

Robot Studio AUAS - Introduction reader

This is a work in progress, and might be a bit rough here and there.

Please send your feedback to: s.b.s.yap@hva.nl

Date of this version: 09-02-2021

Introduction

This reader is created to prepare people to work with the industrial robot arms at the Robotlab of the Amsterdam University of Applied Science.

At some points this reader refers to video tutorials that partly cover the same topic that is also described in text in this reader. Feel free to watch the video's and then skip the text at places where the video already covered the contents. The text is left for those who don't like to learn from videos or just need to see it at their own pace. Ideally you use both this text and the videos because they both contain unique information at some points.

What is an industrial robot

A mechanical manipulator used in the industry to perform certain tasks. Often repetitive tasks, or tasks that require great precision or strength.

degrees of freedom

An industrial robot is often characterised by its degrees of freedom (DOF).

Typically each axis in a robot creates an extra degree of freedom.

A 3D printer has 3 degrees of freedom by its 3 axes. In the below case, these 3 axes are also labeled according to the axis in a cartesian coordinate system: X, Y and Z because they can be linked to such a space.



LulzBot https://www.lulzbot.com/sites/default/files/TAZ_6_Angle_Main_Product_Page.png

But mechanical axes do not always link to axes in the cartesian coordinate system. We can understand that when looking at a delta printer.



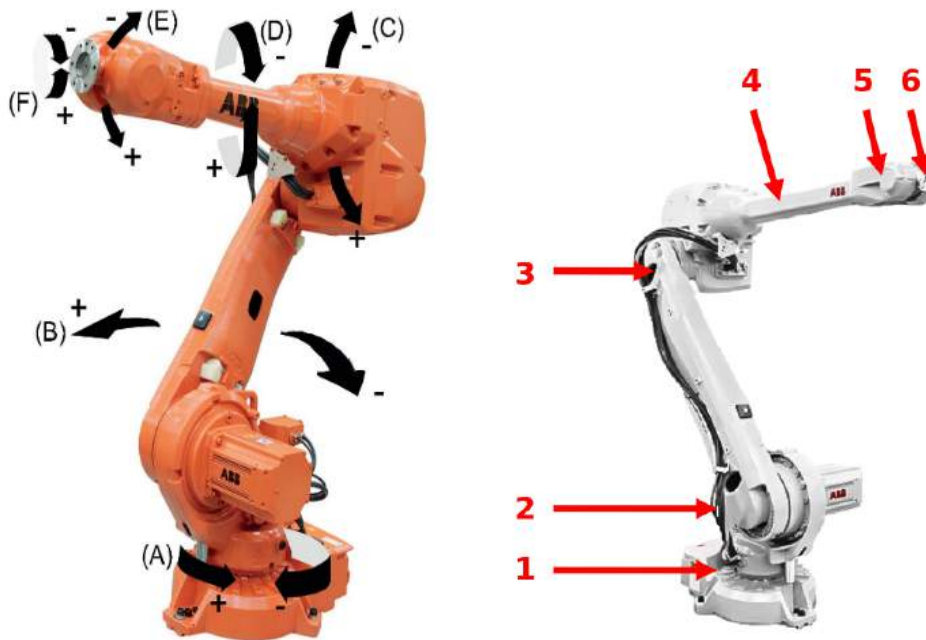
Anycubic Kossel Plus

https://cdn.shopify.com/s/files/1/0245/5519/2380/products/Kossel_Plus_1_1800x1800.jpg?v=1558420052

Industrial robot arms come in many forms, but most of them have 6 DOF.
All robots currently (2020) in the robotlab are 6 DOF.

6 degrees of freedom means the arm can position itself anywhere in its workfield (X,Y,Z) and orientate itself in any direction (3 rotation directions).

These are the axis in such a robot:



Industrial Electronics - Jointed Spherical (Articulated Arm) Robots

<https://www.industrial-electronics.com/industrial-electricity-com/images/14-10.jpg>

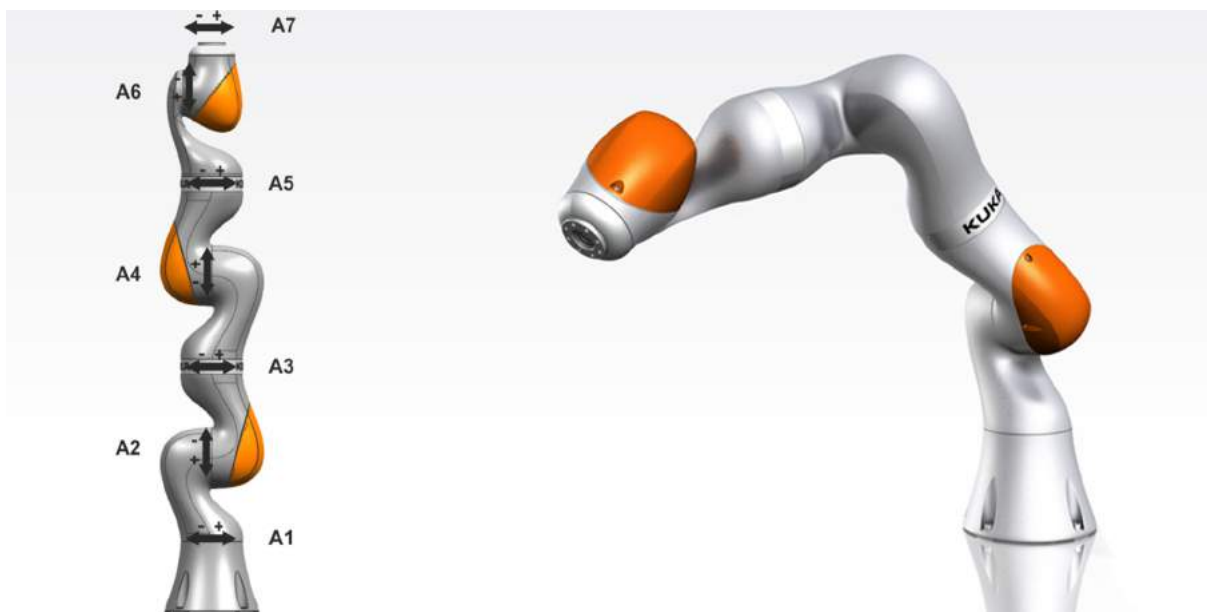
(Be aware that in the above image the arrows for axis 3 are placed slightly confusing after axis 4. Axis 3 is hinged before axis 4.)

An example of industrial robots with a different number of DOF is a SCARA robot:



Fanuc SCARA robots <https://www.fanucamerica.com/images/default-source/scara/sr-6-500x400.png>

Another example is this Kuka iiwa robot.



LBR iiwa – Cobot @unibs.it <https://cobot.unibs.it/dev/wp-content/gallery/iiwa/LBRiiwa.png>

Sometimes industrial robots are complemented with extra external axes.

A workpiece positioner, adding multiple axes, often 2:



Retrade - ABB Robot IRB 2400 M2004

https://www.retrade.eu/i/detail/2088731-IMG_20160913_115110.jpg/ABB_Robot_IRB_2400_M2004.jpg

A track, adding 1 axis:



RAP Technologies - Showcase: Bruil Prefab Printing

https://images.squarespace-cdn.com/content/v1/5d53e7476636c10001f86e1f/1578060787635-XU21LTPGICH6FVTSKZ/ke17ZwdGBTodd18pDm48kKwMFAex-CvLWy4CH9W6Kzt7gQa3H78H3Y0txjaiv_ofDoOvxcdMmMKkDsyUqMSsMWxHk725yiiHCCLfrh8O1z4YTzHvnKhyp6Da-NYroOW3ZGjoBKy3azqku80C789I0ooWhOa5cxQSJsU3rXf8luWjlsKQ6oIf404bDr2RtSPxELRDJld-7Ti-Cj-lgLLDhw/bouwwereld-Bruil-Prefab-Printing.jpg

A gantry, adding multiple axes:



Architecture and Digital Fabrication, ETH Zurich - https://www.dfab.arch.ethz.ch/web/images/content/GKR_Infrastructure_6.jpg

Applications

Typical applications:

- Arc Welding
- Assembly
- Material Handling
- Machine Tending
- Material Removal
- Cleaning/Spraying
- Dispensing
- Packing
- Laser Cutting
- Laser Welding
- etc.

Markets 2019:

28% automotive industry

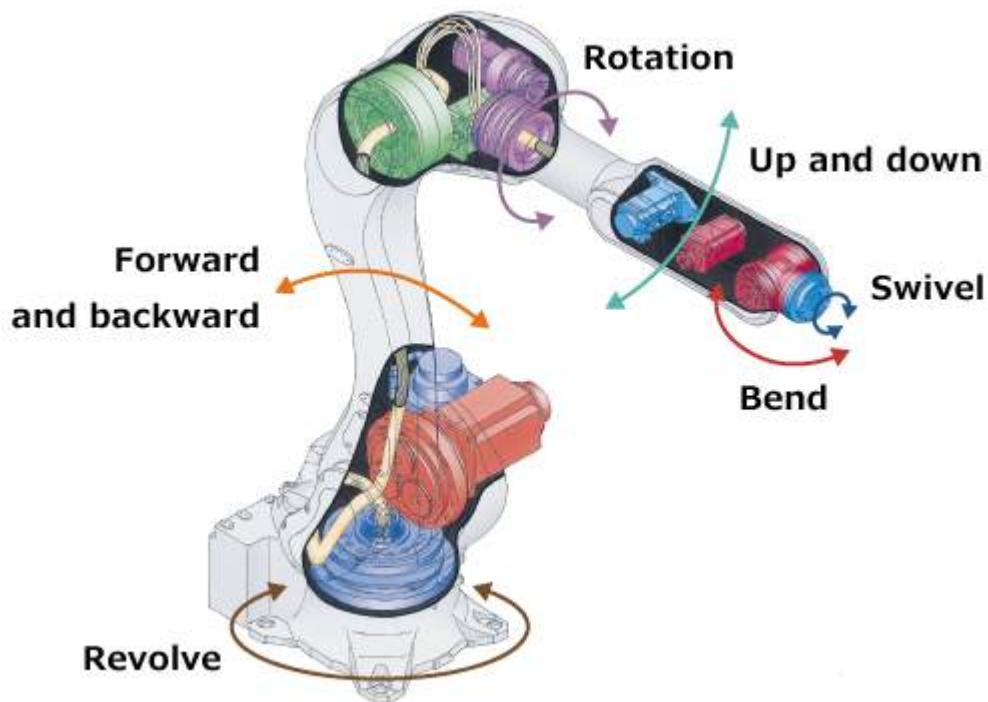
24% electrical/electronics industry

12% metal and machinery industry
5% plastics and chemical industry
3% food and beverages industry

Executive Summary World Robotics 2020 Industrial Robots -
https://ifr.org/img/worldrobotics/Executive_Summary_WR_2020_Industrial_Robots_1.pdf

How does it work

Axes are moved by electro motors often through gearboxes, belts, etc.
Each axis also has a brake, to keep the robot in position when the power is turned off. If these brakes wouldn't be there, the robot would collapse under it's own weight.



How Are Industrial Robots Built? A Guide on the Components and the Movement of Robot Arms -
https://robotics.kawasaki.com/ja1/xyz/en/1804-03/img/1804-03_ph07.png

Only if you are interested to know more, you could check out:
<https://robotics.kawasaki.com/ja1/xyz/en/1804-03/> and possibly
https://www.youtube.com/watch?v=6YiPrytt_Ss for more bonus material.

End-effector

When you buy an industrial robot arm, you cannot do much with it. At the end of it there will be a flange with some threaded holes, quite useless as is. You need to add a tool or something. This is what we call the end-effector.

Controller

The controller houses all the control electronics. It is the brain of the robot. It has a computer that does all the calculations, it has the safety circuits, it has the motor drivers.

This is what the controller for an ABB robot looks like from the outside:



ABB IRC5 Controller - https://cdn.productimages.abb.com/9IBA012392_720x540.jpg

And the inside:



Teach Pendant

The teach pendant is the interface to the user. It is used to control the robot and you can also use it to program.

Every robot manufacturer has a different model and name for it:

Kuka: SmartPad



https://www.kuka.com/-/media/kuka-downloads/imported/9cb8e311bfd744b4b0eab25ca883f6d3/kuka_pb_controllers_en.pdf?rev=d85c3d95195c4fbbbaddc024b1d31d92

ABB: FlexPendant



https://library.e.abb.com/public/d6f68ade0cb24d6aa0fe79220321d187/IRC5_ROB0295EN-Rev.D.pdf

FANUC: iPendant touch



<https://www.fanuc.eu/~-/media/corporate/products/robots/accessories/400x400/400x400/ipendant-touch-compact-view-banner.jpg?w=264>

Specifications

Things that are good to know about the specifications of an industrial robot arm. Examples in this part are all taken from the ABB IRB 4600.

Reach

The reach of an industrial robot arm is often given in an image like this:

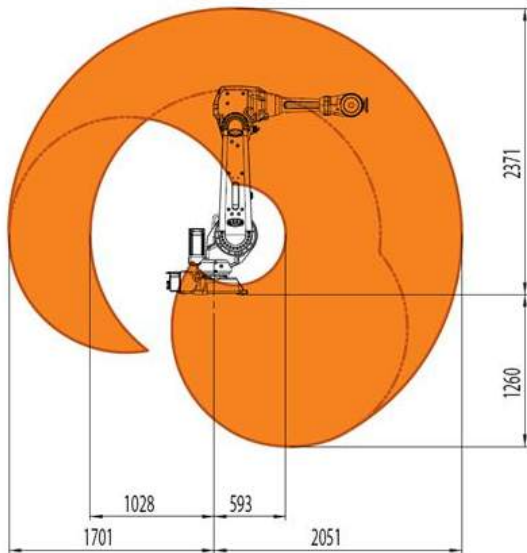
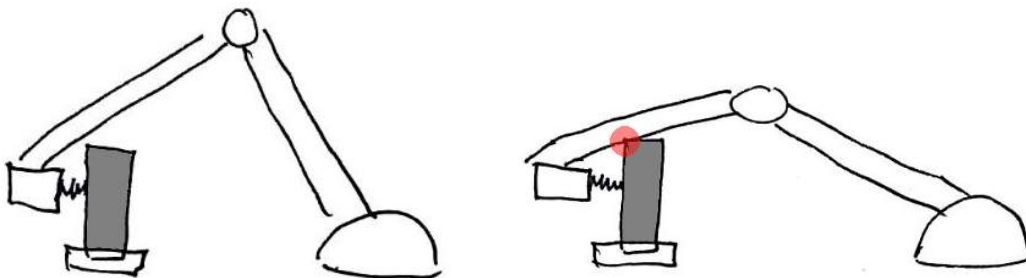


ABB 4600 documentation

You can imagine that this 2D diagram is the cross section of a 3D shape. When the first axis turns, the whole diagram is rotated around that axis.

In this example you see that the forward reach goes from 593 to 2051mm. That does not necessarily mean that you can work on objects of $2051 - 593 = 1458$ mm.

As you can imagine there are a lot more things to consider. For example the maximum position at 2051mm does not allow for rotation, it can only be that far when the wrist is straight. Looking at the drawings below will give you more hints on things to consider.



From these drawings it also becomes clear why adding axes with a positioner (see [degrees of freedom](#)) makes sense sometimes.

Accuracy

In the technical specs of an ABB industrial robot we can find the following:

Performance (according to ISO 9283)

Robot version	Position repeatability RP (mm)	Path repeatability RT (mm)*
IRB 4600 60/2.05	0.06	0.46

0.06 mm is very accurate one could say but we must be careful with what we mean by accuracy.

Although the repeatability is very precise, it could still be that if we move the robot from point A (0,0,0) to point B (1000,0,0) that the distance between these points is not 1000 +/- 0.12mm. They are only specifying repeatability but not absolute accuracy.

The absolute accuracy is affected by:

- Mechanical tolerances in the robotstructure
- Deflection due to load

The ABB robots that we have acquired so far (okt 2020) all have 'Absolute Accuracy', an option at the time of purchase. This means that all absolute position errors are minimised by a calibration process in the factory (for the IRB 4600 absolute accuracy is <1mm in the front of the robot with this option, while an uncalibrated robot can have up to 8mm deviation).

In the calibration process they move the robot in many positions, with some load on the end-effector, with very precise trackers they measure the position at each point. Then based on this data they upload calibration parameters to the robot so it will always compensate.

If you watch a short bit of this video you will understand the principle:

<https://www.youtube.com/watch?v=d3fCkS5xFlq>

(it's not the actual process at the factory).

For the robot to reach the accuracy you always need to define the exact weight and center of gravity of the load on the end-effector.

In some applications you don't need absolute accuracy. If you are implementing a robot that only needs to place seats in a car for years and years, then you have a very limited set of positions that the robot needs to be. If the uncalibrated absolute accuracy of the robot arm is not good enough, you can easily tweak the positions individually to the real world and you

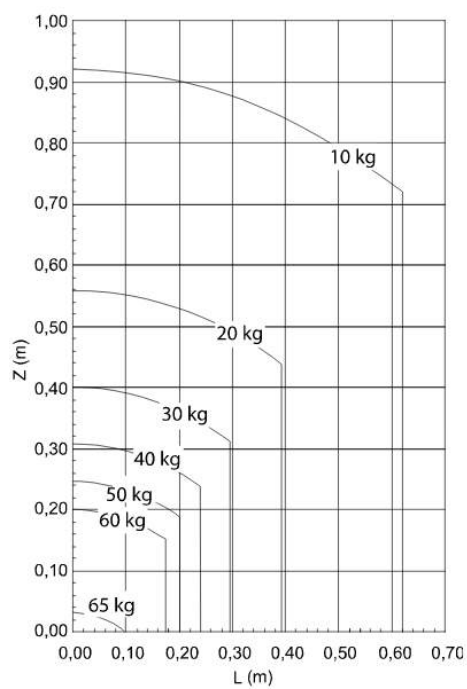
don't care what numbers are linked to a point x,y,z.

Payload

Each robot has a specified payload. The amount of weight that the robot can carry. Behind this number there is more detail.

Specifications regarding: maximum moment of inertia, maximum torque, the position of the load in the work area, and the orientation of the wrist.

Below an example for the specs regarding the position of the load in the work area:



Safety

Industrial robots are potentially very dangerous. They are typically very heavy, powerful and fast, and they don't come with much sensors built in.

Researchers from the US [National Institute for Occupational Safety and Health](https://en.wikipedia.org/wiki/National_Institute_for_Occupational_Safety_and_Health) (NIOSH) identified 61 robot-related deaths between 1992 and 2015 according to wikipedia.

https://en.wikipedia.org/wiki/Workplace_robotics_safety

So this is probably only counting in the USA.

Risks

There are probably more ways to get injured by a robot but roughly we can identify three possible scenarios:

- You get hit by a robot
- You are slowly being crushed by a robot
- You get trapped under a collapsed robot

To get an idea of how lethal a robot can be:

-Robot maximum speeds are specified in degree/s for each axis

-If we take the ABB IRB2600, the first axis has a maximum speed of 175 °/s.

-The reach of the arm (fully extended horizontally) is 1653mm

-So if the first axis would only move and the rest of the robot is fully extended, the distance it travels for 360° would be $2\pi * 1653 = 10386\text{mm}$

-it does 175° in 1 second, so that is $175/360 * 10386\text{mm} = +/- 5\text{m/s}$

-the robot weighs 272kg in total (not all of it is moving all the time of course)

A mechanical unit moving at 5 meter per second is not very safe to be around.

Collision detection

An industrial robot probably has some form of collision detection. The current drawn by the motor controllers is measured, and if that current draw shows certain peaks -because the robot hit something and cannot move any further- the system will halt.

And while you can tweak the sensitivity of this system, it will not protect you from injuries when you would be hit by a robot. Simply because it will only trigger when the impact has already happened.

Brakes

As mentioned earlier, the robot is moved by servo motors, but what happens when the power would be lost for some reason? The robot would collapse under its own weight. Therefore the robot has a brake on each axis that can only be released by supplying energy.

This brake is also engaged when the robot is not moving, to reduce the load on the servo's and controllers, when you use the robot you can hear this when you press the 3 position button.

All the ABB robots that we currently have in the lab have buttons to release the brakes. Never use the buttons. Just to be sure, repeating here:

never press these buttons!



These buttons are there to continue operation in certain conditions after collisions. Also they could be helpful when someone would be stuck. But wrongly using these buttons could result in great damage (the robot collapsing) and making the situation worse. So always ask a staff member to assist you when you would think you need these buttons. In general you need 2 or more people to do such an operation.

In addition to the problems with using these buttons wrongly outlined above, using these buttons also result in calibration problems.

Technical safety measures

To reduce the risks of (lethal) injuries multiple technical safety mechanisms are used.

Robot cells

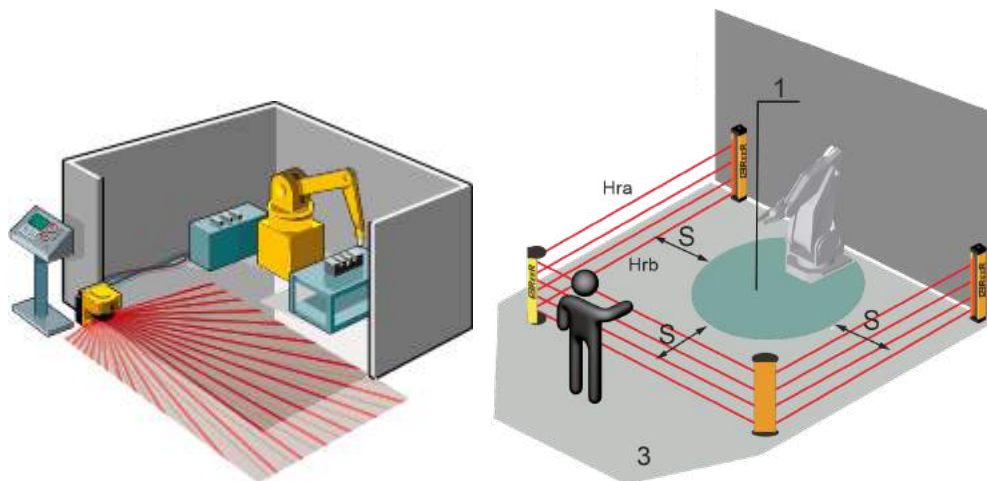
One of the most basic methods to prevent accidents is closing the robot area off. Safety fences are placed around the robot to keep people out. Before starting the robot, the operator will verify that no human being is in the cell. The door is electronically secured and when forced open the robot stops its operation.



Packaging Strategies -

<https://www.packagingstrategies.com/ext/resources/2018-Postings/Supplier-Products/Brenton-for-web.png?1540408434>

Instead of using fences and doors, you can also use more open setups, this requires a lot of expensive sensors. Also you typically need more space since people can run or fall into the area at high speed, if the distance between the sensors and the robot would be too small the robot would have no time to actually stop.



<https://static.thefabricator.com/a/articles/photos/946/fig3.gif>

<https://www.reersafety.com/images/reer/sicurezza/guida/it/4-raggi.png>

Cobots

Cobots are specifically designed to work side by side with humans according to the Power and Force Limited, PFL principle. This is often made possible because they are much

smaller and lighter to start with, and they are built with mechanical compliance or force-feedback detection to stop the robot when it hits a person or object.



The Universal Robots blog Welcome to the Age of the Cobots - <https://blog.universal-robots.com/hubfs/HQ%20Images/Blog/2019/Week%206/welcome%20to%20the%20age%20of%20cobots.jpg>

Reduced speed modes (manual mode vs automode)

To be able to program robots while watching closely, typically a robot has a reduced speed mode. For ABB robots we distinguish between Manual mode and Automode.

In Automode the robot can work on it's full speed without any human action needed.

In manual mode the robot has a highly reduced speed, and the operator needs to constantly press a button while operating, to let the system know that the operator is still there and healthy. This button is a 3 position button. When you let go, the robot stops. But when you press too hard the robot also stops. This is for situations where you are trapped in some way that your muscle tension only increases instead of loosens.

3 Position button being pressed:



Safe use

Working with an industrial robot arm in the lab environment will in most cases mean that you are working in reduced speed mode (called 'manual mode' for ABB robots). This mode is not a safe mode. You can still get hurt.

To stay safe during manual mode, your common sense will bring you a long way. But here are some guidelines for your common sense:

- Before operating the robot you make sure you know all the ways to stop the robot (explained later in this document)
- Before starting any movement of the robot, you make sure no one is close to the robot
- This includes areas where you don't expect the robot to move towards, but are nevertheless reachable by the robot
- You stop the robot as soon as someone comes too close (by letting go of the 3 position button)
- Only when you need to watch something closely (very precise positioning) you as the operator can move closer.
- As the operator you are responsible for the safety of others in the room

Auto mode

If for some reason you think you really need automove, you have to talk to someone of the staff first.

Using an ABB robot on IRC5 controller

First steps

Locating the emergency stop buttons

Before starting any use of the robot. Familiar yourself with the positions of the emergency stop buttons. There is one on the flexpendant, and there is one on the controller.



IRC5 controller



IRC5 compact controller



FlexPendant

Then also realise that stopping the robot in manual mode (not in auto mode) can be done by letting go of the 3 position button.



Turning the robot on and off

Turning on:

For the full size IRC5, turn the top left button to 'on'.

For the small size IRC5 (comes with the small robot IRB1200), turn the bottom left button to 'on'.



IRC5 controller



IRC5 compact controller

Turning off:

There is no software shutdown, you can simply turn the same big button back to 0.

Be aware that after turning the robot off, you shouldn't turn it back on straight away (in a reboot situation, or when you suddenly realise you forgot something), wait one minute before turning the robot back on.

Fans

The controller has 2 sets of fans.

One set for the computer, and one set for the motor controllers.

The fans for the motor controller are very loud. For this reason we have added a switch to these fans to turn them off when the robots are not doing heavy duty work. Please turn them on when the robot is doing heavy work for prolonged times.

The switch is a bit hidden, you have to reach in with your hand.



3 position button

Before making any movement with the robot, you will need to press the 3 position button.

The 3 positions are:

1. Not pressed (robot stopped)
2. Pressed halfway (robot enabled)
3. Pressed all the way (robot stopped)

Jogging

Jogging the robot is done often in an experimental lab environment.

When making mistakes while jogging the robot can cause serious damage to expensive equipment. So be carefull.

This video shows jogging, please watch it first before you continue, because part of what is written below is explained in this video as well.

<https://www.youtube.com/watch?v=1xY-NbJ3coE>

There are 2 ways of jogging:


- Axis by axis -> you can move each axis individually
- Linear/reorientate -> you can move in a cartesian coordinate system (x,y,z) and change the orientation at any point, all axes will rotate to accommodate.

When jogging, always face the robot (the back of the robot is where the cables are coming out). When you don't have any funny settings, then facing the robot will result in the most intuitive response of the robot.


Non funny settings are shown below. For some robots the 'World' is giving identical results. But just be sure you can always set it to 'Base'.

Current selection: Base


Select coordinate system.




World



Base





Tool



Work Object

OK
Cancel

 **Jogging**



Jogging is available all the time with the button next to the touch screen. However there is a specific jogging application/screen that will greatly help you to do the jogging. Please use this at all times in the beginning. (you can switch easily back and forth between the jogging screen).

Some of the values in the jogging screen are explained below. Others we will touch upon somewhere else down the line because they are only needed for more advanced operations.

Mechanical unit

Only relevant for robots on tracks, or multiple robots controlled by one flexpendant. Determines what you want to control, for example on the track with both robots coordinated, you can switch between: robot left, robot right, track left, track right.

If the track is not in coordinated mode but the two robots are separated, the mechanical unit switching is: robot, track.

Motion mode

Here you can see what motion mode you are in, this is quite important to keep an eye on, if you lose track of what mode you are in, you might operate the joystick with unexpected results.

Tool

Often you don't care about the tool, you just need the robot to go somewhere else.

But sometimes it is important to have the right tool selected:

- when you are using the position or rotation numbers as shown in the jogging screen
- when you are reorienting the tool and you need the center of rotation to be correct (it rotates around the tool center point, TCP)

Increment

Often increment is set to 'none' this means the robot moves according to the position of the joystick.

If you press the joystick a little bit, it will move slow. (you can go very slow)

If you press the joystick all the way, it will move at max jogging speed.

In between, the joystick position controls the speed proportional.

If we set the increment to any other value than 'none', the robot will move in steps. Small, medium or large steps. Then the joystick will not behave proportional but will only move one step at a time. If you hold the joystick in a position, it will repeat that step with a small time delay in between steps.

So when do we need increment?

When we are jogging an end-effector very close to an object. If we would have increment none, we can easily slip our joystick and crash into the object. But when we set increment to for example small, then we are somewhat protected from slips.

Understanding the concepts

Coordinate Systems

There are many coordinate systems that ABB uses to make our life easier when programming positions and when talking about them.

[extra video would be nice here]

World and base

In a robot system there is for example a world coordinate system and a base coordinate system.

Let's start with a simple situation. The big robot in the small lab (the IRB4600) has a base coordinate system and a world coordinate system that are exactly on the same location and orientation so we don't need to distinguish between them.

A robot has a front and a back one could say. The back is where the cables are. The front is where the robot is pointing if all axes are on 0°

To operate the robot, and when pondering about coordinate systems, always face the robot toward its front.

Facing the robot, the X axis of the base coordinate system is pointing towards you, and Z-up.

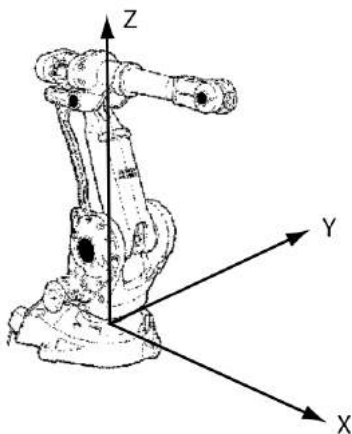


ABB image

So let's consider the track with the IRB2600 robots, here you will understand why the world and the base coordinate system are not the same always. Look at this image:



Both robots share the same world coordinate system.

The base coordinate system can move inside the world coordinate system.

Track

There is a video about jogging with the track, you might want to check that out before you continue reading because you can then skip some of the text.

<https://www.youtube.com/watch?v=edw8ZZrRwS4>

While we are at the track, let's quickly talk about the track axis. Not really important for some of you, so if you are not working with the track you can forget this quick, but just remember that there was something special with it, and you need to reread this when working with the track:

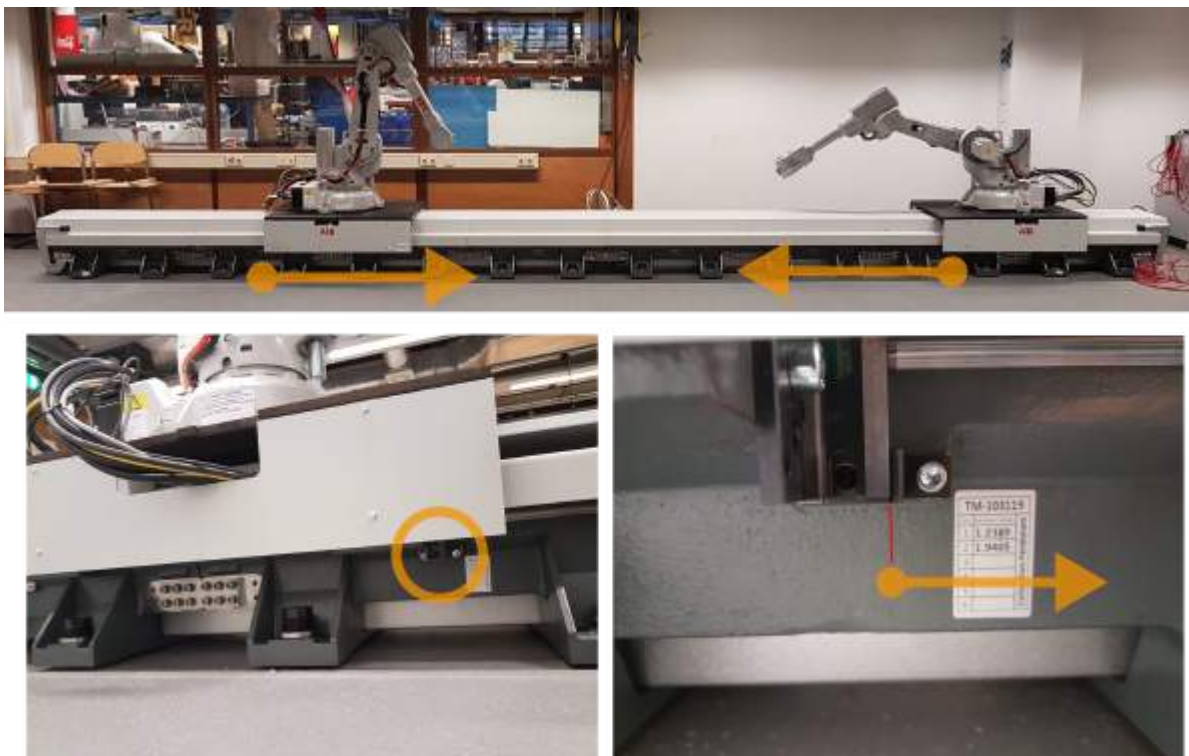
Two notes:

1. The 7th axis is the carriage on the track. This is no coordinate system, just a direction to be aware of.

Both carriage have positive travel towards the middle of the track.

2. Travel is given in mm from the zero point.

The zero point is not at the end of the track. So the carriage can travel towards negative values.



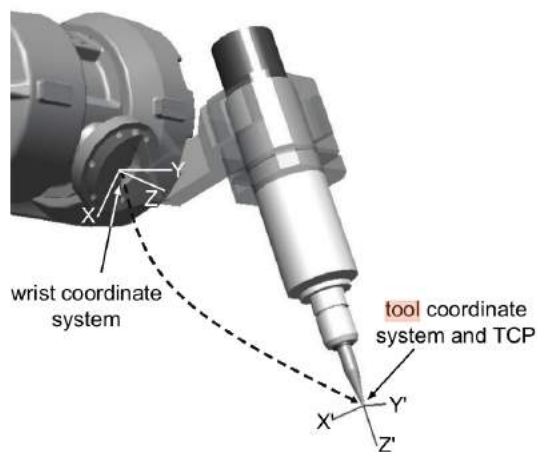
Tool

Instead of 'tool' we could also write 'end-effector' here, but ABB calls it 'tool'.

The minimum that we need to define is the location and orientation of the TCP (tool center point). Ideally we also define the center of the mass, this way the robot can handle the tool and all the forces resulting from moving the tool better.

These points and orientations are defined in relation to an origin on the flange, called wrist coordinate system by ABB.

For example when defining the TCP, first we translate from this origin, and then we rotate from the orientation that the wrist coordinate system has.



Target / point / position

A location and orientation where the robot should move to.

There are stop points and fly-by points.

Stop points are points where the robot moves exactly to the desired point (stand-still) and then executes the next command.

Fly-by points are points where the robot only needs to fly-by through a defined zone from the point. Using these points makes the robot movement smoother and faster.

You can define points by defining all the desired angles for each joint (MoveJ instruction).

You can define points by defining the location in the world coordinate system (x,y,z) and the

rotation in quaternions (more on quaternions later).

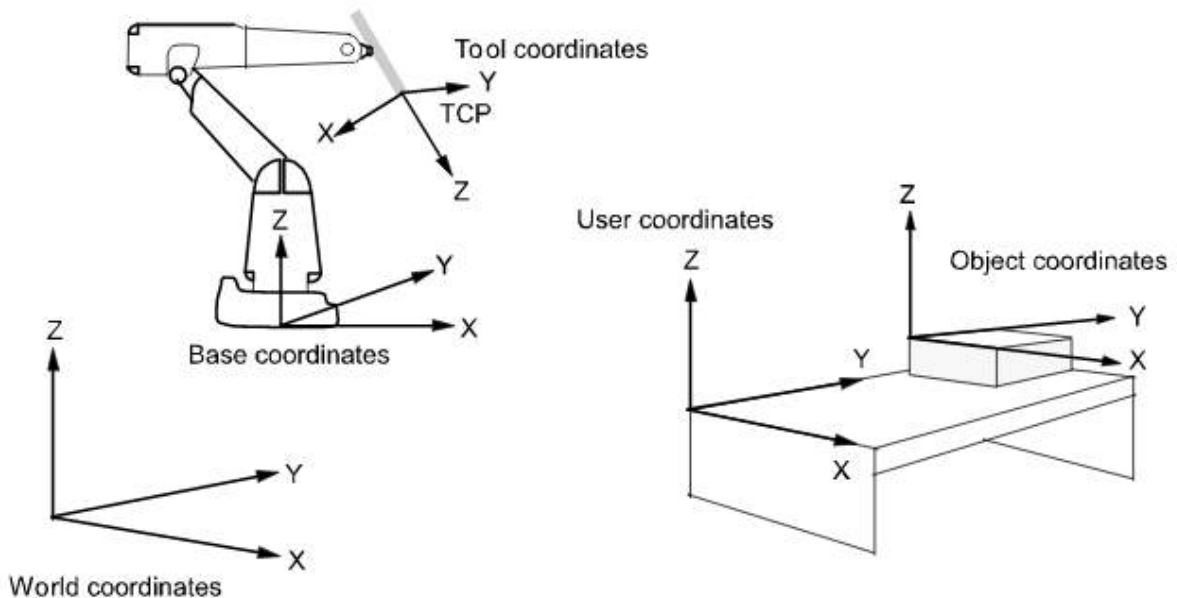


Work object

Work objects are not required. Basically it is a coordinate system that you can add yourself. And you can add as many as you like. And you can put them anywhere in any orientation.

Imagine you want to write a text on an A4 paper with the robot. If you define all the points in the world coordinate system that will work. But as soon as you want to do it again on a second paper, you need to rewrite all the points. If you would have written the points in a work object instead (a workobject coinciding with the A4 paper would make sense), you can just change the definition of the work object and use exactly the same code.

The object coordinate system is defined in the user coordinate system.

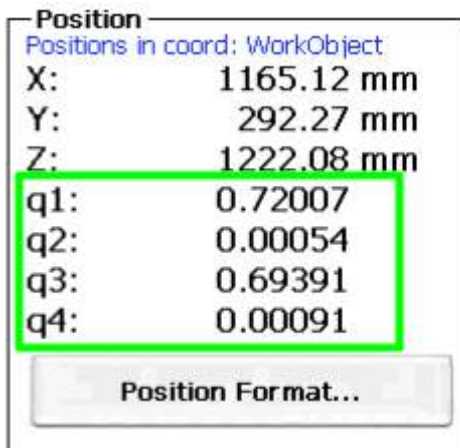


Quaternions

Quaternions are a way to describe 3D rotations. You don't need to know the math behind it. But if you think it is interesting, there are many resources, for example this is a nice introduction: <https://www.youtube.com/watch?v=zjMulxRvygQ>

Quaternions are used in robotics to describe an orientation. It describes an orientation by giving the rotation from a certain coordinate system. It is hard to read for humans, but it is a very powerful notation for the controller, so we have to deal with it.

For now it is good to know that when you see this:



The 4 numbers from q1 to q4 are together representing an orientation.

Starting programs

There is a video about running programs, you might want to check that out before you continue reading because you can then skip some of the text.

<https://www.youtube.com/watch?v=eFI7R45ROq4>

First you open the production window from the menu:

Manual
HvA_2600-112003_left

Guard Stop
Stopped (2 of 2) (Speed 100%)

- HotEdit
- Inputs and Outputs
- Jogging
- Program Editor
- Program Data
- Production Window
- Log Off Default User
- Backup and Restore
- Calibration
- Control Panel
- FlexPendant Explorer
- Lock Screen
- System Info
- Event Log
- Restart

TRACK_1
1/3

This is what the production window looks like when you have multiple tasks, for example when you are working on the track with the two robots in coordinated mode.

Manual

HvA_2600-112003_left

Guard Stop

Stopped (2 of 2) (Speed 100%)

Production Window : program in T_ROB1/mainmodule0/main

<div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin-top: 10px;">T_ROB1</p> </div> <div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center; margin-top: 10px;">T_ROB2</p> </div>	<p>21</p> <p>22 →</p> <p>23</p> <p>24</p> <p>25</p> <p>26</p> <p>27</p> <p>28</p> <p>29</p> <p>30</p> <p>31</p> <p>32</p> <p>33</p> <p>34</p> <p>35</p>	<pre style="color: green;"> PROC main() operation0; ENDPROC PROC operation0() !Set acceleration and acceleration r AccSet 100, 100; !Monitors the configuration during l Confl\On; SingArea\Wrist; CirPathMode\CirPointOri; </pre>
---	---	---

Load Program... PP to Main Debug ▲

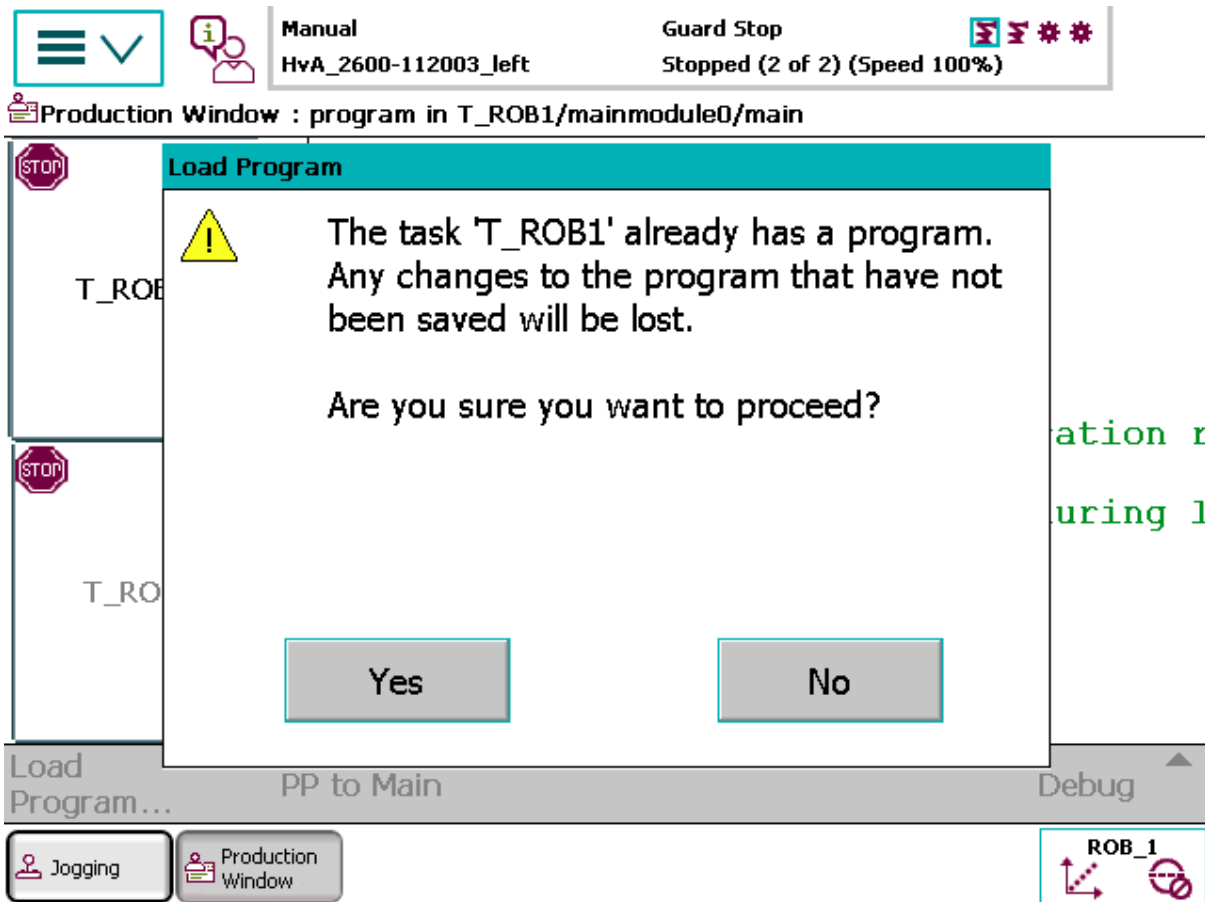
Jogging

Production Window

ROB_1

When you are working with just one robot, and one task, you will not have the tabs on the left. The rest of the window will be the same.

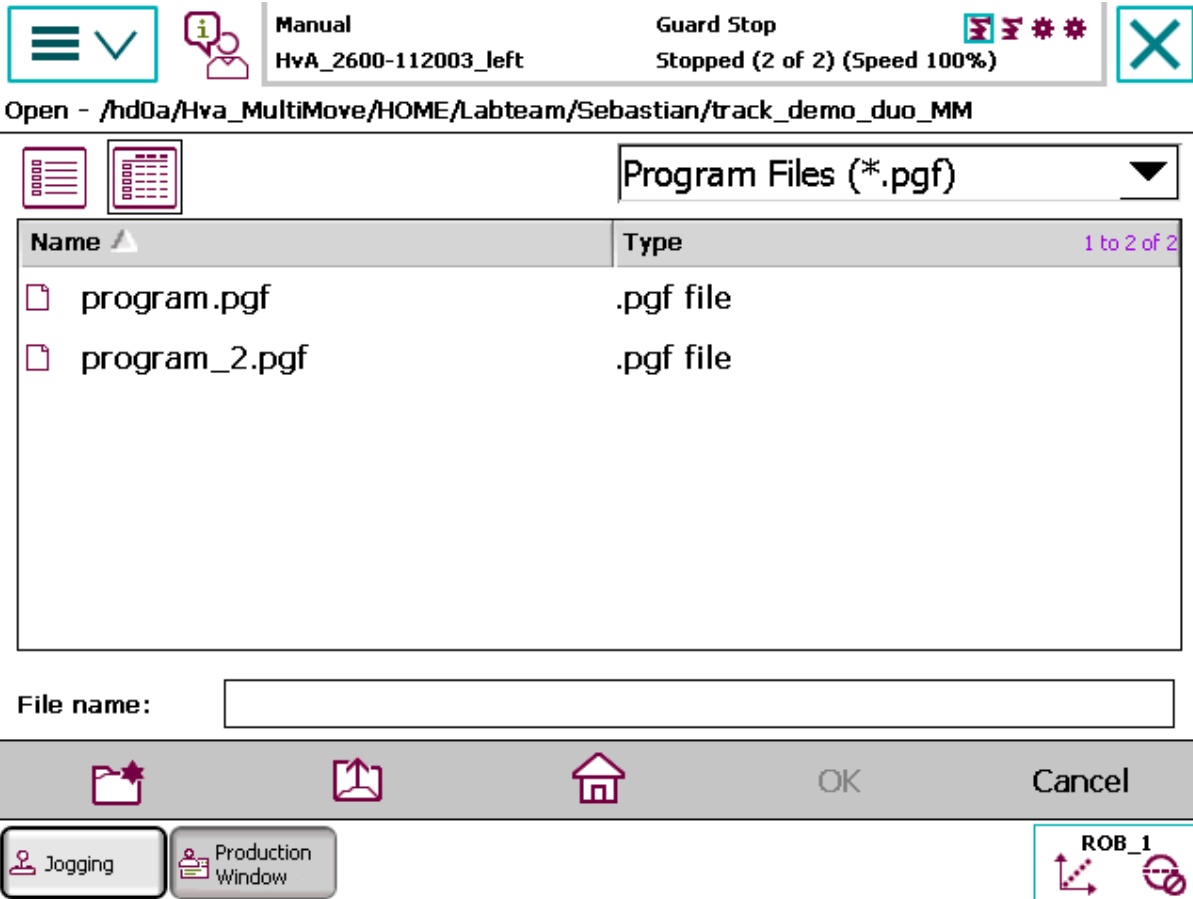
Next is loading your program, press Load Program...



Now confirm with Yes if you want this.

Now you can browse to your file.

This file can either be placed on the robot controller through FTP. Or you can use a USB stick, the flexpendant has a USB stick port.



Speed

Before running any programs, simulate them in a virtual environment to catch any programming mistakes that were made.

But even with a virtual simulation done, most probably you still would like to start careful with your program. You never know what surprises the real world has for you. To do this we will decrease the speed of our program.

Click on the icon bottom right to open this menu.

Click on the speedometer icon to open the speed pane.

Manual
HvA_2600-112003_left

Guard Stop
Stopped (2 of 2) (Speed 100%)

Production Window : program in T_ROB1/

21	PROC main
22	operatio
23	ENDPROC
24	
25	
26	
27	PROC opera
28	!Set acc
29	AccSet
30	!Monitor
31	Confl\O
32	SingArea
33	CirPath
34	
35	

T_ROB1

T_ROB2

Speed

100 %

- 1 % + 1 %

- 5 % + 5 %

0 % 25 % 50 % 100 %

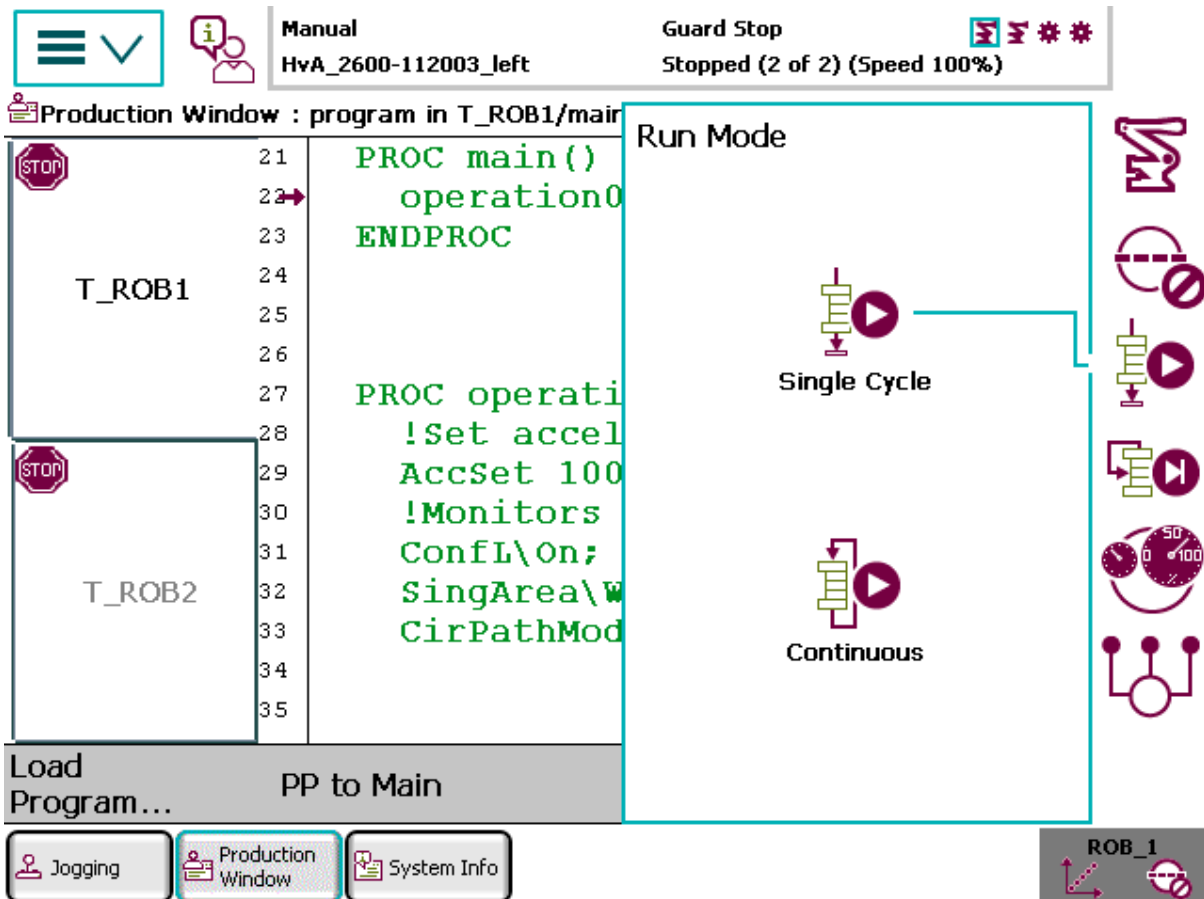
Load Program... PP to Main

Jogging Production Window System Info

ROB_1

Here you can reduce the execution speed of your program.

Also note that you can set the increment in another pane. And also whether or not your programs repeats infinitely:



Error notifications (and what to do/FAQ)

Singularity

Singularity errors with an industrial robot arm mean that the controller is not able to calculate how to move all axes to get the robot to move in a certain direction along a straight line. To move in straight lines the robot controller needs to have a kinematic model of the physical robot and there is quite some math involved to calculate how fast each axis has to rotate to make a straight line in the cartesian coordinate system.

At certain points the math will yield an infinite amount of solution, and here the controller has to give up.

Also when the robot moves closeby such points, some axes need to move very fast to maintain the linear speed. This can also be a reason why you get errors.

In the following video you can see how there are an infinite amount of solutions to the same position in a 6 axis robot.

<https://www.youtube.com/watch?v=ID2HQCxeNoA>

There are various methods to avoid singularities, one of them is to mount the end-effector under an angle like here:



Screenshot from
https://www.linkedin.com/posts/saxion-lectoraat-industrial-design_trudy-at-work-activity-6770688921002115072-ma_p

TODO

out of reach
no program pointer
etc

Programming

Online vs offline programming

In industrial robotics often people are talking about online vs offline programming. What they mean is the following:

Online

Programming in the real world. So you jog the robot to a position you like (possibly only watching the real world, not caring about any numbers). And then you save that position. You can repeat this until your program is done.

Offline

Programming without the real robot. So you program in your computer, possibly with a virtual model, and prepare everything from there.

Getting location in the real world

At the robot lab we often program offline. But we use the robot to get certain positions. For example, you would prepare operations in a work object coordinate system. You clamp your physical object on the table. But now you need to know how to define the location of your work object in the world coordinate system.

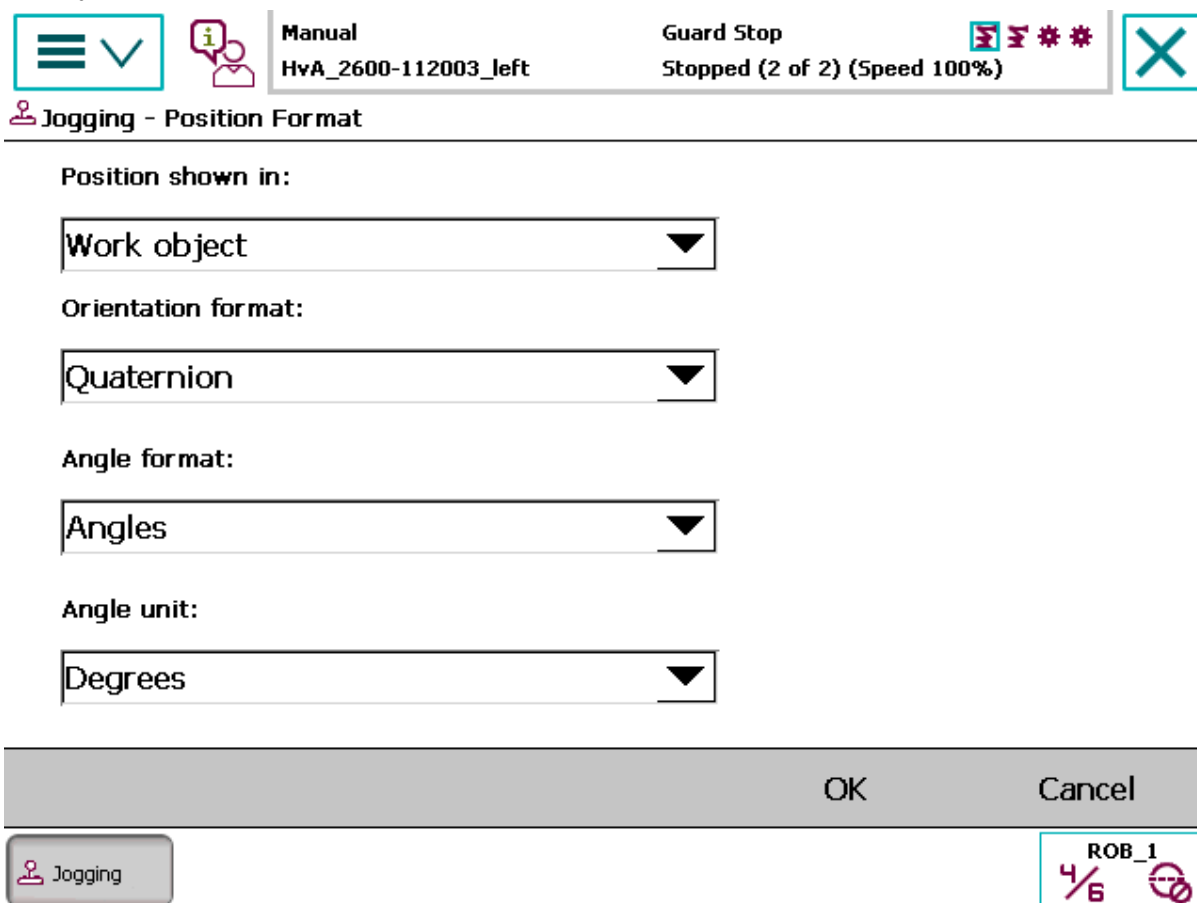
Often you use the robot to do this. There is for example a 3 points method to teach in work objects by jogging with the robot and touching 3 points on the work object.

Check out the video on how to do this:

https://www.youtube.com/watch?v=8E-1LwoE_k

Or sometimes you only need to know the height of something for example, then you jog the robot towards the surface, and you take note of the Z-value.

Be aware that the position shown in you jogging screen should be set to the appropriate value you need:



The screenshot shows a control panel for a robot. At the top, there is a status bar with a menu icon, a checkmark, an information icon, the text "Manual HvA_2600-112003_left", "Guard Stop Stopped (2 of 2) (Speed 100%)", and a close icon. Below this is a title bar "Jogging - Position Format". The main area contains four dropdown menus: "Position shown in:" set to "Work object", "Orientation format:" set to "Quaternion", "Angle format:" set to "Angles", and "Angle unit:" set to "Degrees". At the bottom, there are "OK" and "Cancel" buttons. A "Jogging" button is on the left, and a "ROB_1" status indicator showing "4/6" and a stop icon is on the right.

TODO

Different software tools (Robotstudio, RoboDK, Rhino Grasshopper)

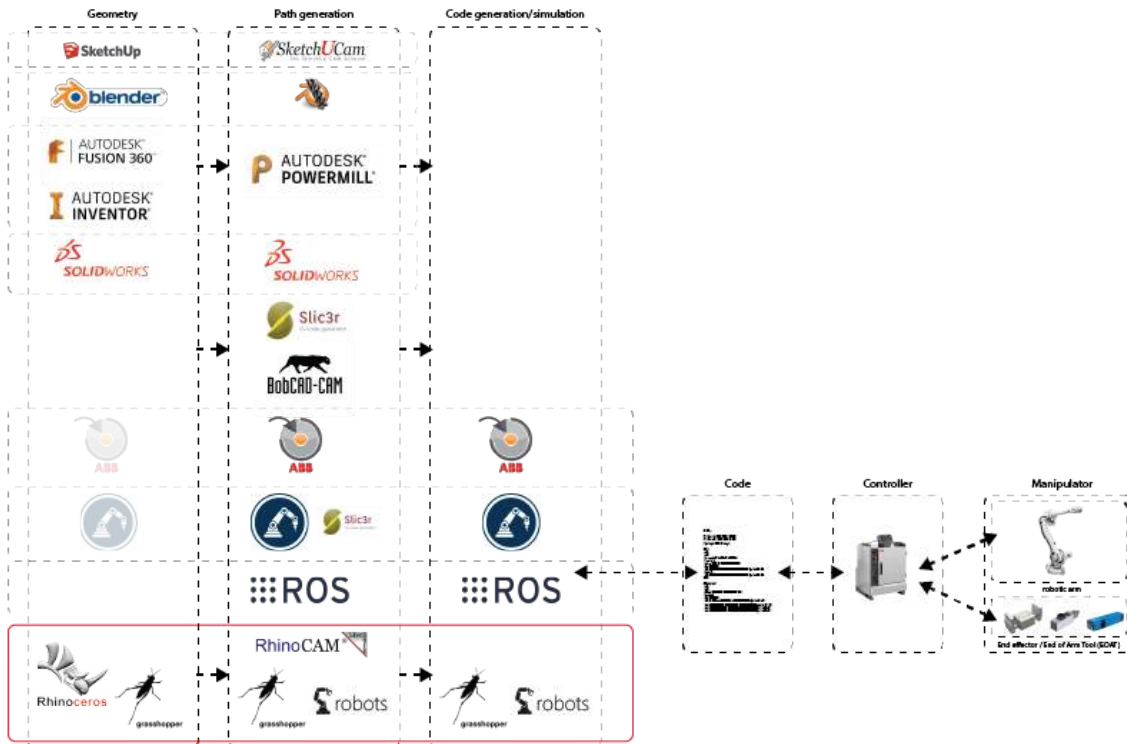


ABB RAPID

Compressed air

pressure relieve / filters / regulators

manifold

ABB IO [video needed]

3D scanner - Trispector

Here is an instruction video on how to configure the Trispector

<https://www.youtube.com/watch?v=XKPMWTLDayM>

List of instruction videos

These videos are also referenced in the text above.

<https://www.youtube.com/watch?v=1xY-NbJ3coE>

<https://www.youtube.com/watch?v=edw8ZZrRwS4>
<https://www.youtube.com/watch?v=eFI7R45ROq4>
https://www.youtube.com/watch?v=8E-1LlwoE_k
<https://www.youtube.com/watch?v=XKPMWTLDayM>