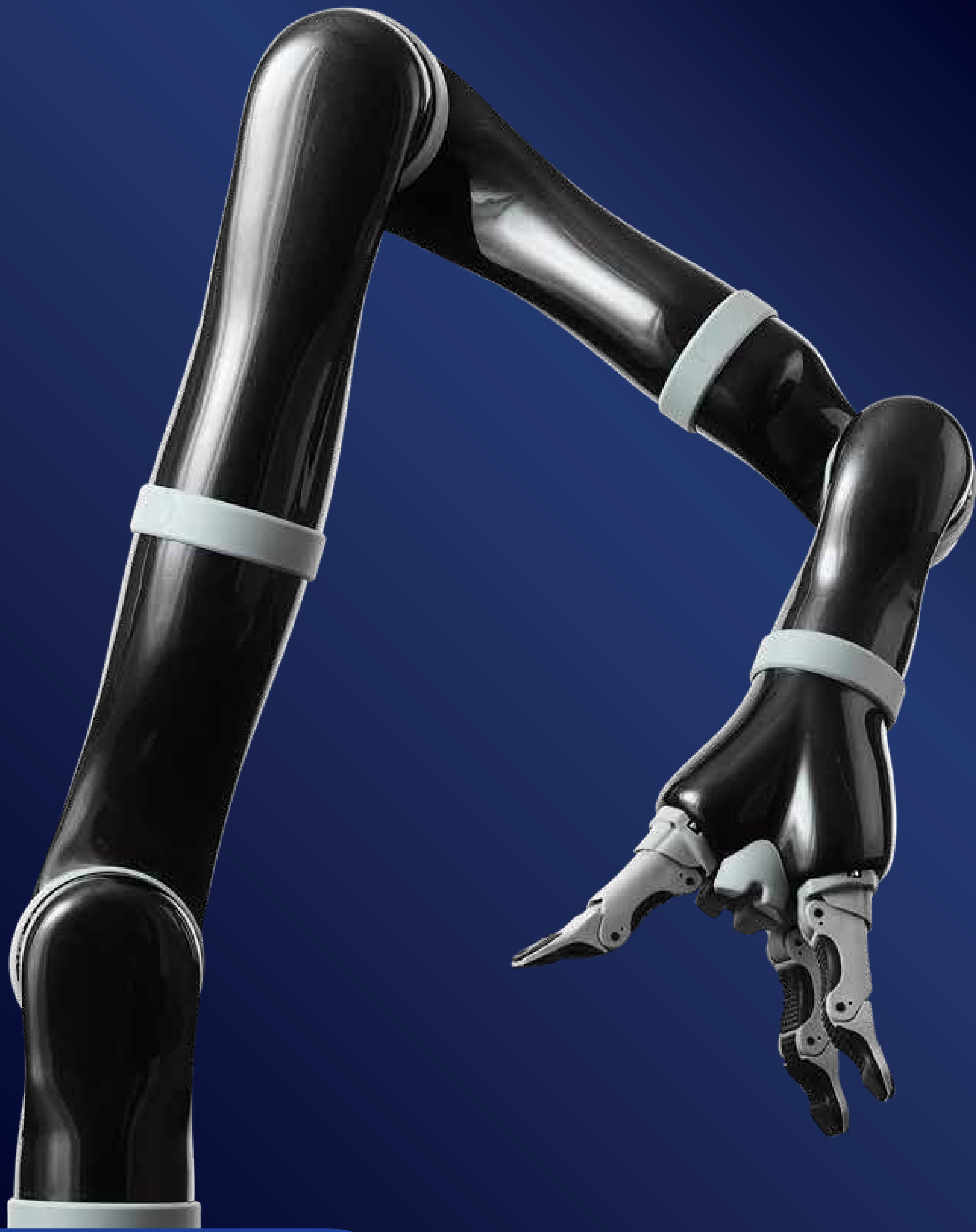


User Guide

KINOVA® *Gen2 Ultra lightweight robot*

KINOVA
Achieve Extraordinary

Contents

Introduction

About this document

This document contains information on the installation and operation of the KINOVA® *Gen2 Ultra lightweight robot*.

⚠ Read all instructions before using this product.

⚠ Read all warnings on the product and in this guide.

⚠ Follow all instructions.

⚠ Keep these instructions for future reference.

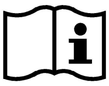
This document contains information regarding product setup and operation. It is intended for:

- Kinova product end users
- Field service, customer support and sales employees of authorized Kinova distributors

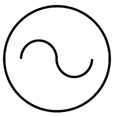
Symbols, definitions, and acronyms

 Important information regarding the safety related to the product and the user.

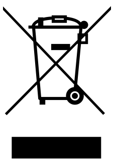
 Tip on the maintenance, operation and manipulation of Kinova's products.

 Refer to accompanying documents.

 Direct current.

 Alternating current.

 Operating temperature range.

 Compliance with WEEE2 directive.

Compliance with ROHS3 directive.

 Type BF Applied Part device.

Warranty

This section describes the Kinova warranty terms.

Subject to the terms of this clause, Kinova warrants to End User that the Products are free of defects in materials and workmanship that materially affect their performance for a period of two (2) years from the date Kinova ships the Products to the End User ("Delivery Date").

Kinova agrees to repair or replace (at Kinova's option) all Products which fail to conform to the relevant warranty provided that:

1. Notification of the defect is received by Kinova within the warranty period specified above.
2. Allegedly defective Products are returned to Kinova, at the End User's expense, with Kinova's prior authorization within thirty (30) days of the defect becoming apparent.
3. The Products have not been altered, modified or subject to misuse, incorrect installation, maintenance, neglect, accident or damage by excessive current or used with incompatible parts
4. The End User is not in default under any of its obligations under this Agreement.
5. Replacement Products must have the benefit of the applicable warranty for the remainder of the applicable warranty period.

If Kinova diligently repairs or replace the Products in accordance with this section, it will have no further liability for a breach of the relevant warranty.

Allegedly defective Products returned to Kinova in accordance with this contract will, if found by Kinova on examination not to be defective, be returned to End User and Kinova may a charge a fee for examination and testing.

The warranty cannot be assigned or transferred and is to the sole benefit of the End User.

Where the Products have been manufactured and supplied to Kinova by a third party, any warranty granted to Kinova in respect of the Products may be passed on to the End User.

Kinova is entitled in its absolute discretion to refund the price of the defective Products in the event that such price has already been paid.

Safety / Warnings

⚠ It is not recommended to use the robot under heavy rain or snow.

⚠ Never use the Home / Retract function when carrying liquid. The home position is preset and the wrist may rotate and drop the liquid.

⚠ Do not manipulate cutting, very sharp or any dangerous tools or objects with the robot.

⚠ When the power is turned off, the robot will fall down and may cause damage to itself, depending on its position at the time of disconnection. Be sure to support its wrist before turning the power off.

⚠ Do not force the fingers beyond their maximal opening. This could damage some internal components.

⚠ Do not immerse any part of the robot under water or snow.

⚠ When lifting weight near the maximum load and reach, and you receive a warning, put down the object in the gripper, bring back the robot to Home or Retract position and wait until the warning goes away before using it again.

⚠ Do not block the robot movement when it is performing a Retract position trajectory.

Disclaimer

KINOVA® and Kinova's logo are trademarks of Kinova Inc., herein referred to as Kinova. All other brand and product names are trademarks or registered trademarks of their respective corporations.

The mention of any product does not constitute an endorsement by Kinova. This manual is furnished under a lease agreement and may only be copied or used within accordance with the terms of such lease agreement. Except as permitted by such lease agreement, no part of this publication may be reproduced, stored in any retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written consent of Kinova.

The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Kinova. Kinova assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Changes are periodically made to the information herein; these changes will be incorporated into new editions of this publication. Kinova may make improvements and/or changes in the products and/or software programs described in this publication at any time.

Address any questions or comments concerning this document, the information it contains or the product it describes through the Kinova website support page:

www.kinovarobotics.com/support

Kinova may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligations to you.

General Information

The KINOVA® *Gen2 Ultra lightweight robot* is a light-weight robot composed of four, six, or seven inter-linked segments. Through the controller or through a computer, the user can move the robot in three-dimensional space and grasp or release objects with the gripper (if a gripper is installed).

⚠ Do not modify equipment without the authorization of the manufacturer.

⚠ The Normal Use definition contains some information fundamental to the proper operation of the robotic arm.

⚠ It is not recommended to use the arm under heavy rain or snow.

Normal use definition

This section describes the normal use of the robot.

The definition of a normal use of the robot includes that you can lift, push, pull or manipulate a maximum load of:

- **Continuously** 2.6 kg from minimum to middle reach for 6 DoF, 6 DoF-S, and 7 DoF-S (45-49 cm distance between the actuator #2 and the load, depending on the configuration) and 4.4 kg from minimum to middle reach for the 4 DoF (35 cm distance between actuator #2 and the load).
- **Temporary** 2.2 kg from middle to full reach for 6 DoF, 6 DoF-S, 7 DoF-S (90-98 cm distance between actuator #2 and the load, depending on the configuration) and 3.5 kg from middle to full reach for 4 DoF (75 cm distance between actuator #2 and the load).

The robot is designed to be able to hold objects in the environment of the user, but it is a manipulator that in some positions and loads near the maximum reach and maximum loads holds for a long period, it can heat. When this occurs, before overheating and being dangerous for either the user or the robot, red lights on the joystick will blink. This is a warning. Simply put down any object in the gripper, and bring back the robot to the HOME or RETRACTED positions and wait until the warning goes away before using the robot.

If you don't use a Joystick in your application, make sure to read all the error statuses and temperature of all actuators modules via the API to ensure that they do not go higher than recommended parameters. If this occurs, the robot should be held in an idle position near the base for a certain time without any object in the gripper to cool down the robot.

⚠ When lifting weight near the maximum load and reach, if the red lights of the controller blinks, put down the object in the gripper, and bring back the robot to HOME or RETRACTED position and wait until the warning goes away before using it.

Note: During normal operation, the joints are subject to heating. The joints are normally covered with plastic rings which will protect the user from any danger that may be occurred by the heating of the metal parts.

The fingers of the robot are made flexible in order to protect the internal mechanism. When using the fingers to push on objects, the user must take special care not flex the fingers beyond their maximal opening as this could damage the internal mechanism.

⚠ Do not force the fingers beyond their maximal opening as this could damage some internal components.

Electromagnetic interference from radio wave sources

This section describes electromagnetic interference considerations for the robot.

Even if the product complies with all relevant standards, your robot may still be susceptible to electromagnetic interference (EMI), which is interfering electromagnetic energy (EM) emitted from sources such as radio stations, TV stations, amateur radio (Ham) transmitters, two way radios, and cellular phones. The interference (from radio wave sources) can cause the product to stop moving for a period of 10 seconds. In this case, the device will simply re-initialize and you will be able to continue to use it. In extremely rare case, it can also permanently damage the control system.

The intensity of the interfering EM energy can be measured in volts per meter (V/m). The product can resist EMI up to certain intensity. This is called "immunity level". The higher the immunity level is, the greater is the protection. At this time, current technology is capable of achieving at least a 20 V/m immunity level, which would provide useful protection from the more common sources of radiated EMI.

There are a number of sources of relatively intense electromagnetic fields in the everyday environment. Some of these sources are obvious and easy to avoid. Others are not apparent and exposure is unavoidable. However, we believe that by following the warnings listed below, your risk to EMI will be minimized.

The sources of radiated EMI can be broadly classified into three types:

1. Gripper-held portable transceivers (e.g. transmitters-receivers with the antenna mounted directly on the transmitting unit, including citizens band (CB) radios, walkie-talkie, security, fire and police transceivers, cellular phones, and other personal communication devices). Some cellular phones and similar devices transmit signals while they are ON, even if not being actively used.
2. Medium-range mobile transceivers, such as those used in police cars, fire trucks, ambulances and taxis. These usually have the antenna mounted on the outside of the vehicle.
3. Long-range transmitters and transceivers, such as commercial broadcast transmitters (radio and TV broadcast antenna towers) and amateur (Ham) radios. Other types of gripper-held devices, such as cordless phones, laptop computers, AM/FM radios, TV sets, CD players, cassette players, and small appliances, such as electric shavers and hair dryers, so far as we know, are not likely to cause EMI problems to your device.

Because EM energy rapidly becomes more intense as one move closer to the transmitting antenna (source), the EM fields from gripper-held radio wave sources (transceivers) are of special concern. It is possible to unintentionally bring high levels of EM energy very close to the control system while using these sorts of devices. Therefore, the warnings listed below are recommended to reduce the effects of possible interference with the control system.

⚠ Do not operate gripper-held transceivers (transmitter-receivers), such as citizens band (CB) radios, or turn ON personal communication devices, such as cellular phones, while the device is turned ON.

⚠ Be aware of nearby transmitters, such as radio or TV stations, and try to avoid coming close to them.

⚠ Be aware that adding accessories or components, close to the device may make it more susceptible to EMI.

⚠ Report all incidents of unintended shut down to your local distributor, and note whether there is a source of EMI nearby.

Markings and labels

This section describes markings and labels on the robot.

Please note that these labels may slightly differ from the ones accompanying your device depending of your country. The following figure depicts the information about the label affixed on the robot controller.

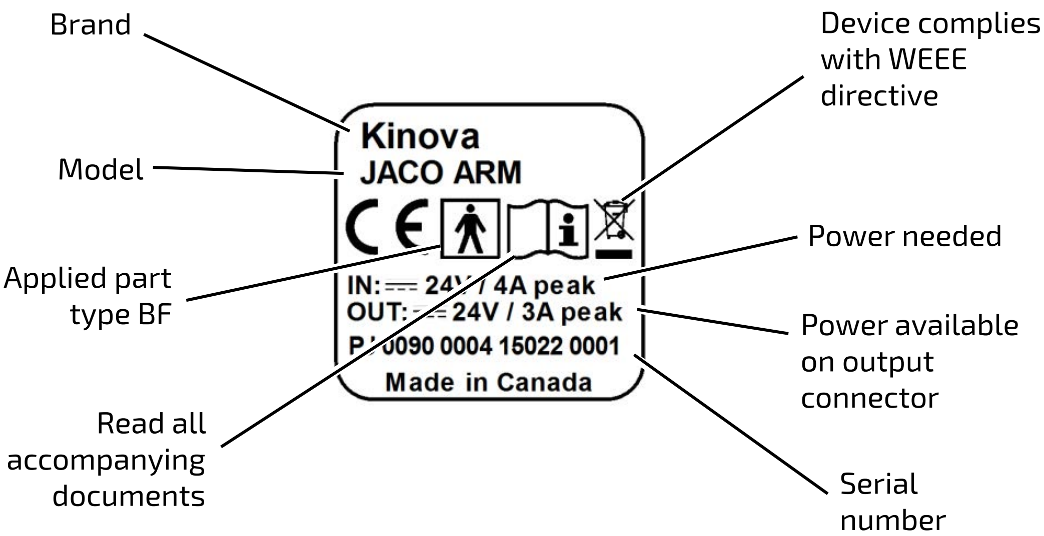


Figure 1: Robot label

Packing materials

The product packing material can be disposed of as recyclable material.

Metal parts

Metal parts can be disposed of as recyclable scrap metal.

Electrical components, circuit boards, and carbon fiber

Please contact your local distributor for information regarding disposal of such parts. You can also address questions directly to Kinova through our website (see Contacting Support).

Robot components and specifications

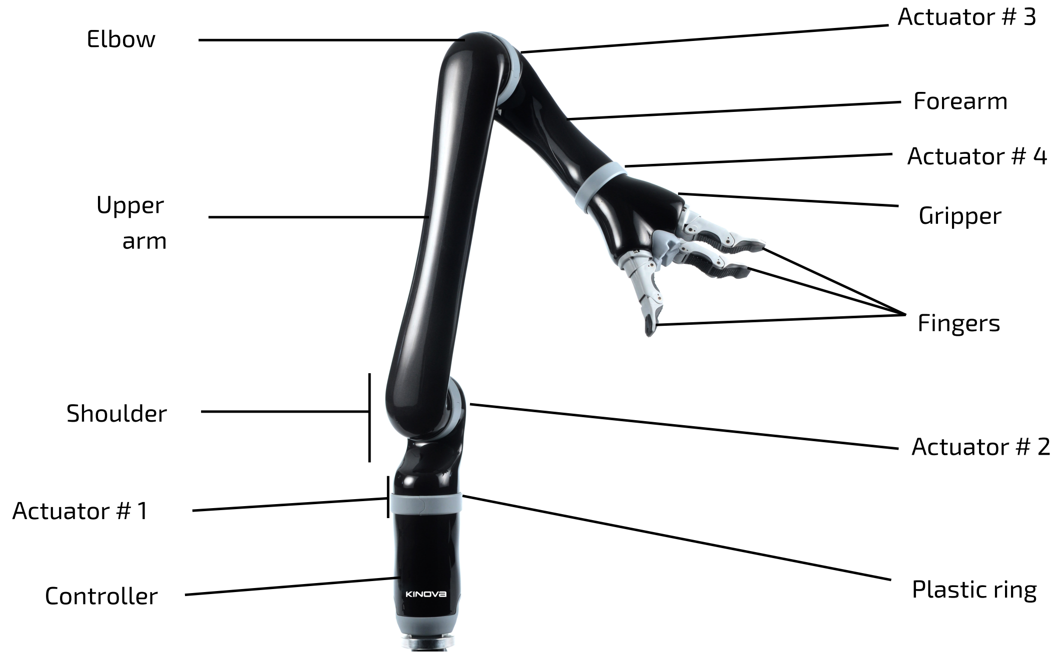
This section describes the different configurations available for the robotic arms.

For Innovation applications, the robot is available in four configurations:

- 4 degrees of freedom (DoF)
- 6 degrees of freedom (DoF) curved wrist
- 6 degrees of freedom (DoF) spherical wrist
- 7 degrees of freedom (DoF) spherical wrist

4 DoF Components

This section shows the components of the 4 DoF robot.



6 DoF Curved Wrist Components

This section shows the components of the 6 DoF curved wrist robot.

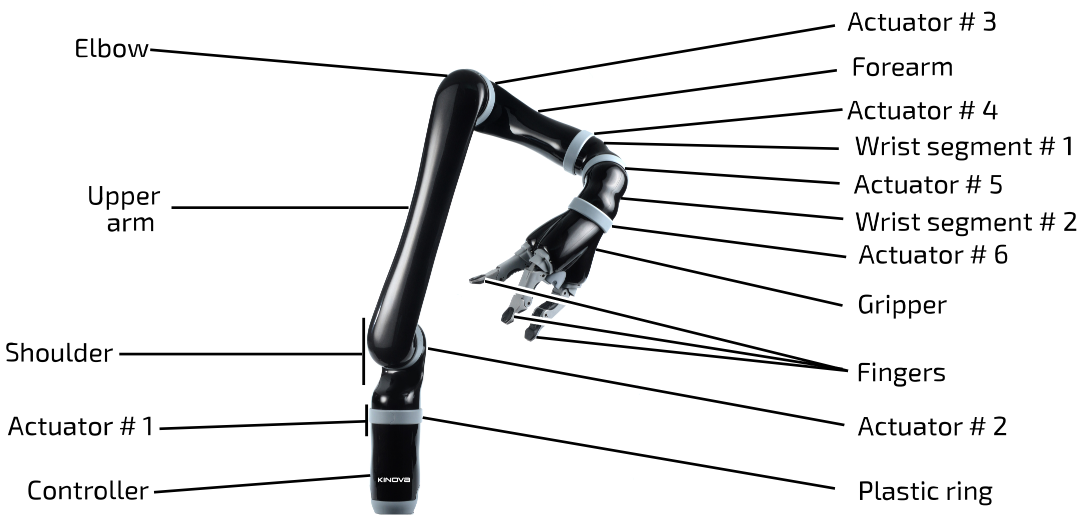
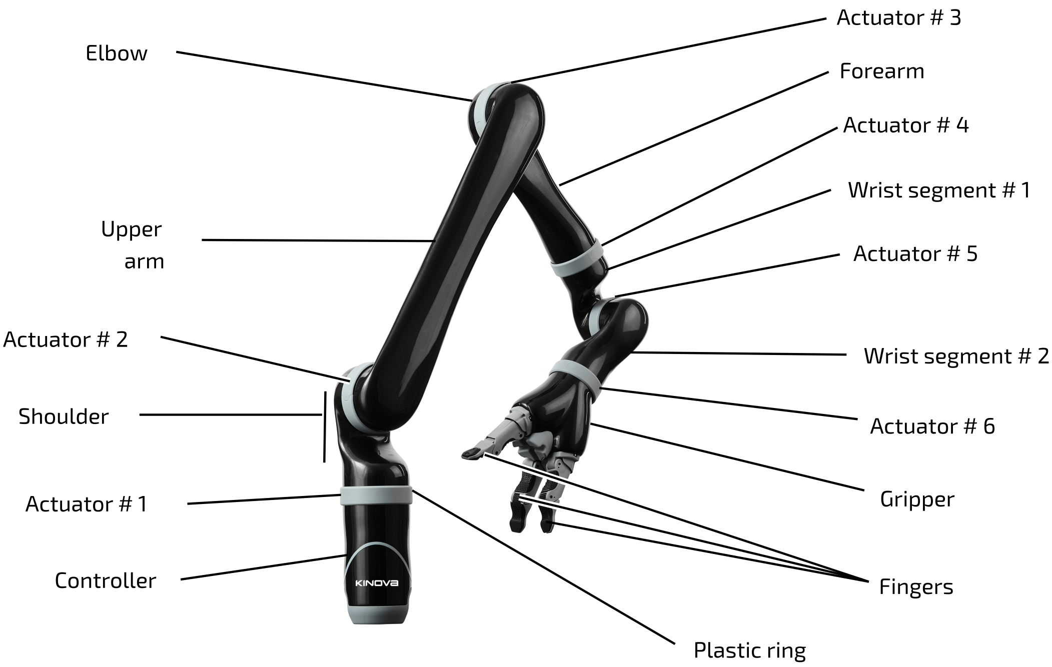


Figure 2: Robot components

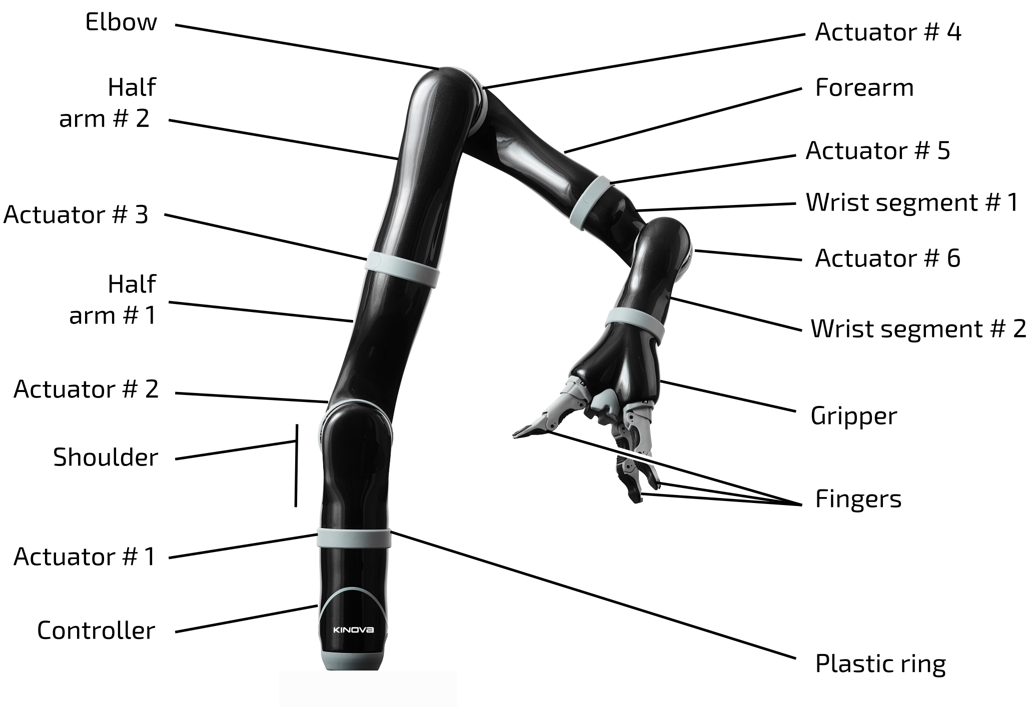
6 DoF Spherical Wrist Components

This section shows the components of the 6 DoF spherical wrist robotic arm.



7 DoF Spherical Wrist Components

This section shows the components of the 7 DoF spherical wrist robotic arm.



Robot external connectors

This section describes the external connectors on the base of the robot.

The following figure shows the external connectors located on the base of the robot.

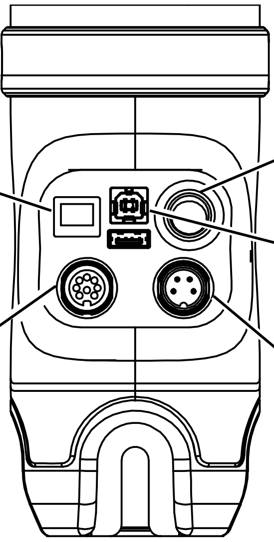


Figure 3: Robot external connectors

The panel at the back of the controller has several connectors and a power on / off switch.

The **power on /off switch** is used to power up or power down the robotic arm.

The four-pin **power connector** is used to connect the robotic arm to electrical power.

The eight-pin **joystick / controls / expansion port** is used to connect wired controllers for the arm.

Note: Two of the pins on this port are available for expansion purposes for researchers and application developers to connect alternate controllers or other devices.

The **USB port** is used to connect a computer for maintenance and configuration purposes.

The **Ethernet port** is used to operate the robotic arm programmatically using API commands.

⚠ The control port and power connector are intended to be connected only with a Kinova-approved device. Connecting other devices may result in poor performance, make the arm inoperable and void your warranty.

⚠ Do not override the safety purpose of the polarized or grounding-type plug. If the provided cable does not fit your outlet, consult an electrician for replacement of the obsolete outlet.

⚠ To prevent risk of fire or electric shock, avoid overloading wall outlets and extension cords.

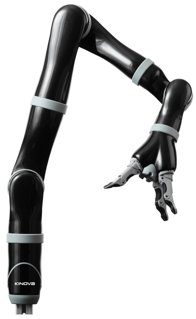
⚠ Protect the cords from being walked on or pinched.

Robot specifications

This section compares the different arm configurations.

Each of the four available robot configurations have their particular strengths. The best option depends on the needs of the specific users / group. The following table compares the four options.

Table 1: Configurations

	4 DoF	6 DoF curved wrist	6 DoF spherical wrist	7 DoF spherical wrist
				
Total weight	3.6 kg	4.4 kg	4.4 kg	5.5 kg
Reach	75 cm	90 cm	98.4 cm	98.4 cm
Maximum payload	• 4.4 kg (mid-range continuous) • 3.5 kg (full-reach peak / temporary)	• 2.6 kg (mid-range continuous) • 2.2 kg (full-reach peak / temporary)	• 2.6 kg (mid-range continuous) • 2.2 kg (full-reach peak / temporary)	• 2.4 kg (mid range continuous) • 2.1 kg (full-reach peak / temporary)
Materials	Carbon fiber (links), Aluminum (actuators)			
Joint range (software limitation)	± 27.7 turns			
Maximum linear arm speed	20 cm / s			
Power supply voltage	18 to 29 VDC			
Average power	25 W (5 W in standby)		25 W (15W standby)	
Peak power	100W			
Communication protocol	RS485			
Communication cables	20 pins flat flex cable			
Water resistance	IPX2			
Operating temperature	-10 °C to 40 °C			

	4 DoF	**6 DoF curved wrist**	**6 DoF spherical wrist**	**7 DoF spherical wrist**
Pros:	• Position control • Simplicity • More payload compared to 6 DoF and 7 DoF	• No pinch points • Proven technology • “Unique feel” wrist motion	• Simpler kinematics • Better reach • More payload than 7 DoF	• Ability to move arm without moving hand • More flexibility with movements • Less difficulties with singularities

Installation and setup

This section describes the main high level steps of the installation process.

The robot's installation consists of four high-level steps:

1. Mechanical integration
2. Electrical integration
3. End-effector electrical integration (optional)
4. Control integration

Mechanical mounting of the robot

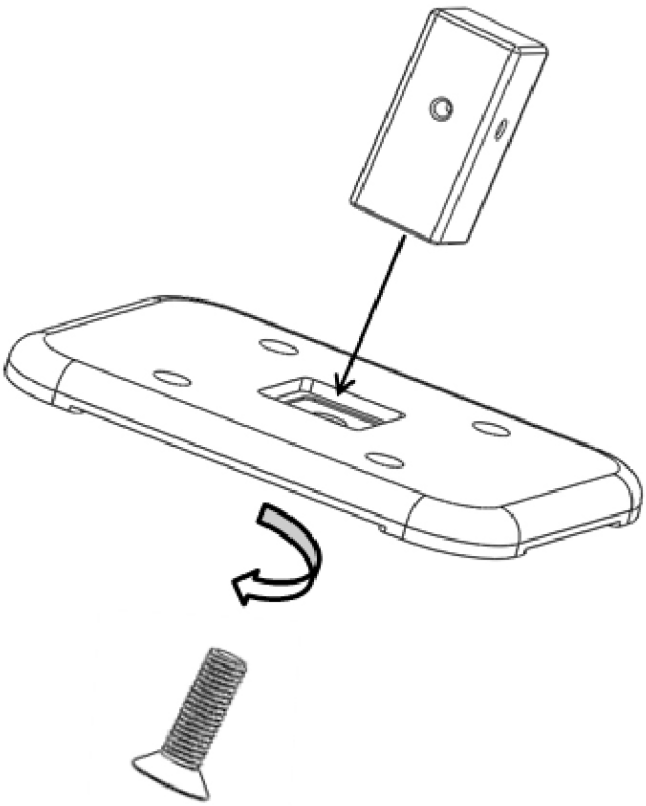
This section describes the steps for mechanical mounting of the robotic arm.

About this task

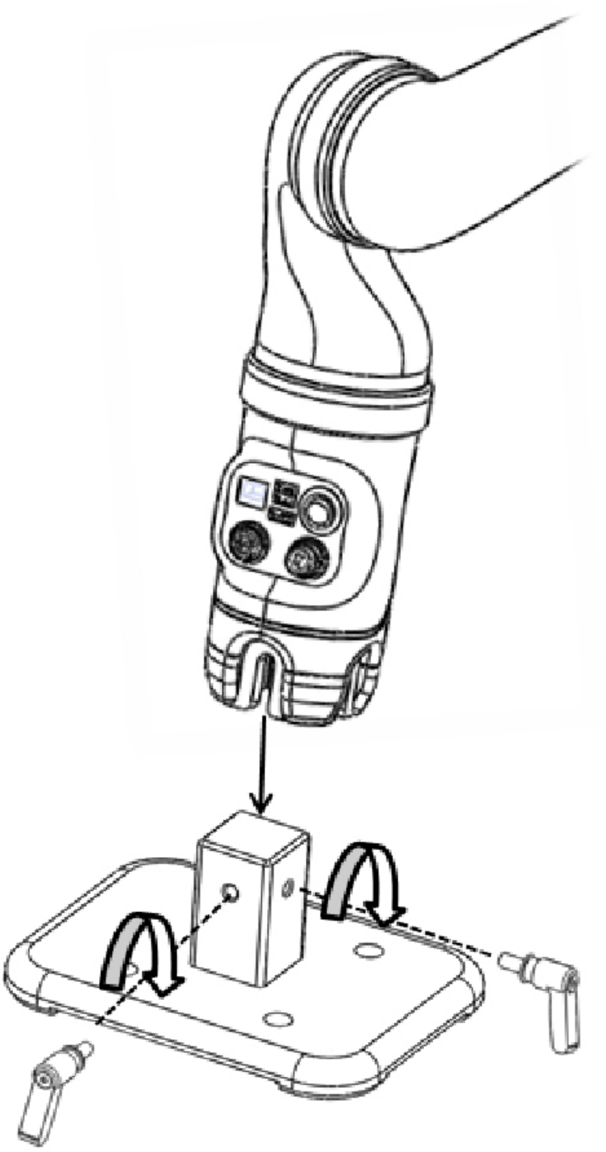
The arm is designed to be installed on a fixed surface or mobile platform. Please make sure the arm is fixed in such a way that its base cannot fall or break during operations involving maximum reach of the arm. Here is a guide on how to install the arm on the mounting kit (XK 0000 0014) supplied with your robot.

Procedure

1. Assemble the mounting kit. Insert the mounting post into the square cavity on the top of the mounting plate and use an 8 mm hex key to attach from the bottom of the mounting plate.



2. Affix the mounting kit to a flat surface. You can either place the larger side of the mounting kit on the edge of a solid flat surface and clamp it as firmly as possible by placing the two clamps supplied with the package on each side of the mounting post or secure four M12 screws through the holes in the mounting plate.
3. Insert the robot arm on the top of the mounting post. Screw the two M8 lever screws into the mounting post, one in the back of the controller and the other on one of the sides of the robot.



Electrical integration

This section describes how to connect the robot to an electrical power source.

There are two ways of powering the robot:

- Wall electrical outlet
- Battery power

Electrical outlet - You can power your robot using a standard 110/220 V power outlet by plugging the power cord (EH 0300 0001 (USA), EH 2500 0001 (EUR), EH 2500 0002 (AUS), EH 2500 0003 (UK)) into the Power Supply Unit (PSU - AE 0000 0029) on one end and into a power outlet on the other. Then plug the PSU into the base controller power connector.

Battery power - You can use the battery power cable (EH 01M8 0003) by plugging one end into the base controller power connector and connecting the four wires at the other end to a 24V battery. The following table shows the relationships between power connector pinout, the signal, and the wire color.

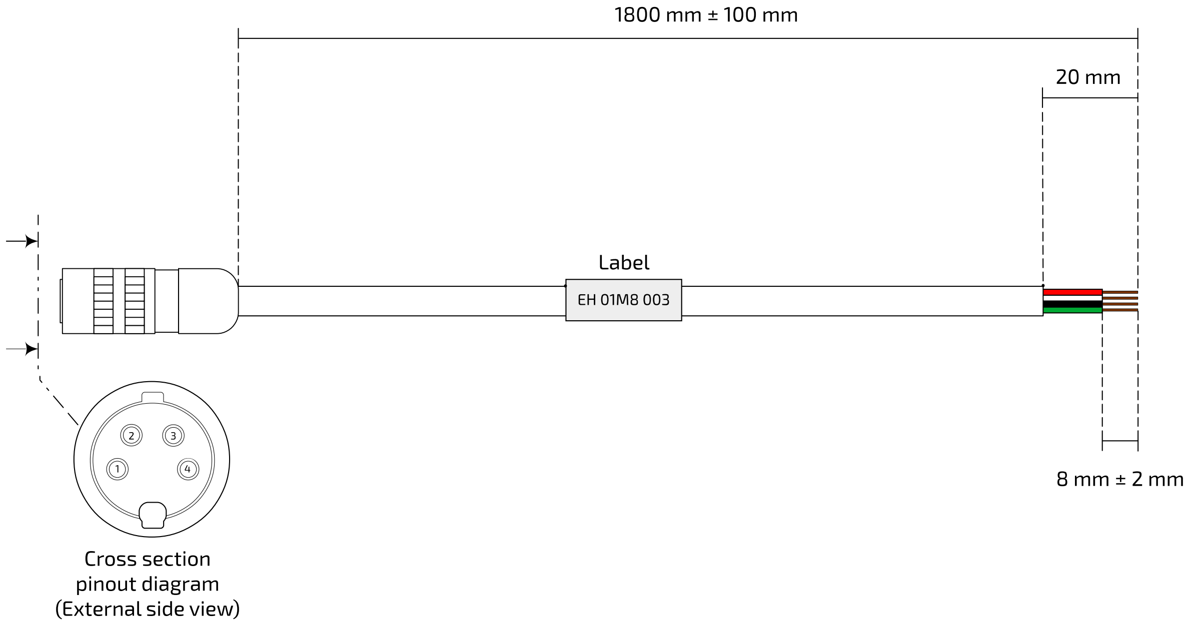


Figure 4: EH 01M8 003 battery power cable

Table 2: Battery power cable pinout

Pinout table		
Pin #	**Signal**	**Wire color**
1	24V	Red
2	24V	White
3	GND	Black
4	GND	Green or blue

⚠ Make sure that your battery respects the electrical specifications of the robot.

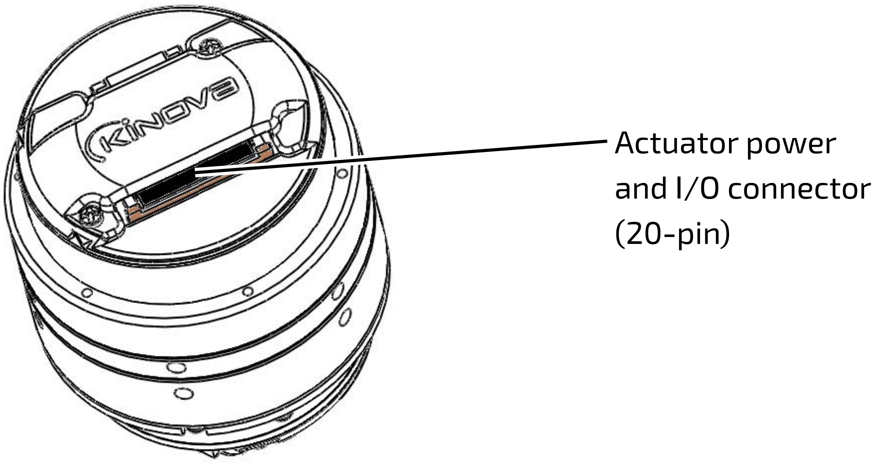
Integrating a new tool (optional)

This section describes how to integrate a new end effector with Kinova actuators, whether in a Kinova robot or custom application.

The small actuators (KA-58), whether they are incorporated at the last joint on a Kinova robot, or used standalone as part of a custom-built robotics application, have the ability to be connected to different types of tools. These actuators are designed to connect easily with Kinova's KG-Series grippers, but also have the ability to integrate with 3rd party end effectors.

Note: To add a new end effector, you will first need to remove the Kinova gripper that comes with the robot.

The output end of each actuator has a 20-pin power and I/O connector. Two of these pins are set aside as dedicated expansion communication and power lines. This allows you the option to connect an additional device to the end of a robot or the end of a chain of actuators in a custom-built set-up.

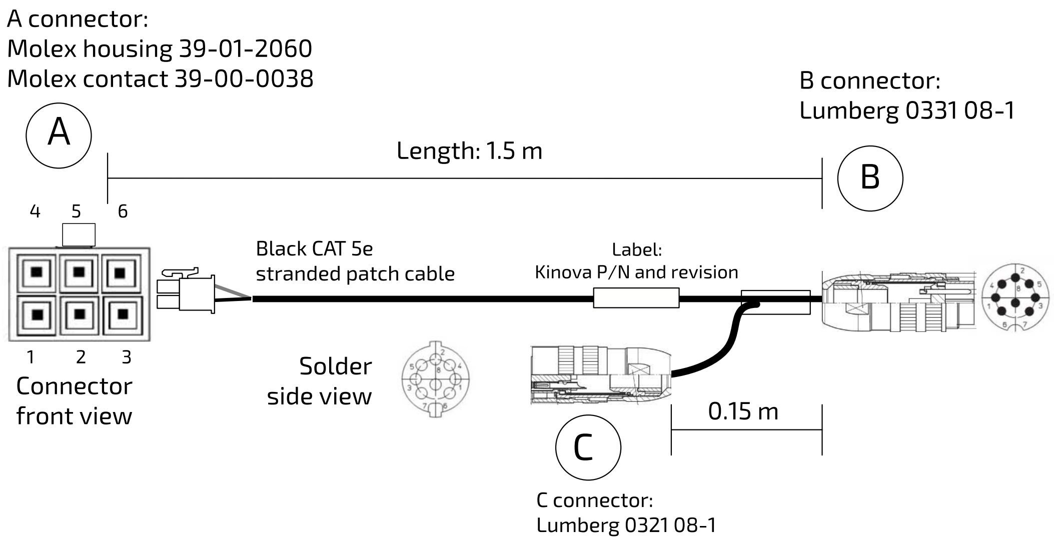


The pins and their function are described in the table below:

Table 3: KA-58 actuator pinout

Pin	Signal
1 to 8	24V input
9 to 16	GND
17	RS-485 low
18	RS-485 high
19	Expansion 0
20	Expansion 1

⚠ Make sure to connect your end effector using only pins 19 and/or 20. Using other pins could severely damage your robot. These two expansion lines are accessible for power and controls purposes via pins at the joystick port. A 'Y' cable is supplied with the robot to allow you to access both the joystick and the expansion lines.



The output pinout of the two expansion lines is indicated in the table below.

Table 4: EH 01M5 0001 pinouts

Connector A		
Pin	**Signal**	**Function**
1	COM1	RS485_low
2	GND	GND
3	COM3	Expansion 0
4	COM2	RS485_high
5	24V (max current: 1.5A)	24V
6	COM4	Expansion 1

Controls integration

This section describes the controls integration for Kinova robotic applications, whether for Kinova robots or custom-built applications.

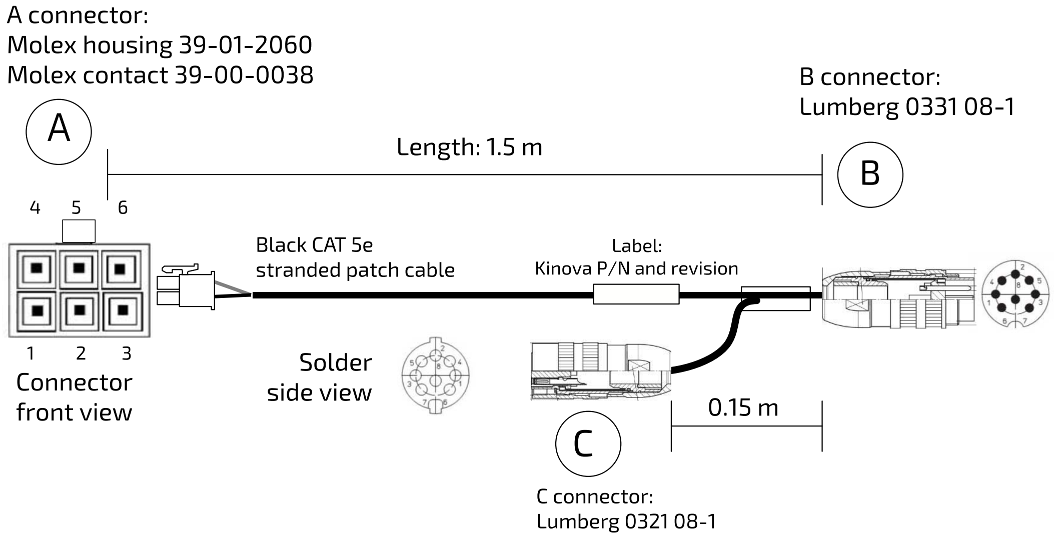
Once the mechanical and electrical integration are completed are completed (as well as the end effector integration if applicable), you can power on the robot by flipping the power switch on the back of the controller to ON. To control the robot, you can use either the Kinova API or the Kinova joystick.

API - Connect your computer to the robot using either the USB cable supplied with your package or an Ethernet cable. Connect one end of the cable to the robot and the other end to the development computer.

Install and open the KINOVA® *Software development kit* (SDK) on the development computer and follow the procedure and documentation included in the SDK.

Note: Refer to the *Software development kit* user guide and the Kinova API documentation for more guidance on controlling the robot via the API.

Kinova Joystick - Connect the joystick directly to the joystick port or to the C connector as shown in the diagram below if you are using the 'Y' cable.



Refer to the **Kinova joystick controller** section in the user guide for all the details regarding the use of the joystick.

Operating / controlling the robot

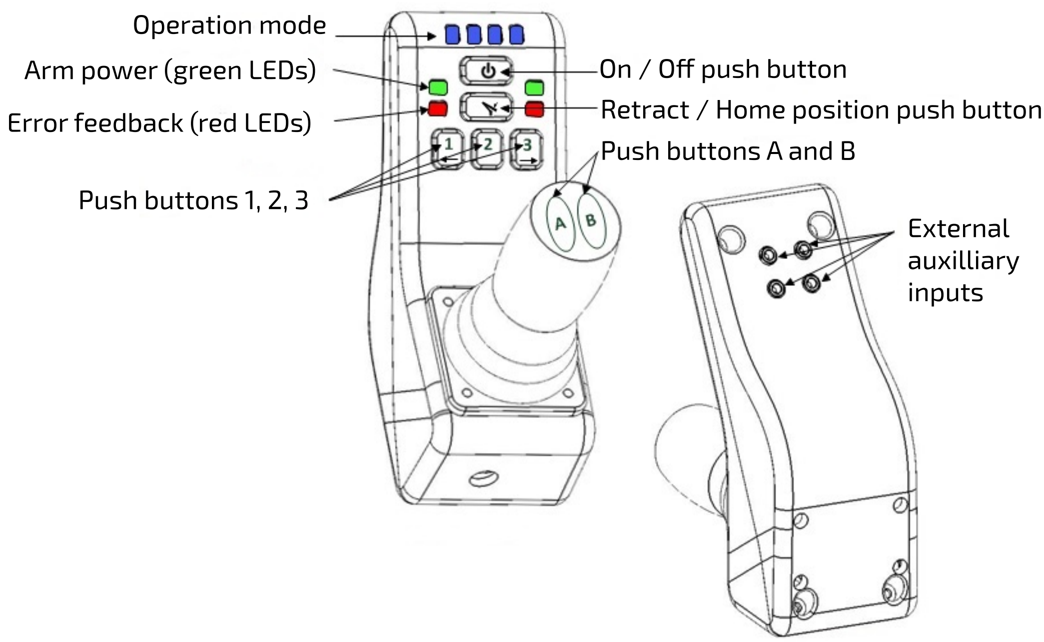
Three options are available to control Kinova robots. Two of these options use the API. The third option is the joystick control. The three options are:

- **(Kinova) joystick control**: This controls the arm in Cartesian velocity (by default as soon as the robot reaches its READY position) or in Angular velocity (if angular control is activated).
- **Kinova software control:** Kinova provides two different software control panels that allow you to control the arm via a graphical user interface.: the *Development Center* and the *Torque Console*. These two software panels allow users to command the arm in position, velocity, and trajectories. It also allows users to activate admittance control (inside the *Development Center*) and direct torque control/ force control (inside the Torque Console). For more details on Kinova software, please see the *Software development kit* User Guide (downloadable from the Kinova website).
- **API control:** Kinova has a library of C++ functions to control its robots. This library of functions is referred to as the Kinova API. The API (.dll files and .h files) is downloadable from Kinova's website as part of the KINOVA® *Software development kit* (SDK). The SDK is supplemented by HTML-based documentation detailing all the available functions. The Kinova API is supported on both Windows and Ubuntu. Kinova also offers developers the possibility for developers to control the robot through a ROS interface. For more information, see the Kinova ROS Github page at https://github.com/Kinovarobotics/kinova-ros.

A fourth option is to control the arm by directly controlling each actuator through applying internal or external forces / torques. The robot becomes reactive to direct control in **admittance control** and direct **torque/force control**.

Kinova joystick controller

The Kinova standard controller is a three-axis (left/right, forward/back, and twist) joystick mounted on a support. The controller includes five independent push buttons and four external auxiliary inputs (on the back side).



Joystick movements and modes

The Kinova joystick allows you to control the robot in a **2-axis** or **3-axis** mode. 2-axis mode will disable the joystick twist rotation.

The following table shows the button default factory settings for using the joystick in a 2-axis and 3-axis modes.

Table 5: Default joystick button settings

Buttons	One click	Hold 2 sec (Hold until position is reached)
⏻	Deactivate / Activate Joystick	Change joystick operating mode (2-axis Vs 3-axis)
	---	Home / Retracted function
	3-Axis	
1	Deactivate / Activate Drinking mode	---
2	---	Set Position
3	---	Go to pre-set position
A	Reach Finger mode	Decrease speed
B	Reach Translation & Wrist mode	Increase speed
Ext1	Reach Finger mode	Decrease speed
Ext2	Reach Translation and Wrist mode	Increase speed
Ext3	--	Home / Retracted function
Ext4	Deactivate / Activate Drinking mode	--
	2-Axis	
1	Deactivate / Activate Drinking mode	---
2	Reach Wrist orientation & Finger mode	Decrease speed
3	Reach Translation-X/Y & Translation-Z / Wrist rotation mode	Increase speed
A	---	---
B	---	---
Ext1	Reach Wrist orientation & Finger mode	Decrease speed
Ext2	Reach Translation-X/Y & Translation-Z / Wrist rotation mode	Increase speed
Ext3	--	Home / Retracted function
Ext4	Deactivate / Activate Drinking mode	--

Operating principles and Cartesian mode

This section describes at a high level the control of the robot using the joystick in Cartesian mode.

Operating principles

The operating principles are very simple and intuitive. The robot may be operated through several controllers. The following sections present the general control principles through Kinova's joystick.

Basic movements

The normal control of the robot with the joystick is said to be Cartesian. The user commands the tool translations (position variations) with respect to the base and the rotations (orientation variations) around the tool reference frame. The different joints are piloted automatically following the given command.

In **Translation mode**, the user controls the position of the gripper / tool in space. The gripper will always keep the same orientation with respect to the robot base.

- Translation X refers to left/right movements of the gripper
- Translation Y refers to front/back movements of the gripper
- Translation Z refers to up/down movements of the gripper

In **Wrist mode**, the user controls the position of the gripper around its center point (reference point) which will not move (or move slightly) when operating in this mode. Lateral orientation refers to a thumb/index circular movement of the wrist around the reference point. Vertical orientation refers to a top/bottom circular movement of the wrist around the reference point. Wrist rotation refers to a circular movement of the gripper around itself.

Drinking mode is to be used with the wrist rotation only. While operating the robot in **drinking mode**, the reference point (normally set in the middle of the gripper), is offset in height and length to produce a rotation around another point in the space of the robot.

Note: The offset is meant to shift the rotation center for wrist rotation from the center of the gripper to a point corresponding to the rim of a typical sized drinking glass held by the gripper. This way, the wrist will rotate, but around a point on the rim of a virtual glass, aided by movement of the elbow of the robot arm. This is similar to the way we coordinate an adjusted rotation of our wrist with movement of our forearm and elbow while drinking with a glass so that the glass tips its rim toward the mouth to pour. This mode is primarily intended for Assistive market users who use the robot to compensate for upper limb movement limitations. However, the mode is available to research users as well and can be useful in applications where the robot needs to pick up and pour from a small container of liquid.

In **Finger mode**, the user controls the opening and closing of the fingers.

Note: The robot will sometimes respond differently to a given command than described in this section. This may be due to the singularity and collision avoidance algorithms embedded in the kinematics. It is a normal protective behaviour of the robot and is position dependent. Both these avoidance algorithms can be deactivated by the user.

Home / Retract positions

This section describes the Home and Retracted positions of the robotic robot.

The robot comes with two configurable factory default pre-set positions:

- Home position and
- Retract position.

Home and retract positions can be configured using the KINOVA® *Development Center* utility.

The Home position refers to the position of the robot when it is ready to be used. In the Home position, the robot is awaiting commands from a control device.

The Retract position refers to the position of the robot when it is not used. The user should always place the robot in the Retract position when it is unused as it decreases the physical volume occupied by the robot. In the Retract position, the robot is in standby mode; control device features are disabled and power consumption is much lower.

⚠ Never use the Home / Retract function when carrying liquid. The Home position is pre-set and if the wrist rotates, it may spill some liquid.

Operating the robot via joystick

This section describes operation of the robot using the joystick.

This section explains how to operate the robot with factory configuration. Contact your reseller for operation instructions in the case of an adapted configuration.

⚠ Before operating the robot, please make sure it is properly installed.

⚠ Do not manipulate cutting, very sharp or any dangerous tools or objects with the robot.

⚠ This equipment is not designed to act as a lift.

⚠ This equipment is not designed to be used in presence of flammable mixture. (Not AP or APG rated).

⚠ Do not install the robot near any heat sources, such as radiators. Do not use it to directly manipulate hot objects.

Joystick control quick start

This section describes how to get started using the Kinova joystick to control the robot in the default configuration.

Before you begin

The joystick needs to be connected to the robot.

About this task

This procedure provides a quick hands-on walkthrough of controlling the robot with the joystick.

Procedure

1. Turn ON the device by pushing the ON/OFF switch located on the robot base.
2. Wait until the green lights on the controller stop flashing.
3. Put the robot in its Home position by holding down the HOME/RETRACTED button until the robot stops moving. The robot will slowly reach the Home position.

   Note: When starting the robot, you are in 3-axis operation mode, **Translation control mode**, meaning that any movement of the joystick will move the center of the gripper parallel to the floor.
4. You may move the 3 axes of the joystick to try out Translation control mode.

   Note: To change the operating mode of the Joystick, hold the ON/OFF button for 2 seconds. At this point, you are in 2-axis mode and the stick rotation is deactivated.
5. One press of Button B will bring you in **Wrist control mode** meaning that any movement of the joystick will result in a rotation of the gripper around its center.

   Note: Another press of Button B will bring you back in Translation control mode.
6. One press of Button 1 will activate **Drinking mode** which may be used only in Wrist mode. When rotating the joystick lever, you will see that the robot's wrist rotation now compensates for the height and radius of a virtual glass. This movement is ideal when trying to use th robot to drink directly from a glass.

   Note: Another press of Button 1 will disable Drinking mode.
7. One press of Button A will bring you in **Finger control mode**. The fingers will move per a left/right inclination of the joystick.

   Note: At any time, you may use the Home / Retracted button until the robot stops moving to bring it back to its Home position.

   Note: If you hold the Home / Retracted button again, the robot will start to move toward the Retracted position.
8. Hold the On/Off Button for 2 seconds to change the operating mode. This will disable the stick rotation. You are now in a 2-Axis Translation control mode. Stick rotation won't have any effect and you will only be able to control the horizontal translation of the robot (X- and Y- axis).
9. One press of Button 3 will bring you to control the vertical translation of the gripper (Translation-Z) and Wrist rotation.

   Note: Another hit on Button 3 will bring you back in Translation-X and Translation-Y control mode.
10. One press of Button 1 will activate Drinking mode again
11. One press of Button 2 will bring you to control the wrist orientation (Lateral orientation and Vertical orientation).
12. Another press of Button 2 will bring you to Finger control mode. The fingers will move according to a left/right inclination of the joystick.

Note: Another press of Button 2 will bring you back in Lateral orientation and Vertical orientation control mode.

Default joystick motion settings - Cartesian three-axis mode

This section describes default motion settings in Cartesian three-axis mode.

Table 6: Three-axis mode joystick controls

Joystick movement	Robot movement	Availability
Translation Mode		
Incline FRONT	Gripper moves forward	4 / 6 / 6S / 7S DoF
Incline BACK	Gripper moves backward	4 / 6 / 6S / 7S DoF
Incline LEFT	Gripper moves left	4 / 6 / 6S / 7S DoF
Incline RIGHT	Gripper moves right	4 / 6 / 6S / 7S DoF
Rotate stick CLOCKWISE	Gripper moves up	4 / 6 / 6S / 7S DoF
Rotate stick COUNTERCLOCKWISE	Gripper moves down	4 / 6 / 6S / 7S DoF
Wrist Mode		
Incline FRONT	Vertical orientation – top side	6 / 6S / 7S DoF
Incline BACK	Vertical orientation – bottom side	6 / 6S / 7S DoF
Incline LEFT	Lateral orientation – thumb side	6 / 6S / 7S DoF
Incline RIGHT	Lateral orientation – index side	6 / 6S / 7S DoF
Rotate stick CLOCKWISE	Wrist rotation clockwise	4 / 6 / 6S / 7S DoF
Rotate stick COUNTERCLOCKWISE	Wrist rotation counterclockwise	4 / 6 / 6S / 7S DoF
Finger Mode		
Incline LEFT	Close Fingers (3-finger mode)	4 / 6 / 6S / 7S DoF
Incline RIGHT	Open Fingers (3-finger mode)	4 / 6 / 6S / 7S DoF
Incline FRONT	Open Fingers (2-finger mode)	6S/ 7S DoF
Incline BACK	Close Fingers (2-finger mode)	6S / 7S DoF

Default joystick motion settings - Cartesian two-axis mode

This section describes default motion settings in Cartesian two-axis mode.

Table 7: Two-axis mode joystick controls

Joystick movement	Robot movement	Availability
Translation-X and Translation-Y		
Incline FRONT	Gripper moves forward	4 / 6 / 6S / 7S DoF
Incline BACK	Gripper moves backward	4 / 6 / 6S / 7S DoF
Incline LEFT	Gripper moves left	4 / 6 / 6S / 7S DoF
Incline RIGHT	Gripper moves right	4 / 6 / 6S / 7S DoF
Translation-Z and Wrist Rotation		
Incline FRONT	Gripper moves up	4 / 6 / 6S / 7S DoF
Incline BACK	Gripper moves down	4 / 6 / 6S / 7S DoF
Incline LEFT	Wrist rotation clockwise	4 / 6 / 6S / 7S DoF
Incline RIGHT	Wrist rotation counter-clockwise	4 / 6 / 6S / 7S DoF
Wrist Orientation		
Incline FRONT	Vertical orientation – Top side	6 / 6S / 7S DoF
Incline BACK	Vertical orientation – Bottom side	6 / 6S / 7S DoF
Incline LEFT	Lateral orientation – Thumb side	6 / 6S / 7S DoF
Incline RIGHT	Lateral orientation – Index side	6 / 6S / 7S DoF
Finger Mode		
Incline LEFT	Close Fingers (3 finger mode)	4 / 6 / 6S / 7S DoF
Incline RIGHT	Open Fingers (3 finger mode)	4 / 6 / 6S / 7S DoF
Incline FRONT	Open Fingers (2 finger mode)	6S / 7S DoF
Incline BACK	Close Fingers (2 finger mode)	6S / 7S DoF

Controlling the robot in angular mode

In angular mode, the user commands each individual actuator rotation. Angular mode must be activated using the *Development Center* application or the Kinova API.

Multiple joints can be commanded simultaneously using the joystick (when diagonal control is enabled). The joystick default settings in angular mode are summarized in the following table.

Table 8: Angular mode joystick controls

Joystick movement	Arm movement	Availability
Translation mode		
Incline LEFT	Joint 1 rotates positively (angle increases)	6S /7S DoF
Incline RIGHT	Joint 1 rotates negatively (angle decreases)	6S /7S DoF
Incline FRONT	Joint 2 rotates negatively (angle decreases)	6S /7S DoF
Incline BACK	Joint 2 rotates positively (angle increases)	6S /7S DoF
Rotate CLOCKWISE	Joint 3 rotates positively (angle increases)	6S /7S DoF
Rotate stick COUNTER-CLOCKWISE	Joint 3 rotates negatively (angle decreases)	6S /7S DoF
Wrist mode		
Incline LEFT	Joint 4 rotates negatively (angle decreases)	6S /7S DoF
Incline RIGHT	Joint 4 rotates positively (angle increases)	6S /7S DoF
Incline FRONT	Joint 5 rotates positively (angle increases)	6S /7S DoF
Incline BACK	Joint 5 rotates negatively (angle decreases)	6S /7S DoF
Rotate CLOCKWISE	Joint 6 rotates positively (angle increases)	6S /7S DoF
Rotate stick COUNTER-CLOCKWISE	Joint 6 rotates negatively (angle decreases)	6S /7S DoF
Finger mode		
Incline LEFT	Close Fingers (3 finger mode)	6S /7S DoF
Incline RIGHT	Open Fingers (3 finger mode)	6S /7S DoF
Incline FRONT	Open Fingers (2 finger mode)	6S /7S DoF
Incline BACK	Close Fingers (2 finger mode)	6S /7S DoF
Rotate CLOCKWISE	Joint 7 rotates positively (angle increases)	7S DoF
Rotate stick COUNTER-CLOCKWISE	Joint 7 rotates negatively (angle decreases)	7S DoF

Joystick LED feedback

The Kinova joystick offers visual feedback:

- Blue LEDs: control mode
- Green LEDs: robot power
- Red LEDs: error condition(s)

Joystick blue LED feedback

This section describes the blue LED feedback on the Kinova controller.

The blue LEDs on the controller give feedback on the current control mode. The interpretation of the blue mode LED indicators is described in the following table.

Table 9: Control mode feedback

Blue LED indication		Control Mode
3-Axis	■□□□	Translation (X-Y-Z)
	□■□□	Wrist
	□□■□	Fingers
	□■□■	Drinking mode (to be used with wrist rotation mode)
	□□□□	Disabled controller
2-Axis	■□□□	Translation (X-Y)
	■■□□	Translation (Z) / Wrist Rotation
	□■□□	Wrist Orientation
	□□■□	Fingers
	□■□■	Drinking mode (to be used with wrist rotation mode)
	□□□□	Disabled controller

When no blue lights are visible, the controller is disabled. To enable the controller, you must either proceed with the following options:

- The On / Off button must be depressed.
- The robot must be set in its HOME position by holding the HOME/RETRACTED function until the robot stops moving.

Joystick green LED feedback

This section describes the green LED feedback on the Kinova joystick.

The green lights offer visual feedback on the power status of the robot.

Table 10: Power status feedback

Green LED indication	Power Status
Flashing	The internal communication is still synchronizing after the robot has been turned on. It is not yet ready to use.
Solid	The robot is powered and ready to use.

Joystick red LED feedback

This section describes the red LED feedback on the Kinova controller.

The red lights offer visual feedback on possible errors that may occur while operating the robot:

Table 11: Error status feedback

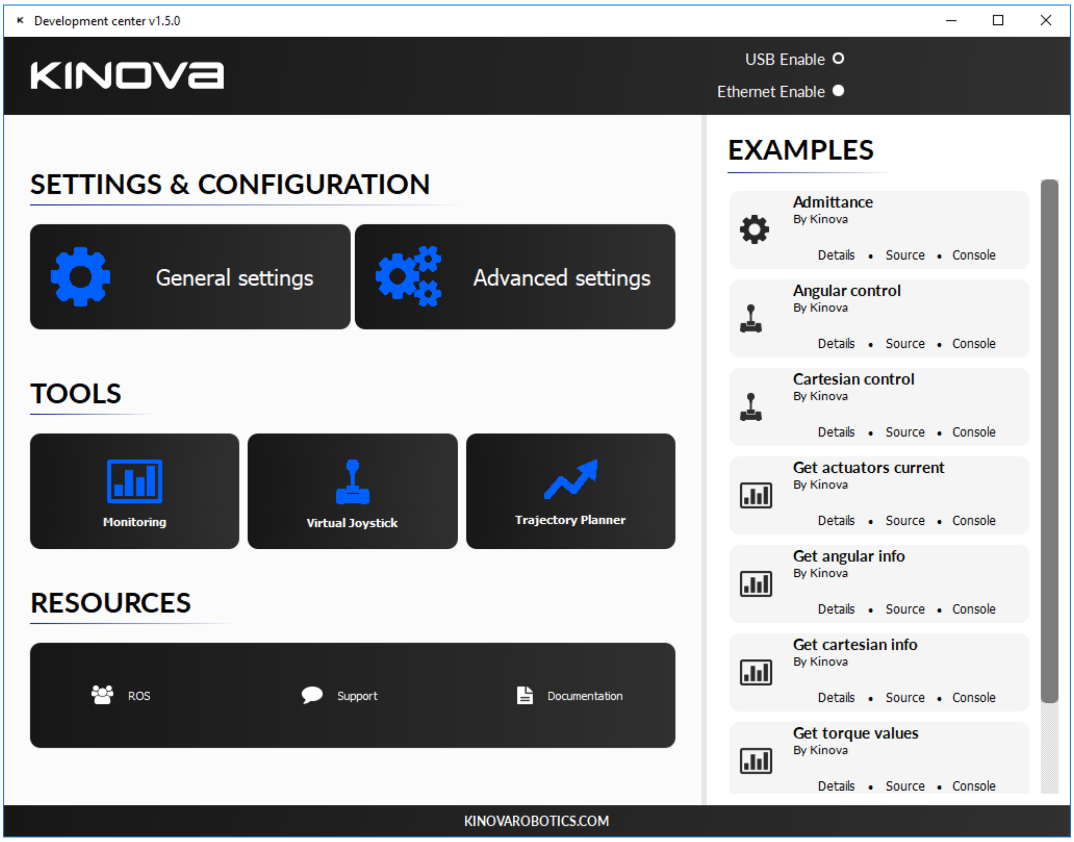
Red LED indication	Cause of the Error Status	Action to resolve the situation
Flashing	The weight being lifted is too heavy, or too much force is applied on the robot.	Safely put down the object, or release force applied on the robot, and wait until red lights turn off.
	The temperature of a section of the robot is too high.	The usage of the robot is excessive and doesn't respect the normal use definition. Safely put down any object that is in the gripper, bring back the robot to its RETRACTED position, and wait until the red lights turn off.
	The input voltage to the robot (or batteries) is too low or too high.	Safely put down any object that is in the gripper, bring back the robot to its RETRACTED position. Ensure the power supply is appropriate, connections are secure and batteries are charged properly before using the robot again.
Solid	The robot is in a fault mode	Turn off the robot and turn it back on. If the problem remains, contact your distributor or Kinova.

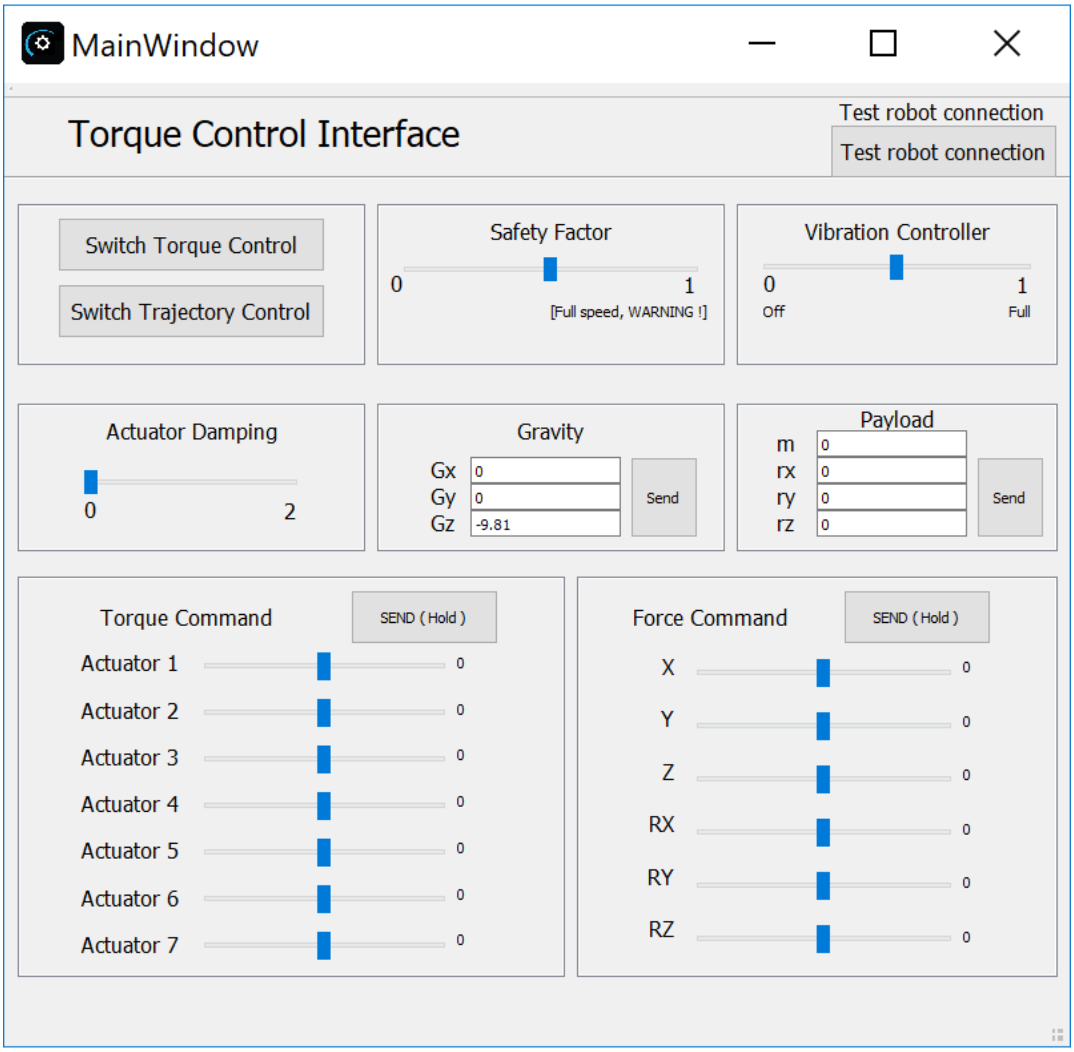
Controlling the robot using Kinova software

This section describes at a high level control of the robot using Kinova software.

Multiple functionalities are offered by Kinova software. Using the *Development Center* and the *Torque Console*, users can (among other things) send trajectories, monitor their robot's state, activate admittance and switch between Cartesian and angular control. Kinova software is also useful to update the robot's firmware and to diagnose different problems.

The *Development Center* and *Torque Console* are available for download on the Kinova website. For a complete list of Kinova software functionalities and use instructions, please see the separate KINOVA® *Software development kit* User Guide.


Development center v1.5.0
KINOVA
USB Enable
Ethernet Enable
SETTINGS & CONFIGURATION
General settings
Advanced settings
TOOLS
Monitoring
Virtual Joystick
Trajectory Planner
RESOURCES
ROS
Support
Documentation
EXAMPLES
Admittance
By Kinova
Details • Source • Console
Angular control
By Kinova
Details • Source • Console
Cartesian control
By Kinova
Details • Source • Console
Get actuators current
By Kinova
Details • Source • Console
Get angular info
By Kinova
Details • Source • Console
Get cartesian info
By Kinova
Details • Source • Console
Get torque values
By Kinova
Details • Source • Console
KINOVAROBOTICS.COM


MainWindow
Torque Control Interface
Test robot connection
Test robot connection
Switch Torque Control
Switch Trajectory Control
Safety Factor
0
1
[Full speed, WARNING !]
Vibration Controller
0
1
Off
Full
Actuator Damping
0
2
Gravity
Gx 0
Gy 0
Gz -9.81
Send
Payload
m 0
rx 0
ry 0
rz 0
Send
Torque Command
SEND (Hold)
Actuator 1 0
Actuator 2 0
Actuator 3 0
Actuator 4 0
Actuator 5 0
Actuator 6 0
Actuator 7 0
Force Command
SEND (Hold)
X 0
Y 0
Z 0
RX 0
RY 0
RZ 0

Controlling the arm using the Kinova API

This section describes at a high level control of the arm using Kinova API.

As with the *Development Center* and the *Torque Console*, information about the API is downloadable from the Kinova website. The *Development Center* comes with intergrated HTML documentation describing the C++ programming functions that can be used to access the API.

A good way to start with the API is to look at the examples provided with the Kinova *Development Center* application. For more details, please see the KINOVA® *Software development kit* User Guide and the incorporated HTML-based API documentation.

Robot control

Control modes overview

This section explains how to operate the robot with factory configuraton.

Kinova robot actuators can be controlled by end effector position, actuators angular position or actuators torque. Kinova robots offer the following control mode options:

Table 12: Control modes

Control mode	Description
Cartesian position	Specify end-effector position and end-effector (tool) orientation (Euler angles, X-Y'-Z'' convention) in the base frame.
Cartesian velocity	Specify end-effector translation velocities in the base frame and end-effector (tool) rotation velocities in the end effector frame.
Angular position	Specify each actuator angle.
Angular velocity	Specify angular (rotational) velocity for each actuator.
Cartesian admittance (Reactive Force control in Cartesian space)	Apply forces and torques on the end-effector and get a Cartesian motion (translation/rotation) in the appropriate direction.
Angular admittance (Reactive Force control in joint space)	Apply torques on actuators and get an angular motion (joint rotation) in the appropriate direction.
Direct torque control	Specify torque for each actuator. By default, each actuator receives its corresponding gravity torque so the robot compensates its own weight.
Force control	Specify forces and torques at the end-effector (tool). The robot automatically computes the torque at each actuator required to generate the appropriate forces/torques at the end-effector.

Control features

Singularity Avoidance

This section discusses singularity-avoidance in the robot.

In Cartesian mode, there are some configurations in which the robot loses one or more degrees of freedom (meaning the robot is not able to move in one direction or the other. These configurations are called singularities and Kinova robots avoid them automatically in Cartesian mode. This means that Cartesian commands sent by the user may be modified somewhat to avoid a singularity.

Singularity auto-avoidance behavior can be deactivated using the `ActivateAutomaticSingularityAvoidance()` API function, setting its parameter to false.

Table 13: 6 DoF Spherical Singularities

Singularity	Explanation	Robot Behavior
Boundary singularity 	The arm is at full reach. It cannot move anymore in the direction it is currently reaching out.	Not possible to bring elbow at 180° in Cartesian mode.
Wrist-over-base singularity 	The wrist point is aligned with the first joint axis. Joint 1 cannot move the hand in translation anymore.	Not possible to bring the wrist inside a virtual cylinder located around the base. The virtual cylinder has a 15 cm radius approximately.
Wrist alignment singularity 	Joint 4 and 6 are aligned and have the same effect. The hand cannot rotate in one direction anymore.	Executed motion when the wrist is near alignment configuration might be a little different from the commanded motion. But the robot is essentially able to go 'though' this singular configuration.

Table 14: 7 DoF Spherical Singularities

Singularity	Explanation	Robot Behavior
Boundary singularity 	The arm is at full reach. Joint 4 is at 180°. The arm cannot move in the direction it is currently reaching out.	When singularity avoidance is activated, it's not possible to bring the elbow at 180° in Cartesian mode.
Joints 2 and 3 singularity 	Joint 2 is at 180° so joints 1 and 3 are perfectly aligned and have the same effect. Joint 3 is at 90° or at 270° so joint 2 and joint 4's axis is perpendicular. The robot cannot move purely along an axis in translation anymore.	When singularity avoidance is activated, it's not possible to bring joint 3 near 90° or 270° when joint 2 is near 180° (or vice versa, to bring joint 2 near 180° when joint 3 is near 90° or 270°) in Cartesian mode. Besides, the fitness function will try to avoid the singularity by moving joint 2 away from 180° and joint 3 away from 90° or 270° while moving in the robot's null space. For more information on the fitness function, please see section "Null space motion."
Joints 2 and 6 singularity 	Joint 2 is at 180° so joints 1 and 3 are perfectly aligned and have the same effect. Joint 6 is at 180° so joints 5 and 7 are perfectly aligned and have the same effect. The hand cannot rotate in one direction anymore.	When singularity avoidance is activated, it's not possible to bring joint 2 near 180° when joint 6 is near 180° in Cartesian mode. Besides, the fitness function will try to avoid the singularity by moving joints 2 and 6 away from 180° while moving in the robot's null space. For more information on the fitness function, please see section "Null space motion."

Singularity	Explanation	Robot Behavior
Joints 5 and 6 singularity 	Joint 6 is at 180° so joints 5 and 7 are perfectly aligned and have the same effect. Joint 5 is at 90° or at 270° so joint 4 and joint 6's axis is perpendicular. The robot can't complete pure rotations around an axis.	When singularity avoidance is activated, it's not possible to bring joint 5 near 90° or 270° when joint 6 is near 180° in Cartesian mode. Besides, the fitness function will try to avoid the singularity by moving joint 5 away from 90° or 270° and joint 6 away from 180° while moving in the robot's null space. For more information on the fitness function, please see section "Null space motion."

Self-collisions auto-avoidance

In Cartesian mode, the robot avoids self-collisions (essentially collisions between the gripper and the base but also between the gripper and the arm segment). This means the Cartesian command sent by the user may be modified by the control software to avoid a self-collision. Singularity auto-avoidance behavior can be deactivated using the `ActivateAutomaticCollsionAvoidance()` API function, setting its parameter to false.

Please note that collisions with other objects present in the environment are not automatically handled by the robot.

⚠ Self-collisions are **not** automatically avoided during angular control. Only joint limits are handled.

7 DoF Spherical Null space motion

The 7 DoF-S robot is redundant because it only requires six degrees of freedom (=six actuators) to move and orient its effector in 3D space although it has seven actuators. A robot with six actuators can only reach a given end-effector position and orientation with a few configurations, which are very different from one another. The 7th degree of freedom (or 7th actuator) gives more flexibility to the robot and lets it reach a given end-effector position and orientation with a multiple/infinite number of configurations.

In fact, the robot can even move its elbow (joint 4) without modifying its end-effector position and orientation. This type of motion is called "null space motion" because it does not affect where the end-effector is (i.e. it has no effect in the task space). During Cartesian control, Kinova automatically optimizes the robot motion in its null space with a special optimisation function called the fitness function. The fitness function will try to find the best compromise to avoid singularities, position the elbow and avoid angular limits of the 7 DoF-S arm without modifying the user's Cartesian command. Right now, the robot's null space motion can be deactivated, but the fitness function parameters cannot be modified.

⚠ Because of null space motion, the robot elbow can move quite a bit during Cartesian control. This is because the robot is trying to avoid singularities and angle limits while trying to keep a preferred angle position and without modifying the user Cartesian command.

⚠ If the currently implemented null space motion algorithm does not suit your needs, you can control the robot in angular space or deactivate the null space motion using the API function.

Protection zones

Protection zones can be defined using the API. Protection zones must be box-shaped. Once a protection zone is defined, the robot will avoid getting inside this zone. If possible, it will slide on the zone. If not, it will stop. By default, a protection zone is defined in reference to the robot.

⚠ If you give a command and the robot stops moving because it is too close to a protection zone, try moving the robot in another direction. If you notice that your robot is not moving under any command, your robot's behaviour is abnormal. Try sending the arm to its Ready position (to continue using it) and contact Kinova support (to report the problem).

⚠ Protection zones are not avoided during angular control. Only joint limits are handled in angular control.

Rotating frame / Fixed frame

This section describes the difference between Rotating frame and Fixed frame for the hand/gripper.

By default, the robot is configured in **Rotating frame**. In Rotating frame, the gripper orientation changes automatically to follow the arm motion in the horizontal plane. This gives a more human-like behavior.

In **Fixed frame**, the gripper orientation will not change unless explicitly commanded.

Users can switch between Rotating frame and Fixed frame using the *Development Center* software or API. Frame selection (Rotating or Fixed) is recorded inside the robot's memory. This means that a robot that was configured in Fixed frame will not reset to Rotating frame at reboot.

Usable workspace

This section describes the usable workspace of the robot.

The robot's usable workspace is represented in the graphic below. The workspace provided is for angular control.

When the robot is in Cartesian mode, its effective usable workspace will be slightly smaller because of singularities, self-collisions and protection zone automatic avoidance algorithms.

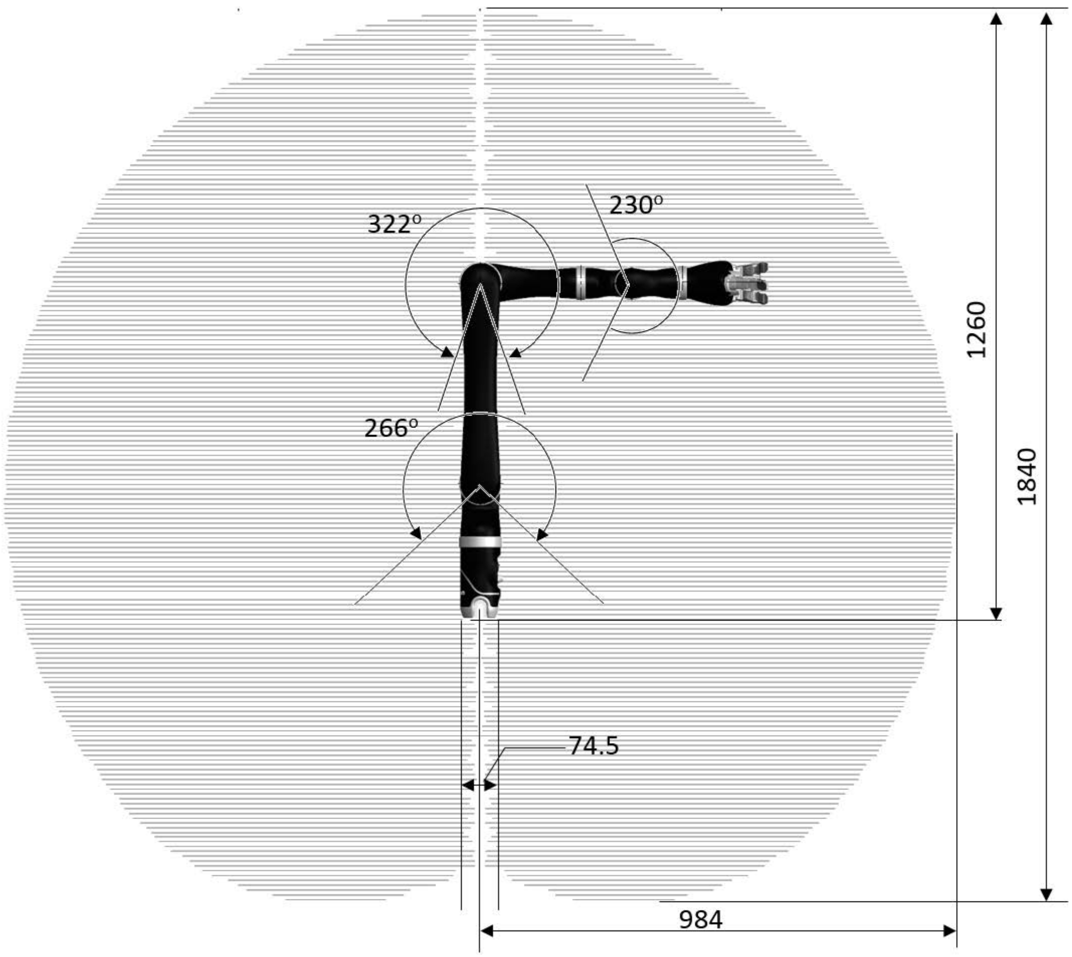


Figure 5: Usable workspace - dimensions in mm. 6 DoF shown for illustrative purposes

Admittance control

This section describes admittance control both in Cartesian and in angular mode.

Admittance control (also called **Reactive Force control**) can be activated and deactivated using the KINOVA® *Development Center* software or API. When admittance control is active, it becomes possible to move the robot by hand. There are two types of admittance control available:

- Cartesian admittance
- angular / joint admittance

For **Cartesian admittance**, the robot must be in Cartesian mode and admittance must be activated.

For **angular/joint admittance**, the robot must be in angular mode and admittance must be activated.

During Cartesian admittance control, the robot continues to avoid singularities, self collisions and protection zones automatically (unless these functions are deactivated using the API). The null space motion will also stay active (unless deactivated using the API). Finally, maximum velocity and acceleration limits stay active during admittance control in angular or Cartesian space.

User force input is automatically bounded between a minimum and a maximum value inside the code. These minimum and maximum values are configurable inside the API. Damping and inertia parameters present in the robot's admittance model are also configurable.

The table below shows default admittance parameters for the 7 DoF spherical configuration. Details will differ for other configurations.

Table 15: Admittance default parameters for 7 DoF Spherical configuration.

Cartesian admittance parameters	Angular admittance parameters
Damping_X=18.18 N/(m/s) Damping_Y=31.82 N/(m/s) Damping_Z=54.55 N/(m/s) Damping_ThetaX=0.85 Nm/(rad/s) Damping_ThetaY=0.85 Nm/(rad/s) Damping_ThetaZ=0.85 Nm/(rad/s) Inertia_X=6.36 N/(m/s2) Inertia_Y=11.14 N/(m/s2) Inertia_Z=19.10 N/(m/s2) Inertia_ThetaX=0.21 Nm/(rad/s2) Inertia_ThetaY=0.21 Nm/(rad/s2) Inertia_ThetaZ=0.21 Nm/(rad/s2) CartesianForceMin_X = 8.0N; CartesianForceMin_Y = 8.0N; CartesianForceMin_Z = 8.0N; CartesianForceMin_ThetaX = 1.5Nm CartesianForceMin_ThetaY = 1.5Nm CartesianForceMin_ThetaZ= 1.5Nm CartesianForceMax_X= 12.0N CartesianForceMax_Y = 15.0N CartesianForceMax_Z = 20.0N CartesianForceMax_ThetaX = 2.5Nm CartesianForceMax_ThetaY= 2.5Nm CartesianForceMax_ThetaZ = 2.5Nm	Damping_Joint1=3.95 Nm/(rad/s) Damping_Joint2=9.2 Nm/(rad/s) Damping_Joint3=3.95 Nm/(rad/s) Damping_Joint4=3.95 Nm/(rad/s) Damping_Joint5=0.5 Nm/(rad/s) Damping_Joint6=0.5 Nm/(rad/s) Damping_Joint7=0.5 Nm/(rad/s) Inertia_Joint1=0.711 Nm/(rad/s2) Inertia_Joint2=1.656 Nm/(rad/s2) Inertia_Joint3=0.711 Nm/(rad/s2) Inertia_Joint4=0.711 Nm/(rad/s2) Inertia_Joint5=0.09 Nm/(rad/s2) Inertia_Joint6=0.21 Nm/(rad/s^2) Inertia_Joint7=0.21 Nm/(rad/s2) AngularForceMin_Joint1 = 4.0Nm AngularForceMin_Joint2 = 7.0Nm AngularForceMin_Joint3 = 2.0Nm AngularForceMin_Joint4 = 2.0Nm AngularForceMin_Joint5 = 1.5Nm AngularForceMin_Joint6 = 1.5Nm AngularForceMin_Joint7= 1.5Nm AngularForceMax_Joint1= 7.0Nm AngularForceMax_Joint2 = 14.0Nm AngularForceMax_Joint3 = 5.0Nm AngularForceMax_Joint4 = 5.0Nm AngularForceMax_Joint5 = 2.0Nm AngularForceMax_Joint6 = 2.0Nm AngularForceMax_Joint7 = 2.0Nm

⚠ In Cartesian admittance mode, you should grab the robot by its hand when you interact with it. If you grab it by another link (e.g. grab the robot at elbow level), the admittance will still work, but the hand might rotate in an unusual way.

⚠ If you have the impression that the robot is moving unusually in admittance mode, please check your torque sensor calibration.

Torque Control

This section describes torque control.

As its name suggests, **torque control** lets the user control the actuators in torque rather than in position. By default, the arm compensates its own weight in torque mode. Users can also send torque commands and force commands using the KINOVA® *Torque Console* application or the Kinova robot API.

By default, the robot also includes some safety features. The first safety is that the arm will not switch from position mode to torque mode unless the torque read by the actuators and the gravity torques computed by the gravity model inside the robot are too different. On reboot, the arm will always go back to position mode.

Once in torque mode, some safety features can bring back the arm in position mode. This is the case when one of the motors' rotation velocity gets too high. Users can disable this safety feature at their own risk using the *Torque Console* or API (Safety Factor to 0). By default, the robot will also switch back to position mode if the hand gets too close from the base or if a specific joint gets too close from its angle limits. Users can disable the base collision avoidance safety feature at their own risk using the API.

⚠ Please note that when the robot is in torque mode, it is NOT possible to move the robot's fingers.

⚠ Do not disable the velocity safety feature unless you are sure that your robot's torque sensors are well calibrated.

⚠ Be very careful when you disable a torque control safety feature. Ideally, keep the robot in an open environment free of near potential obstacles. Please keep in mind that commanding your robot in torque mode with safeties disabled could damage your robot if the motors start turning too fast or if the robot collides with itself or the environment. If you disable the base collision avoidance safety, keep in mind that the robot could collide with itself. A lot of safety parameters are customizable in torque mode (see the HTML-based API documentation in the SDK for more details), but be aware that these customizations require knowledge to use appropriately.

⚠ Moving the robot very quickly in torque mode can lead, in rare occasions, to an unexpected reboot. The robot stops moving or switches back to position mode and waits for a Home/Ready command. If you observe this behaviour, please contact the Kinova Support team (see Contacting Support).

Improving robot behavior in Torque mode

This section describes how to improve behavior of the robot in Torque mode.

Kinova robots allow to compensate the gravity torques associated with their own weight. Two gravity compensation modes are available:

- Manual (default mode)
- Optimal (must be activated by user)

In **Manual mode,** each robot segment mass and center of mass is specified. For a robot with 6 motors, this makes 24 parameters. Users can change segment masses and centers of mass through the function `SetManualInputParam()`.

In **Optimal mode**, the robot computes gravity torques from a series of parameters linking the angle and torque readings. To find these parameters, the robot must perform an automatic trajectory. The accuracy of the gravity torques estimation with the Optimal mode is usually about three times better than with the Manual mode.

To use the Optimal gravity compensation mode, you need to create a custom C++ program calling three main functions from Kinova API:

1. Gravity estimation:
   a. `RunGravityZEstimationSequence()` - for 6 DoF Spherical
   b. `RunGravityZEstimationSequence7DOF()` - for 7 DoF Spherical
2. `SetOptimalZParam()`
3. `SetGravityType()`

The first function to be called is either:

- `RunGravityZEstimationSequence()` for the 6 DoF Spherical, or
- `RunGravityZEstimationSequence7DOF()` for the 7 DoF Spherical.

`RunGravityZEstimationSequence()` takes for input the robot type and an empty array of 16 floats, while `RunGravityZEstimationSequence7DOF()` takes the robot type and an empty array of 19 floats. This will compute the 16 (or 19) optimal gravity parameters and place them in the empty array of 16 (or 19) floats. `RunGravityZEstimationSequence()` will also write the parameters in a text file called `ParametersOptimal_Z.txt`. This file is found in the same folder as your C++ project. This function is not called every time you want to use the robot. Once the optimal parameters have been obtained, this function does not need to be called. Only call this function if you want to recalibrate the gravity parameters (which is the case if you tighten/untighten some screws, or if the robot seems to have an unusual behaviour after resetting the 'zero' of the torque sensors.

The second function to be called is `SetOptimalZParam()`, which takes for input the array of optimal parameters from the previous step. `SetOptimalZParam()` informs the robot on the new optimal parameters.

Note: This function must be called at every reboot.

When you call this function without previously calling `RunGravityZEstimationSequence()`, you must initialize the array of 16 floats with the values found in the text file `ParametersOptimal_Z.txt`.

The third function to be called is `SetGravityType()`, which takes for input either OPTIMAL (activates the Optimal gravity compensation mode) or MANUAL_INPUT (activates the MANUAL gravity compensation mode).

Note: This function must also be called at every reboot.

Optimal mode code example

This section gives an example of C++ code to use Optimal mode with the arm.

The following sample C++ program demonstrates how to use Optimal mode

```
#include <Windows.h>
#include <conio.h>
#include <iostream>
#include "Lib_Examples\CommandLayer.h"
#include "Lib_Examples\CommunicationLayerWindows.h"
#include "Lib_Examples\KinovaTypes.h"
#include <fstream>

using namespace std;

//A handle to the API.
HINSTANCE commandLayer_handle;

//Function pointers to the functions we need
int(*MyInitAPI)();
int(*MyCloseAPI)();
int(*MyGetAngularCommand)(AngularPosition &Response);

int(*MyRunGravityZEstimationSequence7DOF)(ROBOT_TYPE type, float OptimalzParam[OPTIMAL_Z_PARAM_SIZE_7DOF]);
int(*MySetGravityOptimalParameter)(float Command[GRAVITY_PARAM_SIZE]);
int(*MySetGravityType)(GRAVITY_TYPE Type);

int(*MyGetGlobalTrajectoryInfo)(TrajectoryFIFO &Response);

```

```
int main(int argc, char* argv[]) {

//We load the API.
commandLayer_handle = LoadLibrary(L"CommandLayerWindows.dll");
AngularPosition current;
int result;
int programResult = 0;

//We load the functions from the library
MyInitAPI = (int(*)()) GetProcAddress(commandLayer_handle, "InitAPI");
MyCloseAPI = (int(*)()) GetProcAddress(commandLayer_handle, "CloseAPI");
MyGetAngularCommand = (int(*)(AngularPosition &Response))
GetProcAddress(commandLayer_handle, "GetAngularCommand");
MyRunGravityZEstimationSequence7DOF = (int(*)(ROBOT_TYPE type, float
OptimalzParam[OPTIMAL_Z_PARAM_SIZE]))
GetProcAddress(commandLayer_handle,
"RunGravityZEstimationSequence7DOF");
MySetGravityOptimalParameter = (int(*)(float
Command[GRAVITY_PARAM_SIZE])) GetProcAddress(commandLayer_handle,
"SetGravityOptimalZParam");
MySetGravityType = (int(*)(GRAVITY_TYPE Type))
GetProcAddress(commandLayer_handle, "SetGravityType");

//Verify that all functions have been loaded correctly
if ((MyInitAPI == NULL) || (MyCloseAPI == NULL) ||
(MyGetAngularCommand == NULL) ||
(MySetGravityOptimalParameter == NULL) ||
(MyRunGravityZEstimationSequence == NULL) ||
(MySetGravityType==NULL)) {
   cout << "* * * E R R O R D U R I N G I N I T I A L I Z A T I O N * * *" <<
endl;
   programResult = 0;
}
else {
   cout << "I N I T I A L I Z A T I O N C O M P L E T E D" << endl << endl;
   int result = (*MyInitAPI)();
   int resultComm;
   AngularPosition DataCommand;

   // Get the angular command to test the communication with the robot
   resultComm = MyGetAngularCommand(DataCommand);
   KinovaDevice list[MAX_KINOVA_DEVICE];

   // If the API is initialized and the communication with the robot is working
   if (result == 1 && resultComm == 1) {

      // Choose robot type
      ROBOT_TYPE type = SPHERICAL_7DOF_SERVICE;
      float OptimalzParam[OPTIMAL_Z_PARAM_SIZE_7DOF];

      // Run identification sequence
      cout << "Running gravity parameters estimation trajectory..." << endl;
      MyRunGravityZEstimationSequence7DOF (type, OptimalzParam);
      MySetGravityOptimalParameter(OptimalzParam);

```

```
                // informs the robot on the new optimal gravity parameters
                MySetGravityType(OPTIMAL); //sets the gravity compensation mode to
Optimal
            }
            cout << endl << "C L O S I N G A P I" << endl;
            result = (*MyCloseAPI)();
            programResult = 1;
        }

        FreeLibrary(commandLayer_handle);

        return programResult;
```

Important considerations for setting Optimal mode

IMPORTANT: Before launching the calibration trajectory with the function RunGravityZEstimationSequence(), please ensure the robot has enough space to move. The top 1.5 meter-radius spherical space around the base should be free of obstacles. The hand should technically never go below the base, so the base can be fixed on a table. For ideal results, you can attach the base on asmall stand so the space below the robot (except the space below the base) is free.

Before launching the calibration trajectory to find the optimal gravity parameters (`RunGravityZEstimationSequence()` function), reset the torque sensors zero value. Whenever your torque sensors readings seem off, start by resetting the torque sensors zero value, especially if you want to use admittance or torque mode.

The Optimal gravity compensation mode works only when the arm is standing on a flat surface (the gravity vector is in the direction [0, 0, -9.81]).

The robot does not save the optimal gravity parameters. If you want to use the Optimal gravity compensation mode, you will have to load the parameters from the 'ParametersOptimal_Z.txt' text file, send them to the robot using the `SetOptimalZParam()` function and activate the Optimal mode using the `SetGravityType()` function after each reboot.

Robot limits

The robot has certain limits that research users should be cautious of. This includes:

- Joint speed and utilization limits
- Software position limits of actuators
- Software position limits of fingers

Joint speed and utilization limits

	Big actuators (75mm)	Small actuators (58mm)	Fingers actuators
Maximum RPM	6 RPM	8 RPM	600 RPM
Maximum command / sec	36° / sec	48° / sec	300 mm / sec 10800° / sec
Maximum repetitive current	1.5A	1.6A	1.4
Maximum temperature	80°C	80°C	80°C

Utilization over these maximum recommended parameters may affect lifetime of the arm and its modules. Please refer to the specification sheet information for your particular arm configuration for additional information

Software position limits of actuators

This section provides a reference of software position limits of various robotic arm configurations.

The following limits indicate the software limits that are present in the robot base to ensure safety of the robot. These limins are there to protect the arm and its environment.

When moving the actuators, the following minimum and maximum positions should be followed. If the command sent to any of these actuators goes beyond these values, the actuators will stop moving.

Table 16: 4 DoF software limitations

Joint	Min(°)	Max(°)
1	-10 000	10 000
2	47	313
3	19	341
4	-10 000	10 000

Table 17: 6DOF curved wrist software limitations

Joint	Min(°)	Max(°)
1	-10 000	10 000
2	50	310
3	19	341
4	-10 000	10 000
5	-10 000	10 000
6	-10 000	10 000

Table 18: 6 DoF Spherical software limitations

Joint	Min(°)	Max(°)
1	-10 000	10 000
2	47	313
3	19	341
4	-10 000	10 000
5	65	295
6	-10 000	10 000

Table 19: 7 DoF Spherical software limitations

Joint	Min(°)	Max(°)
1	-10 000	10 000
2	47	313
3	-10 000	10 000
4	30	330
5	-10 000	10 000

Joint	Min(°)	Max(°)
6	65	295
7	-10 000	10 000

These limitations are in angular mode. In Cartesian mode, the limits for joints 2 and 3 are different.

For right-hand mode:
- Joint 2 - **min**: 140°, **max**: same as angular
- Joint 3 - **min**: same as angular, **max**: 165°

For left-hand mode:
- Joint 2 - **min**: same as angular, **max**: 220°
- Joint 3 - **min**: 195°, **max**: same as angular.

Software position limitations of fingers

When moving the robot fingers, the following minimum and maximum positions should be respected. If the command sent to any of these fingers goes beyond these values, the fingers will stop moving.

Finger #	Minimum	Maximum
1	0 mm (0°)	18.9 mm (6800°)
2	0 mm (0°)	18.9 mm (6800°)
3	0 mm (0°)	18.9 mm (6800°)

Guidance for advanced users

Kinematics Parameters

There are several sets of useful kinematic parameters:

- Basic geometric parameters of the arm
- Classic Denavit-Hartenberg (DH) parameters
- Directions of joints in angular space

Basic geometric parameters - 4 DoF

This section describes the basic geometric parameters of the 4 DoF configuration.

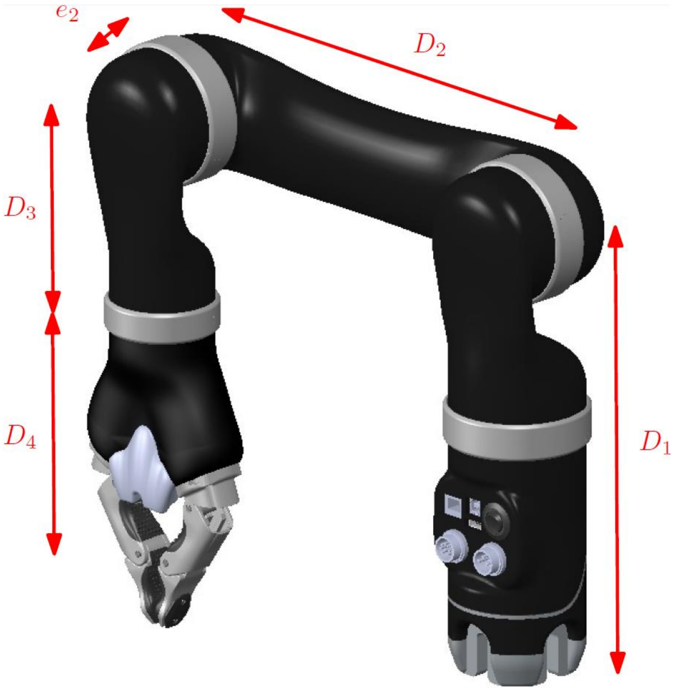


Table 20: 4 DoF basic geometric parameters

Parameter	Description	Length (m)
D1	Base to shoulder	0.2755
D2	Upper arm length (shoulder to elbow)	0.4100
D3	Forearm length (elbow to wrist)	0.2073
D4	First wrist length (center of actuator 4 to center of actuator 5)	0.1600
e2	Joint 3-4 lateral offset	0.0098

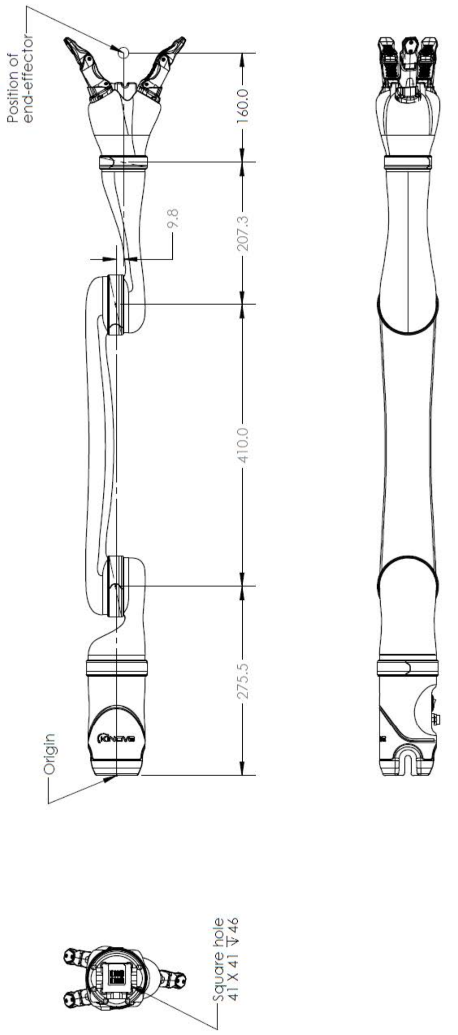


Figure 6: Detailed 4 DoF robot length values (units in mm)

Basic geometric parameters - 6 DoF curved wrist

This section describes the basic geometric parameters of the 6 DoF curved wrist configuration.

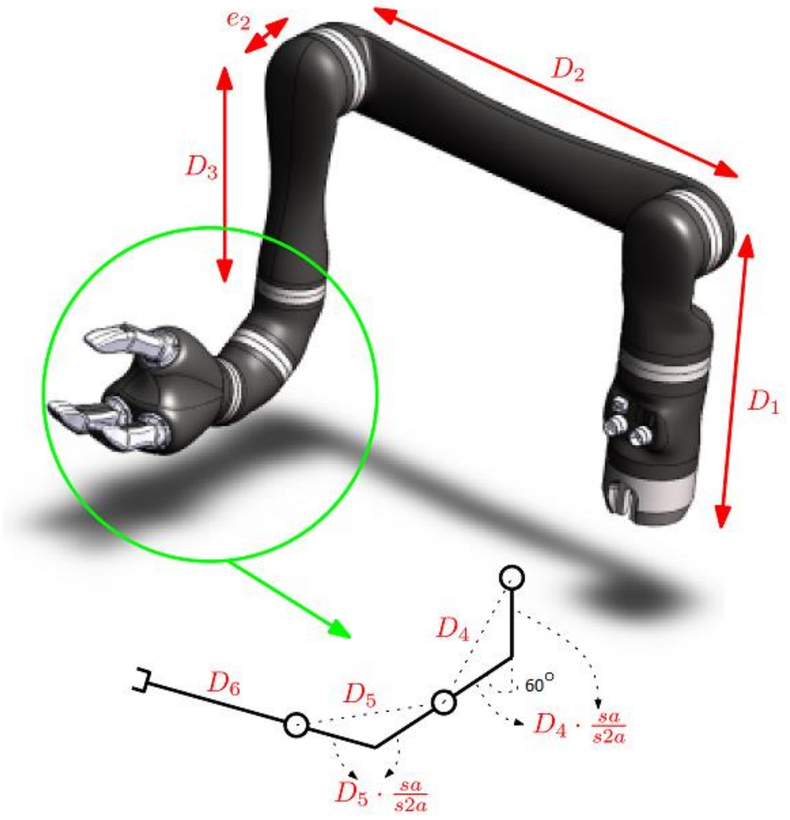


Table 21: 6 DoF curved wrist basic geometric parameters

Parameter	Description	Length (m)
D1	Base to shoulder	0.2755
D2	Upper arm length (shoulder to elbow)	0.4100
D3	Forearm length (elbow to wrist)	0.2073
D4	First wrist length (center of actuator 4 to center of actuator 5)	0.0741
D5	Second wrist length (center of actuator 5 to center of actuator 6)	0.0741
D6	Wrist to center of the hand	0.1600
e2	Joint 3-4 lateral offset	0.0098

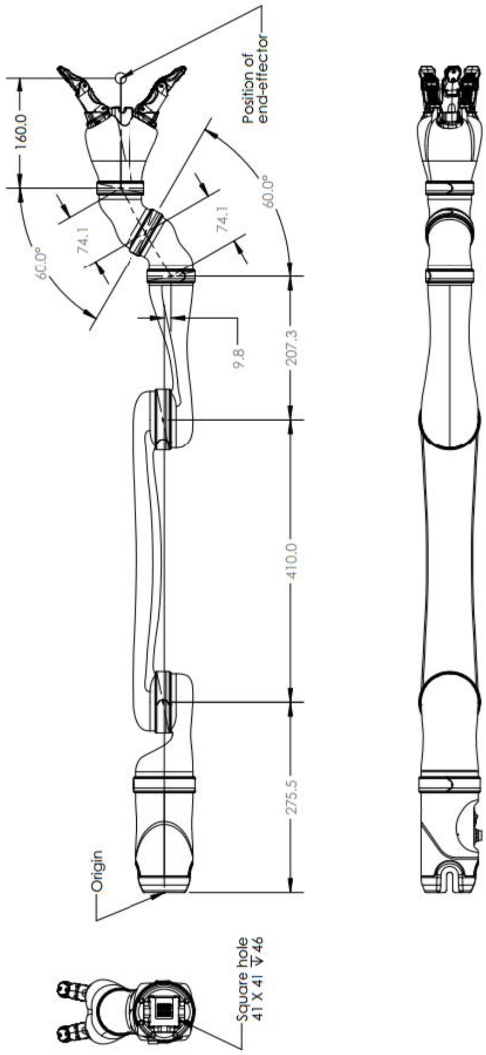
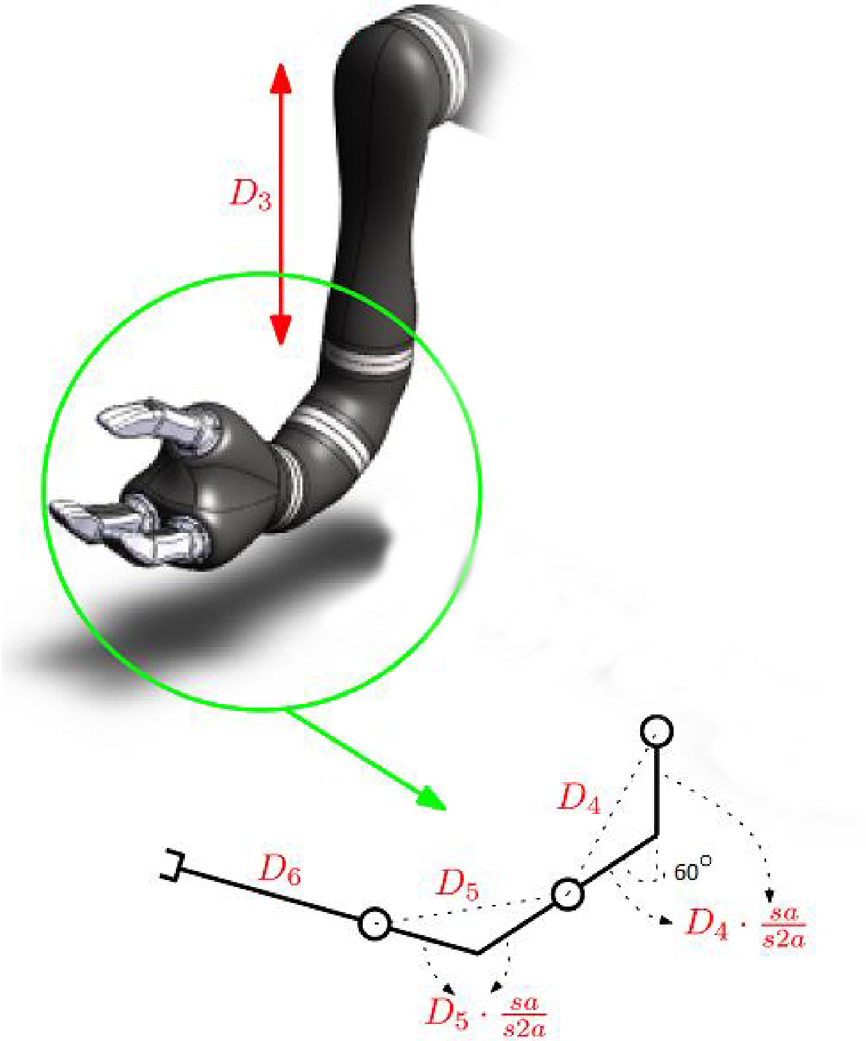


Figure 7: Detailed 6 DoF curved wrist robot length values (units in mm)

Alternate geometric parameters - 6 DoF curved wrist

This section describes alternate parameters that are useful for describing the geometry for kinematics of the 6 DoF curved wrist configuration.

The kinematics of the 6 DoF curved wrist configuration are more complicated than for a spherical wrist due to the more complicated geometry. To simplify the analysis, it is useful to break down each of the two curved wrist segments into two component straight-line sub-segments of equal length, with the second sub-segment angled 60° from the first.



In this way, the arm from the elbow to the center of the hand can be analyzed as three straight-line segments:

- d4b - distance from elbow to end of first sub-segment of first wrist segment
- d5b - distance from end of first sub-segment of first wrist segment to end of first sub-segment of second wrist segment
- d6b - distance from end of second sub-segment of second wrist segment to center of hand

Table 22: Alternate parameters

Parameter	Description	Value
aa	Half of the angle of curvature of each wrist segment (60°), measured in radians	(30.0 * PI) / 180.0
sa	Sine of half the angle of curvature of wrist segment	sin(aa)
s2a	Sine of angle of curvature of wrist segment	sin(2*aa)
d4b	Length of straight-line segment from elbow to end of first sub-segment of first wrist segment.	D3 + (sa / s2a) * D4
d5b	Length of straight-line segment consisting of second sub-segment of first wrist segment and first sub-segment of second wrist segment	(sa / s2a) * D4 + (sa / s2a) * D5
d6b	Length of straight-line segment consisting of second sub-segment of second wrist segment and distance from wrist to the center of the hand	(sa / s2a) * D5 + D6

The DH parameters for the lower part of the robot are most naturally expressed in terms of these alternate parameters.

Basic geometric parameters - 6 DoF spherical wrist

This section describes the basic geometric parameters of the 6 DoF spherical wrist configuration.

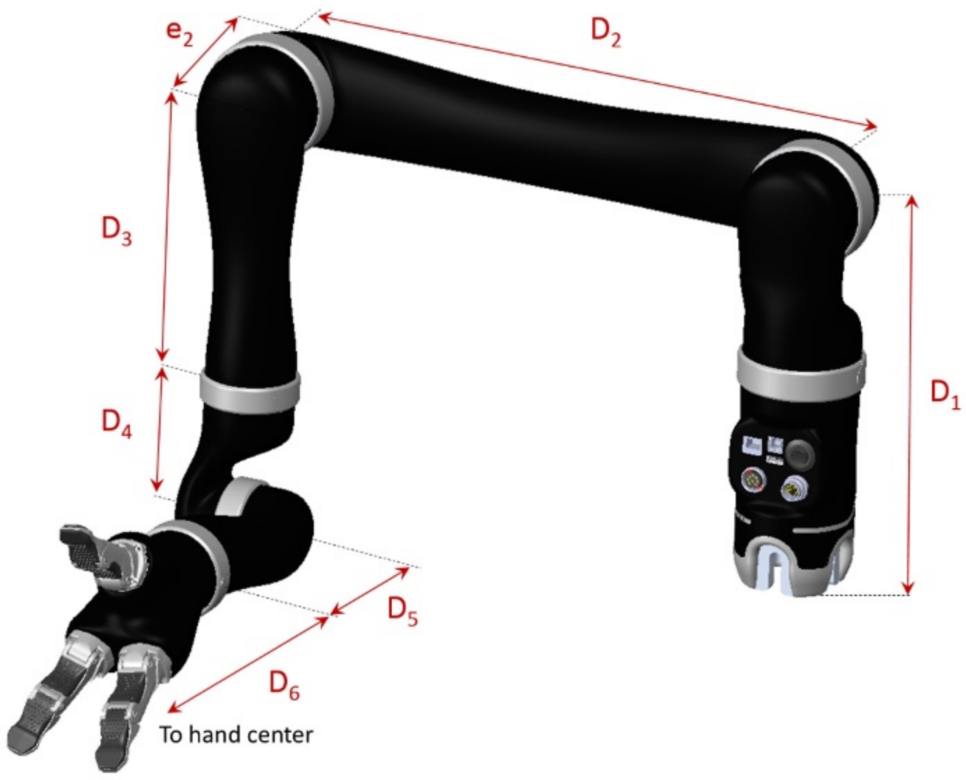


Table 23: 6 DoF spherical basic geometric parameters

Parameter	Description	Length (m)
D1	Base to shoulder	0.2755
D2	Upper length (shoulder to elbow)	0.4100
D3	Forearm length (elbow to wrist)	0.2073
D4	First wrist length	0.1038
D5	Second wrist length	0.1038
D6	Wrist to center of the hand	0.1600
e2	Joint 3-4 lateral offset	0.0098

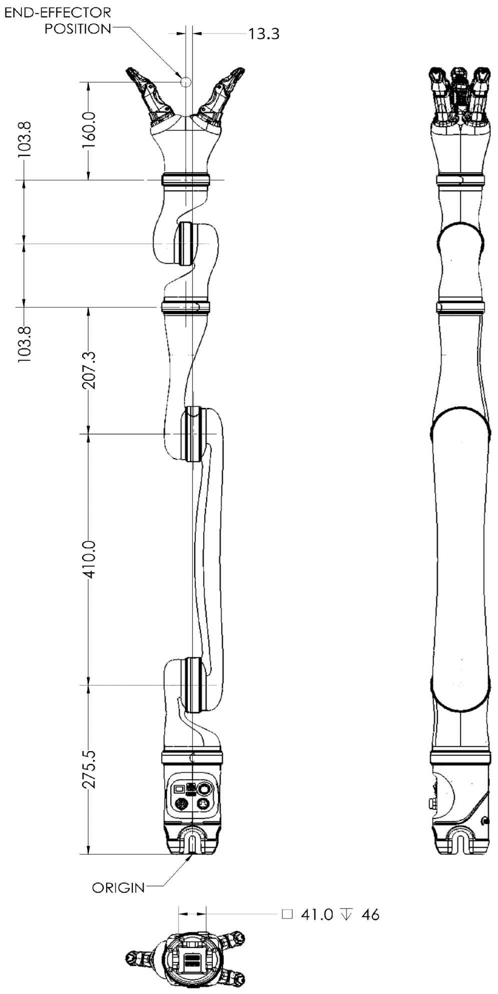


Figure 8: Detailed 6 DoF spherical robot length values (units in mm)

Basic geometric parameters - 7 DoF spherical wrist

This section describes the basic geometric parameters of the 7 DoF spherical wrist.

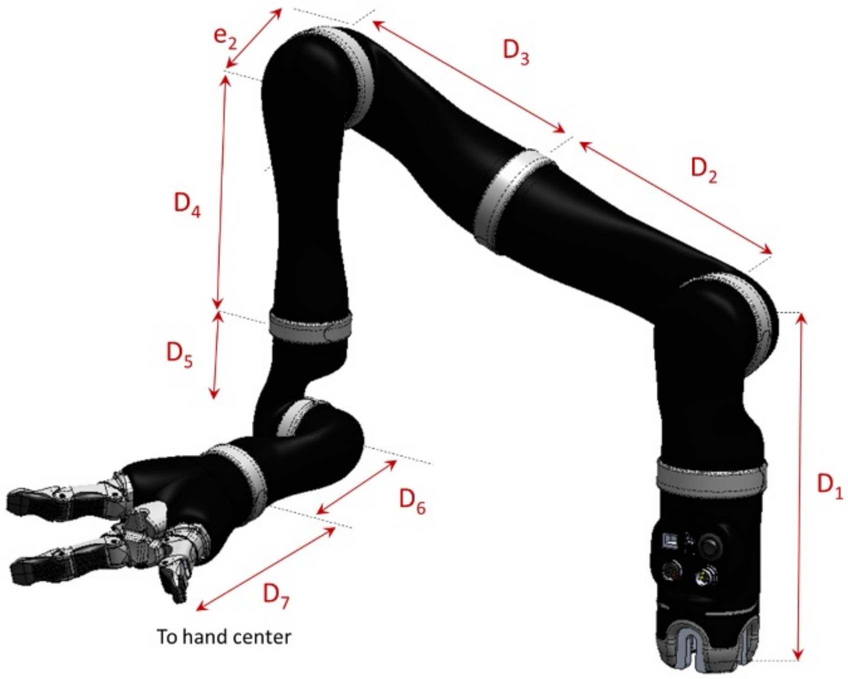


Table 24: 7 DoF spherical basic geometric parameters

Parameter	Description	Length (m)
D1	Base to shoulder	0.2755
D2	First half upper arm length	0.2050
D3	Second half upper arm length	0.2050
D4	Forearm length (elbow to wrist)	0.2073
D5	First wrist length	0.1038
D6	Second wrist length	0.1038
D7	Wrist to center of the hand	0.1600
e2	Joint 3-4 lateral offset	0.0098

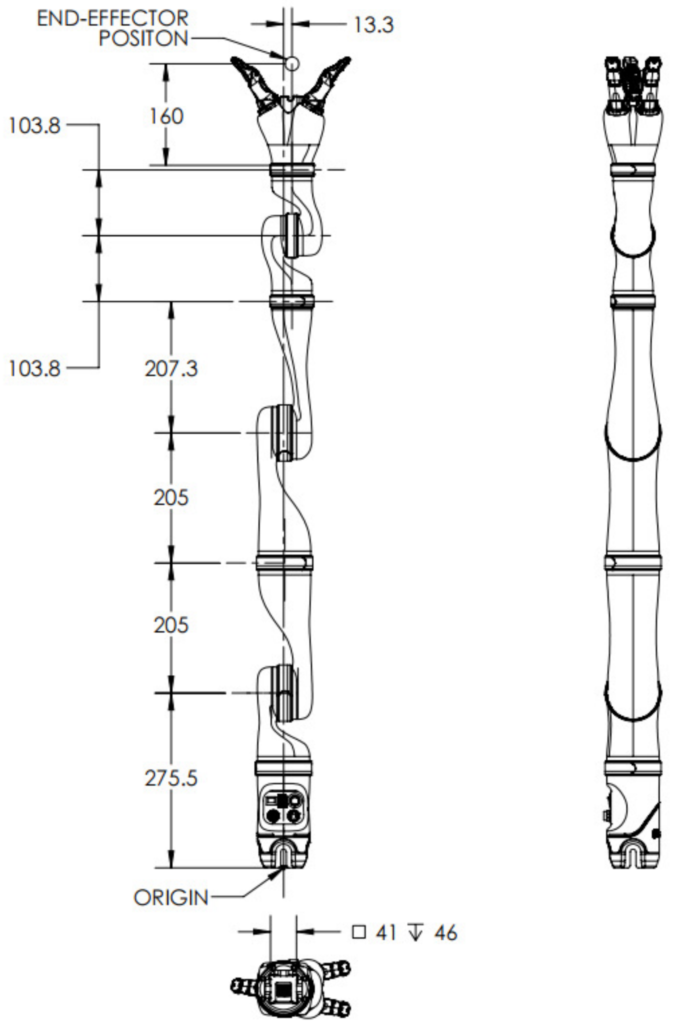


Figure 9: Detailed 7 DoF spherical robot length values (units in mm)

Classic DH parameters - 4 DoF

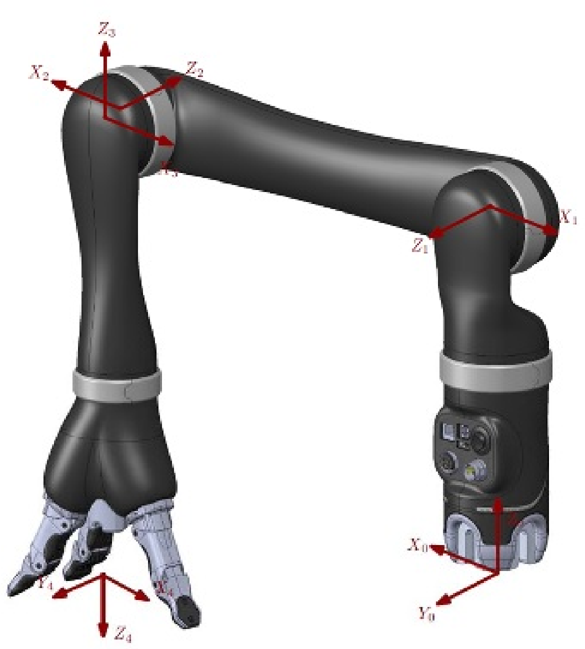
This section describes the Classic DH parameters for the 4 DoF.

Table 25: 4 DoF DH parameters

i	α_i	a_i	d_i	θ_i
1	π / 2	0	D1	q1
2	π	D2	0	q2
3	π / 2	0	-e2	q3
4	π	0	D3 + D4	q4

Table 26: Transformation from DH algorithm to robot physical angles

Q1(physical) = -Q1(DH algo)
Q2(physical) = Q2(DH algo) + 90
Q3(physical) = Q3(DH algo) - 90
Q4(physical) = Q4(DH algo) - 270



Represented angular position [180, 270, 90, 270]

Classic DH parameters - 6 DoF curved wrist

This section describes the Classic DH parameters for the 6 DoF with curved wrist.

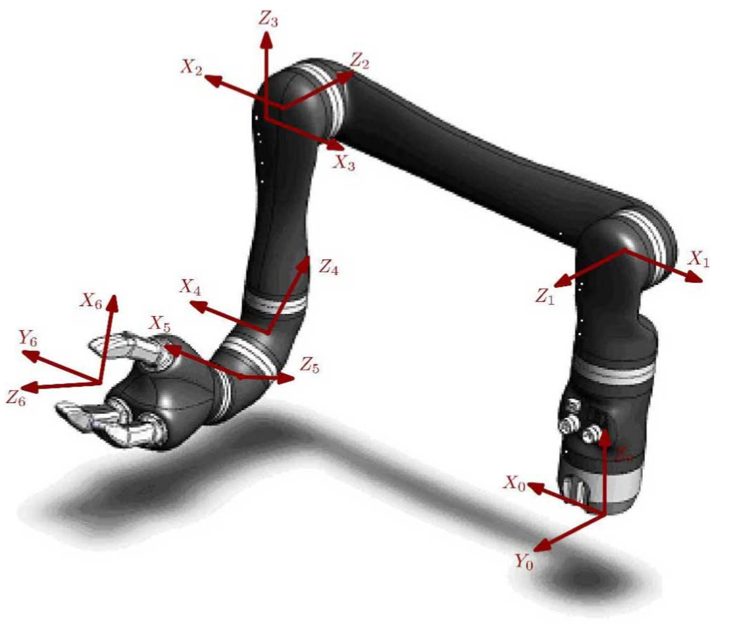
The DH parameters for the 6 DoF curved arm robot are naturally expressed in terms of alternate parameters.

Table 27: 6 DoF curved wrist DH parameters

i	α_i	a_i	d_i	θ_i
1	π / 2	0	D1	q1
2	π	D2	0	q2
3	π / 2	0	-e2	q3
4	2*aa	0	-d4b	q4
5	2*aa	0	-d5b	q5
6	π	0	-d6b	q6

Table 28: Transformation from DH algorithm to robot physical angles

Q1(physical) = -Q1(DH algo)
Q2(physical) = Q2(DH algo) + 90
Q3(physical) = Q3(DH algo) - 90
Q4(physical) = Q4(DH algo)
Q5(physical) = Q5(DH algo) + 180
Q6(physical) = Q6(DH algo) - 90



Represented angular position:
[180, 270, 90, 180, 180, 0]

Classic DH parameters - 6 DoF spherical

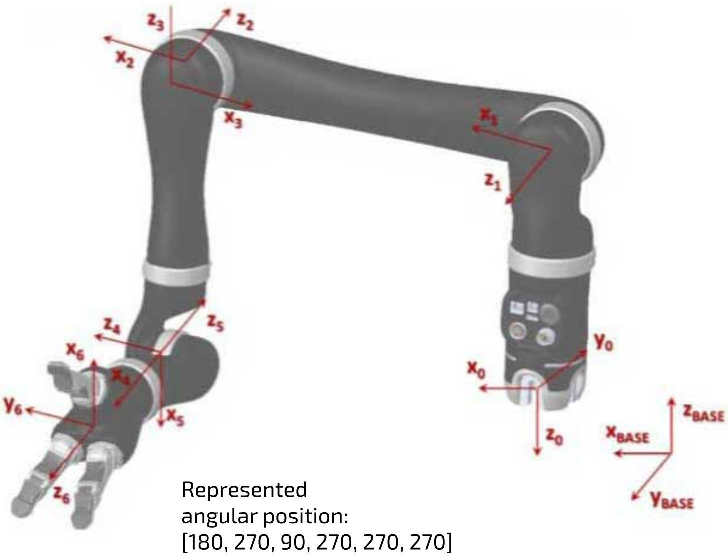
This section describes the Classic DH parameters for the 6 DoF spherical.

Table 29: 6DOF spherical DH parameters

i	α_i	a_i	d_i	θ_i
1	π / 2	0	D1	q1
2	π	D2	0	q2
3	π / 2	0	-e2	q3
4	π / 2	0	-(D3 + D4)	q4
5	π / 2	0	0	q5
6	π	0	-(D5 + D6)	q6

Table 30: Transformation from DH algorithm to robot physical angles

Q1(physical) = Q1(DH algo) + 180
Q2(physical) = Q2(DH algo) - 90
Q3(physical) = Q3(DH algo) - 90
Q4(physical) = Q4(DH algo)
Q5(physical) = Q5(DH algo)
Q6(physical) = Q6(DH algo) + 90



$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Classic DH parameters - 7 DoF spherical

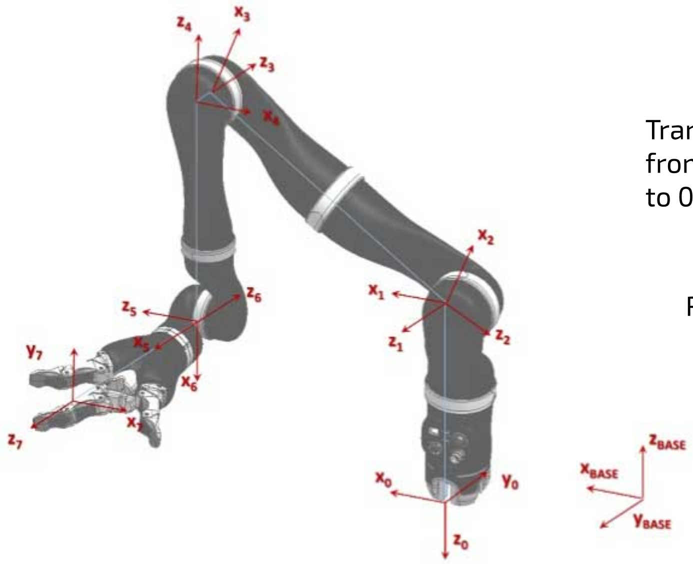
This section describes the Classic DH parameters for the 7 DoF spherical.

Table 31: 7 DoF spherical DH parameters

i	α_i	a_i	d_i	θ_i
1	π / 2	0	-D1	q1
2	π / 2	0	0	q2
3	π / 2	0	-(D2 + D3)	q3
4	π / 2	0	-e2	q4
5	π / 2	0	-(D4 + D5)	q5
6	π / 2	0	0	q6
7	π	0	-(D6 + D7)	q7

Table 32: Transformation from DH algorithm to robot physical angles

Q1(physical) = Q1(DH algo) + 180
Q2(physical) = Q2(DH algo)
Q3(physical) = Q3(DH algo)
Q4(physical) = Q4(DH algo)
Q5(physical) = Q5(DH algo)
Q6(physical) = Q6(DH algo) + 90



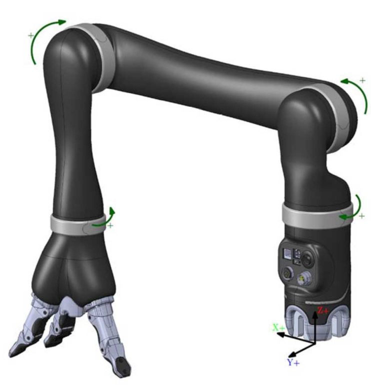
Transformation matrix
from BASE
to 0 axis system:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

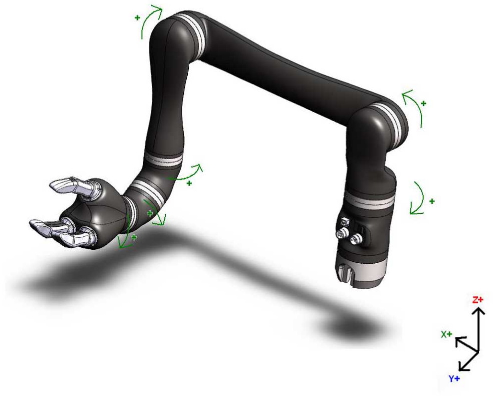
Represented
angular position:
[180, 240, 0, 60, 90, 90, 180]

Directions of joints in angular space

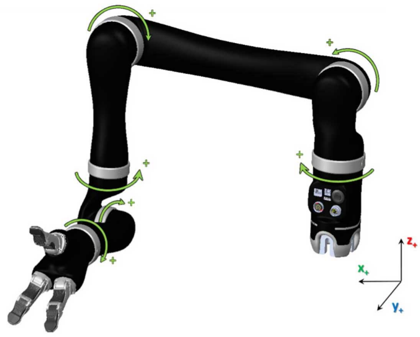
The following image represents the positive direction of rotation of each actuator for the different configurations of the robot:



4 DOF



6 DOF
curved wrist



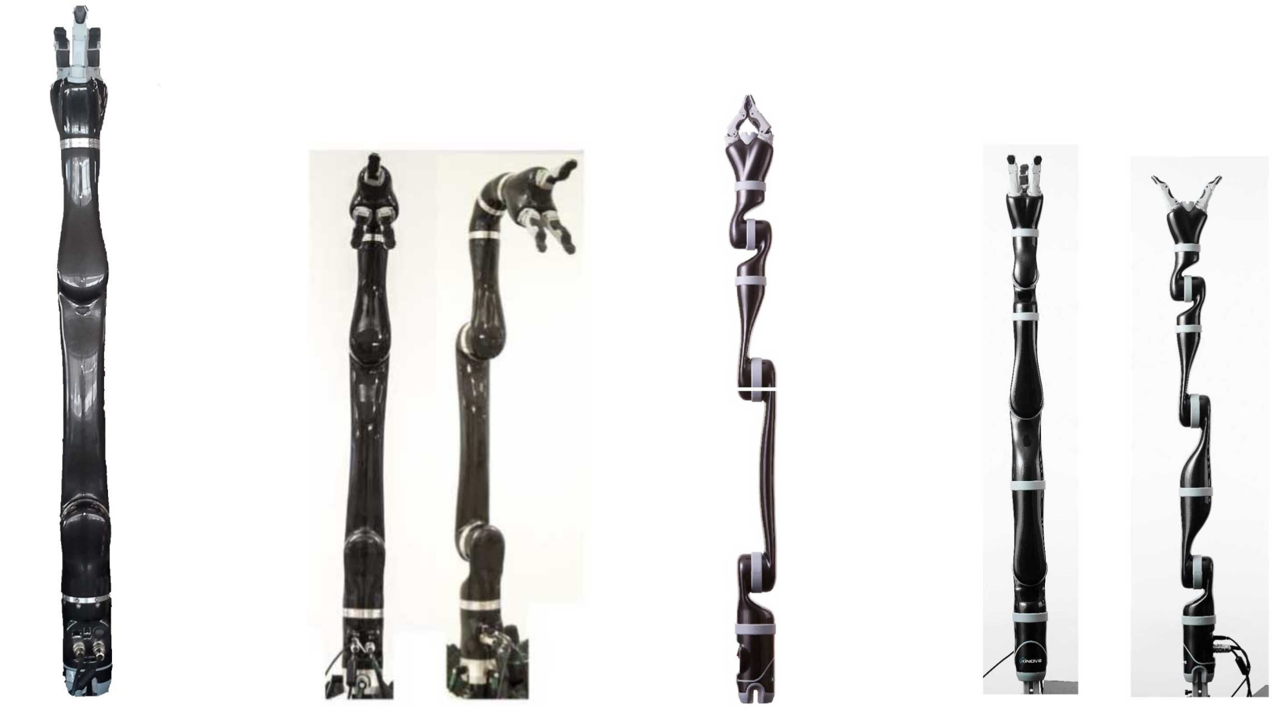
6 DOF
spherical

7 DOF
spherical

Setting zero position and setting torque sensors zero

This section describes considerations for the arm position indexation and torque sensors zeroing processes.

Position indexation



4 DOF
[180°, 180°, 180°, 180°]

6 DOF
curved wrist
[180°, 180°, 180°, 180°, 180°, 180°]

6 DOF
spherical
[180°, 180°, 180°, 180°, 180°, 180°]

7 DOF
spherical
[180°, 180°, 180°, 180°, 180°, 180°, 180°]

Figure 10: Set zero positions joint angles

Resetting torque sensors zero value

In order to reset the torque sensors zero, you must first place the arm in a position where gravity does not influence the joint torques. The set zero position is good but it is suggested to use the positions as indicated in the table below since this position also limits perpendicular torques on the actuators.

Table 33: Recommended reset positions

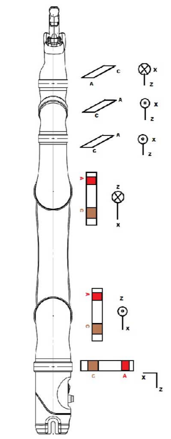
Config	Joint #1 angle	Joint #2 angle	Joint #3 angle	Joint #4 angle	Joint #5 angle	Joint #6 angle	Joint #7 angle
4 DoF	*	180	180	0	-	-	-
6 DoF curved wrist	*	180	180	0	0	180	-
6 DoF spherical	*	180	180	0	0	180	-
7 DoF spherical	*	180	*	180	*	180	*

* Any angle

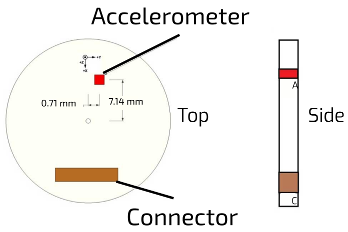
Actuator accelerometers information

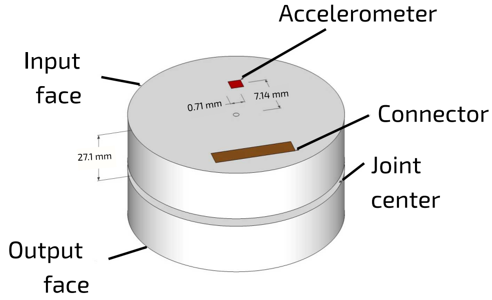
This section has information about the accelerometers in each actuator.

The image below shows information about the accelerometers in each joint actuator.



6 DOF arm shown for
illustration purposes





Maintenance and troubleshooting

Cleaning, maintenance and disposal

This section describes maintenance and disposal considerations.

Cleaning instructions

Only the external surfaces of the product may be cleaned. This is done using a damp cloth and a mild detergent. The following describes the steps for cleaning the product:

- Prepare a water/soap solution using about 2 ml of dish soap for 100 ml of water
- Immerse a clean cotton cloth in the solution
- Remove the cloth and wring out thoroughly
- Gently rub the external surface to be cleaned

⚠ Do not wash more than three times per day.

⚠ Do not immerse any part of the product under water or snow.

⚠ The product is not intended to be sterile. No sterilization process should be undertaken with the product.

⚠ Do not rub the external surfaces with abrasive materials.

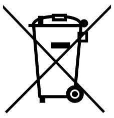
Preventive Maintenance

The product requires no maintenance. Fingers should be cleaned and lubricated every six months.

⚠ Refer all servicing to qualified service personnel. Servicing is required when the apparatus has been damaged in any way, for example if the power-supply cord or plug is damaged, if the product does not operate normally or if it has been dropped.

⚠ The product has no user serviceable parts. Do not open.

Disposal



The product contains parts that are deemed to be hazardous waste at the end of useful product life. For further information on recycling, contact your local recycling authority or Kinova distributor. In any case, always dispose of product via a recognized agent.

Updating robot firmware

This section describes how to update the robot firmware using the KINOVA® *Development center* application.

Before you begin

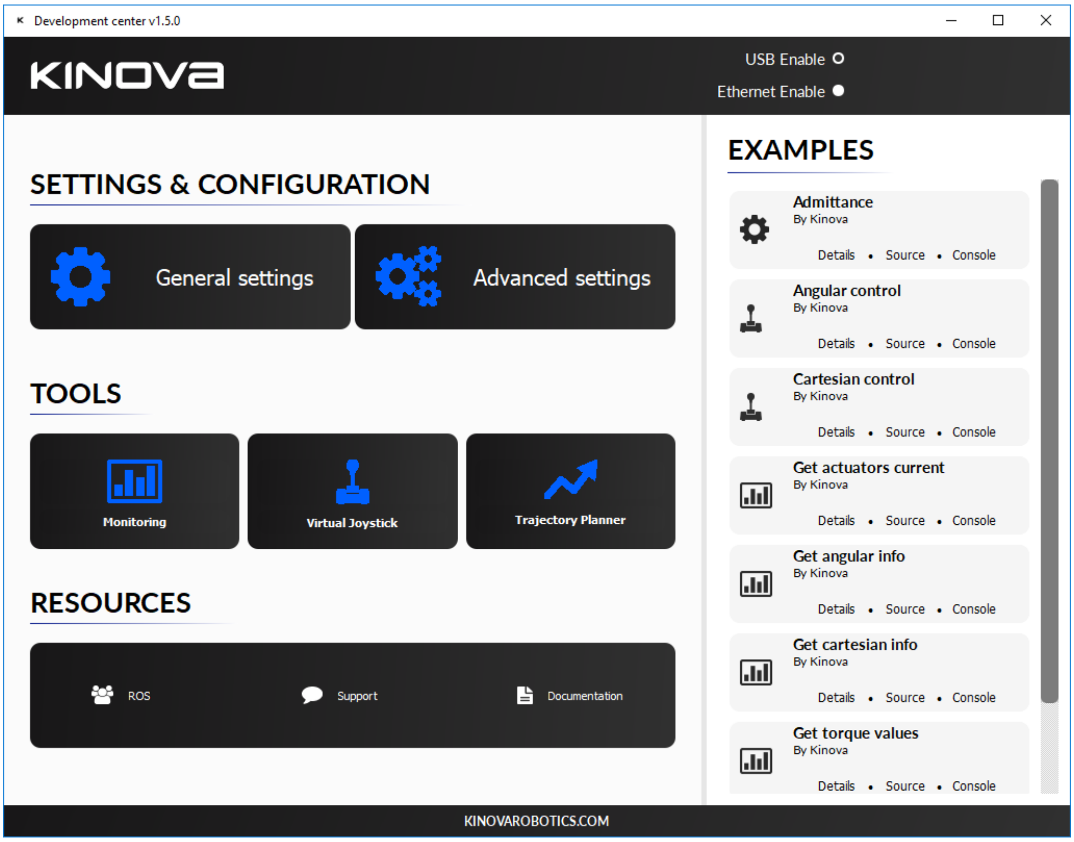
You will need a laptop computer with the KINOVA® *Software development kit* installed and on which the Kinova USB drivers have been installed. You will also need to have previously downloaded the new .hex firmware file. You will also need a USB type A to USB type B cable.

About this task

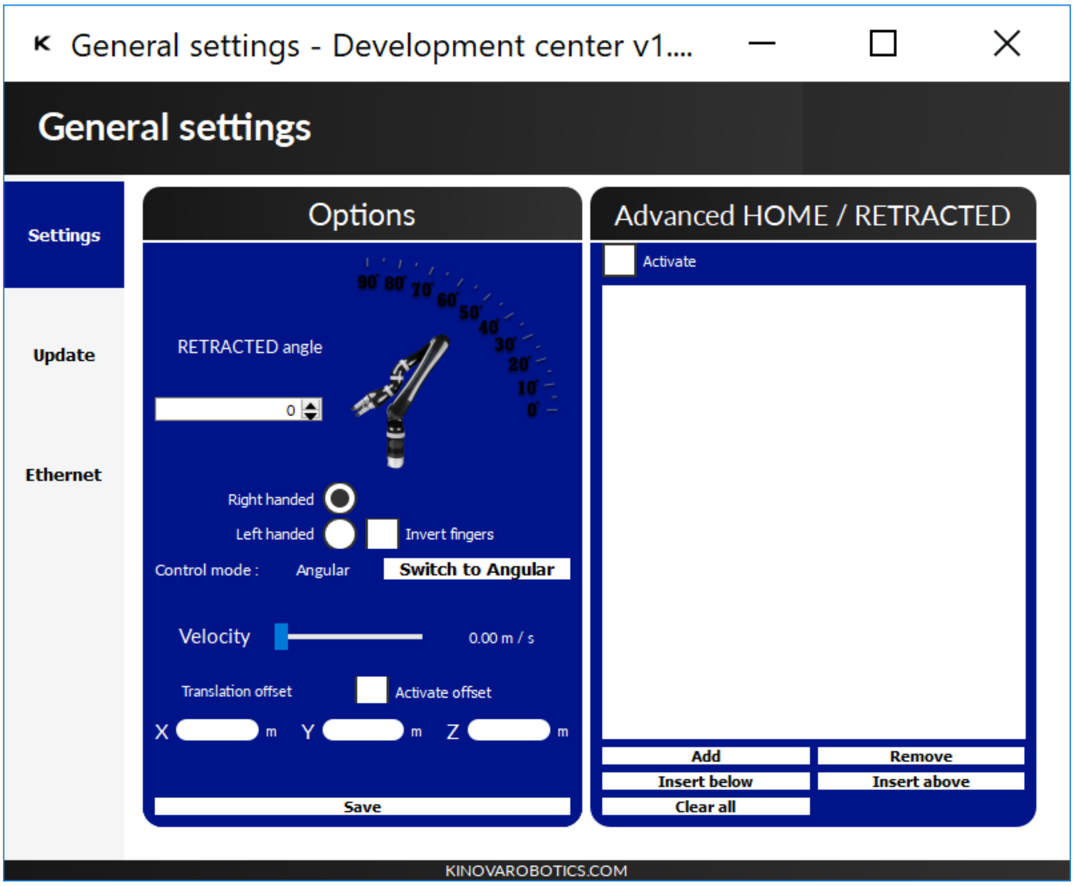
From time to time you may want to update the robot firmware. This procedure explains how to update the firmware with a downloaded firmware file using the *Development center* application.

Procedure

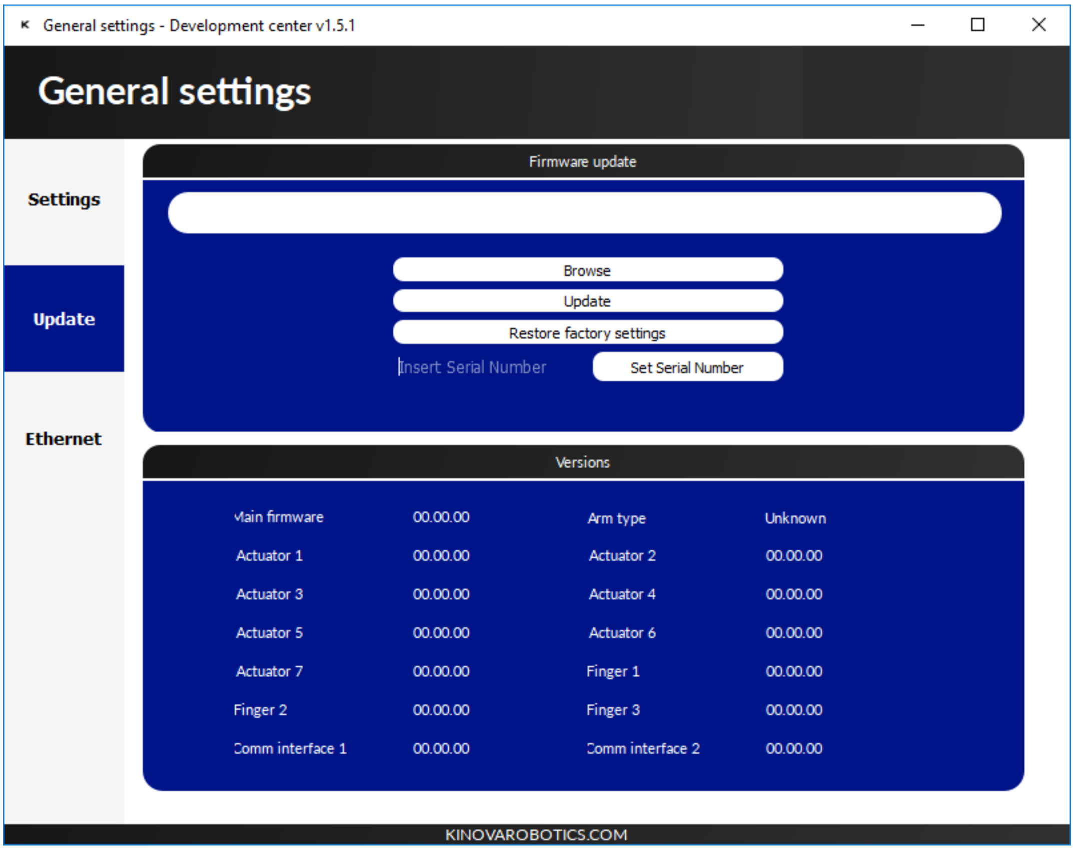
1. Locate the USB type B connector on the connector panel of the robot base. Take your USB cable, and connect the USB type B end to the arm controller.
2. Connect the other end of the cable to a USB type A port on the laptop.
3. If the robot is not powered on, flip the power switch to on.
4. Open the *Development center* software on the laptop.



5. Click the General Settings button to launch the Configuration window.



6. Click the Update button to open the Firmware update tab.



7. Click the Browse button to launch a File Explorer window and find the new .hex firmware file. Select the file and click Open to proceed.
8. Click update. The firmware will be installed on the robot.

⚠ Do not unplug the USB cable while the firmware update process is in progress. You could corrupt the firmware of the robot and render it unusable. Such damage is not covered by warranty.

9. Turn the robot power off and then back on to reboot the robot.

Results

The firmware is now updated on the robot.

Kinova gripper removal

This section describes the procedure to remove the standard gripper from a *Gen2 Ultra lightweight robot*.

Before you begin

Ensure that the arm is powered down before beginning. You will need a 2 mm hex key.

Note: The procedure will require touching the actuators; it is recommended that you wear a grounded anti-static wrist strap to protect against ESD damage to the actuator electronic circuitry.

About this task

Removing the gripper is the first step to replacing a malfunctioning gripper or installing a different type of end effector. This procedure requires about 10 minutes to complete.

Procedure

1. Locate the last actuator at the end of the robotic arm.
2. Remove the plastic ring covering the actuator with a flat screwdriver. This will expose the aluminum rings on either side of the actuator where the robot arm shell and gripper shell are attached to the actuator. You will see two sets of six evenly-spaced screws, one on each ring.

3. Remove the six screws on the output side of the actuator on the gripper shell.

   Note: Make sure to preserve the screws. You will need them to attach the gripper adapter, and also if you want to reattach the Kinova gripper later on.

   Note: Removing the screws on the other (input) side of the actuator will detach the last actuator from the arm shell. This is not necessary for removal of the gripper.
4. Gently pull the gripper away from the wrist.

   Note: Do not pull too hard. There is a flex cable connecting the last robot arm actuator to the gripper, and you might damage the cable if you pull too hard.
5. Detach the flex cable from the last robot arm actuator. There may be glue holding the cable in place. If so, remove the glue, then pull out the small tabs on the side of the connector to loosen the cable. Pull the flex cable free from the connector to fully disconnect it from the actuator.

   Note: Keep the other end of the flex cable connected to the gripper. You will need this later to reconnect the gripper.

Kinova gripper installation

This sections describes the procedure to install a Kinova KG-Series gripper on a Gen2 *Ultra lightweight robot*.

Before you begin

The previous gripper needs to have been removed before installing the new gripper. The arm should be powered down. You will need:

- Loctite 242 (or 243) threadlocker (to hold screws in place)
- hot glue (to hold flex cable connectors in place)
- 2 mm hex key (for tightening screws)
- anti-static wrist strap or equivalent (to protect actuators from ESD)

About this task

To install the new gripper (or re-install a gripper removed for whatever reason), you will need to:

- Ensure that all gripper finger motors are all properly interconnected, fully engaged in their connectors and in the right direction.
- Connect a flex cable from the finger motors to the last actuator on the arm. This provides power and signals to the finger actuators in the hand.
- Connect the shell of the gripper to the last actuator on the arm. This mechanically couples the gripper to the actuator.

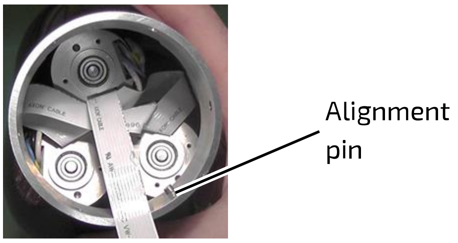
This procedure requires about 15 minutes to complete.

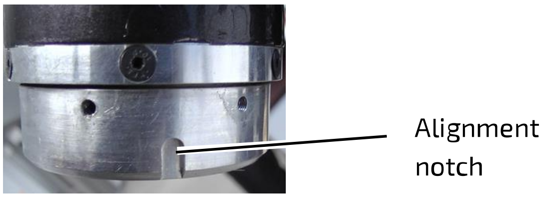
Procedure

1. Bring the new gripper close to the last actuator on the arm. There will be a flex cable coming from finger motor # 1 in the gripper. Open the locking tabs on the cable connector of the last actuator and insert the other end of this flex cable to the actuator connector. Pinch the locking tabs closed to secure the cable.

   Note: The flex cable connector has two sides. Ensure that the side with the metallic contacts is face up (rather than the blue side of the cable connector) with respect to the exposed face of the actuator.
2. Secure the cable by closing the latches on either side of the connector. Apply a thin amount of hot glue to help hold the flex cable in place.

   Note: Do not use any other type of glue.
3. Locate the alignment pin in the shell of the new gripper and the alignment notch on the last actuator of the arm. Line up the pin on the gripper shell with the notch on the actuator and slide the shell of the gripper over the actuator.

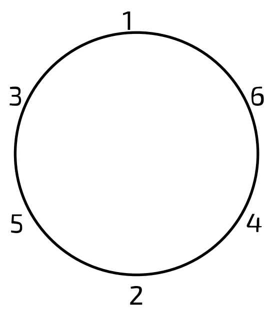




4. Apply Loctite threadlocker to the six screws and loosely screw them in around the ring.
5. Once all the screws are in place, torque the screws in the following star pattern order.

Note: Apply no more than 0.5 N·m (0.74 ft-lb) of torque to these screws.

Star pattern for torquing screws of small actuator



6. Place the plastic ring over the aluminum rings at the actuator junction.

What to do next

Power on the arm. Test out the new gripper by moving the wrist in each direction and perform five open / close cycles in the three finger configuration and again in the one finger configuration.

Troubleshooting / FAQs

This section gives guidance in troubleshooting common issues.

Table 34: Troubleshooting tips

Issues	Suggested actions
My robot will not switch from position mode to torque mode	Make sure nothing is touching your robot when you try to switch to torque mode. Monitor your robot's torque readings, computed gravity-free torques and force/torque at the hand using the *Development Center* or API. Normally, gravity-free torques and force/torque at the hand should be low. If your torque readings seem unusual, redo your torque sensors zero calibration. Normally, a good position to switch from position mode to torque mode is the robot's Ready position (although any position is technically acceptable).
My robot keeps falling down in torque mode	Torque mode is **very sensitive** to gravity compensation. First, check that your torque sensors' zero was adequately calibrated. You can also follow the tips given in the section Improving robot behavior in Torque mode. If your robot keeps falling down, you can try adding a negative (m < 0) virtual mass using the *Torque Console* or API. **Note:** If instead of falling down, your arm is always going up, you should follow the same tips but with adding a positive virtual mass at the end instead of a negative one.

Issues	Suggested actions
I sent a trajectory in Cartesian mode and the robot stops at some point in the trajectory.	This problem can take several forms: • The robot is in a random position when the trajectory is launched, the robot 'reaches' the first point but does not continue passed that. In this case, the robot never really reaches the first point, but goes towards it without reaching it. • The robot starts the trajectory but gets stuck trying to reach a point. • The robot takes a very unusual twist trying to reach a point. All of these problems are in fact one problem: the robot cannot reach the next trajectory point if it gets into a singularity, experiences self-collision or runs into a protection zone. *"But I moved the robot with the joystick to create the trajectory"* When you are using the joystick, the robot receives a succession of small movements that eventually lead it to the points you have recorded in the trajectory. When the robot receives a joystick point (or Cartesian velocity point), it tries to reach it, then erases it and waits for the next point. When the robot receives a Cartesian trajectory point (or Cartesian position), it computes a straight line from its current position to the next trajectory point. Then it computes the next reachable point on this straight line considering maximum velocity and acceleration limits. The path will be deviated by automatic self-collision and singularity avoidance algorithms. Finally, the robot verifies if the desired trajectory point was reached. If the robot did not reach the trajectory point, it repeats the process and it NEVER erases the trajectory point until it reaches it (within a certain error margin). Because the robot is not able to reach the next trajectory point while avoiding singularities and self-collision, and because this trajectory point does not get erased from the list, you get the impression that the robot either does not receive all the trajectory points or that it freezes in the middle of the trajectory. Now, what to do? Here are a few tips: • Record more points in the trajectory, especially for difficult parts of the trajectory (this will also reduce the risk of having your robot doing unusual moves between your trajectory points) • Familiarize yourself with your robot's singularity configurations • If you do not want to be bothered with singularity and self-collision avoidance algorithms, use angular trajectories.
My robot has an unusual behaviour in admittance/ force control mode.	The gravity compensation is probably not right. Please try to follow the tips given in the the section Improving robot behavior in Torque mode (check torque sensors' zero, use optimal gravity model, etc.).
My robot is in angular mode and it will not switch back to Cartesian mode? However, when I send the robot back to its HOME position, it will switch back to Cartesian mode	The robot will not switch back to Cartesian mode if it is not in a valid Cartesian pose, i.e. too close from a singularity, too close from the base for self-collision avoidance, too close from a joint limit and inside a protection zone. The reason why the robot is always switching back to Cartesian control when you bring it to the Home position is because the Home position is chosen as a valid Cartesian position.

Contacting support

If you need help or have any questions about this product, this guide or the information detailed within, please contact Kinova through the support page of our website at www.kinovarobotics.com/support or by phone at 1 (514) 277-3777.

We value your comments!

To help us assist you more effectively with problem reports, please have the following information ready when contacting Kinova or distributor support:

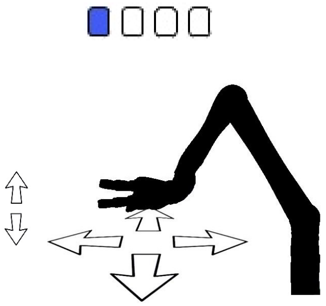
- date and time the problem occurred
- environment where the problem occurred
- actions performed immediately before the problem occurred
- product serial number (this will allow the support agent to access the information regarding your product, such as software version, part revisions and characteristics, etc.).

Appendix - Kinova joystick reference

3-axis mode joystick controls reminder

This section is a visual reminder of the joystick controls in 3-axis mode.

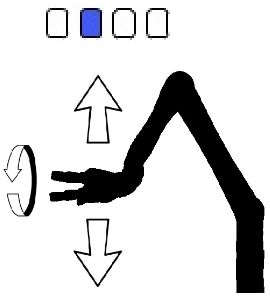
Interface Display



Translation Mode

Joystick control	Hand movement
Joystick forward	Forward
Joystick back	Backward
Joystick left	Left
Joystick right	Right
Stick rotation CW	Up
Stick rotation CCW	Down

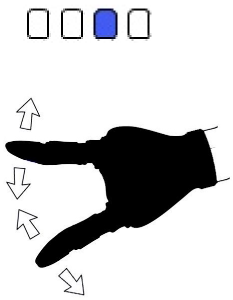
Interface Display



Wrist Mode

Joystick control	Wrist movement
Joystick forward	Top orientation
Joystick back	Bottom orientation
Joystick left	Thumb side orientation
Joystick right	Index side orientation
Stick rotation CW	Rotation of wrist
Stick rotation CCW	Rotation of wrist

Interface Display



Finger Mode

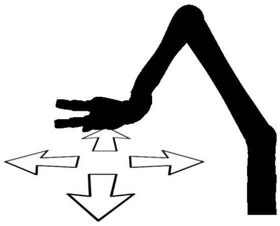
Joystick control	Finger movement
Joystick forward	Open 2 fingers
Joystick back	Close 2 fingers
Joystick left	Close 3 fingers
Joystick right	Open 3 fingers
Stick rotation CW	N / A
Stick rotation CCW	N / A

2-axis mode joystick controls reminder

This section is a visual reminder of the joystick controls in 2-axis mode.

Interface Display

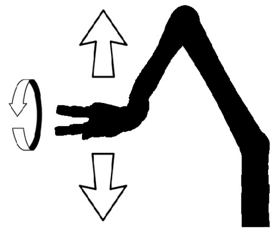
Translation X-Y



Joystick control	**Hand movement**
Joystick forward	Forward
Joystick back	Backward
Joystick left	Left
Joystick right p	Right

Interface Display

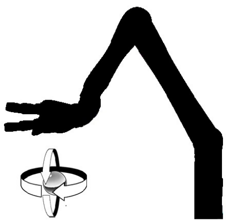
Translation Z / Wrist rotation



Joystick control	**Movement**
Joystick forward	Up
Joystick back	Down
Joystick left	Wrist rotation
Joystick right p	Wrist rotation

Interface Dis lay

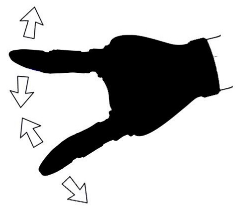
Wrist Orientation



Joystick control	**Wrist orientation**
Joystick forward	Top
Joystick back	Bottom
Joystick left	Thumb side
Joystick right	Index side

Interface Dis lay

Finger Mode



Joystick control	**Finger movement**
Joystick forward	Open 2 fingers
Joystick back	Close 2 fingers

There is no need too small.
No task too great.

kinovarobotics.com

Kinova inc. (Headquarters)
4333, Boulevard de la Grande-Allée
Boisbriand (QC) J7H 1M7
Canada
+1 (514) 277-3777

Kinova Europe GmbH
Großkitzighofer Straße 7 a
86853 Langerringen
+49 8248 8887-928

UG-009_R03

Achieve Extraordinary