Motivation

- Insights in working procedures for industrial robots
- Hands on experience, hardware
- Hands on experience, software
- Other robot brands: working principles are similar for most industrial robots
- Exercises: 1 hands-on, 3 computer exercises which include programming, simulation and upload and testrun on the robot

Geometrical and load characteristic data

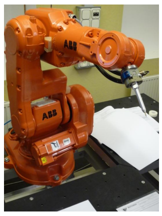
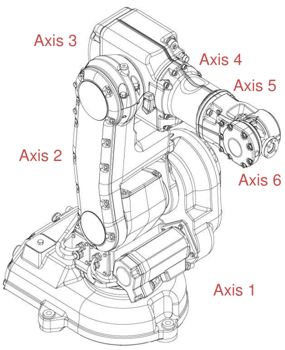
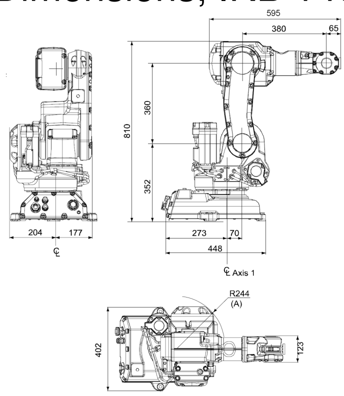


ABB IRB 140 industrial robot

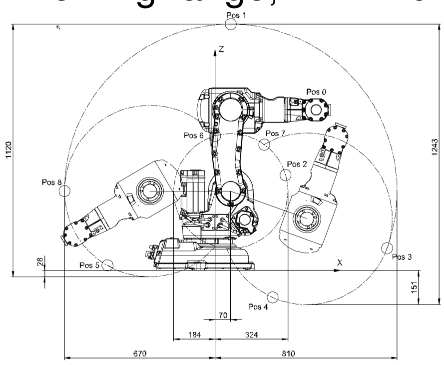


Type of motion	Range of movement
Axis 1: Rotation motion	+ 180° to - 180°
Axis 2: Arm motion	+ 110° to - 90°
Axis 3: Arm motion	+ 50° to - 230°
Axis 4: Wrist motion	+ 200° to - 200°
Axis 5: Bend motion	+ 115° to - 115°
Axis 6: Turn motion	+ 400° to - 400°

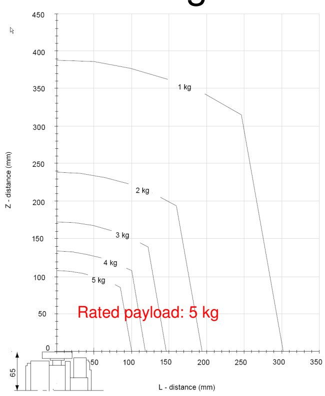
Dimensions, IRB 140



Working range, IRB 140



Load diagram



Load	Description
J5: Maximum static load	T5 = 9.81 x Mass x $((Z + 0.065)^2 + L^2) \leq 8.5$ Nm
J5: Maximum dynamic load	J5 = Mass x $((Z+0.065)^2 + L^2)$ + max $(J_{0L}) \leq 0.35$ kgm^2
J6: Maximum static load	T6 = 9.81 x Mass x L $\leq$ 4.9 Nm
J6: Maximum dynamic load	J6 = Mass x $L^2 + J_{0Z} \leq$ 0.24 kgm^2

Rated load: as indicated in the diagram, the load depends on where the center of gravity is located. The load can be higher, which many times is used in industry

Examples of what the robot can be used for

Packaging and palletizing at Astra Zeneca

Decorating pastry at Hacos

Performance: robot and IRC5 controller

ABB motion control capability

Safe move with MultiMove

Hands-on: Safety first!

Safety issues

- If needed, use emergency stop
  - Located on the teach pendant and the controller cabinet
  - Note: Do not use as a standard stop method. The emergency stop causes extra wear on the robot
- Tools attached to the robot can move freely in 3D space
  - Stay out of reach during program run
- Pneumatic tool (gripper) is stronger than you think
  - Stay out of reach with fingers from the gripper when activated

Safety functions

- The Service Information System (SIS)
  - The service information system gathers information about the robot's usage and determines how hard the robot is used
  - The usage is characterized by the *speed*, the *rotation angles* and the *load* of every axis
  - The *time* the robot is in operation (brakes released) is indicated on the FlexPendant
  - Data can also be monitored over network, using for example WebWare.

Active safety system

- Active brake system
- Self tuning performance
  - Power mode / speed mode
- Electronically stabilized path
  - Load and inertia
- Over-speed protection
- Restricting the working space
- Collision detection

The active safety system includes those software features that maintain the accuracy of the path of the robot and those that actively avoid collisions which can occur if the robot leaves the programmed path accidentally or if an obstacle is put into the path of the robot.

Passive safety system

- Compact robot arm design
- Optionally electronic position switches

The Process Robot Generation has a dedicated passive safety system that by hardware construction and dedicated solutions is designed to avoid collisions with surrounding equipment. It integrates the robot system into the surrounding equipment safely.

Internal safety concept

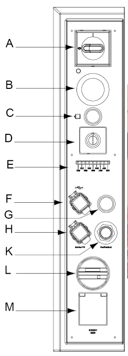
- Safety category 3
  - Detection of malfunctioning component
- Selecting operating mode
- Reduced speed
- Three position enabling device
- Safe manual movement (joystick)
- Emergency stops
- Safe guarded stops / operation

The internal safety concept of the Process Robot Generation is based on a two-channel circuit that is monitored continuously. If any component fails, the electrical power supplied to the motors shuts off and the brakes engage.

Category requirements

Control cabinet and Teach Pendant Unit (TPU)

Operators panel, control cabinet



Pos	Name
A	Mains switch and remote control of power to Drive modules
B	Emergency stop - if pressed in, turn to release
C	MOTORS ON
D	Operating mode selector
E	Safety chain LEDs (option)
F	USB connection
G	FlexPendant Hot Plug pushbutton (option)
H	Service PC connection
K	FlexPendant connection
L	Duty Time Counter (option)
M	Service outlet 115/230 V, 200 W (option)

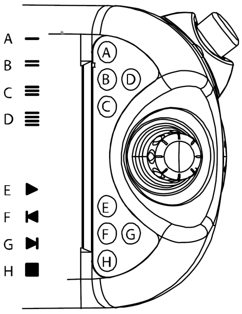
FlexPendant (or TPU Teach Pendant Unit)

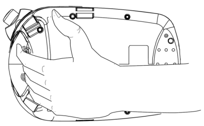
Pos	Description
A	Display
B	User defined keys
C	Emergency stop button
D	Joystick
E	Program execution keys

Enabling device (back side): When moving the robot the push button must be pressed halfway in, which takes the system to MOTORS ON. When released or pushed all the way in, the robot is brought to the MOTORS OFF state.

TPU handling and buttons

A-D	Programmable keys, A-D
E	START button. Starts program execution. Step
F	Step BACKWARD button.
G	Step FORWARD button
H	STOP button. Stops program execution.





The TPU is typically operated by holding the unit and pressing the enabling switch to the middle

Touch Screen Elements

ABB Menu:
- HotEdit
- Inputs and Outputs
- Jogging
- Production Window
- Program Editor
- Program Data
- Backup and Restore
- Calibration
- Control Panel
- Event Log
- FlexPendant Explorer
- System Info

Connect the robot to a PC

RobotStudio Online

RobotStudio Online is a PC software which connects to the robot, as a complement to working from the FlexPendant.
RobotStudio Online is optimized for text based programming

- The *System Builder* for creating, installing and maintaining systems.
- A *configuration editor* for editing the system parameters of the running system.
- A *program editor* for online programming.
- An *event log* for monitoring and saving robot events.
- Tools for *backing up and restoring* systems.
- An *administration tool* for the User Authorization System.
- A *file manager* for transferring files between the PC and controllers.
- A *task window* for operating all kinds of tasks in the controller.
- Other tools for viewing and handling controller and system properties.

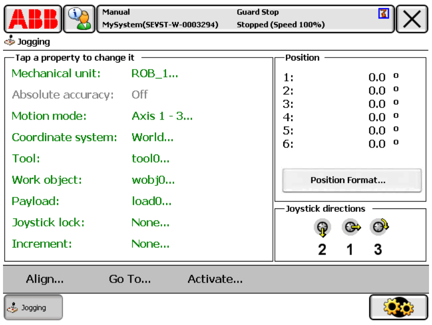
Power up and shut down the system

Start and shut down the system

- Use the main switch on the controller
- The system is quite stable and rebooting is usually not necessary, but shut down and turn on if needed
- Collisions, joint limits, etc
  - The robot will alarm and stop immediately
  - Acknowledge alarm on TPU and jog away from area

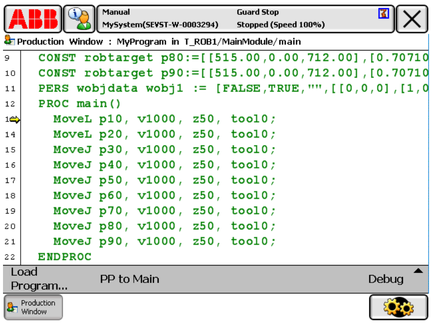
Some useful Operator windows in the TPU

Jogging the robot



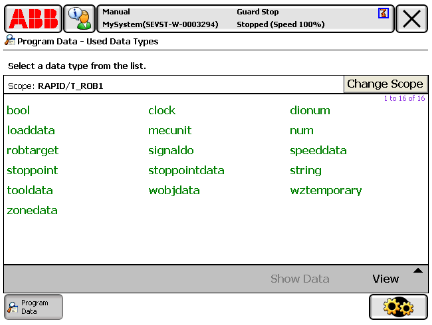
Most common:
- Motion mode
- Coordinate system
- Tool
- Work object
- Payload
- Increment

Production window



Used to view the program code while the program is running
- Load program
- PP to Main: move program pointer to the routine main
- Debug

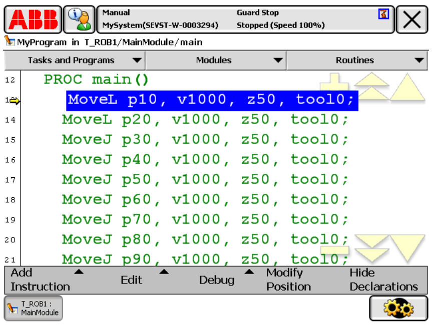
Program data



The Program Data view contains functions for viewing and working with data types and Instances

Often used:
- Tooldata
- Wobjdata

Program editor



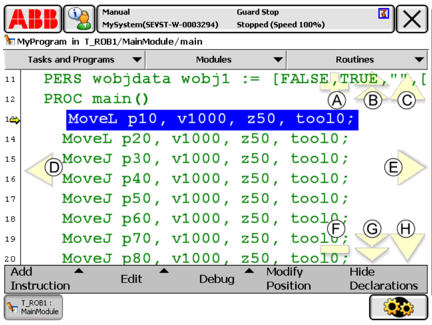
This is where you create or modify programs

- Tasks and Programs: Menu for program operations
- Modules : Lists all modules
- Routines: Lists all routines
- Add Instruction: Opens instruction menu
- Edit: Opens edit menu
- Debug: Functions for moving the program pointer, service routines etc.

Calibration

- Resolver revolution
  – The robot has lost its position and the home position looks strange
- Tool calibration
  – This is included in the hands on exercise and also part of kinematics related exercises

Scrolling and zooming



- A Zoom in (larger text)
- B Scroll up (the height of one *page)*
- C Scroll up (the height of one *line)*
- D Scroll left
- E Scroll right
- F Zoom out (smaller text)
- G Scroll down (the height of one *page)*
- H Scroll down (the height of one *line)*

Jogging

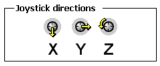
- To jog is to manually position or move robots or external axes using the FlexPendant joystick
- Jogging is done in manual mode
- The selected motion mode and/or coordinate system determines the way the robot moves
- Jogging can be made in custom defined coordinate system, allowing more complicated movements

More on Jogging

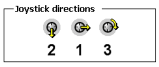
- The joystick response is proportional to how you handle it
  - Slow start
  - Gentle stick move to generate gentle robot motion
- Other robot makers may use push buttons to generate motions instead of a joystick

Jogging mode

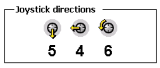
- Linear



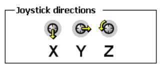
- Axis 1-3



- Axis 4-6



- Reorient



Default settings:
The linear and reorientation motion modes have default settings for coordinate systems

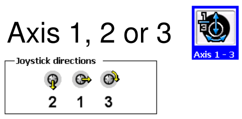
Linear: Base coordinate system
Reorientation: Tool coordinate system

Tool, work object, and payload

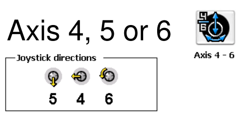
- Selecting tool, work object will define where and how to move in the work space
- A payload must be defined, otherwise the system will almost always stop due to overload errors
- Payload data is included in the tool data together with center of gravity data
- Exact value is not so critical

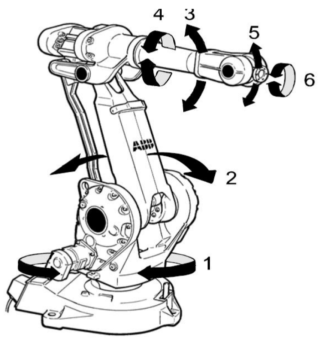
Jog axis by axis

- Axis 1, 2 or 3



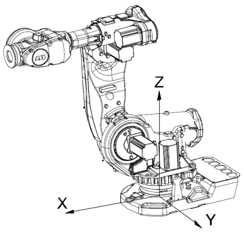
- Axis 4, 5 or 6



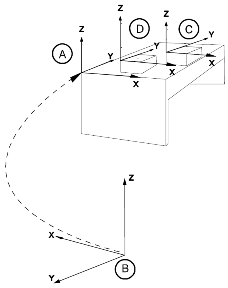


Jog in base coordinates

- Base coordinates are defined from the base of the robot and its setup (floor mounted, upside down, etc)
- World coordinates can be defined and used for jogging, e.g. when two robots should follow the same coordinate axis motion for any jog move instruction



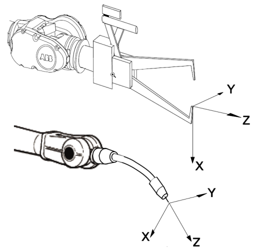
Work object and tool coordinates



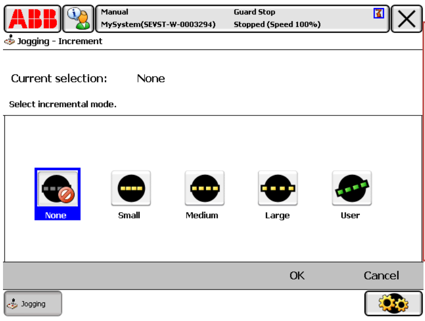
Defined frames
- Work object
- Tool frames

These frames can be used during jogging

This is useful when motions are defined with reference to a work object or a tool



Incremental movement

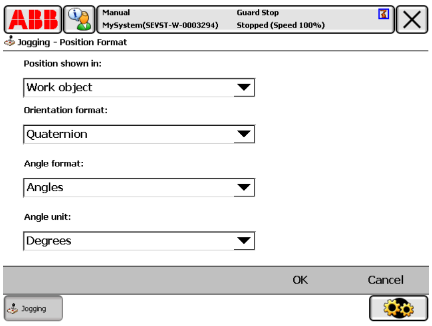


Use incremental movement to jog the robot in small steps, which enables very precise positioning.

This is important to avoid collisions close to other objects

Small: 0.05 mm / 0.005 deg
Medium: 1 mm / 0.02 deg
Large: 5 mm / 0.2 deg

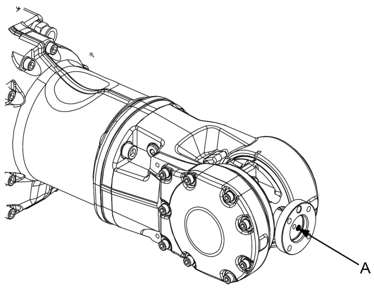
How to read the position



Positions are always displayed as:
- The point in space expressed in the x, y and z tool center point coordinates
- The angular rotation of the tool center point expressed in Euler angles or as a quaternion.

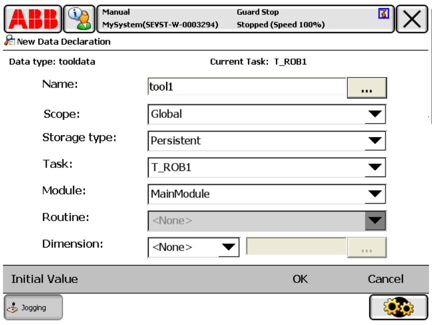
Tool definition and calibration

Creating a tool



- The TCP of the default tool (tool0) is in the center of the mounting flange (A)
- A new TCP can be defined as needed
- Data for the tool can be defined by the robot and / or by typing them
- Examples of data are position and orientation, weight and center of gravity, name

Tool definition menu

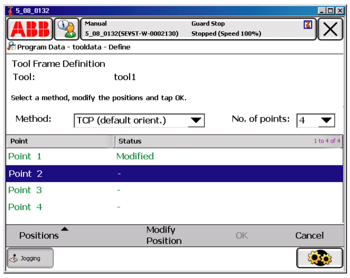


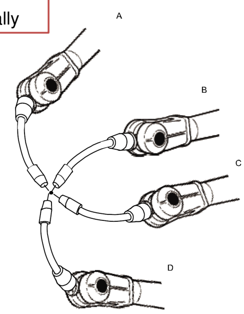
Typical data definition of a tool

Name can be any descriptive name

Defining the tool frame

- TCP can be defined in many ways:
- Here, the 4 point method is shown where the robot is positioned around the same point in space
- The robot then calculates the TCP automatically

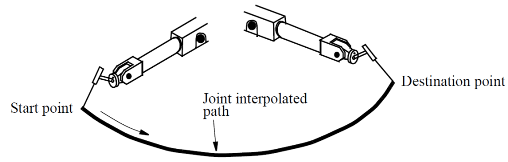




Motion and interpolation during program execution

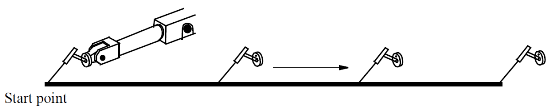
Joint interpolation – MoveJ

- Used to move the tool quickly from one position to another
- Used when the path between points is not important
- Allows movement to any location in a single movement
- The velocity is defined in mm/s but will only be approximate
- All axis which moves will stop at the same time



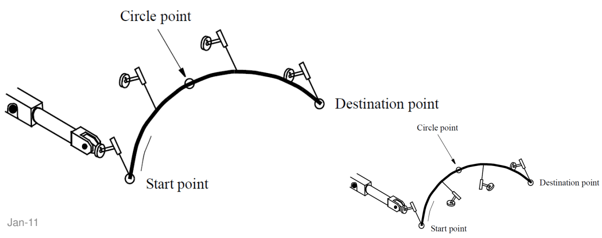
Linear interpolation – MoveL

- Linear movement in Cartesian space
- Movements may not be possible due to joint limits
- The path and velocity through singular areas may be different from defined
- Any reorientation of the tool is made at constant velocity



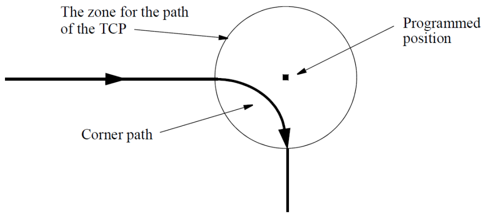
Circular interpolation – MoveC

- A circular path is defined by three programmed poses
- A full circle requires two MoveC
- Behavior of any reorientation must be checked

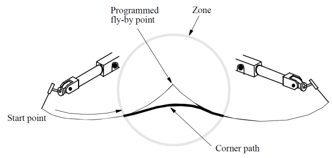


Pose – targets and paths

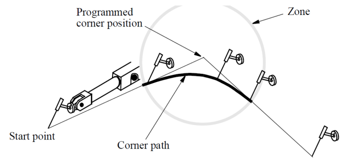
- Movements can use arguments which define interpolation, velocity and how to pass through a pose
- A “Fine” argument means a full stop before continuing
- A “zone” value greater than zero means a “fly-by” motion



Interpolation in corner paths

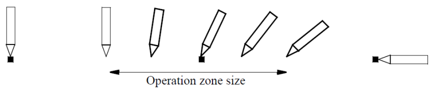


Joint interpolation



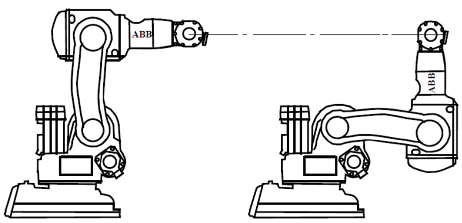
Linear interpolation

Changing orientation

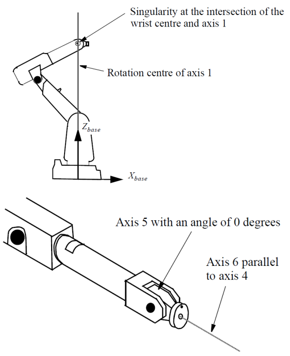


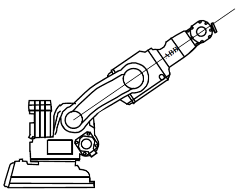
Configurations

- The same pose can be reached using different configurations
- Specific instruction can be used to define how to reach a pose



Singularities





Summing up

- Hands-on exercise will provide an introduction of the basics of operation and programming robots
- Computer exercises will complement this with more advanced tools including simulation
- Making robot programs for complex processes or production systems is an art and requires great skill from involved personnel