

In this Module we will be diving into Robotics and Robotics Simulation by programming and creating movement of complex machines at various scales. We will start with a desktop arm called the UArm Swift Pro. Its inner workings and programming are similar to that of our Kuka's operation and workflow. The UArm allows you to learn the basics at a

small and approachable scale. Gradually you will gain the knowledge required to scale up to the complexities and challenge of the Kuka arm. The assignments will run you through the following programs to operate the various equipment covered: **UArm Factory | RoboDK**

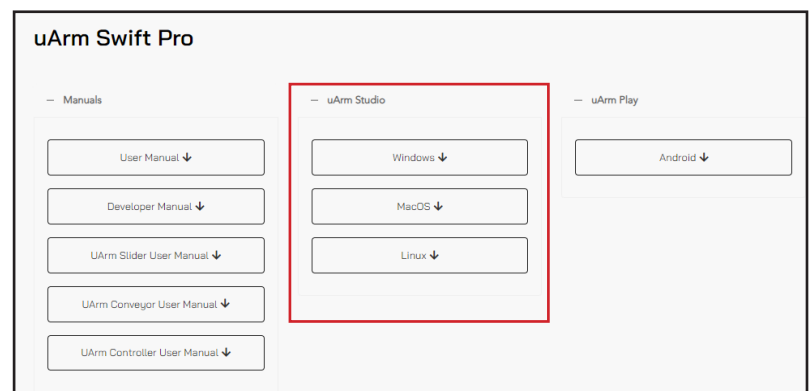
Robotics is the culmination of everything you have learned up until this point. You began with line work and plotting leading to laser cutting moving from the 2D to the 3D world. Once in the 3D world utilizing your laser cut geometry to create motion with the principles of simple mechanics. All of these techniques, principles, and equipment learned combine into a experience and knowledge base to have a deeper understanding of programmable robots that are capable of tasks larger and more efficient than human scale.

We are going to build up in scale in terms of the various programs and equipment using a more approachable, user friendly, smaller scale UArm Swift Pro. Its "brain" is an Arduino Mega 2560 and is using 3 Nema 17 motors and a servo. It is also compatible with the Grove Sensor Kits we have been using in our Electronics Module. This adaptable platform will allow you to experiment with robotic movement and coding that transfer to larger machines such as the Kuka.

The first step is to download UArm Studio which is located under the UArm Swift Pro Section : <https://www.ufactory.cc/download-UArm-robot>

You will need to download either the Windows or Mac versions dependent on your computer. This software controls the robot arm in 3 different ways. One is with a digital joystick, allowing us to move each axis with sliders. Two is with Blockly Coding, similar to Grasshopper but a bit simpler, which allows us to make repeatable actions. Lastly with "Teach" mode allowing us to move the robot with our hands and record its actions to be repeated.

UArm Swift Pro



UArm Studio is the proprietary software that comes with the UArm Swift Pro. It is considered an introductory software that will allow you to adjust each axis and gain an understanding of how the robot operates with approachable software and operation that is user friendly.

It will also show you the wide capabilities of the platform such as sensor implementation learned in previous modules and various tool heads that can be easily interchanged. After you have a basic understanding UArm Studio you will delve deeper into the program's various operational modes.

## Step 1 Open UArm Studio

Step 2 Connect Power and USB to Computer if not already connected. Press the Power button on the base of the robot arm. (You will hear it power up and the light turn on)

Step 3 Ensure that your UArm is connected on the entry screen in UArm STUDIO by checking the right hand of the screen under the image of the UArm. It should have a green dot and say connected.

Step 4 Scroll through the "Getting Started with UArm Studio" page, this will show you a basic overview of how the software operates and the various tools available for the UArm.

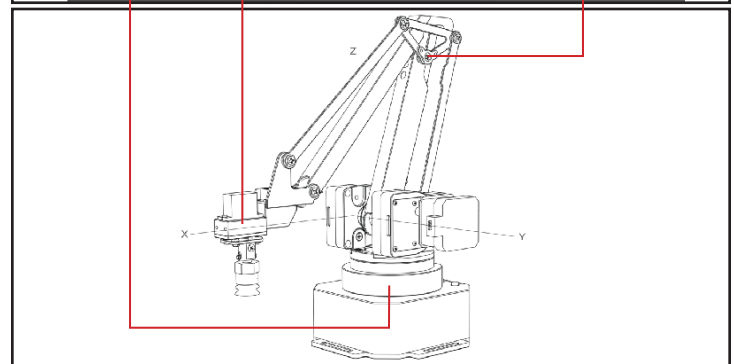
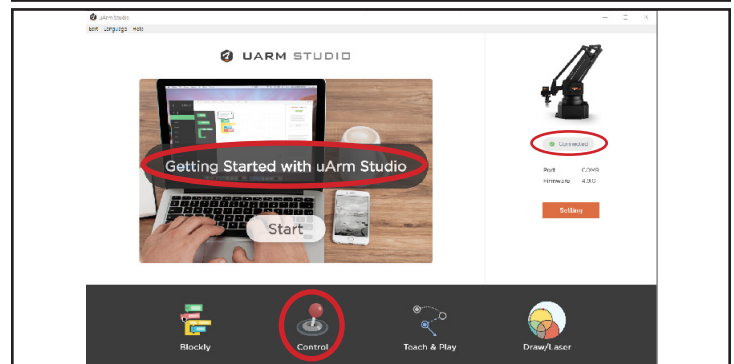
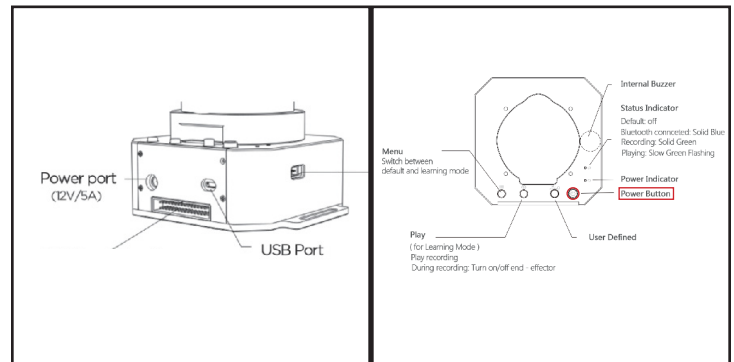
Step 5 Open the Control Tab, and become familiar with the different blue sliders. Check which tool is attached to the UArm and switch the toggle to check if it works (it will turn blue when activated and you will see it move).

Step 6 Now, set to a slower speed, play with each axis of the arm as well as the tool currently attached to the arm. Reference the diagram to the right to see which axis is controlled by which slider.

\*Do not over extend the robot arm, cause any lift of the base off the table, or exceed safe speeds. Please keep area clear of unnecessary objects and stay out of range will in operation.

Step 7 Try to grab the red cube on the desk using the control mode.

Proceed to M6A1b



UArm Studio's "Teach and Play" mode allows for quick programming of the robotic arm's position and tool. There are two options for programming in this mode which are software control and hardware control. As seen in the picture below, there are instructions on how to program the robot arm through the use of the buttons on the base of the arm.

These will allow you to position and activate the tool in 3D space. As an alternative, you can do the same thing through software control by using the "Start Recording" button next to the image.

Step 1 Open UArm Studio

Step 2 Connect Power and USB to Computer if not already connected. Press the Power button on the base of the robot arm (You will hear it power up and the light turn on)

Step 3 Ensure that your UArm is connected on the entry screen in UArm STUDIO by checking the right hand of the screen under the image of the UArm. It should have a green dot and say connected.

Step 4 Review the "Getting Started with UArm Studio" to examine the controls and understand the workflow for recording movement.

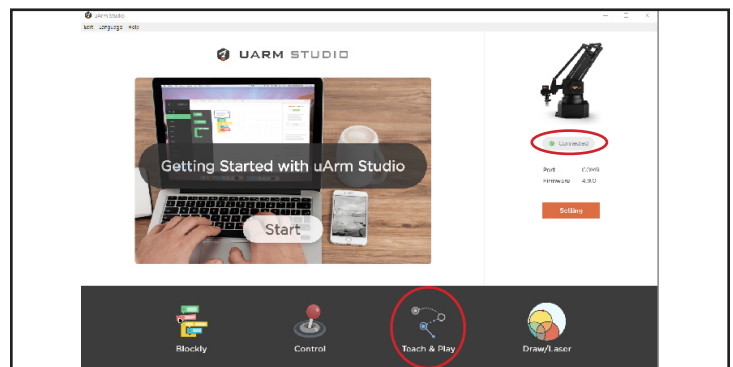
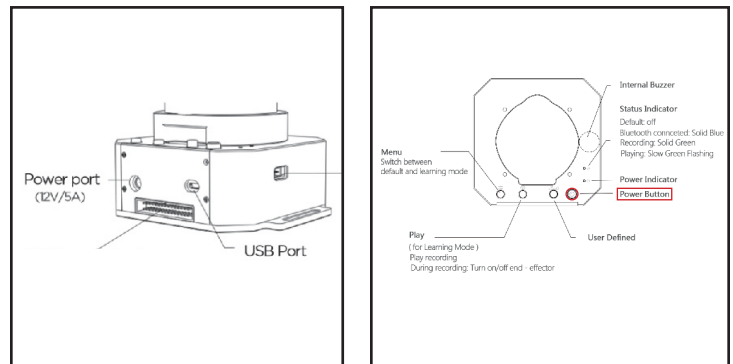
Step 5 Select the "Teach and Play" Mode

\*Make sure the "Locked" toggle is turned off to operate the robot arm. Do not over extend the robot arm, cause any lift of the base off the table, or exceed safe speeds. Please keep area clear of unnecessary objects and stay out of range will in operation.

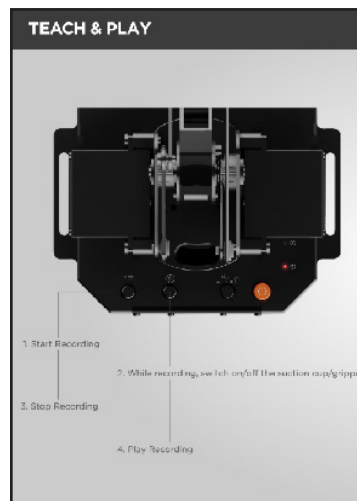
Step 6 Attempt a "Sweeping" recording where you have the arm complete a full range of motions with the software control option. Then finish the recordings and select play to see how it mimics your motions.

Step 7 Now, that you understand the process use both the software control mode to push the red cube around the table (make sure to mark your starting point of the red cube with tape or piece of scrap paper)

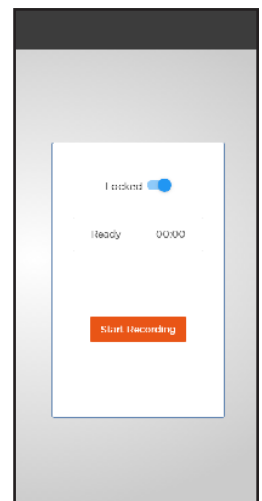
Proceed to M6A1c



Hardware Control



Software Control



After completing the previous pages you now have a better understanding of the basic movement, operation, and workflows. Now we will begin with incorporating the use of simple code to control movement and create scripts. UArm Studio's Blockly control mode illustrates programming of the robotic arm's position and tools with a user friendly puzzle

piece interface. This gives an approachable introduction to the way code functions and creates different operations, which is similar to Arduino Code. Also, it is compatible with our Grove Sensors from Module 5 which act as data inputs. This makes programming complex movements interacting with environmental data and generating loops easier and quicker.

## Step 1 Open UArm Studio

Step 2 Connect Power and USB to Computer if not already connected. Press the Power button on the base of the robot arm (You will hear it power up and the light turn on)

Step 3 Ensure that your UArm is connected on the entry screen in UArm STUDIO by checking the right hand of the screen under the image of the UArm. It should have a green dot and say connected.

Step 4 Review the "Getting Started with UArm Studio" to examine the controls and understand the workflow for recording movement.

\* Do not over extend the robot arm, cause any lift of the base off the table, or exceed safe speeds. Please keep area clear of unnecessary objects and stay out of range will in operation.

Step 5 Select the "Blockly" Mode and select the "Motion" tab to start creating blocks of code with each of the highlighted components (select current attached tool). You will need to complete the missions by following the prompts under the task menu to the right of the screen.

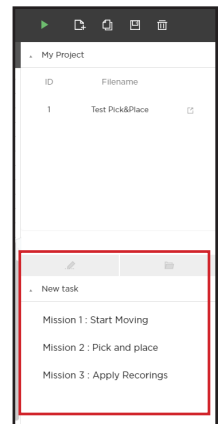
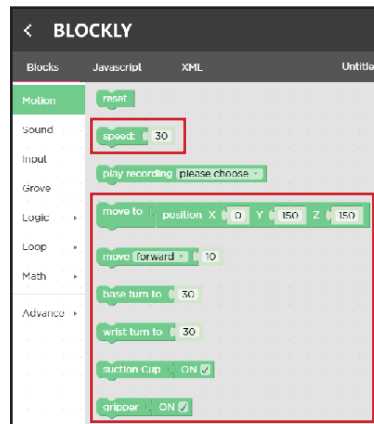
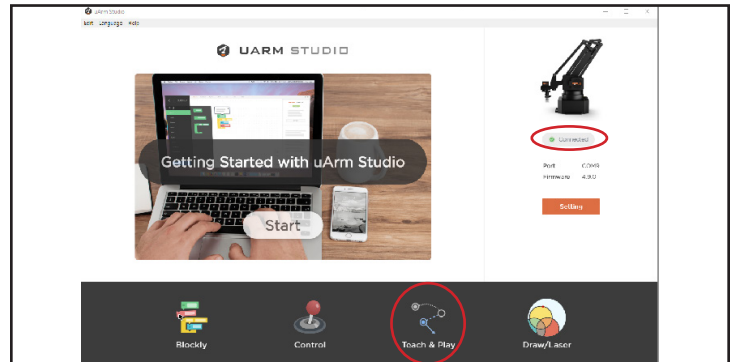
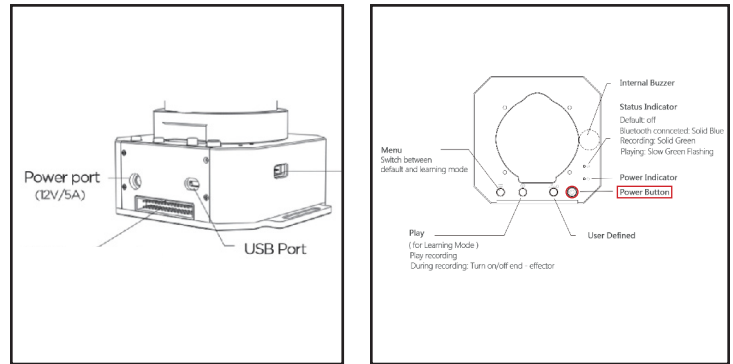
\*You will need to input numbers or select different options from the drop down bars. Make sure to test the numbers before playing script. Wrist and base turn is based on the 180 degree turn radius.

Step 6 Complete "Missions" 1: Start Moving

Step 7 Complete "Missions" 2: Pick and Place

Step 8 Complete "Missions" 3: Apply Recordings

Proceed to M6A2



Since completing the pages with UArm Studio programs you now have the basic knowledge and skills to moving forward to conceptualizing 3D space with the approachable small scale UArm Swift Pro. Progressing to a new software called RoboDK that is an industrial robotics station simulator. It allows users to simulate complex assemblies and visualize

a script in a similar interface to the programs learned in CAP. For now, we will be using this software to connect our usage of a small desktop robotic arm to that of the large industrial scale Kuka in the CAP basement. We will be using an online playlist to explain the ins and outs of the software as it is very well documented by official RoboDK professionals.

Step 1 Download RoboDK and install at (<https://robodk.com/download>)

Step 2 Open Software and select the “Earth” icon on the top left next to the folder and save icon.

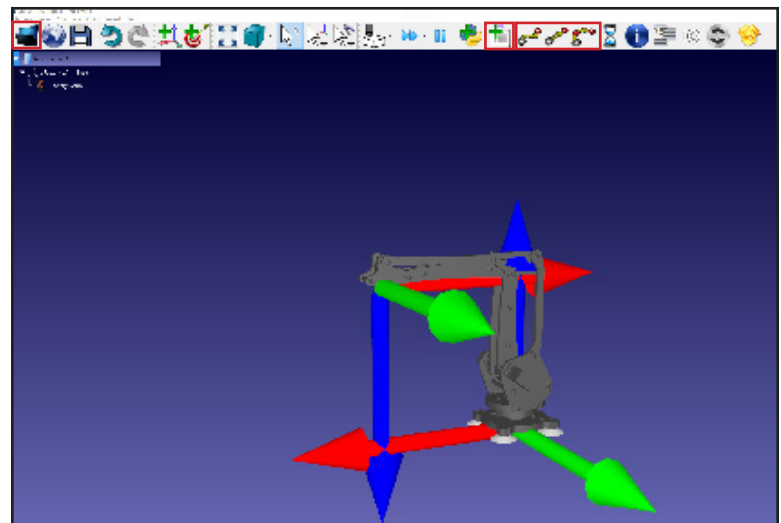
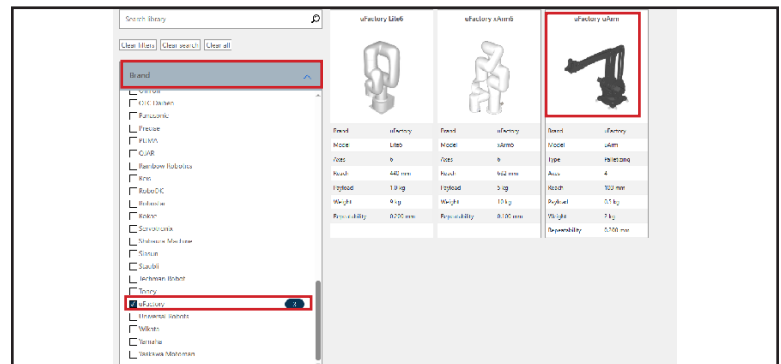
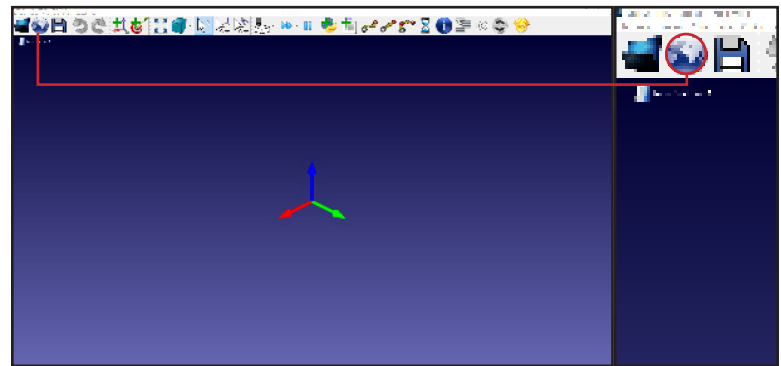
Step 3 Once the browser window opens up go to the left and select Brand>UFactory>Ufactory UArm>Download. Then go back to RoboDK and you will click the folder icon next to the earth icon and then select the UArm.

Step 4 Create a linear, joint, and circular movement and play the simulations of these different types of movement for a staff member to approve. If you need further assistance on how to do this after watching the videos proceed to the next pages for help.

That is the software controls at its most basic level. I highly recommend watching the tutorial series Module 01 CH1- 10 below to further understand how create usable programs.

[https://www.youtube.com/watch?v=qBSQhG8\\_HCQ&list=PLjiA6TvRACQexjP3pB1YOaxT-aTLG0GO2&ab\\_channel=RoboDK](https://www.youtube.com/watch?v=qBSQhG8_HCQ&list=PLjiA6TvRACQexjP3pB1YOaxT-aTLG0GO2&ab_channel=RoboDK)

Make sure to watch the Module 01 CH 11 and CH 12 videos as these will walk you through how to export the code required to post to the robots but feel free to continue on with more videos if you are interested in learning more.



<b>MOD 1 CH1: Set Up</b>	RoboDK download and installation
<b>MOD 1 CH2: User Interface</b>	
Submission Environment	3D workspace
Station Tree	Parent- child folder system
<b>MOD 1 CH3: Navigation</b>	
Pan	Middle mouse button
Zoom	Scroll middle mouse button
Rotate	Right click
Views	3D box icon in toolbar
Hide/Show Reference Frames	Right click robot
Hide/Show Tools	Right click tool in station tree
Enlarge / Reduce Reference Frame Size	+ or -
<b>MOD 1 CH4: Robot Panel</b>	
Robot Panel	Double click robot
Reference Frame	XYZ of a reference plane
Hide/Show Names	/
Hide/Show Robot Workspace	*
<b>MOD 1 CH5: Positioning</b>	
Tool Center Point (TCP)	The central axis of the tool movement
Hide/Show XYZ Labels	Main menu> Tools> Options> Display (click display xyz axis letters)
Show and Move Along Axis, Rotate Along Axis, and or Move Along Shared Planes	Hold ALT and click then drag selected axis
Align/Home	Open robot panel and select align or home button
Jog Robot	Open robot panel, select a reference frame, select an axis, and then click/drag the jog wheel to move (*finer control, click jog and drag out past jog wheel boundary*)
<b>MOD 1 CH6: Targets</b>	
Targets	A recorded position in space (including XYZ position and tool orientation)
Create New Target	Move robot into selected target location and click the bullseye plus icon in the toolbar
Move to Target	Click or double click selected target in station tree child
Rename Target	Click selected target and F2
Adjust Simulation Speed	Click play icon for normal speed, click fast-forward icon for fast speed, and or click drop down menu next to those icons to use a slide bar
Target Panel	Click selected target and F3
<b>MOD 1 CH7: Targets Continued</b>	
Manual Modify Target Position	Hold ALT and select axis, rotation, or plane to modify position and then right click this target in the station tree and select "Teach Current Position"
Cartesian Modify Target Position	Open target panel of selected target and enter new values (click ALT to move to next box) and then click "Teach Current Position"
*Note	When modifying targets if there is an error there will appear a warning sign on a target icon in the station tree
Select Target Without Robot Simulation Movement	CTRL click selected target
Singularity	A point that a robot cannot cross with a linear motion

<b>MOD 1 CH8: Program and Motion</b>	
New Program	Paper icon with plus symbol in the toolbar
Rename Program	F2
New Joint, Linear, or Circular Motion	Select desired program and target in the station tree and then select corresponding motion icon in the toolbar that will appear under parent program
Select Multiple Objects	Hold CTRL and select objects (for objects not back to back) or select first object the SHIFT click last object (for back to back objects, like a column of motions or targets)
Link Targets	Hold CTRL and select motion and target in station tree to link together (linked target name should appear next to motion in station tree)
Run Program Simulation	Double click program icon in station tree
Change Motion Type	Right click motion icon in the station tree and select new motion type
Remove Transparent Robot Arm	Click ESC
Hide/Show Targets	Select Targets and click /
Hide/Show Motion Path	Right click program icon in station tree and click "display path" box
Hide/Show TCP	Right click robot and select "visible TCP" box
Loop	Right click program icon in station tree and select "loop" box and to stop loop press ESC
<b>MOD 1 CH9: Reference Frames</b>	
Reset Robot Position	Right click robot arm in the station tree and select reset
Create New Reference Frame	Select robot arm base in station tree and then select plus sign with a red line icon next to the create new target icon in the toolbar
Change Reference Frame for Targets	Select all targets and right click select "change support" then select the desired reference frame
Active Reference Frame/Tool	Will be displayed as a green dot next to their icons in the station tree
Change Reference Frame for Program	Right click reference frame in station tree under the selected program and right click "set reference link" and click desired reference frame
Slider Control of Program Simulation	The blue slider at the bottom of the submission environment
<b>MOD 1 CH10: Configuration</b>	
Configuration	Different set of joint values that would equal or end up bring the tool to the exact same Cartesian XYZ and rotation position
Configuration Panel	Double click the robot arm and it is at the bottom of the robot panel and use middle mouse button to scroll through the option (select more options to filter configuration by joint orientation)
Linear Motion	Point-to-point motion with no curve that is the fastest possible path
Joint Motion	Also known as Point-to-Point motion, is a method of path interpolation that commands the movement of the robot by moving each joint directly to the commanded position so that all axis arrive to the position at the same time. The path is predictable, however the path will not be linear.
Robot Movement with Respect to Joint Configuration	Single click of target in station tree will be a linear movement which does not respect joint configuration Double click of target will be a joint motion which will respect joint configuration
Set Target Joint Configuration	Select target in station tree and select desired joint configuration then open target panel and click "teach current position"

Since completing the pages with UArm Studio programs you now have the basic knowledge and skills to moving forward to conceptualizing 3D space with the approachable small scale UArm Swift Pro. Progressing to a new software called RoboDK that is an industrial robotics station simulator. It allows users to simulate complex assemblies and visualize

a script in a similar interface to the programs learned in CAP. For now, we will be using this software to connect our usage of a small desktop robotic arm to that of the large industrial scale Kuka in the CAP basement. We will be using an online playlist to explain the ins and outs of the software as it is very well documented by official RoboDK professionals.

### Linear, Joint, and Circular Motion Type

Step 1 Open RoboDK and import your UArm model via the folder button icon on the toolbar

Step 2 Create a set of four targets via the bullseye plus icon in the toolbar.

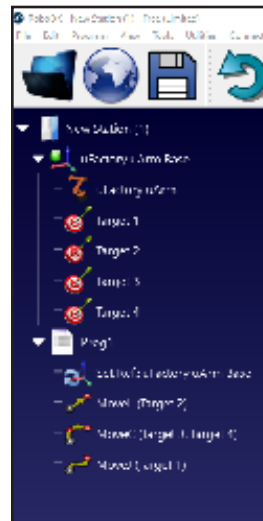
Step 3 Create a new program via the page plus icon in the toolbar

Step 4 Create a set of motions with at least one of each motion type under the new program

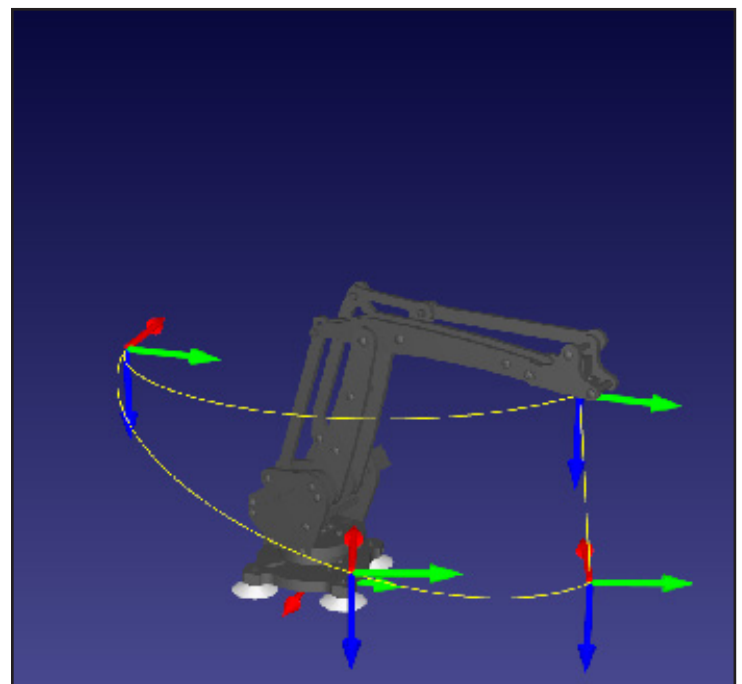
Step 5 Link the different motion types with the different targets by right clicking the motion type in the station tree then selecting target linked and selecting the desired target

Step 6 If you receive a error, modify your target positions to allow for proper space for the motion type. Hold ALT to modify target position and right click the target icon in the station tree and select teach current position.

Step 7 Save your station under the main menu> file> save station as. Then schedule a consultation in service now to have a staff member view your file to move forward with the KUKA M6A3 module. Make sure to note in your consultation request that you have completed M6A2 and need to have your file tested.



Station File Example





In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, tools, and operation. First, we will begin with safety and risks. As with any piece of equipment there are risks and dangers if not used properly especially with such large scale industrial robotics such as our KUKA robot arm.

It is essential to remember and follow these key safety points to avoid injury to yourself and others. In the photos below are examples of the different zones, collision risks, risk of electrocution, and warnings to observe which can seriously harm an individual. Once you have complete understanding of the risks and dangers of operating a KUKA robot arm you can continue on to learn the other items mentioned to the left.

## Key Safety Points

1. One person in the cage at a time while operating the robot arm. If multiple operators are needed to avoid injuries:
  - All persons must have an unimpeded view of the robot
  - Eye-contact between all persons must be possible at all times
  - Openly communicate with those around with confirmation that the communication has been received
2. While operating the robot arm stay inside the safety zone to avoid the risks associated shown in the diagram to the right \*this area will be marked on the floor with colored tape\*
3. Always test any movements planned and when testing a manual movement or a programmed script always test in T1 (manual reduced velocity mode).
4. Never touch or approach the robot arm while it is moving or stand underneath it ever.
5. Handle the areas of high voltage risk (shown to the right) with caution and if necessary the proper PPE.
6. Always properly secure a tool and turn off if necessary to avoid any injuries. When finished always remove and store tool properly.
7. Always set a base and touch point for each project with as much accuracy as possible.
8. Be aware of your surroundings while operating and do not leave it operating unattended.
9. If you are unsure of anything always contact the staff.
10. If something goes wrong or you suspect something to go wrong immediately stop the machine or hit the emergency stop if necessary.
11. Always make sure the space is clean after use and that the doors are locked when finished.
12. When operating in T2 (manual high velocity mode) there should never be anyone in the cage or out of the safety zone.
13. Avoid lock up and over extension of the axes and robot arm.

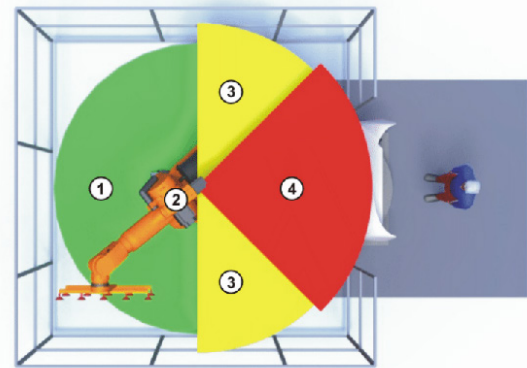


Fig. 2-1: Example of axis range A1

- |   |             |   |                   |
|---|-------------|---|-------------------|
| 1 | Workspace   | 3 | Stopping distance |
| 2 | Manipulator | 4 | Safety zone       |



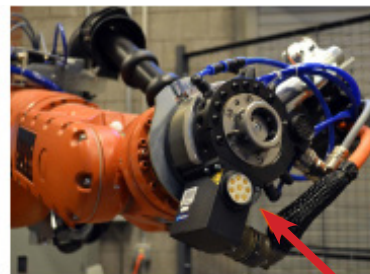
TOOLING RISK



PINCH RISK



STRIKE RISK



HIGH VOLTAGE RISK

# Basics to Robotics- KUKA Parts and Terms 3b

In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with the important terms to remember and the main components of a KUKA robot arm system.

An industrial robot is a re-programmable, multi-functional manipulator designed to move material, machine tools, or specialized devices through variable programmed motions for the performance of a variety of tasks. A robot arm is the combination of the axes, motor, joints, and the links connecting them.

## Important Terms

**Axis Range** of each axis, in degrees or millimeters, within which it may move. The axis range must be defined for each axis.

**Stopping Distance**- Stopping Distance = reaction distance + braking distance. The stopping distance is part of the danger zone.

**Operator (User)**- The user of the industrial robot can be the management, employer or delegated person responsible for use of the industrial robot.

**Danger Zone**- The danger zone consists of the workspace and the stopping distances.

**KCP**- KUKA Control Panel, the KCP has all the operator control and display functions required for operating and programming the industrial robot.

**KUKA smartPAD**- The smartPAD has all the operator control and display functions required for operating and programming the industrial robot.

**Manipulator**- The robot arm and the associated electrical installations.

**Safety Zone**- The safety zone is situated outside the danger zone

**Stop Category 0**- The drives are deactivated immediately and the brakes are applied. The manipulator and any external axes (optional) perform path-oriented braking.

**Stop Category 1**- The manipulator and any external axes (optional) perform path-maintaining braking. The drives are deactivated after 1s and the brakes are applied.

**Stop Category 2**- The drives are not deactivated and the brakes are not applied. The manipulator and any external axes (optional) are braked with a normal braking ramp.

**T1**- Test Mode, manual reduced velocity (<= 250 mm/s)

**T2**- Test Mode, manual high velocity (>250 mm/s permissible)

**External Axis**- Motion axis which is not part of the manipulator but which is controlled using the robot controller

## The Three Primary Components Of An Industrial Robot System



Teach Pendant  
"smartPAD"



Control Cabinet



KR60-3 KUKA Robot ARM

In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with the workspace and other various terms of a KUKA robot arm system and space.

The work cell, work envelope, and barriers of our KUKA robot arm is shown in the photo below. The pinch points safety risks covered in the safety portion will occur within the work envelope shown in the diagrams below within the full extension radius of the robot arm and attached end of arm tooling (EOAT).

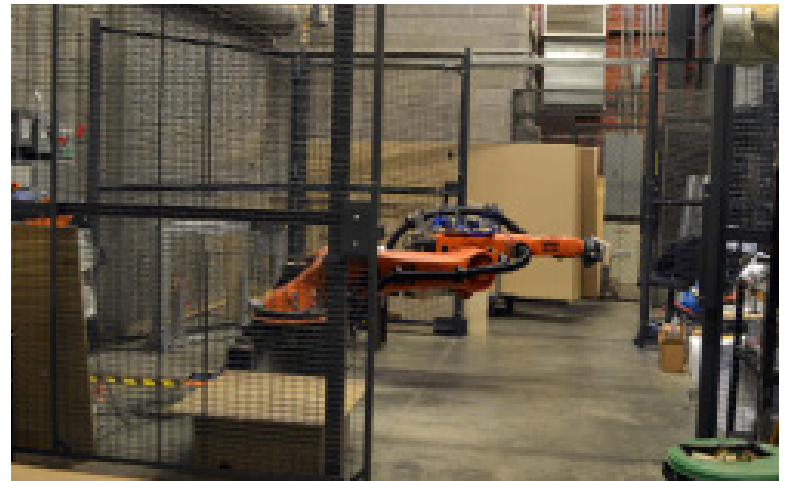
## Terms

### Work Cell: CAP FAB LAB CNC/KUKA Room

The entire room or space that the robot system occupies where people should not be during Auto-mode operations.

### Barriers: Metal Wire Safety Cage

Physical of Safety interlock devices that bound the Work Cell to limit/prevent improper human entrance



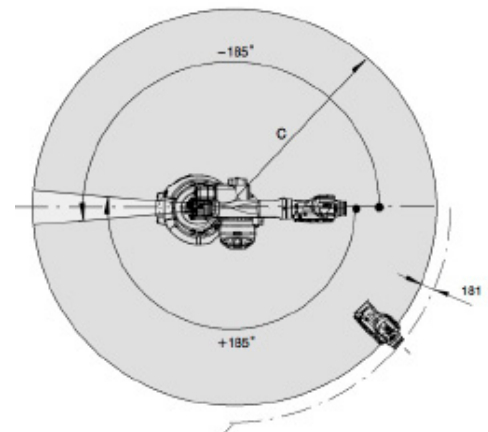
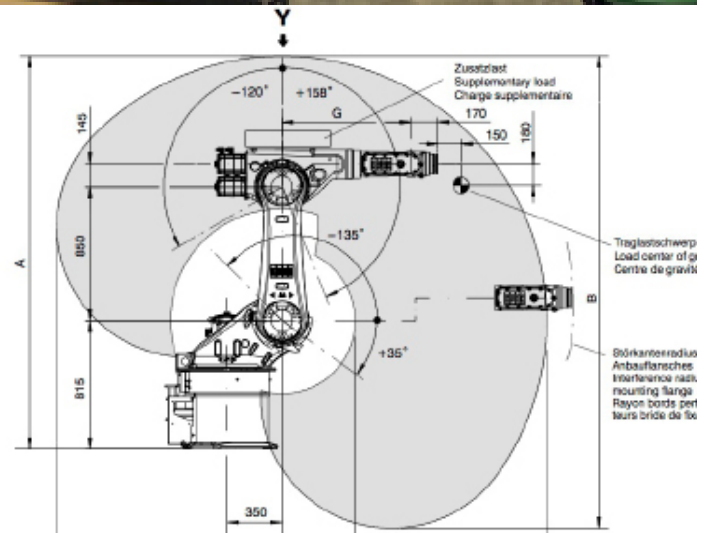
### Work Envelope

Complete space that the Robot Arm + EOAT can reach. IE. the robot's working volume.

### Pinch Points

Please in or on peripheries of the Work Envelope, or Arm, or Tooling, where things can be pinched and/or crushed.

**EOAT = END OF ARM TOOLING**



In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with the different axes and orientation of their rotations and how conceptualize them within a 3D space.

Now that you understand some basic terms and components, the workspaces, and the safety and risks you will need to understand how the industrial robot arm creates movement. The large scale industrial robots move mechanically differently at its joints than the small scale UArm desktop. After this you will move forward to the process of operation of the KUKA robot arm.

## Axis-Specific Motion

### AXIS = JOINT = MOTOR

The center of rotation that connects and rotates the links of the robot arm using a servo-motor.

### 1-3: POSITIONING

Typically used for tool

### 4-6: ORIENTATION

Typically used for tool

### Angle of Rotation About and Axis

Angle A = Z Axis

Angle B = Y Axis

Angle C = X Axis

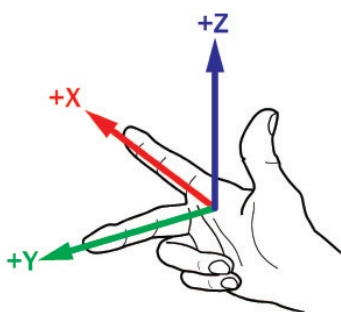
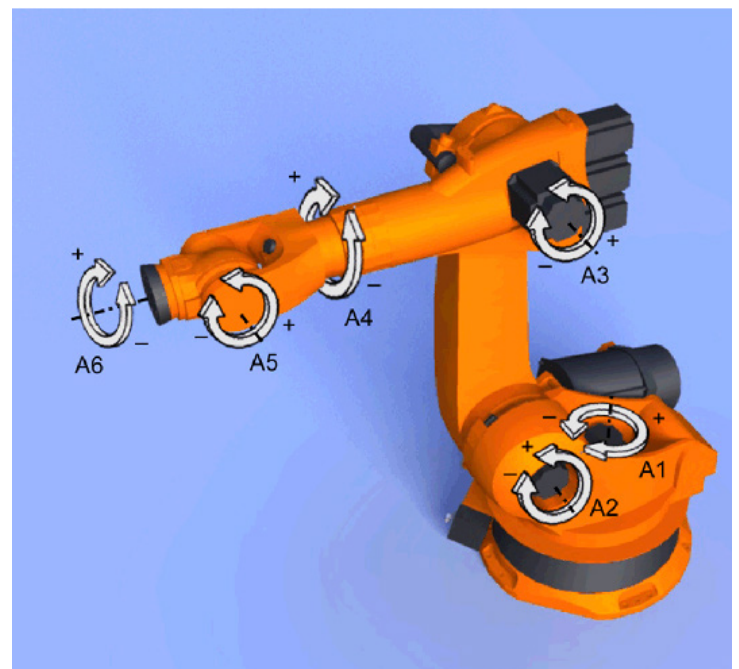
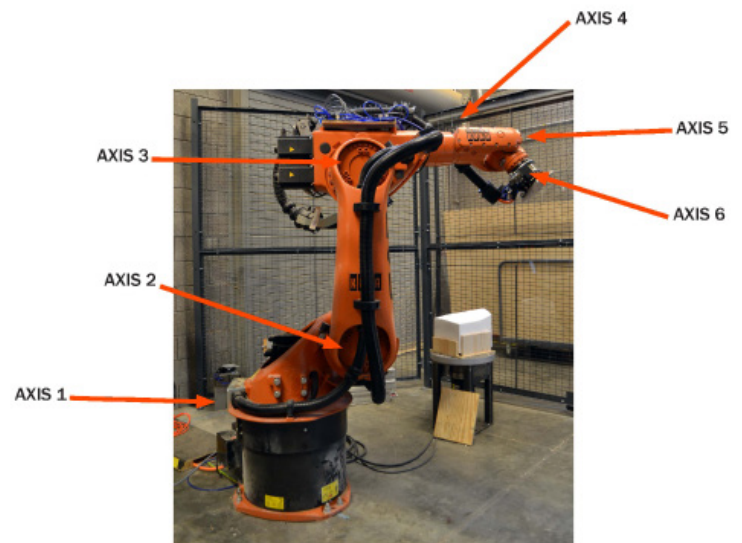
### Right Hand Rule

Used for right-handed coordinate systems

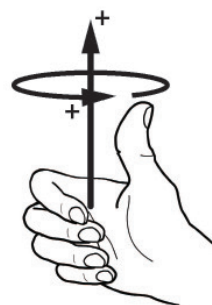
Thumb = Z Axis

Index finger = X Axis

Middle finger = Y Axis



ROBOT RIGHTHAND RULE



RIGHTHAND ROTATION RULE

In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with various types of coordinate systems that the of a KUKA robot arm system requires to understand where, how it needs to move, and understand the space around it.

When creating a new program for an industrial robot it requires different sets of knowledge to be able to complete the program. These sets of knowledge are the guidance system for robot arm to understand what tool it has and what base it uses as a reference frame to know where to go in respect to its own robot base and the world coordinate system around them. These are based on axis directions and targets, or points, that create paths to follow.

## Overview of Coordinate Systems

### WORLD

The WORLD coordinate system is a permanently defined Cartesian coordinate system. It is the root coordinate system for the ROBROOT and BASE coordinate systems. By default, the WORLD coordinate system is located at the robot base.

### ROBROOT

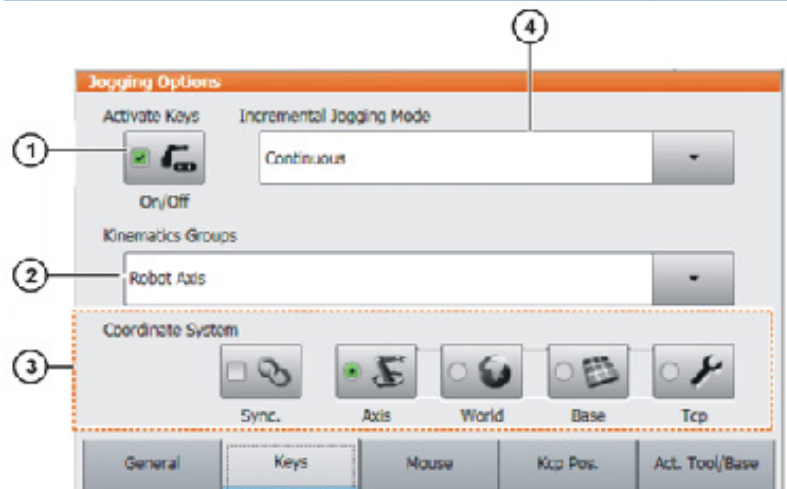
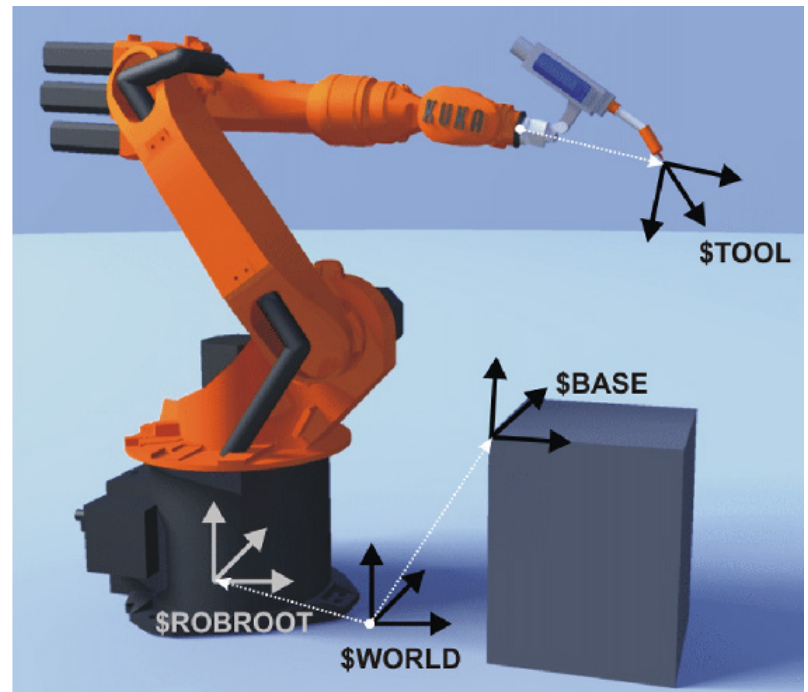
The ROBROOT coordinate system is a Cartesian coordinate system, which is always located at the robot base. It defines the position of the robot relative to the WORLD coordinate system. By default, the ROBROOT coordinate system is identical to the WORLD coordinate system. \$ROBROOT allows the definition of an offset of the robot relative to the WORLD coordinate system.

### BASE

The BASE coordinate system is a Cartesian coordinate system that defines the position of the workpiece. It is relative to the WORLD coordinate system. By default, the BASE coordinate system is identical to the WORLD coordinate system. It is offset to the workpiece by the user.

### TOOL

The TOOL coordinate system is a Cartesian coordinate system which is located at the tool center point. By default, the origin of the TOOL coordinate system is located at the flange center point. (In this case it is called the FLANGE coordinate system.) The TOOL coordinate system is offset to the tool center point by the user.



## Types of Jogging Motion

### Cartesian Jogging

The TCP is jogged in the positive or negative direction along the axes of a coordinate system.

### Axis-Specific Jogging

Each axis can be moved individually in a positive and negative direction.

In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with various types of motion that the of a KUKA robot arm can utilize to move around in a 3D space and how they are each different.

When creating a new program for an industrial robot it requires different sets of knowledge to be able to complete the program. These sets of knowledge are the guidance system for robot arm to understand what type of motion is possible or required to generate the necessary movements. These paths are can vary based on the point data it receives and what is possible as to each has limitations and advantages to how it can preform and move.

## Motion Types

### Point to Point: PTP

The robot guides the TCP along the fastest path to the end point. The fastest path is generally not the shortest path and is thus not a straight line. As the motions of the robot axes are rotational, curved paths can be executed faster than straight paths. The exact path of the motion cannot be predicted.

### Linear: LIN

The robot guides the TCP at a defined velocity along a straight path to the end point.

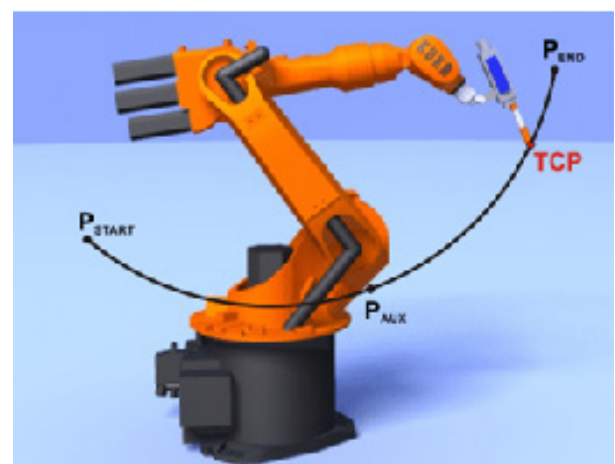
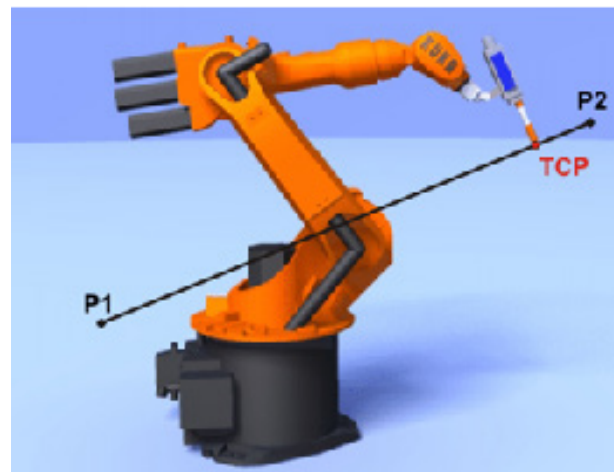
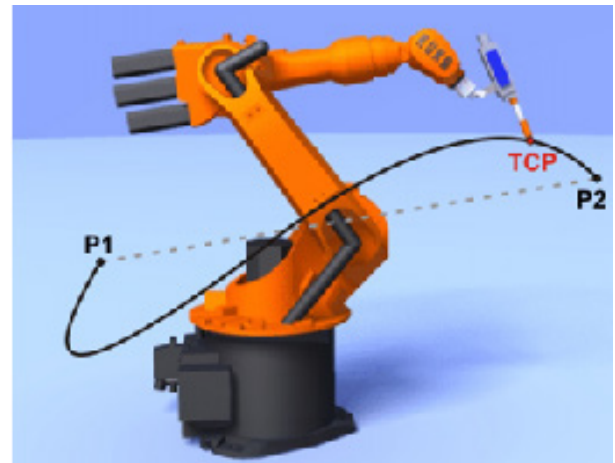
### Circular: CIRC

The robot guides the TCP at a defined velocity along a circular path to the end point. The circular path is defined by a start point, auxiliary point and end point.

### Spline

Spline is a motion type that is particularly suitable for complex, curved paths. Such paths can also be generated using approximated LIN and CIRC motions, but splines have advantages, however. The most versatile spline motion is the spline block. A spline block is used to group together several motions as an overall motion.

The spline block is planned and executed by the robot controller as a single motion block. The motions that may be included in a spline block are called spline segments. They are taught separately. A CP spline block can contain SPL, SLIN and SCIRC segments. A PTP spline block can contain SPTP segments.

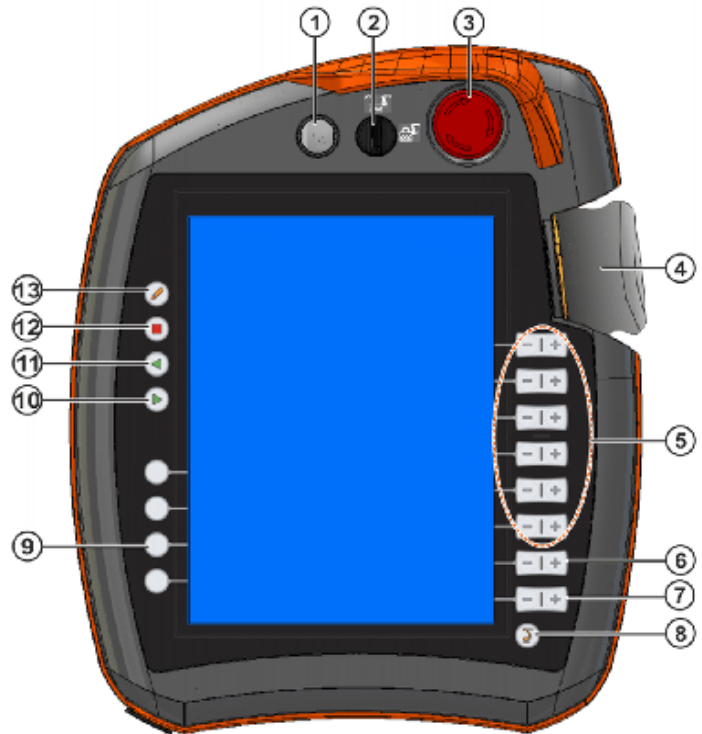


In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with the smartPAD or teach pendant various helpful display screens of a KUKA robot arm system.

The various smartPAD or teach pendant screens are important in the navigation and operation of the industrial KUKA robot arm. This diagram will allow you to familiarize yourself with the teach pendant and navigate the different buttons before operating the industrial KUKA robot arm.

## smartPAD Button Legend: Front

1. Disconnection Button
2. Keyswitch for calling the connection manager
3. Emergency Stop- press to lock and twist to unlock
4. Space Mouse- manual robot movement trigger
5. Jog Keys- manual robot movement of individual axes
6. Key for setting the program override
7. Key for setting the jog override
8. Main Menu Key- shows the menu items on the smartHMI
9. Status Keys- primarily for setting parameters in and dependent of technology packages
10. Start Key- used to start a program
11. Start Backwards Key- used to start a program backwards
12. Stop Key- used to stop a program that is running
13. Keyboard key- displays keyboard



## smartPAD Button Legend: Back

1. Enabling Switch- must be held in the center position in operating modes T1 and T2 in order to be able to jog the manipulator The enabling switch has 3 positions: Not pressed, Center position , Panic position
2. Start Key (Green)- used to start a program
3. Enabling Switch
4. USB Connection
5. Enabling Switch
6. Identification Plate

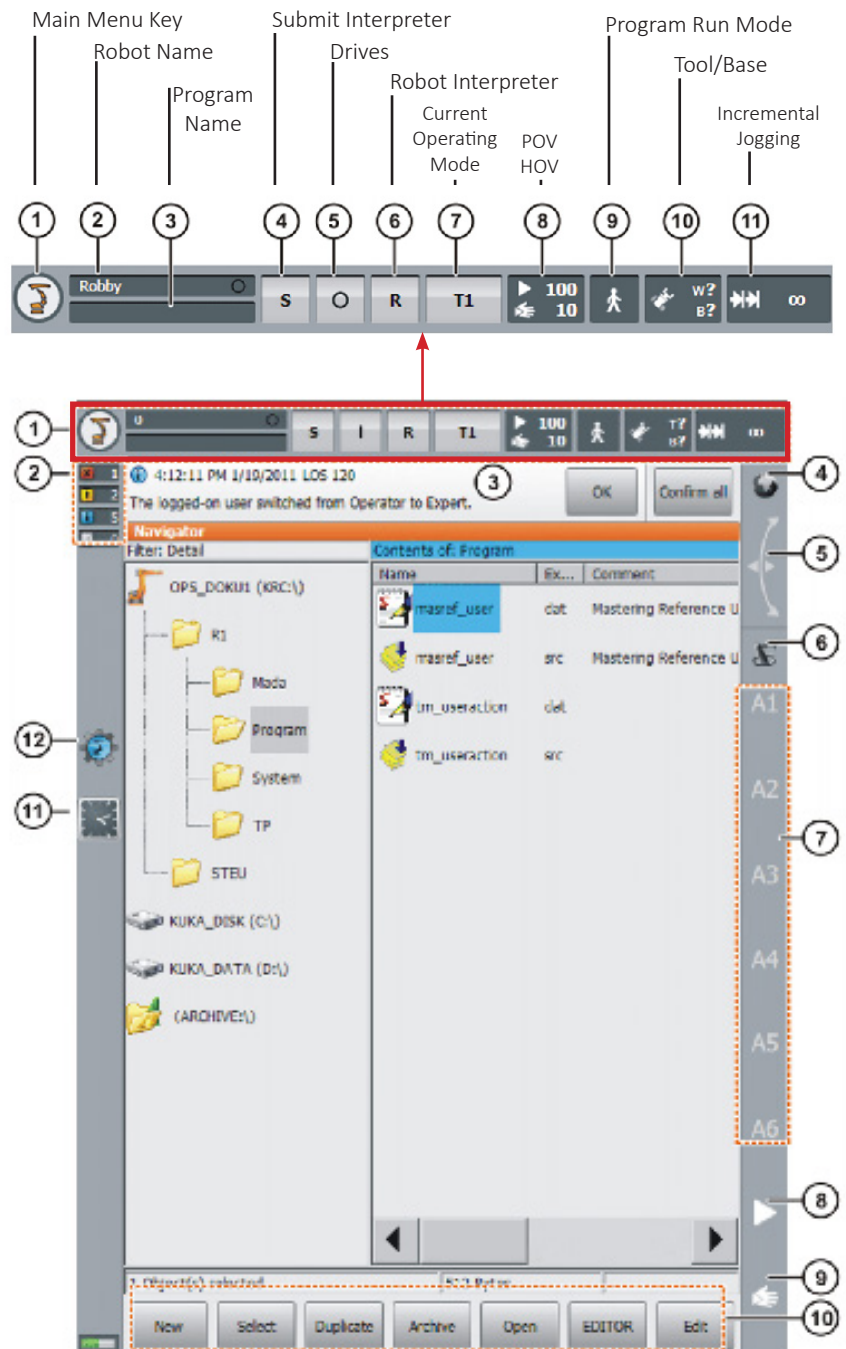


In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with the smartPAD or teach pendant various helpful display screens of a KUKA robot arm system.

The various smartPAD or teach pendant screens are important in the navigation and operation of the industrial KUKA robot arm. These screenshots will allow you to familiarize yourself with the teach pendant and navigate the different menus before operating the industrial KUKA robot arm.

## smartPAD User Interface Legend

1. Status Bar
2. Message Counter- indicates how many active messages there are and touching the button will enlarge the display
3. Message window- displays last message and touching will enlarge to display all active messages
4. Space Mouse Status Indicator- shows the current coordinate system for jogging with the Space Mouse and touching displays all coordinate systems allowing you to change systems
5. Space Mouse Alignment Indicator- touching opens a window in which the current alignment of the Space Mouse is indicated and can be changed
6. Jog Keys Status Indicator- shows current coordinate system for jogging with jog keys and touching displays all coordinate systems allowing you to change systems
7. Jog Key Labels- displays the current selected coordinate system or axis and touching causes the selected kinematics group to be displayed
8. Program Override
9. Jog Override
10. Button Bar- change dynamically and always refer to the window that is currently active in the smartHMI. Within this bar is the EDIT button which can be used to call numerous commands relating to the navigator
11. Clock
12. WorkVisual Icon- touching causes the active project and the OPEN button to be displayed. OPEN takes you to the Project Management window



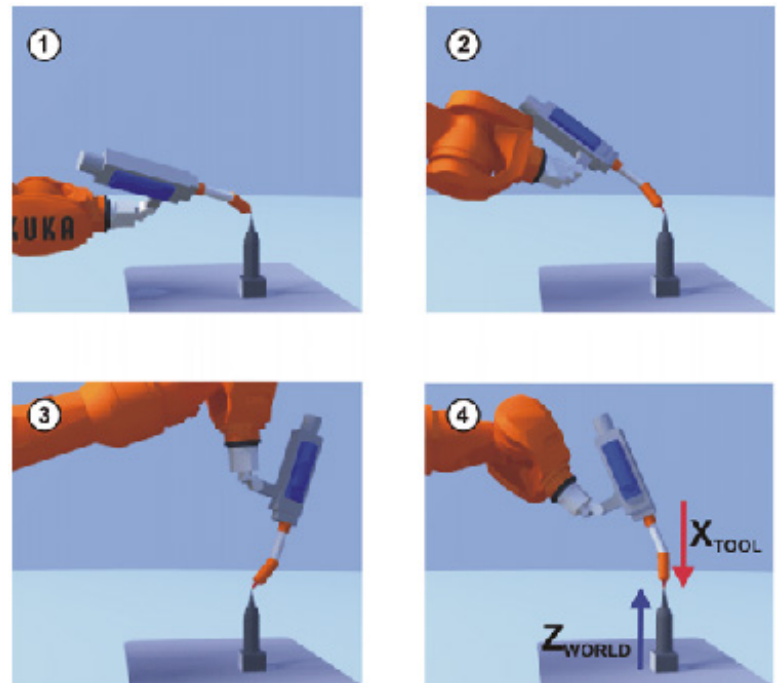


In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with how to teach a new tool for a new project with the KUKA robot arm system. **Teaching a tool to an industrial KUKA robot arm is how you set a accurate tool touch point (TCP) to generate tool**

**paths to follow an orientation so the robot arm knows where and how to move. In this assignment you will set a TCP. During this process it is ideal to turn the movement increments down to get an accurate as possible TCP. The tool to be calibrated is mounted on the mounting flange and should be operating in mode T1. For this particular method the four points need to be far apart as possible and avoid them being more than 180 degrees apart.**

## Teaching A Tool: XYZ 4-Point

1. In the main menu, select Start-up > Calibrate > Tool > XYZ 4-point.
2. Assign a number and a name for the tool to be calibrated. Confirm with Next.
3. Move the TCP to a reference point. Press Calibrate. Answer the request for confirmation with Yes.
4. Move the TCP to the reference point from a different direction. Press Calibrate. Answer the request for confirmation with Yes.
5. Repeat step 4 twice.
6. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.) (>>> 5.8.3 "Entering payload data" Page 134)
7. Confirm with Next.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press Meas. points. Then return to the previous view by pressing Back.
9. Either: press Save and then close the window via the Close icon.

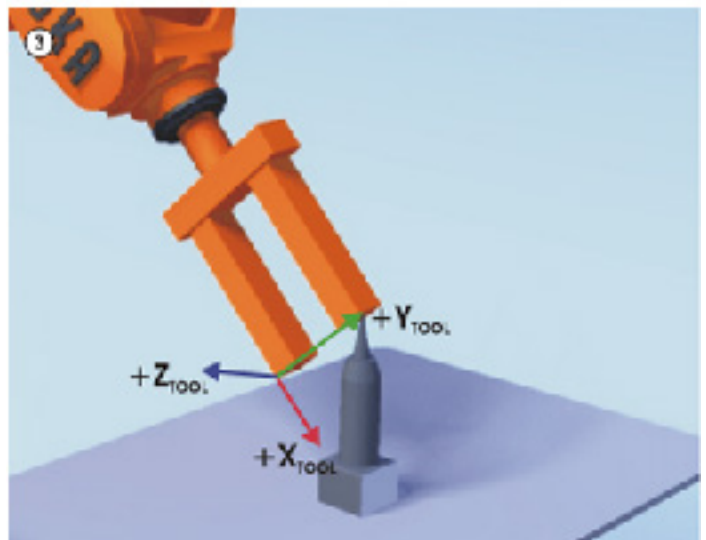
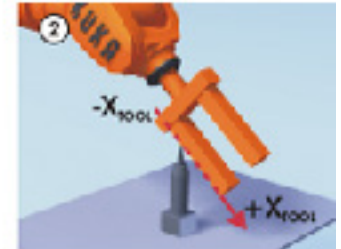


In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with how to teach a new tool for a new project with the KUKA robot arm system. Teaching a tool to an industrial KUKA robot arm is how you set an accurate Tool Control Point (TCP) to generate

tool paths to follow an orientation so the robot arm knows where and how to move. In this assignment you will set a TCP. During this process it is ideal to turn the movement increments down to get as accurate as possible TCP. Next this method is used if it is necessary to define the axis directions with particular precision, a gripper tool as an example. The TCP of the tool has already been measured and should be operating in the mode T1.

## Teaching A Tool: ABC 2-point

1. In the main menu, select Start-up > Calibrate > Tool > ABC 2-point.
2. Enter the number of the mounted tool. Confirm with Next.
3. Move the TCP to any reference point. Press Calibrate. Answer the request for confirmation with Yes.
4. Move the tool so that the reference point on the X axis has a negative X value (i.e. move against the tool direction). Press Calibrate. Answer the request for confirmation with Yes.
5. Move the tool so that the reference point in the XY plane has a negative Y value. Press Calibrate. Answer the request for confirmation with Yes.
6. Enter the payload data. (This step can be skipped if the payload data are entered separately instead.) (>>> 5.8.3 "Entering payload data" Page 134)
7. Confirm with Next.
8. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press Meas. points. Then return to the previous view by pressing Back.
9. Press Save.



In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with how to teach a new base for a new project with the KUKA robot arm system.

Teaching a base to an industrial KUKA robot arm is how you set a plane for your tool paths to follow an orientation so the robot arm knows where and how to move. Before this step you will need to make sure that you have your tool defined and your TCP set. During base calibration, the user assigns a Cartesian coordinate system (BASE coordinate system) to a work surface or the workpiece. The BASE coordinate system has its origin at a user-defined point.

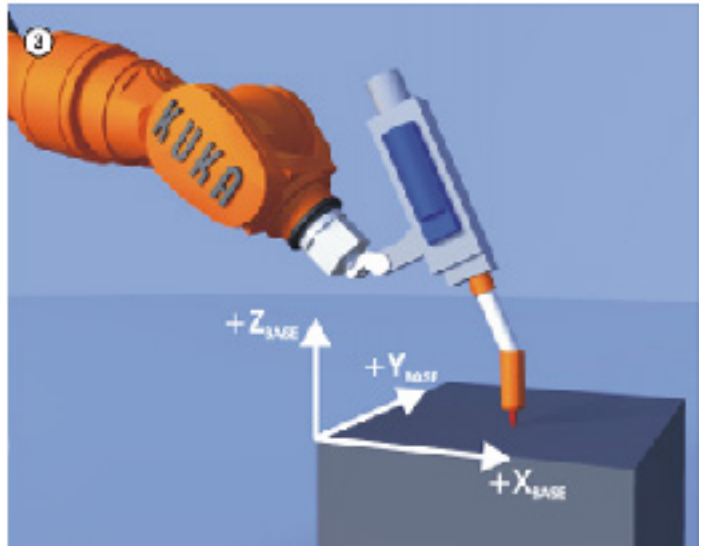
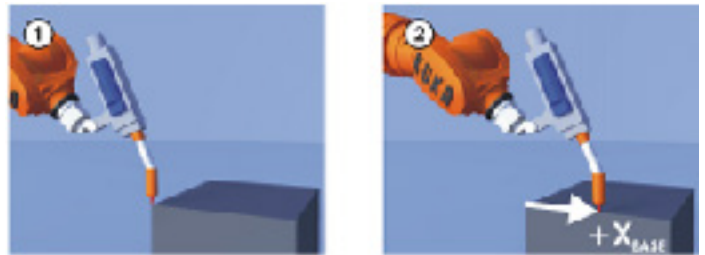
## Teaching A Base: 3- Point

### Advantages

The TCP can be jogged along the edges of the work surface or workpiece.

Points can be taught relative to the base. If it is necessary to offset the base, e.g. because the work surface has been offset, the points move with it and do not need to be retaught.

1. In the main menu, select Start-up > Calibrate > Base > ABC 3-point.
2. Assign a number and a name for the base. Confirm with Next.
3. Enter the number of the mounted tool. Confirm with Next.
4. Move the TCP to the origin of the new base. Press Calibrate. Answer the request for confirmation with Yes.
5. Move the TCP to a point on the positive X axis of the new base. Press Calibrate. Answer the request for confirmation with Yes.
6. Move the TCP to a point in the XY plane with a positive Y value. Press Calibrate. Answer the request for confirmation with Yes.
7. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press Meas. points. Then return to the previous view by pressing Back.
8. Press Save.



In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with how to teach a new base for a new project with the KUKA robot arm system.

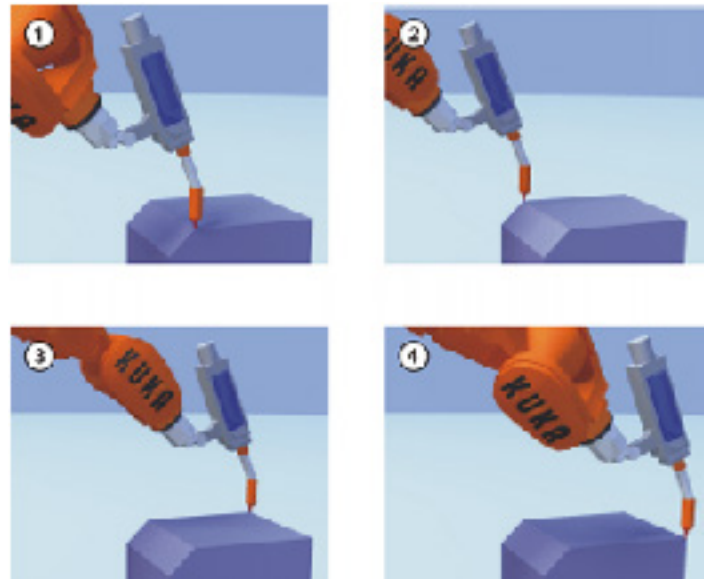
Teaching a base to an industrial KUKA robot arm is how you set a plane for your tool paths to follow an orientation so the robot arm knows where and how to move. Before this step you will need to make sure that you have your tool defined and your TCP set. During base calibration, the user assigns a Cartesian coordinate system (BASE coordinate system) to a work surface or the workpiece. The BASE coordinate system has its origin at a user-defined point.

## Teaching A Base: Indirect

The indirect method is used if it is not possible to move to the origin of the base, e.g. because it is inside a workpiece or outside the workspace of the robot.

The TCP is moved to 4 points in the base, the coordinates of which must be known. The robot controller calculates the base from these points.

1. In the main menu, select Start-up > Calibrate > Base > Indirect.
2. Assign a number and a name for the base. Confirm with Next.
3. Enter the number of the mounted tool. Confirm with Next.
4. Enter the coordinates of a known point in the new base and move the TCP to this point. Press Calibrate. Answer the request for confirmation with Yes.
5. Repeat step 4 three times.
6. If required, coordinates and orientation of the calibrated points can be displayed in increments and degrees (relative to the FLANGE coordinate system). For this, press Meas. points. Then return to the previous view by pressing Back.
7. Press Save.



In this assignment, you will review some basic knowledge of KUKA Robotics, including safety, work envelope, work cell, primary components, screen displays, axis-specific motion, coordinate systems, and tools. Now, we will continue with how to teach a new tool for a new project with the KUKA robot arm system.

Having access and operating this equipment and utilizing a shared space means that there needs to be a level of respect for the technology but also others that may be using the shared space. The space is relatively small and poses a lot of risks if cluttered along with a lose of efficiency of the space. Follow these simple post operation practices to continue to be allowed access to this equipment and space.

## Post Operation Practices

Before shutting down the machine at the end of a session:

1. Remove any tooling from the robot and store it properly.
2. Move arm in a manner that keeps the walkway clear to maintain CNC router material access.
3. Neatly wrap all cords.
4. Place the teach pendant in its holder.
5. Make sure any high voltage power is off at the breaker.
6. Clean up your personal items and remove your fixtures form the lab.
7. Return any borrowed equipment.
8. Make sure to sweep the area if necessary.
9. Make sure to fill out the housekeeping checklist.
10. Make sure to turn all the lights off.
11. Make sure that all doors are locked upon exiting.



The goal of this assignment is to develop your understanding of the KUKA Robot and its functionality by utilizing RoboDK. Through the use of RoboDK, you can work in parallel with the KUKA Robot. This workflow is approachable and will enhance your ability to navigate a maze path and program the robot effectively to do so. Additionally, you

will learn how to import 3D parts and curves from Rhino to RoboDK through the use of the RoboDK Rhino plug-in. This plug-in allows for a stream-lined workflow within a software you have familiarity with and can be built upon in the future within grasshopper using KUKA PRC. In this assignment, you will need **Rhino**, **RoboDK**, and **RoboDK plug-in**.

**Step 01:** Download and Install RoboDK. <https://robdk.com/download>.

**Step 02:** Download and Install RoboDK plug-in Rhino. <https://robdk.com/doc/en/PlugIn-Rhino.html>.

**Step 03:** Download two files: **"M6A4\_WORKCELL\_Maze-2023.stl"** and **"M6A4\_WORKCELL\_Maze-2023.3dm"** from **\*insert link\***

**Step 04:** Open the **"M6A4\_WORKCELL\_Maze-2023.3dm"** file you just downloaded in Rhino and check for the RoboDK toolbar. \*If you do not see RoboDK in your toolbar go to Main Menu> Tools> Toolbar Layout> Select RoboDK and then check the box below to add it. It will appear somewhere on your screen and simply need to drag it to your toolbar to add.\*

**Step 05:** Launch RoboDK and drag and drop the **"M6A4\_WORKCELL\_Maze-2023.stl"** file into the RoboDK work environment. If it does not work check in the Rhino RoboDK toolbar plug-in and click **"Settings"** then go to the **"Object Export Options"** and select **"STL"** in the drop down menu and click **"OK"**.

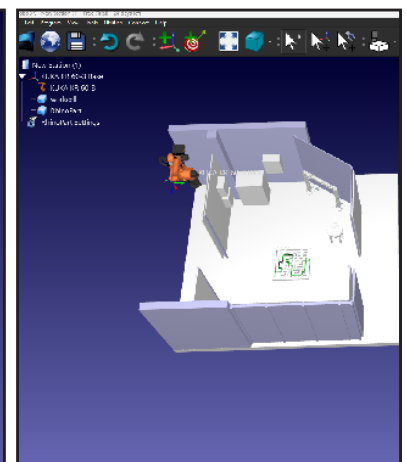
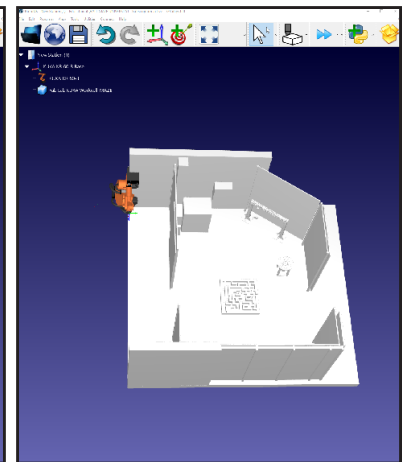
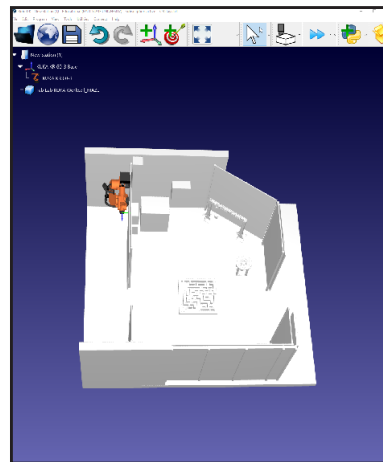
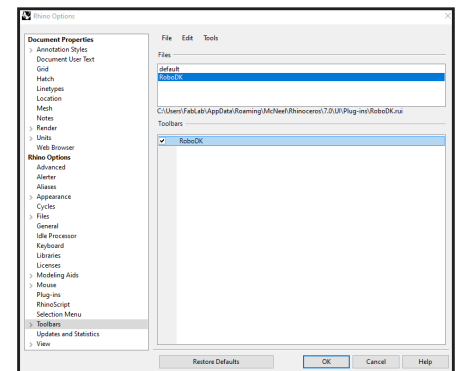
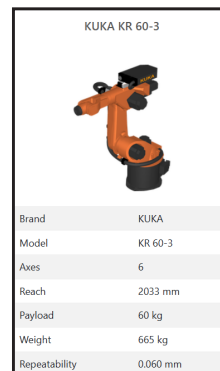
**Step 06:** Click on the **"Open Robot Library"** option and select **"KUKA KR 60-3."** Click **"Open"** to import the robot into RoboDK

**Step 07:** In RoboDK in the station tree click and drag the **"workcell"** into the **"KUKA KR 60-3 Base"** parent folder. Then delete the **"Reference Rhino"** reference Frame.

**Step 08:** Open the rhino file **"M6A4\_WORKCELL\_Maze-2023.3dm"** in the RoboDK plug-in **"Load Curve(s)"** and select the maze solution path and hit enter.

**Step 09:** In RoboDK click and drag the **"RhinoPart"** into the **"KUKA KR 60-3 Base"** and delete the **"Reference Rhino"**

**Step 10:** In RoboDK click **"Add New Reference Frame"** icon (Axis Gumball with a Plus Sign) in the toolbar and right click it in the station tree then select **"Rename"** as Robot Center.



The goal of this assignment is to develop your understanding of the KUKA Robot and its functionality by utilizing RoboDK. Through the use of RoboDK, you can work in parallel with the KUKA Robot. This workflow is approachable and will enhance your ability to navigate a maze path and program the robot effectively to do so. Additionally, you

will learn how to import 3D parts and curves from Rhino to RoboDK through the use of the RoboDK Rhino plug-in. This plug-in allows for a stream-lined workflow within a software you have familiarity with and can be built upon in the future within grasshopper using KUKA PRC. In this assignment, you will need **Rhino**, **RoboDK**, and **RoboDK plug-in**.

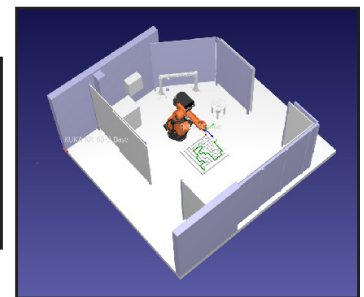
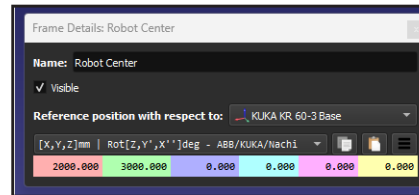
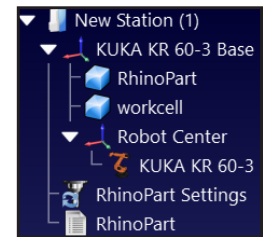
**Step 12:** In RoboDK click and drag the **“KUKA KR 60-3”** and add it to the **“Robot Center”** reference frame parent in the station tree.



**Step 13:** In RoboDK click and drag the **“Robot Center”** and add it to the **“KUKA KR 60-3”** reference frame parent in the station tree.

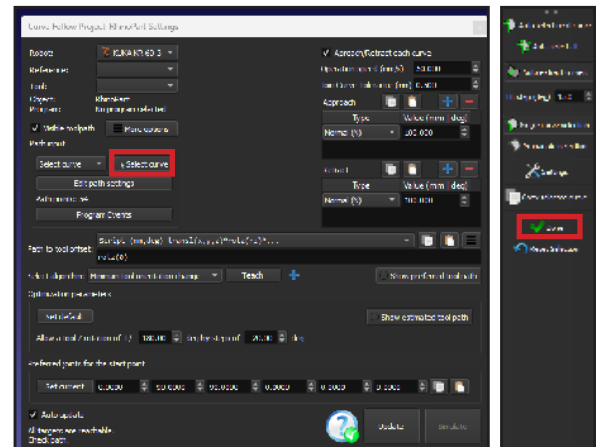


**Step 14:** In RoboDK double click the **“Robot Center”** reference frame in the station tree and change the red (first) to 2000 and green (second) to 3000 axis number.



**Step 15:** In RoboDK in the station tree right-click the **“RhinoPart Settings”** and click **“Options”** to assign the curve in RoboDK.

**Step 16:** Click **“Select Curve”** and choose the curve path uploaded from Rhino which should appear green. Once selected, click **“Done”** and then **“Update and Simulate.”** If done correctly you should now see the robot following the curve path.



**Step 17:** Use the simulation buttons and slider at the bottom of RoboDK to play the simulation.



The goal of this assignment is to develop your understanding of the KUKA Robot and its functionality by utilizing RoboDK. Through the use of RoboDK, you can work in parallel with the KUKA Robot. This workflow is approachable and will enhance your ability to recreate a path and program the robot effectively to do so. Additionally, you will learn

how to import 3D parts and curves from Rhino to RoboDK through the use of the RoboDK Rhino plug-in. This plug-in allows for a stream-lined workflow within a software you have familiarity with and can be built upon in the future within grasshopper using KUKA PRC. In this assignment, you will need **Rhino**, **RoboDK**, and **RoboDK plug-in**.

**Step 01:** Download and Install RoboDK. <https://robotk.com/download>.

**Step 02:** Download and Install RoboDK plug-in Rhino. <https://robotk.com/doc/en/Plugin-Rhino.html>.

**Step 03:** Download two files: **“M6A4\_WORKCELL\_Draw-2023.stl”**, **“M6A4\_WORKCELL\_Draw-2023.3dm”**, and **“Tool 4”** from *\*insert link\**

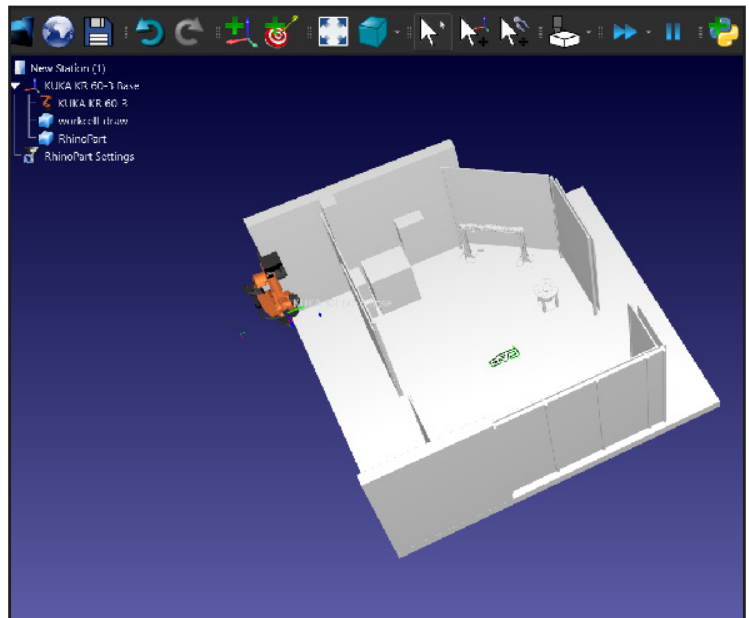
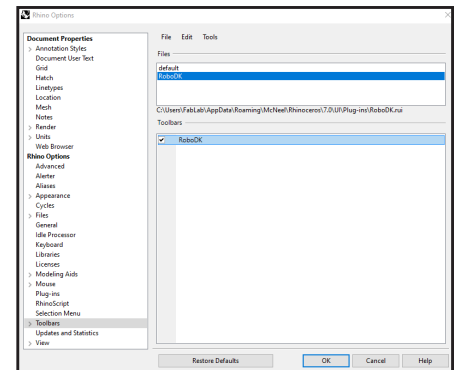
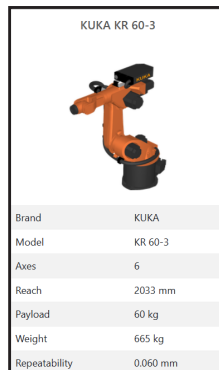
**Step 04:** Open the **“M6A4\_WORKCELL\_Draw-2023.3dm”** file you just downloaded in Rhino and check for the RoboDK toolbar. \*If you do not see RoboDK in your toolbar go to Main Menu> Tools> Toolbar Layout> Select RoboDK and then check the box below to add it. It will appear somewhere on your screen and simply need to drag it to your toolbar to add.\*

**Step 05:** Launch RoboDK and drag and drop the **“M6A4\_WORKCELL\_Draw-2023.stl”** file into the RoboDK work environment. If it does not work check in Rhino in the RoboDK toolbar plug-in and click **“Settings”** then go to the **“Object Export Options”** and select **“STL”** in the drop done menu and click **“OK”**.

**Step 06:** Click on the **“Open Robot Library”** option and select **“KUKA KR 60-3.”** Click **“Open”** to import the robot into RoboDK

**Step 07:** Open the rhino file **“M6A4\_WORKCELL\_Draw-2023.3dm”** in the RoboDK plug-in **“Load Curve(s)”** and select the CAP Logo path and hit enter.

**Step 08:** In RoboDK in the station tree click and drag the **“workcell-draw”** and **“RhinoPart”** into the **“KUKA KR 60-3 Base”** parent folder. Then delete the **“Reference Rhino”** reference Frame.





The goal of this assignment is to develop your understanding of the KUKA Robot and its functionality by utilizing RoboDK. Through the use of RoboDK, you can work in parallel with the KUKA Robot. This workflow is approachable and will enhance your ability to recreate a path and program the robot effectively to do so. Additionally, you will learn

how to import 3D parts and curves from Rhino to RoboDK through the use of the RoboDK Rhino plug-in. This plug-in allows for a stream-lined workflow within a software you have familiarity with and can be built upon in the future within grasshopper using KUKA PRC. In this assignment, you will need **Rhino**, **RoboDK**, and **RoboDK plug-in**.

**Step 09:** In RoboDK click “Add New Reference Frame” icon (Axis Gumball with a Plus Sign) in the toolbar and right click it in the station tree then select “Rename” as Robot Center.

**Step 10:** In RoboDK click and drag the “KUKA KR 60-3” and add it to the “Robot Center” reference frame parent in the station tree.

**Step 11:** In RoboDK click and drag the “Robot Center” and add it to the “KUKA KR 60-3” reference frame parent in the station tree.

**Step 12:** In RoboDK double click the “Robot Center” reference frame in the station tree and change the red (first) to 2000 and green (second) to 3000 axis number.

**Step 13:** Attach “Tool 4” to the KUKA robot. To do this, right-click on “KUKA KR 60-3,” select “Add Tool (TCP) from File,” and attach the STL file of the gripper and pen holder. Make any necessary adjustments using the TCP from the toolbar.

**Step 14:** In RoboDK in the station tree right-click the “RhinoPart Settings” and click “Options” to assign the curve in RoboDK.

**Step 15:** Click “Select Curve” and choose the curve path uploaded from Rhino which should appear green. Once selected, click “Done” and then “Update and Simulate.” If done correctly you should now see the robot following the curve path.

**Step 16:** Use the simulation buttons and slider at the bottom of RoboDK to play the simulation.

