User Guide

KINOVA® *Gen3 Ultra lightweight robot*

KINOVA
Achieve Extraordinary

Contents

Welcome

Welcome to the KINOVA® Gen3 *Ultra lightweight robot*.

Thank you for choosing our robot as a tool for your pathbreaking research needs.

This document is meant to provide you with all the information you need to get up and running with your new robot and get the most out of it.

We are here to help you in your journey. If you need any help or have any questions about how to get to where you're going with the robot, please feel free to contact our support team:

www.kinovarobotics.com/support

About this document

User Guide contents and warnings.

⚠ Read all instructions before using this product and any third-party options.

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

This document contains information regarding product setup and operation. It is intended for Kinova product end users.

All third-party product names, logos, and brands appearing herein are the property of their respective owners and are for identification purposes only. Their use in this document is not meant to imply endorsement by Kinova.

Kinova has made every effort to ensure that this document is accurate, accessible and complete. As part of our commitment to continuous improvement, we welcome any comments or suggestions at www.kinovarobotics.com/support.

From time to time, Kinova will make udpates to this document. To download the most up to date version of this document, visit the Kinova Technical resources page at www.kinovarobotics.com/knowledge-hub/all-kinova-products.

For general inquiries please contact us at +1 (514) 277-3777

Acronyms and abbreviations

API

Application Programming Interface

CIDR

Classless Inter-Domain Routing

CISPR

Comité International Spécial des Perturbations Radioélectriques

EE

End Effector

EMI

Electromagnetic Interference

FOV

Field of View

fps

frames per second

GPIO

General-Purpose Input/Output

HDMI

High-Definition Multimedia Interface

IC

Integrated Circuit

IEEE

Institute of Electrical and Electronics Engineers

I^2C

Inter-Integrated Circuit (bus)

I/O

Input / Output

IP

Ingress Protection or Internet Protocol

IT

Information Technology

ISO

International Organization for Standardization

LED

Light-Emitting Diode

n/c

no connection

NVRAM

Non-Volatile Random-Access Memory

PC

Personal Computer

Rx

Receiver

ROS

Robot Operating System

RPC

Remote Procedure Call

RPM

Revolutions Per Minute

RS

Recommended Standard

SSID

Service Set IDentifier

Tx

Transmitter

UART

Universal Asynchronous Receiver-Transmitter

UDP

User Datagram Protocol

USB

Universal Serial Bus

UL

Underwriters Laboratory

UV

Ultraviolet light

VLAN

Virtual Local Area Network

WEEE

Waste of Electrical and Electronic Equipment

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 caused 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 replaces the Products in accordance with this section, it will be deemed to 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 the End User. Kinova may 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 directives and warnings

Directives, warnings and safety considerations for the KINOVA® Gen3 *Ultra lightweight robot*.

IMPORTANT

Before operating the robot for the first time, ensure that you have read, completely understood and complied with all of the following directives, warnings and cautionary notes. Failure to do so may result in serious injury or death to the user, damage to the robot, or a reduction in its useful life.

Table 1: Safety

⚠ There is no mechanical brake on the robot. If the power supply is cut or an unrecoverable error occurs, be aware that the robot will fall. However, mechanisms are in place within the actuators will slow the descent in the absence of external power.	
⚠ For your personal safety, and that of others, it is strongly recommended that the following be carried out:	- risk assessment, before integration of the robot into a given application.
	- hazard analysis, before integration into an environment which includes atomized flammable dust / particles or explosive / flammable gases, etc.
⚠ For your personal safety, and that of others, **never**:	- use the robot near a flame or source of heat.
	- exceed the maximum specified payload.
	- attempt to stop the robot or prevent its movement by holding it (except in admittance mode).
	- install the robot base within 20 cm of your body (base contains a Wi-Fi transmitter)
	- use the robot to submerge objects in water.
⚠ For your personal safety, and that of others, **always** ensure that:	- the robot does not encounter any obstacles (persons or objects). Although inherently safe in its default configuration, disabling the robot safeties requires that the user be responsible for ensuring a secure working space.
	- the end effector never collides with a hard surface.
	- the grasping of objects by gripper fingers is stable, to prevent the risk of dropped or thrown objects (if using a gripper).
	- the wrist is supported before turning the power off (otherwise it may fall and cause damage).
	- eye protection is worn when manipulating fragile objects with the robot.
	- the working area is safe when containers of hot (or extremely cold) liquids are to be manipulated with the robot.
	- the robot has its base securely fixed to the work surface when in operation.
	- the robot working area is safe if sharp objects are to handled by the robot.
	- before using the robot, it is confirmed that there are no warnings.
	- the robot is protected adequately before being used near any messy process (e.g. welding or painting)

⚠ Do not power on the product if any external damage to the Vision module is apparent.
⚠ Do not attempt to open the Vision module.
⚠ The Vision module depth sensor includes a Class 1 infrared laser. To avoid eyesight injury from wide angle infrared laser light, do not view the front-facing surface of the Vision module through magnifying optical elements.
⚠ The robot should not be used without the provided emergency stop connected.
⚠ Do not operate the robot when the relative humidity exceeds the maximum specified limit. In such a case, put down any object in the gripper, bring the robot to a resting position and wait until the humidity decreases to an allowable value.
⚠ The robot is not certified for use in applications in sterile environments (e.g. food production, pharmaceuticals, medical, surgical).

Table 2: General

⚠ Do not connect the USB ports on the base to one another.
⚠ It is recommended that surge protection be used to protect the robot against external surges on the main AC line which might be caused by lightning or other abnormal conditions.
⚠ The base must be mounted as specified in the installation section, with particular attention to the bolt pattern, strength requirements and any table or tripod-specific mounting.
⚠ The end effector must be mounted as specified in the installation section (including bolt pattern, power requirements, etc.).

Table 3: Maintenance

⚠ Immediately following exposure to saline air conditions, contact Kinova support to schedule maintenance by authorized Kinova technician.
⚠ The controller mating interface must be kept free of dust and moisture to protect the electrical contacts. Wipe down the surface with a soft dry cloth to keep the surface of the interface clean.
⚠ Do not use the robot in heavy rain. If this happens, contact Kinova support to schedule maintenance by an authorized Kinova technician.

Disclaimer

KINOVA® and the Kinova logo are registered trademarks of Kinova inc., herein referred to as Kinova.

KORTEX™ is a trademark of Kinova inc.

All other brand and product names are trademarks or registered trademarks of their respective owners.

The mention of a product name does not necessarily imply an endorsement by Kinova. This manual is furnished under a lease agreement and may only be copied or used in accordance with the terms of such lease agreement. Except as permitted by the lease agreement, no part of this publication may be reproduced, stored in any retrieval system, or transmitted, modified 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 and is subject to change without notice. It 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 and will be incorporated into new editions of this publication. Kinova may make improvements and/or changes to the products and/or software programs described in this publication at any time.

Any questions or comments concerning this document, the information it contains or the product it describes may be addressed through the support page on the Kinova website:

www.kinovarobotics.com/support

Kinova would like to thank you for your contribution, while retaining the right to use or distribute whatever information you supply in any way it believes appropriate (without incurring any obligations to you).

Risk assessment

Before proceeding it is imperative that a risk assessment be performed (note that this is required by law in many countries). As it is a machine, the safety of the robot depends on how well it is integrated with its environment and with other machines.

The recommended international standards for conducting a risk assessment are as follows:

- ISO 12100
- ISO 10218-2

The risk assessment should take into consideration all activities carried out in the context of the robot application, including (but not limited to):

- teaching the robot (during set-up)
- development of the robot installation
- robot troubleshooting
- robot maintenance
- everyday robot operation

The risk assessment must be completed **before** integration of the robot in an application and should address configuration settings as well as the need for any additional emergency stop buttons.

Normal use definition

This section describes the normal use of the robot.

The definition of *normal use* includes lifting, pushing, pulling, or manipulating (without a gripper or other end effector attached) a maximum load of:

- mid-range, continuous: 4 kg
- full-reach, temporary: 4.5 kg
- full-reach, continuous 1.1 kg

The robot is designed to hold, move, and manipulate objects in the user environment. However, for some loads in certain positions (near maximum load and reach), holding an object for an extended period of time may result in heating. To protect the robot hardware from excessive heat, safety thresholds shut down the robot if the temperature rises above a certain threshold.

Before this is reached, an API notifcation will be rendered as a user alert on the KINOVA® KORTEX™ *Web App*.

The robot includes a number of temperature-related safeties:

- base - CPU core and ambient temperatures
- actuators - CPU core and motor temperatures
- interface module - CPU core and gripper motor temperatures

If you receive any temperature warnings, put down any object as soon as is practical and place the robot into a stable rest position to allow it to cool down.

During normal operation, the robot joints are subject to heating. The joints are normally covered in plastic rings to protect the user from the metal surfaces which may become hot.

Getting started

This section describes how to get started with the arm.

The pages that follow lead you through getting started with the robot. This includes:

- unboxing
- physically mounting the robot securely
- provisioning electrical power
- controlling the robot using an Xbox gamepad
- moving the robot in admittance using physical buttons
- connecting a computer to the robot
- connecting to the KINOVA® KORTEX™ *Web App*

What's in the case?

This section describes the KINOVA® *Gen3 Ultra lightweight robot* shipping case contents.

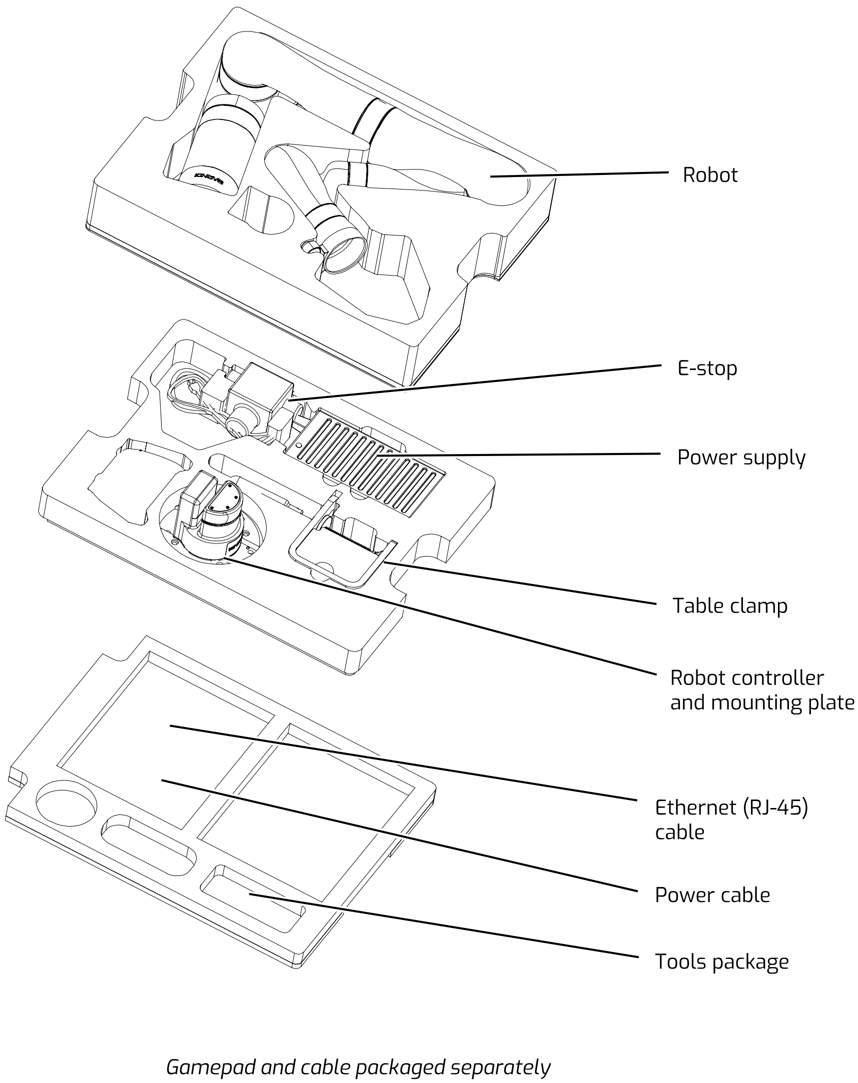


Figure 1: shipping case contents

The shipping case contains the following contents.

At the top of the interior of the box, you will find the **Quick Start Guide**. The Quick Start Guide is a large visual guide printed on sturdy cardboard.

The Quick Start Guide provides a handy reference for first steps, and should have you up and running within 30 minutes. Make sure to keep the Quick Start Guide as a reference for people in your team or organization getting newly acquainted with your robot. The Quick Start Guide is also available on the Kinova website Knowledge Hub:

www.kinovarobotics.com/knowledge-hub/all-kinova-products

The contents of the box are arranged in three layers from top to bottom. These packing layers can be removed from the box to unpack the contents.

In the top layer:

- Robot

In the second layer:

- Power adapter and cable with integrated emergency stop (E-stop) button
- Mounting plate and robot controller

The bottom area contains:

- Ethernet (RJ-45) cable
- Power cable
- Bag with useful tools and fasteners
    - hex keys: 3, 4 and 5 mm
    - M5 x 40 mm screws (qty. 4)

An Xbox gamepad and cable are shipped with the robot, but packaged separately.

There is also space for storage of papers and other items.

Note: The shipping case is also useful for transportation and storage of the robot. Make sure to save it and the packing layers within for future use.

Manipulating the robot joints when the robot is powered off

This describes how to manipulate the robot joints when the robot is powered off.

When the robot is powered on and not in admittance mode, the actuators will hold their position and prevent the joints from moving in response to external forces and torques. When the power is on, the arm will not move except when commanded. The arm joints are stiff and you will not be able to rotate the joints with your hands.

When the robot is powered off, as it is when you first receive the robot, the joints can be moved by hand slowly.

Note: If you move the joints too quickly, you will hear a mechanical noise, and feel resistance. This is a a passive mechanism within the large actuators designed to slow the descent of the robot when the power is cut. This is only triggered if the joints turn too quickly, above a certain threshold with the power off. If you turn slowly, this will not be triggered, and you will be able to move the joints by hand freely.

This moveability of the joints when the robot is unpowered is useful when taking the robot out of the box and setting it up to get started. This lets you arrange the joints of the robot into a stable, balanced position prior to mounting and powering on the robot.

Robot mounting options

This section describes the physical mounting options for the robot.

The first step to getting started with the arm after unboxing is to physically mount the arm in a stable manner so that the arm can be connected and used.

The most basic mounting option uses the mounting plate clamped to a tabletop, but it is possible to mount the arm in different ways, depending on the needs of your particular application.

Mounting the robot on a tabletop

This section describes the procedure for mounting the robot oriented vertically on the edge of a tabletop using the clamp.

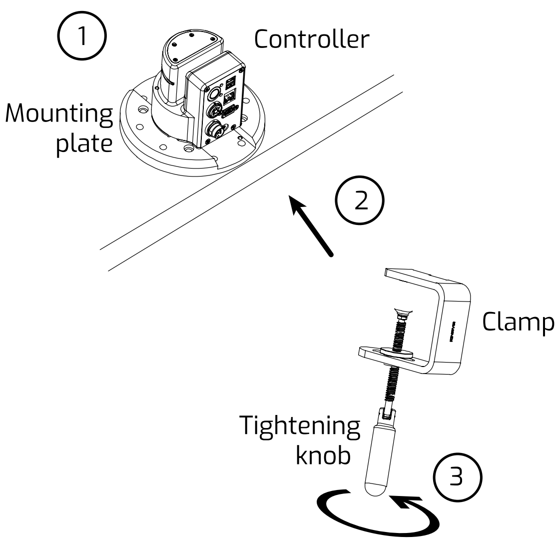
Before you begin

The robot should have the joints of the robot unfolded so that it is in a stable, balanced position ready for mounting.

About this task

The robot is mounted to a tabletop using the base mounting plate and a table clamp.

Note: The table must be large and sturdy to support a tabletop edge mounting. If the table is too small or too flimsy, the weight of the robot at the table edge combined with the movement vibrations may render it unstable.

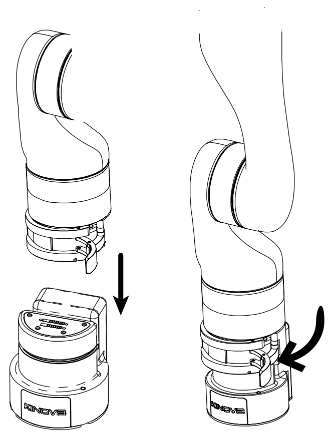


Procedure

1. Place the base controller and mounting plate on the tabletop, next to the edge.

   Note: You can place the controller in one of two orientations. Either the connector panel facing out toward the edge of the table, or the front side of the base controller facing out.
2. Turn the tightening knob on the table clamp to open up the clamp and then slide the clamp into the slot between the mounting plate and the bottom of the base controller.
3. Turn the tightening knob by hand until the mounting plant is firmly clamped to the table top.

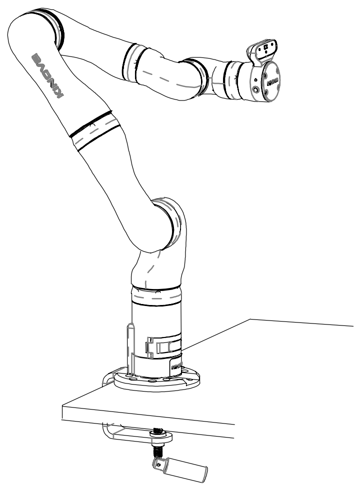
   Note: Do not overtorque.
4. Make sure that the clamp at the bottom of the robot is opened. While holding the robot, you



5. Once the robot is fully lowered onto the base controller, close the clamp to secure the robot in place on the base controller..

Results

The robot is now mounted on the tabletop.



What to do next

You can now proceed to connect the robot to the power supply and E-stop.

Mounting the robot on a horizontal surface without the table clamp

This section describes how to mount the robot on a horizontal surface without the table clamp..

About this task

Here, we describe mounting the robot in a vertical orientation on a flat, horizontal surface, affixing the mounting plate or controller base to the surface using screws and sunk holes in the surface.

Procedure

1. Choose whether to mount the robot base controller directly onto the surface, or whether to use the mounting plate.
2. Using either the mounting plate bolting pattern or the controller bolting pattern or as a guide, drill holes into the surface. If the controller is to be mounted directly to the surface, the holes will have to be drilled all the way through the mounting surface.

3. Use appropriate screws to mount either the base controller or the mounting plate to the surface. If the base controller is mounted directly, the screws will need to go through the mounting surface from the other side.

Controller mounting plate bolting pattern

This section describes the bolting pattern of the mounting plate. This is useful when mounting the robot to a surface using the mounting plate.

Overview

The mounting plate is attached to the bottom of the base controller. The mounting plate has two sets of M8 screw holes (4) and one set of counter-sunk M6 screw holes (4) available for mounting the plate to a surface.

Mounting details

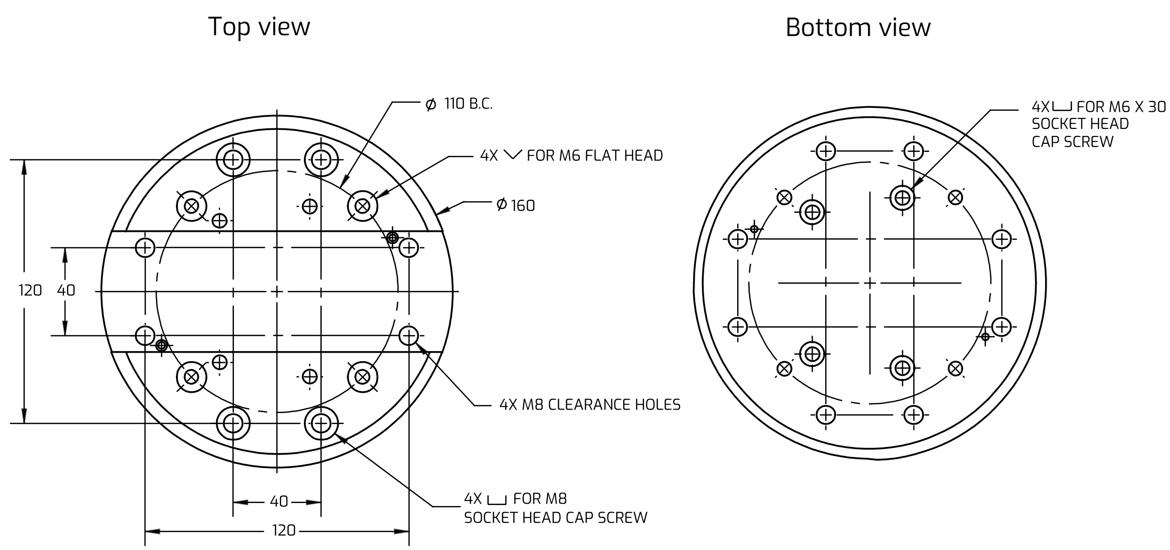


Figure 2: Mounting plate bolting pattern

Base controller underside bolting pattern

This section describes the bolting pattern on the underside of the base controller. This is useful when you want to affix the robot base directly to a surface.

Overview

The underside of the controller has four M6 screw holes for mounting purposes. These holes are used for attaching the mounting plate to the controller. When the mounting plate is removed, these holes can be used for mounting the controller directly to a surface. In that case, holes must be drilled through the surface so that screws can go through from the other side and into the controller mounting holes from underneath.

One of the screw holes in the controller base features an inset locking screw. Turning the locking screw clockwise to the end of its travel (using a 3 mm hex key) while the base shell is clamped to the controller will lock the two together and prevent the clamp from being opened.

Mounting details

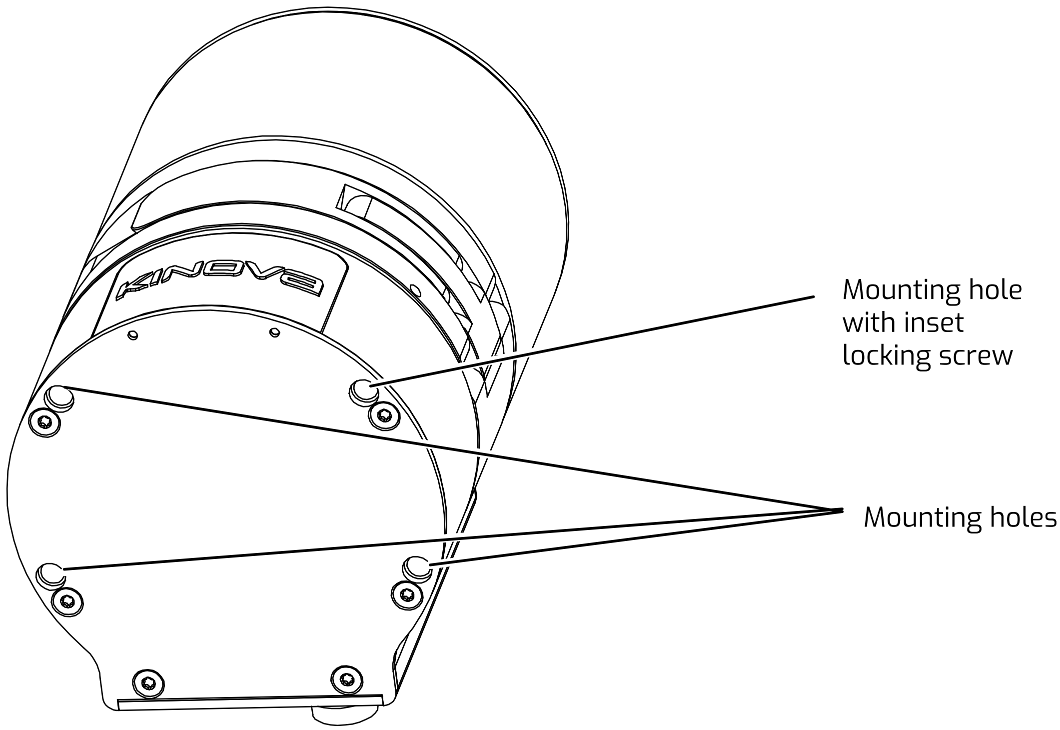


Figure 3: Base and mounting holes

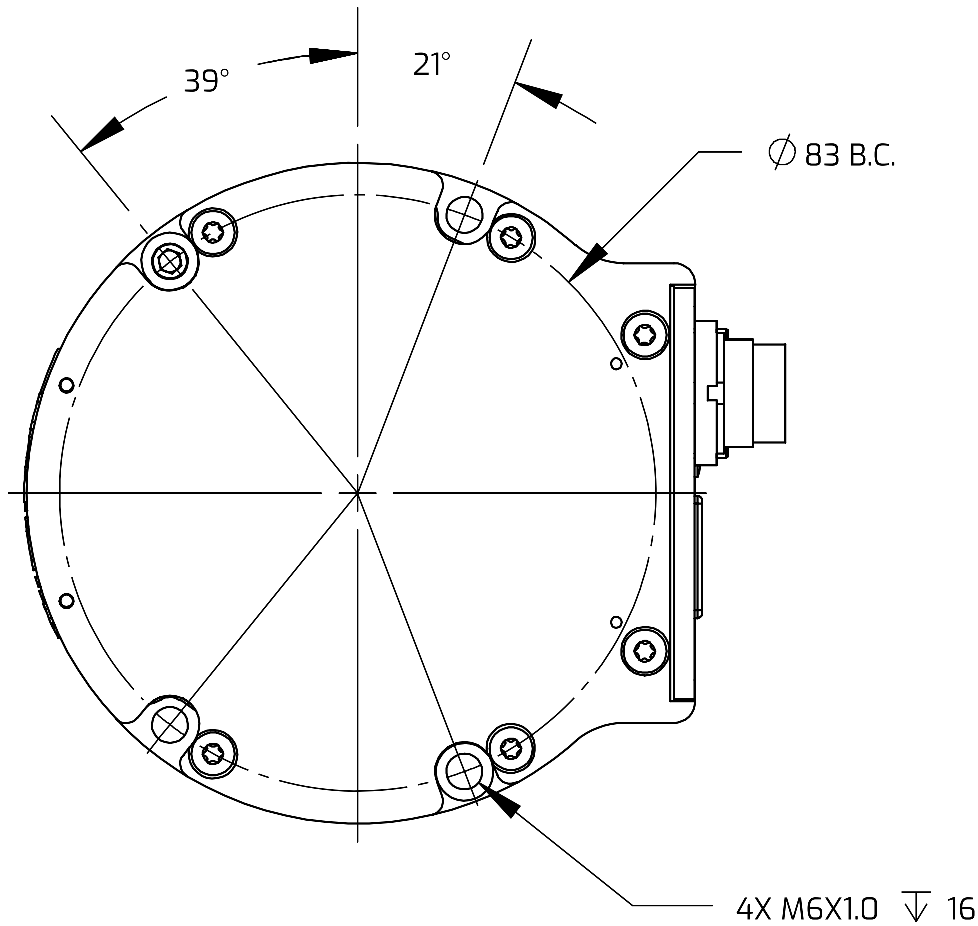


Figure 4: Base mounting holes pattern

Robot power adapter and E-stop

This section describes the power adapter and E-stop.

The power adapter allows power to be supplied to the robot using a wall outlet as a source. The cable from the power adapter connects to the power connector on the base controller using a Lumberg 0322 08 connector.

The cable from the power adapter to the robot includes an integrated push-button E-stop connected in series. The E-stop allows users to shut down the robot quickly in case of an emergency.

To engage the E-stop, press down on the red button on top of the E-stop. This will cut the power to the robot, causing it to shut it down.

⚠ When the power is cut, the robot will descend. There are mechanisms within the large actuators to slow the fall of the arm for safety purposes. However, it is recommended that if possible, users cradle the robot as it falls.

To disengage the E-stop, rotate the button clockwise until it pops up.

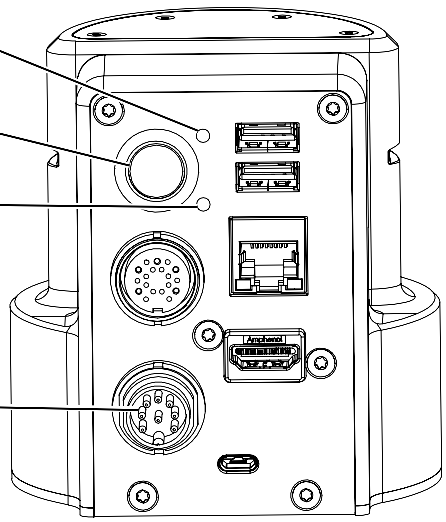
Powering on the robot

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

The robot is powered by a 24V power supply (P/N DTM300PW240D2).

To power up the robot:

1. Connect the captive cable from the power supply to the circular Lumberg connector on the rear connector panel in the controller of the robot, rotating the outer cylindrical locking shell of the connector until it is just tight enough to secure the connector.
2. Plug the power supply into a wall outlet.
3. Push the power button and hold in for 3 seconds to power up the robot. This will initiate the power up sequence.



Note: When the robot is properly powered on, the blue power LED will be illuminated green.

Note: Do NOT hold the power button down for too long. Holding the button for 10 seconds will result in a factory reset.

Power-up, booting, and initialization sequence

This section describes the LED indications during the power-up sequence.

When the power button is held in to initiate a power-up, the robot will go through a regular boot up and initialization sequence.

The base LEDs will provide visual feedback as to the progress through the sequence, as follows:

Table 4: Power-up sequence LEDs indications

Sequence step	LEDs indications
System booting	• Blue power LED, blinking • Status LED, off
System initializing	• Blue power LED, solid • Status LED, amber, solid
System operating normally	• Blue power LED, off • Status LED, green, solid

From start to finish, the process should take no more than 30 seconds, except during a firmware update.

Resetting the robot to factory settings

This section describes how to reset the robot to factory settings.

About this task

At some point you may find it useful or necessary to roll back configurations on the robot to factory defaults. This will return the robot to the state it was in when you received the robot.

Note: This procedure assumes you are starting with the robot powered on. If the robot is already powered off, you can start at step 3.

Note: Be sure that that this is what you want to do. This will erase all user-defined configurations including protection zones, network settings, actions, user profiles, etc.

Procedure

1. Place the robot in a stable position.
2. Press and hold the base power button to power off the robot.
3. Press and hold the power button **for 10 seconds**.
4. The green status LED will go on to confirm the factory reset.
5. Release the power button. The robot will then boot with factory default configuration settings.

Operating the robot

This section gives an overview of the methods of controlling the robot.

There are three ways to operate the robot:

- physical gamepad (Xbox controller)
- virtual joysticks over a network connection (KINOVA® KORTEX™ *Web App* virtual joysticks)
- programmatically (KINOVA® KORTEX™ API)

Supported gamepad controllers

This section describes the supported gamepad controllers of the robot.

The robot currently supports the Xbox gamepad (USB wired connection only; Bluetooth for future development).

Connecting an Xbox gamepad to the robot

This section describes how to connect an Xbox gamepad to the robot.

Before you begin

You will need:

- Xbox gamepad
- micro USB to USB-A cable

About this task

An Xbox gamepad can be used to operate the robot.

Note: The robot currently only supports a wired connection for the gamepad.

Procedure

1. Connect the micro USB connector plug of the cable into the micro USB port on the Xbox gamepad.
2. Connect the USB-A end of the cable into one of the two USB-A connectors on the base controller of the robot.

Default gamepad control mappings - Xbox gamepad

This section describes the default controller mappings between the Xbox gamepad and the actions on the robot.

Gamepad mappings overview

The robot has three default control mappings for the Xbox gamepad.

1. Twist linear (controls the robot by velocity)
2. Twist angular (controls the robot by velocity)
3. Joint (controls the robot joint by joint by velocity)

General controls

Some controls apply the same across all mappings. These are controls for:

- Entering an admittance mode
- Changing the active control mapping to the next or previous mapping in the list
- Opening and closing the gripper
- Clearing faults - a fault state will make itself known through a red LED on the base controller of the robot. Pressing the left bumper clears the fault and returns the LED to green.
- Applying emergency stop - this will stop the robot.
- Reaching home or retract position

The available control mappings are in a sequential list, starting with **Twist linear** and ending with **Joint**, as listed above. Pressing the View or Menu buttons will cause the active control mapping to switch to the previous or next control mapping on the list. The list can be thought of as circular - selecting previous when on the first mapping will cycle around to the last mapping, and conversely, selecting the next mappingwhen on the last mapping will cycle around to the first.

Table 5: General control mappings (common controls applying to ALL mappings)

Action		Control	
Reach defined pose	Retract pose	A (hold down)	button
	Home pose	B (hold down)	
Navigate controller mappings	previous	View button	
	next	Menu button	
Gripper command	close	Left	trigger
	open	Right	
Clear fault		Left	bumper
Stop robot		Right	

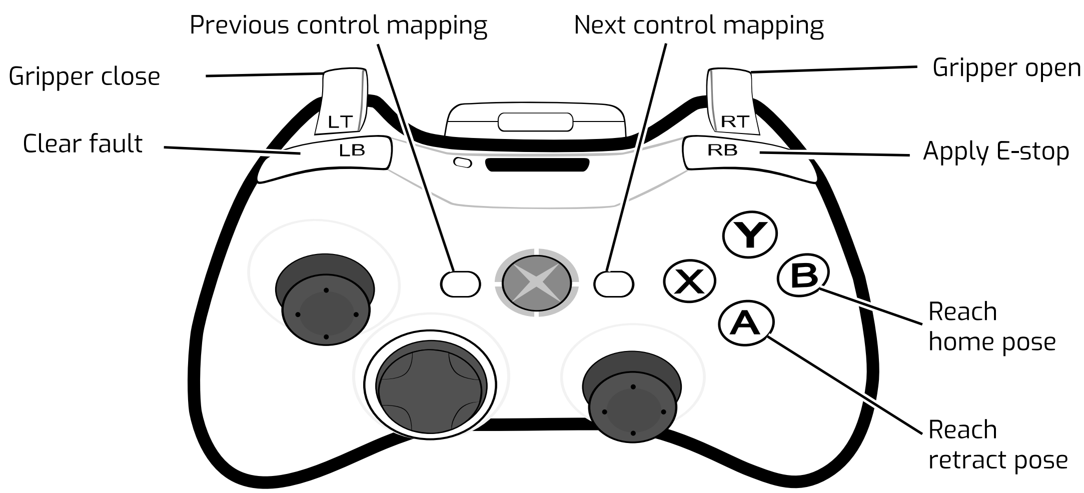


Figure 5: General control mappings with Xbox gamepad

Twist linear mapping

Twist linear is the default gamepad mapping when the robot is turned on and the controller is connected. In this mode the end effector is translated in space with respect to the Cartesian base frame. The end effector orientation does not change in this mapping. The user controls the linear velocity of the end effector, including the linear speed.

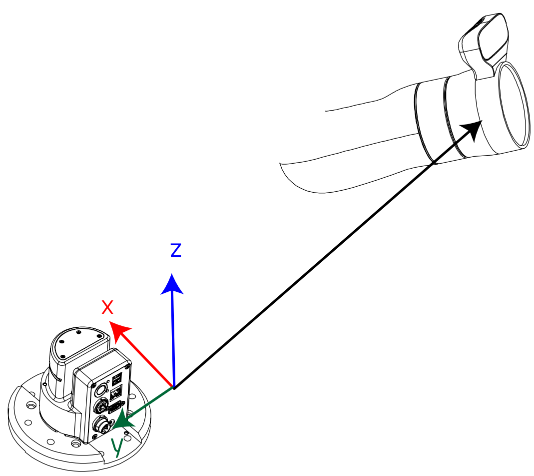


Table 6: Twist linear - general controls plus:

Action		Control	
Toggle admittance	Cartesian	X	button
	Nullspace	Y	
Cartesian translation command Y	+	left	Left stick
	-	right	

Action		Control	
Cartesian translation command X	-	down	
	+	up	
Cartesian translation command Z	-	down	Right stick
	+	up	
Speed	decrease	down	D-pad
	increase	up	

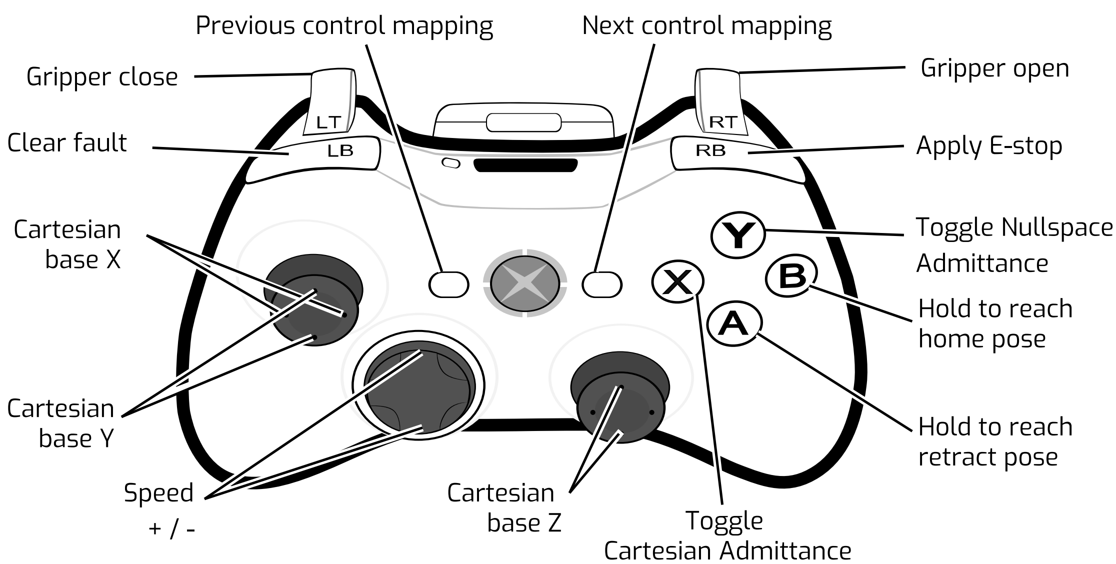
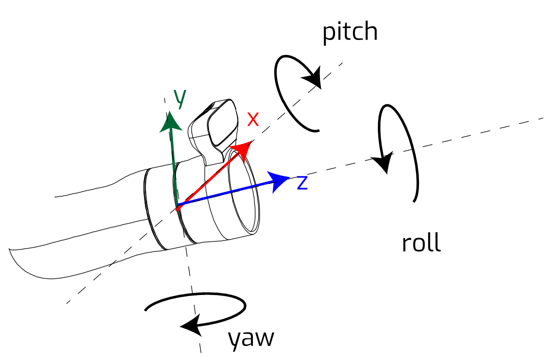


Figure 6: Twist linear controls with Xbox gamepad

Twist angular mapping

Twist angular can be thought of as a companion to the Twist linear control mode. In Twist linear, the end effector is translated with respect to the base reference frame while leaving the orientation unchanged. In Twist angular, the control is pure rotation of the end effector within the end effector reference frame, around the three axes of that frame. The user controls the angular velocity of the end effector in relation to those three axes.



Twist linear and Twist angular together specify a twist (consisting of three linear velocity terms and three angular velocity terms) to be applied to the end effector (Cartesian control).

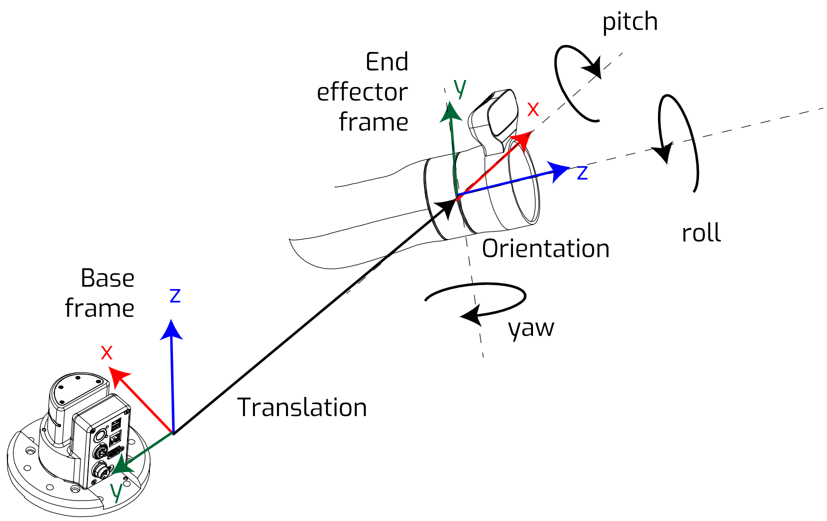


Table 7: Twist angular - general controls plus:

Action		Control	
Toggle admittance	Cartesian	X	button
	Nullspace	Y	
Cartesian rotation command Y (yaw)	yaw right	left	L stick
	yaw left	right	
Cartesian rotation command X (pitch)	pitch up	down	
	pitch down	up	
Cartesian rotation command Z (roll)	roll left	left	R stick
	roll right	right	
Speed	decrease	down	D-pad
	increase	up	

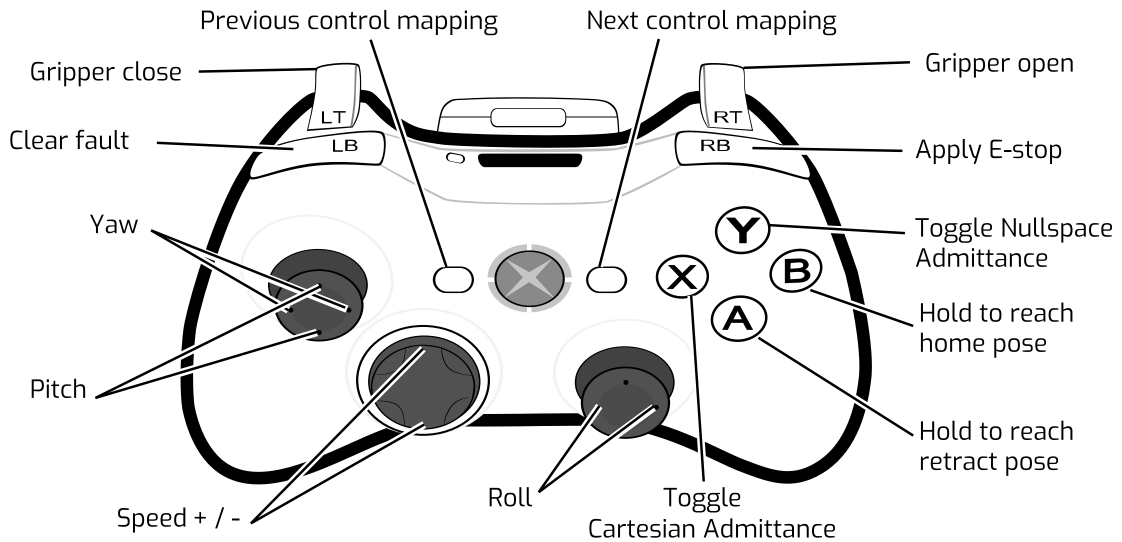


Figure 7: Twist angular controls with Xbox gamepad

Joint mapping

Joint control offers direct control of the rotational movement of the joint actuators. In this mode you can toggle through the joints (actuators) one by one, starting with the first and going through in increasing order. On reaching the last actuator, it will then cycle back to the first. The joint angular speed (ω) can be controlled.

Table 8: Joint - general controls plus:

Action		Control	
Toggle admittance	Joint	X	button
Joint speed	ω-	left	L stick
	ω+	right	
Speed	increase	up	D-pad
	decrease	down	
Navigate joints	Previous	left	
	Next	right	

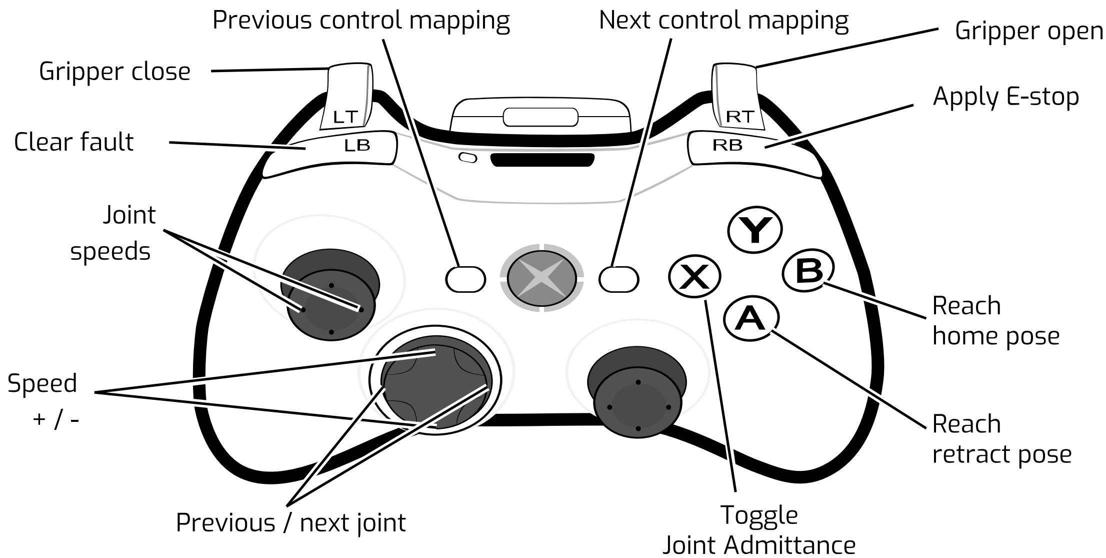


Figure 8: Joint controls with Xbox gamepad

Putting the robot into admittance using the interface buttons

This section describes how to put the robot into admittance modes using the buttons on the sides of the interface module.

The interface module has two buttons on its side that can be used to temporarily put the robot into admittance. This can be a convenient way to take ahold of the robot and move it into a desired position, or to explore the flexibility of the arm at a particular position.

The two interface module buttons each offer access to one admittance mode.

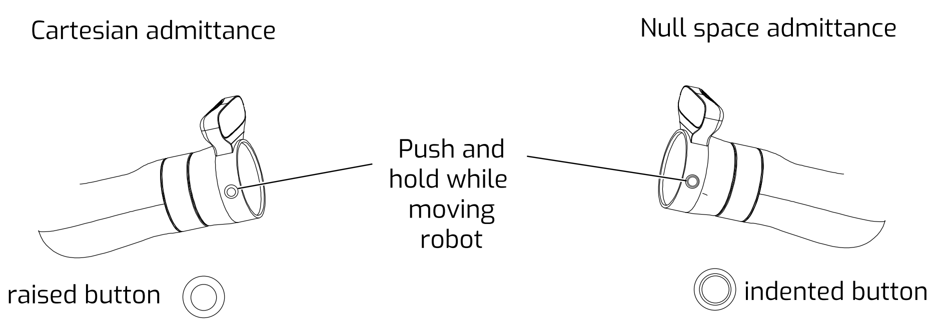


Figure 9: Interface module admittance buttons

The button with the raised solid circle shape is for **Cartesian admittance**, in which the end effector of the robot moves in response to force exerted on it.

The button with the indented or ring shape is for **null space admittance**. In this mode the end effector stays in a fixed position and orientation, while the other joints move within the null space available at the given end effector (seven degrees of freedom to specify six coordinates of position and orientation gives a free degree of liberty to move within different solutions of the inverse kinematics of a given pose).

To engage one of the admittance modes, hold down the button and exert a moderate amount of force on the robot. The arm will be in admittance mode as long as the button is held down.

When the button is released, the robot will no longer be in admittance mode and will return to the previously engaged control mode.

Connecting a computer to the robot

This section gives an overview of the methods available to connect a computer to the robotic arm.

There are two ways of connecting a computer to the robotic arm:

- Ethernet (direct or over a small local network)
- Wi-Fi

Connecting a computer to the robot via Ethernet (for the first time)

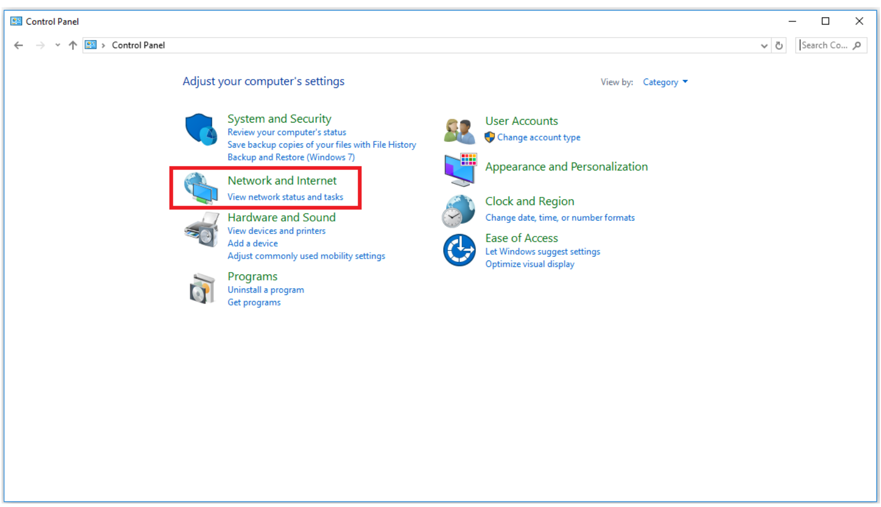
This section describes the procedure to connect a computer to the robot via a wired connection for the first time. This procedure requires some configuration of the computer's network adapter.

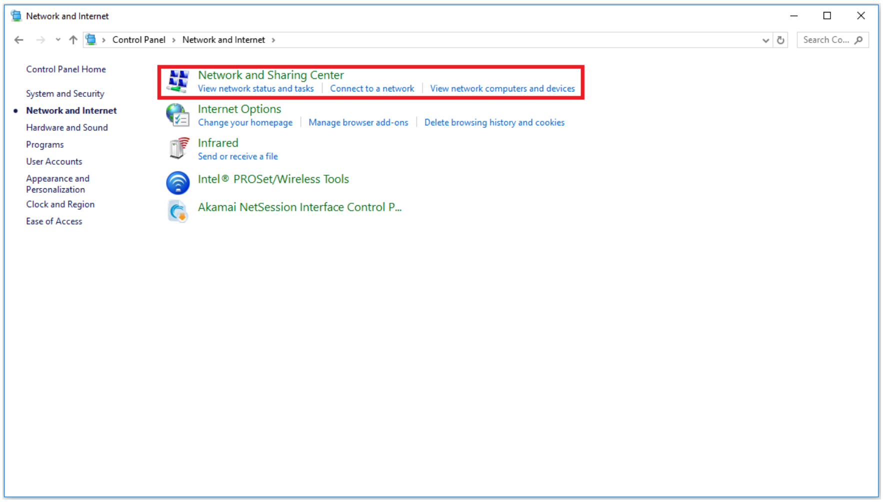
About this task

This procedure is required to connect a computer to the computer for the first time via a wired Ethernet connection. This requires some configuration of the computer. The following procedure describes details for Windows 10. The details will be somewhat different for other OS platforms, but the high level steps will be the same.

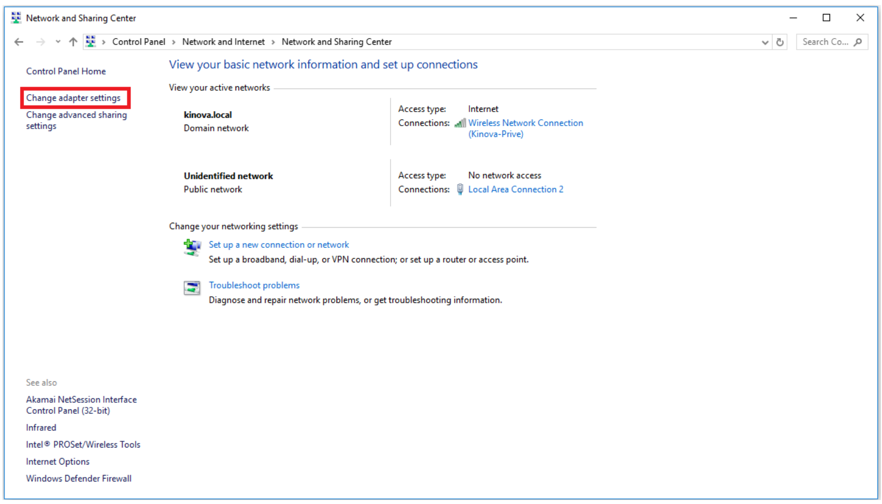
Procedure

1. Connect an RJ-45 Ethernet cable from your computer's wired network adapter to the base controller Ethernet port.
2. On your computer, open **Control Panel** > **Network and Internet** > **Network and Sharing Center**

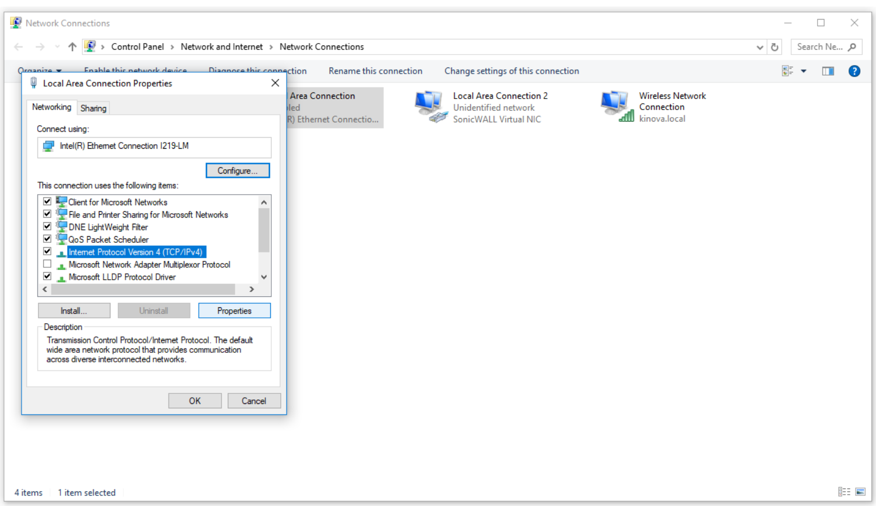




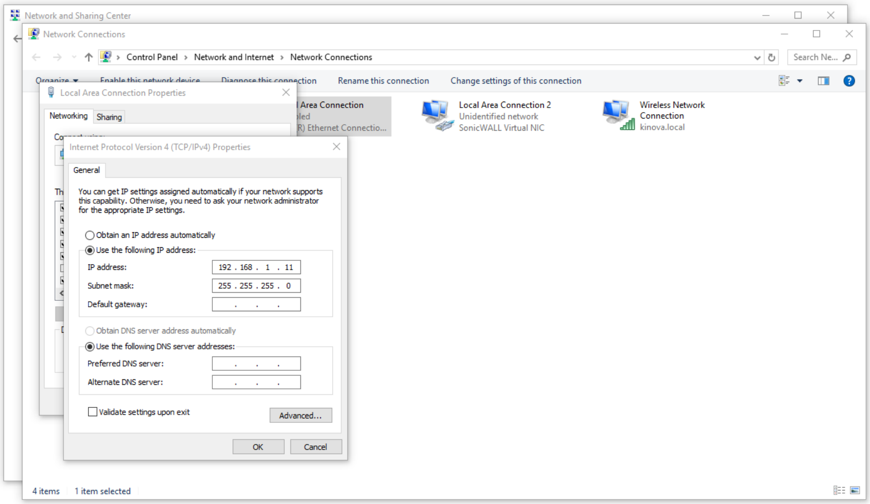
3. Select **Change adapter settings**



4. Select wired Ethernet adapter (i.e. **Local Area Connection**) and choose **Properties**.
5. Select **Internet Protocol Version 4 (TCP/IPv4)** and choose **Properties**.



6. Select **Use the following IP address** and enter IPv4 address:`192.168.1.11` and Subnet mask: `255.255.255.0`



7. Press OK.

Results

Your computer is now connected physically to the robot and ready to communicate.

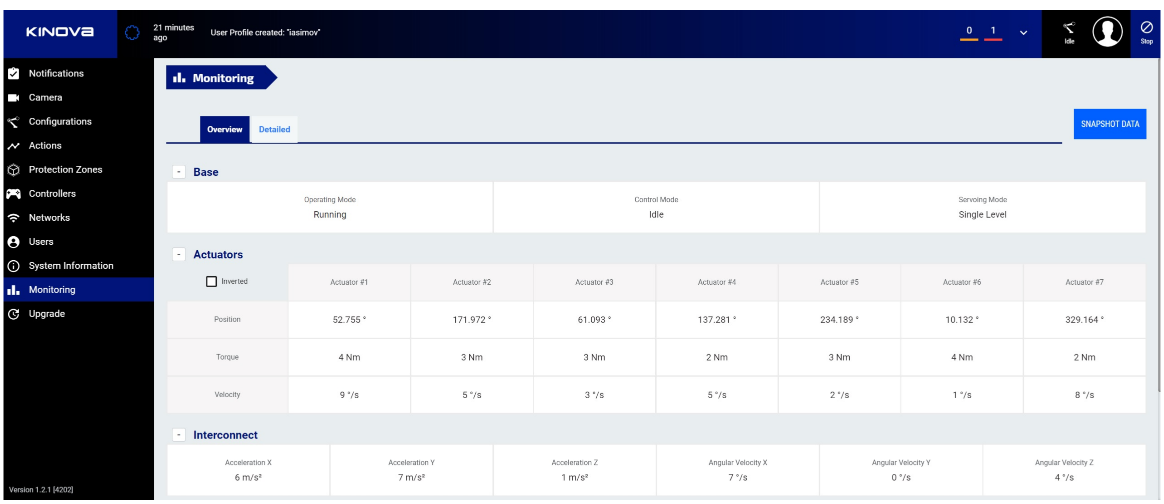
What to do next

You can now access the *Web App*.

KINOVA® KORTEX™ *Web App*

This section gives an overview of the KINOVA® KORTEX™ *Web App*.

The KINOVA® KORTEX™ *Web App* provides a HTML Web browser based GUI to interact with the arm and perform basic tasks without using programming commands.



The *Web App* allows users to control and configure the robot via the GUI.

This includes:

- Real-time control of the robot in different modes using different virtual joysticks
- Setting the arm into admittance modes to manipulate the arm using external forces / torques
- Viewing the feed from the Vision module color sensor
- Configuring
    - robot performance parameters and safety thresholds

    - protection zones
    - network settings
    - backup management
    - user profiles
- Reading
    - system information
    - notifications
- Defining robot poses and trajectories
- Managing control mappings for physical controllers
- Monitoring robot parameters
- Upgrading the robotic arm firmware

The *Web App* can be run from either a desktop / laptop PC connected by wired Ethernet to the arm, or from any computer on the same local network. This includes local Wi-Fi neworks. The *Web App* is a responsive web application, and can be run from both mobile devices (smartphone or tablet) or desktop computers.

The Web app is described in detail in the KINOVA® KORTEX™ *Web App* User Guide section.

Accessing the KINOVA® KORTEX™ *Web App*

This section describes how to launch the *Web App*.

Before you begin

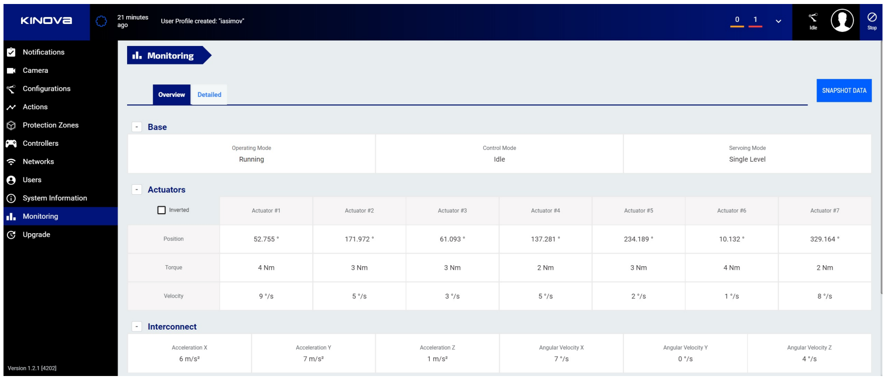
You should be using a computer that is connected to the robot either over a wired (direct or over local area network) or wireless connection and you should have the IP address of the robot on the network over which you are connected.

About this task

Procedure

1. From the computer web browser, enter the appropriate IP address for the arm base to access the *Web App*.

   Note: By default, the IP address to use here is `192.168.1.10`. If you have configured the arm so that the computer and arm are both connected to the same local area network, whether wired or over Wi-Fi, use the new configured IP address.
2. If the connection between the arm and computer is configured correctly, the Web application should launch and present a login window. In the login window, enter the following credentials:
    - username: admin
    - password: admin
3. Click CONNECT. The application will initialize. If all is successful, the application will open to a Monitoring screen that displays live parameters for the robot.



Changing the robot wired connection IP address

This section describes how to change the robot's wired connection IP address.

Before you begin

You need to have already connected the computer to a wired network connection to configure the robot for a wired connection (you will need information about the available IP addresses on your local area network).

About this task

This procedure is used to configure the robot so that you can connect a computer to the robot remotely over your local area network.

Note: For security reasons, we do not recommend connecting the robot to a WAN. The network should be a simple local area network with low traffic.

Procedure

1. Open the Web application and go to the Networks page. Open the Ethernet tab.
2. Modify the IPv4 address, IPv4 subnet mask, and IPv4 gateway to match an available IP address with the IP address range of your network.

   Note: Once you modify the robot network parameters, your client computer will lose connection with the robotic arm.
3. Physically disconnect the robot from your computer and connect it via Ethernet cable to your local network at a network switch.
4. Restore appropriate local network settings on your computer and connect your computer to your network.
5. From your computer, ping the robot at the newly configured robot IP address.

What to do next

From your computer web browser, enter the new robot IP address to access the Web App.

Connecting a computer to the robot via Wi-Fi

This section describes the procedure to connect a computer to the robot via Wi-Fi .

Before you begin

You will need to have a wired connection between the computer and robot to carry out this procedure.

About this task

The robot features a Wi-Fi adapter. This allows the arm to connect to a local Wi-Fi connection. Once this connection is established, other devices on the same Wi-Fi network can then connect to the robot wirelessly.

Procedure

1. On a computer connected to the arm via Ethernet, open the *Web App* and connect to the arm.
2. Select Networks in the main navigation panel of the *Web App* to go to the Networks page.
3. Select the Wi-Fi tab.
4. The Wi-Fi tab will list all of the detected Wi-Fi networks. Choose one of the networks, and click the corresponding **Connect** text button.

   Note: It is not recommended to connect to Wi-Fi networks which are potentially insecure. Security settings of at least WPA2 are recommended.
5. A pop-up window will appear to sign in to the network, with information about the signal strength and security settings. Enter the password for the network and click the CONNECT button. Take note of the IPv4 address that the robotic arm obtains after clicking the CONNECT button
6. On any wireless device connected to the same Wi-Fi network, open a Web browser and type the IP address that the robot obtained at Step 5 (This address corresponds to the robot's address on the Wi-Fi network).
7. At the Login screen, enter the appropriate user name and password, and click the CONNECT button.

Results

You are now connected to the Web Application through the Wi-Fi network adapter of the robot. You can now configure, monitor, and control the robot wirelessly.

Warning: A Wi-Fi connection is **not recommended** for 1 kHz (low-level) control of the robot due to potential latency issues - a wired connection must be used for this purpose.

Robot components

This page describes the main components of the KINOVA® Gen3 *Ultra lightweight robot.*

The robot consists of:

- base (base shell and controller)
- actuators
- interface module
- vision module

The following image shows the main components of the robot.

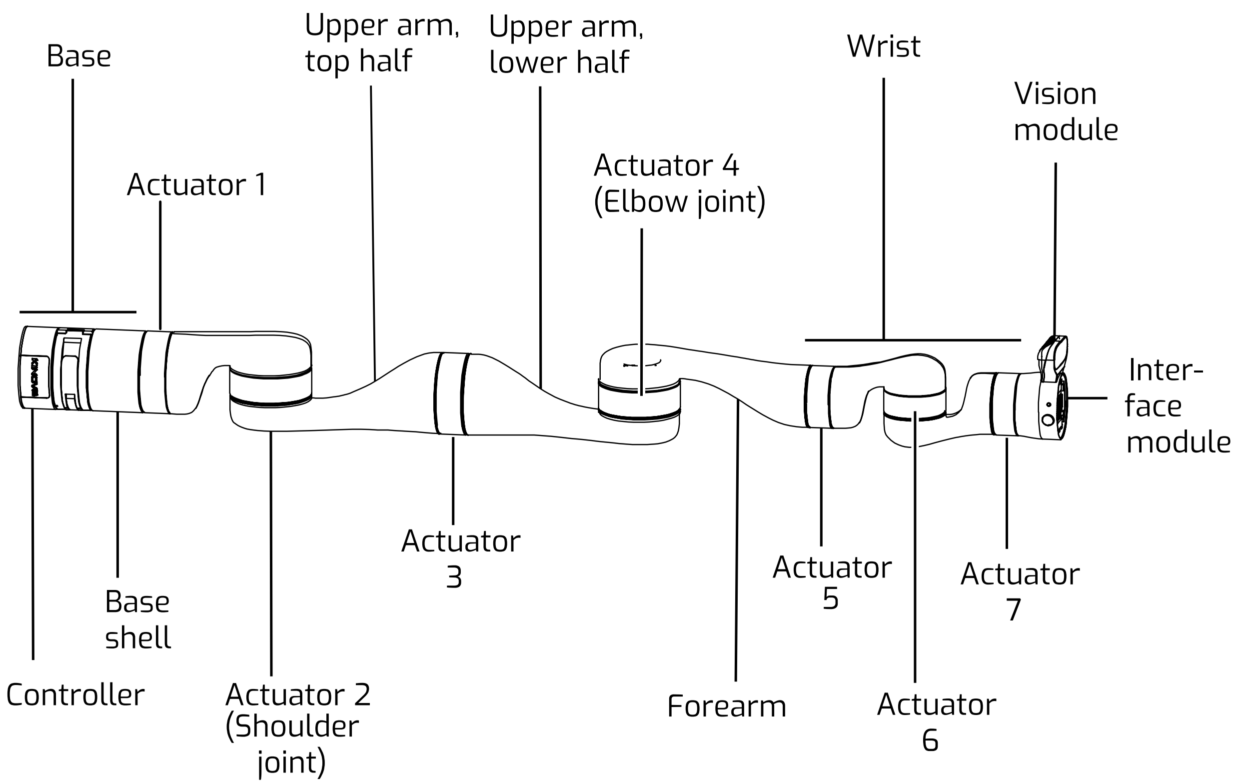


Figure 10: Robot main components (7 DoF model shown)

Base

This section describes the purpose, components, and functionalities of the robot base.

The robot is secured onto its physical mounting point and connects to power and control signals using a two-part structure consisting of:

- base shell
- controller

The base shell is the bottom part of the robotic arm shell connected to the first actuator. It mounts onto the controller and is secured in place with an integrated clamp.

The controller includes a connector panel at the rear for connecting to power and external devices. The controller has four mounting holes (M6) on its underside. The controller is shipped connected with screws to a circular mounting plate with through holes for mounting to surfaces and a slot to put a clamp between the robot and the plate for tabletop mounting.

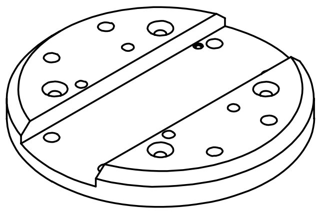


Figure 11: Mounting plate

The mounting plate can also be removed from the controller by removing the screws, giving access to the four mounting holes and allowing the controller to be mounted directly on the surface.

A mating interface to which the base shell connects establishes an electrical connection between the base shell and the controller. The controller also contains the key components for the control and connectivity of the arm.

Note: The controller mating interface needs to be kept free of dust and moisture to protect the electrical contacts. Wipe down with a soft dry cloth to keep the interface clean.

The internal components of the controller include:

- CPU
- Wi-Fi / Bluetooth adapter (Only Wi-Fi is used at present)
- Ethernet switch
- USB hub
- temperature sensor
- accelerometer/gyroscope

A Linux web server runs on the controller and manages connectivity between the controller and the arm devices, and between the controller and an external computer.

Controller quick connect system

This section describes the controller quick connect system.

The base of the robot is equipped with a quick connect system that allows for simple connect / disconnect of the base shell and controller. This allows the arm to be quickly detached from the mounting point of the arm without disconnecting any cables. This can be useful for transport, for removal of the arm for servicing or for convenient re-siting of the arm between multiple installation sites.

Note: Be careful to avoid damage to the electrical contacts on the mating interface of the controller when the base shell is disconnected. Make sure to keep the surface dry and free from dust.

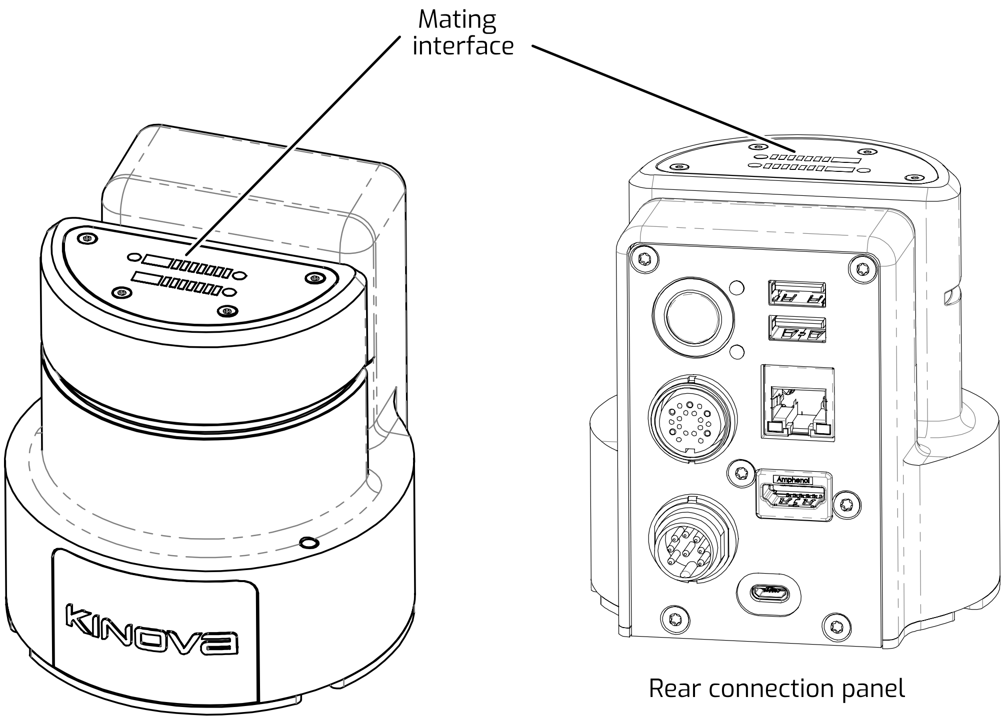


Figure 12: Controller quick connect

The base shell slides over and onto the controller, establishing an electrical connection with the arm. The base shell is secured in place on the controller by closing the front clamp.

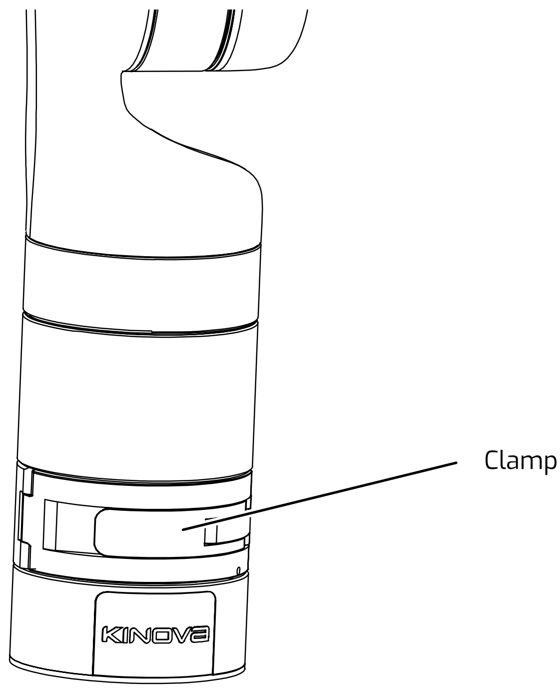


Figure 13: Base shell installation

To remove the arm from the controller, flip open the clamp and slide off the base shell.

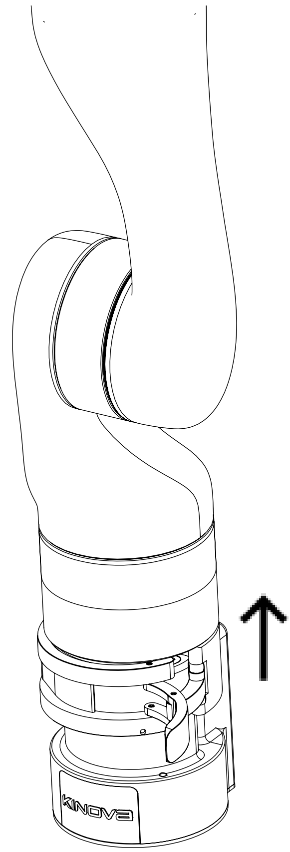


Figure 14: Base shell and arm removal

The controller features a locking screw within the mounting hole on the front bottom left (from the perspective of an observer behind the connector panel). Turning the locking screw with a 3 mm hex key clockwise will cause the screw to go forward and protrude through a hole above the top surface of the controller a few mm until it reaches the end of its travel. If the base shell is already clamped onto the controller when this is done, the set screw will interface with a mechanism on the clamp and prevent the clamp from opening until the set screw is withdrawn. This serves as a safety mechanism. There is a hole on the clamp where the end of the lock screw can be seen when it is fully engaged. Confirm visually that the lock screw is not engaged before trying to open the clamp.

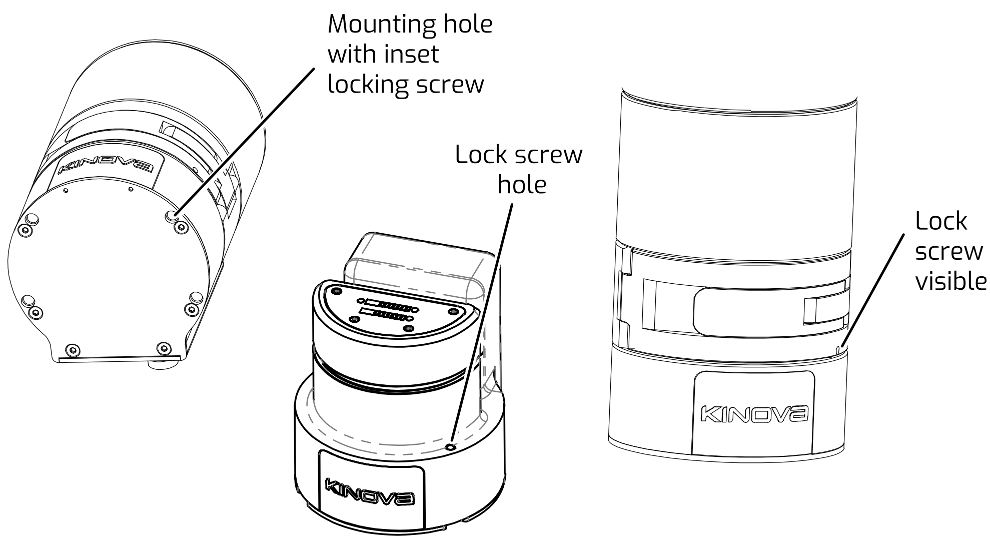


Figure 15: Lock screw mechanism

Controller connector panel

This section describes the controller connector panel of the robotic arm.

The controller connector panel is located at the rear of the controller. It features the following elements:

- On / Off power switch
- blue power LED indicator
- red / amber / green status LED indicator
- HDMI Out (camera video*)
- Micro USB (for firmware updates)
- USB 2.0, type A - qty 2 - for wired controller. Top port 1 A for charging. Bottom port 500 mA max, for peripherals.
- RJ-45 Gigabit Ethernet (LAN)
- Binder-USA 09 0463 90 19 (joystick, discrete I/O, E-Stop, expansion)
- Lumberg 0317 08 (power)

Note: Cables connected to the base controller must be less than 3 m in length. If not, you must perform a risk analysis. Cables longer than 3 m can potentially have an effect on radio frequency emissions and the immunity of the product.

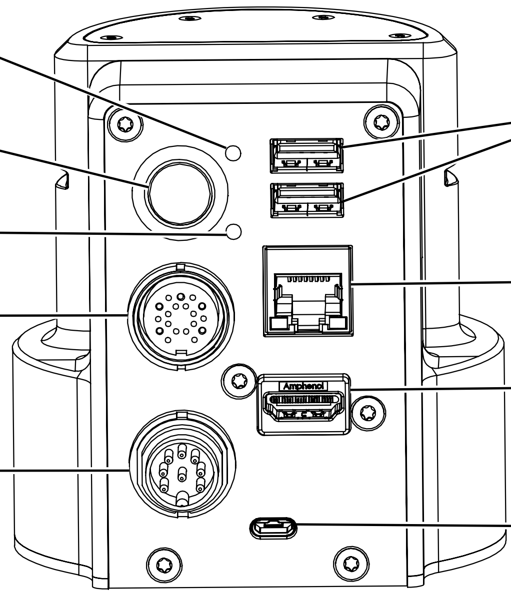


Figure 16: Controller connector panel

* to be implemented in future software release

Actuators

This section provides an overview of the robot actuators.

The rotational motion at each of the joints of the robot is powered by rotary actuators. There is one actuator for each joint. Each actuator allows for potentially unlimited rotation in either direction (There are software limits however on some joints however to avoid collisions between robot shell segments).

There are two sizes of actuator:

- small
- large

Each actuator has:

- torque sensing
- current and temperature sensing on each motor phase

Wrist joints use small actuators, while large actuators are used for other joints. All actuators are equipped with a 100:1 strain wave gear for smooth motion.

The actuators are connected to each other and to the interconnect board using a series of 41-pin flex cables. These cables convey:

- power
- 2 x full-duplex 100 Mbps Ethernet
  - one for 1 kHz control
  - one for vision / expansion data traffic

Actuator Specifications:

- actuator speed (maximum, unloaded):
  - 25 RPM (small)
  - 17 RPM (large)
- actuator torque (small):

  - 13 N·m (nominal)
  - 34 N·m (peak)
- actuator torque (large):
  - 32 N·m (nominal)
  - 74 N·m (peak)

Interface module

This section describes the interface module.

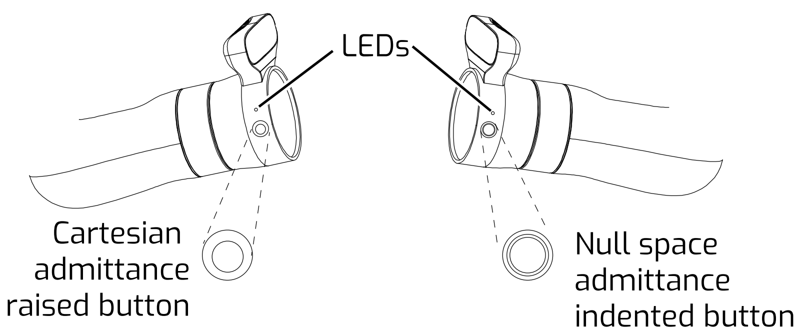
The interface module provides an interface for connecting a gripper or other tools at the end of the arm. The interface module also provides a mounting point and connection for the Vision module.

The interface module has a connection interface at the end of the arm, and is surrounded on the sides by a bracelet shell. The Vision module is mounted on the top of the bracelet.

The bracelet includes two buttons used to activate admittance modes to interact with the robot. By default the button on the right hand side (viewed from behind) puts the arm into Cartesian admittance while the button on the left puts the arm into null space admittance. The two buttons can be distinguished easily by touch without looking; the Cartesian admittance mode button sticks out from the surface in the center, while the null admittance mode button is slightly indented in in the center and ring-shaped.

Note: Only one of the buttons can be active at any given time. If you press the two buttons together or in close succession, the button pressed later will take effect.

The bracelet also includes two amber LEDs.



The interface module takes a 41-pin input from the last actuator of the robot.

The interface exposes connectors that allow different end effectors to be integrated with the robot. It features:

- Kinova internal end-effector interface
- 10-pin spring-loaded connector with RS-485 (compatible with Robotiq Adaptive Grippers)
- 20-pin user expansion interface

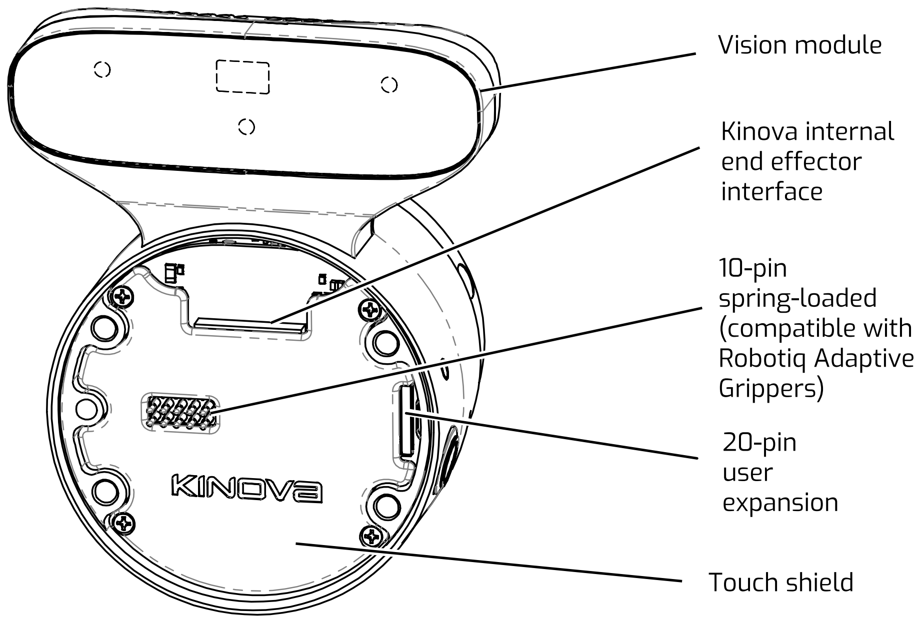


Figure 17: Interface module

The interface also includes four mounting holes for physical mounting of an end effector and a position key hole used for alignment of the end effector in the right orientation.

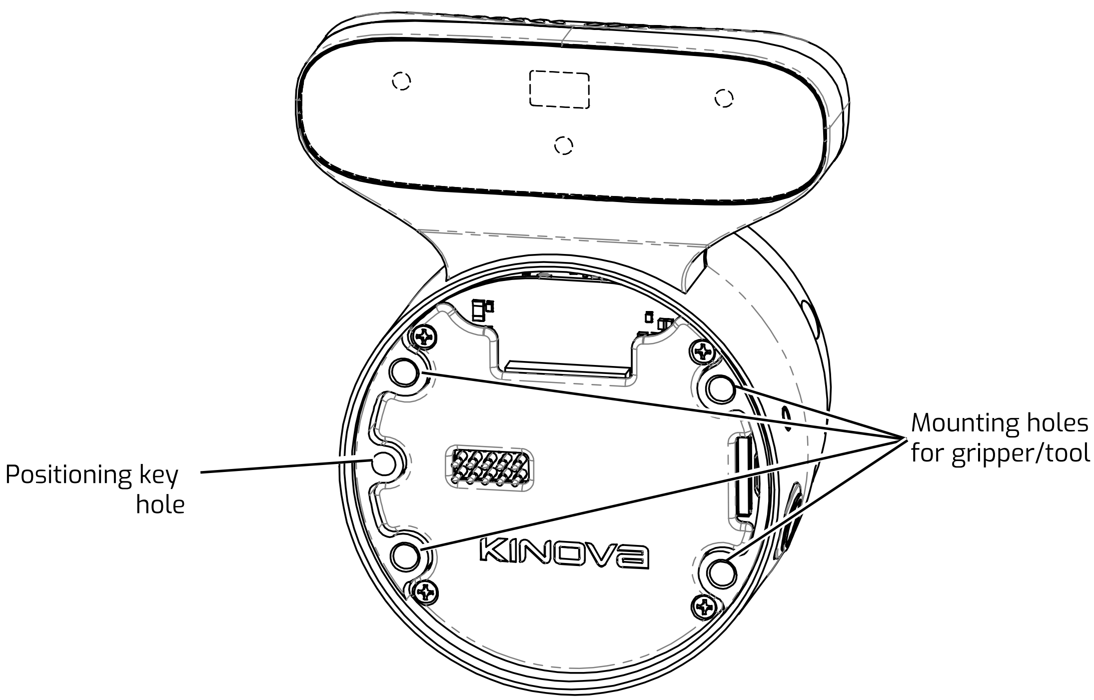


Figure 18: Mounting holes and positioning key hole

The interface module includes a 6-axis accelerometer / gyroscope. The module also includes an Ethernet switch to route connectivity and control data between the interface module and the vision module and any connected tool (e.g. gripper).

Note: The printed circuit board (PCB) of the interface module is partially covered with a touch shield with holes to expose only the output connectors - 10-pin spring loaded connector, 20-pin user expansion connector, and Kinova internal end effector interface.

Note: When there is no end effector present, it is recommended to place an end cap over the face of the interface module. Kinova provides an end cap with the robot. This end cap is attached to the interface with screws using the mounting holes on the interface. The end cap needs to be removed to attach an end effector to the robot.

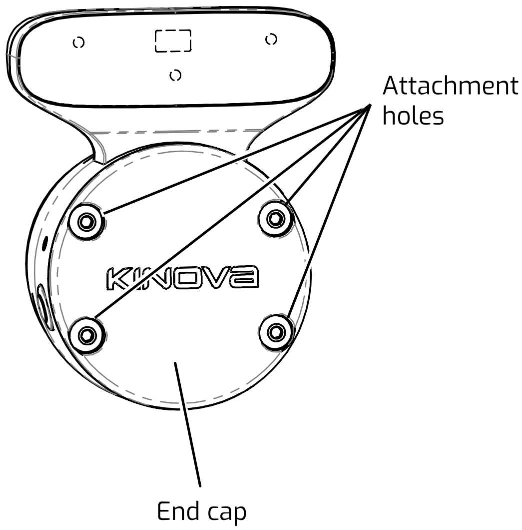


Figure 19: End cap

Vision module

This section describes the Vision module.

The Vision module is a module provided by Kinova to enable robotic computer vision applications.

The Vision module is mounted on the top side of the Interconnect module. A housing containing sensors protrudes from the top of the Interconnect module. The sensors are contained on the front face of the housing, facing out parallel to the axis of the last actuator.

The Vision module is used to capture and stream image data captured looking in the direction the end of the arm / end effector is pointed. The Vision module includes both a 2D RGB camera (**Omnivision OV5640**) and a 3D stereo depth sensor (**Intel® RealSense™ Depth Module D410**).

Together, the two cameras allow the capture of RGBD (color and depth) data. Both camera sensors can be configured using the KINOVA® KORTEX™ `VisionConfig` interface.

Performance for the Vision module depth sensor may be degraded at temperatures below 0° C. For more details, please consult the depth sensor data sheet.

The color and depth sensors data streams are made accessible to developers through a computer with a connection to the robot. For more information on accessing these data streams programatically, see here.

Vision module specifications

Color sensor:

- resolution and framerates:
  - 1920 x 1080 @ 30, 15 fps (16:9)
  - 1280 x 720 @ 30, 15 fps (16:9)
  - 640 x 480 @ 30, 15 fps (4:3)
  - 320 x 240 @ 30, 15 fps (4:3)
- FOV - 77° (diagonal)
- focal length (range) - 30 cm to ∞

Depth sensor:

- resolution and framerates:
  - 480 x 270 @ 30, 15, 6 fps (16:9)
  - 424 x 240 @ 30, 15, 6 fps (16:9)
- FOV - 77° (diagonal)
- minimum depth distance (min-Z) - 18 cm

Dimensions, specifications, and capabilities

Schematic and dimensions - 7 DoF spherical wrist

This section provides a schematic diagram of the system and its physical dimensions.

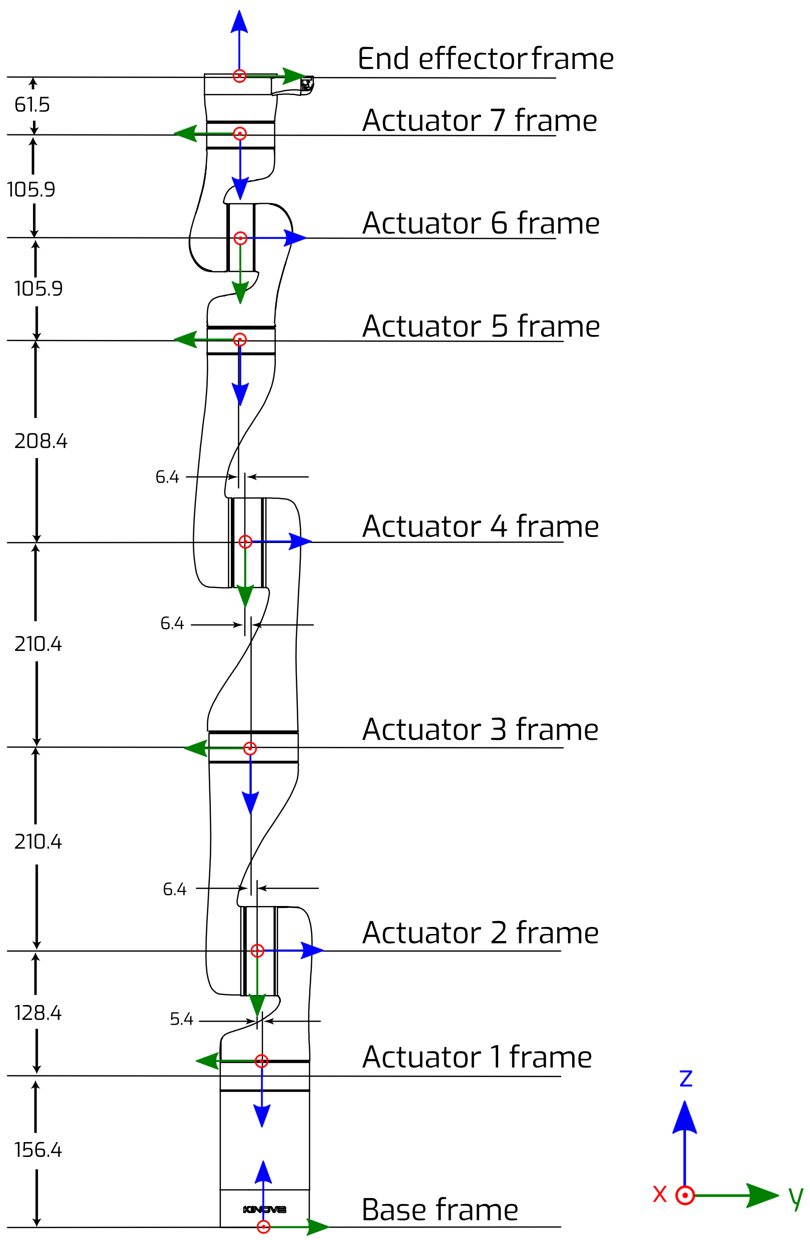


Figure 20: 7 DoF spherical wrist frames definition and dimensions

The image above defines reference frames for the base, joints (when all joint angles = 0) and end effector. Each frame is defined in terms of the previous frame via a transformation matrix. The diagram also indicates the link lengths and lateral offset values (measurements in mm).

The maximum reach of the robot, as defined by the distance from the shoulder (Actuator 2 frame) to the end effector frame, is 90.2 cm.

Table 9: 7 DoF spherical wrist robot geometric parameters

Description	Length (mm)
Base to actuator 1	156.4
Base to shoulder	284.8
First half upper arm length	210.4
Second half upper arm length	210.4
Forearm length (elbow to wrist)	208.4
First wrist length	105.9
Second wrist length	105.9
Last actuator to end efector	61.5
Joint 1-2 offset	5.4
Joint 2-3 offset	6.4
Joint 3-4 offset	6.4
Joint 4-5 offset	6.4

Table 10: Joint limits - 7 DoF spherical wrist

Actuator	Angular range	
	Lower limit	Upper limit
1	- ∞	+ ∞
2	- 126°	+ 126°
3	- ∞	+ ∞
4	- 147°	+ 147°
5	- ∞	+ ∞
6	- 117°	+ 117°
7	- ∞	+ ∞

Technical Specifications

This section provides the technical specifications for the KINOVA® Gen3 *Ultra lightweight robot*, categorized for ease of reference. Some of these also appear within the main body of the text.

Table 11: Safety / Security

Feature	Detail
Safety alarm (power monitor)	≥ 10A (maximum current)
Position monitoring	default and user-defined protection zones.
Thermal monitor	warning / shutdown above maximum operating temperature

Table 12: Environmental

Parameter	Value(s)
Temperature	-30 °C to 35°C (operating)
	-30 °C to 50 °C (storage)
Robot ingress protection	IPX3 (liquid)**
	IP3X (solid)
Relative humidity (non-condensing)	15% to 90% (operating)
Pressure	70 kPa to 106 kPa **
Sound pressure level	< 55.5 dBA
Universal Power Supply (external)	300 W
Input Voltage	100- 240 VAC
Input Frequency	50 - 60 Hz
Power supply ingress protection	IP42

Table 13: Controller (base)

Feature	Detail
power indicator	blue LED
status indicator	red/amber/green LED
USB 2.0 (two ports)	Xbox gamepad connect; 1 A charging (top), 500 mA USB peripherals (lower)
Gigabit Ethernet (RJ-45)	for development PC connection
Wi-Fi (IEEE 802.11a/b/g/n)	KINOVA® KORTEX™ *Web App* and API
HDMI 1.4a	(for future use)
circular connector [Binder-USA 09 0463 90 19]	joystick, discrete I/O, emergency line assert, expansion
circular connector [Lumberg 0317 08]	power
sensors	voltage, current, temperature, accelerometer and gyroscope

Table 14: Robot

Parameter	Value(s)
Weight	8.3 kg (with vision module, no gripper)
Payload	4 kg (mid-range continuous; no gripper) 4.5 kg (full-reach peak / temporary; no gripper) 1.1 kg (full-reach continuous; no gripper)
Maximum reach (fully extended)	902 mm (7 DoF)
Degrees of freedom	7 DoF
Actuators	qty 3 (small)
	qty 4 (large)
Wrist interaction buttons	qty 2 (user-configurable*; default for null space and Cartesian admittance control)
Power supply voltage	24 VDC (nominal, 18 to 30 V)
Materials	Carbon fiber shell
	Aluminum
Communciations and control	100 Mbps Ethernet for real-time 1 kHz control
	100 Mbps Ethernet for Vision module / expansion
Average power	45 W

Table 15: Actuators

Feature	Value(s)
Sensors	current sensors (motor), temperatures (motor), voltage, torque, position

Table 16: Interface module

Feature	Function
Vision module	color and depth sensing
Wrist status LEDs	admittance mode indication
Wrist pushbuttons	null-space / Cartesian admittance mode; programmable*
Kinova internal end-effector interface connector	Kinova internal use*
10-pin spring-loaded connector	RS-485 (compatible with Robotiq Adaptive Grippers)
20-pin user expansion connector [AVX/Kyocera 046288020000846+]	100 Mbps Ethernet
	UART (3.3V)*
	I^2C (3.3V)*
	GPIO (3.3V, qty 4)*
	24V @ 0.5A
	3.3V @ 0.1A for signaling
Sensors	accelerometer and gyroscope, voltage, temperature

Table 17: Vision

Feature	Detail
Depth sensor	480 x 270 @ 30, 15, 6 fps (16:9) 424 x 240 @ 30, 15, 6 fps (16:9) FOV: 77° (diagonal) focal length (range) - 18 cm to ∞
Color sensor	1920 x 1080 @ 30, 15 fps (16:9) 1280 x 720 @ 30, 15 fps (16:9) 640 x 480 @ 30, 15 fps (4:3) 320 x 240 @ 30, 15 fps (4:3) FOV: 77° (diagonal) focal length (range) - 30 cm to ∞

Table 18: Software / control

Feature	Detail
Low-level control	torque*, position, velocity, current
High-level control	Cartesian position/velocity, joint position/velocity
High-level control features (mode dependent)	protection zones - rectangular, spheric, cylindrical
	singularity handling
Servoing modes	high-level, low-level
Data recorder*	logging of position, speed, temperature, torque, battery, etc. (user-configurable)
Maximum velocity	user-configurable
Angular position of individual joints	
Angular speed of individual joints	
Supported ROS distribution	Kinetic Kame
Boot time	45 s
Internal communication frequency	1 kHz

* to be implemented in future software release

** subject to future change

Sensors

This section describes the robot sensors.

The robot contains a number of sensors to provide feedback on the status of the robot. This data is used by the robot for internal monitoring and control.

The robot components contain the following sensors:

Base sensors

- voltage
- current
- temperature

- arm present detection
- 6-axis accelerometer / gyroscope

Actuator sensors

- motor phases current sensors (one per phase)
- motor phases temperature sensors (one per phase)
- CPU temperature sensor
- input voltage sensor
- Hall effect sensors for BLDC motor drive
- absolute rotary position encoder
- incremental rotary position encoder
- torque sensors

Interface module sensors

- voltage monitoring (future enhancement)
- temperature sensor (CPU, accelerometer and dedicated)
- 6-axis accelerometer / gyroscope

Access to sensors data

Data from some sensors can be read by users using the APIs or through the Monitoring page of the Web Application.

The API method `RefreshFeedback()` in the `BaseCyclic` API service returns a data structure with readings from sensors in the base, actuators, and interface. For detailed information on how to unpack this data in an application, see the `BaseCyclic` API documentation.

The following tables give more information about the sensor data.

Base readings available

Table 19: Base readings available through API

Field name	Description
`arm_voltage`	arm voltage in V
`arm_current`	arm current in A
`temperature_cpu`	CPU temperature in °C
`temperature_ambient`	ambient temperature in °C
`imu_acceleration_x`	IMU measured acceleration (X-Axis) of base in m / s^2
`imu_acceleration_y`	IMU measured acceleration (Y-Axis) of base in m / s^2
`imu_acceleration_z`	IMU measured acceleration (Z-Axis) of base in m / s^2
`imu_angular_velocity_x`	IMU measured angular velocity (X-Axis) $\omega_x$of base in ° / s
`imu_angular_velocity_y`	IMU measured angular velocity (Y-Axis) ω_y of base in ° / s
`imu_angular_velocity_z`	IMU measured angular velocity (Z-Axis) ω_z of base in ° / s

Actuators readings available

Table 20: Actuators readings available via API

Field name	Description
`position`	angular position of the actuator in °
`velocity`	angular velocity of the actuator in ° / sec
`torque`	torque in N·m
`current_motor`	motor current in A
`voltage`	main board voltage in V
`temperature_motor`	actuator motor temperature in °C (highest of three 3 phases)
`temperature_core`	microcontroller temperature in °C

Interface readings available

Table 21: Interface readings available via API

Field name	Description
`imu_acceleration_X`	IMU measured acceleration (X-Axis) of interconnect in m / s^2
`imu_acceleration_Y`	IMU measured acceleration (Y-Axis) of interconnect in m / s^2
`imu_acceleration_Z`	IMU measured acceleration (Z-Axis) of interconnect in m / s^2
`imu_angular_velocity_X`	IMU measured angular velocity (X-Axis) ω_x of interconnect in ° / s
`imu_angular_velocity_Y`	IMU measured angular velocity (Y-Axis) ω_y of interconnect in ° / s
`imu_angular_velocity_Z`	IMU measured angular velocity (Z-Axis) ω_z of interconnect in ° / s
`voltage`	main board voltage in V
`temperature_core`	microcontroller temperature in °C

End effector readings available

Table 22: End effector readings available via API

Field name	Description
`tool_pose_x`	Measured Cartesian position (X-axis) of the end effector in m
`tool_pose_y`	Measured Cartesian position (Y-axis) of the end effector in m
`tool_pose_z`	Measured Cartesian position (Z-axis) of the end effector in m

Field name	Description
`tool_pose_theta_x`	Measured Cartesian orientation (X-axis) of the end effector in °
`tool_pose_theta_y`	Measured Cartesian orientation (Y-axis) of the end effector in °
`tool_pose_theta_z`	Measured Cartesian orientation (Z-axis) of the end effector in °
`tool_twist_linear_x`	Measured cartesian linear velocity (X-Axis) of the end effector in m / s
`tool_twist_linear_y`	Measured cartesian linear velocity (Y-Axis) of the end effector in m / s
`tool_twist_linear_z`	Measured cartesian linear velocity (Z-Axis) of the end effector in m / s
`tool_twist_angular_x`	Measured cartesian angular velocity (X-Axis) of the end effector in ° / s
`tool_twist_angular_y`	Measured cartesian angular velocity (Y-Axis) of the end effector in ° / s
`tool_twist_angular_z`	Measured cartesian angular velocity (Z-Axis) of the end effector in ° / s

Effective workspace

This section provides information on the effective workspace of the robot.

Effective workspace overview

The effective workspace refers to the region in three-dimensional space which is reachable by the robot end effector. This is impacted by several factors, including the number and length of the links, the joint ranges, and the shape of the links

There are two definitions of effective workspace, the first being larger than the second.

1. **Nominal (or reachable) workspace -** the set of all locations in the three-dimensional space reachable by the end effector through at least one combination of end effector position and orientation
2. **Dextrous workspace -** the subset of the nominal workspace in which the end effector still has the full freedom to move, both in translation (three degrees of freedom) and in rotation (three degrees of freedom)

Detailed information

The following graphic illustrates a two-dimensional cross-section of the nominal workspace for the robot.

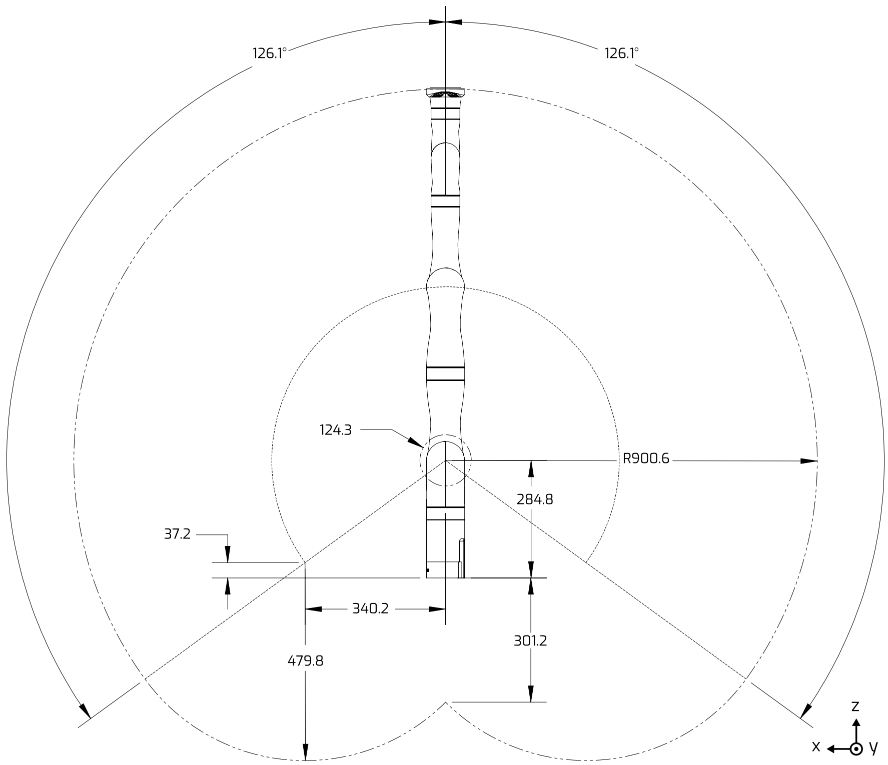


Figure 21: 7 DoF robot nominal workspace (measurements in mm)

Payload vs. workspace

This section describes the variation of payload over the workspace and depending on the type of use.

Overview

The payload of the robot is the maximum mass that the robot can hold up at the end effector.

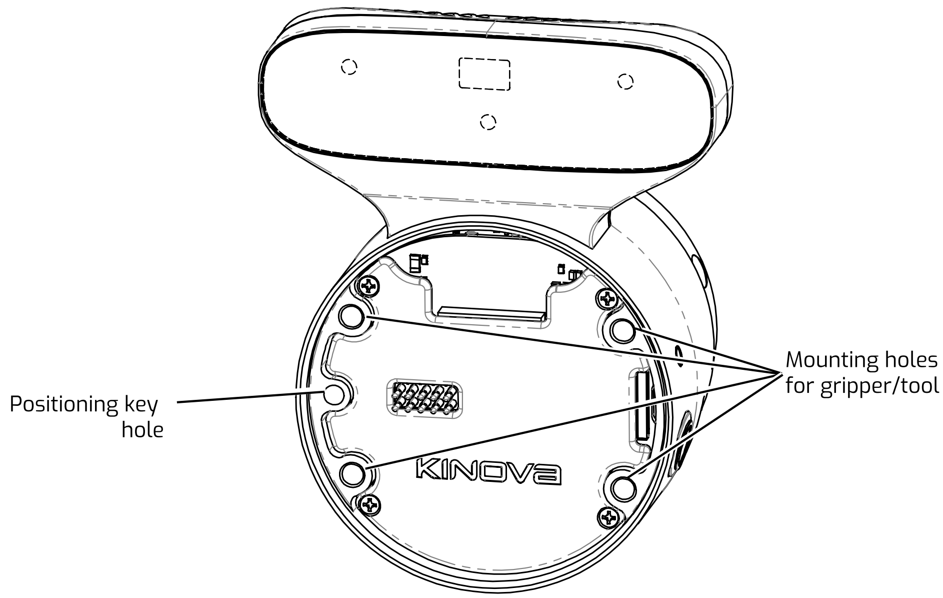
This is generally not one constant figure, but will depend on a few factors.

- radial distance from the base - the payload will be highest closest to the base, and will go down as the end effector gets farther out from the base axis.
- temporary vs. continuous - the robot will have a maximum payload that can be handled temporarily for a short period of time. However, continued use of the arm with that payload for an indefinite period will cause the arm to heat up, as the heat generated by the strain on the actuator exceeds the rate at which heat can be dissipated. However, a smaller mass can be handled for an indefninite period. This is referred to as the continuous payload limit.

The payload will also depend on whether a gripper is attached or not, with some of the payload capability reduced to lift the weight of the gripper.

Interface module expansion - tips for installing tools

This section describes what is needed to install a new tool onto the interface module.

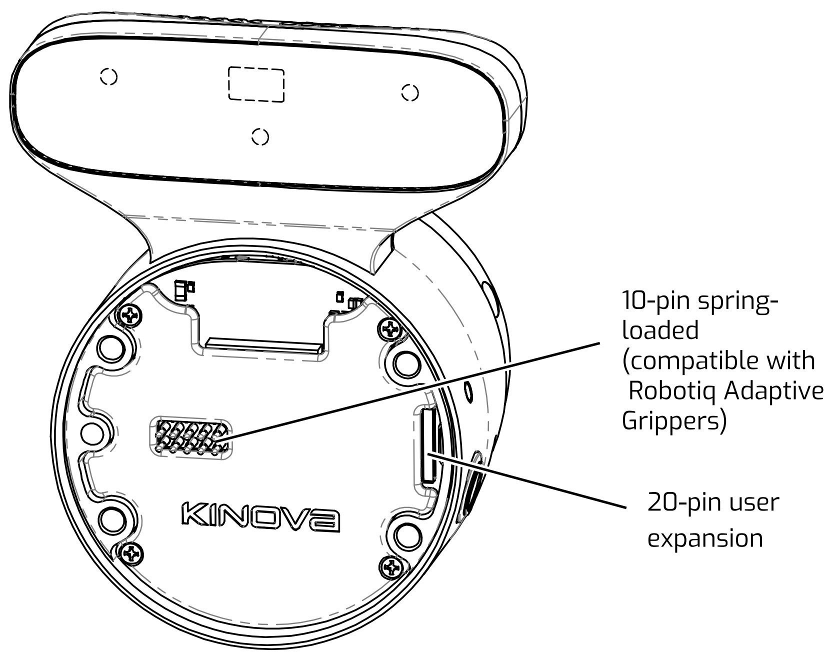


At some point, you may want to install a new tool such as a gripper or sensor onto the robot.

Generally, this involves two steps.

1. Physically mounting the tool using the screw holes available on the Interface module face.

   Note: The holes on the Interface module face are laid out to allow easy installation of Robotiq Adaptive Grippers using the four supplied M5 X 40 mm Socket Head Cap Screws (SHCS include O-rings for compliance with the IP rating for sealing). For other third-party tools, it may be necessary to create a mounting structure matching the provided interface module bolting pattern, as discussed in the End effector reference design section.



2. Integration of Robotiq Adaptive Grippers to the robot power and control signals uses the 10-pin spring-loaded connector. Other third-party tools can use the signals on this and / or the 20-pin user expansion connector. Currently power and Ethernet expansion are available via the user expansion connector. The pinout details are described in later sections.

Note: If designing or installing your own tool or end effector, remember to take into consideration the field of view of the depth sensor when designing the length of the tool to avoid hindering the effectiveness of the vision module depth sensor.

End effector reference design

This section provides guidance to developers on developing and integrating a new end effector with the robot.

Introduction

The KINOVA® *Gen3 Ultra lightweight robot* is designed for maximum flexibility. As such, the robot has a user expansion interface designed to simplify the development required to incorporate different sensors, end effectors or other tools/boards. The supported user interfaces are listed in the section Interface module user expansion connector pinout.

Kinova provides a reference design package which includes mechanical and electrical modular interfaces.

Mechanical interface

The mechanical interface is a flanged circular structure which converts the interface module bolting pattern to the ISO 9409-1-A50 mounting plate pattern common to many industrial robots.

Kinova recommends that the mechanical interface part of your end effector be machined from solid aluminium, though for applications where no payload will be attached (only sensors or PCBs), a 3D-printed interface part may suffice.

Note: If a 3D printed interface is used, perform an evaluation of the forces involved to ensure that adequate safety factors are observed.

Note: The structure includes openings at the top and bottom of the structure for the passage of cabling for the electrical interface. As a result, the interface does not provide ingress or EMI protection.

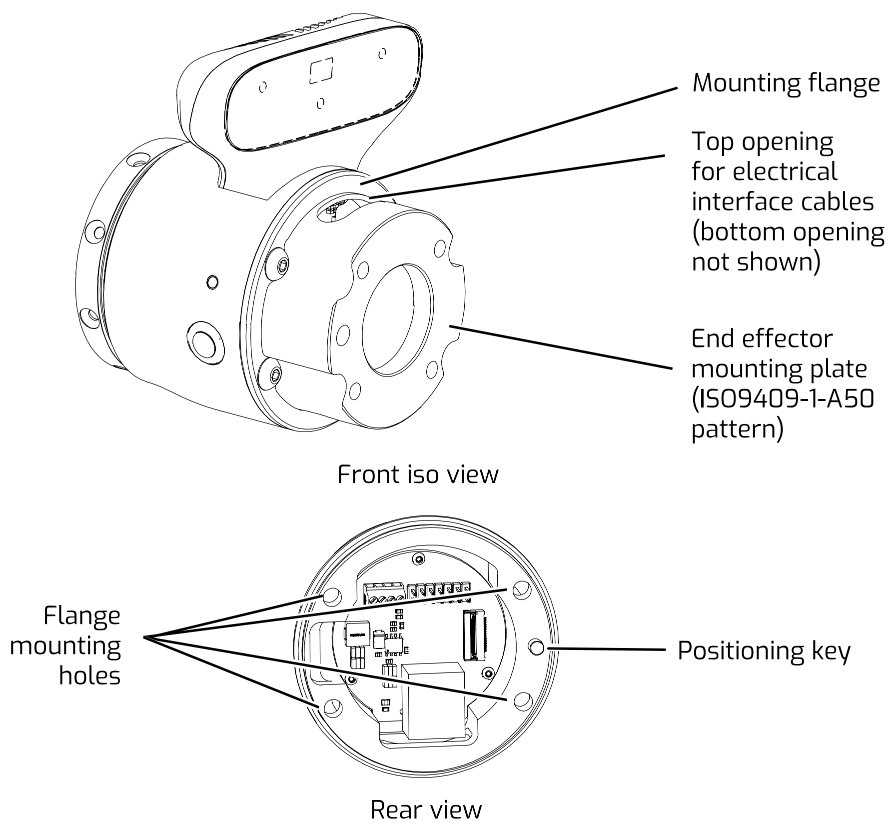


Figure 22: Reference design mechanical interface (details in reference design package)

Electrical interface

The electrical reference design acts as a breakout, giving access to:

- 24 V / 0.5 A
- 5 V / 2.5 A (through a DC-DC buck converter)
- Expansion Ethernet (100 Mbit, through a RJ-45 port)
- GPIO, I^2C and UART*

* to be implemented in future software release

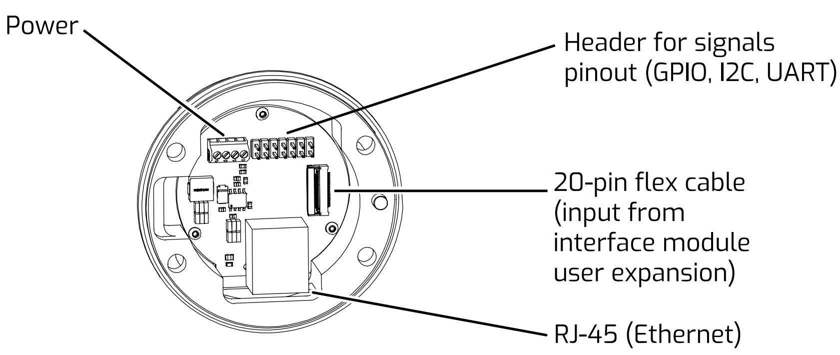


Figure 23: Reference design electrical interface (details in reference design package)

Reference design package

A reference design package is available on the Kinova website. www.kinovarobotics.com/knowledge-hub/technical-resources

The package includes several useful files to help you with building an end effector. You may use these files as is or as a starting point to design your own end effector. The package contents include:

- STL file of the mechanical interface, for 3D-printing
- STEP file and PDF drawing of the mechanical interface, for machining
- STEP file of the Kinova breakout PCB for integration into CAD programs
- KR13933.ASY file which directs the assembly of the PCB (including the BOM)
- KR13933.PCB which directs the PCB fabrication (including Gerber files)
- KR13933.SCH which includes the circuit board schematic diagrams

Remove end cap from Interface module

This section describes how to remove the end cap from the Interface module.

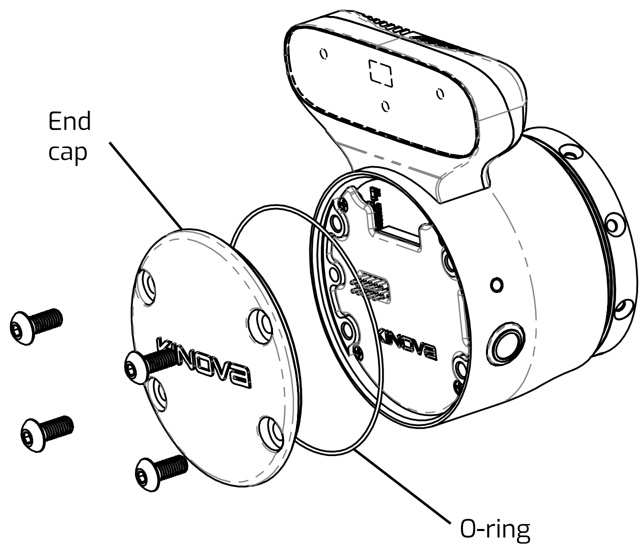
Before you begin

You will need a 3 mm hex key.

About this task

The robot ships originally with an end cap over the interface. Attaching an end effector to the robot requires removing the end cap first. Removing the end cap exposes expansion and end-effector connection points.

When removing the end cap, there is an O-ring exposed which **must be conserved**. The O-ring is used to provide protection against water ingress and EMI at the junction between the robot interface and the end effector.



Procedure

1. The end cap is held onto the robot interface using four M5 button head cap screws. Using a 3 mm hex key, remove the screws and preserve the screws.
2. Remove the end cap, and set aside with the screws.
3. You will see an O-ring on removing the end cap. The O-ring will be needed when attaching an end effector.

Note: Set aside the O-ring with the screws and end cap for safe keeping.

Robotiq Adaptive Grippers installation (optional)

This section describes the procedure for installing a Robotiq Adaptive Grippers on the robot.

Before you begin

You will need four M5 X 40 mm Socket Head Cap Screws and 4 mm hex key (supplied).

You will also need to have removed the interface end cap (robot comes with end cap connected). You will need the O-ring that was exposed when the end cap was removed.

About this task

This procedure describes the installation for Robotiq Adaptive Grippers (Robotiq 2F-85, 2F-140, or Hand-E Gripper) on the interface module of the robot. The interface module allows easy mounting of Robotiq Adaptive Grippers. This procedure mechanically mounts the gripper on the robot and integrates the gripper with the robot in terms of electrical power and control. The interface module has four mounting holes corresponding to the bolt pattern on the gripper. The 10 spring-loaded pins on the interface mate with a contact plate on the inside of the Robotiq Gripper to establish electrical supply and controls.

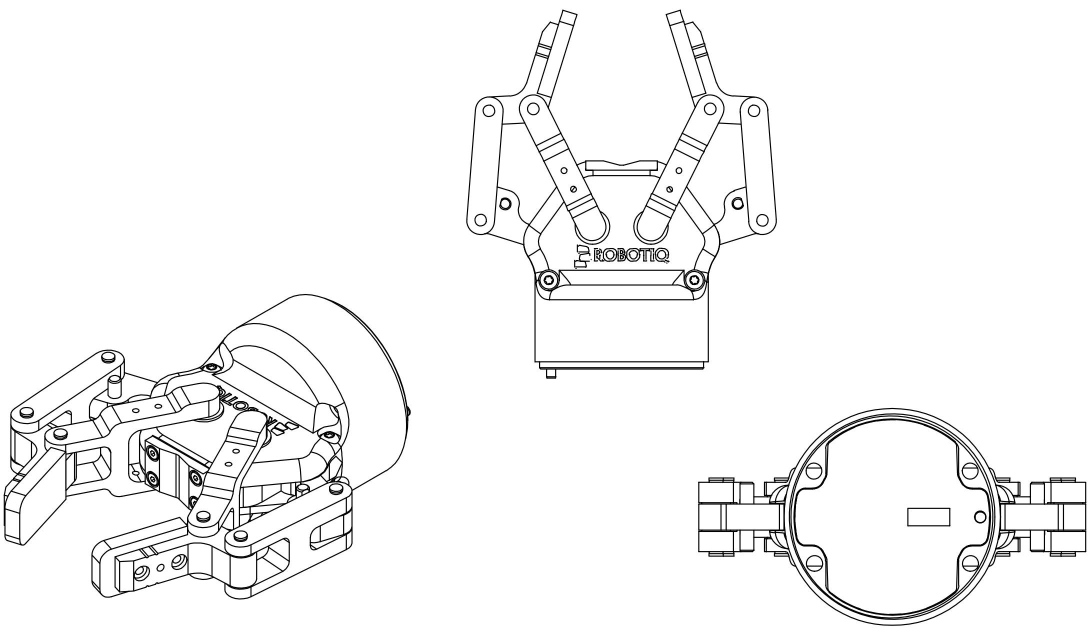
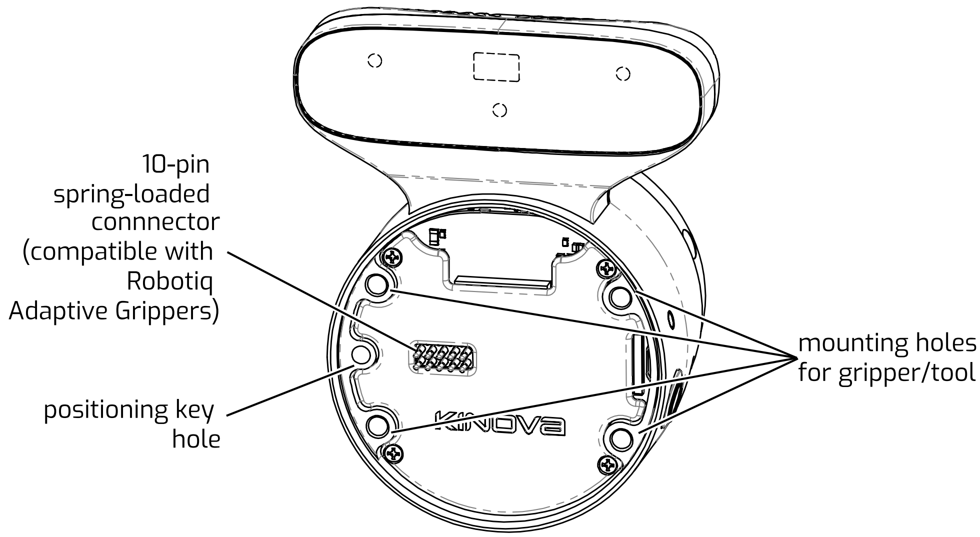


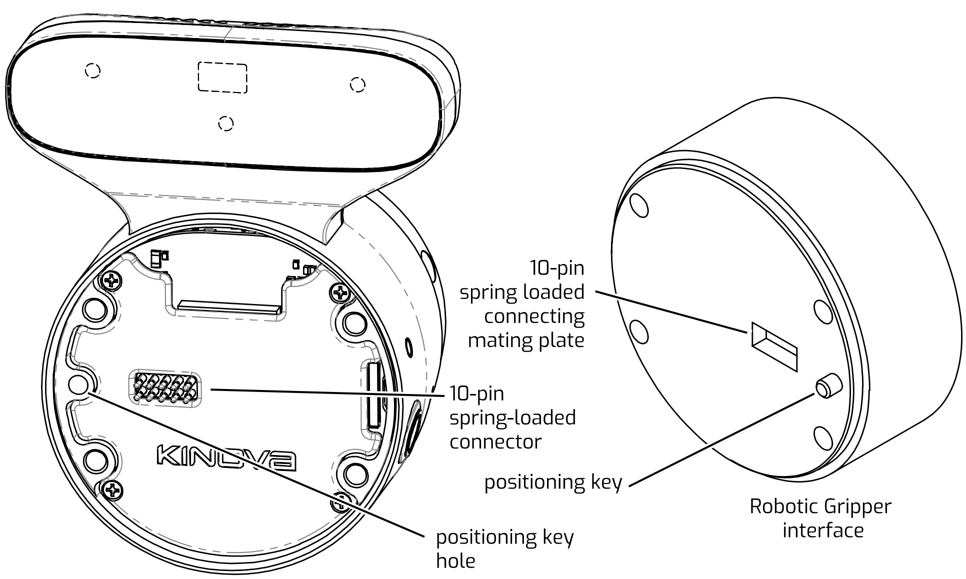
Figure 24: Robotiq Adaptive Gripper (Robotiq 2F-85 Gripper shown)

Procedure

1. Prepare the four supplied M5 X 40mm Socket Head Cap Screws and a 4 mm hex key.



2. Place the O-ring around the diameter of the gripper. The O-ring protects the junction between the interface module and gripper from water ingress and EMI.
3. Locate the positioning key on the Robotiq Gripper and the corresponding hole on the interface module face.



4. Position the gripper interface against the Interface module interface so that the positioning key of the gripper is in the positioning key hole of the interface module and the 10-pin spring-loaded connector of the interface module is aligned with the corresponding mating interface on the gripper.
5. Insert the four screws through the front face of the gripper. Tighten each screw in sequence until they are all snug (do not overtighten).

Results

The Robotic Gripper will now be mechanically installed on the robot. The gripper is also fully integrated with the robot for power and controls. The robot provides power to the gripper, and the gripper can be controlled using either the provided gamepad or the KINOVA® KORTEX™ *Web App* virtual joysticks.

What to do next

⚠ For your personal safety, it is strongly recommended that you read the user documentation for the Robotiq Gripper before use.

Interface module bolting pattern

This section describes the bolting pattern for a tool interfacing with the Interface module.

Drill pattern for mounting screws and position key

The drill pattern below is for the four mounting screws. Openings to accommodate the required cables connections with the connectors also need to be added. The use of a 4 mm dowel pin to accurately localize the positioning key hole is optional but strongly recommended.

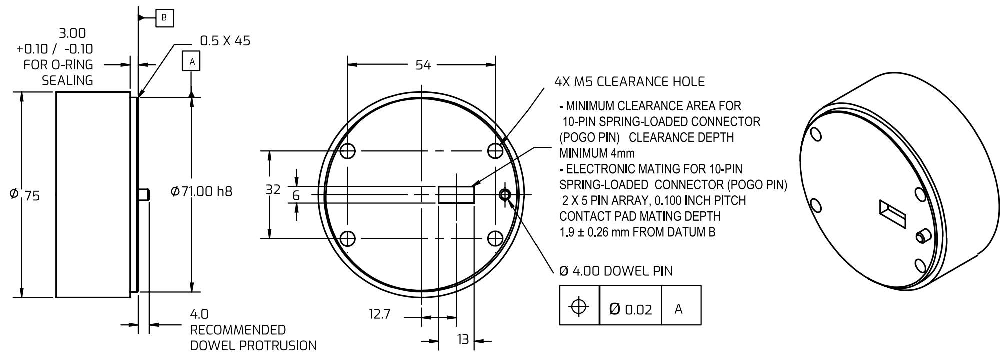


Figure 25: Mounting holes for gripper/tool (all dimensions in mm)

Interface module user expansion connector pinout

This section describes the functionality available at the interface module user expansion connector.

The interface module user expansion connector pin assignment is described in the table below.

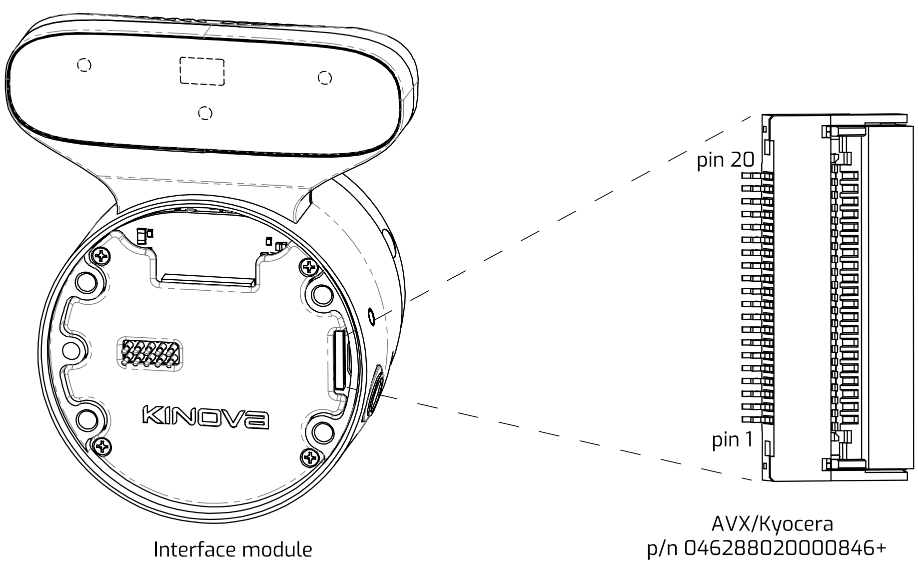


Figure 26: Interface module user expansion connector

Table 23: Interface module user expansion pinout

Pin	Name	Comment
1	+24V USER	24V / 0.5A power; a protection device limits current shared between gripper and user expansion port to 1A total.
2	+24V USER	
3	GND	power return path
4	GND	
5	ETH_RX_P	Ethernet Rx 100Mbps (connected with EXP bus)
6	ETH_RX_N	
7	GND	signal return path
8	ETH_TX_P	Ethernet Tx 100Mbps (connected with EXP bus)
9	ETH_TX_N	
10	GND	signal return path
11	+3V3	3.3V / 100 mA; can be used for small IC or sensor*
12	UART_TXD	signal 3.3V*
13	UART_RXD	signal 3.3V*
14	GND	signal return path*
15	I2C_SCL	I^2C clock - 3.3V*
16	I2C_SDA	I^2C data - 3.3V*
17	GPIO1	General Purpose Input / Output 3.3V*
18	GPIO2	
19	GPIO3	
20	GPIO4	

* to be implemented in future software release

Spring-loaded connector pinout

This section describes the pinout of the spring-loaded connector.

The spring-loaded connector pin assignment is described in the table below.

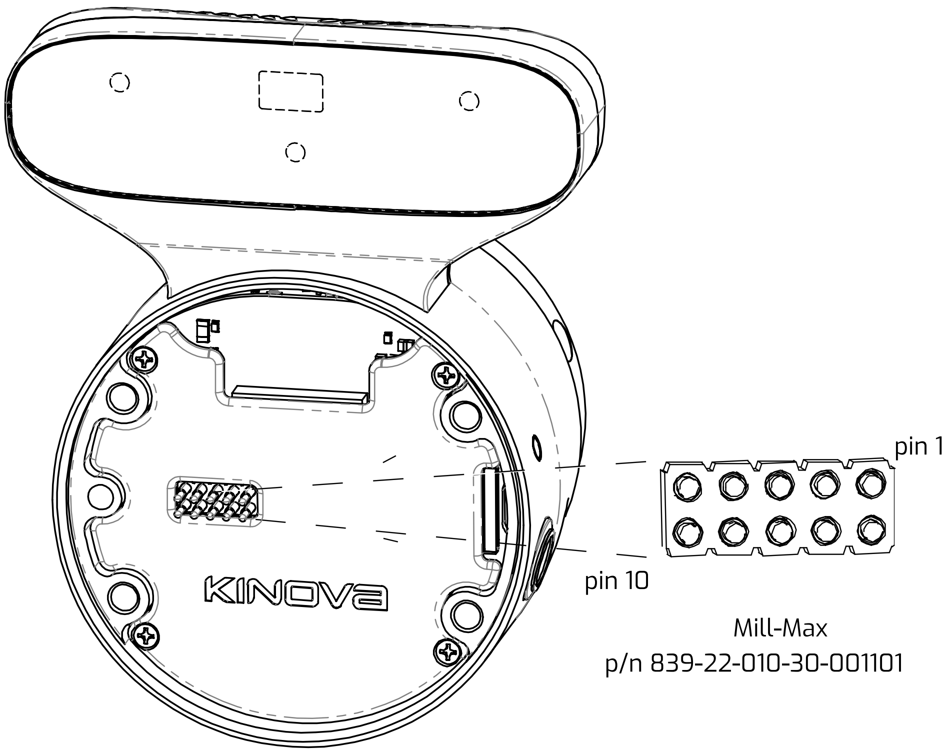


Figure 27: Spring-loaded connector

Table 24: Spring-loaded connector pinout

Pin	Name	Comment
1	GND	power return path
2	GND	
3	+24V	24V / 1A power for end device (current limit shared with interface module user expansion port)
4	+24V	
5	PRESENT	end device presence detection (connect to GND on end device)
6	N/C	no connection
7	RS485_N	RS-485 signal pair (bidirectional)
8	RS485_P	
9	GND	signal return path
10	N/C	no connection

Robot communications and network interfaces

This section describes communications and network interfaces within the robot.

The devices in the robot, from the base of the arm through the chain of actuators, to the interface module at the end of the arm, are daisy chained together using 41-pin flex cables which carry power and communications.

The base, actuators, and interface module each contain an Ethernet switch. The Ethernet port on the connector panel of the base controller allows an external computer to connect to the Ethernet switch of the base.

The Kinova vision module and any 3rd party tool that makes use of Ethernet communications user expansion pins in the interface connect directly to the interface module Ethernet switch. Other tools (for example any gripper interfacing using the 10-pin spring loaded connector on the interface) will interface instead with the interface module CPU (which is connected to the Ethernet switch).

Together, this enables dual Ethernet networks between all the devices (base, actuators, interface, Vision module, and end effector tools) with data carried between the base and interface over the 41-pin flex cables. This is accessible from a client computer via the 1 Gbps Ethernet port on the base controller connector panel.

The flex cables carry two distinct 100 Mbps Ethernet communications channels.

- one is for control and monitoring of actuators, interface module, and gripper (if present)
- the other is for data transmission for the vision module and expansion.

Each device connected to one of the Ethernet switches has an IP address to allow routing of communications, transmitted using UDP.

The actuators and interface module have the following default IP addresses:

Table 25: Actuator and gripper IP addresses

Device	IP address
Actuator 1	`10.10.0.10`
Actuator 2	`10.10.0.11`
Actuator 3	`10.10.0.12`
Actuator 4	`10.10.0.13`
Actuator 5	`10.10.0.14`
Actuator 6	`10.10.0.15`
Actuator 7	`10.10.0.16`
Interface module	`10.10.0.17`

The expansion devices (Vision module and expansion tool peripherals) have the following IP addresses:

Table 26: Expansion IP addresses

Expansion Devices	IP address
Vision module	`10.20.0.100`
Expansion device	`10.20.0.200/24`*

The robot Ethernet network features three VLANs:

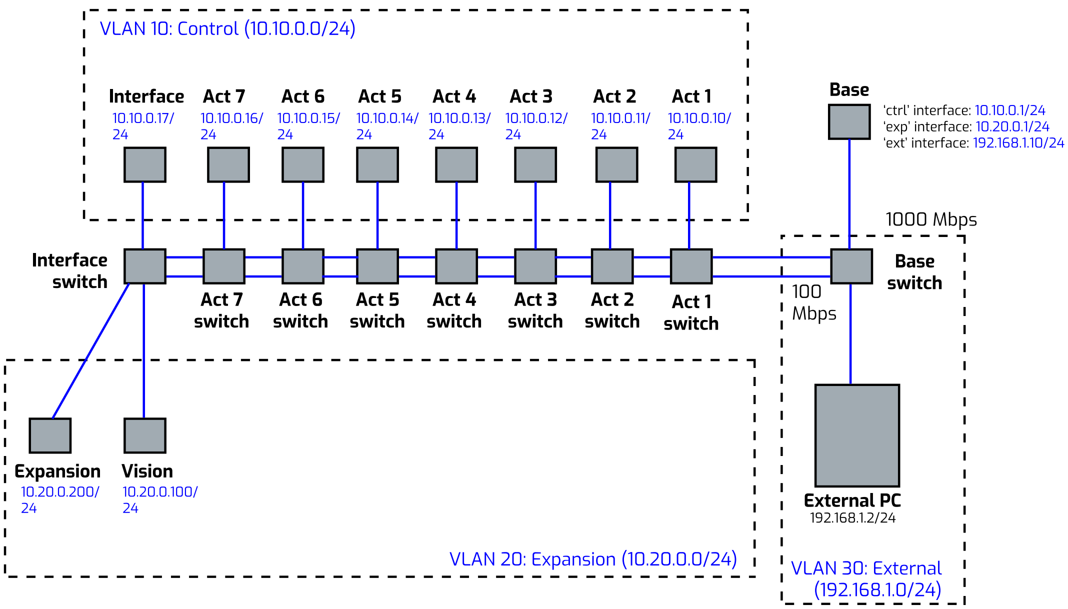
- VLAN 10 : control
- VLAN 20 : expansion
- VLAN 30 : external

The base has network interfaces to all three of these VLANs:

Table 27: Base network interface IP addresses

VLAN	IP address
CTRL interface IP address	`10.10.0.1/24`*
EXP interface IP address	`10.20.0.1/24`*
EXT interface IP address	`192.168.1.10/24`*

The graphic below illustrates the topology of the networks.



* CIDR notation

Accessing Vision module color and depth streams

This section describes access to the video module color and depth streams.

The video module sensors capture two video streams:

- color
- depth

The data from these streams is sent from the vision module back to the base controller via the vision / expansion channel carried over the internal flex cable links.

These two streams are accessible via the Real Time Streaming Protocol (RTSP) on a computer connected to the robot (with transport over real-time transport protocol (RTP)).

Using the default IP address settings for the base controller, the two streams are accessible via:

- color sensor stream: `rtsp://<base IPv4 address>/color`
- depth sensor stream: `rtsp://<base IPv4 address>/depth`

For the default configuration of the base controller network interface, this would give:

- `rtsp://192.168.1.10/color`
- `rtsp://192.168.1.10/depth`

Note: Examples in the documentation will use the default base controller IP setting for simplicity.

Streams specifications

- pixel format: Z16 pixel format - 16 bits LSB transferred as grayscale
- H.264 baseline profile (constant bitrate)
- RTSP server listening on port 554 (default)
- maximum of two simultaneous connections per stream
- inactivity timeout of 30 seconds

The KINOVA® KORTEX™ Developer Guide section of the user guide describes in more detail how to work with the vision module camera streams.

Concepts and terminology

Robot key concepts

This section describes some important concepts related to the robot.

Actions

An action is something that the user wants the robot to do. This can include (but is not limited to):

- a command to control the robot
- toggling a mode
- replaying a predefined movement
- changing a position or motion parameter
- setting a limitation
- adding a delay

The full set of action types is defined in the Base API.

Control modes

A control mode is one of several modalities of controlling the motion of the robot while it is in run mode. Different modes provide different means to describe or guide the desired motion. The control modes for the arm are:

- angular joystick
- angular trajectory
- vision joystick
- Cartesian joystick
- Cartesian trajectory
- Cartesian admittance
- joint admittance
- null space admittance

Controller

A controller is a device or interface through which a user can produce actions on the robot.

Factory settings

Factory settings are the configuration settings of the robot as they were when the robot arrived from the manufacturer. A robot can be returned to factory settings, which includes the base configuration and the network settings.

Mappings

A mapping is a one-to-one correspondance between an input on a controller and a resulting action on the robot.

Notifications

A notification is a log of an event that happens related to the robot while a user is using the robot. Events are of different types, including (but not limited to):

- user login
- controller input
- safety
- action or sequence
- connection / disconnection of arm, controller, or end effector
- configuration change or backup

- factory restore
- protection zone reached / entered / exited
- change in control, operation, or servoing mode

A notification will include the user profile, type of event, details of the event (if applicable), and a timestamp.

Operating mode

Operating modes are the different operational states of the robot. The operating modes for the arm are:

- update - in process of update
- update completed - update is completed successfully
- update failed - update process started but failed to complete successfully
- shutting down - arm is in process of shutting down
- run - normal operating mode. Arm is ready to accept control inputs.
- fault - robot is in an error state

Protection zone

A protection zone defines a three-dimensional region with respect to the robot base where the end effector or arm is either prevented from entering or where its speed is limited. Protection zones are used for enabling obstacle avoidance. For the robot, protection zones can be one of three shapes (or combination thereof):

- cylinder
- rectangular prism
- sphere

Sequence

A sequence is an ordered list of actions.

Servoing mode

A servoing mode is a modality through which commands are transmitted to robot devices during operation. The servoing modes are as follows:

- high-level servoing - user(s) send commands to the base, which routes the commands to the desired device. The base also manages a 1 kHz control loop.
    - single-level - a single user sends commands to a base
- low-level servoing - the user sends commands to the base for routing to the desired device. There are no high-level kinematics or control features available.

Upgrade package

An upgrade package contains firmware images for all devices on the robot, not just the ones which are being ugraded.

User profiles

A user profile is a collection of basic information about the person using the robot, along with credentials (username and password) for access. A user profile allows access to the robot to be controlled based on login credentials, and allows permissions for reading, updating, and deleting different configuration items to be controlled. The user profile also allows notifications for events happening during a user's session to be associated with the user. Notifications that were sent by the robot can be viewed in the `Web App > Notifications` page if the Web App is open and connected to the robot before the notifications were sent.

Terminology reference

The following sections give an overview of the terminology of the robotic arm.

For ease of reference, the terminology reference section has been divided into the following categories:

- Acronyms
- General mathematics and robotics
- Features, components and functionalities
- Control and Operation Modes

General mathematics and robotics

Axis
A fixed line for the measurement of coordinates or angles, in relation to which is specified the robot motion (in linear or rotational fashion).

Base Frame
The reference frame located at the center of the bottom surface of the arm's base. This serves as the origin frame in Cartesian space.

Cartesian Space
The Euclidean space described by x, y and z of the Cartesian coordinate system.

Center of Mass
The center of mass is a useful reference point for calculations in mechanics that involve masses distributed in space.

Closed Loop control system
Control system of a robot manipulator by means of sensor feedback. As a manipulator is in action, its sensors continually feed information back to the robot's controller which is used to further guide the manipulator within the given task. Many sensors are used to provide information about manipulator placement, speed, torque and applied forces, as well as the location of a targeted moving object, etc.

Coordinate System
A system used to represent a position in three-dimensional space, consisting of three coordinate axes and an origin.

Degrees of Freedom (DoF)
The number of independent directions or joints of the robot, which would allow the robot to move its end effector through the required sequence of motions. For arbitrary positioning, six degrees of freedom are needed: three for position (x, y, z) and three for orientation (yaw, pitch and roll).

Endpoint
The nominal commanded position that a manipulator will seek at the end of a motion path.

Euler Angle

- Describes the orientation of a rigid body with respect to a fixed coordinate system.
- Another less complicated way to express vector orientation.

Gravity Compensation
The gravity model computes the forces and torques due to gravity at each joint. The gravity torques are then subtracted from the measured torques, resulting in a gravity-free torque vector.

Joint Angle
Describes the position of every joint of a robot as as series of angles.

Joint Space
The set of all possible joint positions.

Null Space
The mathematical space of joint speeds where the robot can change its configuration (generate joint speed and motion) without changing the end-effector pose (Null Twist at the end effector).

Orientation

- The angle formed by the major axis of an object relative to a reference axis. It must be defined relative to a three-dimensional coordinate system, i.e. the angular position of an object with respect to the robot's reference system.
- The end effector's position consists of two things, a translation (x, y and z) and an the orientation that can be expressed in several ways. Think of it as a vector. This would be the orientation of this vector.

Path
The continuous locus of points (or positions in three dimensional space) traversed by the tool center point and described in a specified coordinate system.

Path (Angular)
The set of at least two angular poses, through which the actuator values angles should pass during motion.

Path (Cartesian)
The set of at least two Cartesian poses, through which the tool of the robot should pass during motion.

Pose
Describes the position and orientation of a rigid body in Cartesian space.

Position
The definition of an object's location in 3D space, usually defined by a 3D coordinate system using X, Y, and Z coordinates.

Quaternion
A quaternion is a group of four numbers used to express a vector's orientation.

Reference Frame
A system of geometric (coordinate) axes in relation to which measurements of size, position, or motion can be made. Robotic manipulators normally include several strategically placed reference frames, located at the base, joints and end effector.

Tool Frame
A coordinate system attached to the end effector of a robot (relative to the base frame).

Trajectory
A specific path in the Cartesian robot workspace (e.g. a straight line trajectory for the end effector) can be defined by the user.

Twist
Generalized velocity vector, which is a combination of translationnal velocity and rotational velocity.

Wrench
Generalized force (vector which is a combination of linear force and torques).

Vector
Mathematical representation of physical quantities that have both magnitude and direction, expressed in terms of a Reference Frame.

X DoF
Undefined number of degrees of freedom.

Features, components and functionalities

Actual Position
The position (or location) of the tool frame point. Note that this will not be exactly the same as the requested position due to a multitude of un-sensed errors (such as link deflection, transmission irregularity, tolerances in link lengths, etc.).

Admittance
Used in several control modes wherein the arm is moved directly by the user touching the arm and applying force and torque to the end effector, the wrist, or to individual joints.

Base
Refers to the stationary base structure of a robot arm that supports the first arm joint.

Base support
The stable platform to which the base is attached

Continuous Path
Describes the process whereby a robot is controlled over the entire path traversed, as opposed to a point-to-point method of traversal. Used when the end-effector smooth trajectory is of vital importance to provide the constant motion required by applications such as spray painting.

End Effector
The device at the end of a robotic arm, designed to directly interact with the environment, is known as the end effector (EE).

In-motion
Describes the operations/computations done while the robot is moving (i.e. inverse kinematics and trajectory generation).

Joint
Section of the manipulator system which allows one rotational degree of freedom.

Off-motion
An instruction fed to the robot by means of the human-to-machine input device. This command is interpreted by the robot's controller system. The proper instruction is then fed to the robot's actuators, which enable it to comply with the initial command.

Path Planning
Off-motion computation of an optimal path to reach a goal pose (i.e. while avoiding singularities and collisions).

Payload - Maximum
The maximum mass that the robot can manipulate at a specified speed, acceleration/ deceleration, center of gravity location (offset), and repeatability in continuous operation over a specified working space, specified in kilograms.

Pinch Point
Any location on the robot arm (or its accessories) which poses a risk of injury to fingers or other appendages close by.

Point-to-Point
Manipulator motion in which a limited number of points along a projected path of motion is specified. The manipulator moves from point to point rather than along a smooth continuous path

Protection Zone
A volume in space where the robot can be limited in speed, acceleration and force.

Protection Zone Management
For assistive cases, removal of the part of the translation command directed towards the inside of the protection zone. Use of the fitness function to avoid the elbow getting inside the protection zone.

Redundancy
Occurs when the manipulator (robot) has more degrees of freedom than it needs to execute a given task.

Redundancy Optimization
One of the effective methods to avoid a singularity is to use the redundant degrees-of-freedom motion.

Safeties
Hardware current limits and torque mismatches which are checked to increase robot safety.

Singularity Avoidance
Strategy to avoid configurations where the robot loses its ability to move the end effector in a given direction no matter how it moves its joints.

Tool
A working apparatus mounted to the end of the robot, such as a hand, gripper, welding torch, screw driver, etc.

Control and Operation Modes

Angular Mode
Independent joint control, whereby each axis of the manipulator is controlled separately.

Cartesian Mode
Translation and orientation of the end effector (EE) are defined in the task space. Then, a generalized Inverse Kinematics (IK) solver determines the robot joint movements needed to reach the target motion.

Cartesian Admittance Mode
Allows the application of external force to the EE, so as to guide the arm to a new position.

Joint Admittance Mode
Allows the application of external force at the links, without affecting the EE pose.

Null Space Admittance Mode
Robot configuration can be changed by applying external forces at the links without affecting the EE pose.

Trajectory Mode
A specific trajectory in the robot Cartesian workspace (e.g., a straight line trajectory for the end effector) can be defined bu the user.

Control features

This section gives an overview of control features of the robot.

The robot has the following control features that improve the safety and usability of the robot, and protect it from damage:

- singularity avoidance
- protection zones

Singularity avoidance

This section describes the Singularity avoidance feature of the robotic arm.

A singularity refers to any robot configuration (set of joint angles/orientations) which causes the Jacobian transformation matrix relating actuator rotations to end effector linear motion to be ill-conditioned, thus rendering the solution mathematically unstable (determinant of the joint space to Cartesian space Jacobian matrix becomes 0).

At a singularity, the mobility of the robot is reduced, meaning the arbitrary motion of the manipulator in a Cartesian direction is lost (losing a degree of freedom). This occurs when two or more robot axes are aligned, leading to unpredictable / extreme velocities when trying to attain a certain Cartesian pose. For example, when two axes become aligned in space, rotation of one can be canceled by counter-rotation of the other, leaving the actual joint location indeterminate. Near a singularity a small linear end effector motion requires disproportionately large angular velocities of the actuators.

The robot controller firmware features capabilities to handle / avoid singularities in any 'Cartesian' mode. As a singularity cannot occur unless inverse kinematics are calculated, singularities do not occur in any of the 'joint' modes.

Note: The robot behavior may change somewhat at or near a singularity. For example, the tool speed may be reduced or the motion may deviate from the commanded motion.

Protection zones

This section describes the protection zones feature of the robot.

With this feature, the user defines protection zones programmatically or by using the *Web App*, based on a few basic geometric shapes. Moreover, the user can specify a threshold, or slow-zone, in the area surrounding each protection zone.

The end effector of the robot will never enter protection zones. If the robot is commanded to enter or pass through a protection zone, the arm movement will stop any motion toward the protection zone at the outer boundary of the protection zone. Protected parts of the robot will be able to "slide" on the outer surface of the zone but not enter inwards.

The robot can travel however within adjacent slow-zones, but at a reduced speed.

One or more protection zones can be configured to define geometric volumes about the robot base, where the motion of the robot end effector is either limited or precluded.

By defining suitable protection zones, the robot can be set to avoid collisions with known fixed obstacles in the immediate environment of the robot while in operation.

Protection zones can be defined using one of three basic shape types:

- rectangular prism - position of center, length, width, and height dimensions, and angular orientation of the rectangular prism are configurable
- cylindrical - position of center, radius, height, and angular orientation of the cylinder are configurable.
- spherical - position of center and radius of sphere are configurable

A planar or disc-shaped protection zone can be defined by setting the thickness of the zone to zero in either a rectangular prism or cylindrical protection zone.

Protection zones can be defined, edited, and deleted using either the Web App or the developer APIs.

Control modes overview

This section gives an overview of the control modes of the robot.

The robot is controllable in a number of control modes:

- trajectory modes
  - angular trajectory
  - Cartesian trajectory
- joystick control modes
  - Cartesian joystick
  - angular joystick
- admittance modes
  - Cartesian admittance
  - joint admittance
  - null space admittance

The control mode can be set using API commands or the KINOVA® KORTEX™ *Web App*. Depending on the API command used, the robot control mode will change.The control mode will also change depending on the Virtual Joystick interface being used.

Trajectory control modes

This section describes the Trajectory Control modes of the robot.

Using Trajectory Control modes, the user can send a desired endpoint pose or set of desired joint angles to the robot. The robot controller computes an interpolated trajectory (between current pose and target pose) to reach the final position, and commands the robot to follow this trajectory.

In **Cartesian Trajectory** mode the endpoint pose is defined in terms of the desired Cartesian space position of the end of the robot. This mode enables singularity avoidance.

In **Angular Trajectory** mode the endpoint pose is defined in terms of the desired joint angles for the actuators.

To set the Trajectory control mode and execute trajectories, use the appropriate API methods, or the *Web App* Actions page.

Joystick control modes

This section describes the Joystick Control modes of the robot.

Joystick Control modes provide the user the ability to move the robot by sending speed commands or twist (velocity and angular velocity) to the robot using joystick commands or directly using cyclic commands.

In **Cartesian Joystick** mode the end effector of the robot is moved using twist (linear and angular velocity) or joystick commands. Linear speed is defined in the base frame, while angular speed is defined in the tool frame. This mode provides for singularity avoidance and obstacle avoidance (protection zones).

In **Angular Joystick** mode the joints of the robot are moved in angular space using angular velocity or joystick commands provided to the actuators. The joints can be moved individually or together.

In **Vision Joystick** mode, the end effector is made to move, both in translation and rotation, in relation to the Tool frame.

Joystick control modes are used by any connected gamepad and by the *Web App* virtual joysticks.

To enable Joystick Control Mode, use API methods or take control of the robot using either a connected gamepad or the *Web App* control panel virtual joystick controls.

Admittance modes

This section describes the Admittance modes of the arm.

By setting the control mode of the arm to admittance mode, the user can manually apply an external torque and/or force (wrench) to the arm and it will move accordingly.

In **Cartesian Admittance** mode, the end effector moves according to the wrench (Force + Torque) applied. This mode provides for singularity avoidance and obstacle avoidance (protection zones).

In **Joint Admittance** mode the joints of the arm move according to the torques applied.

In **Null Space Admittance** mode, the end effector stays in the same pose while the user manipulates the joints of the arm (within the null space). The arm moves within the null space according to the torques applied. This mode provides for singularity avoidance.

There are three ways to put the robot into admittance:

- use the method `SetAdmittance` in the API.
- Web App control panel admittance controls
- admittance mode physical buttons on the interface module.

Note: Motion in admittance modes is constrained by internal safety limits for the robot on velocity and torques. This includes Cartesian linear velocity limits and joint limits for angular velocity and torque. Admittance mode performance is also tuned using configurable parameters (damping, inertia, etc). The values for these parameters are not currently user-configurable, but will be available for configuration in a future software release.

Configurations and safeties

Configurable parameters

This section lists the configurable parameters of the robotand gives guidance on how to configure them.

The robotic arm includes a number of parameters that can be configured to customize the operation of the robot.

These parameters can be configured using the appropriate APIs. For more details on how to perform configuration using the APIs, see the API documentation..

Some of these parameters can also be configured using the *Web App* GUI, which can be accessed as follows:

1. Open the *Web App*
2. Navigate to the Robot Configurations page
3. Open the Configurations tab

The following tables give a summary of the configurable parameters.

Base configuration

Table 28: Base configuration (`Kinova.Api.Base`)

Configurable item	Description
User Profiles	Create, read, update and delete user profiles
Protection Zone	Create, read, update and delete protection zones (for obstacle avoidance). Configurable parameters are: • enabled / disabled • zone shape type (rectangular prism, cylinder, sphere) • zone dimensions • envelope thickness • zone limitation types (force, acceleration, velocity) and values • envelope limitation types (force, acceleration, velocity) and values
Control Mapping	Create, read control mapping
Action	Create, read, update, delete action
Sequence	Create, read, update, delete a sequence of actions
IPv4	Set IPv4 configured (for specified network adapter): • IP address • subnet mask • default gateway
Communication Interface	Enable communication interface: • network type (Wi-Fi or Ethernet) • enabled/disabled

Configurable item	Description
Wi-Fi	Set: • SSID • security key • automatic connection allowed
Admittance	Set robot admittance mode (Cartesian, null-space, angular). This allows you to move the robot by applying forces and torques by hand to the robot and joints.
Twist Wrench Reference frame	Set reference frame to use with Twist and Wrench commands. Set to either Tool or Mixed.
Operating mode	Set operating mode (Run or Update)
Servoing mode	Set servoing mode (high-level, low-level)
Factory defaults	Delete all configuration and restores factory defaults

Actuators configuration

Table 29: Actuator configuration (`Kinova.Api.ActuatorConfig`)

Configurable item	Description
Axis offsets	Set actuator axis offset position
Torque offset	Set actuator torque offset value
Control mode	Set actuator control mode. Options: • position • velocity • current • custom
Activated control loop	Define the contents included in activated control loop. Set bit mask, 1 to include the data item, 0 to exclude: • bit 0: joint position • bit 1: motor position • bit 2: joint velocity • bit 3: motor velocity • bit 4: joint torque • bit 5: motor current
Control loop parameters	Configure an individual control loop parameter (joint position, motor position, joint velocity, motor velocity, joint torque, or motor current). Configure: • error saturation value • output saturation value • kAz • kBz

Configurable item	Description
Vector drive parameters	Set vector drive parameters: • kpq • kiq • kpd • kid
Encoder derivative parameters	Set encoder derivative parameters: • maximum window width • minimum encoder tick count
Command mode	Set command mode. Options: • cyclic • asynchronous • cyclic jitter compensation using only position inputs • cyclic jitter compensation using position and velocity inputs
Servoing	Enable servoing.

Device configuration

Table 30: Device configuration (`Kinova.Api.DeviceConfig`)

Configurable item	Description
Run mode	Set device run mode (Run, Calibration, Configuration, Debug, Tuning)
IPv4 settings	Set device IPv4 address, subnet mask, default gateway

Vision configuration

Table 31: Vision configuration (`Kinova.Api.VisionConfig`)

Configurable item	Description
Sensor	Set several discrete vision sensor settings: **Color sensor** • resolution - 320 x 240, 640 x 480, 1280 x 720, 1920 x 1080 • frame rate - 15, or 30 fps • bit rate - 10, 15, 20, or 25 Mbps • focus mode - manual, continuous normal, or continuous extended **Depth sensor** • resolution - 424 x 240, 480 x 270 • frame rate - 6, 15, or 30 fps **Note:** Higher frame rate or resolution requires a higher data rate for best results
Options	Set the value of any **one** of a number of camera settings options. **Color image settings**: • brightness • contrast • saturation

Configurable item	Description
	Depth camera settings • exposure • gain • enable auto-exposure • visual presets • frames queue size • enable error polling • enable output trigger • depth unit setting in m • stereo depth camera baseline distance in mm between 1^{st} and 2^{nd} images

Safety items

This section is a reference for Safety items viewable and configurable in the *Web App* Configuration page.

Overview

Safety items, and their associated warning and error thresholds are viewable within the Configuration page of the *Web App*. There are three categories of safeties:

- Base (controller) safeties
- Actuators safeties
- Interface module safeties

The tables that follow give more information about the safeties, including:

- **Description -** significance of the safety item
- **Hard limit (lower)** - the minimum allowable value for the item
- **Hard limit (upper)** - the maximum allowable value for the item
- **Default warning / error threshold** - default configurations for the safety thresholds.

Base (controller) safeties

The following Base-related Safety items are viewable in the *Web App*.

Table 32: Base Safety items

Safety Item	**Description**	**Hard limit** (**lower** / **upper**)	**Default threshold** (**warning** / **error**)
Firmware Update Failure	Indicates a failure in the firmware update process.	n/a	n/a
Maximum Ambient Temperature	The ambient temperature is above upper limit	0.0 °C 90.0 °C	70.0 °C 80.0 °C
Maximum Core Temperature	The core temperature is above upper limit	0.0 °C 100.0 °C	75.0 °C 85.0 °C
Joint Fault	At least one joint is in a fault state	n/a	n/a
Joint Detection Error	The number of detected joints does not match the configured arm joint count.	n/a	n/a
Network Initialization Error	Arm is detected but control bus link is down	n/a	n/a
Maximum Current	The base current reading is above upper limit	0.0 A 12.0 A	9.0 A 10.0 A
Minimum Voltage	The base voltage reading is below lower limit **Note:** The minimum voltage must be lower than the maximum voltage	16.0 V 24.0 V	18.0 V 16.0 V
Maximum Voltage	The base voltage reading is above upper limit. **Note:** The maximum voltage must be higher than the minimum voltage	24.0 V 31.0 V	30.0 V 31.0 V
Emergency Stop Activated	Emergency stop activated. **Note:** electronic protection cannot be deactivated	n/a	n/a
Inrush Current Limiter Fault	Inrush current limiter fault triggered. **Note:** electronic protection cannot be deactivated	n/a	n/a

Actuators safeties

The following actuator-related Safety items are viewable and configurable in the Web application.

Table 33: Actuator Safety items

Safety Item		Description	Hard limit lower / upper	Default threshold warning / error
Following Error		The error between the command and the reported position is above upper limit. Note: Only active when in servoing state	0°	3.0°
			10°	5.0°
Max Velocity	Small	The computed velocity of the actuator is greater than threshold °/sec.	0 °/s	180 °/s
			250 °/s	200 °/s
	Large		0 °/s	100 °/s
			150 °/s	120 °/s
Maximum Torque	Small	Torque reading higher than x N·m.	0 N·m	29.4 N·m
			52 N·m	51.3 N·m
	Large		0 N·m	56.7 N·m
			105 N·m	104.5 N·m
Magnetic Position		Position step of more than threshold °/ms has been read on the magnetic sensor.	0°	3.0°
			20°	5.0°
Hall Position	Small	Position step of more than threshold °/ms	0°	n/a
			10°	0.4285°
	Large		0°	n/a
			10°	0.2145°
Hall Sequence		Invalid Hall sequence detected.	n/a	n/a
Input Encoder Hall Mismatch		The Hall sensor position value doesn't match the input optical encoder position within +/- threshold degrees.	0°	1.5°
			10°	2.0°
Input Encoder Index Mismatch		Input encoder index position mismatch	0	500
			2000	1000

Safety Item		Description	Hard limit lower / upper	Default threshold warning / error
Input Encoder Magnetic Mismatch		The input optical encoder position value doesn't match with the magnetic encoder position within +/- threshold degrees.	0°	10°
			45°	15°
Maximum Motor Current	Small	The measured current of the motor is above upper limit	0 A	6.0 A
			8.0 A	7.0 A
	Large		0 A	10.0 A
			12.0 A	11.0 A
Minimum Voltage		The voltage reading is below lower limit. **Note:** The minimum voltage thresholds must be lower than the maximum voltage thresholds.	16.0 V	18.0 V
			24.0 V	16.0 V
Maximum Voltage		The voltage reading is above upper limit. **Note:** The maximum voltage thresholds must be higher than the minimum voltage thresholds.	24.0 V	30.0 V
			31.0 V	31.0 V
Maximum Motor Temperature		Motor temp is above upper limit	0.0 °C	60.0 °C
			80.0 °C	75.0 °C
Maximum Core Temperature		Core temp above upper limit	0.0 °C	80.0 °C
			100.0 °C	90.0 °C
Non-Volatile Memory Corrupted		Non-volatile memory corrupt	n/a	n/a
Motor Driver Fault		Driver chip reported a major fault **Note:** electronic protection cannot be deactivated	n/a	n/a
Emergency Line Asserted		Emergency line asserted. Motor drive disabled	n/a	n/a
Watchdog Triggered		Watchdog was triggered	n/a	n/a

Interface module safeties

The following Interface module-related Safety items are viewable and configurable in the Web application.

Table 34: Interface Safety items

Safety Item	Description	Hard limit	lower / upper	Default threshold	warning / error
Maximum Motor Current	The measured motor current in the connected 3rd party gripper (if compatible gripper is attached) is above the higher limit. If gripper is not present the safety is disabled.	n/a		n/a	
Maximum Motor Temperature	The motor temperature of the connected 3rd party gripper (if compatible gripper is attached) is above the higher limit. If gripper is not present the safety is disabled.	n/a		n/a	
Minimum Voltage	Voltage reading below lower limit **Note:** minimum voltage thresholds must be below maximum voltage thresholds	16.0 V 24.0 V		18.0 V 16.0 V	
Maximum Voltage	Voltage reading above upper limit **Note:** maximum voltage thresholds must be above minimum voltage thresholds	24.0 V 31.0 V		30.0 31.0	
Maximum Core Temp.	Core temperature above upper limit	0.0 °C 100.0 °C		80.0 °C 90.0 °C	
Non-Volatile Memory Corrupted	Non-volatile memory corrupt	n/a		n/a	
Emergency Line Asserted	Emergency line asserted. Motor drive disabled	n/a		n/a	
Watchdog Triggered	Watchdog triggered	n/a		n/a	

KINOVA® KORTEX™ *Web App* User Guide

Introduction

The following sections describe the KINOVA® KORTEX™ *Web App*. The *Web App* is a useful interface for controlling, configuring and monitoring the robotic arm.

This pages that follow describe the purpose, layout, and use of the *Web App*.

Purpose

This section describes the purpose of the *Web App.*

The *Web App* is an HTML GUI (Graphical User Interface) that runs on the robot. This web interface allows users to configure, control and monitor the robot through a web browser interface from a computer connected to the robot over a wired Ethernet or Wi-Fi connection.

Device availability of *Web App*

This section describes the device availability of the *Web App.*

The *Web App* is a responsive web application. It is designed to adapt itself to various aspect ratios and resolutions enabling it to run on multiple platforms that support the Google Chrome browser. This includes:

- desktop / laptop computer
- tablet computer
- smartphone

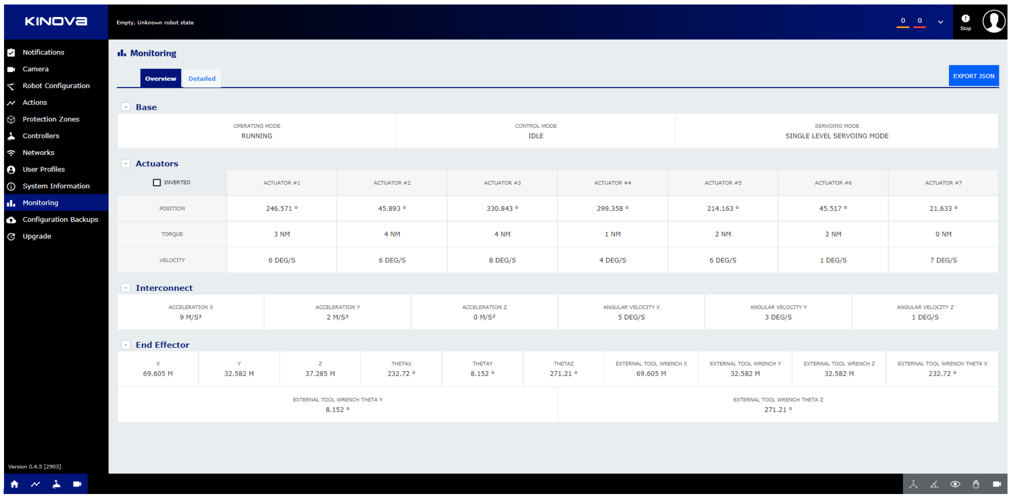


Figure 28: Desktop

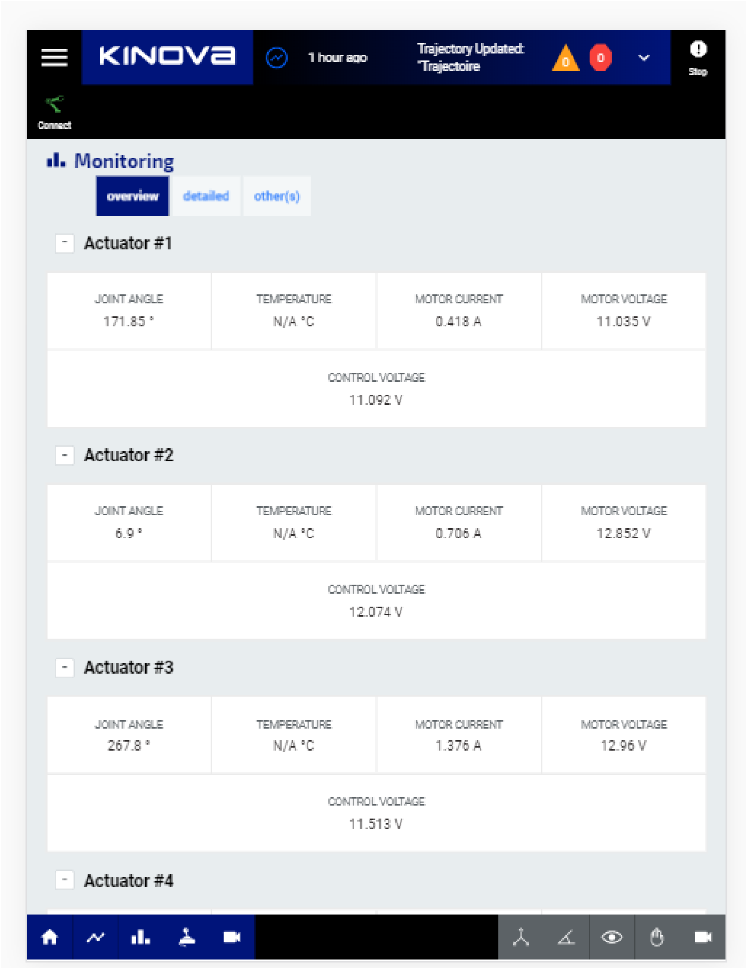


Figure 29: Tablet

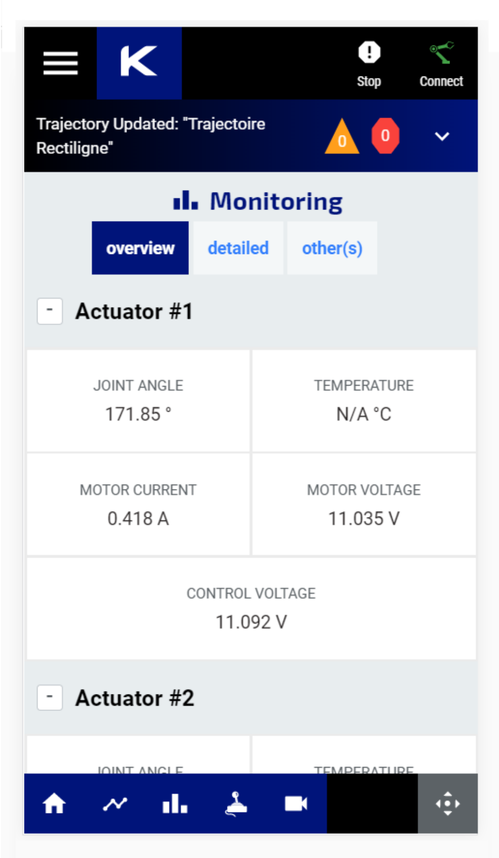


Figure 30: Smartphone

Platform and browser support

This section describes platform and browser support for the *Web App.* The

Web App has the following platform and browser support.

Browser support

Support for Google Chrome version 64+ is available on the following platforms:

Operating system support

- Microsoft Windows 7/8/10
- Ubuntu LTS 16.04
- Android 8.1 and higher

Other browsers and platforms are not currently supported - some features may work differently in those cases. Support for other major browsers is planned for the near future.

User login

This section describes how to log in to the *Web App.*

After establishing a network connection between your device and the robot, open a web browser and enter the IP address for the robot base external interface.

The *Web App* will launch, ending in a login popup.

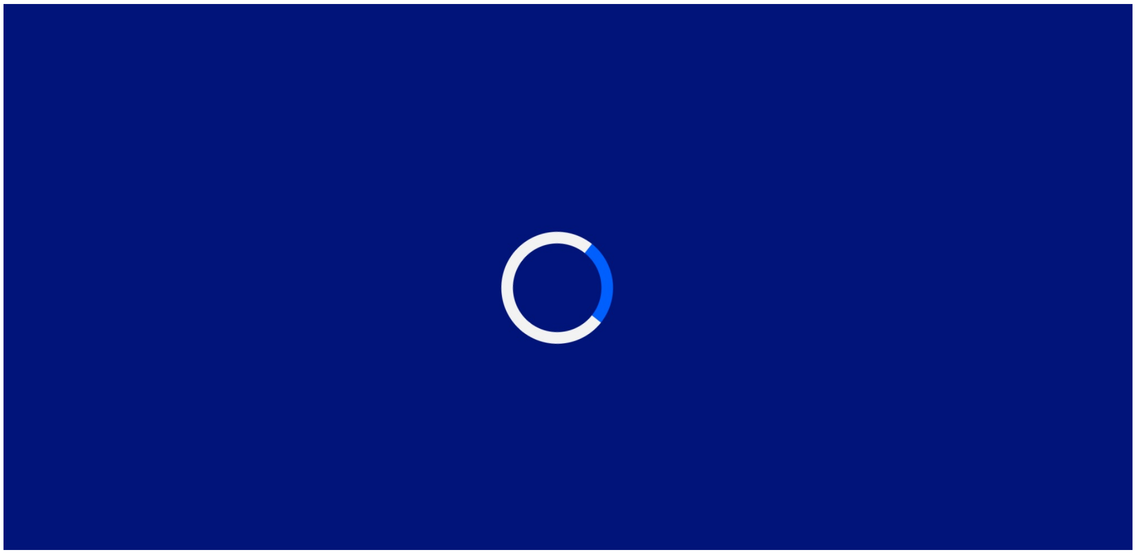


Figure 31: Launching...

Enter your user name and password and press the CONNECT button.

The default username and password when the robot first arrives are:

- username: `admin`
- password: `admin`

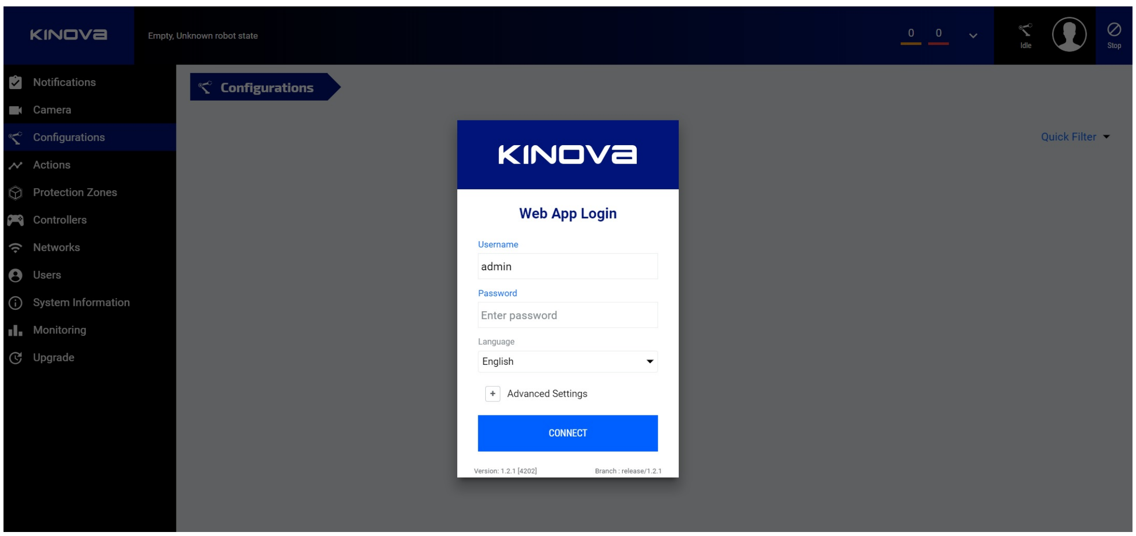


Figure 32: User login

On pressing CONNECT, the *Web App* will launch and initialize. While it is doing this, the *Web App* will give visual feedback to the user on the status of initialization of the application and loading of robot configurations into the application.

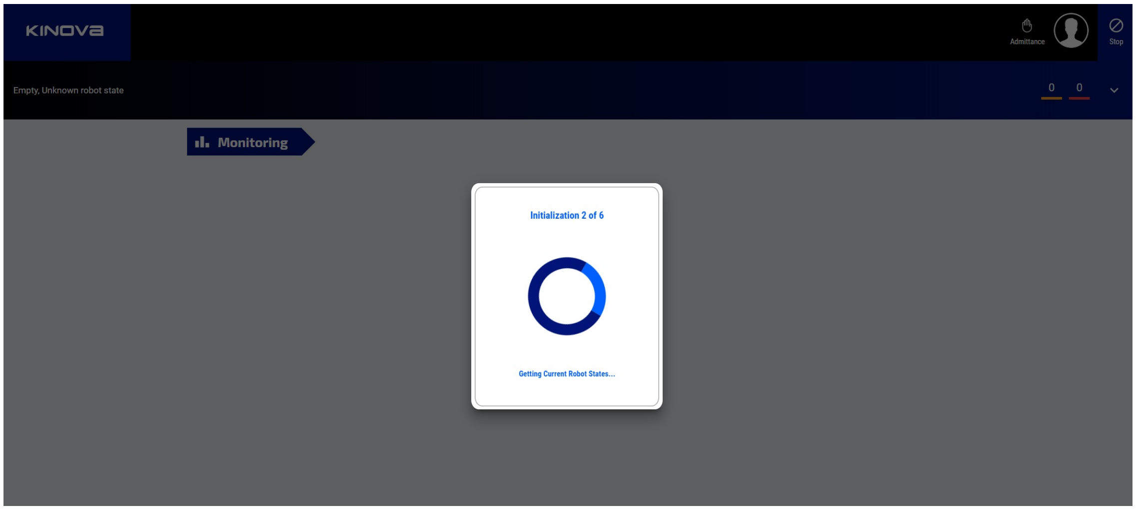


Figure 33: Initializing...

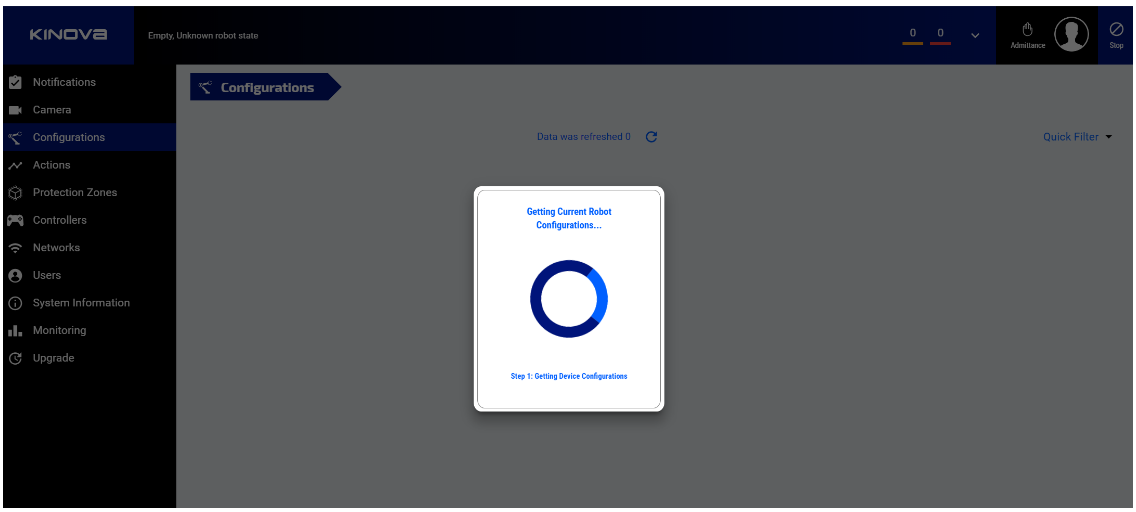


Figure 34: Getting robot configurations

Web App layout and navigation

This section describes the layout and navigation of the *Web App*.

The *Web App* screen is divided into several main sections:

- Main navigation panel
- Main information panel
- Notification bar
- Shortcuts panel
- Robot control panel
- Mode indicator, user icon, and E-stop

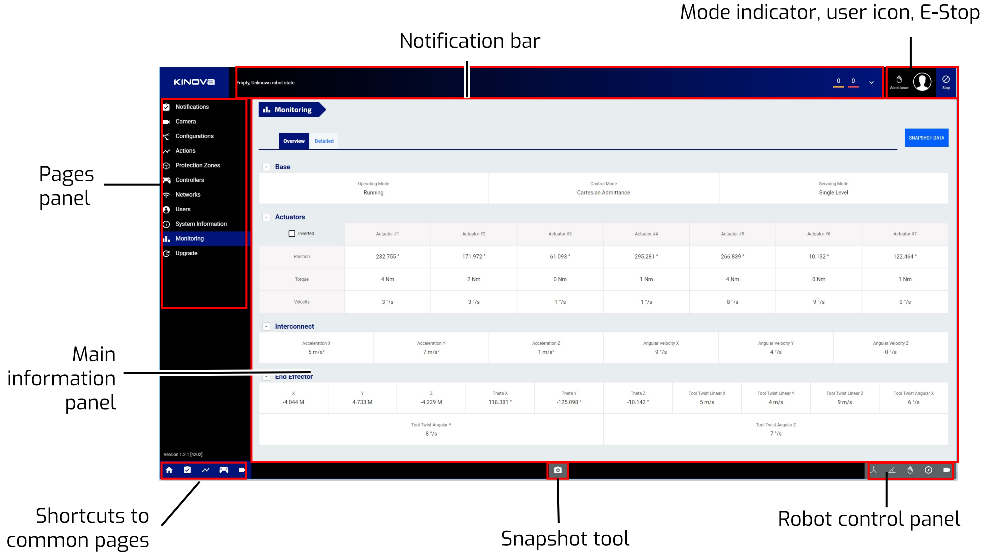
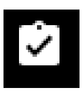
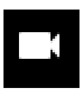
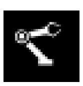
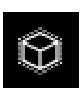
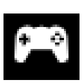
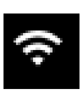


Figure 35: Web application Layout

Pages panel

In the middle of the screen is the main information panel containing the contents of each page of the application.

The pages panel on the left of the screen gives easy access to the main pages of the application:

-  Notifications
-  Camera
-  Configuration
-  Actions
-  Protection Zones
-  Controllers
-  Networks
-  Users
-  System Information
-  Monitoring
-  Upgrade

Note: When the screen or window size is small (on a tablet or smartphone browser, or when the app is viewed in a small desktop window), the navigation panel will be hidden by default. If you click on the menu icon in the upper left hand corner of the screen, the navigation panel will be made visible.

Information about the current version of the *Web App* is visible below the main navigational panel. Version 1.2.1 [4202]

Shortcuts panel

A small panel of shortcut icons to useful common pages appears in the lower left of the screen, as follows:

- Home (monitoring page) - information panel for monitoring the robot stats
- Notifications page - view notifications for the robot during the session
- Actions - Create actions: poses and sequences of poses of the robot that can be saved and replayed
- Controllers - configure and choose active gamepad control mode
- Camera - view camera feed from robot video module

Notification bar

The notification bar at the top of the page gives a visible, high-level summary of information in the Notifications Center. It displays the most recent events coming from the robot, as well as a count of outstanding warnings and errors.

In the upper right hand corner of the screen are three indicators / controls:

- control mode - status of the control mode situation of the robot. There are four icons to indicate the mode / state:
  - Fault Fault - robot is in fault
  - Idle Idle - robot is not currently being controlled by any user session; waiting
  - Running Running - robot being actively controlled in Cartesian or angular control
  - Admittance Admittance - robot is being controlled in an admittance mode
- User profile icon - Shows the user session icon
- Stop Emergency Stop (E-stop) - button control which when pressed / tapped will initiate the emergency stop of the robot.

Clicking on any of these items displays a pop-up showing further information.

Robot control panel

In the lower right is the virtual joysticks / admittance panel. This panel consists of a group of five buttons. Three are to launch pop-up windows for virtual joystick controls and admittance mode toggles

- Cartesian Virtual Joystick
- Angular mode control
- Admittance controls

The virtual joysticks allow you to control the movement of the robot without the use of a physical controller.

Admittance mode lets you to move the robot with your hands in one of three (Cartesian, joint, and null space) admittance modes.

In the same area there are two other controls:

- Play action - play a selected action
- Camera feed - view camera feed

Clicking on one of the buttons in the robot control panel will launch a smaller window from the bottom of the screen, revealing the selected control panel.

Clicking the same button again will clear the smaller window at the bottom.

Snapshot tool

The snapshot tool gives the ability to capture a "snapshot" of a robot pose. This is a useful feature to help with building pre-set sequences (for demos or to capture / program a fixed set of movements). There are three types of snapshots:

- Cartesian pose
- Joints pose
- Gripper pose

The snapshot tool is available at the bottom on all pages.

Robot control panel

Cartesian virtual joystick control

This section describes the Cartesian virtual joystick control interface of the *Web App.*

The Cartesian virtual joystick panel allows users to control the position and orientation of the end effector through the *Web App* using a mouse (on laptop or desktop computer) or touch control (on a tablet or smartphone).

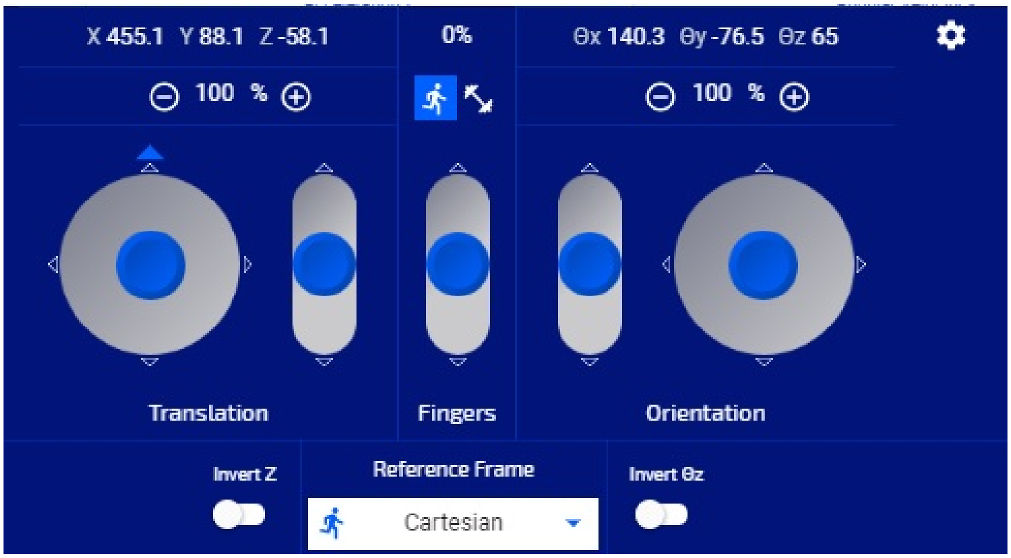
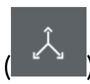


Figure 36: Cartesian joystick panel

The Cartesian virtual joystick panel is launched by clicking the first button () on the robot control panel.

Translation / orientation joystick controls

The Cartesian virtual joystick controls allow you to control the translation and orientation of the end effector. There are two sets of joysticks:

- translation (to apply a translation to the end effector)
- orientation (to apply a rotation to the end effector at the current position)

Together, the three position coordinates and the three orientation directions specify the Cartesian pose of the robotic arm.

Each set of joysticks features a 2-axis joystick for controlling the x and y axes, and a 1-axis joystick to control in the z-axis. For the 2-axis translation joystick, the user can configure the joystick axis that is assigned to control the y direction movement.

As the controls are moved, a display is provided for the current position (x, y, z) and orientation (θ_x, θ_y, θ_z) of the end effector.

Finger controls

It is possible to open and close the fingers using a single 1-axis joystick control (if a gripper is installed). Push the control up to open the fingers and down to close. The fingers position can be controlled between 0% (fully closed) and 100% (fully open).

Additional settings

By clicking the settings gear icon, additional settings controls are revealed. These let the user change the position reference frame (Cartesian or tool), or switch the defaults for the z and θ_z controls.

Z and θ_z toggles

The default for the 1-axis z-direction controls is that 'up' increases the z-position or z-angle, while 'down' decreases it. This can be reversed using the Invert Z and Invert θ_z toggles.

Reference frames

The position of the end effector can be specified in one of two reference frames:

- the fixed base reference frame
- the tool reference frame

The tool reference frame is useful when controlling the robot using visual feedback from the camera sensors. The joystick will control movement of the end effector with respect to the tool reference frame, which is very close to the reference frame of the camera. This makes it easy to command the robot in relation to what is seen through the camera.

The translational speed (for change of position) and rotational speed (for change of orientation) can be adjusted between 0 and 100% of the hard limits for the robot using the ⊕ and ⊖ buttons.

Angular virtual joystick control

This section describes the angular virtual joystick control interface of the *Web App.*

The angular virtual joystick panel allows users to control the robot joint angles through the *Web App* using a mouse (or touch control on a tablet or smartphone).

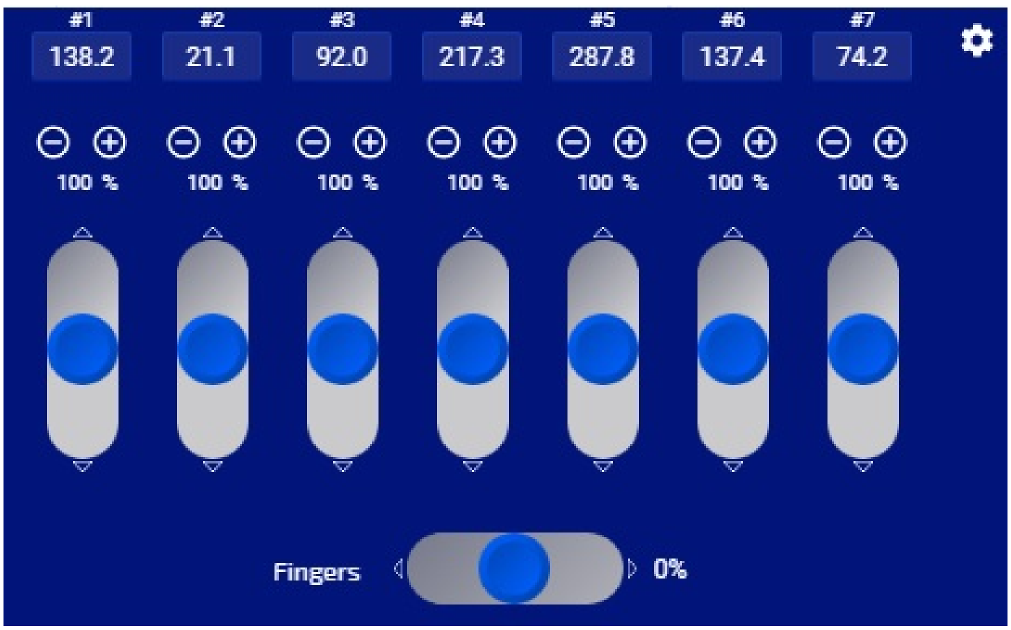


Figure 37: Angular virtual joystick panel

The joint angles can be controlled through:

- angular velocity (control the angular speed of each actuator)

The angular joystick panel is launched by clicking the second button on the robot control panel.

The virtual joystick controls allow you to control the angle of each actuator as well as the opening and closing of the fingers (if a gripper is installed). As the virtual joystick controls are manipulated, the robot arm joints respond accordingly.

Note: For joints 2, 4 and 6 there are physical limits to how far the joint can turn without the arm shell segments running into each other. The robot enforces software joint angle limits to prevent these joints from reaching the physical limits. When you control these joints, the software will cause the arm joints to stop responding when the limits are reached on the actual robot.

The value of each angle is displayed in degrees. The value displayed will be restricted to one full rotation.

The maximum angular speed (or torque) for each actuator can be adjusted using the ⊕ and ⊖ buttons. Each actuator can be adjusted between 0 and 100% of the hard maximums for the joints.

There are two ways to control the angle of each actuator:

- type in a numerical value
- use the virtual joystick controls to apply a velocity or torque in the given direction. Pushing the joystick up causes the angle to increase, while pushing it down causes it to decrease. The further up or down the joystick is pushed, the higher the angular speed (or torque applied) for the joint, up to the set limit. The angle will continue to change as long as the joystick is being pushed.

Another joystick allows users to control the end effector finger position (if an end effector is installed). The values for the finger state range between 0% (fully closed) and 100% (fully open). Push the joystick to the right to increase the percentage (and open the fingers). Push the joystick to the left to decrease the percentage (and close the fingers).

Virtual joystick keyboard shortcuts

This section describes keyboard shortcuts for the *Web App* virtual joysticks.

Introduction

The virtual joysticks for the *Web App* are controllable with mouse or touch inputs. Some people (particularly those with a background in PC gaming) may find it more natural to control using keyboard shortcuts. If you are accessing the *Web App* using a desktop device that has a keyboard (such as a desktop or laptop PC) there are handy keyboard shortcuts available for the joystick controls.

Cartesian joysticks keyboard shortcuts

Table 35: Pose translation joystick shortcuts

Control		Shortcut
X	increase translation	D
	decrease translation	A
Y	increase translation	W
	decrease translation	S
Z	increase translation	R
	decrease translation	F

Table 36: Pose orientation joystick shortcuts

Control		Shortcut key
X	increase angle	right arrow
	decrease angle	left arrow
Y	increase angle	up arrow
	decrease angle	down arrow
Z	increase angle	right shift

Control		Shortcut key
	decrease angle	right control

Angular joystick keyboard shortcuts

Control		Shortcut key
Actuator 1	increase angle	1
	decrease angle	Q
Actuator 2	increase angle	2
	decrease angle	W
Actuator 3	increase angle	3
	decrease angle	E
Actuator 4	increase angle	4
	decrease angle	R
Actuator 5	increase angle	5
	decrease angle	T
Actuator 6	increase angle	6
	decrease angle	Y
Actuator 7	increase angle	7
	decrease angle	U

Put another way, the angles for joints 1-7 can be increased using the keys 1-7 on the top of the keyboard.

The angles can be descreased using the letter keys QWERTYU on the top row of the keyboard.

Admittance modes panel

This section describes the admittance mode panel of the *Web App*.

The Admittance Mode panel is brought up by clicking the hand icon  in the robot control panel. The panel slides up from the bottom of the screen.

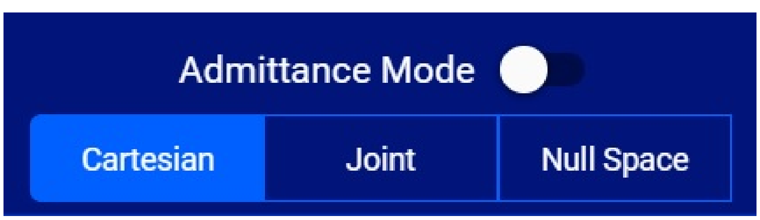


Figure 38: Admittance modes panel

An admittance mode is one in which the control systems of the robot take into account external force / torque feedback from its environment.

From this panel, the arm can be set into one of three types of admittance mode:

- Cartesian admittance mode - end effector moves linearly according to the wrench (force + torque) applied on the end effector
- Joint admittance mode - joints of the arm rotate according to the external torques applied at the joint

- Null Space admittance mode - end effector stays in the same pose while the user manipulates the joints of the arm (within the null space). The arm moves within the null space according to the external torques applied.

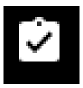
It is also possible to take advantage of the admittance mode to manually move the robot into a pose and then capture a snapshot of that pose for future reference (Normally, when not in an admittance mode, the joints will resist being manipulated by external forces / torques).

From the admittance mode panel it is possible to add a Cartesian or Angular pose by clicking the snapshot button .

Main pages

Notifications

This section describes the notification center of the *Web App.*

 The Notifications page is a central location for viewing all notifications related to the robot. The Notification Center can be accessed either from the main navigational panel or from the top notification bar (by selecting All).

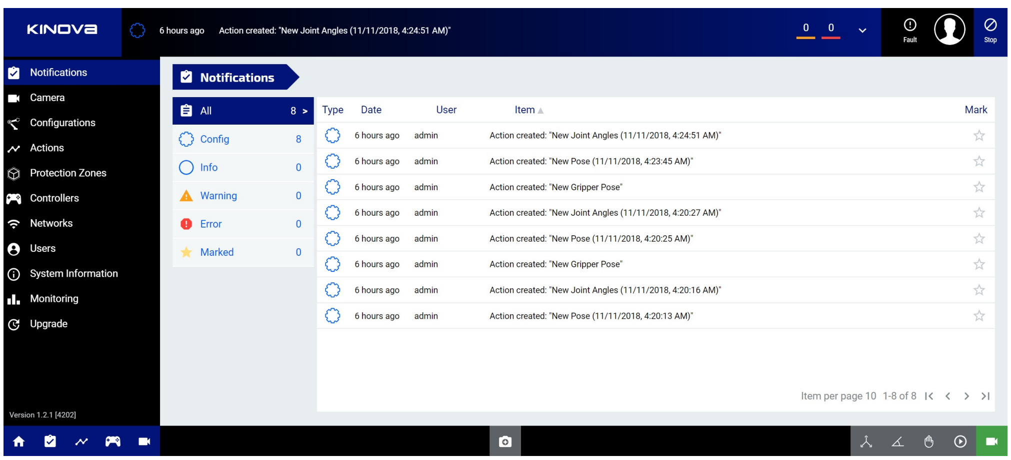


Figure 39: Notification Center

Users can either view all notifications, or view notifications by type. There are four types of notifications:

- Config - indicates a change in robot configuration such as creating, modifying, deleting, or activating a User Profile, Protection Zone, Action or Control mapping
- Info - information message for the user
- Warning - Warning state reached. Arm parameter is moving out of the normal range and will reach Error state if the trend continues.
- Error - Error state reached. Arm parameter is significantly outside normal bounds. Triggers an emergency stop.

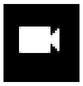
The notifications list appearing in the window can be sorted by Type, Date, User, and Item, either in increasing or decreasing order.

If there are many notifications of a particular type, they will be grouped into pages.

Note: In the current release of the web application, information displayed in the Notification Center is stored on the client device where the web application is accessed. The events displayed are only those that have occurred since the web application was opened. These events will be lost when the application is closed.

Camera

This section describes the Camera page of the *Web App.*

 The Camera page allows you to see the video feed from the installed vision module.

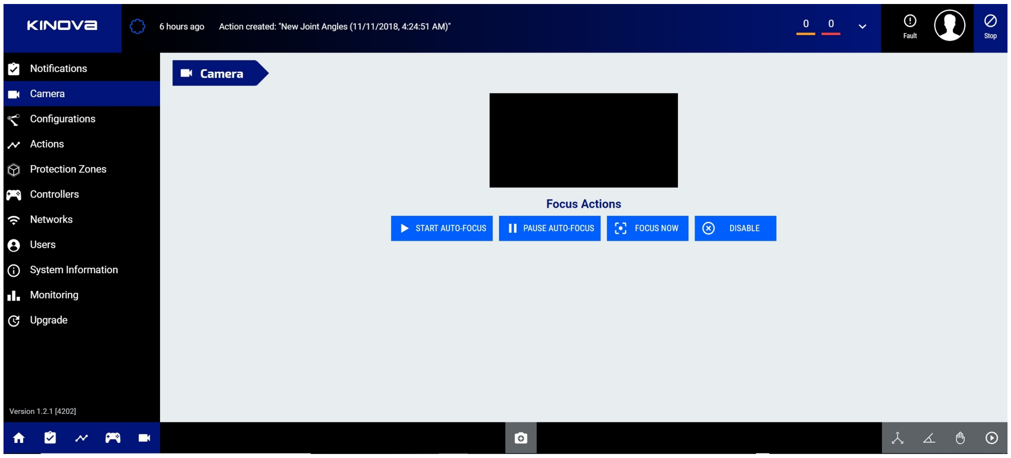
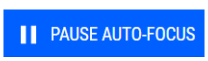
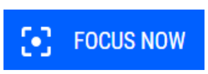


Figure 40: Camera page

There are four controls available on this page:

- 

Start auto-focus
- 

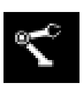
Pause auto-focus
- 

Focus now
- 

Disable auto-focus

Configurations

This section describes the Configurations page of the *Web App.*

 The Configurations page allows users to configure different aspects of the arm hardware and set safeties.

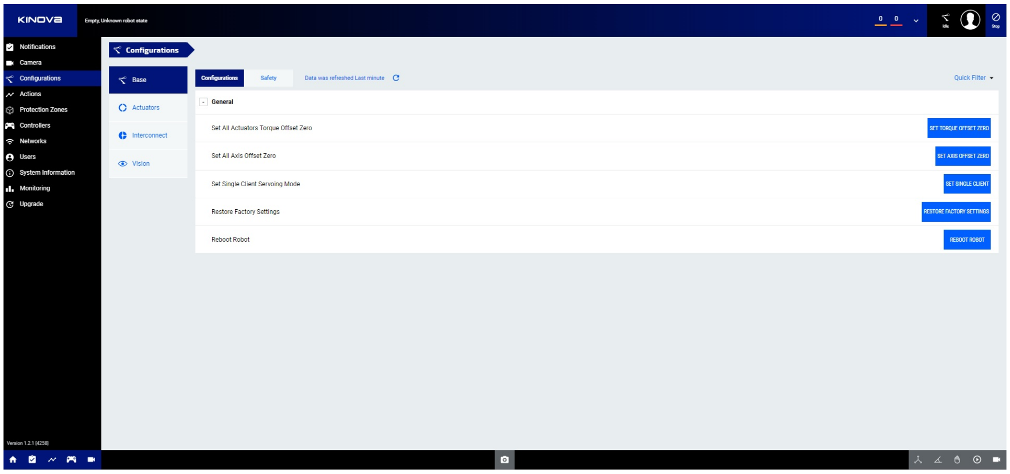


Figure 41: Configurations page

The settings menu is divided into four tabs:

- Base
- Actuators
- Interconnect (interface module)
- Vision

Each of these tabs contain two sub-tabs:

- Configurations
- Safety

Configurations tab

This section describes the Configurations tab of the *Web App* Configurations page.

Configurations

The Configurations tab of the Configurations page lets you use a GUI to adjust the configurable parameters of the robot hardware to customize its behavior.

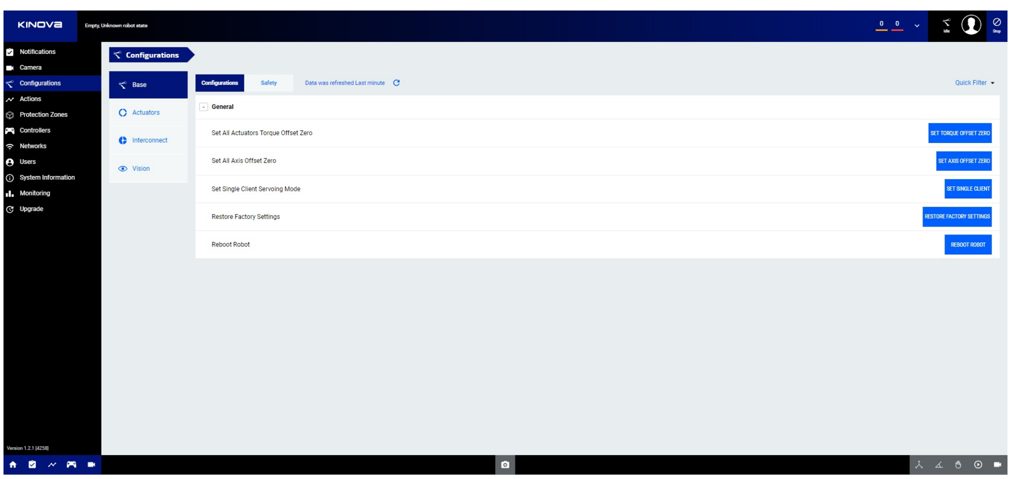


Figure 42: Configurations

The configurable items are broken into four sections:

- Base
- Vision
- Actuators

Most of the configurable parameters of the robot can be configured on this page. Some other configurable items are handled on their own pages:

- Protection Zones
- Control Mappings
- Actions
- User Profiles
- Network Settings

For more information on configurable parameters, refer to the section on Configurable parameters.

Safety tab

This section describes the Safety tab of the *Web App* configurations page.

Safety

The Safety tab allows users to view safety thresholds. There are two types of safety thresholds:

- **Error -** An error is a departure from normal parameters that is more serious than warnings and represents a situation which could damage the robot or endanger the user. The thresholds for errors are set at a more extreme level than warning thresholds.

  Note: An error triggers an emergency stop for the robot.
- **Warning -** A warning serves to signal that the robot is moving away from normal operational status toward an error state. A warning will not stop the robot.

  Note: Some safety items do not have warning thresholds, only error thresholds.

In the *Web App*, warning thresholds are marked in orange and error thresholds are marked in red.

Note: In the current release, safety items can be viewed but not modified in the *Web App*. In addition, all safety items are enabled and cannot be deactivated. It will be possible to modify and disable some safeties in a future release.

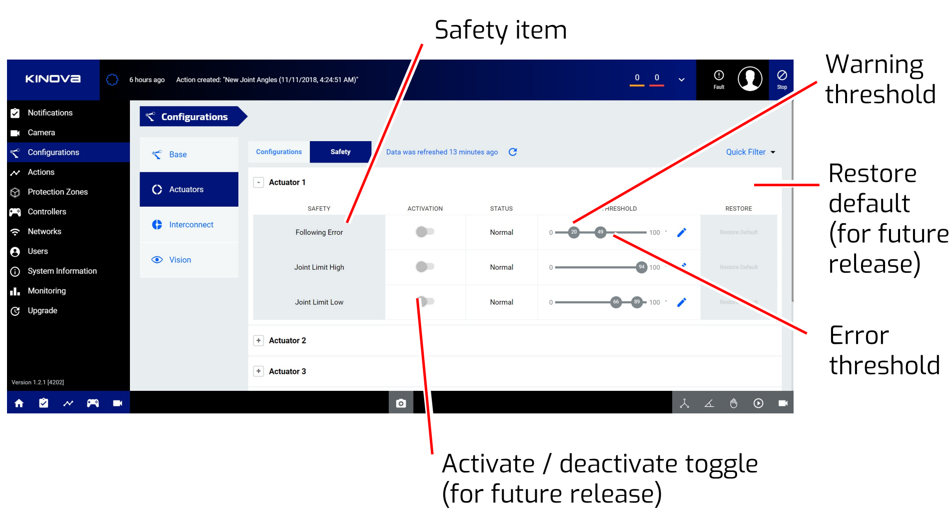


Figure 43: Safety tab contents

Activating / deactivating safety items

Not currently supported. To be supported in future release.

Restoring defaults

Not currently supported. To be supported in future release.

Detailed safeties information

For more detailed information on robot safety thresholds, see here.

Actions

This section describes the Actions page of the *Web App.*

The Actions page allows user to define, view, and edit robot actions, as well as build sequences and play back actions and sequences.

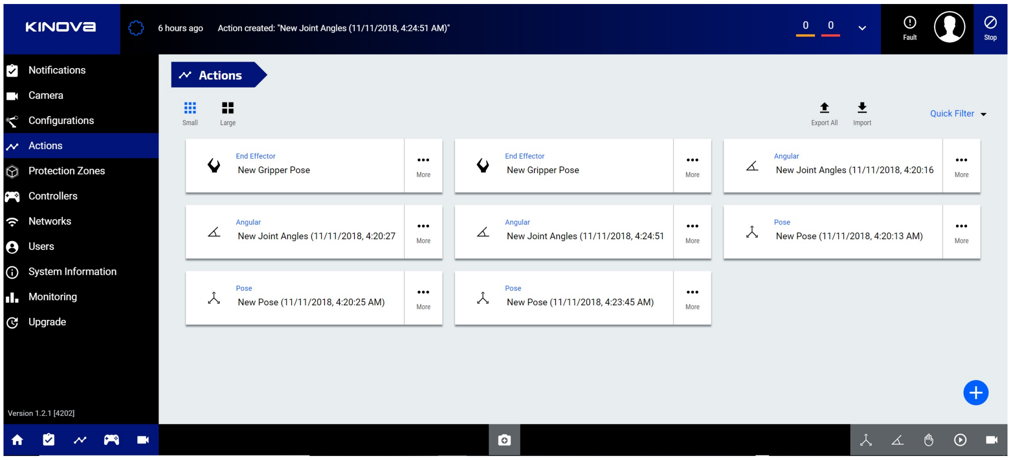


Figure 44: Actions page

Actions available in the *Web App* are:

- Pose (Cartesian pose)
- Angular (Joint angle combinations)
- End Effector (Gripper pose)
- Sequence

Cartesian poses represent a single Cartesian pose for the robot. A pose consists of x, y, and z coordinates representing the position of the end effector, and the three angles θ_x, θ_y, and θ_z representing the orientation of the end effector.

Angular represents the set of joint angles for each of the arm joints.

End effector represents the gripper state, from 0% (fully open) to 100% (fully closed).

Note: Currently the Robotiq 2-finger gripper is supported.

A **Sequence** is defined as a sequence of 'keyframe' poses on a timeline, which are interpolated by the software in the robot base to produce intermediate frames to generate smooth motion. Sequences are a sequential combination of Cartesian poses, angular settings, and end effector poses. Sequences may also include timed delays between movements.

The main information panel of the page shows cards with all the defined actions and sequences. New actions or sequences can be added with the + icon in the lower right of the main panel. This launches a menu where you can select the type of new item to create.

Figure 45: '+' menu

If the screen gets crowded after a while, you have the option to filter to show only the chosen types of actions.

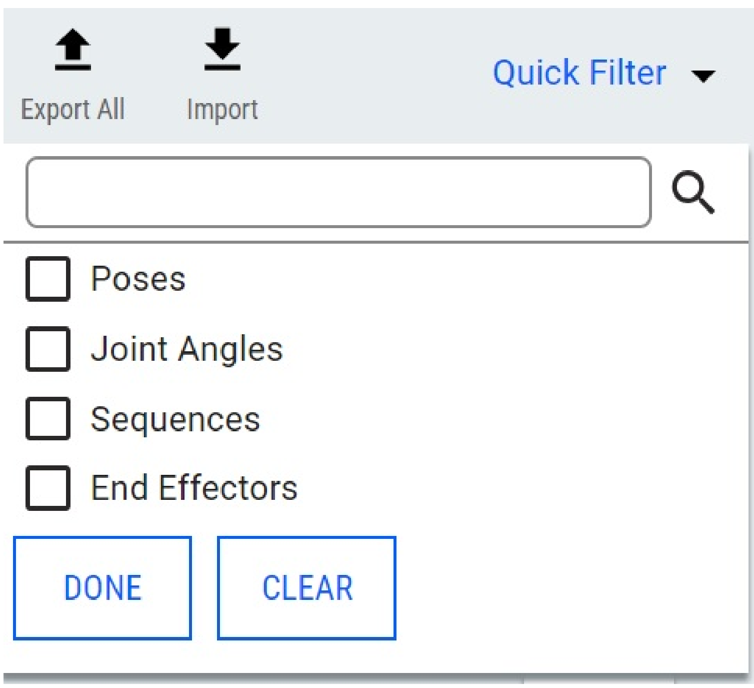


Figure 46: Action filter

Information about defined actions is viewable in information cards which come in two sizes, small and large, which can be toggled with buttons at the top of the main panel. The large cards carry full information about the action.

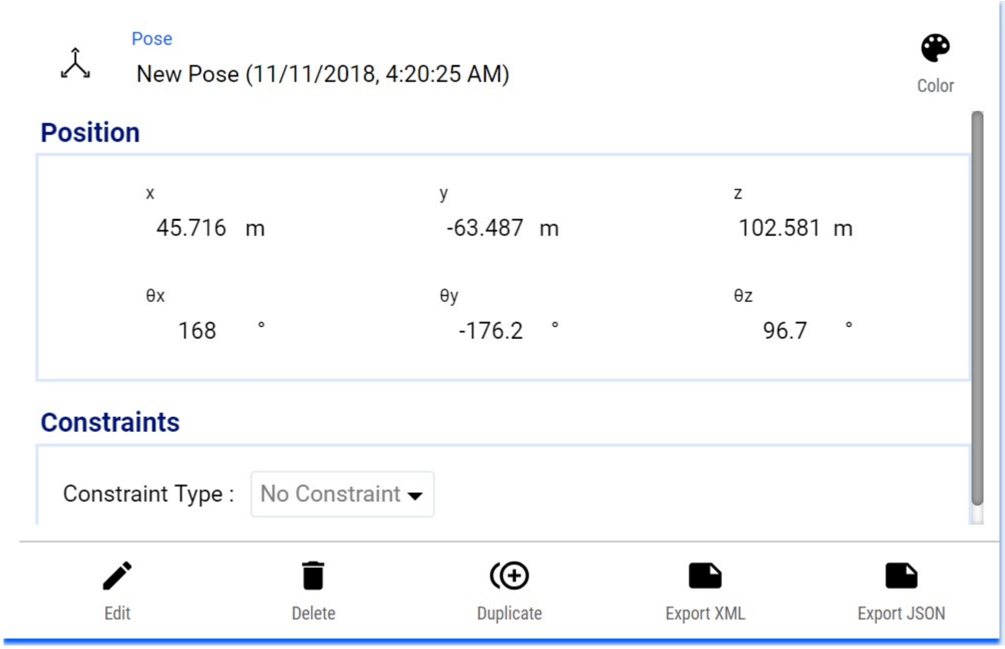


Figure 47: Information card

From the large cards, you are able to:

- edit the action
- delete the action
- duplicate the action
- assign a color code to the card
- Export a representation of the action to XML or JSON

Small cards on the other hand show a more compact view, with the functions other than Play hidden. By clicking More, you can access a pop-up with the remaining functions.

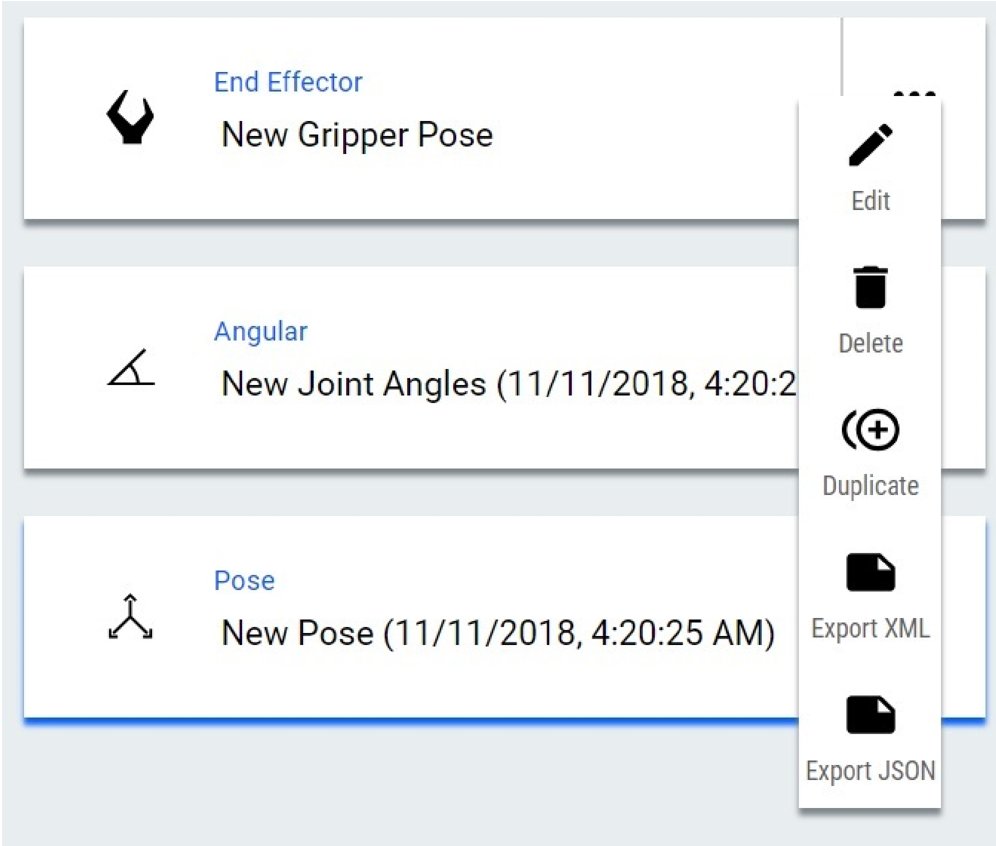


Figure 48: 'More' pop-up menu

Chosing Edit brings up an interface to modify the parameters of the action.

Figure 49: Parameter modifications

Creating actions using snapshot tool

The snapshot button () at the bottom of the screen can be used to capture the current robot Cartesian pose, angular pose, or gripper state. Any pose captured by the snapshot tool will show up on the actions page. For more information, see the snapshot tool page.

Sequence editor

This section describes the sequence editor on the *Web App* Actions page.

Choosing **More** > **Edit** for a sequence on the Actions page brings up a sequence editor panel.

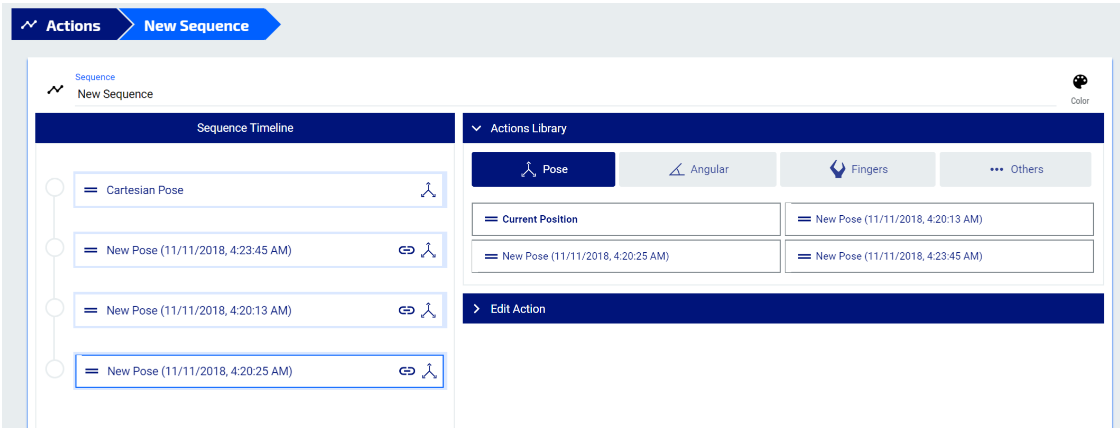


Figure 50: Sequence editor

The sequence editor is made up of three sections:

- sequence timeline
- actions library
- edit action

The **sequence timeline** shows the steps in the sequence. The link icon () on a step indicates that the action is a reference to an existing standalone action.

Note: If that standalone action is modified outside this sequence it will have an impact on the sequence.

The **actions library** shows the already saved actions, organized into tabs grouped by type. Clicking one of the available actions adds it to the sequence. There is also an option to add the Current Position to the sequence. Depending on what actions tab you are in, this will add the current position as a Cartesian pose, as joints position, or as gripper fingers position.

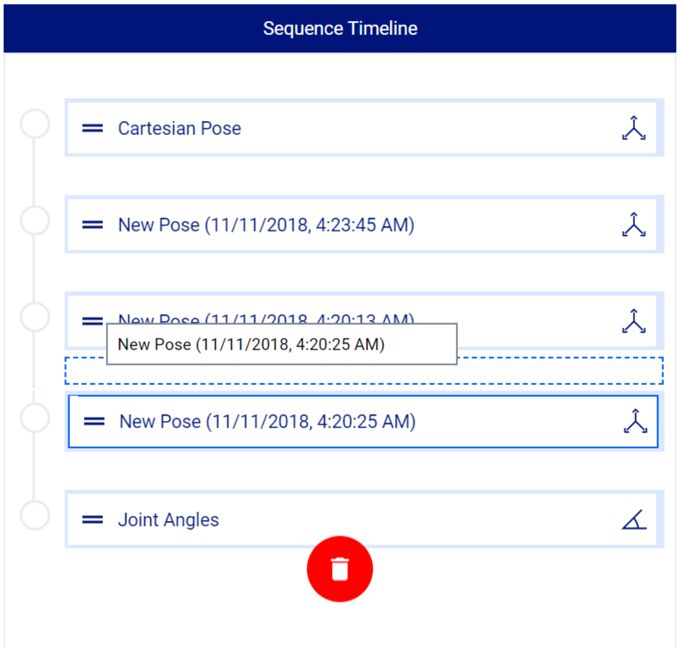
The edit action section lets you directly edit the parameters of the action.

The link icon indicates that the action referred to in the Sequence is a reference to an existing standalone action. If that standalone action is modified outside this sequence it will have an impact on the sequence. By opposition, if the action (for example the first one in your screen capture) does not have that link icon, it means that it does not exist outside the Sequence.

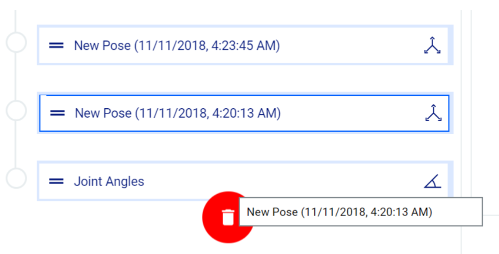
Deleting actions or re-ordering the steps of a sequence

If you want to change the position of an action in the sequence, or remove it from the sequence entirely, simply hover over the left side of the box for the action until the cursor changes from a hand icon to a two-dimensional arrow icon.

Click and drag to move the selected action. To move it to another position, drag the action and release it in the desired position. The dotted outline will indicate where the action is in the sequence.



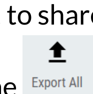
Notice that a red garbage can icon will also appear when an action is selected to move. To delete the action from the sequence, drag the action onto the garbage can until the garbage can visibly expands.



Then release to delete.

Importing and exporting actions or sequences

This section describes functionalities available on the Actions page for importing and exporting actions or sequences.

You can Export All defined actions or sequences as XML or JSON, to share with others. Similarly, a JSON or XML action file can be imported from the computer. The 

Export All and 

Import functions are available at the top of the main panel.

Playing back actions and sequences

This section describes how to play back actions and sequences

When an action or sequence is selected by clicking on its card, the item is loaded in a playback bar at the bottom of the page.

When the play button is pressed, the robot will move directly to execute the described action or sequence.

The hold to play toggle (by default, activated) controls the playback. When the toggled on, the playback will only continue as long as the play button is held down. When not toggled on, a single press of the play button will suffice for the playback to execute completely.

The pause button will stop the playback while keeping the playhead at the same position. When the play button is pressed again, the motion will continue exactly where it left off.

The stop button will stop the movement and return the playhead to the beginning.

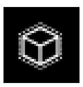
In the case of a Cartesian, angular, or gripper pose, the robot (or gripper) will interpolate linearly between the present pose and the target pose and move smoothly and directly to the target pose.

For a sequence, the robot will first go directly to the the first item in the sequence, and then will trace out a smooth path that goes through the poses on the sequence. A progress bar above the playback bar shows the progress of the playback through the steps.

For sequences, an additional looping control can be toggled on or off. When toggled on, a sequence will play through all the steps and then go directly to the pose of the first step. This is useful for demonstrations.

Protection Zones

This section describes the Protection Zones page of the *Web App.*



The Protection Zones page allows user to define three-dimensional geometric volumes about the robot where the robot either cannot go or where the maximum speed is reduced. A protection zone is intended to limit the possibility of the robot running into either the user or objects near the robot. Protection zones only work when controlling the robot in Cartesian mode - when controlling the robot in Angular mode, they are ignored.

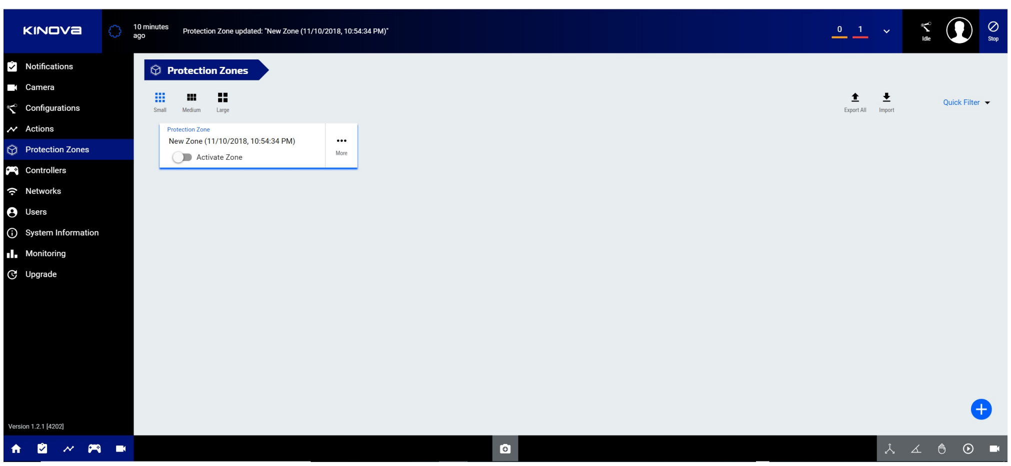


Figure 51: Protection Zones page

The Protection Zones page allows for defining multiple protection zones. The defined protection zones are displayed as cards in the main information panel. The cards can be displayed in three different sizes:

- Small
- Medium
- Large

The card sizes can be toggled using buttons at the top of the main information window. The large cards show fuller detail about the settings for the protection zone, namely whether the Envelope Translation Speed Limit and Envelope Orientation Speed Limit are turned on. It also includes controls to activate, edit, delete, and duplicate protection zones, as well as to apply a color code to the card.

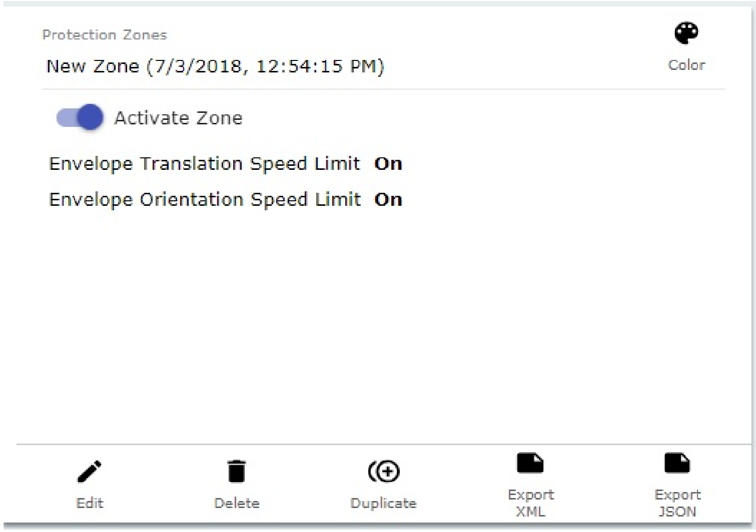


Figure 52: Protection Zone card

Medium cards are somewhat smaller than the large, with all the same information and controls, except for the ability to apply a color code to the card.

Small cards show a more compact view. Only the name of the protection zone and a toggle to activate the zone are displayed. By clicking More, a pop-up menu gives options to edit, delete, or duplicate the protection zone.

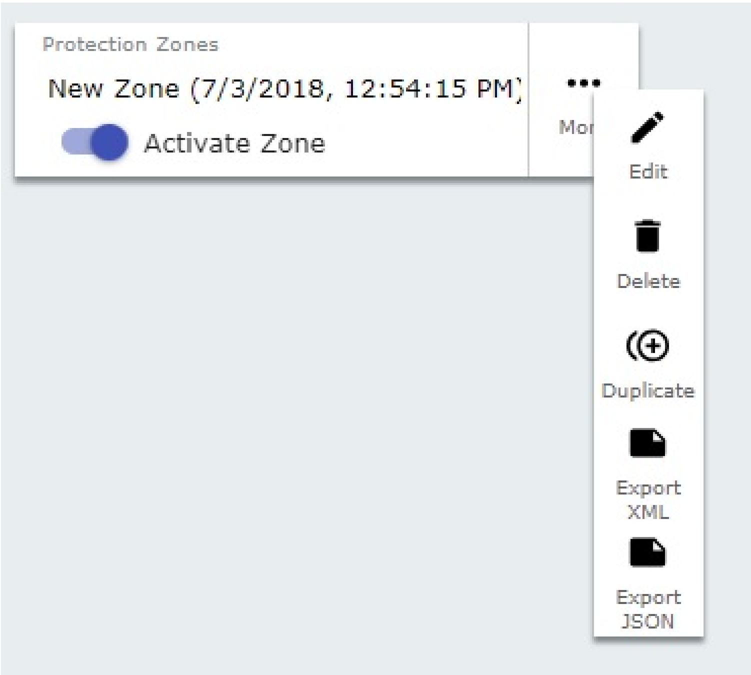


Figure 53: 'More' menu

Multiple protection zones can be active at the same time.

A new protection can be added by clicking the + sign icon in the lower right of the main information panel. This will create a new, empty protection zone, and add a new card to the page. By clicking Edit, a new interface window is launched where you can configure the protection zone.

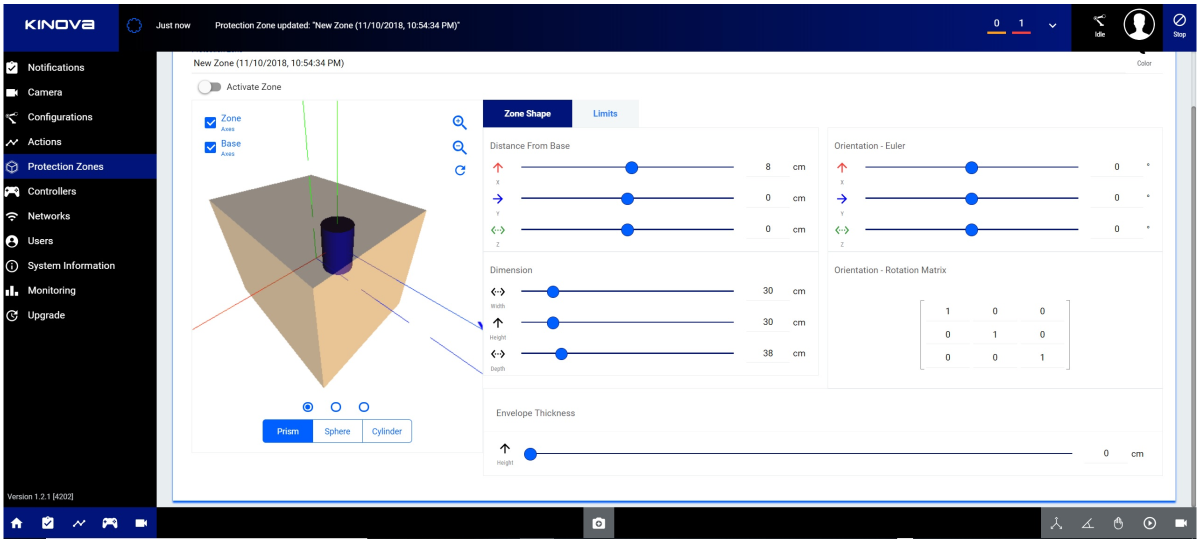


Figure 54: Configure protection zone

There are two tabs:

- Zone Shape - for defining the protection zone geometry
- Limits - for setting limits on velocity within an envelope surrounding the shape.

Zone Shapes

Three protection zone shapes are available:

- Rectangular prism
- Sphere
- Cylinder

The dimensions of the zone (in cm) are configurable:

- For a Prism shape, this means height, width, and depth
- For a cylinder, this means the height and radius
- For a sphere, it means the radius

In addition, a thickness (in cm) can be defined around the outside of the protection zone where motion is allowed but velocity is limited.

The default is for the protection zone to be centered on the robot base, with its z-axis aligned with the base z-axis. However, this is also configurable. The center of the protection zone can be displaced in the x, y, and z directions. It can also be rotated around each of the base axes.

Limits

Figure 55: Limits tab

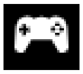
Under the Limits tab, limitations can be set for the protection zone. Limits between 0 and 10 cm/s can be set for:

- Envelope Translation Speed
- Envelope Orientation Speed

In addition, the thickness of the envelope can be configured between 0 and 1000 cm.

Controllers

This section describes the Controllers page of the *Web App.*

The Controllers page lets you view and toggle between the defined control mappings for any physical controllers associated with the robot.

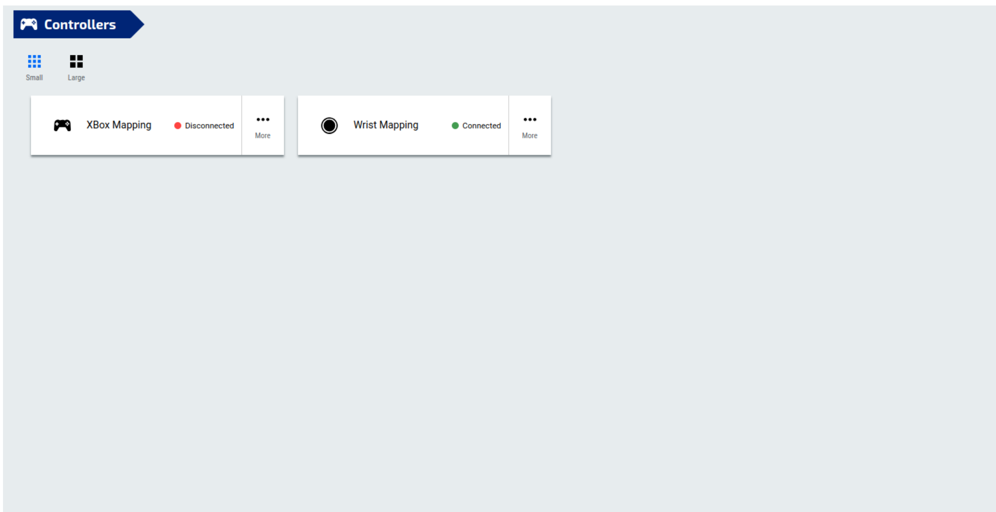


Figure 56: Controllers page

A mapping is a correspondence between the different controls on the controller and the resulting action produced in the robot.

There are two default controllers:

-  Xbox Mapping
-  Wrist Mapping

The **XBox Mapping** is for a generic Xbox gamepad. The robot has three preset control mappings defined for the Xbox gamepad. These mappings correspond to the mappings that can be toggled using the physical buttons on the gamepad.

- Xbox 360 Twist Linear
- Xbox 360 Twist Angular
- Xbox 360 Joint

The **Wrist Mapping** is for the two buttons on the wrist of the robot. The preset mapping corresponds to the factory default settings for the two wrist buttons - namely enabling Cartesian or Nullspace admittance mode.

The main information panel of the Controllers page has cards with control mapping information for each controller. The cards are in two sizes, large and small.

Hitting the edit button on a card brings up a window with tabs for the different control mapping modes available for the controller.

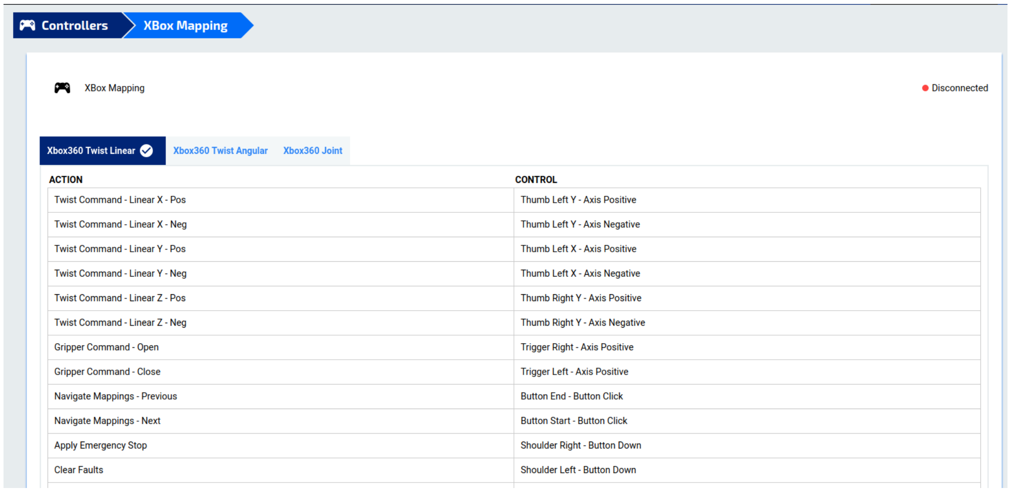


Figure 57: Xbox Mapping tab

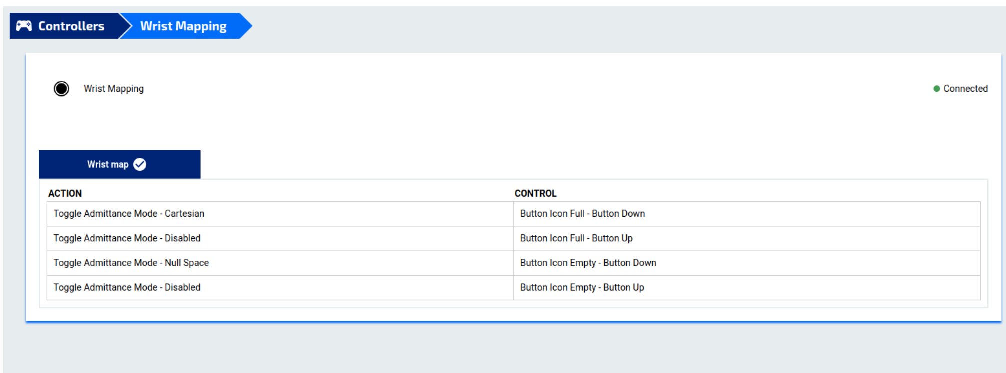
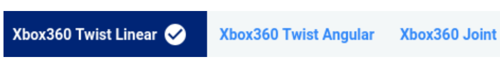


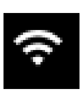
Figure 58: Wrist Mapping tab

At any given time, one map is set as active for the controller. The active controller map is indicated with a check mark icon. Another mapping can be set as active by clicking on the tab for the mapping.



Networks

This section describes the Networks page of the *Web App.*


The Networks page is used to set network parameters for:

- Ethernet
- Wi-Fi

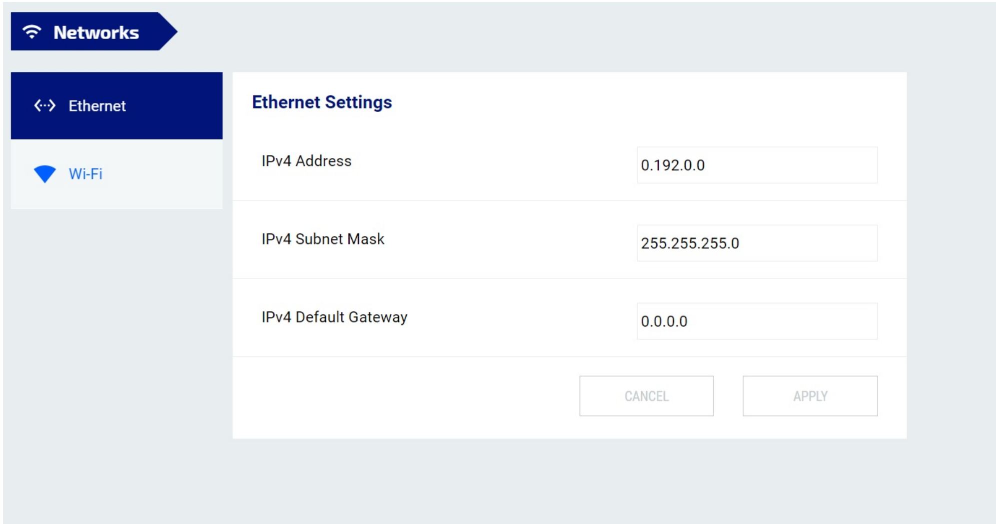


Figure 59: Networks page

The page has two tabs, one for each connection method.

The Ethernet Settings tab allows you to configure:

- IPv4 address
- IPv4 subnet mask
- IPv4 default Gateway

The Wi-Fi Settings tab allows you to enable Wi-Fi networking with the robot and find and connect to available Wi-Fi networks.

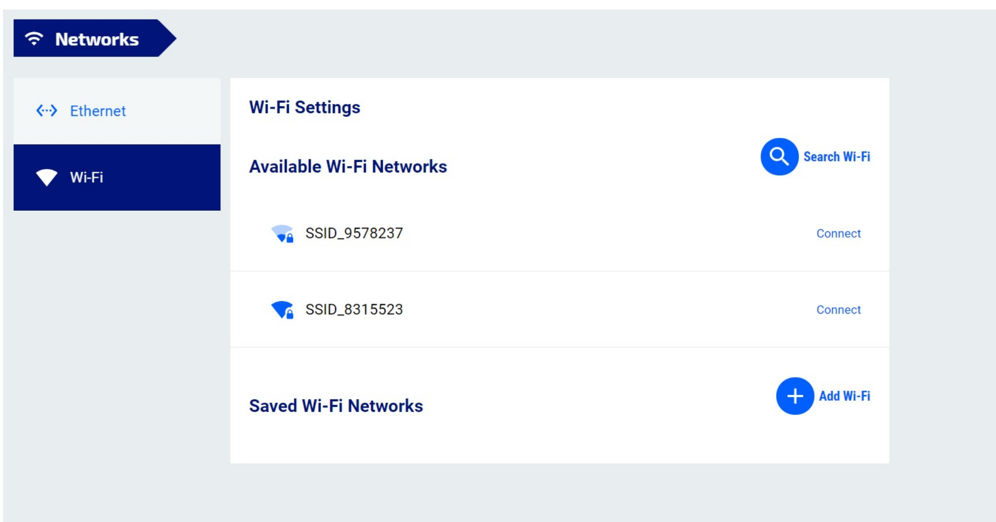


Figure 60: Wi-Fi Settings tab

Users

This section describes the Users page of the *Web App.*


The Users page is used to define, set, and edit user profiles for the robot.

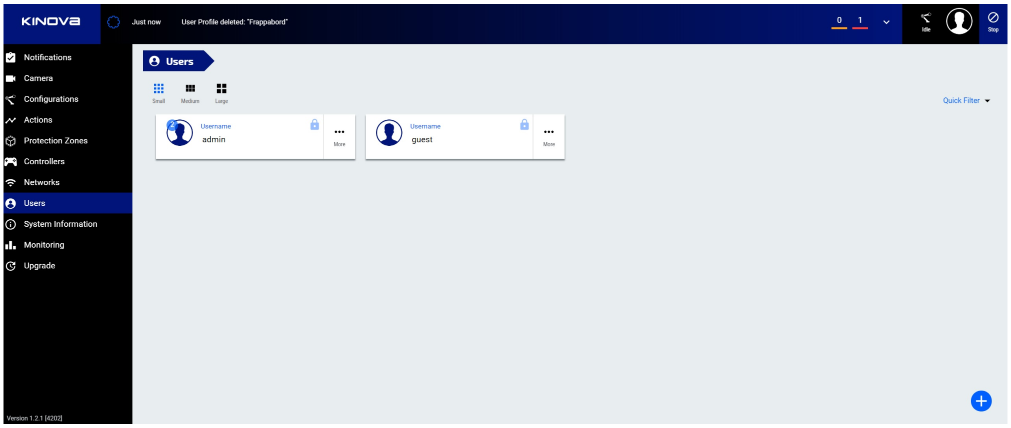


Figure 61: Users page

Defined profiles are displayed as information cards on the main information panel of the page. The cards are in three different sizes:

- large
- medium
- small

Card sizes can be toggled using buttons on the upper right of the main information panel.

Large cards show a full set of information. The large card displays the user name and language.

Medium cards are slightly smaller.

Small cards show a more compact view. By clicking the More button, a pop-up menu is revealed to allow you to View, Edit, Delete, or Duplicate the profile.

Clicking Edit brings you to an editing interface where it is possible to configure the profile.

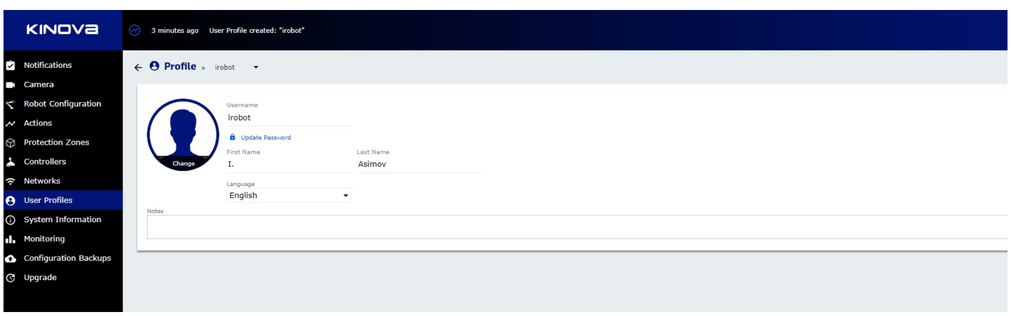


Figure 62: Card editing interface

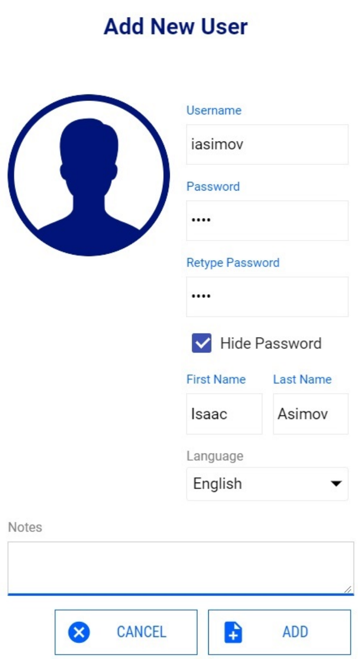
Creating a new user profile

This section describes how to create a new user profile in the *Web App.*

About this task

Procedure

1. On the Users page, press the  button to add a new empty user profile. This will bring up a window to enter information for the profile.



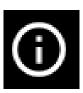
2. Enter the information for the user profile including name, user name, and password.
3. When you are done adding information, press ADD to create the new user profile.

Results

The new user profile will be created. The next time you log on to the *Web App*, you will be able to log in with these credentials.

System Information

This section describes the System Information page of the *Web App*.

 The System Information page gives a quick high level view of hardware and firmware configuration details.

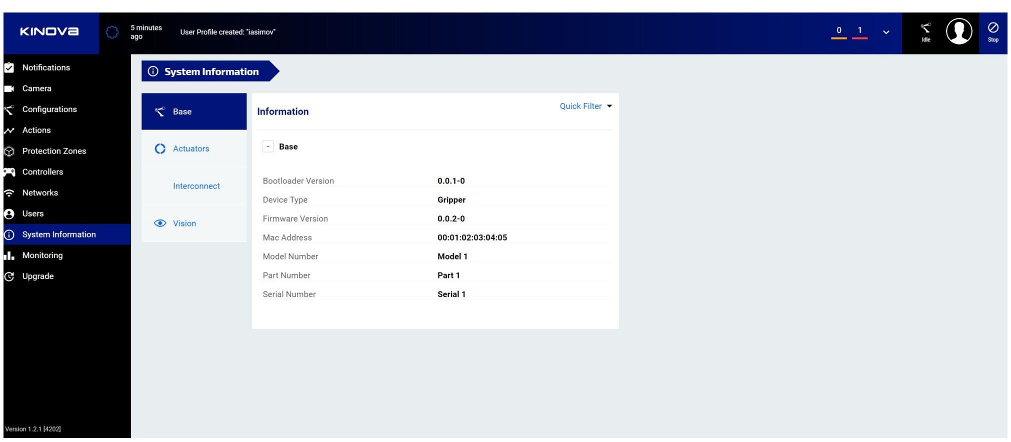


Figure 63: System Information page

The information on the page is displayed within different tabs:

- Base
- Actuators

- Interconnect (interface module)
- Vision

For all devices, information is given on:

- bootloader version
- device type
- firmware version
- MAC address
- model, part, and serial number

Monitoring

This section describes the Monitoring page of the *Web App.*

The Monitoring page allows for real-time monitoring of status and performance information for the robot. The monitoring page is the first page that opens when opening a new session using the *Web App*.

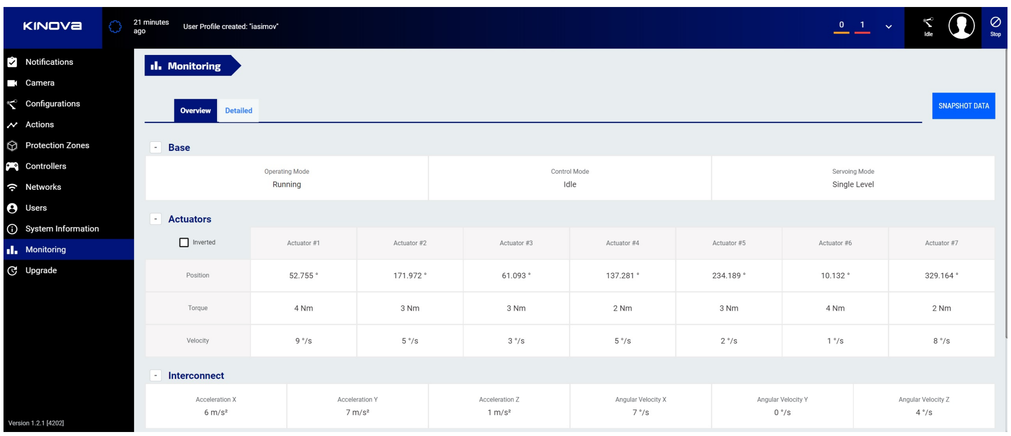


Figure 64: Monitoring page

The monitoring information is divided into six sections:

- Base
- Actuators
- Interconnect
- End effector

There are two tabbed views available, selectable through three tabs at the top of the screen:

- Overview
- Detailed

Overview tab contents

The Overview tab shows the following information in each section:

- Base
  - operating mode (maintenance, update, shutting down, run, in fault)
  - control mode (angular joystick, Cartesian joystick, torque control, Cartesian admittance, null space admittance)
  - servoing mode (single level (high level), low level)
- Actuators - for each joint:
  - measured position (°)

  - measured torque (N·m)
  - measured velocity (° / s)
- Interconnect
  - acceleration - x, y, and z (m / s^2)
  - angular velocity x, y, and z of interconnect (° / s)
- End effector
  - position and orientation - x, y, z, θ_x, θ_y, θ_z
  - velocity - x, y, z, θ_x, θ_y, θ_z
  - tool Twist - x, y, z, θ_x, θ_y, θ_z

Detailed tab contents

The detailed tab shows the following information in each section:

- Base
  - operating mode (maintenance, update, shutting down, run, in fault)
  - control mode (angular joystick, Cartesian joystick, torque control, Cartesian admittance, null space admittance)
  - servoing mode (single level (high level), low level)
  - arm voltage (V)
  - arm current (A)
  - CPU core temperature (°C)
  - ambient temperature (°C)
  - acceleration x, y, z of the base (m / s^2)
  - angular velocity x, y, z of base (° / s)
- Actuators - for each joint:
  - measured position (°)
  - measured velocity (° / s)
  - measured torque (N·m)
  - motor current (A)
  - voltage (V)
  - motor temperature (°C)
  - core temperature (°C)
- Interconnect
  - acceleration x, y, z of interface (m / s^2)
  - angular velocity x, y, z of interface (° / s)
  - voltage (V)
  - core temperature (°C)
- End effector
  - position and orientation - x, y, z, θ_x, θ_y, θ_z
  - tool Twist - x, y, z, θ_x, θ_y, θ_z

Exporting a snapshot of monitoring data

It is possible to export a snapshot of the currenting monitoring data for the robot.

By pressing the snapshot data button (SNAPSHOT DATA), you have the ability to save a dump of the monitoring data locally on your computer to JSON format. This can be useful information to share with Kinova support for troubleshooting purposes.

Upgrade

This section describes the Upgrade page of the *Web App.*

 The Upgrade page provides a simple interface to perform upgrades to the robot.

Robot upgrade files are bundled as a package (.swu file).

The robot upgrade package includes:

- robot devices firmware updates:
  - base controller
  - actuators
  - interface module
  - vision module
- *Web App* upgrade package
- KINOVA® KORTEX™ API upgrade package

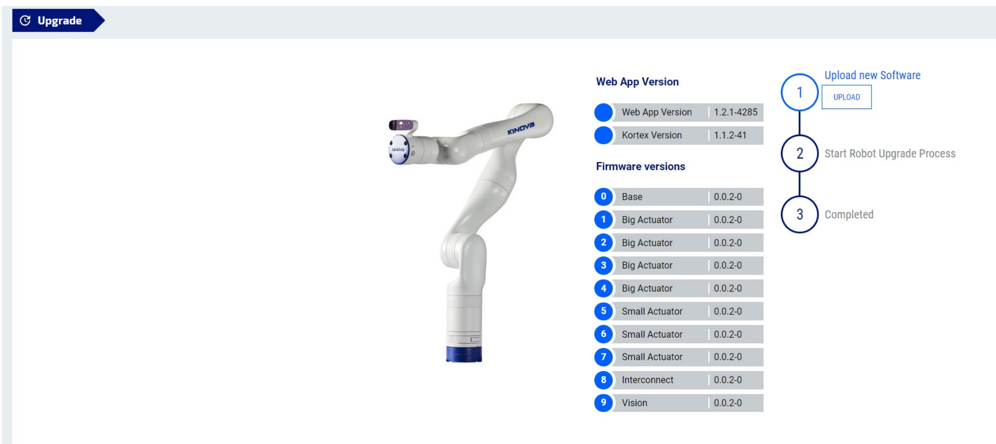


Figure 65: Upgrade page

The upgrade page provides an interface to upload a new upgrade package and initiate the upgrade.

The page also provides information on the current *Web App* and KINOVA® KORTEX™ API versions, as well as the current firmware versions of the robot devices.

Upgrading the robot firmware and software

This section describes the process to upgrade the robot firmware and software using the *Web App*.

Before you begin

- A new robot update package needs to have been previously downloaded to the development computer.
- The development computer needs to be connected to the robot, either via wired Ethernet connection or via Wi-Fi.
- The user needs to have a *Web App* session open on the robot.

About this task

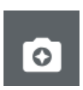
The *Web App* is used to upgrade the robot firmware and software using a new upgrade package on the development computer. The upgrade package covers all devices in the arm, and all devices are upgraded as part of this process.

Procedure

1. Browse to the *Web App* Upgrade page.
2. Click the Upload button on "Upload New Software."
3. Browse the development PC disk to select the new firmware package. The new package will upload to the robot. If the upload is unsuccessful, you will receive an error message. If it is successful, the process will continue.
4. The upgrade process will proceed automatically as soon as the firmware package uploads successfully. The *Web App* will indicate when the process is finished.

Snapshot tool

This section describes the snapshot tool.

 The snapshot tool lets users capture a snapshot of a current pose.

Pressing the snapshot button reveals a set of three snapshot pose options:

-  Cartesian pose
-  Joints pose
-  Gripper pose

Pressing one of the respective snapshot buttons will capture a snapshot of that type of pose the robot is currently in. The pose will be saved, and will show up as one of the saved Actions viewable in the Actions page.

KINOVA® KORTEX™ Developer Guide

Introduction

This section of the documentation provides guidance on developing custom software applications for the robot.

Your robot is enabled by KINOVA® KORTEX™, the new Kinova software framework and application development platform. This growing and evolving framework will allow you to configure and control the robot programmatically, adapting to your specific needs and supporting you in integrating new Kinova products into robotics applications. The KINOVA® KORTEX™ API is currently available only for the KINOVA® *Gen3 Ultra lightweight robot*, but will also support future robot products from Kinova as a cross-hardware development framework.

APIs are currently provided for the following languages:

- C++
- Python

Kinova also offers ROS packages covering most of the same functionalities.

The pages that follow describe the general philosophy and approach of the APIs.

The GitHub kinovarobotics/kortex and kinovarobotics/ros_kortex respositories contain additional developer guidance and resources, including detailed API documentation, setup instructions, and source code examples.

Devices and services

This section describes the concept of devices and services in the robot.

The API consists of **services** which define interfaces implemented and available on the various robot **devices**.

The robot consists of several devices:

- base controller
- actuators (each actuator is a distinct device)
- interface module
- vision module

A service consists of methods and communication exchange data structures. The devices in the robot each implement a particular set of services, some of which are available across multiple devices. The methods available as part of a service on a device are accessed via remote procedure calls (RPC).

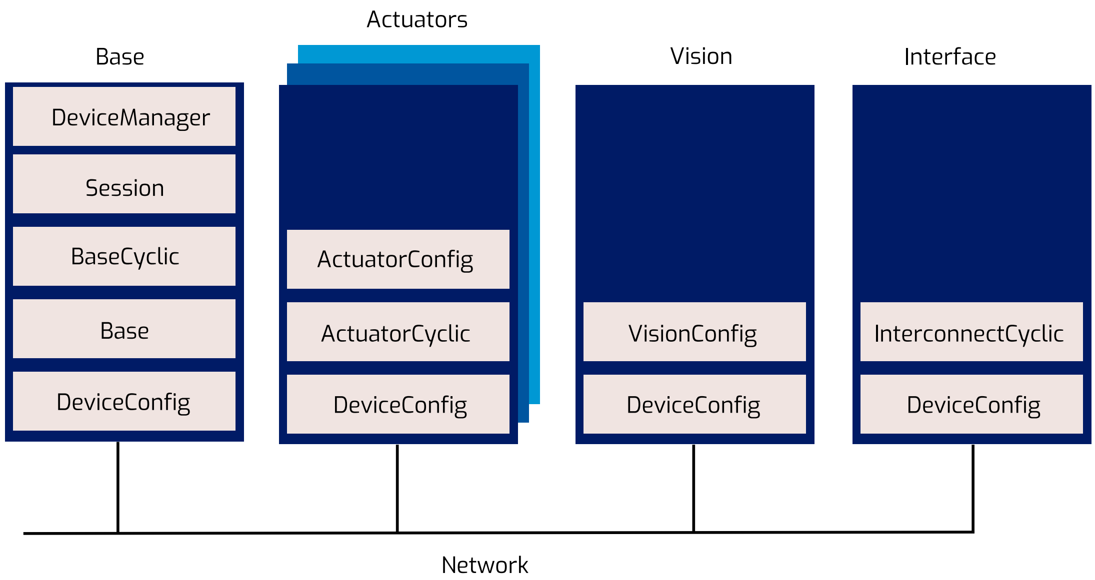


Figure 66: Services on multiple devices

Available services

This section lists the available robot services.

Kinova makes available a number of services for developers, each of which includes functions and data types supported for C++ and Python.

- `Session` - provides functions for opening and closing sessions with the robot. This service is used at the beginning and end of every session with the robot to authenticate the user. **Note:** In practice, end users will not use the Session service directly, but will use a SessionManager object. See the GitHub documentation for more details.
- `Base` - broadly useful service. Provides functions for configuring a range of base-related functionalities as well as high-level control for the robot.
- `DeviceManager` - provides a list of device information used for internal communication routing purposes.
- Cyclic data communications (sending commands to devices and/or receiving status feedback on a periodic or as-requested basis). Cyclic data communications are used with low-level servoing, and are intended to be called by API clients as part of a user-defined 1 kHz control loop.
  - For low-level servoing cyclic communication
    - `BaseCyclic`- sending commands to actuators and interface module, obtain feedback from base, actuators and interface module
- Configuration related
  - `ActuatorConfig` - get / set actuator configuration
  - `DeviceConfig` - get / set general device configuration
  - `VisionConfig` - get / set vision module configuration

For full details on available services, see the KINOVA® KORTEX™ GitHub repository.

Users, connections and sessions

This section describes the concept of connections and sessions in the API.

Introduction

A user has a **connection** with the robot when communication is established between the client application and the robot.

A **session** is active when the user has used the connection to log in to the robot with credentials. The default session credentials for the robot are:

- user: `admin`
- password: `admin`

A session is opened using a SessionManager object.

Sessions and robot control

Multiple users can connect to the same robot simultaneously, and have multiple sessions open on the robot.

A session must be created before commands can be received by the base (otherwise they will be discarded). Sessions are only supported for communications routed through the base. Sessions are **not** supported for communications between a client computer and a device.

Currently, high-level servoing for the robot only work in single-level servoing mode. Multi-level servoing is not currently supported. What this means is that multiple sessions can be active on the robot, and multiple users can pull data from the robot. However, only one session can actively control the robot at any given time. Rules are in place on the robot base to manage which session has control of the robot at any given time.

Services, methods, and messages

This section describes the concept of messages used by functions within services.

The API services offer a set of RPC and pub/sub methods. The methods exchange data which are structured as Google Protocol Buffer message objects.

KINOVA® KORTEX™ API and Google Protocol Buffer

This section describes the use of Google Protocol Buffer for the KINOVA® KORTEX™ API.

On Google Protocol Buffer

The KINOVA® KORTEX™ API is based on the Google Protocol Buffer 3 mechanism for serializing structured data. Using Protocol Buffer, the Kortex API is made available in C++ and Python languages.

Developers accustomed to Protocol Buffer can see .proto files on the KINOVA® KORTEX™ GitHub repository. These files are published as a means to document the services and methods offered via the API.

The API data structures are based on Google Protocol Buffer messages. Extensive documentation has been made available by Google explaining the different mechanisms offered to:

- set a field in a message
- read a OneOf element in a message
- go through a nested object

- set a nested object
- get/set a collection

For more details on how the above works, check out the following documentation on the Google Protocol Buffer website:

- C++ tutorial: https://developers.google.com/protocol-buffers/docs/reference/cpp-generated
- Python tutorial: https://developers.google.com/protocol-buffers/docs/reference/python-generated

Service client-server model

This section describes the client-server model for services.

Services operate on a **client-server** model. The **server** component of the service runs on the device itself. The **client** component runs on the client computer.

Services offer a set of device functionalities which are transparently exposed to the end-user via RPC and pub/sub methods.

The API is built on a transparent client/server communication protocol which allows an end-user (client side) to call methods on robot devices.

Notifications

This section describes the concept of notifications in the API.

The robot base can provide notifications on different topics as requested by a client application that has a session open with the robot.

The robot base uses a Publish/Subscribe design pattern. That is, rather than needing to poll periodically for updates, the client application subscribes to a list of Topics. Whenever a change happens related to that topic, whether caused by the same client session, or another, a publisher sends a notification to all subscribers. Notifications are surfaced to clients via the API, and are also displayed in the Notifications page of the Web app.

Client applications can also unsubcribe from a topic.

Methods for subscribing and unsubscribing from notification topics are descrcibed in the API documentation on the KINOVA® KORTEX™ GitHub respository.

Blocking and non-blocking calls

This section describes the concept of blocking vs. non-blocking calls in the context of RPCs on the robot.

The API defines interfaces of methods to be executed on devices in the robot.

The methods can be one of two types, depending on what the client application does while waiting for the response:

- blocking
- non-blocking

With a blocking call, the flow of the client application will pause and wait for the remote procedure call to return a response before proceeding. With non-blocking call, the procedure call is sent, and the flow of the application carries on while waiting for the response. When the response arrives, the caller will handle the response.

For the Python API, only blocking calls are enabled.

In the context of C++, remote procedure calls in the API can in general be set as either *blocking* or *non-blocking*.

There are two types of non-blocking calls available in the C++ API:

- Future / Promise
- Registered callback

For more information on how this works, see the API documentation on the KINOVA® KORTEX™ GitHub respository.

Robot servoing modes

This section describes the concept of servoing modes on the robot.

There are multiple servoing modes on the robot. A servoing mode is a modality through which commands are transmitted to robot devices during operation. Depending on the servoing mode chosen, the details involved in controlling via the API will be different.

There are two servoing modes:

- `High-level`
- `Low-level`

High-level servoing

This section describes the concept of high-level servoing with the robot.

High-level servoing is the default servoing mode for the robot on bootup.

In high-level servoing, users connect to the base through the API (whether directly, or through the *Web App* built on top of the API), sending command inputs. The base routes commands to the actuators, and manages a 1 kHz control loop.

High-level servoing is the recommended servoing mode for non-advanced users.

High-level servoing allows a client to control the robot by sending it a target (angular or Cartesian) position or velocity via an API method which is sent once (i.e. no high frequency client-controlled communication between the client PC and the robot). High level API calls are redirected to the robot control library to calculate inverse kinematics (breaking down the command into commands for actuators) and apply limits (protection zones, singularity management, self-collision avoidance).

The base then manages the execution of the command via the 1 kHz communications with the actuators.

Low-level servoing offers lighter and faster API methods, but at the cost of having to manage these details yourself.

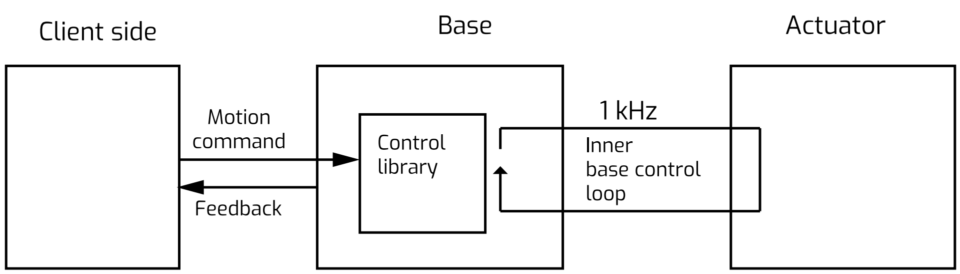


Figure 67: High-level servoing

High-level servoing can in theory be either single-level (one user controls the robot at a time in high-level servoing) or multi-level (multiple users simultaneously control the robot at the same time).

Note: Currently only single-level servoing is supported by the API.

Sessions and control permissions

As soon as someone takes control of the robot by sending a control command (whether from API calls, *Web App* session, or Xbox gamepad input) to the robot, the control mode changes from IDLE to SERVOING. In this mode, control commands from other sessions sent via the *Web App* or API methods will be blocked while the control mode is in SERVOING and this session has control. However, after a predefined "grace period" of 7.5 seconds elapses with no new control commands from the user, the robot control mode returns to IDLE and someone else can take control by sending control inputs via the *Web App* or API calls.

Override by physical controls

Physical controls of the robot via a connected Xbox gamepad or the buttons on the robot wrist override user session control of the robot via *Web App* or API calls. These physical controls always take precedence immediately, without having to wait for the grace period to elapse.

Low-level servoing

This section describes the concept of low-level servoing with the robot.

In low-level servoing, the API client connects to the base and sends commands through the base for routing.

The base ensures device routing and internal communications with the actuators at 1 kHz, but the high-level functionalities for the base control loop (robot kinematics, trajectory management, etc.) are no longer available.

Low-level servoing allows clients to control each actuator individually by sending position (or velocity) increments at 1 kHz frequency (bypassing the kinematic control library).

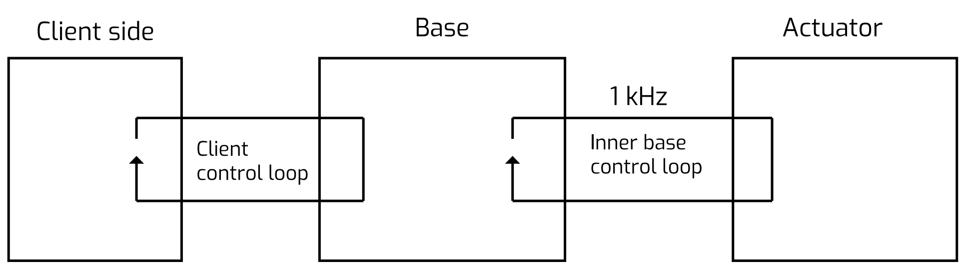


Figure 68: Low-level servoing

Device routing

This section describes device routing..

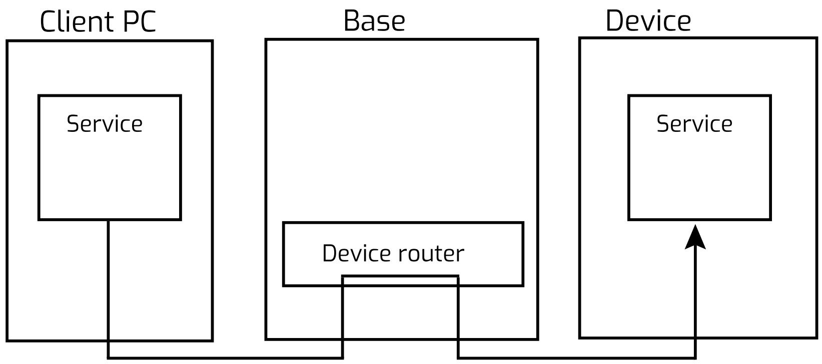


Figure 69 Device routing

The API allows you to communicate with the robot devices. Using a device identifier the RPC and pub/sub methods of the API are simply routed by robot base and directly bridged to the intended device.

Error management

This section describes the concept of error management with the robot.

When an API method is called, sometimes an error will result.

There are three main categories of errors:

- Protocol server errors
- Protocol client errors
- Device errors

The first two categories of errors include all errors relating to the the internal communication protocol. (ex: invalid, unsupported or unknown calls, out of session call, etc.)

The other category is for errors coming from the target device.

For each high level category, there are also more detailed and specific errors.

For more information about the error codes that can be produced, see the KINOVA® KORTEX™ GitHub documentation.

KINOVA® KORTEX™ GitHub repository

This section describes the KINOVA® KORTEX™ GitHub repository.

For more detailed information about developing applications using the API visit the KINOVA® KORTEX™ GitHub repository at: github.com/kinovarobotics/kortex

The repository offers access to a number of resources for developers.

- setup instructions and release notes
- detailed API documentation by language
- code examples

KINOVA® KORTEX™ ROS and KINOVA® KORTEX™ ROS GitHub overview

This section describes the ROS packages for the robot (and all other products enabled by KINOVA® KORTEX™).

Introduction

KINOVA® KORTEX™ ROS is the official repository containing ROS packages to interact with Kortex and related products. It consists of a number of ROS packages built on top of the client Kortex API.

These ROS packages are designed to work with ROS Kinectic Kame. Note that ROS Kinectic Kame is primarily targeted for Ubuntu 16.04 (Xenial) LTS (although other platforms are also supported to different degrees).

Methods provided by the underlying API are offered as ROS services and topics, depending on the method.

- RPC methods are exposed via ROS services
- pub/sub methods are exposed via ROS topics

The ROS Messages correspond to the message type definitions of the underlying API.

The ROS interface can be accessed using either Python (rospy) or C++ (roscpp).

Detailed documentation of the packages is available on the Kinova `ros_kortex` GitHub repository at github.com/kinovarobotics/ros_kortex

The repository includes various packages related to ROS development:

- setup instruction and release notes
- `kortex_actuator_driver` (ROS node package for interfacing with a single actuator)
- `kortex_api` (package containing header files and libraries needed to use the C++ Kortex API)
- `kortex_description` (package contains URDF and STL files of the robot)
- `kortex_device_manager` (ROS node package to allow basic communication with every device supported by the Kortex framework)
- `kortex_driver` (ROS node package to allow direct communication with a robot base)
- `kortex_examples` (examples needed to understand the basics of ros_kortex)
- `kortex_vision_config_driver` (ROS node package to allow direct communication with the robot Vision module)

Working with camera streams using GStreamer

This section describes how to work with vision module camera streams using the GStreamer framework.

GStreamer

Kinova recommends that developers use the GStreamer framework for handling the camera's sensor streams. GStreamer offers a pipeline-based framework that allows you to link together plugins for image-processing workflows.

Requirements:

- GStreamer version: 1.8.3 and above
- Supported operating systems:
  - Windows 7 and later
  - Ubuntu 16.04 and later

Using GStreamer

GStreamer pipelines can be called from the command line using the gst-launch-1.0 utility.

For integration with applications, GStreamer offers a number of language bindings, which include both C++ and Python.

Official documentation for the GStreamer framework, including installation instructions, application development guidance, and tutorials can be accessed here: https://gstreamer.freedesktop.org/documentation/

Windows command examples

This section provides examples of using GStreamer with the robot vision module camera streams on the Windows command line.

Color stream CLI example: command to display the color stream

```
gst-launch-1.0.exe rtspsrc location=rtsp://192.168.1.10/color
latency=30 ! rtph264depay ! avdec_h264 ! autovideosink
```

Unpacking the example:

- `gst-launch-1.0.exe`: launch GStreamer
- `rtspsrc location=rtsp://192.168.1.10/color latency=30`: connect to the RTSP server at the color stream URL with latency of 30 ms.
- `rtph264depay`: Extract H.264 video payload from RTP packets
- `avdec_h264`: decode H.264 video
- `autovideosink`: search computer registry for video sink (player) and plays decoded video stream.

Depth stream CLI example: command to display the depth stream

```
gst-launch-1.0.exe rtspsrc location=rtsp://192.168.1.10/depth
latency=30 ! rtpgstdepay ! videoconvert ! autovideosink
```

Unpacking the example:

- `gst-launch-1.0.exe`: launch GStreamer
- `rtspsrc location=rtsp://192.168.1.10/depth latency=30`: connect to the RTSP server at the color stream URL with latency of 30 ms.
- `rtpgstdepay`: extract GStreamer buffers from RTP packets
- `videoconvert`: automatically convert the video to a format understandable to the chosen video sink in the next step
- `autovideosink`: search computer registry for video sink (player) and plays decoded video stream.

Linux command examples

This section provides examples of using GStreamer with the robot vision module camera streams on the Linux command line.

Color stream CLI example: command to display the color stream

```
gst-launch-1.0 rtspsrc location=rtsp://192.168.1.10/color
latency=30 ! rtph264depay ! avdec_h264 ! autovideosink
```

Unpacking the example:

- `gst-launch-1.0`: launch GStreamer
- `rtspsrc location=rtsp://192.168.1.10/color latency=30`: connect to the RTSP server at the color stream URL with latency of 30 ms.
- `rtph264depay`: extract H.264 video payload from RTP packets
- `avdec_h264`: decode H.264 video
- `autovideosink`: search computer registry for video sink (player) and play decoded video stream.

Depth stream CLI example: command to display the depth stream

```
gst-launch-1.0 rtspsrc location=rtsp://192.168.1.10/depth
latency=30 ! rtpgstdepay ! videoconvert ! autovideosink
```

Unpacking the example:

- `gst-launch-1.0`: launch GStreamer
- `rtspsrc location=rtsp://192.168.1.10/depth latency=30`: connect to the RTSP server at the color stream URL with latency of 30 ms.
- `rtpgstdepay`: extract GStreamer buffers from RTP packets
- `videoconvert`: automatically convert the video to a format understandable to the chosen video sink in the next step
- `autovideosink`: search computer registry for video sink (player) and play decoded video stream.

Guidance for advanced users

This section gathers together reference information on advanced topics.

Introduction

The following contents are intended for advanced users.

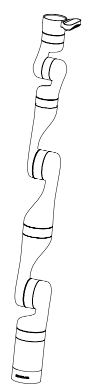
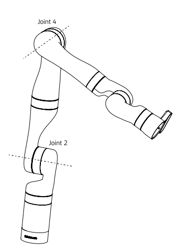
7 DoF singularity configurations

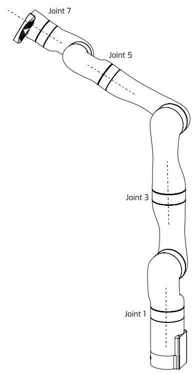
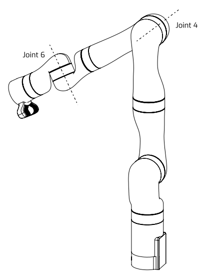
This section describes the singularity configurations of the 7 DoF robot.

Singularity configurations overview

Singularities generally occur when a particular angular configuration of the robot causes axes to be aligned, causing the robot to lose degrees of freedom and experience limitations in movement in some directions while operating the robot in Cartesian mode. There are many ways that this could potentially happen, and an exhaustive listing would be difficult. The following table highlights some important singularities for the 7 DoF robot, explaining how they occur and how the robot behavior is altered near the singularity while in Cartesian mode.

Table 37: Selected singularity configurations description

Singularity	Description	Robot behavior
Boundary singularity 	The arm is at full reach. Joint 4 (elbow) is at 0°. The arm cannot move any farther in the direction it is currently reaching out.	Due to singularity avoidance, it's not possible to bring the elbow to 0° in Cartesian mode.
Joints 2 and 3 singularity 	Joint 2 is at 0° so joints 1 and 3 are perfectly aligned and have the same effect. Joint 3 is at 90° or at 270° so that the axes of joint 2 and joint 4 are perpendicular. The robot can no longer move purely along an axis in translation.	Due to singularity avoidance, it's not possible to bring joint 3 near 90° or 270° when joint 2 is near 0° (or vice versa, to bring joint 2 near 0° when joint 3 is near 90° or 270°) in Cartesian mode. The control algorithms will try to avoid the singularity by moving joint 2 away from 0° and joint 3 away from 90° or 270° while moving in null space.

Singularity	Description	Robot behavior
Joints 2 and 6 singularity 	Joint 2 is at 0° so that joints 1 and 3 are perfectly aligned and have the same effect. Joint 6 is at 0° so that joints 5 and 7 are perfectly aligned and have the same effect. The hand cannot rotate in one direction anymore.	Due to singularity avoidance, it's not possible to bring joint 2 near 0° when joint 6 is near 0° in Cartesian mode. The control algorithms will try to avoid the singularity by moving joints 2 and 6 away from 0° while moving in null space.
Joints 5 and 6 singularity 	Joint 6 is at 0° so that joints 5 and 7 are perfectly aligned and have the same effect. Joint 5 is at 90° or at 270° so that the axes of joint 4 and joint 6's axis are perpendicular. The robot can no longer complete pure rotations around an axis.	Due to singularity avoidance, it's not possible to bring joint 5 near 90° or 270° when joint 6 is near 0° in Cartesian mode. The control algorithms will try to avoid the singularity by moving joint 5 away from 90° or 270° and joint 6 away from 0° while moving in the robot's null space.

Reference frames and transformations

Homogeneous transforms

This section describes the homogeneous transforms for the robotic arm.

Introduction

The forward kinematics of the robotic arm are determined by homogeneous transform matrices. These matrices represent the transformations from one frame (base, joint, or end effector) to the next along the kinematic chain.

The overall transformation from the base frame to the end effector frame is given by:

$${}^{B}T^{*}_{EE} = {}^{B}T^{*}_{1}\,{}^{1}T^{*}_{2}\,{}^{2}T^{*}_{3}\,{}^{3}T^{*}_{4}\,{}^{4}T^{*}_{5}\,{}^{5}T^{*}_{6}\,{}^{6}T^{*}_{7}\,{}^{7}T^{*}_{EE}$$

Where:

$$ {}^{i-1}T^{*}_{i} = {}^{i-1}T_{i} * R_z(q_i) $$

${}^{i-1}T_i$ = Transform from frame i-1 to frame i for qi = 0

$$ R_z(q_i) = \begin{pmatrix} cqi & -sqi & 0 & 0 \\ sqi & cqi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

q_i = joint angle for joint i

$cq_i = \cos(q_i)$ $sq_i = \sin(q_i)$

${}^{i-1}T_i$ is the transform from the previous frame [i-1] to the current frame [i] where q_i, the angle for joint i, is 0.

$R_Z(q_i)$ is the transformation matrix for a rotation of q_i around joint i (the z axis for the joint frame is always defined to be along the joint axis of rotation.)

${}^{i-1}T^{*}_{i}$ is the matrix for the general transformation matrix from frame [i-1] to frame [i].

Homogeneous transform matrices - 7 DoF spherical wrist

This section is a reference for the homogeneous transform matrices for the 7 DoF spherical wrist robot.

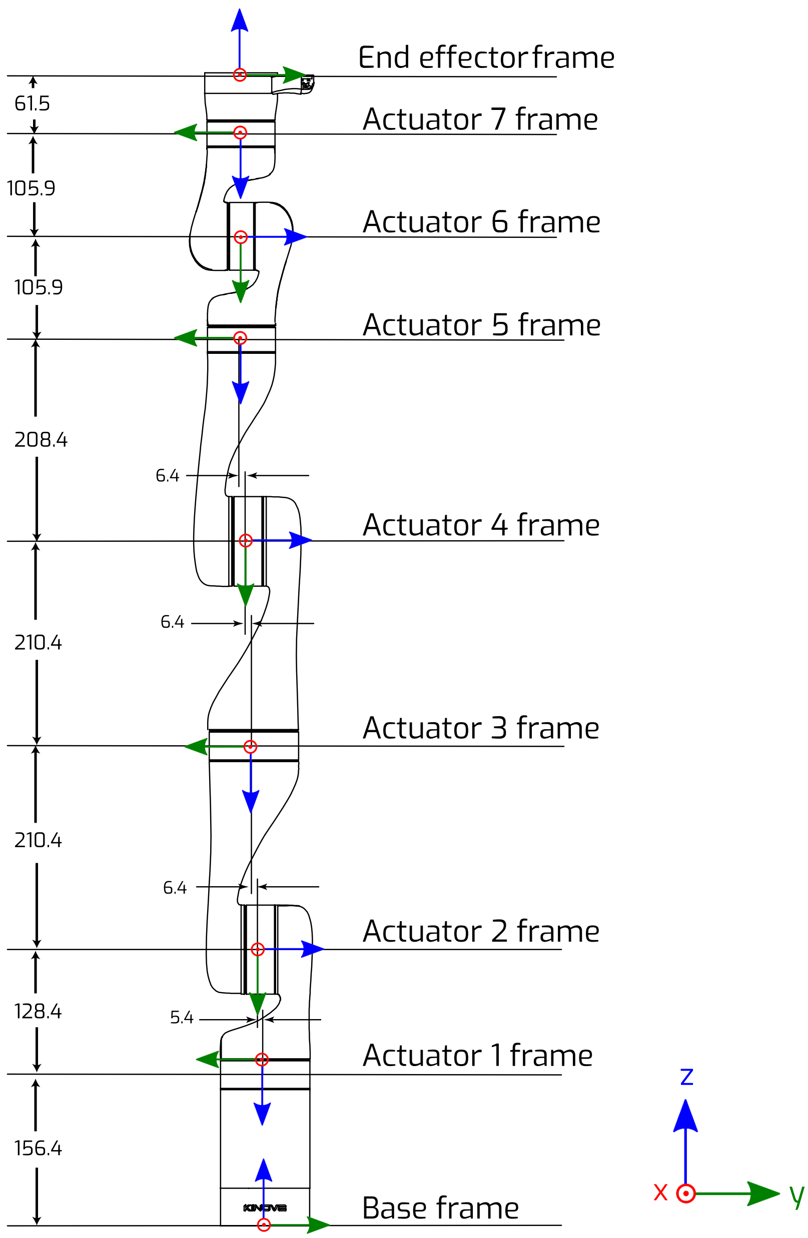


Figure 70: Frame definitions and dimensions (all joints at 0 position, dimensions in mm)

Table 38: Transformation matrices

Transformation	${}^{i-1}T_i$	${}^{i-1}T^*_i$
Base to frame 1	${}^{B}T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.1564 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{B}T^*_1 = \begin{bmatrix} cq1 & -sq1 & 0 & 0 \\ -sq1 & -cq1 & 0 & 0 \\ 0 & 0 & -1 & 0.1564 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 1 to frame 2	${}^{1}T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0.0054 \\ 0 & 1 & 0 & -0.1284 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{1}T^*_2 = \begin{bmatrix} cq2 & -sq2 & 0 & 0 \\ 0 & 0 & -1 & 0.0054 \\ sq2 & cq2 & 0 & -0.1284 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 2 to frame 3	${}^{2}T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -0.2104 \\ 0 & -1 & 0 & -0.0064 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{2}T^*_3 = \begin{bmatrix} cq3 & -sq3 & 0 & 0 \\ 0 & 0 & 1 & -0.2104 \\ -sq3 & -cq3 & 0 & -0.0064 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 3 to frame 4	${}^{3}T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.0064 \\ 0 & 1 & 0 & -0.2104 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{3}T^*_4 = \begin{bmatrix} cq4 & -sq4 & 0 & 0 \\ 0 & 0 & -1 & -0.0064 \\ sq4 & cq4 & 0 & -0.2104 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 4 to frame 5	${}^{4}T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -0.2084 \\ 0 & -1 & 0 & -0.0064 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{4}T^*_5 = \begin{bmatrix} cq5 & -sq5 & 0 & 0 \\ 0 & 0 & 1 & -0.2084 \\ -sq5 & -cq5 & 0 & -0.0064 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 5 to frame 6	${}^{5}T_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -0.1059 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{5}T^*_6 = \begin{bmatrix} cq6 & -sq6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ sq6 & cq6 & 0 & -0.1059 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 6 to frame 7	${}^{6}T_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -0.1059 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	${}^{6}T^*_7 = \begin{bmatrix} cq7 & -sq7 & 0 & 0 \\ 0 & 0 & 1 & -0.1059 \\ -sq7 & -cq7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 7 to end effector	${}^{7}T_{EE} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -0.0615 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

Note: units are in meters for homogeneous transform translations in the right-hand column of each matrix.

Standard robot frames

This section describes the standard frames of the robotic arm.

The robot has three standard frames:

- base frame (base reference frame)
- rotating frame (actuator 1 reference frame)
- tool frame (end-effector reference frame)

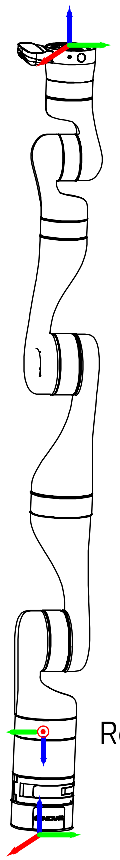


Figure 71: Three standard frames

Different control modes make use of different frames.

Dynamic parameters of the 7 DoF robot

This section describes the dynamic parameters of the 7 DoF robot.

Overview

The following tables describe the key dynamic parameters of the link segments of the 7 DoF robot, including masses, centers of masses, and and moment tensors (moments of inertia). The center of mass of a link is always expressed in the coordinates of the precedent joint frame. The mass of a link segment includes the shell and portions of the input and output actuators (as applicable) that are enclosed within the link and move rigidly with the link.

Overview	
moments of inertia taken at the center of mass and aligned with the output coordinate system	$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$

Table 39: Base

Physical quantity	Value	
mass (kg)	1.697353	
center of mass coordinates (m)	(0.000648, -0.000166, 0.084487)	
moments of inertia ($kg \cdot m^2$)	$\begin{bmatrix} 0.004622 & 0.000009 & 0.000060 \\ 0.000009 & 0.004495 & 0.000009 \\ 0.000060 & 0.000009 & 0.002079 \end{bmatrix}$	

Table 40: Link 1

Physical quantity	Value	
mass (kg)	1.377353	
center of mass coordinates (m)	(-0.000023, -0.010364, -0.073360)	
moments of inertia ($kg \cdot m^2$)	$\begin{bmatrix} 0.004570 & 0.000001 & 0.000002 \\ 0.000001 & 0.004831 & 0.000448 \\ 0.000002 & 0.000448 & 0.001409 \end{bmatrix}$	

Table 41: Link 2

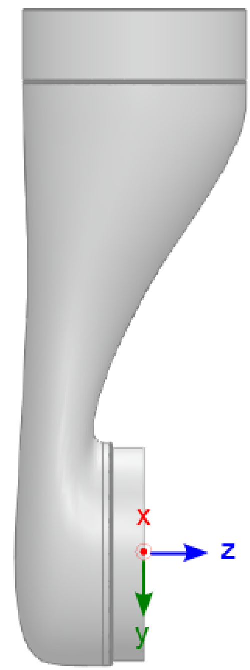
Physical quantity	Value	
mass (kg)	1.163667	
center of mass coordinates (m)	(-0.000044, -0.099580, -0.013278)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.011088 & 0.000005 & 0.000000 \\ 0.000005 & 0.001072 & -0.000691 \\ 0.000000 & -0.000691 & 0.011255 \end{bmatrix}$	

Table 42: Link 3

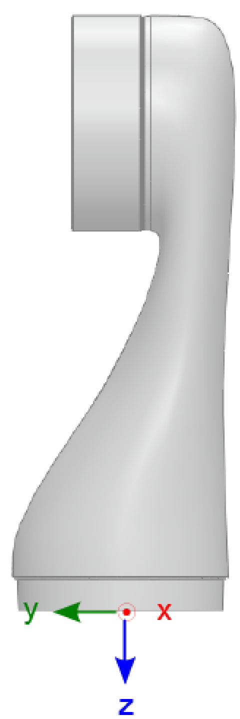
Physical quantity	Value	
mass (kg)	1.163667 kg	
center of mass coordinates (m)	(-0.000044, -0.006641, -0.117892)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.010932 & 0.000000 & -0.000007 \\ 0.000000 & 0.011127 & 0.000606 \\ -0.000007 & 0.000606 & 0.001043 \end{bmatrix}$	

Table 43: Link 4

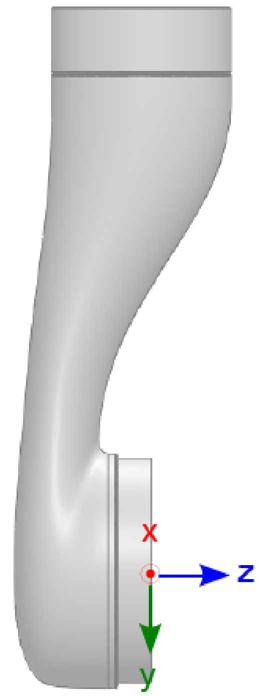
Physical quantity	Value	
mass (kg)	0.930287	
center of mass coordinates (m)	(-0.000018, -0.075478, -0.015006)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.008147 & -0.000001 & 0.000000 \\ -0.000001 & 0.000631 & -0.000500 \\ 0.000000 & -0.000500 & 0.008316 \end{bmatrix}$	

Table 44: Link 5

Physical quantity	Value	
mass (kg)	0.678106	
center of mass coordinates (m)	(0.000001, -0.009432, -0.063883)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.001596 & 0.000000 & 0.000000 \\ 0.000000 & 0.001607 & 0.000256 \\ 0.000000 & 0.000256 & 0.000399 \end{bmatrix}$	

Table 45: Link 6

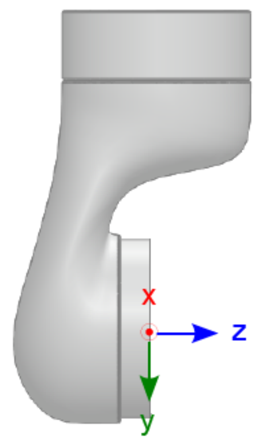
Physical quantity	Value	
mass (kg)	0.678106	
center of mass coordinates (m)	(0.000001, -0.045483, -0.009650)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.001641 & 0.000000 & 0.000000 \\ 0.000000 & 0.000410 & -0.000278 \\ 0.000000 & -0.000278 & 0.001641 \end{bmatrix}$	

Table 46: Interface Module

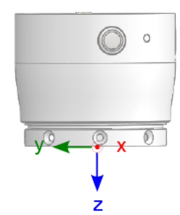
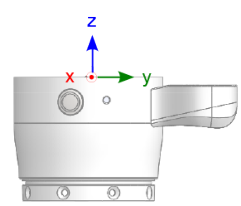
Physical quantity	Value	
mass (kg)	0.364223	
center of mass coordinates (m)	(-0.000093, 0.000132, -0.024666)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.000206 & 0.000000 & 0.000001 \\ 0.000000 & 0.000215 & -0.000002 \\ 0.000001 & -0.000002 & 0.000240 \end{bmatrix}$	

Table 47: Interface & Vision Module

Physical quantity	Value	
mass (kg)	0.500657	
center of mass coordinates (m)	(-0.000281, -0.011402, -0.031080)	
moments of inertia $(kg \cdot m^2)$	$\begin{bmatrix} 0.000570 & 0.000002 & 0.000003 \\ 0.000002 & 0.000352 & 0.000111 \\ 0.000003 & 0.000111 & 0.000609 \end{bmatrix}$	

Maintenance

This section describes maintenance tasks for the robot.

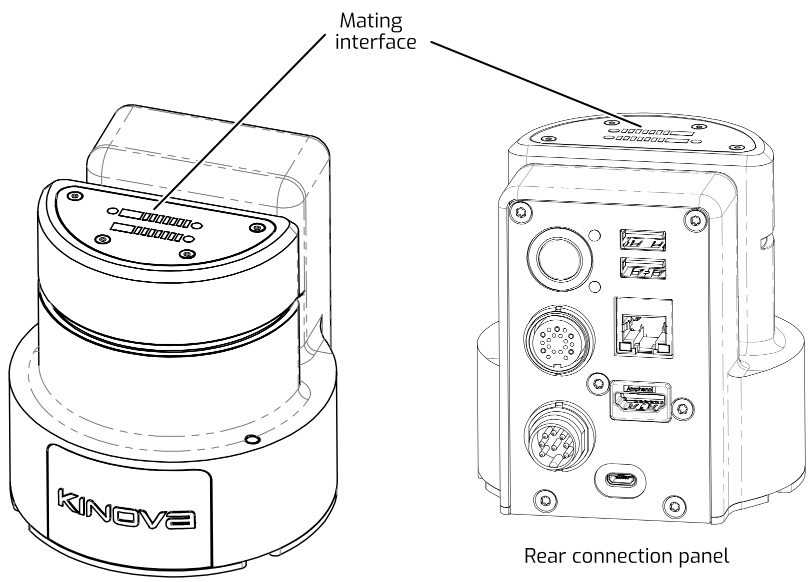
Maintenance overview

Currently, none of the components of the robot are field replaceable. Contact Kinova for assistance in the case of any component breakdown or malfunction.

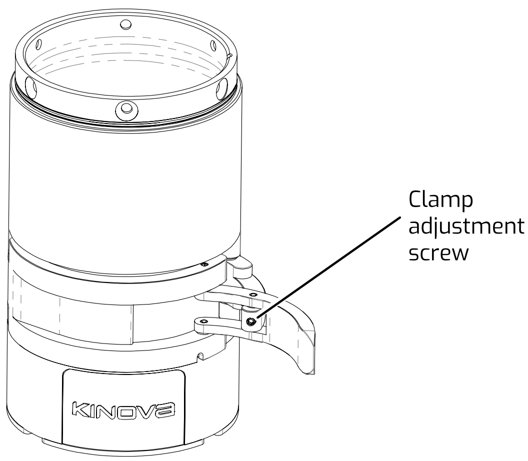
Preventive Maintenance

Some preventive maintenance tasks are helpful for protecting your robot and getting the most out it over time:

- **Cleaning contacts on base controller -** keep contacts clear of dust and contamination, wiping electrical contacts regularly with a soft moistened cloth.



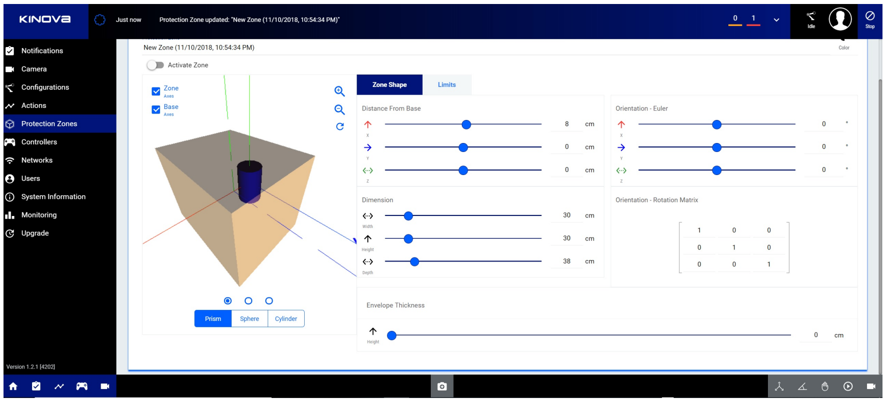
- **Fine adjustment of base clamp -** Some adjustment may be needed for the base clamp to ensure that the robot is firmly clamped onto the base controller. Within the clamp is an adjustment screw which can be adjusted to tighten the clamp as needed.



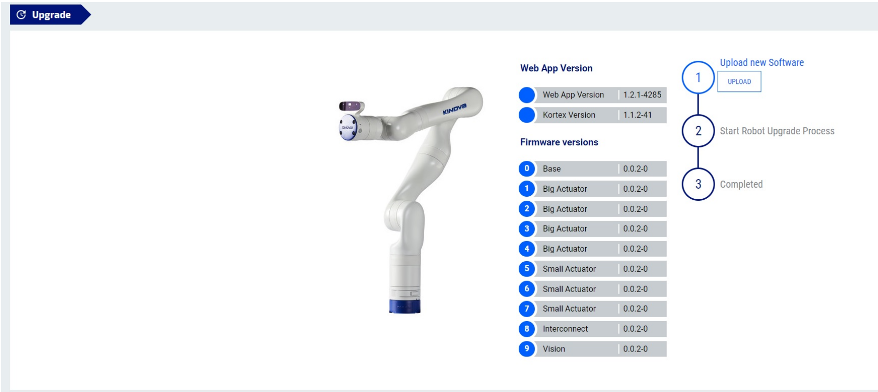
To tighten the clamp turn the screw clockwise using a 2 mm hex key in small, ¼ turn increments, testing the clamp after each increment.

Note: Overtightening the clamp can damage the clamp and base. Always make sure that the clamp can be closed using a reasonable amount of force.

- **Cleaning glass on vision module -** the cameras on the vision module are covered in glass. For best results, keep the glass clear of contamination that could block the view of the sensors. Wipe the glass regularly with a soft moistened cloth and wipe dry with a soft dry cloth.
- **Setting protection zones -** volumetric protection zones should be established around the robot to protect it from potential damage caused by collisions with known obstacles. Protection zones can be set using the KINOVA® KORTEX™ *Web App*.



- **Updating firmware -** Kinova will periodically release updates to robot and robot device firmware to fix known bugs and expand the capabilities of the robot. For best results, it is recommended to regularly update firmware using the *Web App*.



- **Updating development packages -** Kinova will periodically release updates for the KINOVA® KORTEX™ API and KINOVA® KORTEX™ ROS packages on the Kinovarobotics/kortex and Kinovarobotics/ros_kortex GitHub repositories. These updates will fix known bugs and expand the capabilities of the robot.

Troubleshooting

This section describes troubleshooting for the robot

Troubleshooting resources

There are several resources that can be used to help diagnose issues when they occur:

- KINOVA® KORTEX™ *Web App* notifications
- *Web App* monitoring - the monitoring page provides useful status information on the robot components, including the base, all actuators, and the interface. Notably, currents, voltages, CPU core temperatures and motor temperatures from the sensors are updated in real-time on the monitoring page
- *Web App* safeties page - when a safety item's warning or error threshold is exceeded, the safety item will be highlighted in the Robot Configurations Safety page.
- Base controller LED indicators - LEDs on the robot base controller connector panel provide visual feedback on the robot status
- API errors
- GitHub - information on known issues and workarounds

General tips for troubleshooting issues with the robot

When the robot enters a fault state, the robot will become unresponsive until the fault is cleared. The gamepad can be used to clear faults - press the left bumper once and proceed.

Open the *Web App* and check the monitoring page for high-level status information on various components.

Check the *Web App* Notifications page for any recent notifications.

Check the *Web App* Safeties page to see if the robot has passed a warning or error threshold. If any safety us triggered, the safety item will be Look up the information on the safety for guidance on handling.

Remember that the behavior of the robot will change as the robot nears singularities or enters the envelope of protection zones. If robot behavior deviates from what you expect, verify whether one of these two cases applies.

For API-related errors, check the reference tables for guidance on the source of the error and how to deal with it.

Kinova recommends updating robot firmware and KINOVA® KORTEX™ API packages regularly to keep up with the latest bug fixes and ensure optimal performance.

If all else fails, try rebooting the robot.

If you're still experiencing issues, contact Kinova support via the website.

Base controller LEDs

This section describes the meanings of the LED indicators on the robot base controller.

Overview

The base controller has two LEDs, one blue and one red / green.

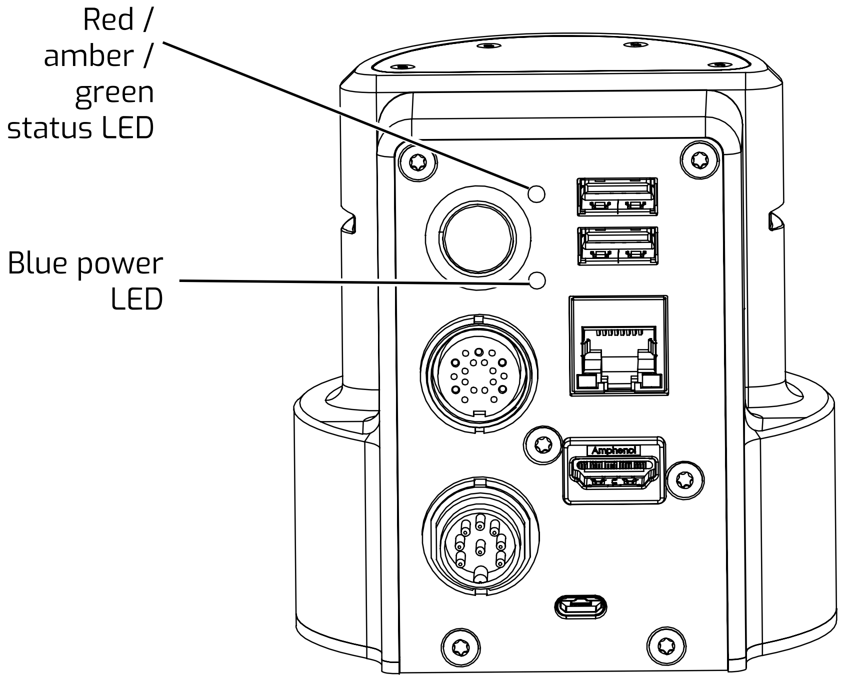


Figure 72: Base controller LEDs

Base controller LED details

Table 48: LED interpretation

Power LED		Status LED		Description
color	status	color	status	
n/a	off	n/a	off	system not powered
blue	blinking	n/a	off	system booting
blue	solid	amber	solid	system initializing
n/a	off	green	solid	system operating normally
blue	blinking	amber	solid	system currently updating at least one component
blue	solid	red	solid	system in error state

How to respond to safety warnings and errors

This section describes how to respond to safety warnings and error states experiened when operating the robot.

Overview

The robot has a number of warning and error thresholds set for safety purposes. These are viewable (and in some case configurable) in the Web application. The following tables give more guidance as to the source of the problem when a safety threshold is triggered.

Safeties handling details

Table 49: Base safeties handling

Safety	Most Probable Cause
Incompatible Firmware version	• Firmware issue
Firmware Update Failure	• Firmware issue • Communication issue
Maximum Ambient Temperature	• CPU heat sink issue • Unknown thermal issue
Maximum Core Temperature	• CPU heat sink issue • Unknown thermal issue
Joint Fault	• Joint error / warning state
Joint Detection Error	• Communication issue
Network Initialization Error	• Base CPU board issue
Maximum Current	• Shorted phases on a joint • Payload exceeded
Maximum Voltage	• Power supply issue • Electronic component failure
Minimum Voltage	• Power supply issue • Electronic component failure
Emergency Stop Activated	• XBox gamepad emergency stop button clicked • Web application emergency stop button clicked
Emergency Line Asserted	• Joint not programmed • Joint in a boot loop • Electrical component failure
Inrush Current Limiter Fault	• Payload exceeded • Electrical component failure

Table 50: Actuators safeties handling

Safety	Most Probable Cause
Following error	• Communication issue • Firmware issue
Maximum velocity	• Communication issue • Firmware issue

Safety	Most Probable Cause
Maximum torque	• Strain gauge improperly soldered • Incorrect torque calibration
Magnetic position	• Magnet improperly glued
Hall position	• Hall sensor major malfunction
Hall sequence	• Hall sensor major malfunction(s)
Input encoder Hall mismatch	• Dirt and/or particles on encoder disk
Input encoder index mismatch	• Dirt and/or particles on encoder disk
Input encoder magnetic mismatch	• Dirt and/or particles on encoder disk • Detached magnet on magnetic encoder
Maximum motor current	• Shorted phases • Bad motor
Non-volatile memory corrupted	• Incomplete calibration(s • No system information entered • No torque calibration
Motor driver fault	• Shorted phases • Hall sensor issue
Watchdog triggered	• Firmware issue

Contacting Kinova support

Here's where to turn for related support and advice.

For support and advice on hardware related issues, please don't hesitate to contact us through the support form on our website:

www.kinovarobotics.com/support.

For development guidance and software-related questions, check out the KINOVA® KORTEX™ and KINOVA® KORTEX™ ROS GitHub repositories at:

github.com/kinovarobotics/kortex
github.com/kinovarobotics/ros_kortex

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
Friedrich-Ebert-Allee 13
53113 Bonn, Deutschland
+49 (0) 228-9293-9148

Kinova Asia
4th Floor, Building 5, Zizhu Cyberport
No. 555 Dongchuan Road, Minhang District
200241 Shanghai P.R.China
+86 1862 1111 822

UG-014_2018-12-R01

KINOVA
Achieve Extraordinary