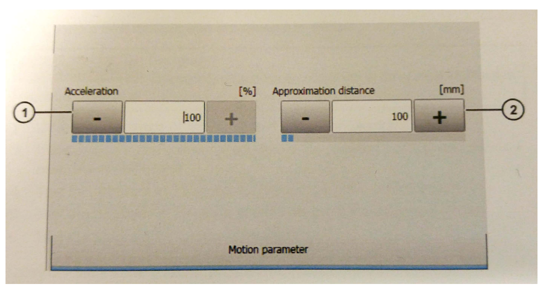


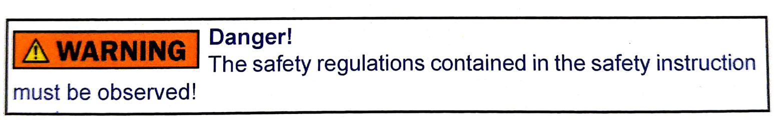
Figure 33. Motion Parameter Window (PTP)

Item	Description
1	Acceleration on leading axis (1 to 100%);
2	This shows when `CONT` selected. Used to calculate approximate end position;

9. Hands-on Activity III - Creating a Program with PTP

Your task here is to create your own program for the robot to execute. Follow the steps below. Some of the tasks will require that you recall what you have seen through this experiment, or read in this document.

1. First of all, use the picture below as a reminder;



2. Create a new program module, and name it Activity03
3. Teach the robot the 12 points shown on the worktable (the green numbers highlighted on Figure 34), using the *green* base

coordinate (the one you have used before) and `Pen1` as the tool. This refers to the pen that the robot will hold, that is part of the workstation.

Point number 1 should be the *home* and *end* position, or where the motion begins and ends;

The jog velocity should be set to 10%;

Make sure the longitudinal axis of the tool is *always* perpendicular to the path on the worktable;

Leave box number 3 of the inline form (for PTP motion) empty. This refers to Figure 31.

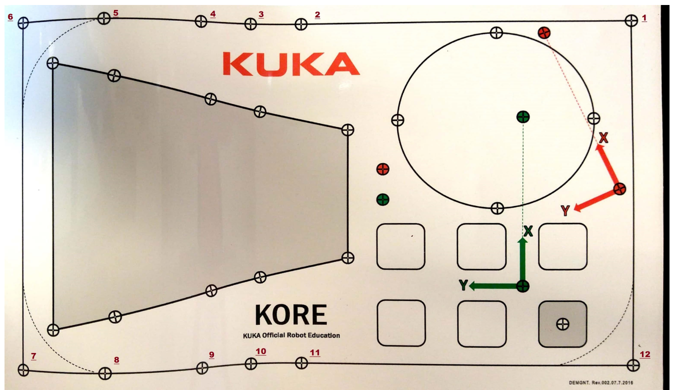


Figure 34. Worktable with trajectory points highlighted

4. Test your program in `T1 mode`. Follow the necessary safety advice given before in this document;
5. Run your program in `Automatic mode`. Again, make sure that the safety advice is followed;
6. Now set box number 3 (refer to Figure 31) of the inline form for PTP motion to `CONT`. Set acceleration parameter to XXXX and approximate distance to YYYY
7. Test your program in `T1 mode`. Pay attention to the safety recommendations;
8. Run the program in `Automatic mode`. Again, do so safely.

10. Conclusion

The experiment today lead you to explore the industrial robotic workstation in our robotics lab. The instructions and steps you have followed today relate to how a robot *operator* would prepare such station to perform a task such as pick-and-place, welding, etc. You are being trained to be a *roboticist*, so it is very important for you to understand how to robot operation is conducted, so that you can assist in the design of robots for different applications. It is hoped that in this experiment you had the opportunity to develop a greater appreciation for the application of robotics and robot operation.