User Guide

KINOVA® *Gen3 lite robot*

KINOVA
Achieve Extraordinary

Contents

Introduction 6
Welcome 6
About this document 6
Normal use definition 6
Risk assessment 7
Safety directives and warnings 7
Warranty 9
Disclaimer 10
Acronyms and abbreviations 11

Robot components 13
Overview 13
Base 13
Base connector panel 14
Base joystick / expansion connector pinout 14
Base LED indications 15
Base mounting accessories 16
Quick connect system 16
Mounting plate 18
Table mounting clamp (optional) 19
Actuators 20
Robot structure and skin 20
Wrist and Gripper 20
Robot communications and network interfaces 21

Getting started 23
Overview 23
What's in the box? 23
Manipulating the robot joints when the robot is powered off 25
Robot mounting options 25
Securing the mounting plate on a table with the table mounting clamp 26
Attaching the mounting plate to a surface with screws 27
Mounting plate bolting pattern 28
Installing the robot on the quick connect base 29
Installing the quick connect system 30
Attaching the robot base to the mounting plate 32
Mounting the robot directly onto a surface 33
Base underside bolting pattern 35
Robot power adapter 36
E-stop (optional) 36
Powering on the robot 37
Power-up, booting, and initialization sequence 37
Operating the robot 38
Overview of operating the robot 38

Supported control devices....38
Home and retract positions....44
Connecting a computer to the robot....45
Establishing a communication link between the robot and a computer via USB (RNDIS)....45
Connecting to a computer using a USB type-A to Ethernet adapter and Ethernet cable (Optional)....47
KINOVA® KORTEX™ *Web App*....49
Connecting a computer to the robot via Wi-Fi....51

Dimensions, specifications, and capabilities....53
Schematic and dimensions....53
Technical Specifications....54
Sensors....56
Base readings available....57
Actuators readings available....58
Wrist readings available....58
Gripper pose and twist readings available....59
Gripper finger motor readings available....59
Effective workspace....60
Payload vs. workspace....61

Concepts and terminology....63
Robot key concepts....63
Terminology reference....65
General mathematics and robotics....65
Features, components and functionalities....67
Control Modes....68

Robot control....69
High-level and low-level robot control....69
High-level and low-level robot control methods reference....69
Control features....71
Singularity avoidance....71
Protection zones....72
Joint limits....73
Cartesian limits....74
High-level control modes description....75
Trajectory control modes....75
Joystick control modes....76
Low-level control detailed description....77

Configurations and safeties....79
Configurable parameters....79
Base configuration....79
Actuators configuration....80
Device configuration....81
Safety items....82
Base safeties....82

Actuators safeties....83
Gripper safeties....86

KINOVA® KORTEX™ Web App User Guide....88
Introduction....88
Purpose....88
Device availability of *Web App*....88
Platform and browser support....90
User login....91
Web App layout and navigation....92
Robot control panel....95
Pose virtual joystick control....95
Angular virtual joystick control....97
Virtual joystick keyboard shortcuts....98
Actions panel....100
Snapshot tool....100
Main pages....100
Configurations page group....100
Safeties....104
Operations page group....105
Systems page group....115
Users....120

KINOVA® KORTEX™ Developer Guide....123
Introduction....123
Devices and services....123
Available services....124
Services, methods, and messages....125
KINOVA® KORTEX™ API and Google Protocol Buffer....125
Service client-server model....125
Blocking and non-blocking calls....126
Device routing and transport....126
Robot servoing modes....127
High-level servoing....127
Low-level servoing....128
Notifications....129
Error management....129
KINOVA® KORTEX™ GitHub repository....129
KINOVA® KORTEX™ ROS packages and GitHub repository overview....130

Guidance for advanced users....131
Overview....131
Reference frames and transformations....131
Standard robot frames....131
Homogeneous transforms....132
Homogeneous transform matrices....132
Denavit-Hartenberg (DH) parameters....134
Singularity configurations....136
Inertial parameters definition....138

Inertial parameters of the robot....139

Maintenance and troubleshooting....142
Maintenance....142
Troubleshooting....143
How to respond to safety warnings and errors....144
Contacting Kinova support....145

Introduction

Welcome

Welcome to the KINOVA® Gen3 lite 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 at support@kinova.ca.

About this document

⚠ 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 updates to this document. To download the most up to date version of this document, visit the Gen3 lite Technical Resources section of our website(https://www.kinovarobotics.com/en/resources/gen3-lite-technical-resources)).

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

Normal use definition

This section describes the normal use of the robot.

The definition of *normal use* includes lifting, pushing, pulling, or manipulating a maximum load of:

- mid-range, continuous: 600 g
- full-range, continuous 500 g

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

© Kinova inc 2021. All rights reserved.

- actuators - CPU core and motor temperatures
- gripper - micro-controller core and gripper motor temperatures

During normal operation, the robot joints are subject to heating. 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.

The expected service life of the robot is five years when the robot is used continuously for one hour per day. The warranty period is two years.

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 must be kept up to date with any changes in the parameter settings, work environment, or tasks of the robot. The risk assessment should address configuration settings as well as the need for any additional emergency stop buttons.

Safety directives and warnings

Directives, warnings and safety considerations for the KINOVA® Gen3 lite 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 that will slow the descent in the absence of external power.

© Kinova inc 2021. All rights reserved.

⚠ 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.
⚠ 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.
	- attempt to backdrive / force the robot joints to turn by hand while the robot is powered on. This may cause damage to the robot. The joints may be turned slowly by hand only when the robot is powered off.
	- install the robot base within 20 cm of your body (base contains a Wi-Fi transmitter).
	- use the robot to submerge objects in water.
	- operate or store the robot outside the recommended temperature ranges.
	- manipulate containers with hot or extremely cold liquids using the robot.
	- operate the robot in an environment which includes atomized flammable dust / particles or explosive / flammable gases, etc.
	- manipulate sharp objects with the robot.
	- drop or hit the robot, or force parts of the robot against itself. If this happens, before using the robot again, move each robot through the entirety of its joint range and then run the demo sequence, verifying no errors or warnings are present. In case of any doubts, contact Kinova support.
	- move the robot recklessly in such a way that there is a realistic risk of injuring bystanders or breaking objects in the operating environment of the robot.
	- power up and boot, reboot, or upgrade firmware of the robot unless the robot is in a stable position.
⚠ 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.
	- children are **not** left unsupervised with the robot.
	- 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).
	- eye protection is worn when manipulating fragile objects with the robot.
	- the robot has its base securely fixed to the work surface when in operation.
	- 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).

© Kinova inc 2021. All rights reserved.

⚠ 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.
⚠ For transport, the robot and accessories should be placed in the original packaging with the shipping box secured in place to avoid excessive jostling during transport.
⚠ Do not expose the robot to heavy rain.

Table 3: Maintenance

⚠ Perform regular cleaning of the robot, as well as visual inspection for damage and wear, particularly the gripper and fingers.
⚠ If the robot is exposed to rain, contact Kinova support to schedule maintenance by an authorized Kinova technician.
⚠ Immediately following exposure to saline air conditions, contact Kinova support to schedule maintenance by authorized Kinova technician.
⚠ Verify robot functionality if the robot is suspected to be potentially damaged following improper manipulation of the robot or if the robot is submitted to an excessive shock or drop.
⚠ Perform a quick routine check when starting the robot to catch defects that could occur with time.
⚠ If the robot loses a part (for example a shell due to impact) or a part breaks, shut down the robot safely and leave it off. Contact Kinova technical support.

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 inc 2021. All rights reserved.

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.

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 user guide 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 or via support@kinova.ca.

© Kinova inc 2021. All rights reserved.

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).

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

GPIO
General-Purpose Input/Output

IC
Integrated Circuit

IEEE
Institute of Electrical and Electronics Engineers

I/O
Input / Output

IP
Ingress Protection or Internet Protocol

IT
Information Technology

ISO
International Organization for Standardization

© Kinova inc 2021. All rights reserved.

LED
Light-Emitting Diode

NVRAM
Non-Volatile Random-Access Memory

PC
Personal Computer

ROS
Robot Operating System

RPC
Remote Procedure Call

RPM
Revolutions Per Minute

RS
Recommended Standard

SSID
Service Set IDentifier

TCP
Transmission Control Protocol

UART
Universal Asynchronous Receiver-Transmitter

UDP
User Datagram Protocol

UL
Underwriters Laboratory

UV
Ultraviolet light

VLAN
Virtual Local Area Network

WEEE
Waste of Electrical and Electronic Equipment

© Kinova inc 2021. All rights reserved.

Robot components

Overview

This section describes the main components of the KINOVA® Gen3 lite robot.

The robot consists of:

- base with controller
- actuators
- robot structure and skin
- wrist and gripper

The following image shows the main components of the robot.

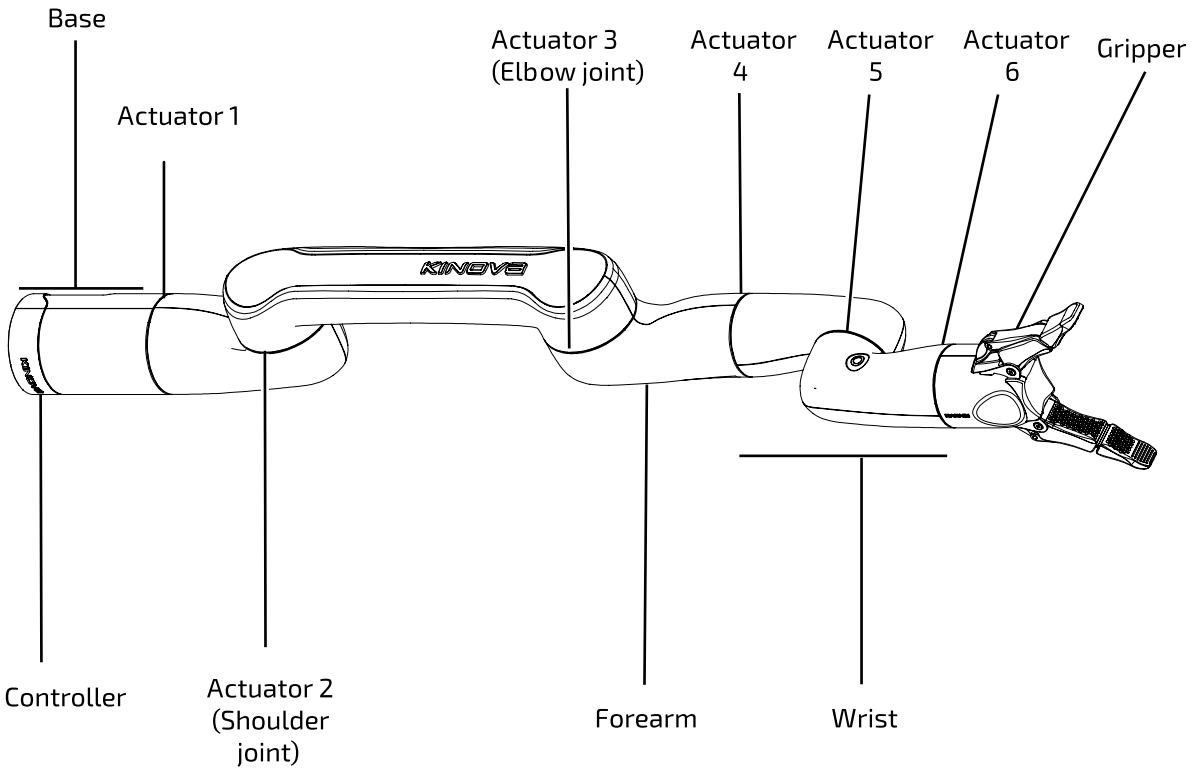


Figure 1: Robot main components

Base

The robot base provides for physical mounting of the robot and for power and communication / control. The base is the brains of the robot and contains a number of important internal components.

The robot robot base includes a built-in the controller for the robot. The base also includes a connector panel at the rear for connecting to power and external devices. The base has four mounting holes on its underside.

The internal components of the controller include:

- CPU

© Kinova inc 2021. All rights reserved.

- Wi-Fi / Bluetooth adapter (Only Wi-Fi is enabled at present)
- Ethernet switch and Ethernet interface
- USB interface
- piezoelectric buzzer for audio warnings and alarms
- 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. The base also includes a DHCP server.

Base connector panel

The base connector panel is located on the back of the controller part of the robot base. This provides a connection point for the power supply and various cables. It also contains the power on/ off switch and an LED indicator.

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

- On / Off power switch
- blue / green / red / amber LED indicator
- Micro USB B
- USB 2.0, type A
- 14-pin mini D ribbon joystick / expansion connector
- Lumberg 0317 08 (power and RNET connector)

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.

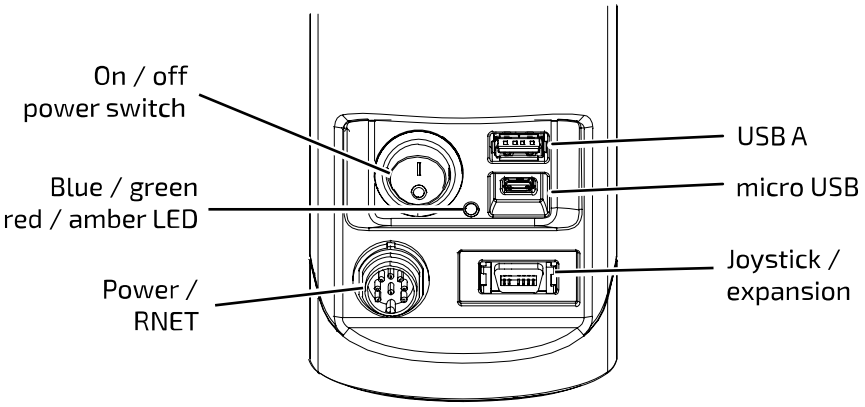


Figure 2: Controller connector panel

Base joystick / expansion connector pinout

This section describes the pinout details for the base joystick / expansion connector.

Joystick / expansion 14-pin connector

The base joystick / expansion connector pin assignment is described in the table below.

Table 4: Connector pinout

Pin	Name	Description
1	JOY_UP_SPEED	Joystick up input

© Kinova inc 2021. All rights reserved.

Pin	Name	Description
2	GND	Ground
3	GND	Ground
4	EXP_GPI1_STOP	Expansion GPI 1 or E-Stop
5	JOY_DOWN_DRI	Joystick down input
6	JOY_LEFT_REF	Joystick left input
7	24 V output (500mA)	24 V output at 500 mA
8	GND	Ground
9	EXP_GPI2	Expansion GPI 2
10	JOY_RIGHT	Joystick right input
11	JOY_BTN1_DET	Joystick button 1 input
12	EXP_GPI3	Expansion GPI 3
13	JOY_BTN2_SS	Joystick button 2 input
14	EXP_GPI4	Expansion GPI 4

Base LED indications

This section describes LED indicators on the base.

Introduction

The base connector panel includes a multi-color LED indicator. The following table explains the interpretation of the base LED indicator.

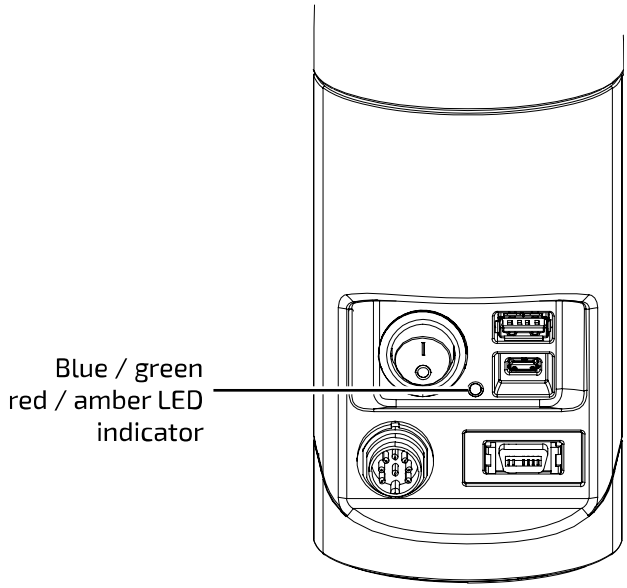


Figure 3: LED indicator

© Kinova inc 2021. All rights reserved.

Table 5: LEDs interpretation

Color	Meaning
Blue	The robot is booting
Green	The robot is ready to be used and there are no alarms and no faults
Amber	The robot is in an alarm condition
Red	The robot is in a fault condition

Base mounting accessories

The robot includes a number of accessories used to help with mounting the robot in different ways on a horizontal surface.

For the robot, there are different options for mounting the robot on a horizontal surface for use.

The robot package includes mounting accessories to make it easier to mount the robot. Some of these come standard with the robot, while others are optional.

- Quick connect system
  - Quick connect base
  - Mounting knob (pre-installed on the bottom of the robot base)
- Mounting plate
- Table mounting clamp (optional)

The robot base also includes four mounting holes on the bottom of the base that allow the robot to be mounted directly to a surface.

Quick connect system

The quick connect base is a two-part structure. This allows a quick connect base to be mounted in one place while allowing the robot to be quickly attached and detached.

The base of the robot is equipped with a quick connect system that allows for simple connect / disconnect of the robot base from the mounting point. This allows the robot to be quickly detached from the mounting site. This can be useful for:

- transport
- removal of the arm for servicing, or
- convenient re-siting of the arm between multiple installation sites

The quick connect system is supported out of the box and is the simplest and quickest way to get set up with the robot.

© Kinova inc 2021. All rights reserved.

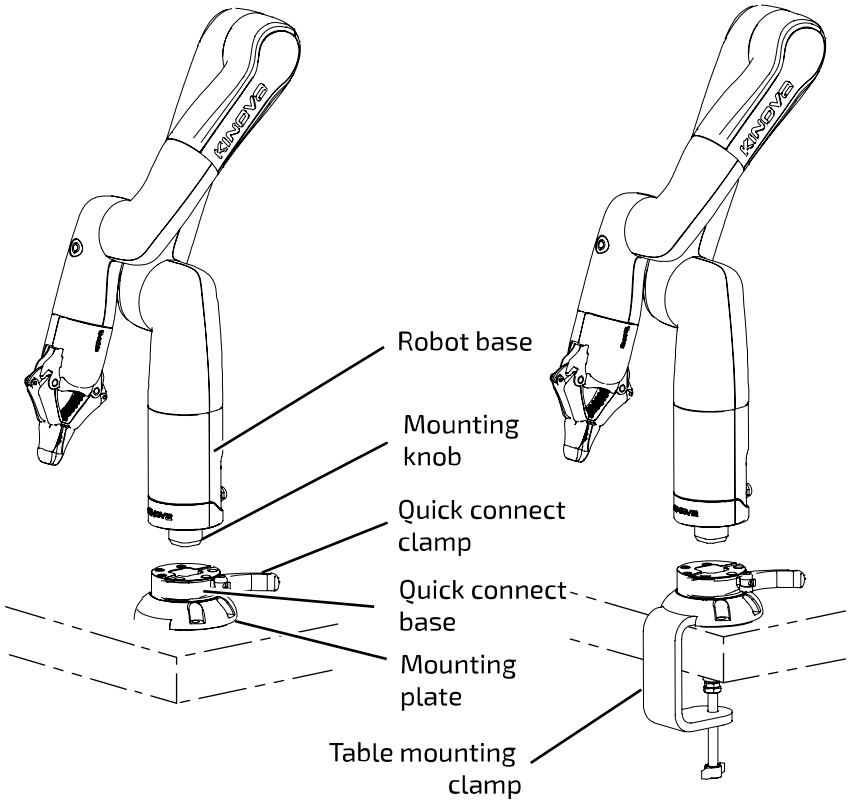


Figure 4: Quick connect - table screw mounting (left) and table clamp (right) mounting

A **mounting plate** attaches firmly to the mounting surface, either semi-permanently with screws or temporarily using a table mounting clamp. A **quick connect base** attached to the mounting plate provides an easily detachable docking point for the robot.

A (removable) **mounting knob** affixed to the bottom of the robot slides into a corresponding hole on the quick connect base, and a **quick connect clamp** locks the mounting knob to secure the robot in place. The robot ships with the mounting knob pre-installed by default.

© Kinova inc 2021. All rights reserved.

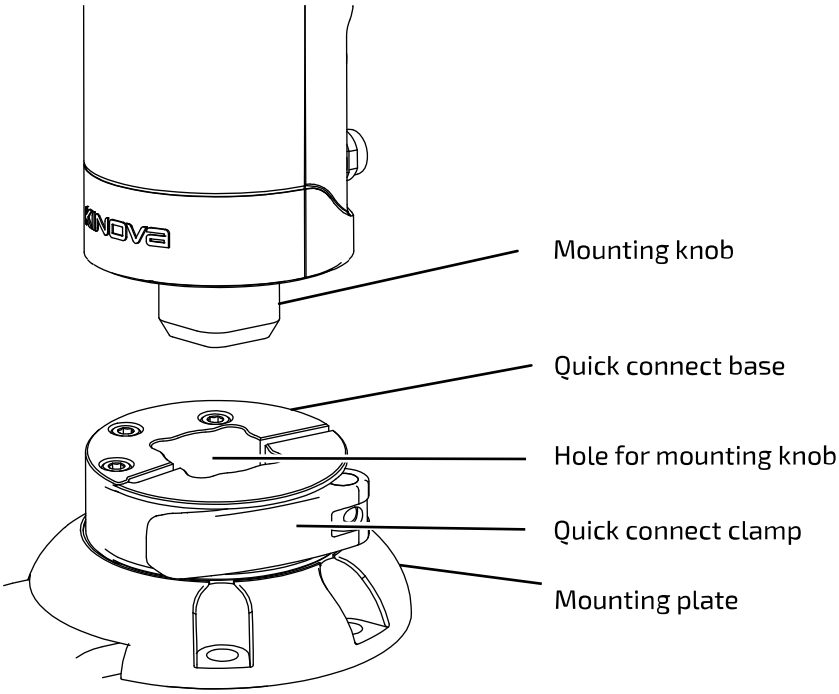


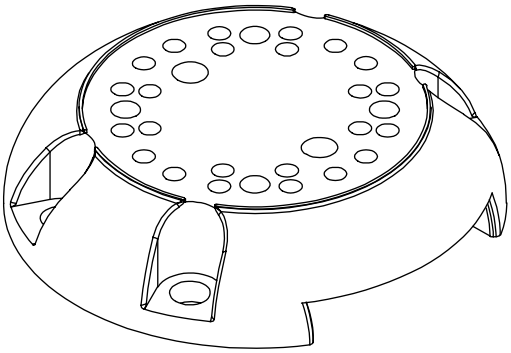
Figure 5: Quick connect (closeup)

To remove the robot from the quick connect base, flip open the quick connect clamp, detach any connected cables, and slide the robot out of the quick connect base.

Mounting plate

The mounting plate is useful for mounting the robot in different configurations.

The mounting plate provides a foundation for mounting the robot onto a horizontal surface. The mounting plate attaches to the surface, and the robot can be mounted on top of it.



The mounting plate can be secured to the surface using either screws or the table mounting clamp.

The table mounting clamp option allows for faster set up and removal, and is a good option for initial setup of the robot, for setting up demos, and in general for any temporary installation.

Securing the mounting plate to a surface using screws provides a more durable solution, but requires extra set-up time to drill holes into the surface according to the mounting plate holes pattern.

The mounting plate has a number of holes drilled through it to support various mounting options:

- 4 mounting holes for mounting on table or aluminum extrusion

© Kinova inc 2021. All rights reserved.

- 24 mounting holes to attach quick connect base (3 holes x 8 different possible orientations)
- 4 mounting holes to mount robot base without quick connect
- 2 M8 threaded mounting holes to attach table mounting clamp

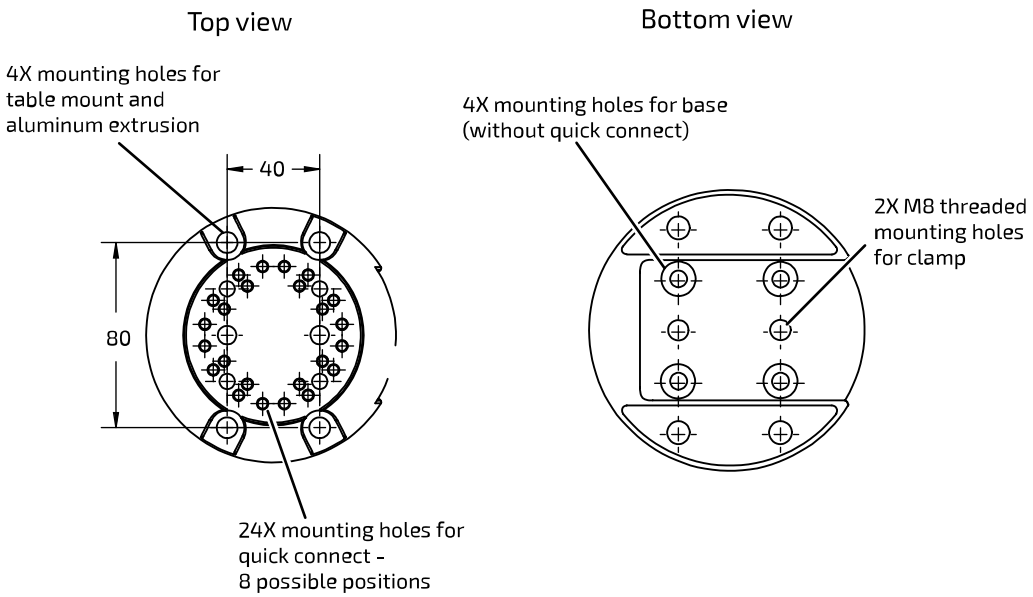


Figure 6: Mounting plate mounting holes patterns (dimensions in mm)

Table mounting clamp (optional)

The table mounting clamp provides a quick way to secure the mounting plate on a table top.

The clamp attaches to the bottom of the mounting plate with screws. The clamp can then be secured onto the table using a tightening knob on the clamp.

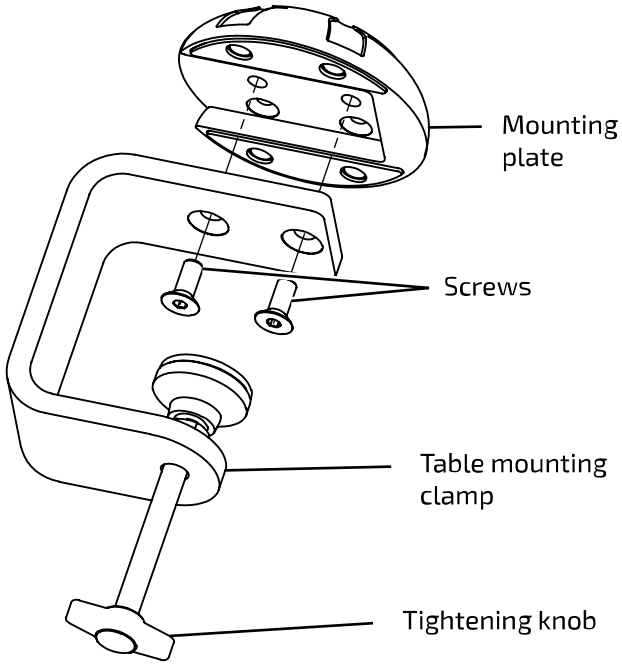


Figure 7: Table mounting clamp and mounting plate

© Kinova inc 2021. All rights reserved.

Actuators

The robot contains actuators which power the rotational movement at the robot joints. The robot contains three sizes of actuators with distinct performance capabilities. The actuators contain sensors.

The rotational motion at each of the joints of the robot is powered by rotary actuators. There is one actuator for each joint.

There are three types of actuator:

- small
- medium
- large

Each actuator is equipped with the following:

- current sensing
- temperature sensing (motor)
- temperature sensing (CPU)
- relative position and velocity sensing
- absolute position sensing

Wrist joints (joints 4 to 6) use small actuators. Medium actuators are used for joints 1 and 3. A single large actuators is used for the shoulder joint (joint 2).

Actuators use either high-efficiency planetary gearboxes (small and medium actuators) or low-backlash strain wave gearing (large actuator).

The actuators are connected to each other using a series of internal cables. These cables convey:

- power
- Ethernet data transmission

Robot structure and skin

The robot is held together by a structural skeleton serving as a mounting point for actuators and providing passages for internal wiring. The robot is covered by outer skin panels.

The robot structure is made up of two main elements:

1. Robot skeleton - provides structural integrity for the robot, mounting interface for actuators and control boards, and passage for internal wiring
2. Robot skin - protects the inner components of the robot while giving an appealing look to the robot

Note: In normal operations, there is no need or reason to remove skin panels.

Wrist and Gripper

The wrist is made up of the last three actuators of the robot and includes buttons. A two finger gripper is attached allowing a range of different grip options.

© Kinova inc 2021. All rights reserved.

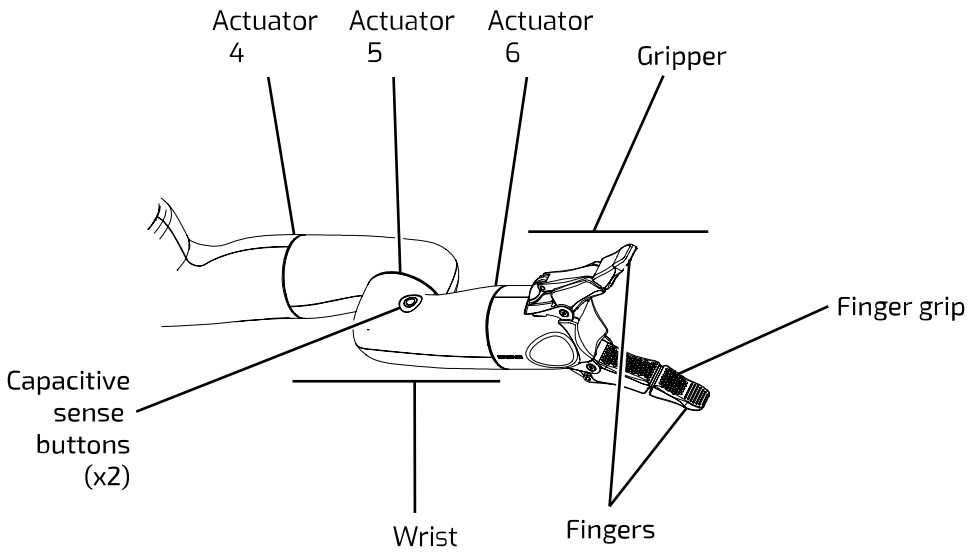


Figure 8: Wrist and gripper

The robot has a three-axis wrist with the range of motion enabled by actuators four through six.

The robot incorporates a two finger gripper. The finger body is composed of two different materials. The finger structure is made of a hard, but flexible plastic for structural integrity and strength, and a soft, rubber-like plastic for the grip.

The gripper fingers are actuated by a linear actuator inside the robot wrist and can open and close to accommodate a variety of grip types and object sizes.

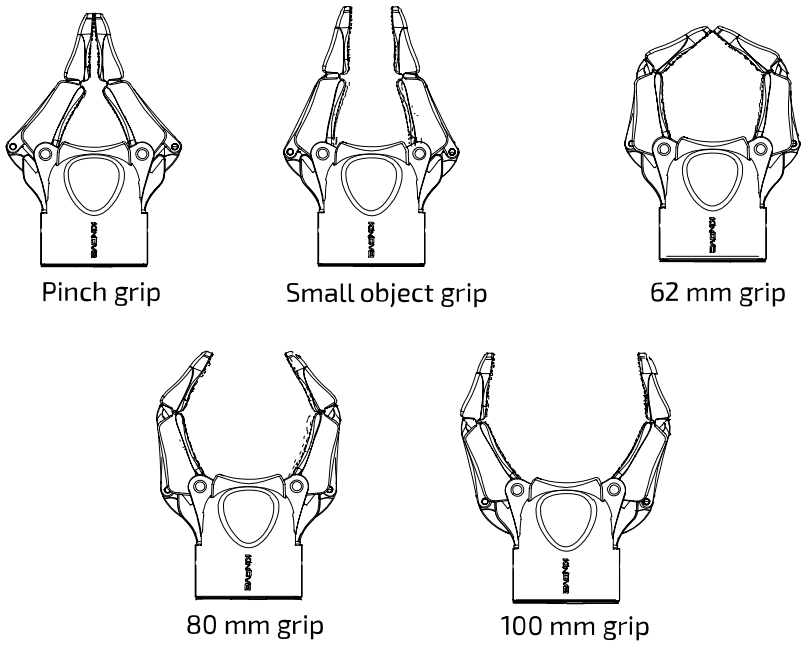


Figure 9: Grip options

Robot communications and network interfaces

The robot is made up of a number of devices. These devices contain Ethernet switches and are connected together in a network.

© Kinova inc 2021. All rights reserved.

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 internal cables which carry power and communications.

The base, actuators, and gripper each contain Ethernet switches.

Together, this enables Ethernet communications between all the devices (base, actuators, and gripper) with data carried between the base and the actuators and gripper over the internal cables. Users can connect to the robot base from a developer computer to control and configure the robot via wired connection in either of two ways:

- via the USB type-A port on the robot base. An optional USB type-A to Ethernet adapter is available for this purpose.
- over RNDIS using a micro-B USB port on the base. A micro-B USB to USB type-A cable is provided for this purpose.

The first option using the optional USB A to Ethernet adapter allows for connecting to the robot base over a local area network (LAN).

A Wi-Fi adapter in the base also allows for connecting to the robot base wirelessly over a local Wi-Fi network.

Each device connected to one of the Ethernet switches has an IP address to allow internal routing of communications.

The actuators and gripper have the following default IP addresses:

Table 6: 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`
Gripper	`10.10.0.18`

© Kinova inc 2021. All rights reserved.

Getting started

Overview

This section describes how to get started with your robot.

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
- connecting a computer to the robot
- connecting to the KINOVA® KORTEX™ *Web App*

A **quick start guide** which covers much of the same material in a simplified visual format is provided in a printed form in the box. A downloadable digital version is also made available on the Kinova website in the Technical Resources (https://www.kinovarobotics.com/en/resources/gen3-lite-technical-resources) section.

What's in the box?

This section describes the KINOVA® Gen3 lite robot shipping case contents.

The interior of the shipping box has two layers.

On opening the box, you see the top layer, which contains the printed quick start guide. There are handles on the left and right side of the top layer allowing you to lift out the top layer.

© Kinova inc 2021. All rights reserved.

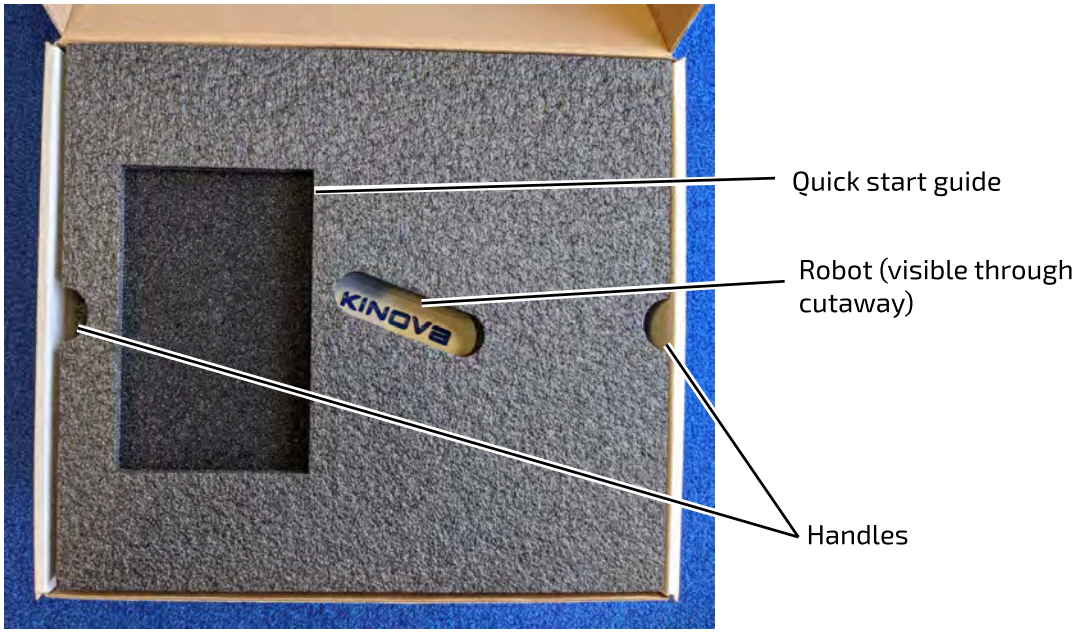


Figure 10: Shipping case contents (top layer)

Lifting out the top layer reveals the bottom layer containing the robot and associated accessories.

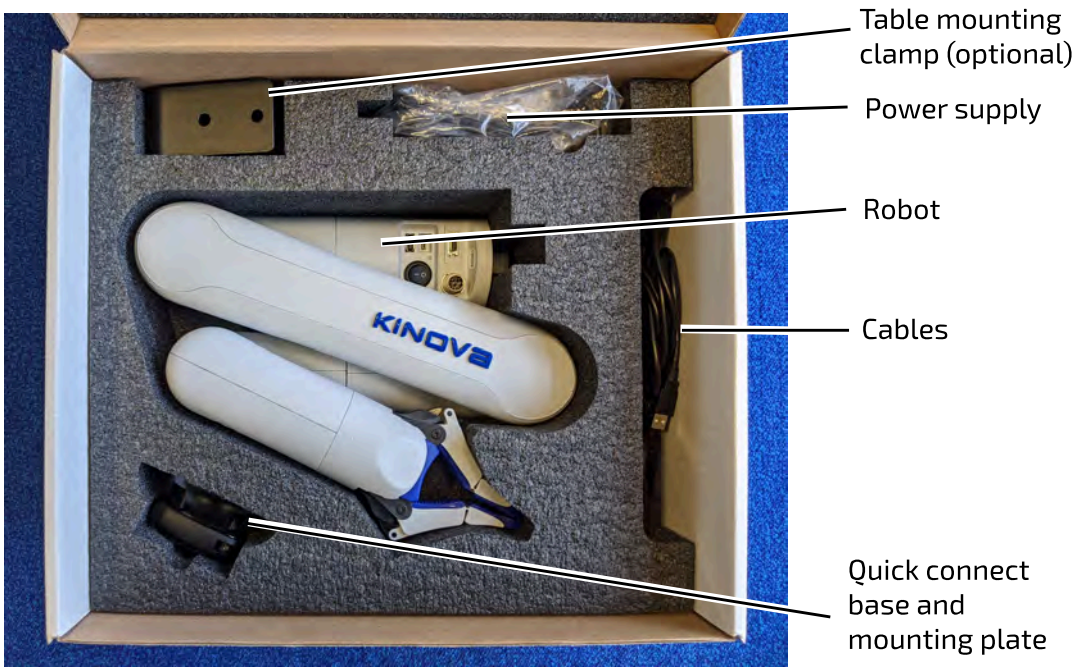


Figure 11: Shipping case contents (bottom layer)

The contents of the bottom layer of the box are as follows:

- robot
- power adapter and cable
- mounting plate and quick connect base
- table mount clamp (optional)
- micro-B USB to USB type-A cable
- tools and fasteners

© Kinova inc 2021. All rights reserved.

The following **optional** items, when included, are shipped with the robot, but packaged separately:

- Xbox gamepad and cable
- power cable E-stop
- Expansion I/O and E-stop cable
- USB-A to Ethernet adapter (to facilitate connecting robot to computer Ethernet port or local area network)

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

The robot joints can be manipulated by hand when the robot is powered off. This is useful when setting up the robot for the first time.

When the robot is powered on, 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: The shoulder joint (joint 2) is more stiff than the others, but can still be moved.

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 robot can be connected and used.

It is possible to mount the robot in different ways, depending on the needs of your particular application:

- Robot mounted using quick connect system, with installation using
  - table clamp
  - mounting plate affixed to surface with screws
- Robot mounted onto to mounting plate, with mounting plate secured using:
  - table clamp
  - screws, affixed to surface
- Robot base mounted directly onto surface using screws

© Kinova inc 2021. All rights reserved.

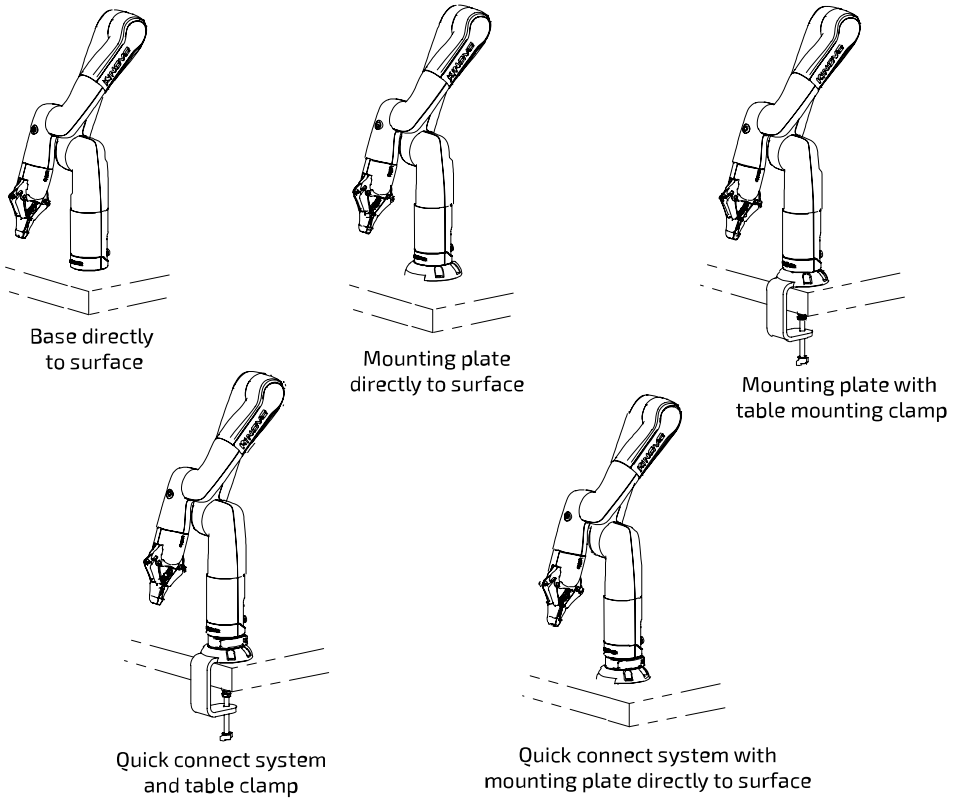


Figure 12: Robot mounting options

Securing the mounting plate on a table with the table mounting clamp

This section describes how to secure the mounting plate on a table using the table mounting clamp.

Before you begin

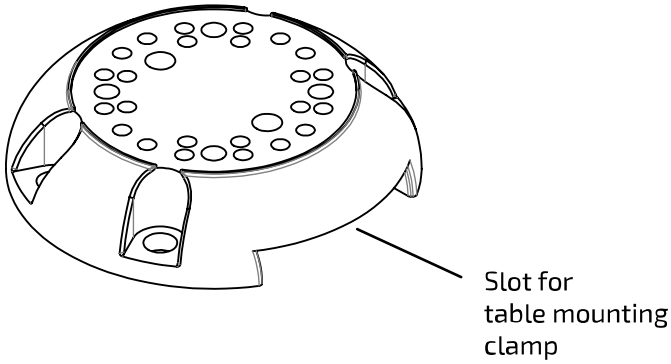
You should determine in advance whether you want to mount the robot in a fixed way onto the mounting plate, or whether you want to use a quick connect setup. If you want a fixed mounting of the robot onto the mounting plate, you will need to do that first, because you will need access to the bottom of the mounting plate to affix the robot base to the mounting plate.

About this task

The table clamp allows for a quick temporary setup of the robot on a table top. The table clamp must be affixed to the mounting plate using screws.

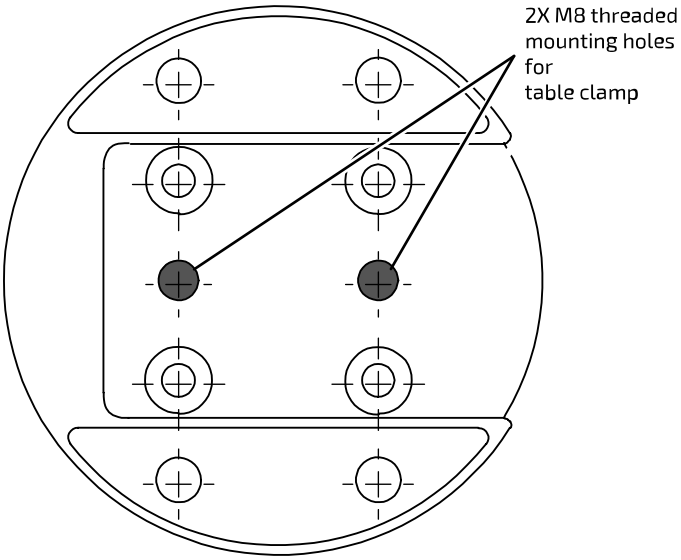
Procedure

1. Place the table mounting clamp into the slot on the bottom of the mounting plate.



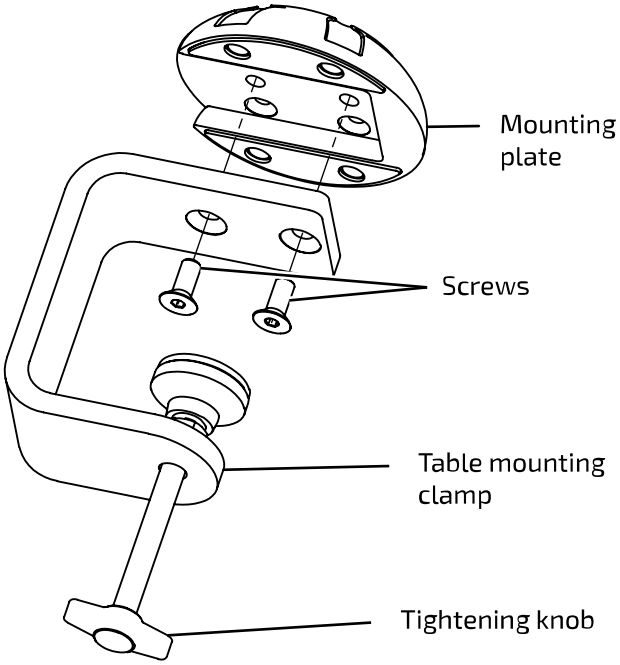
© Kinova inc 2021. All rights reserved.

2. Align the two holes on the table mounting clamp with the two corresponding holes on the bottom of the mounting plate.



Bottom view of mounting plate

3. Attach the table clamp to the mounting plate with screws.



4. Turn the tightening knob on the table clamp to open up the clamp and then slide the clamp, with the mounting plate attached, over the edge of the table.
5. Turn the tightening knob by hand until the mounting plate is firmly clamped onto the table top.

   Note: Do not overtorque.

Attaching the mounting plate to a surface with screws

This section describes how to attach the mounting plate to a surface with screws.

© Kinova inc 2021. All rights reserved.

Before you begin

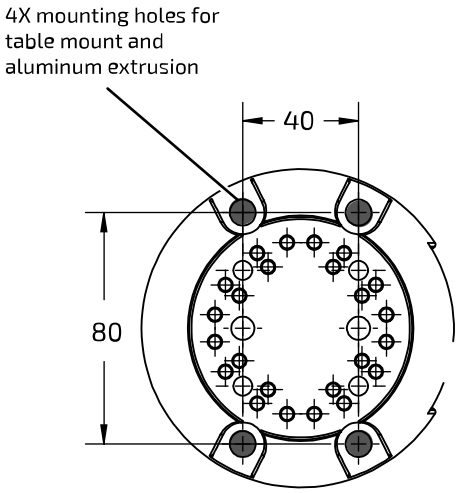
You will need four M8 socket head cap screws (SHCS) for attaching the mounting plate. You should determine in advance whether you want to mount the robot in a fixed way onto the mounting plate, or whether you want to use a quick connect setup. If you want a fixed mounting of the robot onto the mounting plate, you will need to do that first, because you will need access to the bottom of the mounting plate to affix the robot base to the mounting plate.

About this task

The mounting plate can be mounted to a surface using either a table mounting clamp or with screws. The clamping mount offers a faster, simpler, and more flexible solution that is good for temporary setups like for demos or initial experimentation. Attaching the mounting plate with screws however offers a more stable, permanent solution that works better for a fixed installation at a site.

Procedure

1. Drill holes in the mounting surface based on the bolting pattern of the mounting plate.



Top view of mounting plate

2. Place the mounting plate on the surface, aligning the mounting holes with the holes in the surface.
3. Pass the screws through the mounting holes from the top of the mounting plate and affix the mounting plate to the surface using the four M8 screws.

What to do next

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

Mounting plate bolting pattern

This section describes the bolting pattern of the mounting plate for affixing it to a surface or aluminum extrusion with screws. This is useful when mounting the robot to a surface using the mounting plate.

Overview

The mounting plate has four screw holes available for mounting the plate to a flat horizontal surface or to an aluminum extrusion.

© Kinova inc 2021. All rights reserved.

Mounting details

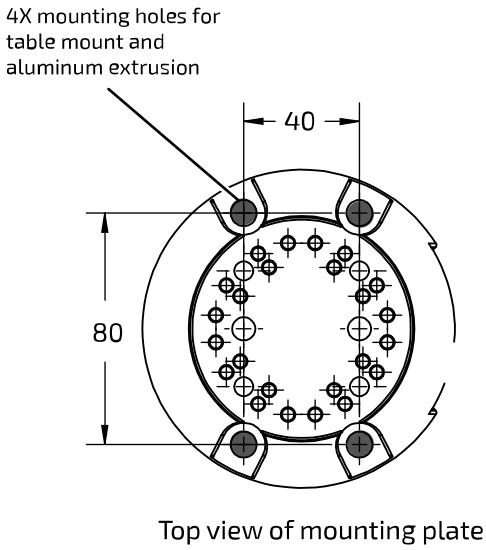


Figure 13: Mounting plate bolting pattern

Installing the robot on the quick connect base.

This section describes how to install the robot on the quick connect base.

Before you begin

Install the quick connect system.

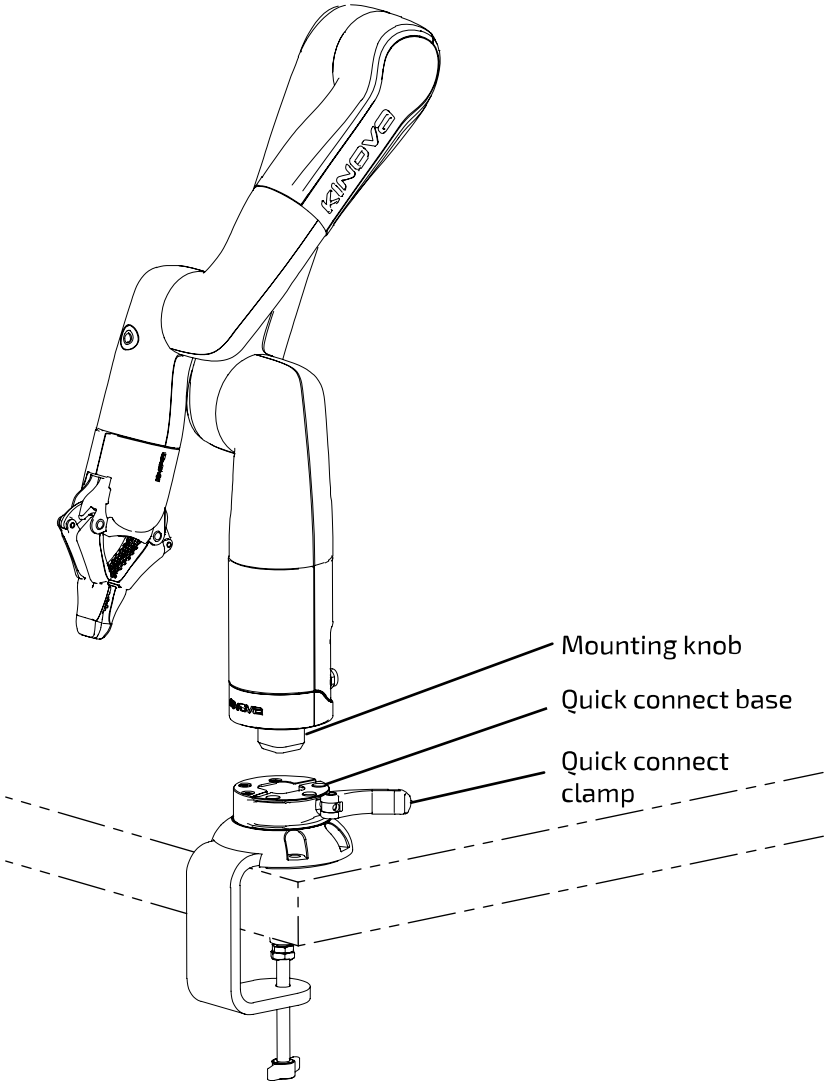
About this task

Procedure

1. Make sure that the quick connect clamp on the quick connect base is opened.

© Kinova inc 2021. All rights reserved.

2. Lower the base of the robot so that the mounting knob goes into the opening on the quick connect base.



3. Once the robot is fully lowered onto the quick connect base, close the quick connect clamp to secure the robot in place.

Installing the quick connect system

This section describes how to install the quick connect system, including the quick connect base and the base mounting knob.

Before you begin

You will need screws to install the quick connect base on the mounting plate and a screw to install the mounting knob on the bottom of the robot. The mounting plate should already be mounted to a surface, either with screws or using the table mounting clamp.

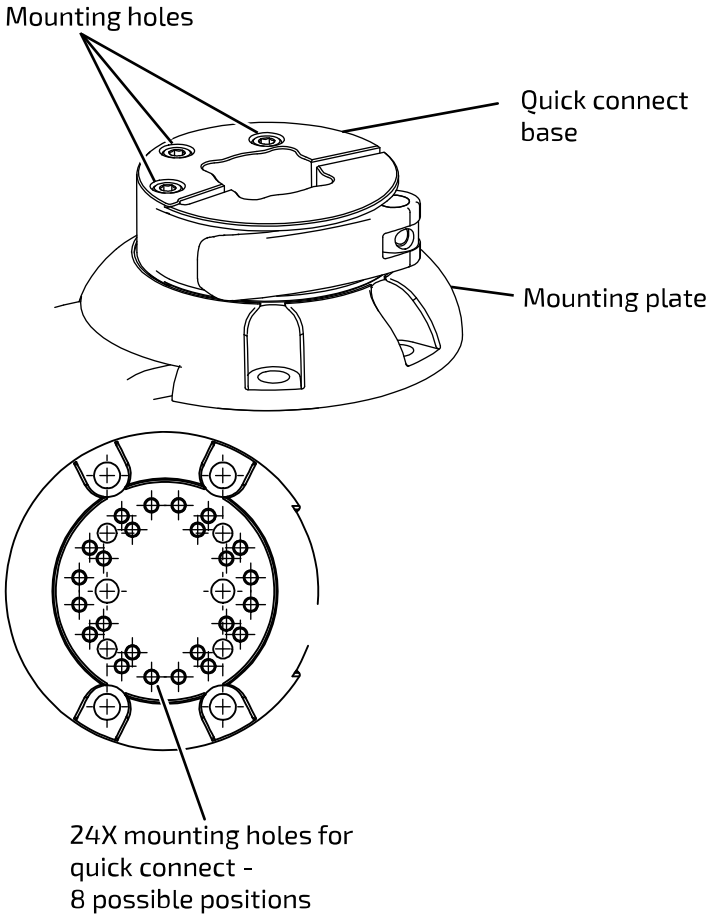
About this task

The quick connect system offers a mechanism to easily move a robot between multiple installation sites. The quick connect system includes a mounting base where the robot can be quickly docked and undocked without having to go to the trouble of a full installation and deinstallation.

© Kinova inc 2021. All rights reserved.

Procedure

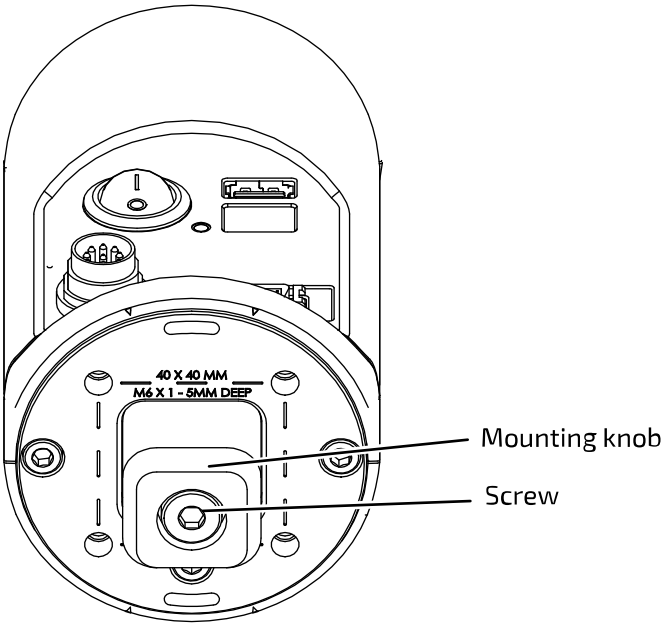
1. The mounting plate has 24 same-sized mounting holes that allow the quick connect base to be attached in eight different orientations. The quick connect base has three corresponding-sized mounting holes. Align the quick connect base on the mounting plate so that holes line up.



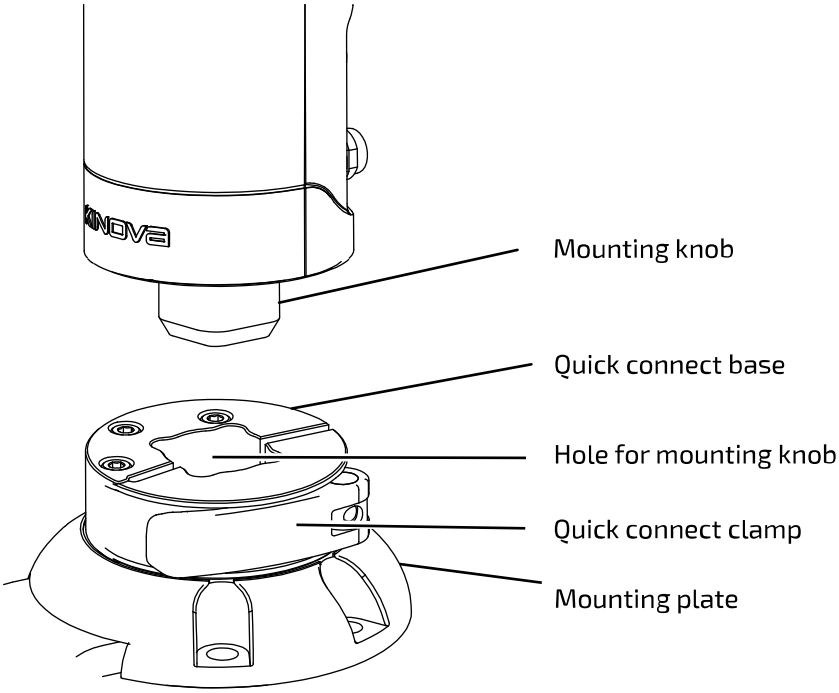
2. Attach the quick connect base to the mounting plate using screws.
3. When the robot is initially shipped, the mounting knob will already be attached to the bottom of the robot base. If the knob has since been removed (to mount the robot previously directly on a surface or onto the mounting plate), you will need to re-attach it to use the quick connect. Attach the mounting knob to the bottom of the robot using a single screw passing through the

© Kinova inc 2021. All rights reserved.

center of the mounting knob. If the mounting knob was still already attached, there is nothing more to do.



Results



Attaching the robot base to the mounting plate

This section describes how to attach the robot base to the mounting plate.

Before you begin

If the robot has the mounting knob attached for use with a quick connect system, the mounting knob needs to be removed first. Also, to attach the robot base to the mounting plate, you will need access to the bottom of the mounting plate. So if the mounting plate is affixed to a surface, you will need to detach it first to perform the following procedure.

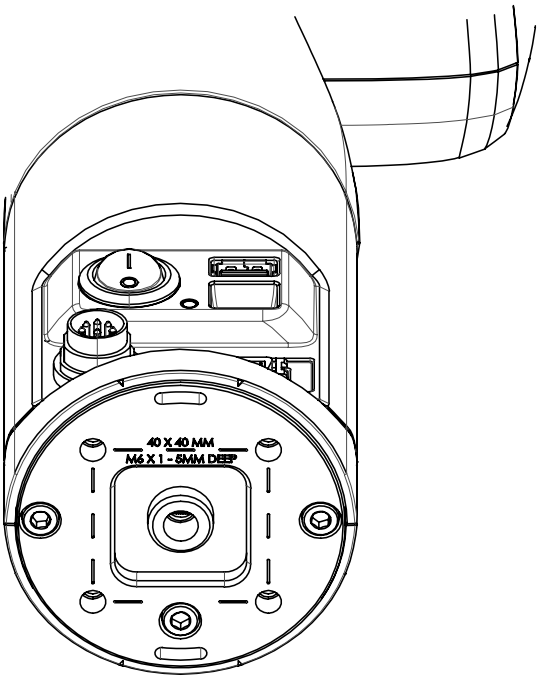
© Kinova inc 2021. All rights reserved.

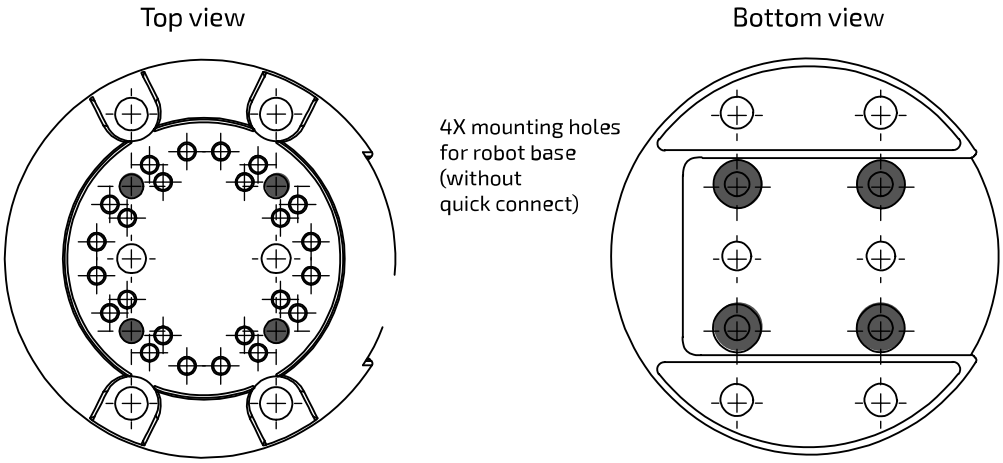
About this task

The mounting plate has four M6 mounting holes on the top in a square pattern matching the bolting pattern on the bottom of the robot base. This can be used to mount the robot base onto the mounting plate.

Procedure

1. Bring together the top surface of the mounting plate with the bottom of the robot base with the square hole pattern on the mounting plate aligned with the holes on the bottom of the robot base.





2. Pass M6 screws through the mounting plate from below into the holes in the bottom of the robot base. Turn the screws until tight.

Mounting the robot directly onto a surface

This section describes how to mount the robot directly on a flat, horizontal surface .

© Kinova inc 2021. All rights reserved.

Before you begin

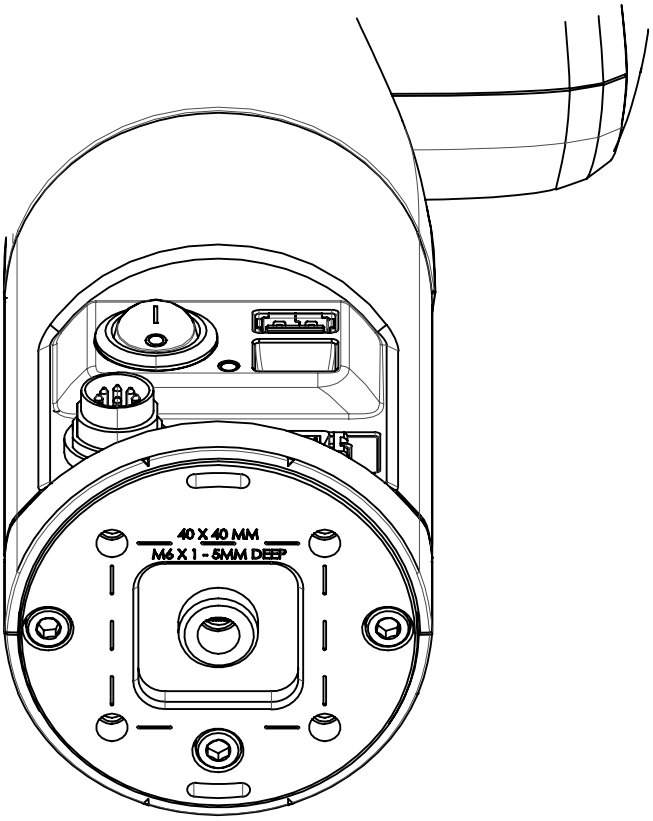
You will need four M6 screws to affix the robot to the surface. If the robot has the mounting knob attached for use with a quick connect system, the mounting knob needs to be removed first.

About this task

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

Procedure

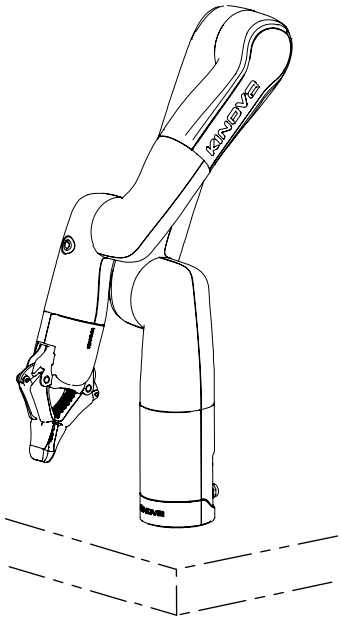
1. Using the base bolting pattern as a guide, drill holes through the surface. Since the base is to be mounted directly to the surface, and the holes on the robot base are on the underside, the holes in the surface will have to be drilled all the way through the mounting surface.



2. Use appropriate length M6 screws to mount the robot base controller or the mounting plate to the surface. The screws will need to go through the mounting surface from the other side.

© Kinova inc 2021. All rights reserved.

Results



What to do next

You can now proceed to connect the robot to the power supply.

Base underside bolting pattern

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

Overview

The underside of the base has four M6 screw holes for mounting purposes, with the centers of the holes at the corners of a 40 mm x 40 mm square. These holes can used for attaching the robot base to either the mounting plate or directly to a surface. In the second case, mounting directly onto a surface, holes must be drilled through the surface so that screws can go through from the other side and into the base mounting holes from underneath.

© Kinova inc 2021. All rights reserved.

Mounting details

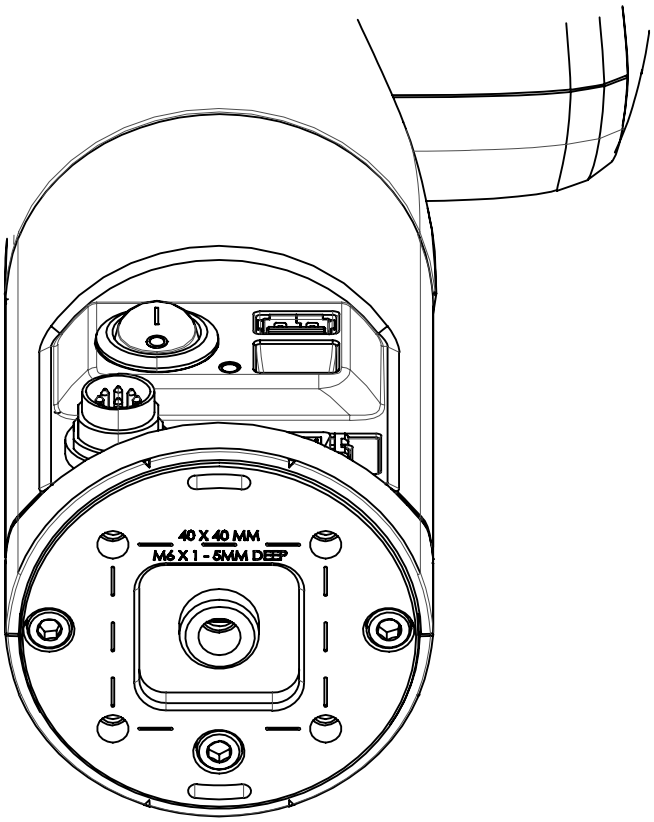


Figure 14: Base and mounting holes

Robot power adapter

The robot comes with a power adapter and cable to supply the robot with power from a wall outlet.

The robot is powered by a 24V power supply (Delta 150AAS24B).

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 0321 08 connector.

⚠ When the power is cut, the robot will descend. There are mechanisms within all the 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.

E-stop (optional)

An optional emergency stop (E-stop) allows users to shut down the robot quickly in case of an emergency.

Integrated E-stop with power adapter cable

This option includes a push-button E-stop integrated in series with the power adapter.

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

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

© Kinova inc 2021. All rights reserved.

Behavior of the robot after cut to power supply

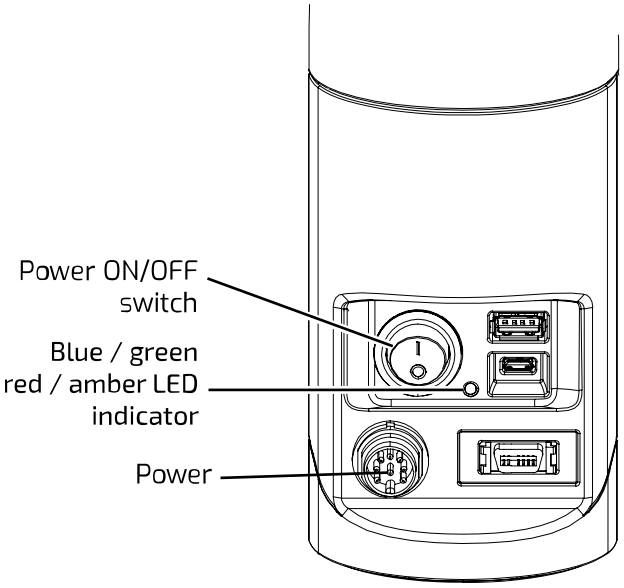
After the power supply to the robot or actuators is cut, the robot will fall, gracefully, under its own weight.

Powering on the robot

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

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 base 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. Flip the power switch. This will initiate the power up sequence.



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

To power down the robot, simply flip the power switch. The robot will undergo a shutdown procedure.

Note: When the robot has recently been powered down, wait at least five seconds before attempting to power the robot back on.

Power-up, booting, and initialization sequence

On pressing the power button of the robot, the robot goes through a boot up and initialization sequence. The LED indications on the base give visual feedback during the power-up sequence.

When the power switch is flipped 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:

© Kinova inc 2021. All rights reserved.

Table 7: Power-up sequence LEDs indications

Sequence step	LEDs indications
Robot booting	Blue LED
Robot ready for use	Green LED

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

Operating the robot

Overview of operating the robot

This section describes the supported control devices for 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 control devices

This section describes the supported control devices for the robot.

The robot currently supports the Xbox gamepad.In particular, the Xbox 360, Xbox One, Xbox Series S, and Xbox Series X.

Connecting an Xbox gamepad to the robot, wired

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

Before you begin

You will need:

- Xbox gamepad
- micro-B USB to USB type-A cable (included)

Procedure

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

Default control device mapping - Xbox gamepad

This section describes the default control device mappings between the Xbox gamepad and the actions on the robot. Customize the mapping between the gamepad and the robot using the API or the Web App.

Gamepad maps overview

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

1. Twist linear (controls the robot end effector translations by velocity)

© Kinova inc 2021. All rights reserved.

2. Twist angular (controls the robot end effector rotations by velocity)
3. Joint (controls the robot joint by joint by velocity)

General controls

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

- Changing the active control map to the next or previous map in the list
- Opening and closing the gripper
- Clearing faults or E-Stop - a fault state or the triggering of an E-Stop signal will make itself known through a red LED on the base controller of the robot. The robot cannot be moved this if state is present. Pressing the left bumper clears the fault (removes the E-Stop) and returns the LED to green, returning normal control.
- Applying emergency stop signal - this will stop the robot. As with a fault, the LED will change to red and control of the robot will be prevented until this state is cleared.
- Reaching home or retract position

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

Table 8: General control map elements (common controls applying to ALL maps)

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

© Kinova inc 2021. All rights reserved.

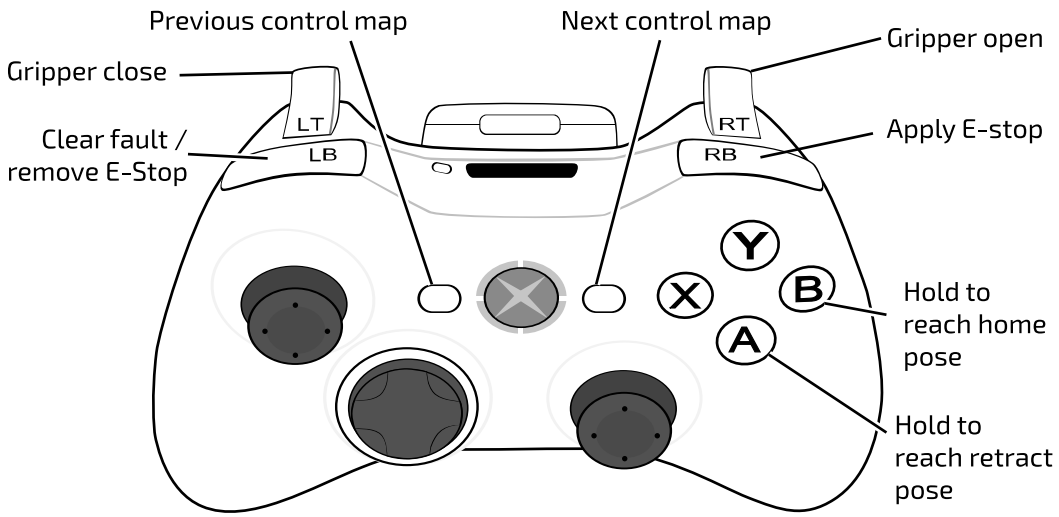
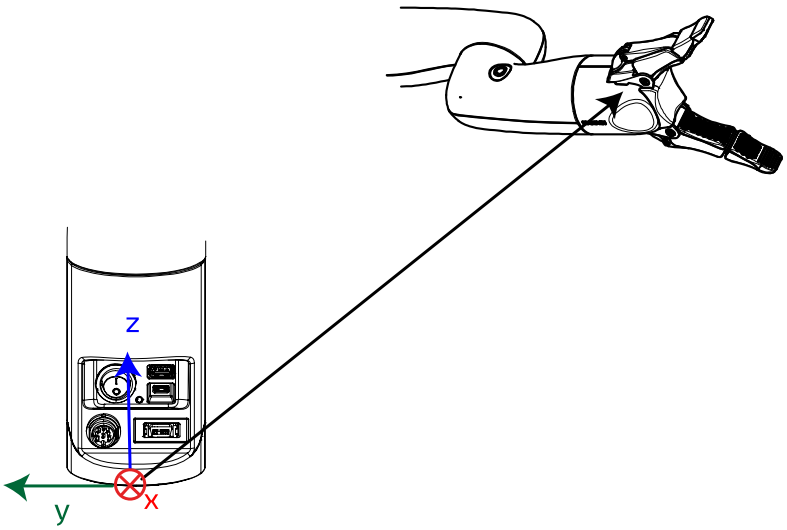


Figure 15: General control map elements with Xbox gamepad

Twist linear map

Twist linear is the default gamepad map when the robot is turned on and the controller is connected. In this mode the tool is translated in space with respect to the configured Cartesian translation frame (by default the base frame). The tool orientation does not change in this map. The user controls the linear velocity of the tool, including the linear speed.



-->

© Kinova inc 2021. All rights reserved.

Table 9: Twist linear - general controls plus

Action		Control	
Cartesian X translation	-	down	Left stick
	+	up	
Cartesian Y translation	+	left	
	-	right	
Cartesian Z translation	-	down	Right stick
	+	up	
Speed	decrease	down	D-pad
	increase	up	

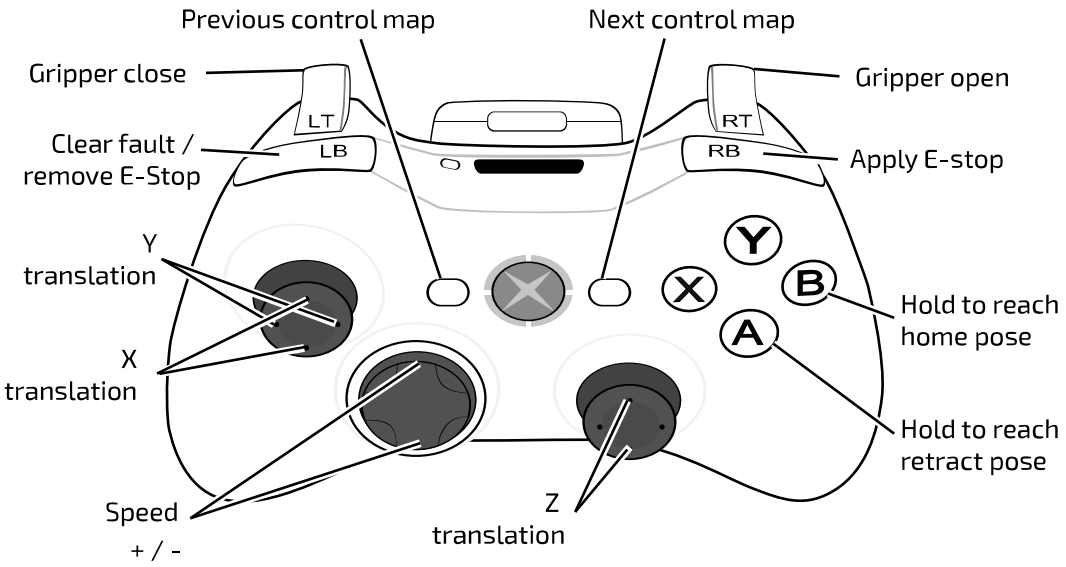


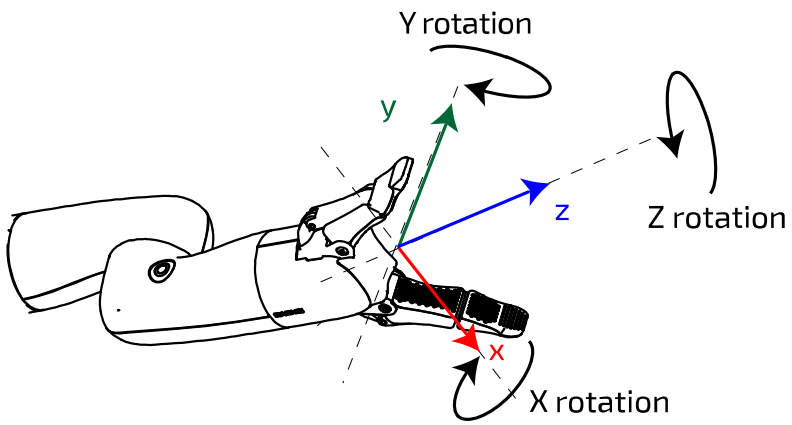
Figure 16: Twist linear controls with Xbox gamepad

Twist angular map

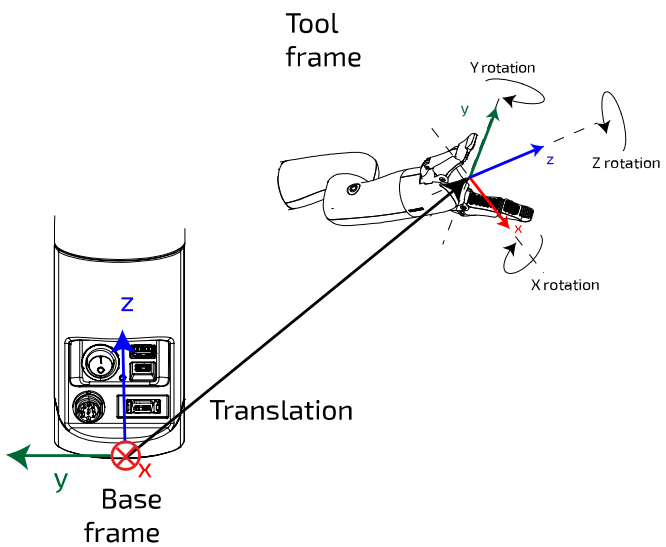
The twist angular map is used to change the orientation of the gripper/tool without translating the gripper/tool.

Twist angular can be thought of as a companion to the Twist linear control map. In Twist linear, the tool is translated with respect to the configured Cartesian translation frame while leaving the orientation unchanged. In Twist angular, the control is pure rotation of the tool within the configured Cartesian orientation frame (by default the tool reference frame), around the three axes of that frame. The user controls the angular velocity of the tool in relation to those three axes.

© Kinova inc 2021. All rights reserved.



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).



© Kinova inc 2021. All rights reserved.

Table 10: Twist angular - general controls plus:

Action		Control	
Cartesian Y rotation	-	left	L stick
	+	right	
Cartesian X rotation	-	down	
	+	up	
Cartesian Z rotation	-	left	R stick
	+	right	
Speed	decrease	down	D-pad
	increase	up	

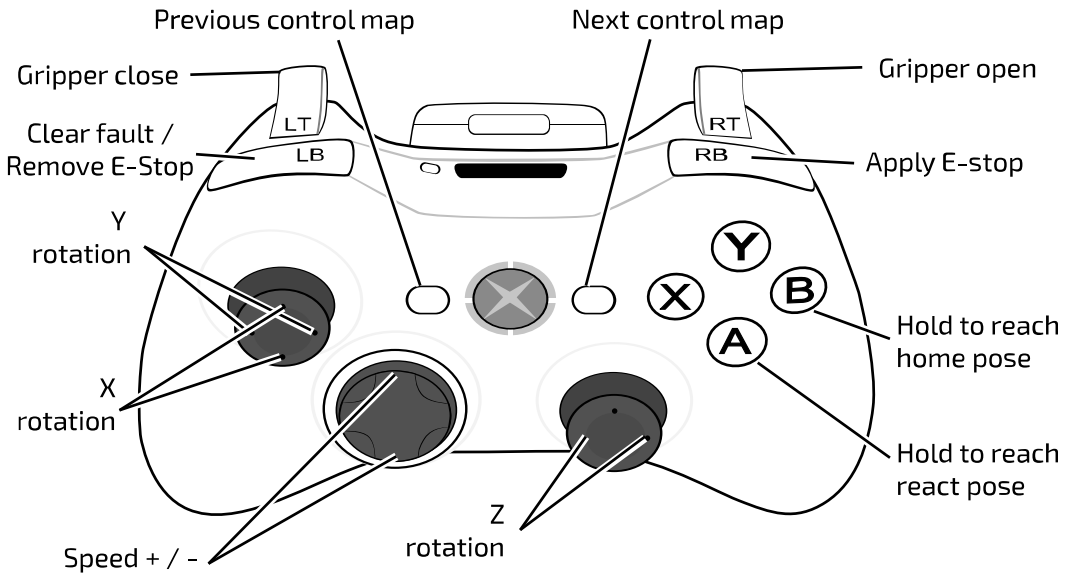


Figure 17: Twist angular controls with Xbox gamepad

Joint map

The joint map allows actuators to be chosen and moved one by one.

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 11: Joint - general controls plus:

Action		Control	
Joint speed	ω-	left	L stick
	ω+	right	
Speed	increase	up	D-pad
	decrease	down	

© Kinova inc 2021. All rights reserved.

Action		Control	
Navigate joints	Previous	left	
	Next	right	

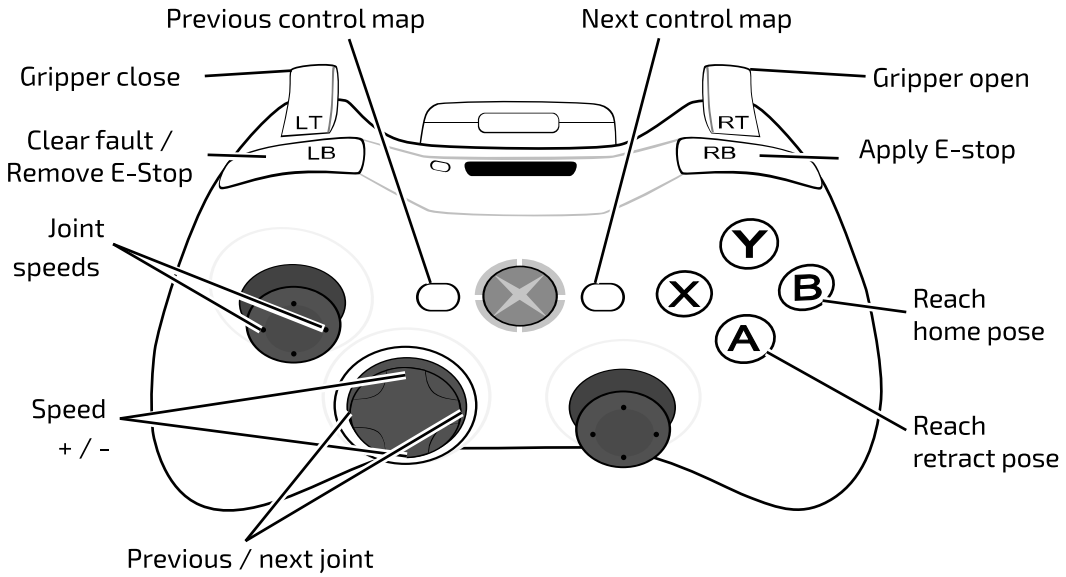


Figure 18: Joint controls with Xbox gamepad

Home and retract positions

Pre-configured Xbox gamepad maps set aside two buttons to move the robot to the home and retract positions of the robot.

The robot includes two pre-configured poses that can be reached using the Xbox gamepad.

The **home** position sets the robot into a convenient "ready" position. The home position is reached by holding down the **B** button on the Xbox gamepad.

The **retract** position folds the robot up into a compact pose. This can be a useful position to out the robot into for periods when it will not be in use. The retract position is reached by holding down the **A** button on the Xbox gamepad.

Figure 19: Reach home and retract controls

© Kinova inc 2021. All rights reserved.

Connecting a computer to the robot

Establishing a communication link between the robot and a computer via USB (RNDIS)

Describes steps to connect the robot to a computer via USB.

Before you begin

You will need a micro-B USB to USB type-A cable (provided with the robot). The robot needs to have been turned on.

About this task

The default method to connect the robot to a computer is to connect the micro USB of the robot base to a USB-A port on the computer. The DHCP server on the robot base is enabled by default and will assign an IP address to the computer RNDIS adapter to allow Ethernet over USB communication between the robot and the computer. No configuration is necessary to enable the communication. This is the most straightforward way to connect the robot to a computer, and provides simple point-to-point connectivity between robot and computer.

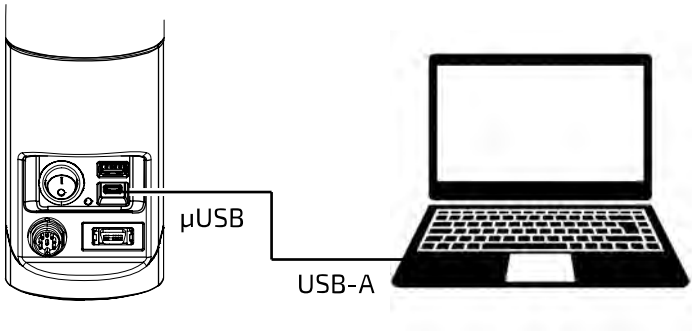
Note: This method does not allow for connection to a computer over a local area network.

Note: The description and illustrations that follow are for a computer with Windows 10. Details for other operating systems will be different, but similar. For more details, see the documentation for your OS.

To enable connecting to a computer over a local area network, you will need to use an alternate method using a USB type-A to Ethernet adapter and an Ethernet (RJ-45) cable.

Procedure

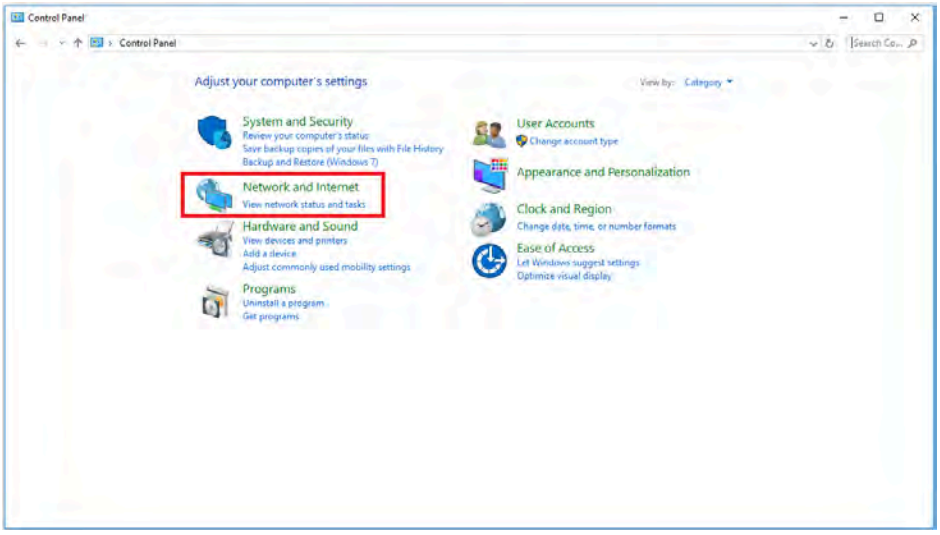
1. Connect one end of the micro-B USB to USB type-A cable to the micro-B USB connector in the robot base.
2. Connect the other end of the cable to a USB type-A port on the computer.

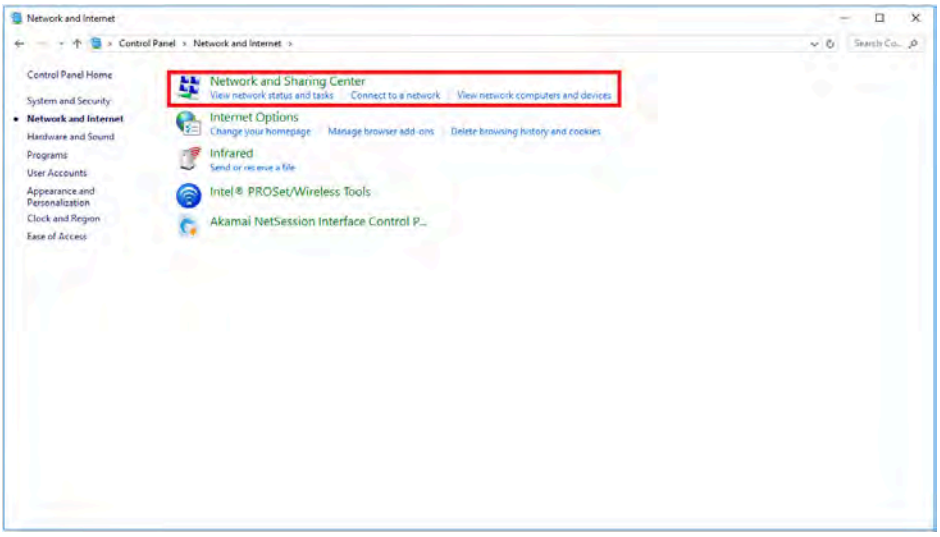


3. If the DHCP server on the base has worked correctly, the computer should automatically be assigned an IP address. In this case no additional configuration should be necessary. To confirm that this worked correctly, try to connect to the robot via the *Web App* interface. If this works, you are done. If not, you will need to manually configure the computer RNDIS adapter settings using the following steps:

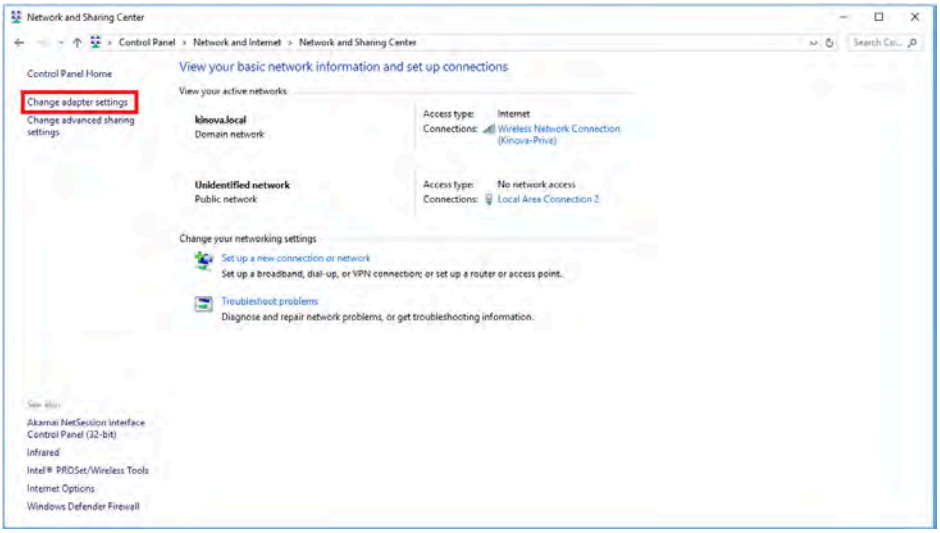
© Kinova inc 2021. All rights reserved.

4. On your computer, in Windows, open **Control Panel** > **Network and Internet** > **Network and Sharing Center**





5. Select **Change adapter settings**



6. Select the RNDIS adapter and choose **Properties.**

© Kinova inc 2021. All rights reserved.

7. Select **Internet Protocol Version 4 (TCP/IPv4)** and choose **Properties**.
8. Select **Use the following IP address** and enter IPv4 address: `192.168.1.11` and Subnet mask: `255.255.255.0`
9. Press OK.

Results

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

Connecting to a computer using a USB type-A to Ethernet adapter and Ethernet cable (Optional)

Describes how to connect the robot to a computer using a USB type-A to Ethernet adapter and Ethernet cable.

Before you begin

You will need:

- USB type-A to Ethernet adapter (optional item for robot packages)
- Ethernet (RJ-45) cable

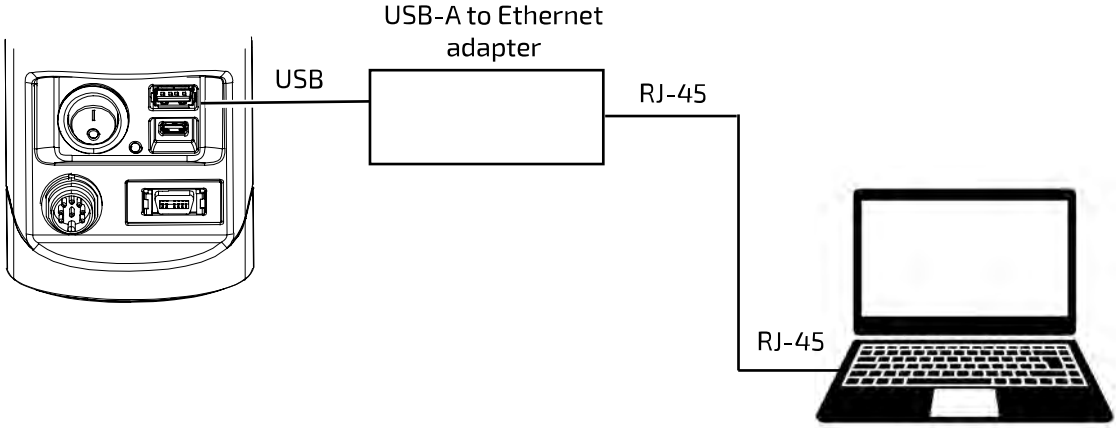
About this task

The default way to establish a communication link between the robot and a computer for communication and control is via a USB cable connection. (over micro-B USB to USB type-A cable using RNDIS / Ethernet over USB). However if you want to be able to connect a computer to the robot over a local area network, it will be necessary to use this alternate method.

Note: The description and illustrations here are for Windows 10. The steps may follow a different order or interface for other operating systems.

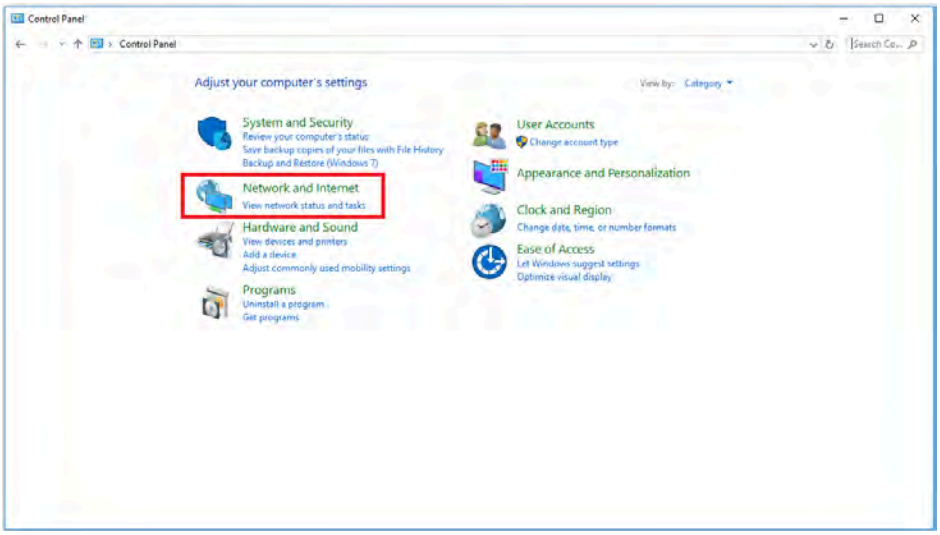
Procedure

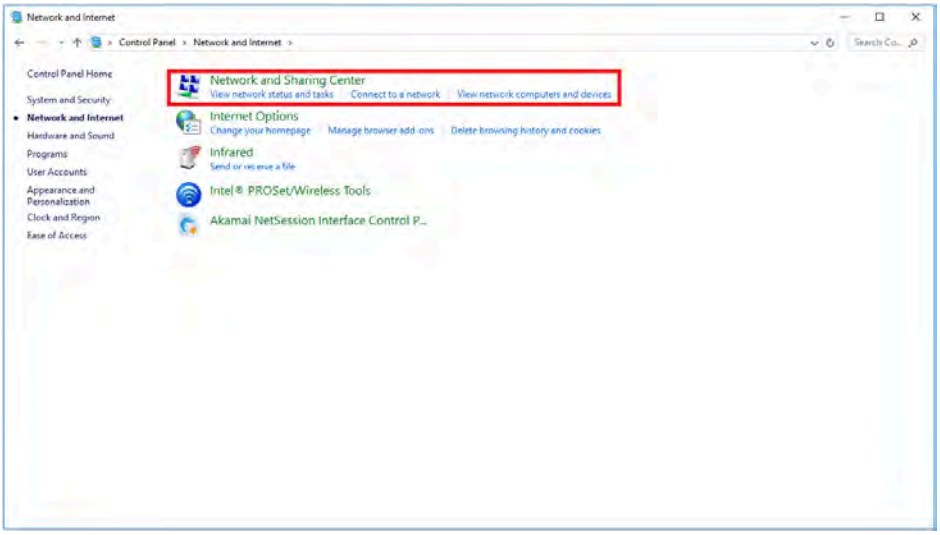
1. Connect the USB type-A to Ethernet adapter to the USB type-A port in the robot base. Connect an RJ-45 Ethernet cable from your computer's wired network adapter to the adapter Ethernet port.



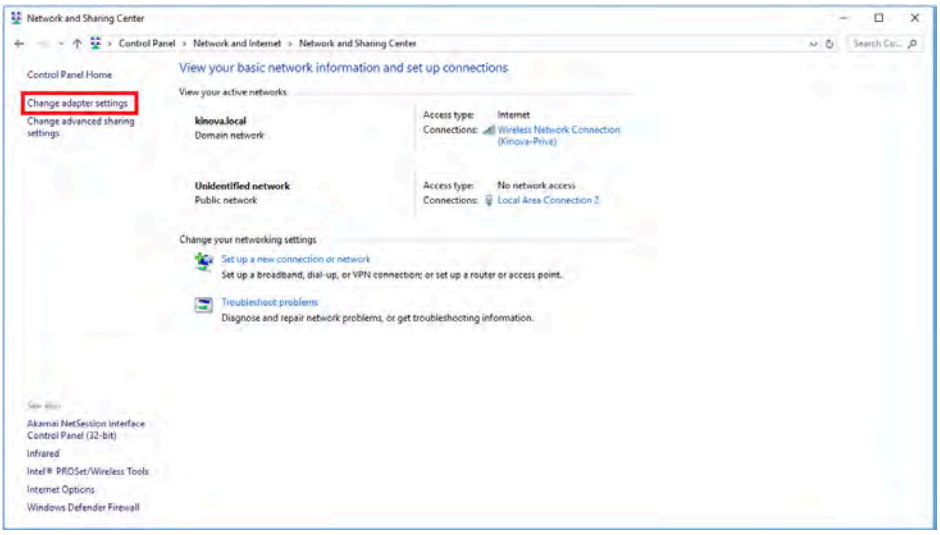
© Kinova inc 2021. All rights reserved.

2. On your computer, open **Control Panel** > **Network and Internet** > **Network and Sharing Center**





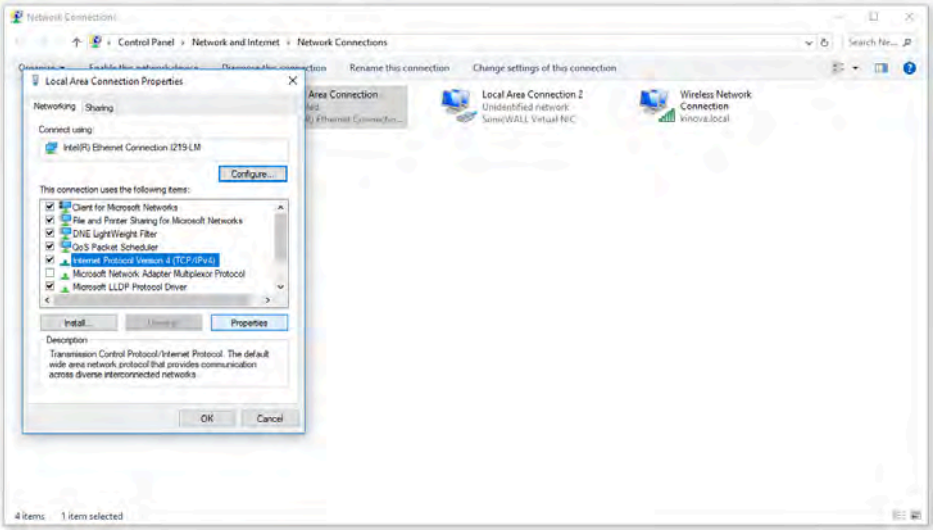
3. Select **Change adapter settings**



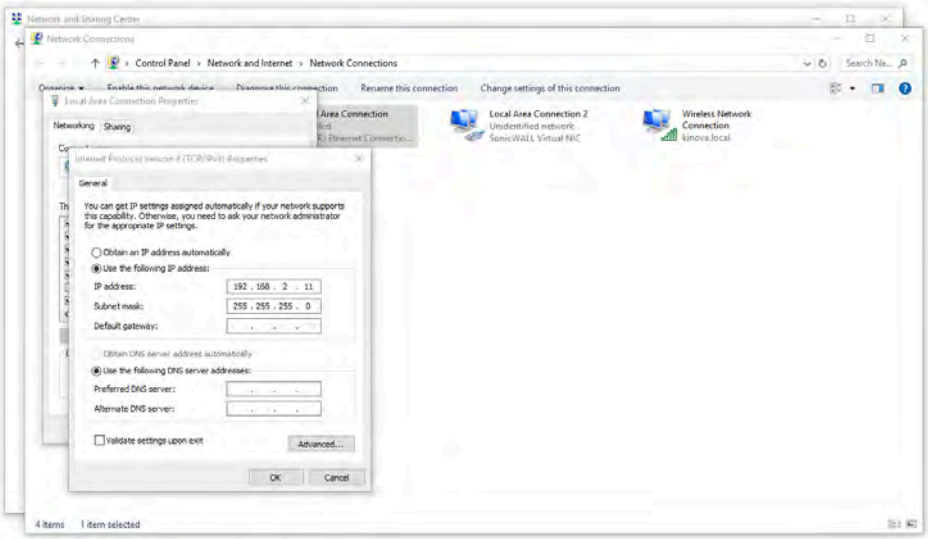
4. Select wired Ethernet adapter (i.e. **Local Area Connection**) and choose **Properties**.

© Kinova inc 2021. All rights reserved.

5. Select **Internet Protocol Version 4 (TCP/IPv4)** and choose **Properties**.



6. Select **Use the following IP address** and enter IPv4 address: `192.168.2.11` and Subnet mask: `255.255.255.0`



7. Press OK.

Results

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

What to do next

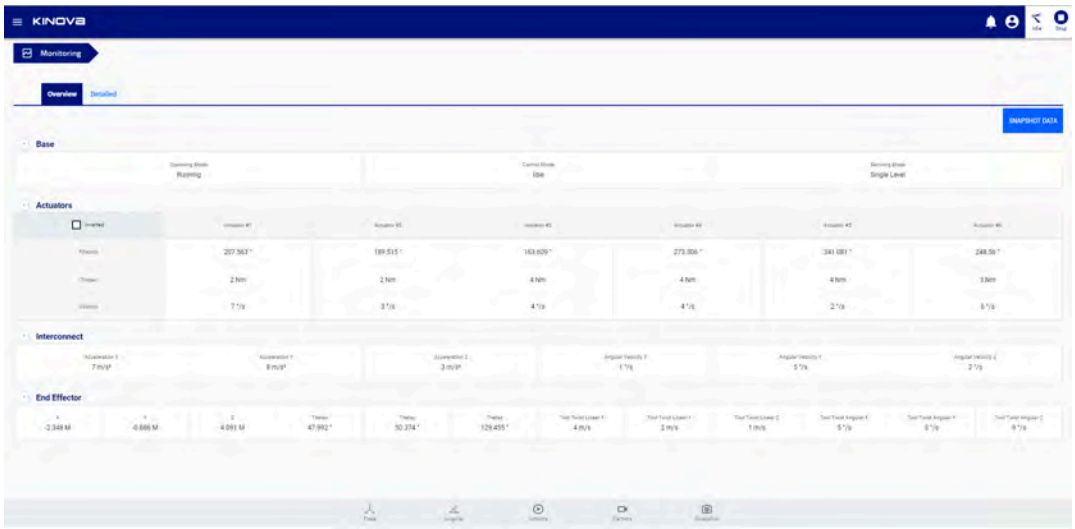
You are now ready to connect to the robot from the computer using the *Web App*.

KINOVA® KORTEX™ *Web App*

This section is an overview of the *Web App*. The *Web App* allows you to interact with the robot and perform basic tasks wuthout using programming commands.

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

© Kinova inc 2021. All rights reserved.



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
- 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 robot firmware

The *Web App* can be run from a desktop or laptop PC connected by wired connection to the robot, or from any device on the same local network. Devices include desktops, laptops, smartphones, and tablets. The local network type includes local Wi-Fi neworks.

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

© Kinova inc 2021. All rights reserved.

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 connecting point to point with USB cable over RNDIS
   - `192.168.2.10` if connecting over Ethernet cable using USB-type A to Ethernet adapter

   If you have configured the robot with a different IP address so that the computer and robot are both connected to the same local area network, whether wired or over Wi-Fi, use the newly 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.

Connecting a computer to the robot via Wi-Fi

This section describes how 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 prior to carrying out this procedure. The Wi-Fi router will need to be set to broadcast its SSID so that the robot can find the network. The router will also need to be configured to accept new devices. You will also need the Wi-Fi password.

About this task

The robot features an integrated 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 using the IP address assigned for the robot on the wireless network.

Procedure

1. On a computer connected to the robot via wired Ethernet, open the *Web App* and connect to the robot.
2. Under the Configurations page group, select Wireless & Networks in the main navigation panel of the *Web App* to go to the Wireless & Networks page.
3. Once on the page, select the Wi-Fi tab.

© Kinova inc 2021. All rights reserved.

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.

   Note: A Connect Automatically toggle is available here. Toggling this on will allow the robot base to automatically connect to the wireless network using the previously entered password whenever the robot is powered on.
6. Take note of the wireless network IPv4 address that the robot obtains after clicking the CONNECT button.
7. On any wireless device (laptop, tablet, smartphone) connected to the same Wi-Fi network, open a Web browser and type the IP address that the robot obtained at Step 6 (This address corresponds to the robot's address on the Wi-Fi network).
8. The *Web App* login screen will appear. At the login screen, enter the appropriate user name and password, and click the CONNECT button.

Results

You are now connected to the *Web App* through the Wi-Fi network adapter of the robot. You can now configure, monitor, and control the robot wirelessly via either the *Web App* or via API control.

Note: 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.

© Kinova inc 2021. All rights reserved.

Dimensions, specifications, and capabilities

Schematic and dimensions

This section provides a schematic diagram of the robot and the key physical dimensions of the robot.

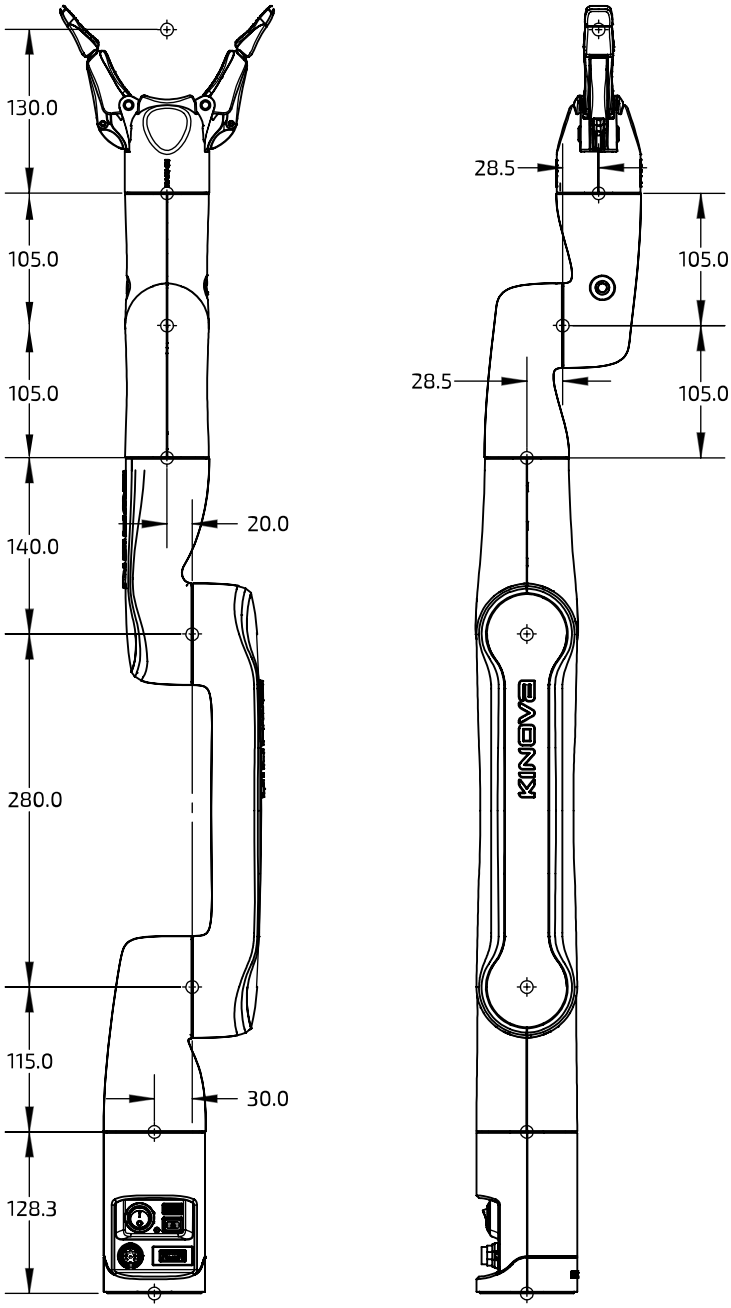


Figure 20: Robot dimensions (measurements in mm)

The image above shows the dimensions of the robot.

The maximum reach of the robot, as defined by the distance from the shoulder (Actuator 2 frame) to the middle of the gripper, is 76.0 cm.

© Kinova inc 2021. All rights reserved.

Table 12: Robot geometric parameters

Description	Length (mm)
Base to actuator 1	128.3
Base to shoulder	243.3
Upper arm length	280.0
Forearm length (elbow to wrist)	140.0
First wrist length	105.0
Second wrist length	105.0
Last actuator to gripper center	130.0
Joint 1-2 offset	30.0
Joint 2-3 offset	0.0
Joint 3-4 offset	20.0
Joint 4-5 offset	28.5
Joint 5-6 offset	28.5

Technical Specifications

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

Table 13: Safety / Security

Feature	Detail
Power monitoring	current and voltage monitoring
Position monitoring	user defined protection zones, joint limits
Thermal monitoring	warning / shutdown near maximum core and motor temperatures

Table 14: Environmental

Parameter	Value(s)
Temperature	-25°C to 40°C (operating)
	0°C to 40°C (startup)
Robot ingress protection	IP22
Relative humidity (non-condensing)	15 to 90%
Pressure	70 to 106.0 kPa
Sound pressure level (nominal)	< 60 dBa
Power supply input voltage	90 - 264 VAC

© Kinova inc 2021. All rights reserved.

Table 15: Controller (base)

Feature	Detail
LED indicator	blue/green/amber/red LED
USB type-A receptacle	For charging small devices such as phones, connecting gamepad wired, connecting USB-A to Ethernet adapter
micro-B USB receptacle	Ethernet over USB (RNDIS)
Wi-Fi (IEEE 802.11a/b/g/n)	KINOVA® KORTEX™ *Web App*
14-pin joystick / expansion connector	joystick*, discrete I/O*, emergency stop, expansion
circular connector [Lumberg 0317 08]	power
piezo-electric buzzer	configurable for audio warnings and alarms*
sensors	gyroscope and accelerometer
Web server	to support *Web App*
DHCP server	simplified (no configuration) connectivity with computer

Table 16: Robot

Parameter	Value(s)
Weight	5.4 kg
Payload	600 g (mid-range, continuous)
	500 g (full-range, continuous)
Maximum reach (fully extended)	760 mm (to gripper center reference)
Maximum Cartesian translation speed	low-level - 30 cm/s
	high-level - 25 cm/s
Degrees of freedom	6
Actuators	small - actuators 4, 5, 6
	large - actuator 2
	medium - actuators 1, 3
Power supply voltage	18 to 30 VDC (24 VDC nominal)
Robot power usage	20 W (average)
	100 W (peak)
Materials	Aluminum
	Polycarbonate, reinforced nylon
Internal communciations and control	Ethernet (100 Mbps)

© Kinova inc 2021. All rights reserved.

Table 17: Actuators

Feature	Value(s)
Sensors	Position, current (motor), voltage, temperature (motor and CPU)

Table 18: Wrist and gripper

Feature	Function
Gripper	2 finger gripper actuated by one linear actuator
Sensors	accelerometer and gyroscope, voltage (motor and microcontroller), temperature (motor and micontroller), motor current, finger position and velocity

Table 19: Software / control

Feature	Detail
Low-level control	position, velocity, current
High-level control	Cartesian twist (linear and angular velocity), joint speed, Cartesian and angular trajectory
High-level control features (mode dependent)	protection zones - rectangular, spheric, cylindrical
	singularity handling
Servoing modes	High level, low-level
Supported ROS distribution	Kinetic Kame, Melodic Morenia
Boot time	30 seconds
Internal communication frequency	1 kHz

Sensors

The robot contains a number of different sensors. There are sensors in the base, actuators, and wrist and gripper. Sensors data can be accessed using the cyclic communications with the base.

The robot components contain the following sensors:

Base sensors

- voltage
- current
- temperature
- 6-axis accelerometer / gyroscope

Actuator sensors

- motor current
- motor temperature
- CPU temperature
- Angular Hall sensors
- magnetic angular sensor

© Kinova inc 2021. All rights reserved.

- 24 V monitor

Wrist and gripper sensors

- capacitive sensors (qty. 2)
- microcontroller temperature
- microcontroller board voltage
- 6-axis accelerometer / gyroscope
- gripper motor current
- gripper motor temperature
- gripper motor voltage
- gripper finger position and velocity (from gripper motor)

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 returns a data structure with readings from sensors in:

- base
- actuators
- wrist and gripper
  - wrist
  - gripper pose and twist
  - gripper finger motors

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

Base sensor readings available via cyclic communications.

Table 20: Base readings available through API

Field name	Description
`arm_voltage`	arm voltage in V
`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

© Kinova inc 2021. All rights reserved.

Field name	Description
`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

Actuators sensors readings available via cyclic communications.

Table 21: Actuators readings available via API

Field name	Description
`position`	angular position of the actuator in °
`velocity`	angular velocity of the actuator in ° / s
`torque`	Approximate equivalent torque derived from motor current in N·m
`current_motor`	motor current in A
`voltage`	main board voltage in V
`temperature_motor`	actuator motor temperature in °C (one of the three phases)
`temperature_core`	microcontroller temperature in °C

Wrist readings available

Wrist sensors readings available via cyclic communications.

Table 22: Wrist readings available via API

Field name	Description
`imu_acceleration_x`	IMU Measured acceleration (X-Axis) of the gripper. (m / s^2)
`imu_acceleration_y`	IMU Measured acceleration (Y-Axis) of the gripper. (m / s^2)
`imu_acceleration_z`	IMU Measured acceleration (Z-Axis) of the gripper. (m / s^2)
`imu_angular_velocity_x`	IMU Measured angular velocity (X-Axis) of the gripper. (° / s)
`imu_angular_velocity_y`	IMU Measured angular velocity (Y-Axis) of the gripper. (° / s)
`imu_angular_velocity_z`	IMU Measured angular velocity (Z-Axis) of the gripper. (° / s)

© Kinova inc 2021. All rights reserved.

Field name	Description
`voltage`	Voltage of the main board in V
`temperature_core`	Microcontroller temperature. (°C)

Gripper pose and twist readings available

Gripper pose and twist sensors readings available via cyclic communications.

Table 23: Gripper readings available via API

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

Gripper finger motor readings available

Gripper finger motor sensors readings available via cyclic communications.

Table 24: Finger readings available via API

Field name	Description
`position`	Position of the gripper fingers in percentage (0-100%)

© Kinova inc 2021. All rights reserved.

Field name	Description
`velocity`	Velocity of the gripper fingers in percentage (0-100%)
`current_motor`	Current consumed by the gripper motor (mA)
`voltage`	Motor Voltage (V)
`temperature_motor`	Motor temperature. (° / s)

Effective workspace

This section provides information on the effective workspace. The effective workspace is the region in space reachable by the gripper 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.

© Kinova inc 2021. All rights reserved.

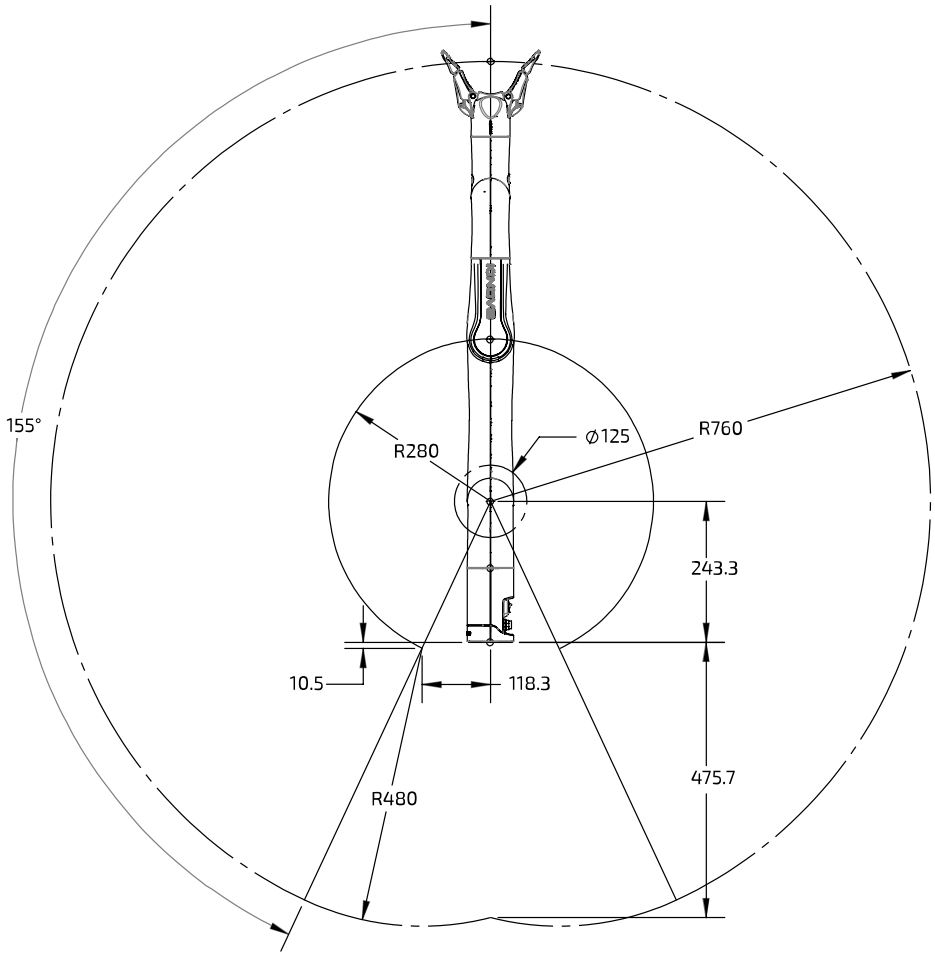


Figure 21: Robot nominal workspace (measurements in mm)

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.

Examples of acceptable payloads include:

- 500 mL bottle of water
- cup
- a tool (screwdriver, fork, spoon, etc)

© Kinova inc 2021. All rights reserved.

- a phone

If the maximum payload is exceeded, any of the following symptoms may appear:

- arm power cut and drop
- arm low voltage
- actuator following error
- actuator fault (motor driver fault)
- Hall sequence error

© Kinova inc 2021. All rights reserved.

Concepts and terminology

Robot key concepts

Actions

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

- reaching an end effector pose or set of joint angles
- changing a position or motion parameter
- applying emergency stop or clear faults
- adding a delay
- sending a gripper command

The full set of action types is defined in the `Kinova.Api.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
- Cartesian joystick
- Cartesian trajectory

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.

Map

A map is a set of associations between controller device inputs and actions to be triggered by those inputs when the map is active.

Mappings

A mapping is a full definition of the possible correspondances between controller device inputs and actions that are triggered by those inputs when the mapping is active. A mapping can consist of multiple **maps**, for example to enable multiple different modes on the same controller device.

Notifications

A notification is a log of an **event** related to a particular topic that happens while a user is using the robot.

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

© Kinova inc 2021. All rights reserved.

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** executed one after the other. A sequence can be started, paused, resumed, and stopped.

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) control the robot by sending a single command to the base. The base manages the low-level details of executing the command, breaking it down, applying any relevant high-level protections, and routing commands to the desired devices via a 1 kHz communication loop with the devices.
  - single-level - a single user sends commands to a base
- low-level servoing - the user controls the robot by sending a series of actuator commands to the base via a user-controlled loop. The base routes these commands to the desired device via its own 1 kHz communication loop with the actuators.

Snapshot

A snapshot is a capture of the instantaneous position of the robot. This can be of different types:

- Cartesian position
- joint angles
- gripper position
- combined robot position and gripper position

© Kinova inc 2021. All rights reserved.

Topic

A set of related robot **events** to which the user can subscribe and receive notifications as part of a Publisher-Subscriber (pub-sub) arrangement. There are a number of different topics, including:

- user
- controller device input
- safety
- action or sequence
- connection / disconnection of arm, controller, or tool
- configuration change or backup
- factory restore
- protection zone
- control, operation, or servoing mode

Upgrade package

An upgrade package is an image containing firmware updates for all modules on the robot (base, actuator)

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:

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

General mathematics and robotics

Axis

A fixed line with direction and units. It is used 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.

© Kinova inc 2021. All rights reserved.

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

Center of Mass
Unique point of a rigid body where an applied force will generate only linear acceleration (and no angular acceleration)

Closed loop control
Control of a device where the device is controlled in relation to the different between sensed current state and goal state. Used on the robot for control of actuators.

Coordinate System
A system used to represent a position in three-dimensional space, consisting of three coordinate axes and an origin. The term "frame" is also used to designate a coordinate system.

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 and three for orientation.

Endpoint
The nominal commanded position that the robot will try to reach with the tool center point at the end of a motion path.

Euler Angle
Describes the rotation in three dimensions of a rigid body in terms of a sequence of three rotations with respect to a coordinate system.

Gravity Vector
Vector representing direction and magnitude of the local force of gravity, expressed in terms of the robot base frame.

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 orientation of a rigid body describes the pure rotations that should be applied to the body to move it from a reference placement to its current placement.

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.

© Kinova inc 2021. All rights reserved.

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.

Payload
The object that is carried or manipulated by the robot tool.

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.

Tool Configuration
Configurations made to enable use of a tool at the end effector position of the robot. This consists of the mass and center of mass of the tool, and the tool transform.

Tool Frame
A coordinate system attached to the end effector tool.

Tool Transform
Transformation (translation and orientation) between the interface module reference frame and the reference frame of the tool attached to the end of the robot.

Trajectory
A time-parametrized path in the robot workspace that can be defined by the user.

Twist
Generalized velocity vector, which is a combination of translational velocity and rotational velocity. Term comes from Screw Theory.

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

Features, components and functionalities

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

End Effector
The device at the end of a robot, designed to directly interact with the environment. Also referred to as tool.

© Kinova inc 2021. All rights reserved.

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

Path Planning
Computation of a path to reach a goal pose subject to applicable constraints and criteria.

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 (or its accessories) which poses a risk of injury to fingers or other appendages close by.

Protection Zone
A volume in space where the motion of the robot can be limited for safety purposes.

Safeties
Hardware limitations which are monitored 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
See End Effector.

Control Modes

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

Cartesian Mode
Mode used to control the velocities (translation and orientation) of the tool in Cartesian space.

Trajectory Mode
Mode allowing user to specify an endpoint (in joint space or Cartesian space) that the robot should reach.

© Kinova inc 2021. All rights reserved.

Robot control

High-level and low-level robot control

The robot can be controlled in either high-level or low-level robot control. There are benefits to both forms of control. High-level control is easier with more protections, while low-level control offers faster commands and finer-grained control with less protections.

High-level is the default on boot-up and offers the safest and most straight-forward control.

In both high-level and low-level, commands are sent through the robot base.

In high-level control, commands are sent to the base via a single command using the `Kinova.Api.Base` API. These commands are processed by Kinova robot control libraries.

The robot control library applies high-level control features like:

- singularity avoidance
- protection zones
- Cartesian and joint limits

The control library also breaks down the command into smaller commands that the base will send incrementally to the individual robot actuators via its 1 kHz communication loop with the actuators.

In low-level control, the user sends a series of small, incremental commands to the actuator and gripper as part of a user-defined loop (at a rate up to 1 kHz) using the `Kinova.Api.BaseCyclic` API. The base receives these commands and routes them to the appropriate actuators and the gripper via its own 1 KHz communication loop with robot devices.

High-level control is simpler to use and offers added protections. However, it is slower due to the overhead of processing by the Kinova control library. High-level control runs at 40 Hz.

Low-level control offers lighter and faster commands and finer-grained control, but the user is responsible for more of the low-level details without the protections offered by the high-level control libraries.

For more information, see the KINOVA® KORTEX™ Developer Guide section of the User Guide.

High-level and low-level robot control methods reference

Reference for key methods available for controlling robot movement through the API.

High-level (`Kinova.Api.Base`)

Table 25: Send or play a trajectory (Cartesian or joint)

Method	Description
`PlayCartesianTrajectory (ConstrainedPose)`	Moves to the specifed pose (with specified Cartesian constraint on trajectory) **Note:** This method will be deprecated in a future software version.

© Kinova inc 2021. All rights reserved.

Method	Description
`PlayCartesianTrajectoryPosition (ConstrainedPosition)`	Moves to the specifed position (with specified Cartesian constraint on trajectory) **Note:** This method will be deprecated in a future software version.
`PlayCartesianTrajectoryOrientation (ConstrainedOrientation)`	Moves to the specifed orientation (with specified Cartesian constraint on trajectory) **Note:** This method will be deprecated in a future software version.
`PlayJointTrajectory (ConstrainedJointAngles)`	Moves to the specified joint angles (with specified joint constraint on trajectory) **Note:** This method will be deprecated in a future software version.
`PlaySelectedJointTrajectory (ConstrainedJointAngle)`	Moves specifed joint to the specifed joint angle (with specified joint constraint on trajectory). **Note:** This method will be deprecated in a future software version.
`PlayPreComputedJointTrajectory (PreComputedJointTrajectory)`	Play pre-computed angular trajectory

Table 26: Send Cartesian command to tool

Method	Description
`SendTwistCommand (TwistCommand)`	Sends a twist command to tool (velocity and angular velocity)
`SendTwistJoystickCommand (TwistCommand)`	Sends a twist joystick command to tool. The twist values sent to this call are expected to be a ratio of maximum value (between -1.0/+1.0)

Table 27: Send command to joints

Method	Description
`SendJointSpeedsCommand (JointSpeeds)`	Sends a joint speeds command, that is the desired speed of one or many joints
`SendSelectedJointSpeedCommand (JointSpeed)`	Sends a speed command for a specific joint
`SendJointSpeedsJoystickCommand (JointSpeeds)`	Sends the desired joystick speeds for one or multiple joints. Values sent to this call are expected to be a ratio of maximum value (between -1.0/+1.0)
`SendSelectedJointSpeedJoystickCommand (JointSpeed)`	Sends a joystick speed for a specific joint. Value sent to this call is expected to be a ratio of maximum value (between -1.0/+1.0)

© Kinova inc 2021. All rights reserved.

Table 28: Send gripper commands

Method	Description
`SendGripperCommand (GripperCommand)`	Sends a command to move the gripper. Commands the fingers of the gripper in either position or velocity.

Low-level (Kinova.Api.BaseCyclic)

Table 29: Low-level cyclic commands

Method	Description
`Refresh (Command)`	Send a new incremental refresh command to the actuators and gripper for the current 1 ms interval. Command actuators position, angular velocity, and motor current, as well as gripper finger motors. Receive feedback on current status of base, actuators, tool, and gripper.
`RefreshCommand (Command)`	Send a new incremental refresh command to the actuators and gripper for the current 1 ms interval. Command actuators position, angular velocity, and motor current, as well as gripper finger motors. No feedback provided.
`RefreshFeedback ()`	Receive feedback on current status of base, actuators, tool, and gripper.

Control features

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
- angular limits
- Cartesian limits

Singularity avoidance

The singularity avoidance feature of the robot control library handles or avoids singularities in Cartesian control modes. The robot behavior, including tool speed may be altered near a singularity.

A singularity refers to any robot configuration (set of joint angles) which causes the Jacobian transformation matrix relating actuator rotation speed to end effector velocities to be ill-conditioned, thus rendering the solution mathematically unstable (determinant of the Jacobian matrix loses rank).

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 become colinear, leading to unpredictable / extreme joint velocities when trying to attain a certain Cartesian pose. For example, when two axes become colinear in space, rotation of one can be canceled by counter-rotation of the other, leaving the actual joint location

© Kinova inc 2021. All rights reserved.

indeterminate. Near a singularity a small linear end effector motion requires disproportionately large angular velocities of the actuators.

Note: 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.

For more information on robot singularity configurations, see here.

Protection zones

Users can define geometric protection zones where the movement of checkpoints on the robot is either prevented or slowed. These can be defined using either the *Web App* or developer API.

Overview

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 speed limitation in the **envelope** of defined thickness surrounding each protection zone.

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.

Note: Protection zones are active only in **Cartesian** control modes. These protections are not available in **angular** control modes.

Note: If multiple protection zones are used, we recommend that the same envelope speed limitations be used for each.

Note: There is no hard limit on the number of protection zones that can be simultaneously active, but for best results it is recommended to activate no more than four protection zones at one time.

Note: Envelope-only protection zones, where all the dimensions of the protection zone are set to zero but the envelope thickness is set to a non-zero value are NOT supported.

Robot tool behavior

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

The tool can move within the surrounding envelope, but at a reduced speed.

Checkpoints and behavior of checkpoints

Additional checkpoints are used for protection zones and are defined for the robot at the centers of actuators 3-6 and near the center of actuator 2.

For these checkpoints, the motion will stop at the outer surface of the protection zone. Checkpoints will move within the envelope surrounding a protection zone, but at a reduced speed.

Protection zone shapes

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

© Kinova inc 2021. All rights reserved.

- 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.

Editing protection zones

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

Joint limits

The robot has joint limits used in robot high-level control. This impacts the position, speed, acceleration, and torque applied for the joints.

Overview

When controlling the robot in high-level, the robot control library applies a number of different joint limits for safety purposes. This includes limits on:

- joint position
- joint speed
- joint acceleration
- joint torques

The limits applied in a particular situation depend on the current high-level control mode.

Gen3 lite robot joint limits

Table 30: Joint position limits

Actuator	Joint limits	
	Lower limit	Upper limit
1	- 154.1°	+ 154.1°
2	- 150.1°	+ 150.1°
3	- 150.1°	+ 150.1°
4	- 148.98°	+ 148.98°
5	- 144.97°	+ 145.0°
6	- 148.98°	+ 148.98°

Table 31: Joint speed limits

Actuator	Limit (magnitude)
joints 1-5	57.3 ° / s (1.0 rad / s)
joint 6	90.0 ° / s (1.57 rad / s)

© Kinova inc 2021. All rights reserved.

Table 32: Joint acceleration hard limits

Actuator	Limit (magnitude)
joint 1	57.3 ° / s^2 (1.0 rad / s^2)
joint 2	28.65 ° / s^2 (0.5 rad / s^2)
joint 3	22.92 ° / s^2 (0.5 rad / s^2)
joint 4	57.3 ° / s^2 (1.0 rad / s^2)
joints 5 - 6	572.95 ° / s^2 (10.0 rad / s^2)

Table 33: Joint torques soft limits

Actuator	Limit (magnitude)
joints 1 - 6	9.0 N * m

Control modes and relevant joint limits

Control mode	Joint limits applied
Cartesian trajectory	joint position, joint speed
angular trajectory	joint position, joint speed, joint acceleration, joint torque (for pre-computed trajectories)
Cartesian joystick	joint position, joint speed
angular joystick	joint position, joint speed, joint acceleration

Behavior at joint limits

When joint limits are reached, the behavior of the robot will be altered depending on the type of limitation.

Limit	Behavior when limit is reached
joint acceleration	joint acceleration will max out at acceleration limit
joint speed	acceleration go to zero and joint speed will max out at speed limit
joint position	joint acceleratation and joint speed goes to zero and motion of the joint will stop at the position limit

Cartesian limits

The robot has limits on tool motion applying in high-level Cartesian control. These limits apply on top of underlying joint limits.

Overview

Cartesian limits on the motion of the tool in Cartesian modes (these limits do not apply in angular modes) are applied as follows:

- In Cartesian joystick, the magnitude (vector norm) of the linear and rotational speeds of the tool are capped. Twist commands whose linear and / or rotation speed exceed these limits will be re-

© Kinova inc 2021. All rights reserved.

scaled proportionally so that the **magnitudes** of tool linear and angular speeds will not exceed the limits, but the commanded **directions** of tool movement will be respected.

- For all Cartesian modes, the individual joint speeds will be capped at their limits. The speed of all joints will be proportionally scaled so that:
  - no individual joint exceeds the speed limit
  - the desired direction of motion is respected
- For all Cartesian modes, joint positions will be capped at their limits.

Gen3 lite robot Cartesian limitations

Limit		Value
twist limits	linear	0.25 m / s
	angular	45.8 ° / s (0.80 rad / s)

High-level control modes description

Overview of the high-level control modes of the robot.

The robot is controllable via a number of high-level control modes:

- trajectory modes
  - angular trajectory
  - Cartesian trajectory
- joystick control modes
  - Cartesian joystick
  - angular joystick

Trajectory control modes

Trajectories define a path of the robot through space, either Cartesian or joint, over a time period. There are different ways available to define trajectories.

Using trajectory control modes, the user can command the robot to a desired endpoint.

There are a few different ways a trajectory can be defined.

Reach endpoint

Users can provide an **endpoint** to reach, and the let the control software of the robot compute how to follow a path to reach the endpoint. The robot controller computes an interpolated trajectory (between current pose and target pose) to reach the final position while satisfying configured limits, and commands the robot to follow this trajectory. These trajectories, once defined, can be played back. This is a good setting when you want the robot to go to a desired end state but you are not overly concerned about the exact path it follows getting there and exactly how long it takes.

Pre-computed trajectory

Users can supply a **pre-computed trajectory** to the robot. A pre-computed trajectory is generally auto-generated by some sort of path planning software or algorithm rather than built manually. A pre-computed trajectory is defined as a time series of settings for joints angular positions, velocities, and accelerations at each timestamp. The robot control software will verify that the trajectory is valid and reasonable, satisfying configured limits. Users can indicate a desired continuity mode for the trajectory against which the trajectory can be checked (position, or position and velocity, or position, velocity and acceleration).

© Kinova inc 2021. All rights reserved.

The robot control library will perform the following validations on the pre-computed trajectory:

- trajectory is non-empty
- for each trajectory point, position, velocity, and acceleration values must be provided for each joint in the robot
- trajectory contains no `NaN` values
- timestamp of the first trajectory point must be 0.000 seconds and the difference in time stamps between successive points must be 0.001 seconds
- joint positions, speeds, accelerations, and torques must be within robot joint limits
- continuity - the trajectory is continuous in terms of (user can specify which of these continuity checks to apply):
  - position - joint position variation between successive timesteps is less than the maximum variation (based on joint speed limit)
  - speed - speed values are consistent (within tolerances) with derivative of position
  - acceleration - acceleration values are consistent (within tolerances) with derivative of speed
- trajectory execution (when starting the trajectory) - the joint positions and speeds for the first point in the trajectory must match the initial robot joint positions and speeds

Note: For safety reasons, trajectories failing any of the validation checks will be rejected, and you will receive an error notification. To get more detailed information about why a particular trajectory failed, use the method `GetTrajectoryErrorReport()` in `Kinova.Api.Base`.

Trajectory modes

In **Cartesian Trajectory** mode the positions are defined in terms of the desired Cartesian space pose of the tool frame. This mode enables singularity avoidance. Cartesian trajectory mode can be activated by sending a pose endpoint. Cartesian Trajectory mode is not compatible with pre-computed trajectories.

In **Angular Trajectory** mode the positions are defined in terms of the desired joint angles for the actuators. Angular Trajectory mode can be activated by sending an endpoint joint state or a pre-computed trajectory.

Users have the option to apply constraints to trajectories. There are three options available for constraints:

- duration - the time period (in seconds) in which the trajectory is to be carried out
- speed - the maximum speed (meters / second for Cartesian, degrees / second for angular) of the motion while carrying out the trajectory
- no constraint - the robot will go to the endpoint without the time or speed being specified

Note: If a requested trajectory constraint will cause angular limits to be exceeded in the course of the trajectory, (e.g. the duration is too short and requires a speed or acceleration that is not feasible) the trajectory will be rejected.

In angular trajectories, all three options are available (duration, speed, no constraint).

In Cartesian trajectories, the speed constraint or no constraint options are available.

Trajectory control modes are set by sending a trajectory using the appropriate API methods, or the *Web App* `Actions` page.

Joystick control modes

Joystick control modes allow the robot to be controlled by sending motion (speed) commands by either Cartesian or joint.

© Kinova inc 2021. All rights reserved.

Joystick Control modes provide the user the ability to create a desired motion of the robot by sending commands to the robot. This is done using joystick control inputs (physical gamepad or virtual joystick) or directly using API commands.

In **Cartesian Joystick** mode the motion of the robot end effector, both linear and angular, is controlled. Cartesian Joystick is entered by:

- sending API twist commands (specify linear and angular velocity of the end effector)
- activating Xbox gamepad Twist linear and Twist angular modes
- sending *Web App* Pose virtual joystick control inputs in velocity control

This mode provides for singularity avoidance and obstacle avoidance (protection zones).

For Cartesian control, the reference frames for translation and orientation control are configurable:

Table 34: Cartesian reference frames options

Cartesian frame mode	Reference frame for translation	Reference frame for orientation
mixed mode	base	tool
tool mode	tool	tool
base mode	base	base

Note: Orientations are defined using an **z-y-x Tait-Bryan extrinsic convention**. That is, rotation about the fixed x-axis, followed by rotation about the fixed y-axis, followed by rotation about the fixed z-axis.

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

Angular joystick mode can be entered by:

- sending API joint speed commands
- activating Xbox gamepad Joint mapping
- sending commands via *Web App* Angular virtual joystick

Low-level control detailed description

Overview of low-level control options.

Overview

Low-level control of the robot is carried out through the `Kinova.Api.BaseCyclic` API.

To use the `BaseCyclic API`, a real-time session is configured by the user on the robot to enable high-frequency real-time communications.

Users are responsible for creating a high-frequency command loop (1 kHz) and for decomposing desired robot movements into small command increments for actuators and gripper motors. These command increments are sent sequentially to the robot base at a 1 kHz frequency using this real-time session. The robot base then routes these individual command increments to the actuators and gripper via its own 1 kHz cyclic communications with the robot devices. Each invidual command increment must be executed within the 1 ms time interval.

Low-level commands

© Kinova inc 2021. All rights reserved.

Commands consist of a command for each actuator in the robot and a command for each gripper finger motor.

Actuator commands can be of one of the following types:

- joint position (angular)
- joint speed (angular)
- actuator motor current

Gripper motor commands can be of one of the following types:

- gripper finger position
- gripper finger velocity
- force

Low-level cyclic feedback

The user can simply send a command increment (`RefreshCommand(Command)` method), or also request, in the same command, to receive status feedback (`Refresh(Command)` method) during the interval. This provides feedback on the current status of:

- base
- tool
- actuators
- wrist and gripper

Cautions

Low-level control in general is intended for **advanced users only**.

Note: Low-level control bypasses **completely** the Kinova control library and all its soft limits and protections.

Although the Kinova APIs, including the low-level `BaseCyclic` API are supported in both C++ and Python, note that Kinova strongly recommends low-level control applications be done in C++ for best results.

Further information

For more details on how to implement low-level control, including API documentation and code examples, please see the kortex GitHub repository.

© Kinova inc 2021. All rights reserved.

Configurations and safeties

Configurable parameters

There are a number of parameters of the robot that are configurable to customize the operation of the robot. These can be configured using the API and *Web App*.

These parameters can be configured using the appropriate KINOVA™ KORTEX® 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. Access configurable parameters on one of the available tabs.

The following tables give a summary of the configurable parameters.

Base configuration

Base configurations cover a range of core configurable items. These can be configured via the API or the *Web App*.

Table 35: Base configuration

Configurable item	Description	Where to configure it
User Profiles	Create, read, update and delete user profiles	`Kinova.Api.Base` API Web App > Users
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 origin and orientation • zone dimensions • envelope thickness • zone limitation types (velocity) and values • envelope limitation types (velocity) and values	`Kinova.Api.Base` API Web App > Operations > Protection Zones
Control Mapping	Create, read control mapping	`Kinova.Api.Base` API Web App > Configuration > Controllers
Action	Create, read, update, delete action	`Kinova.Api.Base` API Web App > Operations > Actions

© Kinova inc 2021. All rights reserved.

Configurable item	Description	Where to configure it
Sequence	Create, read, update, delete a sequence of actions	`Kinova.Api.Base` API Web App > Operations > Actions
IPv4	Set IPv4 configured (for specified network adapter): • IP address • subnet mask • default gateway	`Kinova.Api.Base` API Web App > Configurations > Wireless & Networks
Communication Interface	Enable communication interface: • network type (Wi-Fi or Ethernet) • enabled/disabled	`Kinova.Api.Base` API Web App > Configurations > Wireless & Networks
Wi-Fi	Set: • SSID • security key • automatic connection allowed • country code	`Kinova.Api.Base` API Web App > Configurations > Wireless & Networks
Restore factory settings	Delete all configurations and reverts settings to factory defaults (except network settings)	`Kinova.Api.Base` API Web App > Configurations > Robot > Arm
Restore factory product configuration	Restore product configuration to factory product configuration	`Kinova.Api.Base` API

Actuators configuration

Actuators configurations set items related to the functioning of the actuators. Most of these can be configured only via the API. Zeroing of the actuators torque offsets can be done via the *Web App*.

Table 36: Actuator configuration (`Kinova.Api.ActuatorConfig`)

Configurable item	Description	Where to configure it
Axis offsets	Set actuator axis offset position	`Kinova.Api.ActuatorConfig` API
Torque offset	Set actuator torque zero configuration	`Kinova.Api.ActuatorConfig` API Web App > Configurations > Robot > Actuators
Control mode	Set actuator control mode. Options: • position • velocity • current • torque	`Kinova.Api.ActuatorConfig` API

© Kinova inc 2021. All rights reserved.

Configurable item	Description	Where to configure it
Control loop parameters	Configure an individual control loop parameter: • joint or motor position • joint or motor velocity • joint torque • motor current Configure: • error saturation value • output saturation value • kAz • kBz • error dead band value	`Kinova.Api.ActuatorConfig` API
Vector drive parameters	Set vector drive parameters: • kpq • kiq • kpd • kid	`Kinova.Api.ActuatorConfig` API
Encoder derivative parameters	Set encoder derivative parameters: • maximum window width • minimum encoder tick count	`Kinova.Api.ActuatorConfig` API
Command mode	Set command mode. Options: • cyclic - cyclic data only • asynchronous - configuration messages only • cyclic jitter compensation using position or posi tion and velocity inputs **Note:** These options are available in the API but should NOT be used	`Kinova.Api.ActuatorConfig` API
Servoing	Enable servoing	`Kinova.Api.ActuatorConfig` API

Device configuration

Device configurations impact settings for robot devices. These can be configured via the API.

Table 37: Device configuration

Configurable item	Description	Where to configure it
Run mode	Set device run mode (Run, Calibration, Configuration, Debug, Tuning)	`Kinova.Api.DeviceConfig` API

© Kinova inc 2021. All rights reserved.

Configurable item	Description	Where to configure it
IPv4 settings	(For devices other than base) Set device IPv4 address, subnet mask, default gateway	`Kinova.Api.DeviceConfig` API
Safeties	Enable / disable, set warning and/or error threshholds for specific safeties	`Kinova.Api.DeviceConfig` API

Safety items

There are a number of safety items related to different components of the robot. These are viewable in the *Web App* Safeties page.

Overview

Safety items, and their associated warning and error thresholds are active on the robot and are viewable within the Safeties page of the *Web App*. There are several categories of safeties:

- Base (arm) safeties
- Actuators safeties
- Gripper 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 safeties

The following base-related safety items are active on the robot and are viewable in the *Web App*.

Table 38: Base Safety items

Safety Item	Description	Threshold	warning / error
Maximum Core Temperature	CPU temperature is above upper limit. CPU automatically downclocks at 80 °C and shuts down at 115 °C.	75.0 °C	warning
		85.0 °C	error
Minimum Voltage	The base voltage reading is below lower limit	18.0 V	warning
	Note: The minimum voltage must be lower than the maximum voltage	16.0 V	error
Maximum Voltage	The base voltage reading is above upper limit.	30.0 V	warning
	Note: The maximum voltage must be higher than the minimum voltage	31.0 V	error
Inrush current limiter fault	Inrush current limiter fault triggered	n/a	
NVRAM Corrupted	Non-volatile memory corrupt	n/a	

© Kinova inc 2021. All rights reserved.

Safety Item	Description	Threshold	warning / error
Network Initialization Error	Arm is detected but arm internal connectivity not available	n/a	
Joint Detection Error	The number of detected joints does not match the configured arm joint count	n/a	
Network Error	Loss of actuator cyclic data response	n/a	
Joint Fault	At least one joint is in a fault state	n/a	
Emergency Stop Activated	Emergency stop activated	n/a	
Emergency Line Activated	Emergency line activated	n/a	
Discrete Input Stuck Active	Joystick Control indicator (GPI) stuck	n/a	
Incompatible firmware version	Firmware incompatible with hardware board revision	n/a	
Firmware Update Failure	Indicates a failure in the firmware update process	n/a	

Actuators safeties

The following actuator-related safety items are active on the robot and are viewable in the *Web App*.

Table 39: General actuator safety items (apply the same to all actuator sizes)

Safety Item	Description	Hard limit	lower / upper	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°	
Maximum Velocity	The computed velocity of the actuator is greater than threshold °/sec.	0 °/s		180 °/s	
		250 °/s		200 °/s	
Magnetic Position	Position step of more than threshold °/ms has been read on the magnetic sensor.	0°		1.25°	
		2°		1.50°	
Hall Position	Position step of more than threshold °/ms	0°		0.1758	
		1°		0.3516	
Hall Sequence	Invalid Hall sequence detected.	n/a		n/a	

© Kinova inc 2021. All rights reserved.

Safety Item	Description	Hard limit	lower / upper	Threshold	warning / error
Hall magnetic mismatch	Compares the current absolute position (as seen from Hall sensors) versus the current absolute position (as seen from the magnetic sensor). Makes sure the difference between those values are inside the defined tolerances.	0.0°		4.0°	
		10.0°		4.5°	
Maximum motor current	Checks the motor current. Makes sure it is below the defined upper limit.	0 A		11 A	
		12 A		12 A	
Minimum Voltage	The voltage reading is below lower limit. **Note:** The minimum voltage thresholds must be lower than the maximum voltage thresholds.	14.0 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		28.0 V	
		36.0 V		32.0 V	
Maximum Core Temperature	Core temp above upper limit	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	
Watchdog Triggered	Watchdog was triggered	n/a		n/a	

© Kinova inc 2021. All rights reserved.

Table 40: Small actuator specific safeties

Safety item	Description	Hard limit	lower / upper	Threshold	warning / error
Joint Limit Low	Checks if the actuator is within the defined lower position limit.	-154°		-150°	
		0°		-152°	
Joint Limit High	Checks if the actuator is within the defined upper position limit.	0°		150°	
		154°		152°	
Maximum motor temperature	Checks the motor temperature. Makes sure it is below the defined upper limit.	0°		60°	
		80°		70°	

Table 41: Medium actuator specific safeties

Safety item	Description	Hard limit	lower / upper	Threshold	warning / error
Joint Limit Low	Checks if the actuator is within the defined lower position limit.	-159°		-155°	
		0°		-157°	
Joint Limit High	Checks if the actuator is within the defined upper position limit.	0°		155°	
		159°		157°	
Maximum motor temperature	Checks the motor temperature. Makes sure it is below the defined upper limit.	0°		60°	
		90°		75°	

Table 42: Large actuator specific safeties

Safety item	Description	Hard limit	lower / upper	Threshold	warning / error
Joint Limit Low	Checks if the actuator is within the defined lower position limit.	-159°		-155°	
		0°		-157°	
Joint Limit High	Checks if the actuator is within the defined upper position limit.	0°		155°	
		159°		157°	
Maximum motor temperature	Checks the motor temperature. Makes sure it is below the defined upper limit.	0°		60°	
		90°		70°	

© Kinova inc 2021. All rights reserved.

Gripper safeties

The following gripper-related safety items are active on the robot and are viewable in the *Web App*.

Table 43: Gripper Safety items

Safety Item	Description	Hard limit	lower / upper	Threshold	warning / error
Following error	Compares the position command given by the base versus the current command as seen by the actuator. Makes sure the difference between those values are inside the defined tolerances.	0°		8°	
		15°		12°	
Maximum velocity	Checks if the velocity as seen by the actuator is within the defined limits.	0°/s		150°/s	
		200°/s		170°/s	
Joint limit low	Checks if the actuator is within the defined lower position limit.	-5°		-1°	
		0°		-2°	
Joint limit high	Checks if the actuator is within the defined upper position limit.	0°		105°	
		115°		110°	
Hall position	Compares the current absolute position (as seen from Hall sensors) versus the last absolute position (as seen from the Hall sensors). Makes sure the difference between those values are inside the defined tolerances.	0		0.1758	
		1		0.3516	
Hall sequence	Checks if the Hall sequence is valid. This is done evaluating a parameter that does this in the process phase.	n/a		n/a	
Maximum motor current	Checks the motor current. Makes sure it is below the defined upper limit.	0 A		1.2 A	
		1.8 A		1.4 A	
Minimum Voltage	Checks the entry voltage. Makes sure it is above the defined lower limit. **Note:** The minimum voltage must be lower than the maximum voltage	14.0 V		18.0 V	
		24.0 V		16.0 V	

© Kinova inc 2021. All rights reserved.

Safety Item	Description	Hard limit	lower	Threshold	warning
			upper		error
Maximum Voltage	Checks the entry voltage. Makes sure it is below the defined upper limit. **Note:** The maximum voltage must be higher than the minimum voltage	24.0 V		28.0 V	
		36.0 V		32.0 V	
Maximum motor temperature	Checks the motor temperature. Makes sure it is below the defined upper limit.	0 °C		60 °C	
		90 °C		70 °C	
Maximum core temperature	Checks the mcu core temperature. Makes sure it is below the defined upper limit.	0 °C		80 °C	
		100 °C		90 °C	
Non-volatile memory (NVRAM) corrupted	Checks the non-volatile memory used for storage. Makes sure the memory isn't corrupted.	n/a		n/a	
Motor driver fault	Checks if the motor gate driver is in fault. This is done by polling the pin assigned for this purpose on the driver board.	n/a		n/a	
1000 ms		200 ms		n/a	
Watchdog triggered	Checks if the watchdog got triggered.	n/a			

© Kinova inc 2021. All rights reserved.

KINOVA® KORTEX™ Web App User Guide

Introduction

Overview of the *Web App* section of the user guide.

The following sections describe the KINOVA® KORTEX™ *Web App*. The *Web App* is used for controlling, configuring and monitoring the robot.

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

Purpose

The *Web App* provides a convenient Web-based GUI for configuring, controlling, and monitoring the robot.

The *Web App* is a Web 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*

The *Web App* is available on desktop computers, tablets, and smartphones.

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

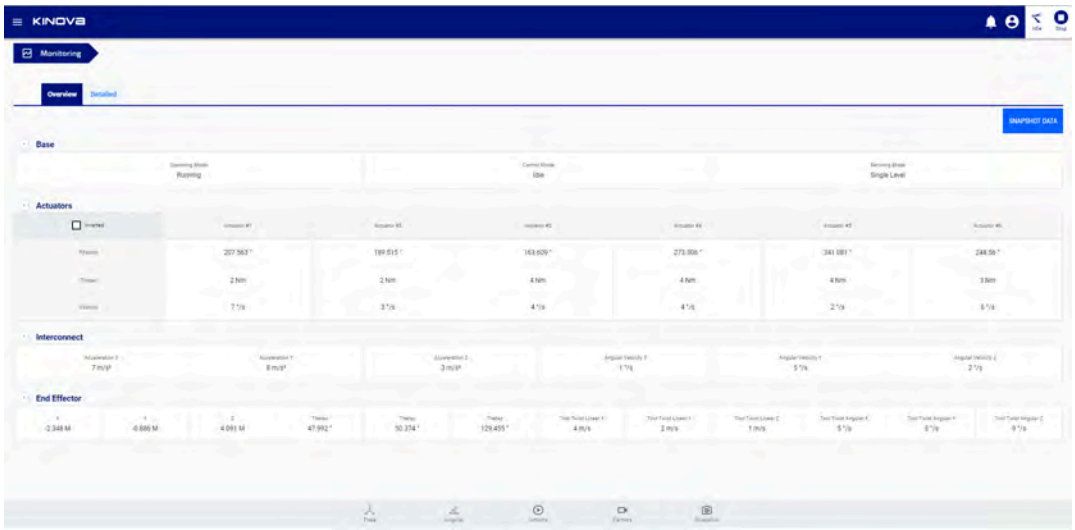


Figure 22: Desktop

© Kinova inc 2021. All rights reserved.

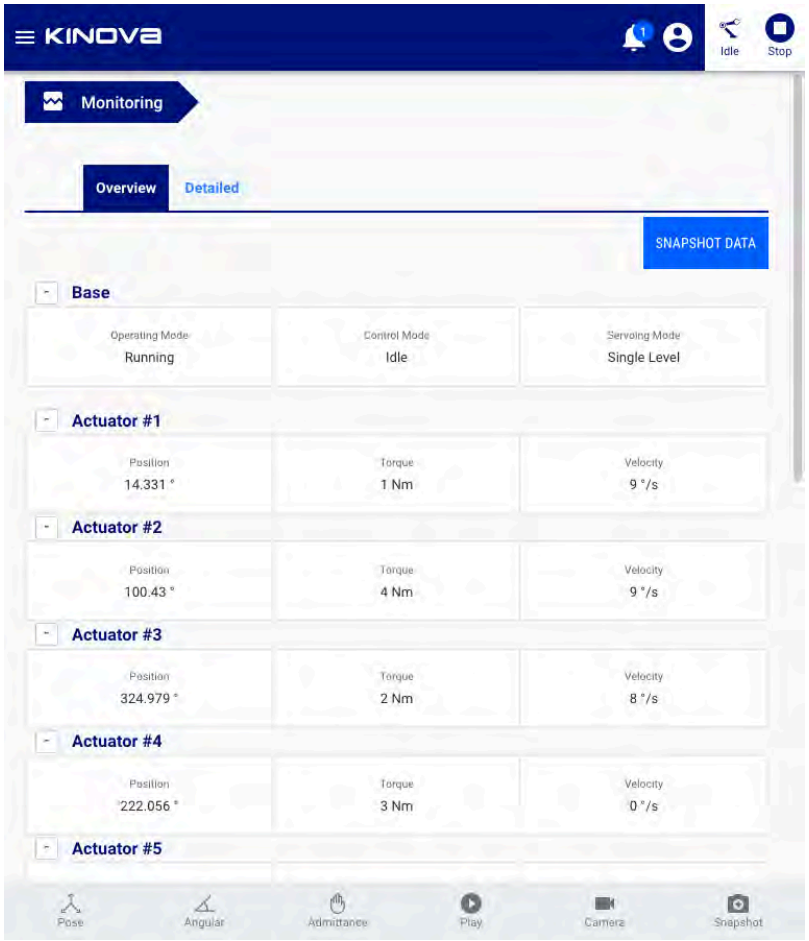


Figure 23: Tablet

© Kinova inc 2021. All rights reserved.

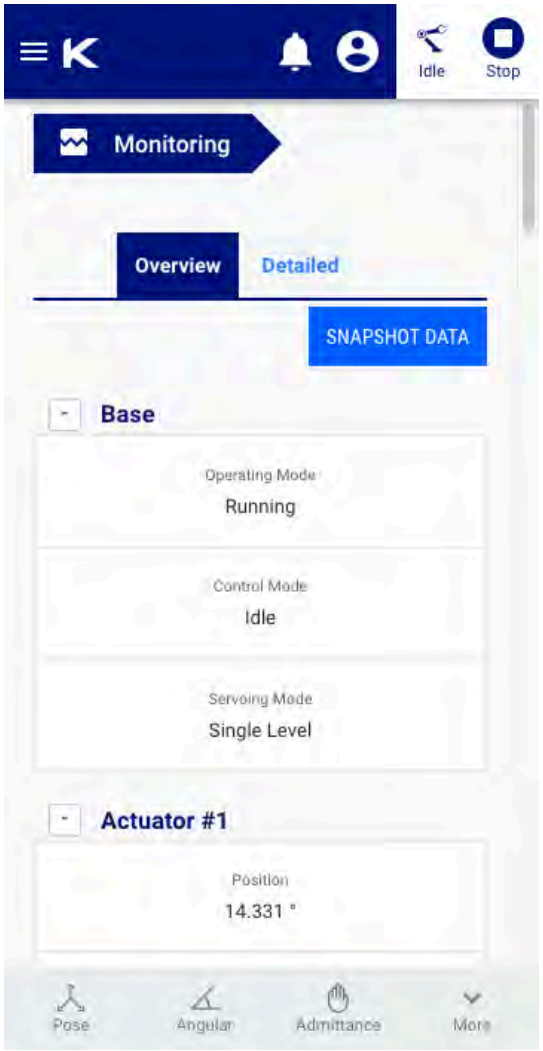


Figure 24: Smartphone

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.

Operating system support

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

Browser support

- Chrome

Other platforms and browsers are not currently supported - some features may work differently in those cases.

© Kinova inc 2021. All rights reserved.

User login

Given the robot base IP address and a computer connected to the robot, users can log in to the *Web App* with username and password via a Web browser.

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.

Enter your username and password and press the CONNECT button.

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

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

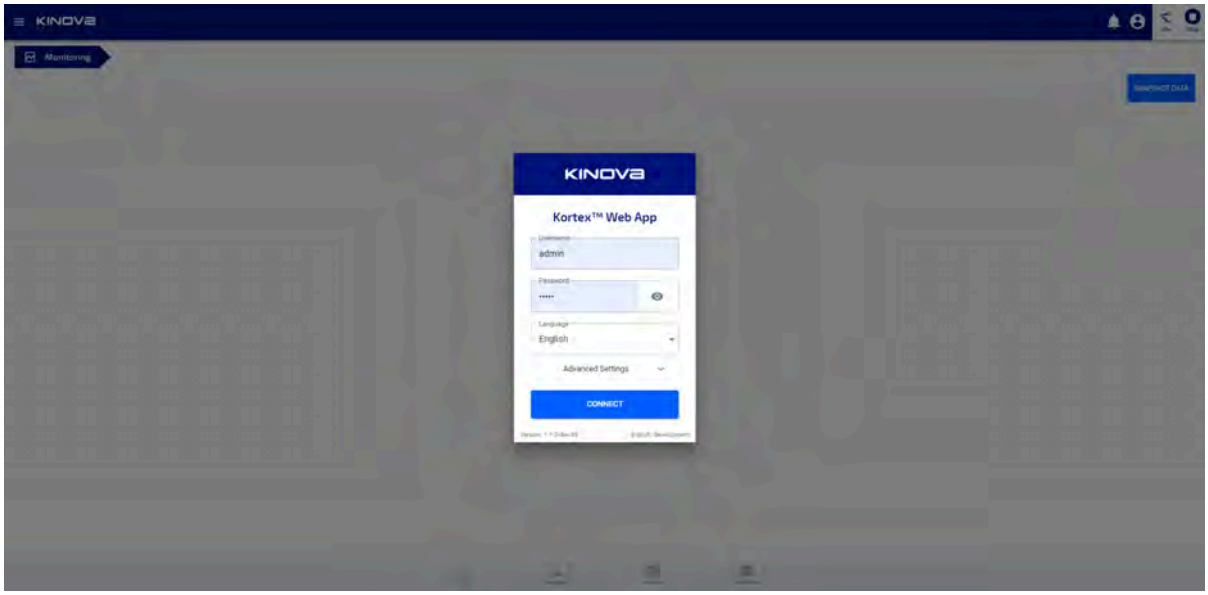


Figure 25: User login

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.

© Kinova inc 2021. All rights reserved.

Figure 26: Initializing...

Web App layout and navigation

This section describes the basic 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

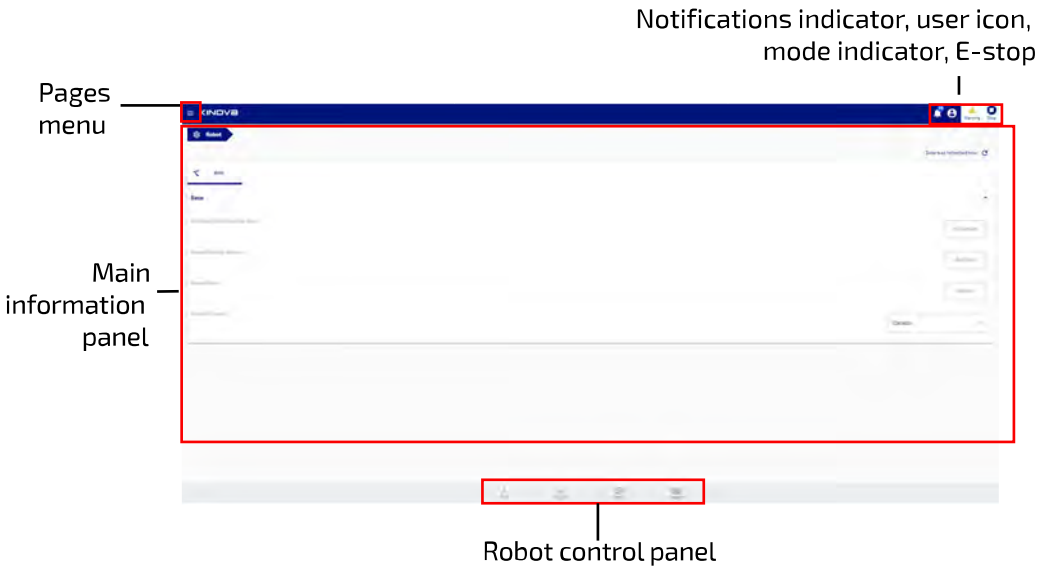
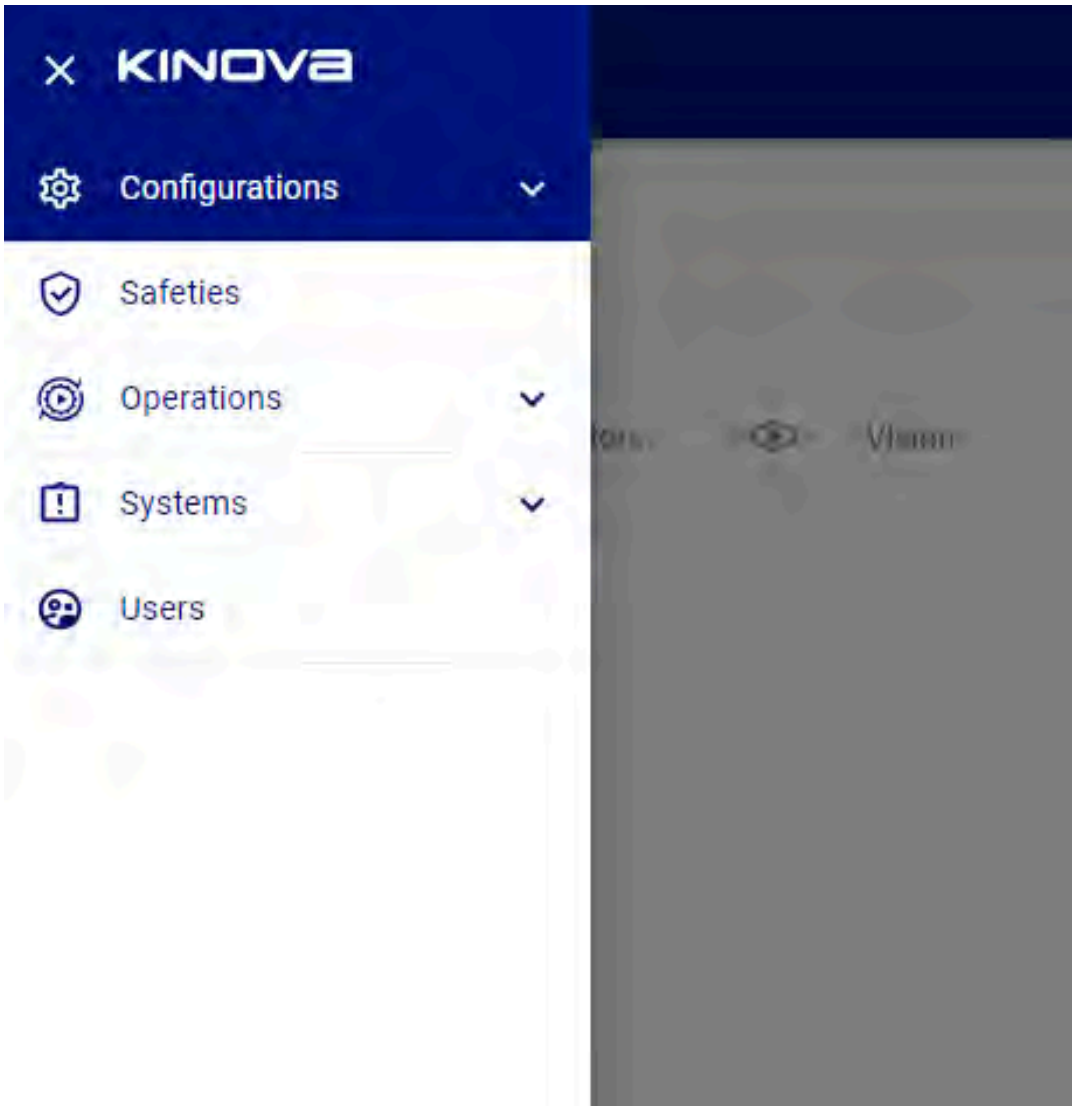


Figure 27: Web App layout

Pages menu

© Kinova inc 2021. All rights reserved.

In the middle of the screen is the main information panel containing the contents of each page of the application. The page can be changed from a pages menu on the left of the screen. This menu is hidden by default, but can be lauched by clicking / tapping the menu icon in the upper left.



The page options are organized into groups:

- Configurations
  - Robot
  - Controllers
  - Wireless & Networks
- Safeties

© Kinova inc 2021. All rights reserved.

- Operations
  - Actions
  - Protection Zones
- Systems
  - System Information
  - Monitoring
  - Upgrade
- Users

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

- Notifications indicator - number of most important notifications. Clicking allows notifications to be read.
- User profile icon - Shows the user session icon
- Control mode - status of the control mode situation of the robot. There are four icons to indicate the mode / state:
  - Warning - robot in warning state
  - Error - robot is in a fault state
  - Idle - robot is not currently being controlled by any user session; waiting
  - Playing - a sequence is being played on the robot
  - Running - the robot is being actively controlled by a user
- Stop Emergency Stop (E-stop) - button control which when pressed / tapped will initiate the emergency stop of the robot.

© Kinova inc 2021. All rights reserved.

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

Robot control panel

The control panel is on the bottom of the screen, and consists of a group of four buttons. Two are to launch pop-up windows for virtual joystick controls:

-

Pose Virtual Joystick
-

Angular mode control

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

In the same area there are two other controls:

-

Play action - brings up a window to play a selected action
-

Snapshot - allows user to capture a Cartesian, angular, or gripper pose

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.

Robot control panel

Pose virtual joystick control

The Pose virtual joystick provides an interface to control Cartesian movement of the robot from the *Web App*.

The Pose 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).

© Kinova inc 2021. All rights reserved.

Figure 28: Pose joystick panel

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

Translation / rotation joystick controls

The Pose virtual joystick controls allow you to control the linear and angular motion of the tool. There are two sets of joysticks:

- linear (to apply a translational motion to the end effector)
- angular (to apply a rotational motion to the end effector at the current position)

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 linear 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.

Note: The orientation representation uses a z-y-x Tait-Bryan extrinsic convention.

Finger controls

It is possible to open and close the fingers using a single 1-axis joystick control. 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 caret icon on the upper right-hand side, additional controls and displays are revealed.

- current linear and angular velocity display
- invert z and θ_z toggles

© Kinova inc 2021. All rights reserved.

- reference frame selection
- gripper speed control
- speed limit

Speed control

The actuators speed and finger speed can be adjusted between 0 and 100% of the hard limits for the robot.

Reference frames

The position of the end effector can be specified in one of three reference frame conventions:

- Mixed - linear in base reference frame, angular in tool reference frame
- Base - linear and angular in base reference frame
- Tool - linear and angular in tool reference frame

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.

Angular virtual joystick control

The Angular virtual joystick provides an interface to control the join by joint movement of the robot from the *Web App*.

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

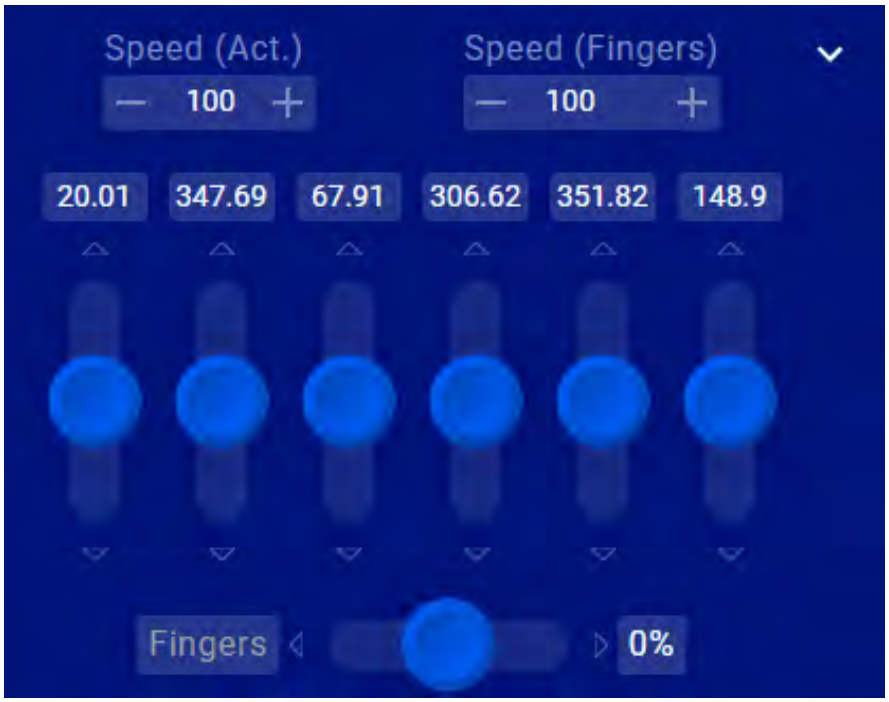


Figure 29: Angular virtual joystick panel

The joint angles are controlled through angular velocity (controlling the joint speed for each actuator)

© Kinova inc 2021. All rights reserved.

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 with joint rotation limits, 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.

The value of each angle is displayed in degrees. The value displayed is restricted to one full rotation (0 - 360°).

Additional settings

By clicking the caret icon on the right-hand side, additional controls are revealed. These let the user control the maximum actuators and finger speed, between 0 and 100% of maximums.

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 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 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

Keyboard shortcuts are available for *Web App* virtual joysticks when accessing the *Web App* from a desktop device with keyboard. This serves as an alternate to mouse and touch for controlling the robot via the 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 44: Pose translation joystick shortcuts

Control		Shortcut
X translation	+	W
	-	S
Y translation	+	A

© Kinova inc 2021. All rights reserved.

Control		Shortcut
	-	D
Z translation	+	Q
	-	E

Table 45: Pose orientation joystick shortcuts

Control		Shortcut key
X rotation	+	I
	-	K
Y rotation	+	L
	-	J
Z rotation	+	U
	-	O

Table 46: Gripper controls

Control		Shortcut key
Gripper fingers	open gripper fingers	C
	close gripper fingers	V

Angular joystick keyboard shortcuts

Table 47: Actuators controls

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

© Kinova inc 2021. All rights reserved.

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

The angles can be decreased using the letter keys QWERTYU on the second row of the keyboard.

Table 48: Gripper controls

Control		Shortcut
Gripper fingers	open gripper fingers	C
	close gripper fingers	V

Actions panel

The Actions panel of the *Web App* gives a control panel for the currently selected Action.

The

Actions panel provides controls to play, stop, or loop a selected Action.

The panel pops up from the bottom of the screen when an Action is selected in the Actions page. Clicking the control panel Actions icon will toggle the visibility of the panel.

Snapshot tool

The snapshot tool in the *Web App* captures and saves one of three types of instantaneous snapshots of the robot / gripper positioning.

The snapshot tool lets users capture a snapshot of a current position. This is a useful feature to help with building pre-set sequences (for demos or to capture / program a fixed set of movements).

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

- Cartesian pose
- Joints position
- Gripper position

Pressing one of the respective snapshot buttons will capture the chosen type of snapshot of the current robot position. This will be saved, and will show up as one of the saved Actions viewable in the Actions page.

Main pages

Configurations page group

The Configurations page group of the *Web App* includes several pages useful for robot configuration.

© Kinova inc 2021. All rights reserved.

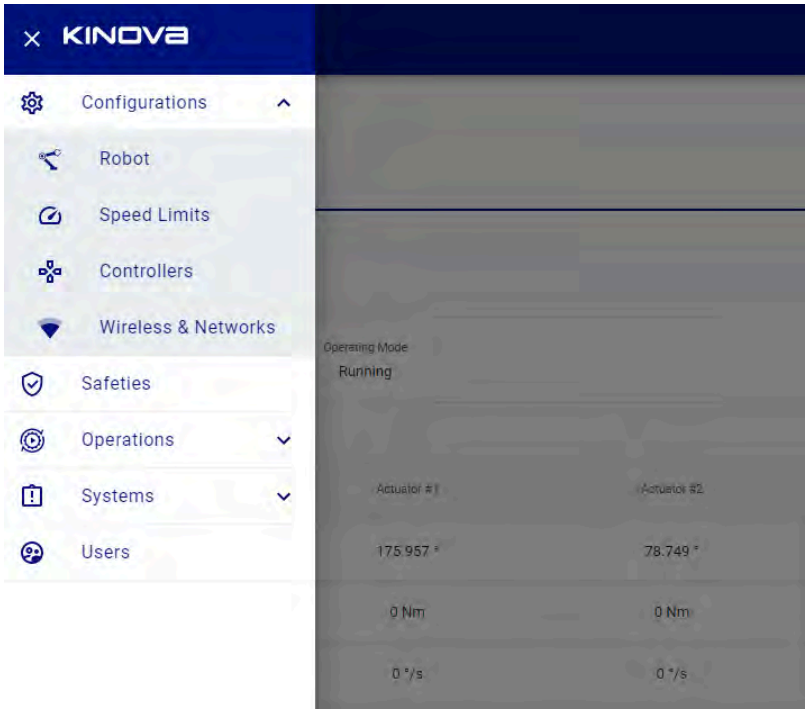


Figure 30: Configurations page group

The Configurations page group contains pages allowing users to perform preliminary set up and configuration for robot hardware.

This includes the following pages:

- Robot
- Controllers
- Wireless & Networks

Robot configurations

The Robot configurations page of the *Web App* allows for the configuration of a broad range of robot parameters.

The Robot configurations page provides a GUI to adjust the configurable parameters of the robot hardware in order to customize its behavior.

© Kinova inc 2021. All rights reserved.

Figure 31: Robot configurations page

The configurable items are broken into sections by devices:

- Arm

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

- Protection Zones
- Controllers (switch active control mapping)
- Actions (define actions)
- Users (edit user profiles)
- Wireless & Networks (update network settings)

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

Controllers

The Controllers page of the *Web App* lets users view and select control mappings for the robot.

The Controllers page lets you view and toggle between the defined control mappings for any physical controllers associated with the robot. It also provides tools for creating new control mappings.

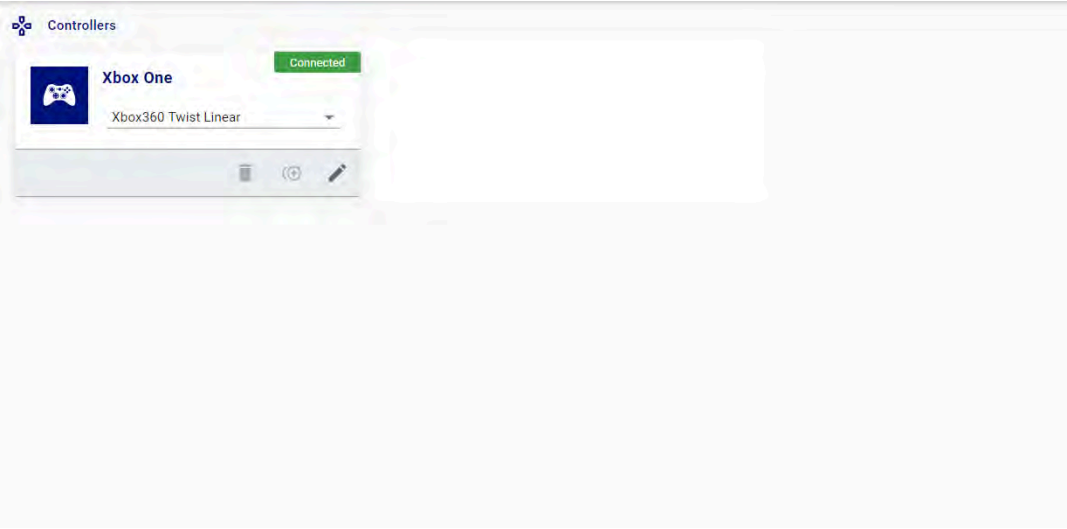


Figure 32: Controllers page

© Kinova inc 2021. All rights reserved.

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

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.

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

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

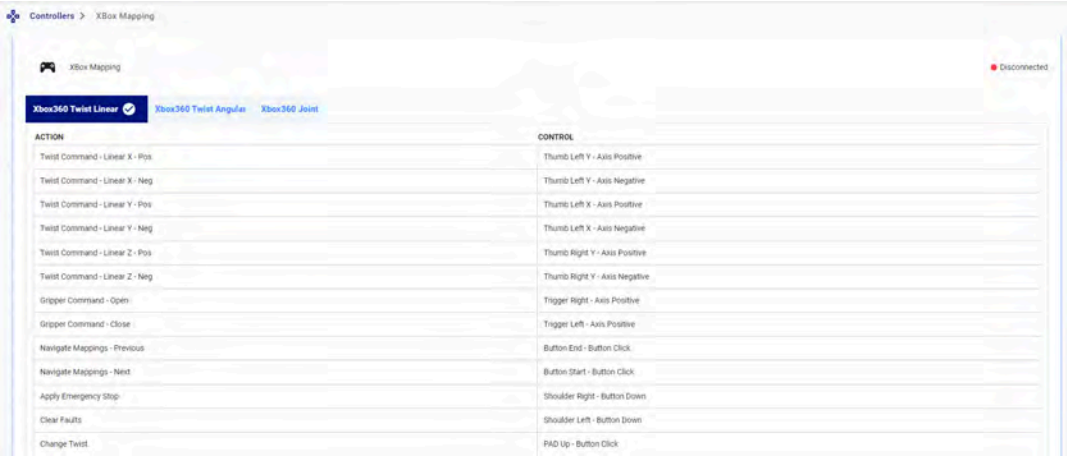


Figure 33: Xbox Mapping tab

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.

Wireless & Networks

The Wireless & Networks page of the *Web App* allows users to configure networks settings.

The Networks page is used to set network parameters for:

- Wi-Fi
- RNDIS/Micro USB
- USB-A Dongle

© Kinova inc 2021. All rights reserved.

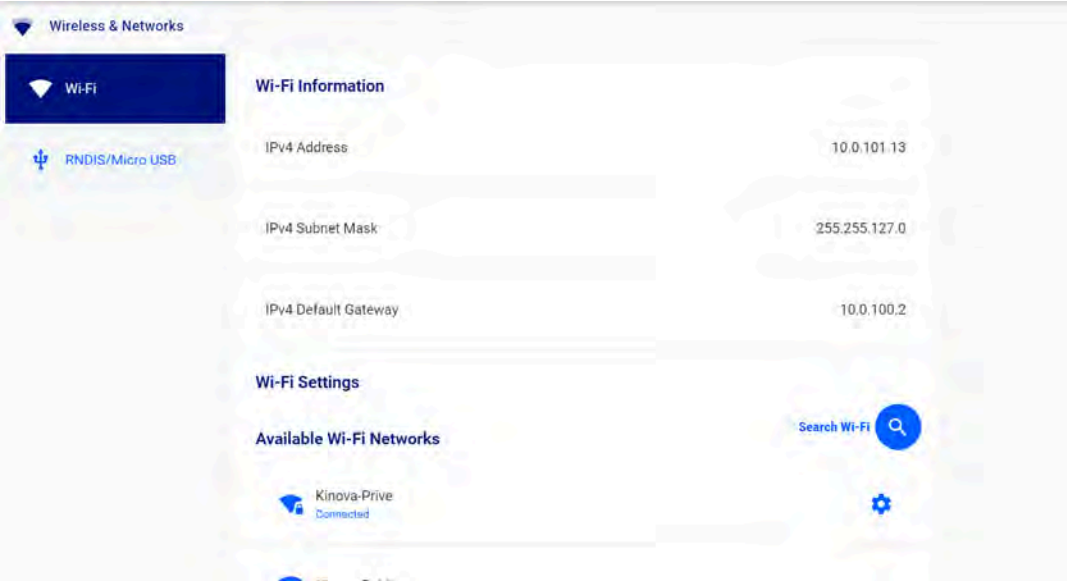


Figure 34: Wireless & Networks page

The page has tabs for each currently available connection method.

The RNDIS / Micro USB Settings tab allows enabling and disabling DHCP server.

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

Safeties

The Safeties page of the *Web App* allows users to view safety settings on the robot.

The Safeties page allows users to view safety thresholds.

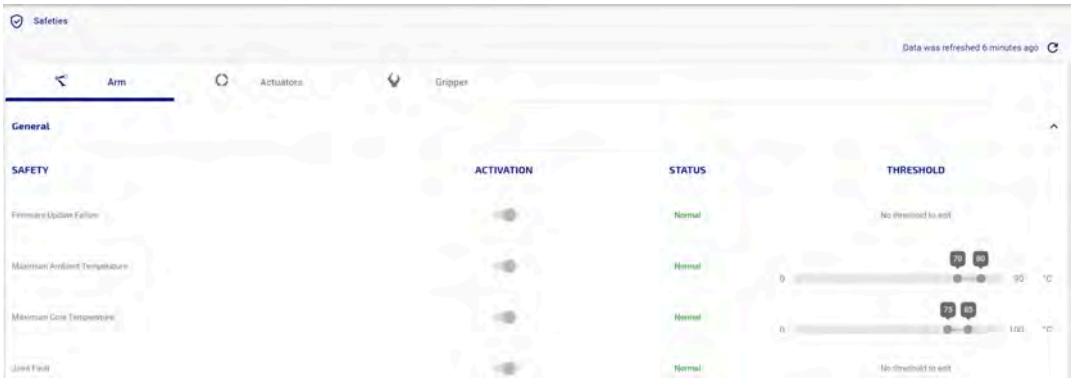


Figure 35: Safeties page

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.

© Kinova inc 2021. All rights reserved.

- **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.

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

Operations page group

The Operations page group of the *Web App* groups together pages useful to operating the robot.

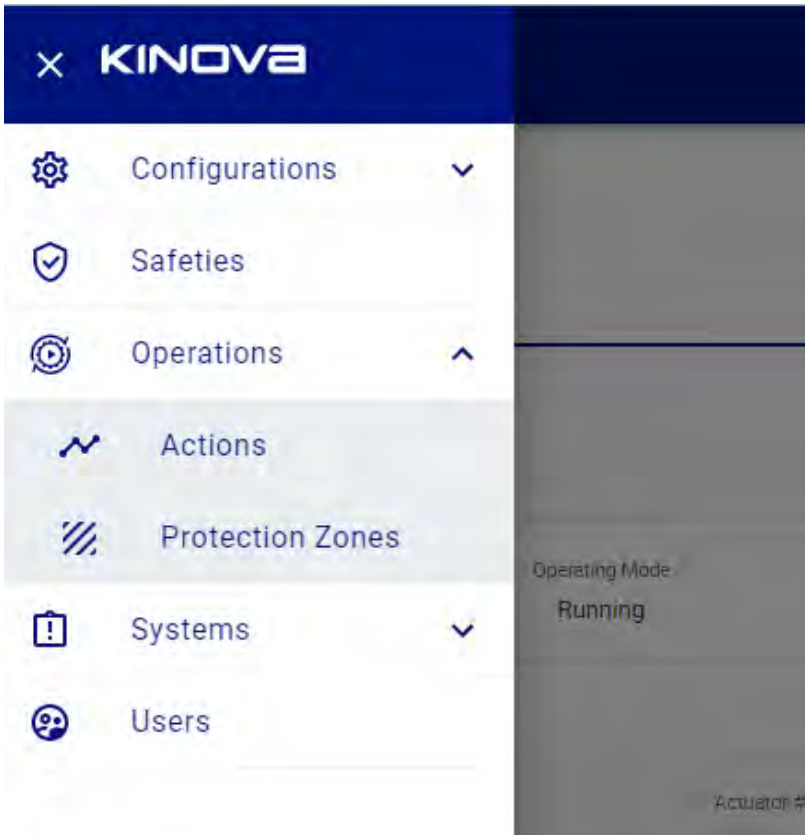


Figure 36: Operations page group

The Operations page group contains pages allowing users to fine tune the configuration and performance of the robot after the initial setup.

This includes the following pages:

- Actions
- Protection Zones
- Camera

© Kinova inc 2021. All rights reserved.

Actions

The Actions page of the *Web App* allows the user to define and edit robot actions of various types. It also allows users to play back actions and assemble them together into sequences of action.

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

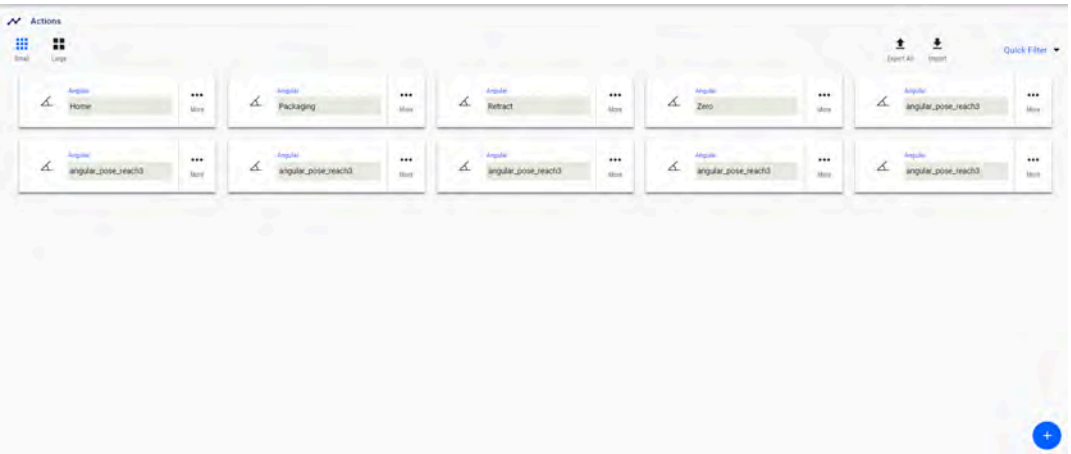


Figure 37: Actions page

Actions available in the *Web App* are:

- Pose (go to a Cartesian pose)
- Angular (go to a combination of joint angles)
- End Effector (change the gripper state)
- Sequence (take a series of actions one after the other)

A **Cartesian pose** represents a single Cartesian endpoint 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).

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 smooth the 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.

© Kinova inc 2021. All rights reserved.

Figure 38: '+' menu

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

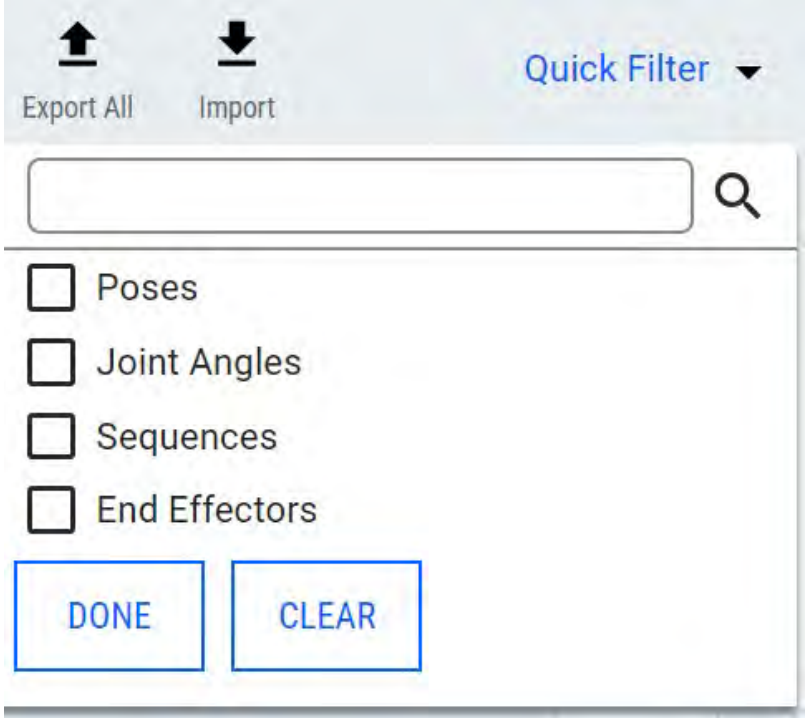


Figure 39: Action filter

© Kinova inc 2021. All rights reserved.

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.

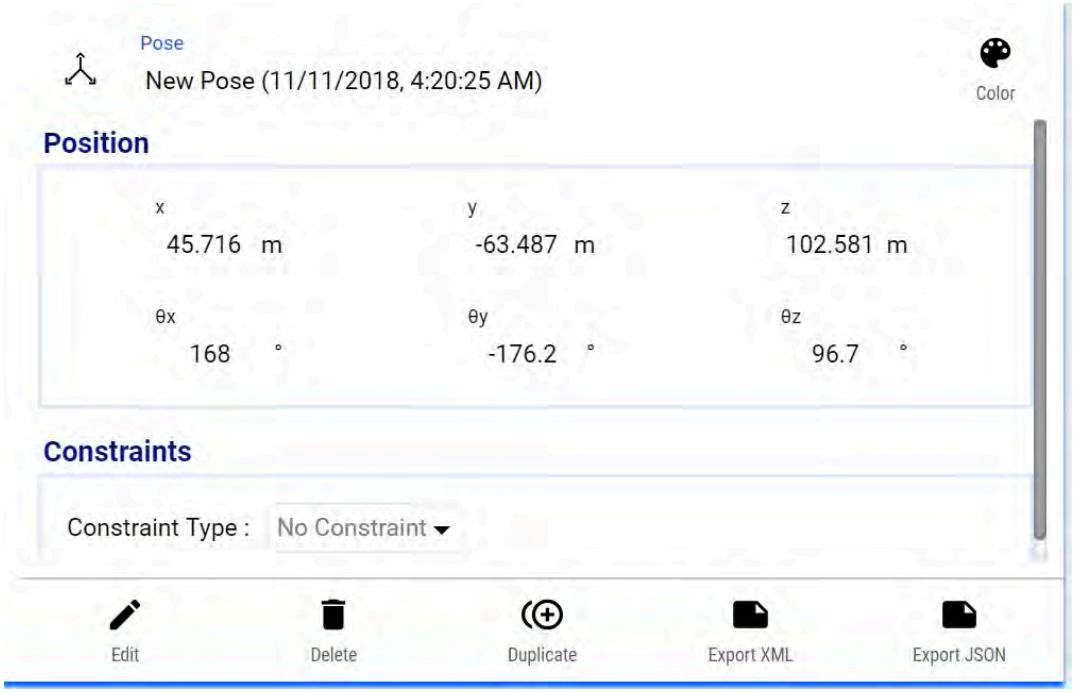


Figure 40: Information card

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
- apply relevant constraints (speed and / or duration) on the action, depending on the type of action

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.

© Kinova inc 2021. All rights reserved.

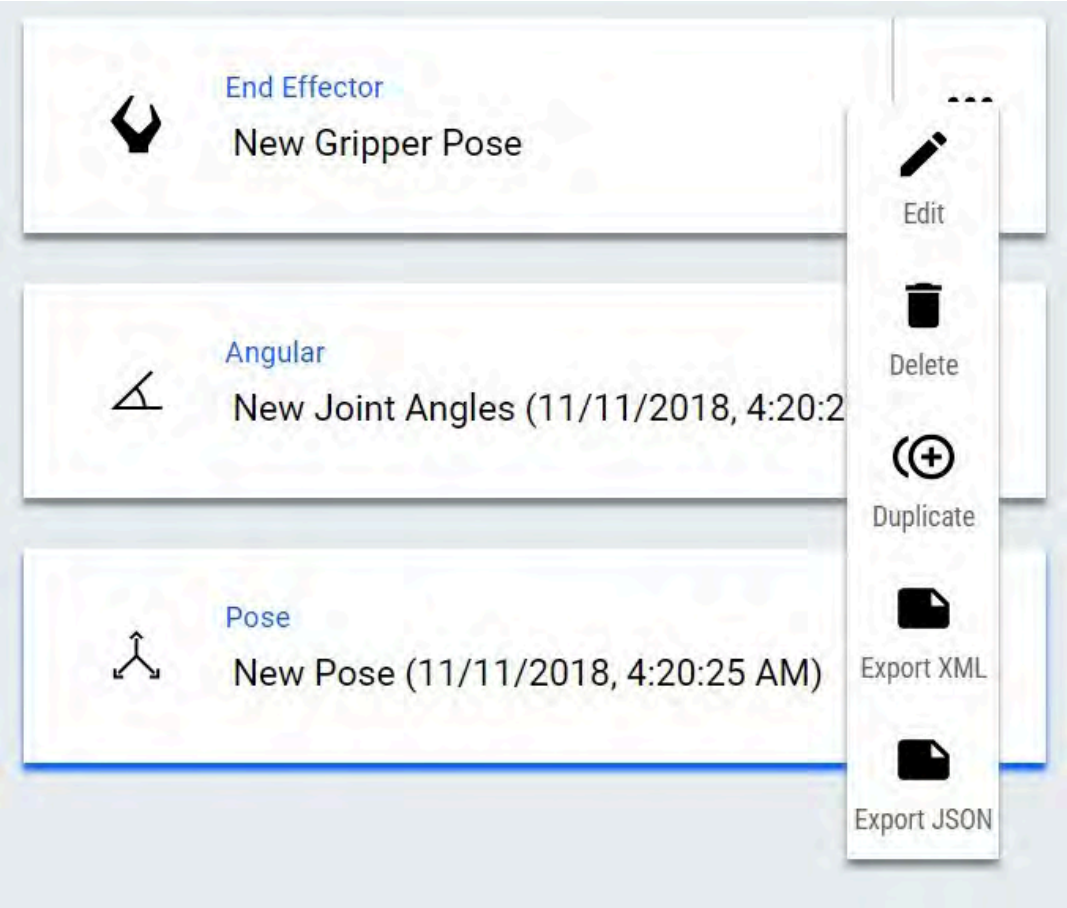


Figure 41: 'More' pop-up menu

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

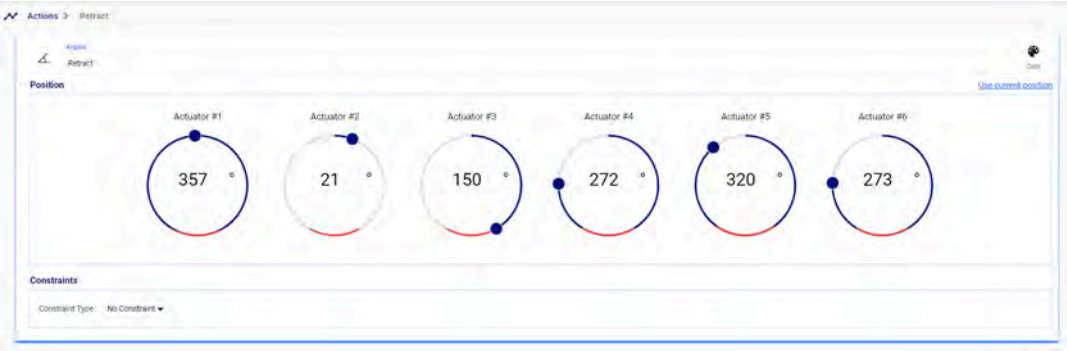


Figure 42: Parameter modifications

Creating actions using snapshot tool

The snapshot button (Snapshot) at the bottom of the screen can be used to capture the current robot Cartesian pose, angular positions, 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

The sequence editor on the *Web App* Actions page is used to build and edit sequences of actions.

© Kinova inc 2021. All rights reserved.

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

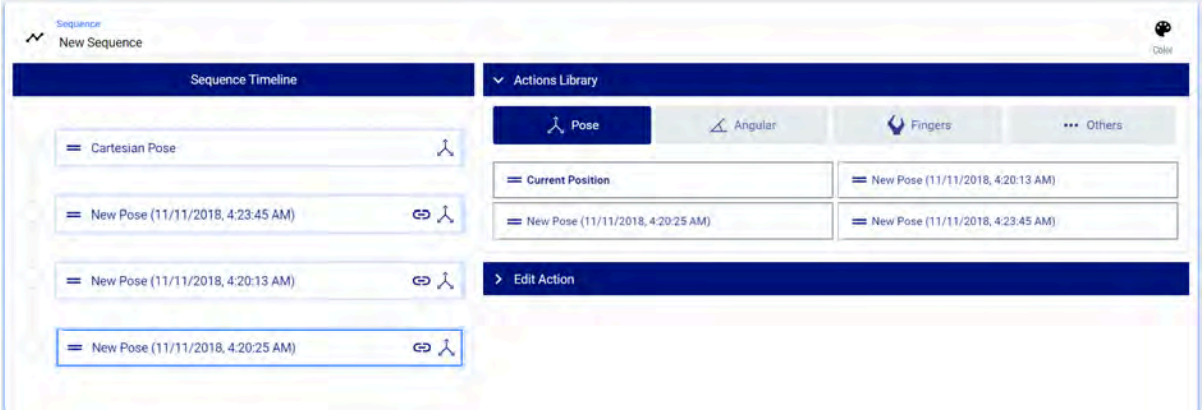


Figure 43: Sequence editor

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.

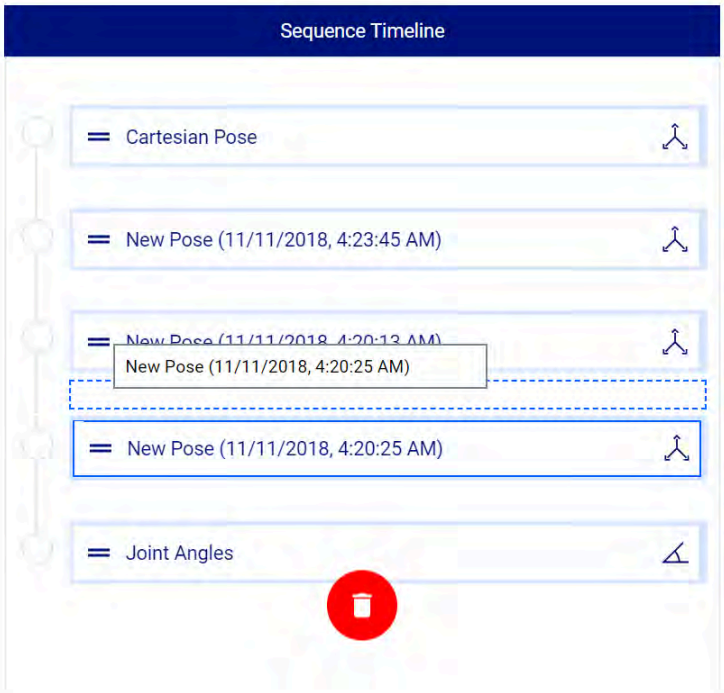
The link icon indicates that the action referred to in the Sequence is a reference to an existing standalone action. If that standalone action is modified outside this sequence it will have an impact on the sequence. By opposition, if the action (for example the first one in your screen capture) does 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

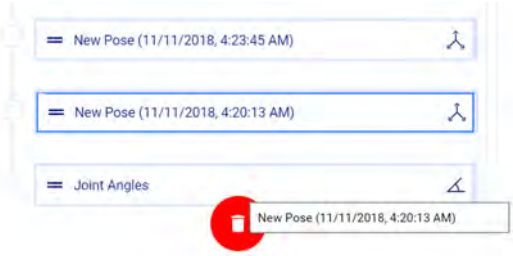
If you want to change the position of an action in the sequence, or remove it from the sequence entirely, simply hover over the left side of the box for the action until the cursor changes from a hand icon to a two-dimensional arrow icon.

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.

© Kinova inc 2021. All rights reserved.



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 exporitng 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 Export All and Import Import functions are available at the top of the main panel.

Playing back actions and sequences

Actions and sequences can be played back from the Actions page, and controls are provided to manage playback.

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.

© Kinova inc 2021. All rights reserved.

Figure 44: Actions playback bar

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 **Play** button plays the sequence.

The **Pause** button (the Play button changes to a Pause button while the sequence is playing) 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 pose, angular position, or gripper position, the robot (or gripper) will interpolate linearly between the present position and the target position and move smoothly and directly to the target position.

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 positions on the sequence. A progress bar above the playback bar shows the progress of the playback through the steps.

For sequences, an additional **Loop toggle** 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

The Protection Zones page of the *Web App* allows users to define zones around the robot where movement is prevented or limited.

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.

© Kinova inc 2021. All rights reserved.

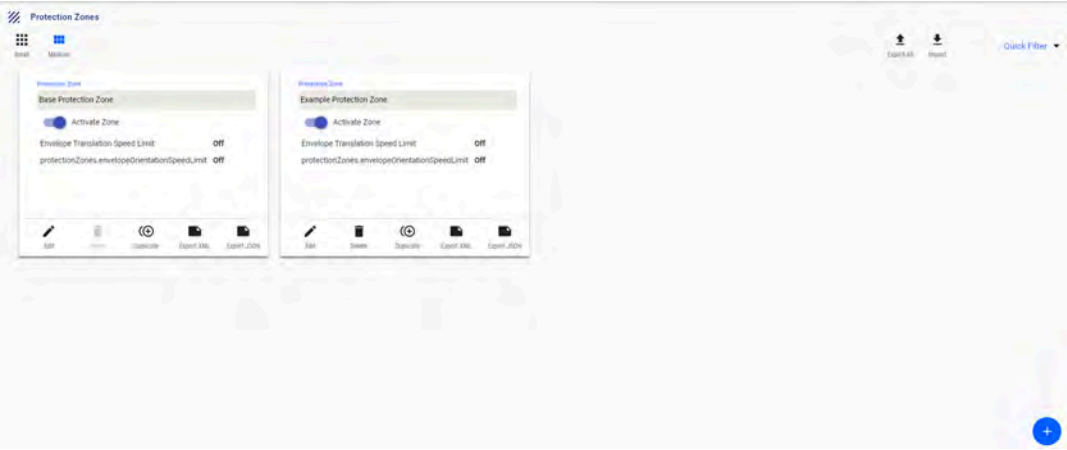


Figure 45: Protection Zones page

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 two different sizes:

- Small
- Medium

The card sizes can be toggled using buttons at the top of the main information window. The medium 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,.

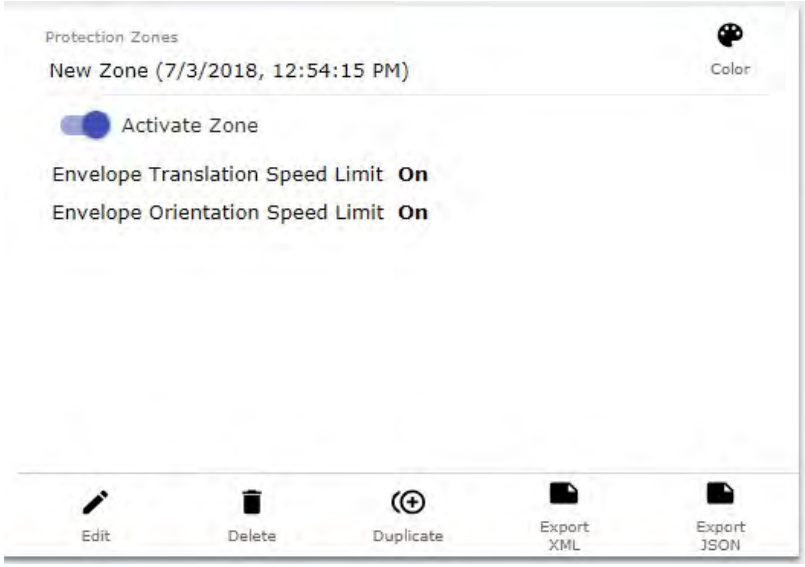


Figure 46: Protection Zone card

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.

© Kinova inc 2021. All rights reserved.

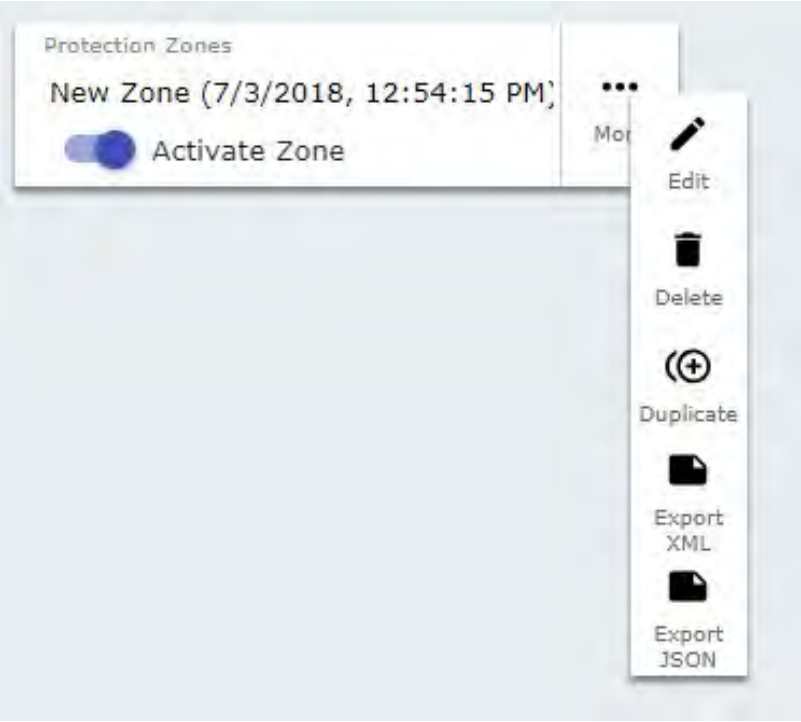


Figure 47: 'More' menu

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.

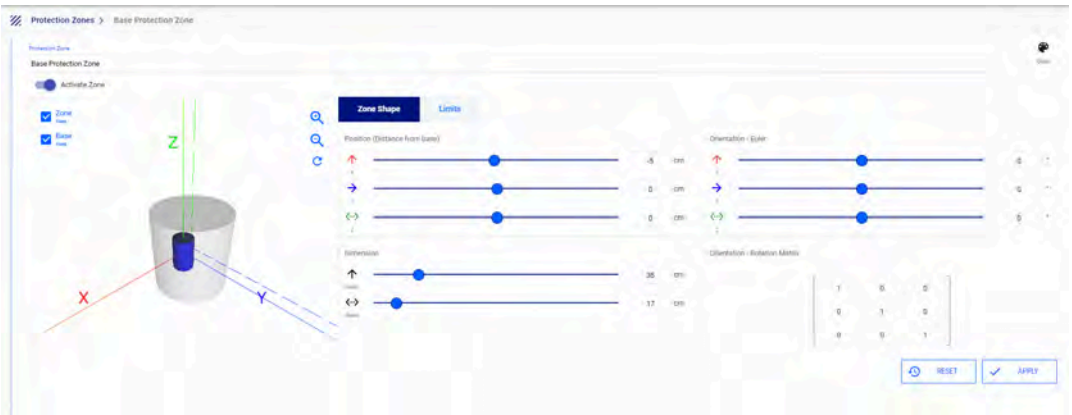


Figure 48: Configure protection zone

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

© Kinova inc 2021. All rights reserved.

- 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 49: Limits tab

Under the Limits tab, speed limitations can be set for the protection zone. Limits can be set for:

- Envelope Translation Speed (between 0 and 10 cm/s)

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

Systems page group

The Systems page group of the *Web App* groups together pages to update and view status and configiration information for the system.

© Kinova inc 2021. All rights reserved.

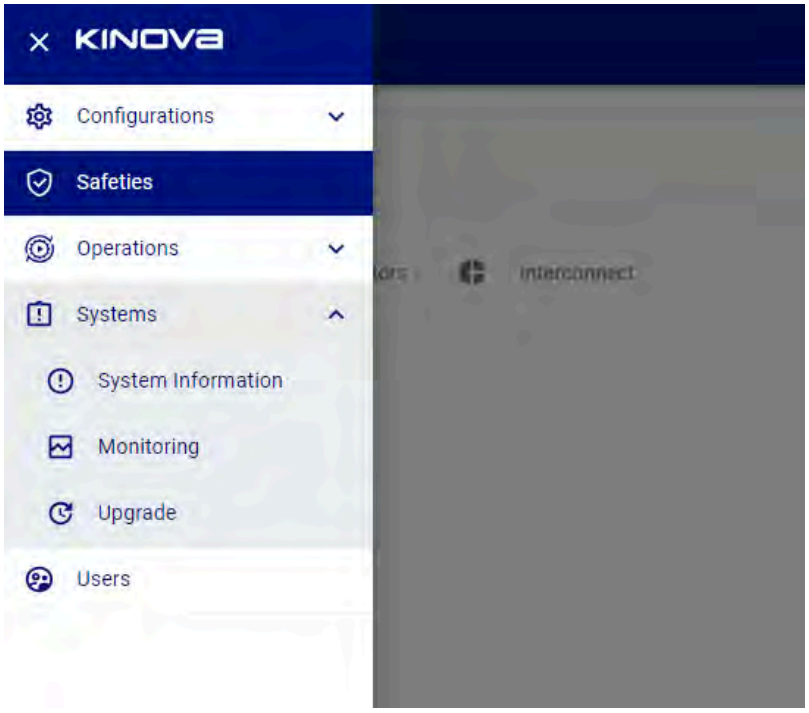


Figure 50: Systems page group

The Systems page group contains pages for robot devices information, monitoring, and software / firmware upgrades.

It contains the following pages:

- System Information
- Monitoring
- Upgrade

System Information

The System Information page of the *Web App* provides high-level hardware and firmware details for the robot.

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

© Kinova inc 2021. All rights reserved.

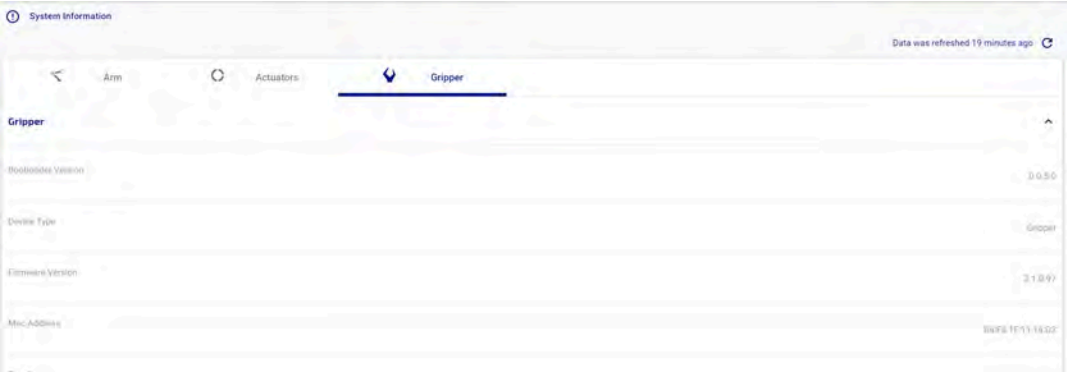


Figure 51: System Information page

The information on the page is displayed within different tabs:

- Arm (product and base)
- Actuators (for each of the individual actuators in robot)
- Gripper

For the base, actuators, gripper, information is given on:

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

Product gives product-level information:

- KIN (Kinova information number)
- model ID
- model year
- assembly plant
- degrees of freedom
- base type
- end effector type
- arm laterality
- wrist type

Monitoring

The Monitoring page of the *Web App* allows users to view real-time monitoring information on the status and performance of the robot.

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*.

© Kinova inc 2021. All rights reserved.

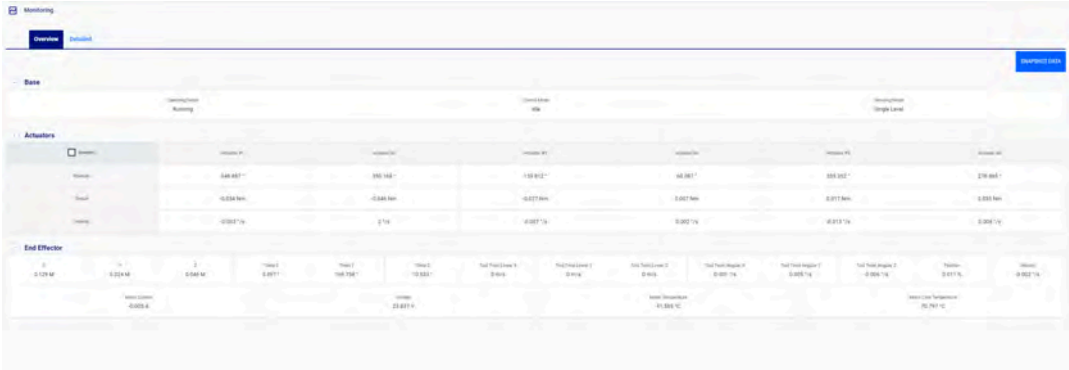


Figure 52: Monitoring page

The monitoring information is divided into sections:

- Base
- Actuators
- 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)
  - servoing mode (single level (high level), low level)
- Actuators - for each joint:
  - measured position (°)
  - measured torque (N·m)
  - measured velocity (° / 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:

© Kinova inc 2021. All rights reserved.

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

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 (

), 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

The Upgrade page of the *Web App* is used to update the software of the robot with a downloaded software package.

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
- *Web App* upgrade package
- KINOVA® KORTEX™ API upgrade package

© Kinova inc 2021. All rights reserved.

Figure 53: 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 how 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. Once the firmware package uploads successfully, you will be able to start the upgrade. The *Web App* will refresh twice during the upgrade, and each time you will have to login again. If you login during the upgrade, the Upgrade page will indicate that the upgrade is in progress.

Results

The *Web App* will indicate when the process is finished.

Users

This section describes the Users page of the *Web App* allows for editing user profile information.

© Kinova inc 2021. All rights reserved.

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

Figure 54: 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.

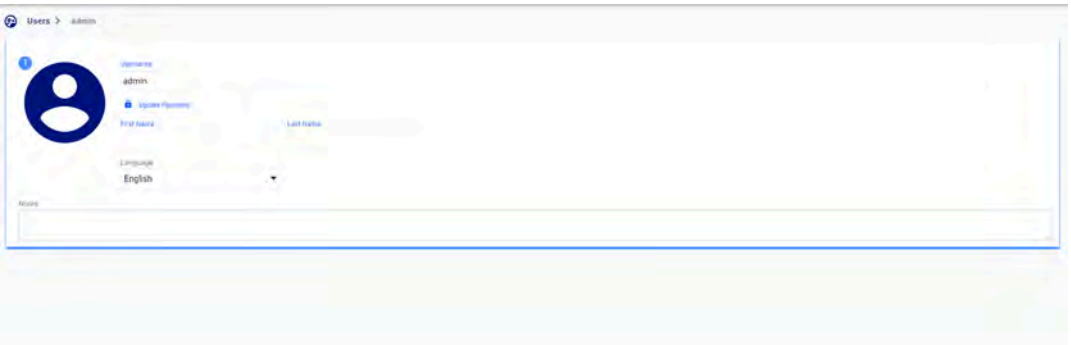


Figure 55: Card editing interface

Click on the button in the lower right corner of the page to create a new user. This launches a menu to add information for the new user.

© Kinova inc 2021. All rights reserved.

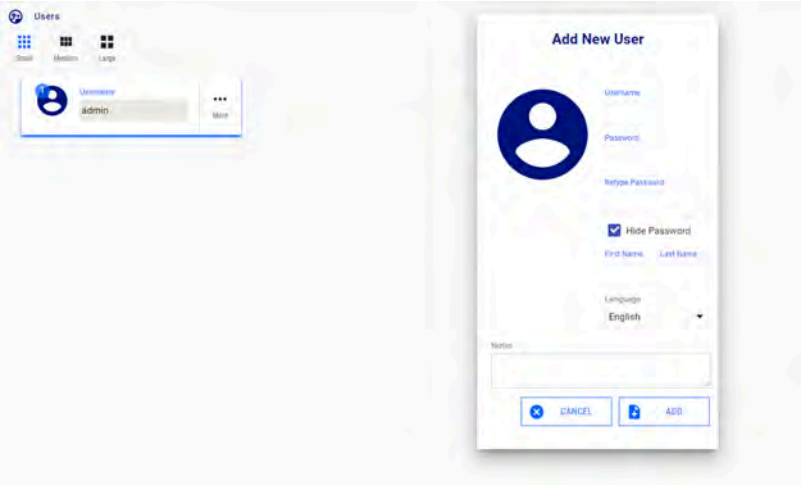


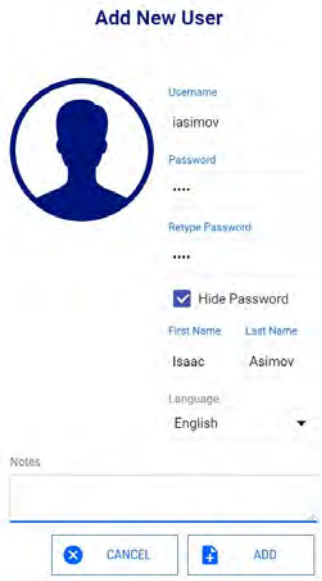
Figure 56: Create new user

Creating a new user profile

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

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.

© Kinova inc 2021. All rights reserved.

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 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 supports multiple robot products from Kinova as a cross-hardware development framework.

Note: Some of the specific features of the API are hardware dependent and may not be available on your robot.

APIs are currently provided for the following languages:

- C++
- Python
- MATLAB® (simplified API supporting a subset of Kortex functionality)

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 following GitHub respositories contain additional developer guidance and resources, including detailed API documentation, setup instructions, and source code examples:

- KINOVA® KORTEX™ API: kinovarobotics/kortex
- KINOVA® KORTEX™ ROS: kinovarobotics/ros_kortex

Tutorial videos are available on the KINOVA® YouTube channel (https://www.youtube.com/watch?v=zQewb08M4sA).

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)
- gripper

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).

© Kinova inc 2021. All rights reserved.

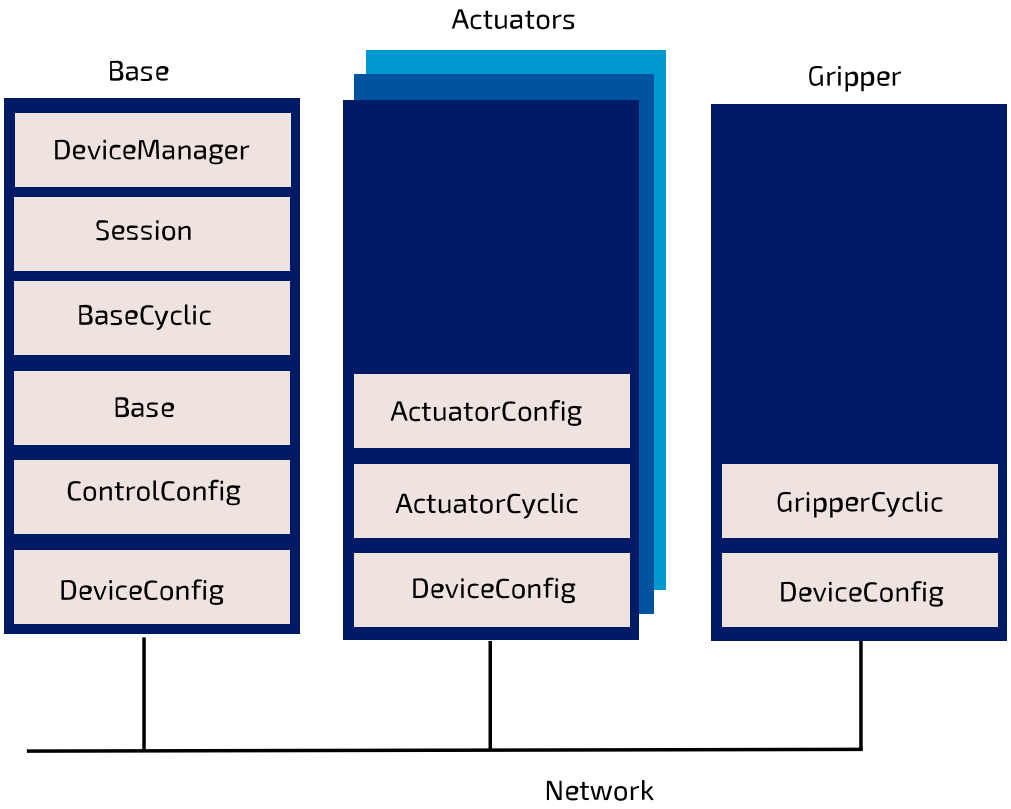


Figure 57: Services on multiple devices

Available services

This section lists the available robot services for the robot.

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.

© Kinova inc 2021. All rights reserved.

- 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 control loop.
  - For low-level servoing cyclic communication
    - `BaseCyclic`- sending commands to actuators and gripper motors and obtain feedback from base, actuators, wrist interface, and gripper motors.
- Configuration related
  - `ControlConfig` - get / set control library configuration
  - `ActuatorConfig` - get / set actuator configuration
  - `DeviceConfig` - get / set general device configuration

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

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

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 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.

© Kinova inc 2021. All rights reserved.

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.

Blocking and non-blocking calls

Remote procedure calls on the robot are generally available in both blocking and non-blocking forms. Non-blocking calls are available using Future / Promise and by registering callback functions.

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 functions

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

Device routing and transport

This section describes the concept of device routing and low-level transport.

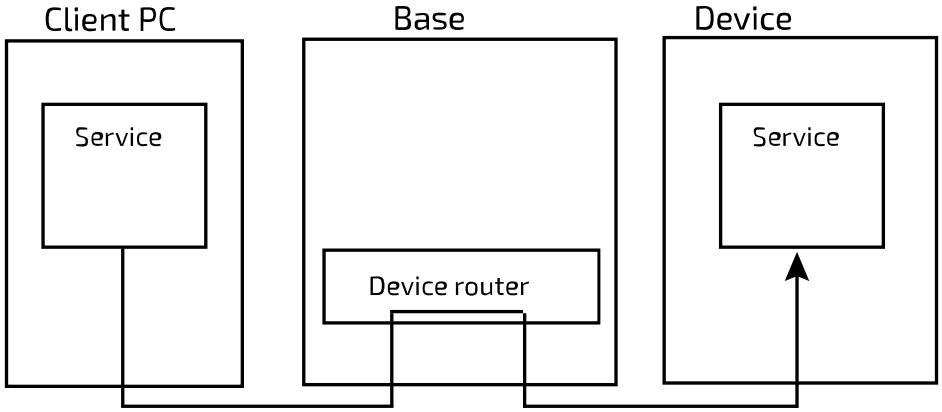


Figure 58: Device routing

© Kinova inc 2021. All rights reserved.

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

The routing from the client computer, through the base to a device is managed by a `RouterClient` object in the API. A Session on the robot depends on a `RouterClient` object.

RouterClient objects depend, for low-level details of the transport of the routed methods, on `Transport` objects. A `Transport` object uses a specific underlying network transport protocol. The API defines two types of Transport objects, one using TCP as the protocol, and another using UDP.

TCP `Transport` objects are used for high-level robot control and configuration.

UDP `Transport` object are used for low-level (real-time) cyclic communication with the robot.

Note: Since Sessions require a `RouterClient`, and a `RouterClient` is tied to a `Transport` object using a specific protocol, separate Sessions must be created if high-level control / configuration and low-level control are used in the same application.

For more details, see the API documentation on GitHub.

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 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 communications with the actuators.

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

© Kinova inc 2021. All rights reserved.

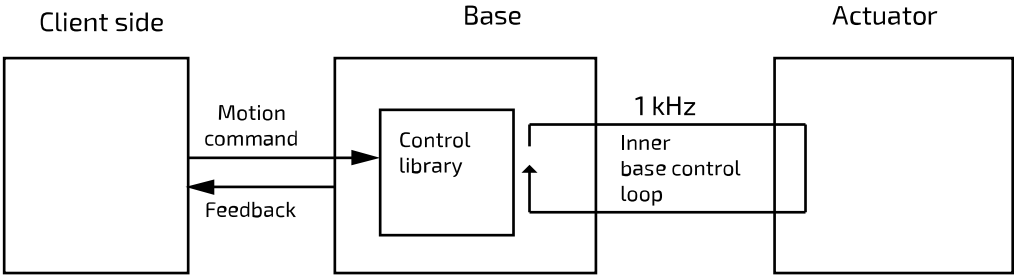


Figure 59: High-level servoing

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, 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 command increments (bypassing the kinematic control library).

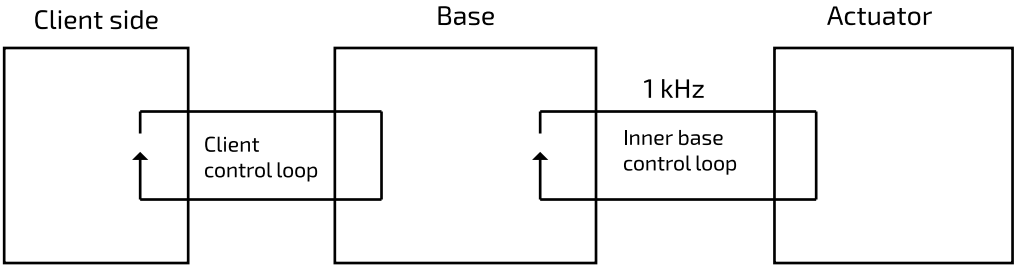


Figure 60: Low-level servoing

© Kinova inc 2021. All rights reserved.

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.

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 contents of 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 inc 2021. All rights reserved.

KINOVA® KORTEX™ ROS packages and GitHub repository overview

This section describes the contents of 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 and ROS Melodic Morenia.

Note that ROS Kinetic Kame is only compatible with Ubuntu 16.04 (Xenial) LTS.

ROS Melodic is compatible with Ubuntu 18.04 (Bionic)

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).

Support is included for Gazebo and MoveIt.

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_api` (package containing header files and libraries needed to use the C++ Kortex API)
- `kortex_control` (package contains configuration files for the ros_control controllers used to control the simulated robot
- `kortex_description` (package contains URDF and STL files of the robot)
- `kortex_driver` (ROS node package to allow direct communication with the robot base)
- `kortex_examples` (examples needed to understand the basics of ros_kortex)
- `kortex_gazebo` (package contains files to simulate the robot)
- `kortex_moveit_config` (contains all the auto-generated MoveIt! configuration ROS packages)

© Kinova inc 2021. All rights reserved.

Guidance for advanced users

Overview

The advanced guidance section gathers together reference information on topics useful to advanced users of the robot.

Introduction

The following contents are intended for advanced users.

Reference frames and transformations

Standard robot frames

This section describes the standard frames of the robotic arm.

The robot has two standard frames:

- base frame (base reference frame)
- tool frame (end-effector reference frame)

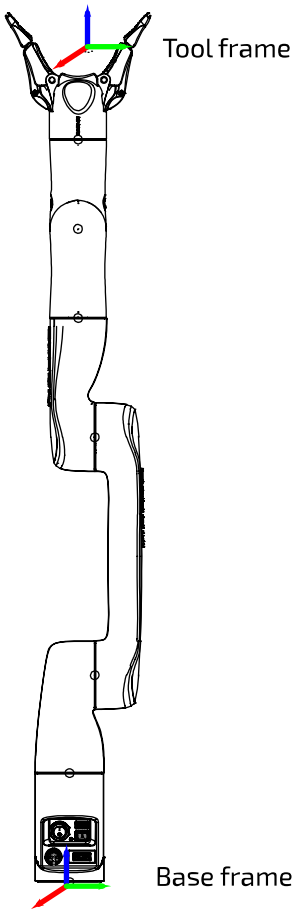


Figure 61: Standard frames

Different control modes make use of different frames.

© Kinova inc 2021. All rights reserved.

Homogeneous transforms

Homogeneous transforms for the robot are provided. Homogeneous transforms define transformations between successive reference frames in the robot, allowing coordinates in one frame to be expressed in terms of another.

Introduction

The forward kinematics of the robot are determined by homogeneous transform matrices. These matrices represent the transformations from one frame (base, joint, or interface) to the next along the kinematic chain. These transformations allow for the coordinates in one frame to be expressed in terms of another.

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

$$^{B}T^{*}_{TOOL} = {}^{B}T^{*1}_{1}T^{*2}_{2}T^{*3}_{3}T^{*4}_{4}T^{*5}_{5}T^{*6}_{6}T^{*}_{TOOL}$$

Here:

$$^{i-1}T^{*}_{i} = {}^{i-1}T_{i} * R_{z}(q_{i})$$

Where:

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

$^{i-1}T_{i}$ is the transform from the previous frame [i-1] to the current frame [i] when q_i, the angle for joint i, is 0 (Here, q_i is the angle in radians).

$R_z(q_i)$ 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.):

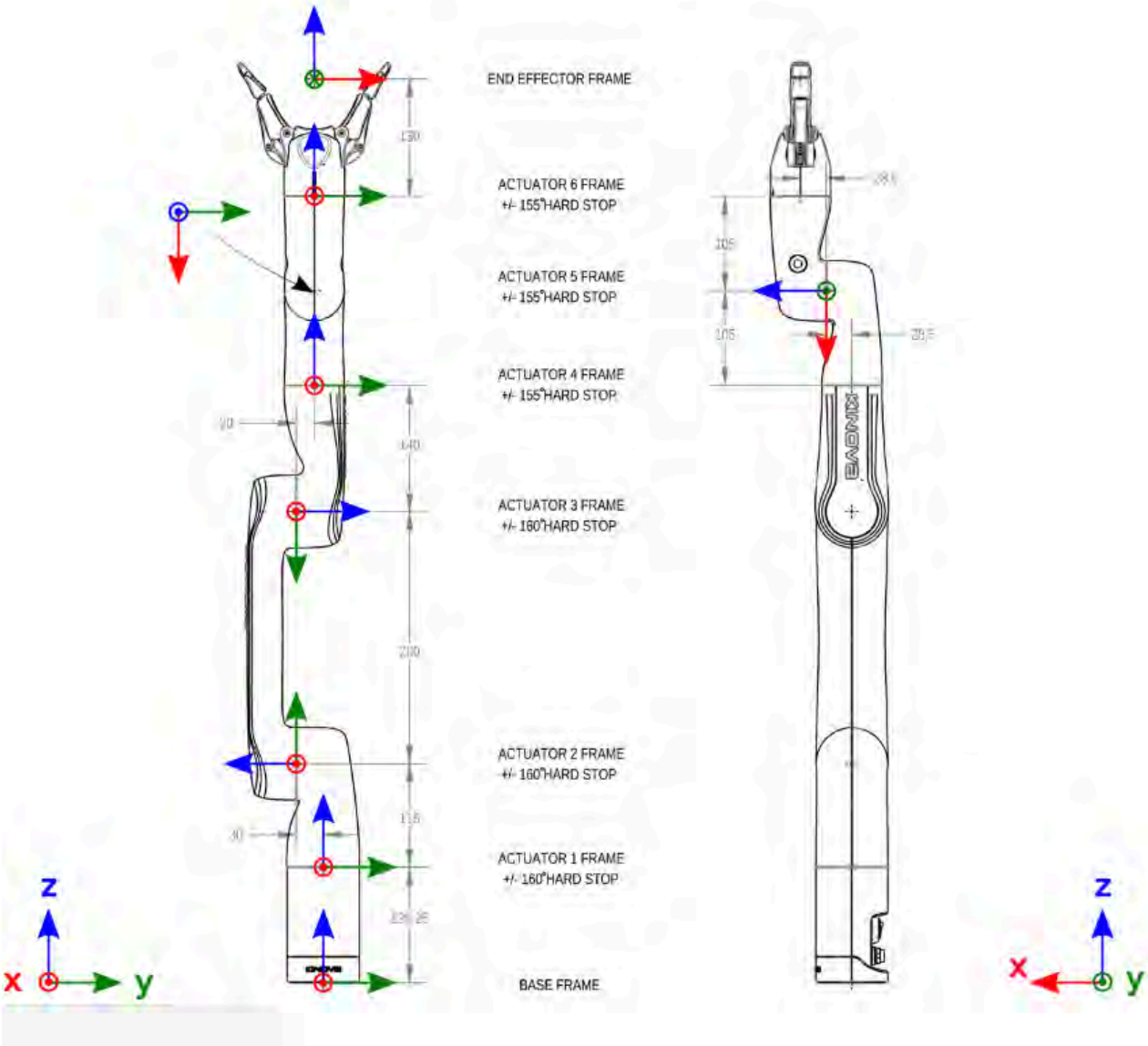
$$R_{z}(q_{i}) = \begin{bmatrix} cq_{i} & -sq_{i} & 0 & 0 \\ sq_{i} & cq_{i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Homogeneous transform matrices

This section is a reference for the homogeneous transform matrices of the robot.

© Kinova inc 2021. All rights reserved.



© Kinova inc 2021. All rights reserved.

Table 49: Transformation matrices

Transformation	$^{i-1}T_i$	$^{i-1}T^*_i$
Base to frame 1	$^{B}T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1283 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$^{B}T^*_1 = \begin{bmatrix} cq_1 & -sq_1 & 0 & 0 \\ sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & 0.1283 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 1 to frame 2	$^{1}T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.0300 \\ 0 & 1 & 0 & 0.1150 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$^{1}T^*_2 = \begin{bmatrix} cq_2 & -sq_2 & 0 & 0 \\ 0 & 0 & -1 & -0.0300 \\ sq_2 & cq_2 & 0 & 0.1150 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 2 to frame 3	$^{2}T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0.2800 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$^{2}T^*_3 = \begin{bmatrix} cq_3 & -sq_3 & 0 & 0 \\ -sq_3 & -cq_3 & 0 & 0.2800 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 3 to frame 4	$^{3}T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.1400 \\ 0 & 1 & 0 & 0.0200 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$^{3}T^*_4 = \begin{bmatrix} cq_4 & -sq_4 & 0 & 0 \\ 0 & 0 & -1 & -0.1400 \\ sq_4 & cq_4 & 0 & 0.0200 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 4 to frame 5	$^{4}T_5 = \begin{bmatrix} 0 & 0 & 1 & 0.0285 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.1050 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$^{4}T^*_5 = \begin{bmatrix} 0 & 0 & 1 & 0.0285 \\ sq_5 & cq_5 & 0 & 0 \\ -cq_5 & sq_5 & 0 & 0.1050 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 5 to frame 6	$^{5}T_6 = \begin{bmatrix} 0 & 0 & -1 & -0.1050 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0.0285 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$^{5}T^*_6 = \begin{bmatrix} 0 & 0 & -1 & -0.1050 \\ sq_6 & cq_6 & 0 & 0 \\ cq_6 & -sq_6 & 0 & 0.0285 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 6 to gripper	$^{6}T_{EE} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1300 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

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

Denavit-Hartenberg (DH) parameters

This section describes the DH parameters for the robot.

© Kinova inc 2021. All rights reserved.

DH parameters

Denavit-Hartenberg (DH) parameters offer another convenient way to specify the reference frame transformations for the robot kinematic chain.

The Classical DH parameters for the transformation between frames are defined as:

$$^{i-1}T_i = \begin{bmatrix} cos(\theta_i) & -cos(\alpha_i)sin(\theta_i) & sin(\alpha_i)sin(\theta_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\alpha_i)cos(\theta_i) & -sin(\alpha_i)cos(\theta_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 63: Classical DH parameters frame definitions

The following table gives the Classical DH parameters for the robot.

Table 50: Classical DH parameters

i	α_i	a_i	d_i	θ_i
1	π / 2	0.0	(128.3 + 115.0)	q_1
2	π	280.0	30.0	q_2 + π / 2
3	π / 2	0.0	20.0	q_3 + π / 2
4	π / 2	0.0	(140.0 + 105.0)	q_4 + π / 2
5	π / 2	0.0	(28.5 + 28.5)	q_5 + π
6	0	0.0	(105.0 + 130.0)	q_6 + π / 2

Here q_i refers to the joint angles.

The DH parameters use the following frames definitions:

© Kinova inc 2021. All rights reserved.

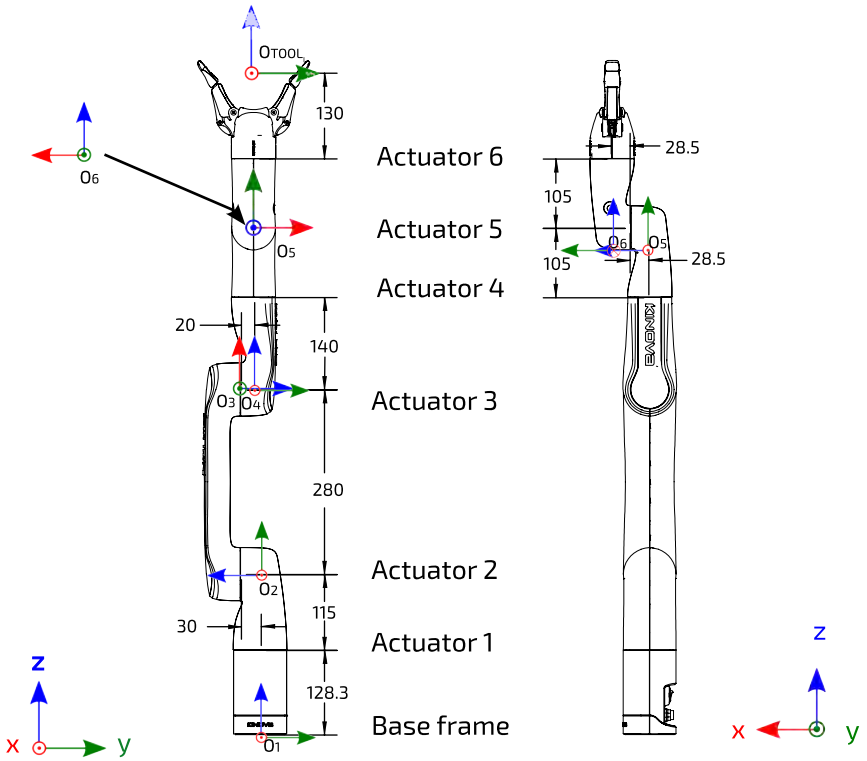


Figure 64: DH parameters frames definitions

Singularity configurations

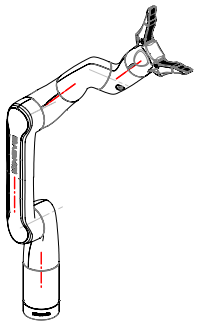
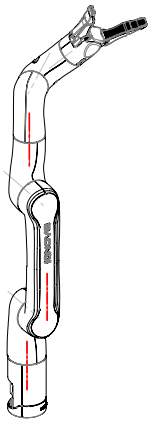
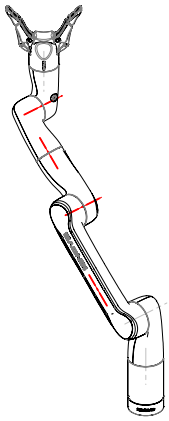
This section describes the singularity configurations of the robot.

Singularity configurations overview

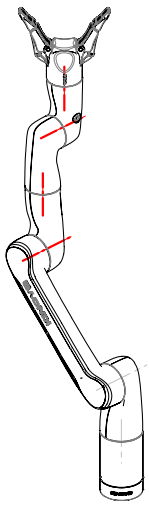
Singularities generally occur when a particular angular configuration of the robot causes axes to be aligned, causing the robot to lose degrees of freedom and experience limitations in movement in some directions while operating the robot in Cartesian mode. There are many ways that this could potentially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

© Kinova inc 2021. All rights reserved.

Table 51: Selected singularity configurations description

Singularity class	Definition	Description
Joint angles shown: [45, 0, 45, 45, 0, 45] 	q2 = 0 and q5 = 0	The axis of joint 1 is aligned with the joint 2 - joint 3 link. The axes of joints 4 and 6 are also aligned.
Joint angles shown: [45, 0, 0, 45, 45, 45] 	q2 = 0 and q3 = 0	There is a 3-way alignment between the axis of joint 1, the joint 2 - joint 3 link, and the axis of joint 4.
Joint angles shown: [45, 45, 0, 90, 45, 45] 	q3 = 0 and q4 = π / 2	There is an alignment between the joint 2 - joint 3 link and the axis of joint 4. The axes of joints 3 and 5 are also aligned.

© Kinova inc 2021. All rights reserved.

Singularity class	Definition	Description
Joint angles shown: [45, 45, 45, 90, 0, 45] 	q4 = π / 2 and q5 = 0	The axes of joints 3 and 5 are aligned. The axes of joints 4 and 6 are aligned.

Inertial parameters definition

Inertial parameters are provided for the robot.

Overview

The following tables describe the key inertial parameters of each independently moving rigid link segment of the 7 robot. This includes:

- mass in kg
- centers of masses in meters
- moments of inertia in kg * m^2

The six distinct moments of inertia (Ixx, Ixy, Ixz, Iyy, Iyz, Izz) are presented in tabular form.

Conventions used

The following conventions are used:

1. The center of mass of a link is always expressed in terms of the reference frame of the precedent joint.
2. The mass of a link segment includes the shell and portions of the actuators at each end of the link (as applicable) that are enclosed within the link and move rigidly with the link.
3. The moments of inertia of the link segments are taken in a frame defined at the center of mass of the link segment and aligned with the precedent joint frame.

Note: These conventions align with those used in URDF files in the `<inertial>` section of `<link>` definitions.

© Kinova inc 2021. All rights reserved.

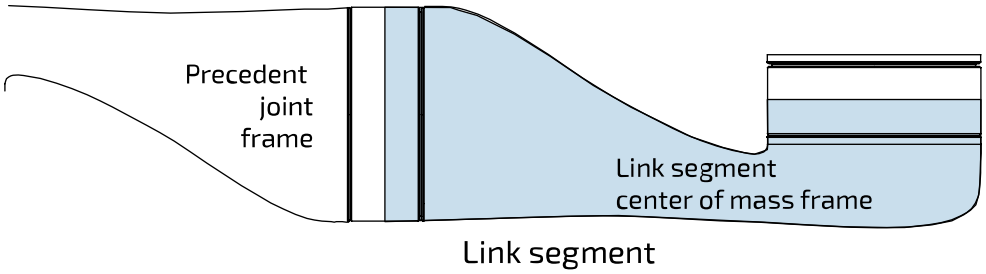


Figure 65: Inertial parameters conventions

Inertial parameters of the robot

This section describes the inertial parameters of the robot.

Overview

The following tables describe the key inertial parameters of the link segments of the robot.

Table 52: Base

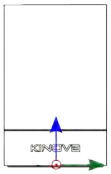
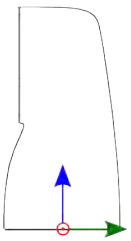
Physical quantity	Value		
mass (kg)	1.1461		
center of mass coordinates (m)	[0.002443, 0.000156, 0.086167]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.0033585	
	Ixy	0.00000039	
	Ixz	0.00010989	
	Iyy	0.00331100	
	Iyz	0.00000191	
	Izz	0.00077158	

Table 53: Link 1

Physical quantity	Value		
mass (kg)	0.9597		
center of mass coordinates (m)	[0.000025, 0.022135, 0.099377]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.0016595	
	Ixy	0.00000002	
	Ixz	0.00000036	
	Iyy	0.0014036	

© Kinova inc 2021. All rights reserved.

Physical quantity	Value		
	Iyz	0.00034927	
	Izz	0.00089493	

Table 54: Link 2

Physical quantity	Value		
mass (kg)	1.1776		
center of mass coordinates (m)	[0.029983, 0.21155, 0.045303]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.0114930	
	Ixy	0.0000010	
	Ixz	0.00000016	
	Iyy	0.0010285	
	Iyz	0.0014077	
	Izz	0.0113350	

Table 55: Link 3

Physical quantity	Value		
mass (kg)	0.59768		
center of mass coordinates (m)	[0.030156, 0.095022, 0.007356]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.0016326	
	Ixy	0.0000071	
	Ixz	0.0000015	
	Iyy	0.0002980	
	Iyz	0.0000960	
	Izz	0.0016909	

Table 56: Link 4

Physical quantity	Value		
mass (kg)	0.52693		
center of mass coordinates (m)	[0.005752, 0.010004, 0.087192]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.00069098	
	Ixy	0.00000024	
	Ixz	0.00016483	

© Kinova inc 2021. All rights reserved.

Physical quantity	Value		
	Iyy	0.00078519	
	Iyz	0.00000074	
	Izz	0.00034115	

Table 57: Link 5

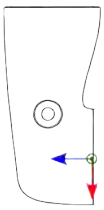
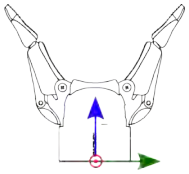
Physical quantity	Value		
mass (kg)	0.58097		
center of mass coordinates (m)	[0.080565, 0.009804, 0.018728]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.00021268	
	Ixy	0.00000521	
	Ixz	0.00000291	
	Iyy	0.0010637	
	Iyz	0.00000011	
	Izz	0.0010847	

Table 58: Gripper

Physical quantity	Value		
mass (kg)	0.2018		
center of mass coordinates (m)	[0.00993, 0.00995, 0.06136]		
moments of inertia ($kg \cdot m^2$)	Ixx	0.0003428	
	Ixy	0.00000019	
	Ixz	0.0000001	
	Iyy	0.00028915	
	Iyz	0.00000027	
	Izz	0.00013076	

© Kinova inc 2021. All rights reserved.

Maintenance and troubleshooting

Maintenance

This sectin describes maintenance tasks for the robot.

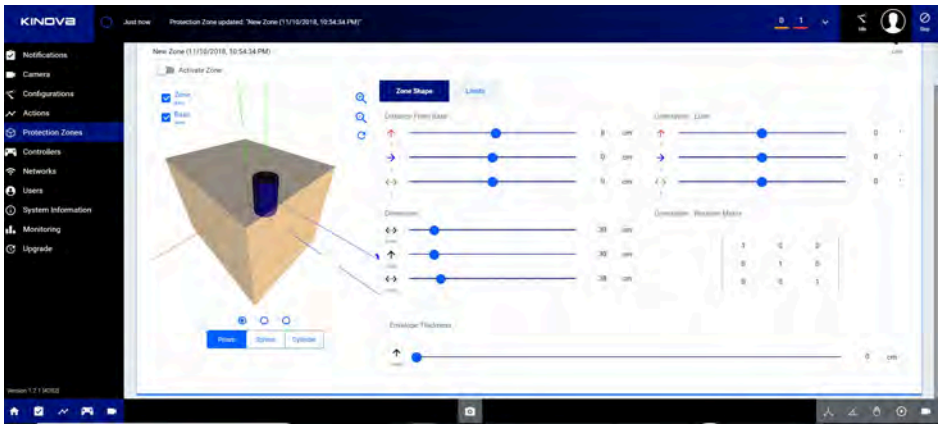
Maintenance overview

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:

- **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™ 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.
- **Cleaning -** Clean the product with a damp cloth and all-purpose household cleaner.

© Kinova inc 2021. All rights reserved.

- **Gripper fingers visual inspection** - Check for signs of wear and replace in case of excessive wear or breakage every six months. Never submerge fingers / gripper when connected to the robot. If necessary, the fingers can be removed with a Phillips T2 screwdriver and cleaned with warm water and mild soap.

Troubleshooting

Tips for troubleshooting robot issues.

If the robot loses a part (for example a shell due to impact) or if a part breaks, shut down the robot safely and leave it off. Contact Kinova technical support.

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
- release notes

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 Xbox gamepad can be used to clear faults - press the left bumper once and proceed.

Consult the *Web App*:

- Check the Monitoring page for high-level status information on various components.
- Check the Notifications page for any recent notifications.
- Check the 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.

As part of periodic software updates, Kinova will publish release notes on the Kinova website. These notes describe known issues, limitations, and workarounds, as well as information about new features and previous bugs fixed in the release.

If all else fails, try rebooting the robot.

© Kinova inc 2021. All rights reserved.

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

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 App*. The following tables give more guidance as to the source of the problem when a safety threshold is triggered.

Safeties handling details

Table 59: 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 - internal Ethernet or power cable • Base EEPROM issue
Network Initialization Error	• Base CPU board issue • Joint 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 App* emergency stop button clicked

© Kinova inc 2021. All rights reserved.

Safety	Most Probable Cause
Emergency Line Asserted	• Joint not programmed • Joint in a boot loop • Electrical component failure
Inrush Current Limiter Fault	• Payload exceeded • Electrical component failure

Table 60: Actuators safeties handling

Safety	Most Probable Cause
Following error	• Communication issue • Firmware issue
Maximum velocity	• Communication issue • Firmware issue
Magnetic position	• Magnet improperly glued
Hall position	• Hall sensor major malfunction
Hall sequence	• Hall sensor major malfunction(s)
Maximum motor current	• Shorted phases • Bad motor • Payload exceeded
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 is 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 our GitHub repositories:

- KINOVA® KORTEX™ GitHub: github.com/kinovarobotics/kortex
- KINOVA® KORTEX™ ROS GitHub: github.com/kinovarobotics/ros_kortex

© Kinova inc 2021. All rights reserved.

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

kinovarobotics.com

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

Kinova Europe GmbH

Innovation

Großkitzighofer Straße 7a
86853 Langerringen
T: +49 (0)8248 8887928

Kinova Europe GmbH

Assistive Technologies

Vorderbreitenthann 150
91555 Feuchtwangen,
Germany
T: 0800 5466822

UG-017_R02

KINOVA
Achieve Extraordinary