DESCRIPTION

Compute Box
Robot Controller Interface

Edition E6

OnRobot Compute Box Version 3.1.3

June 2018

1 Content

Information provided within this document is accurate to the best of our knowledge at the time of its publication. There may be differences between this document and the product if the product has been modified after the edition date.

OnRobot A/S. does not assume any responsibility for any errors or omissions in this document. In no event shall OnRobot A/S. be liable for losses or damages to persons or property arising from the use of this document.

The information within this document is subject to change without notice. You can find the latest version on our webpage at: **https://onrobot.com/**.

The original language for this publication is English. Any other languages that are supplied have been translated from English.

2 Preface

2.1 Target Audience

This document is intended for robot manufacturers who intend to use the OnRobot Sensor together with a robot, currently not supported by OnRobot. Personnel working with the integration are expected to have the following expertise:

Basic knowledge of programming principles

Basic knowledge of TCP/IP communication

Basic knowledge of electronic and electrical systems

Basic knowledge of the robot system

2.2 Intended Use

The Compute Box is designed to perform calculations based on force and torque measurements from the OnRobot sensor, positional information from the robot system, and commands requested on the Compute Box Robot Controller Interface (CBRCI). The results of the calculations are speed values sent to control the robot system. OnRobot is not liable for any damage or injury resulting from misuse.

2.3 Typographic Conventions

The following typographic conventions are used in this document.

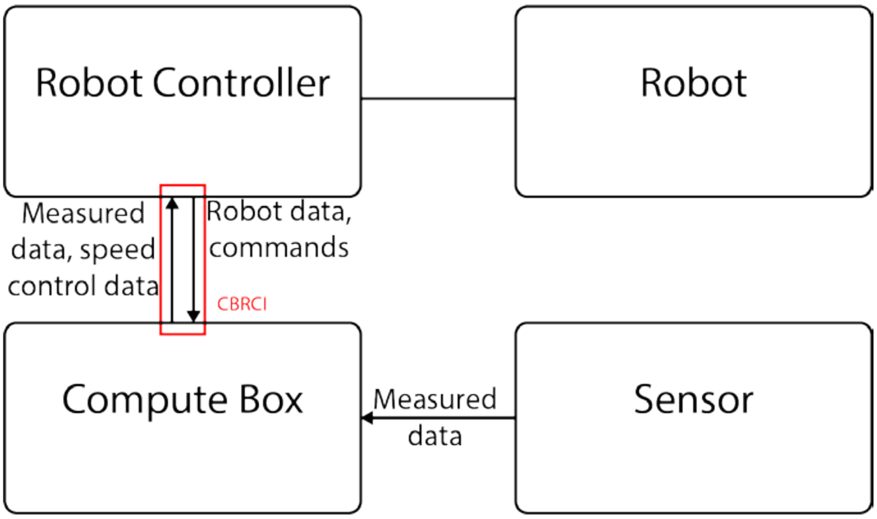
Table 1: Conventions

`Courier Text`	File paths and file names, code, user input and computer output.
Italicized text	Citations and marking image callouts in text.
Bold text	UI elements, including text appearing on buttons and menu options.
Bold, dark red text	External links, or internal cross-references.
<angle brackets>	Variable names that must be substituted by real values or strings.
1. Numbered lists	Steps of a procedure.
A. Alphabetical lists	Image callout descriptions.

3 Introduction

3.1 Robot and Sensor System Architecture

The image below shows the robot and sensor system architecture.



The sensor measures the force and torque data and transfers this data to the Compute Box. The Compute Box also receives data from the Robot controller about the current robot position. After receiving a command, the Compute Box performs the calculations necessary to provide instructions to the robot controller, to perform the movements that implement the commands sent by the robot controller. The instructions come in the format of speed and rotational speed values that the Robot must follow.

3.2 Conventions and Default Behavior

The CBRCI uses TCP over IP over Ethernet protocol stack for communication with the robot controller. It is highly recommended to connect the Robot Controller and the Compute Box directly, without any networking equipment, such as switches or routers.

If one part of the interface cannot establish a connection, close the connection on all interfaces.

4 Compute Box Discovery

To make connection configuration between the Compute Box and the Robot Controller easier, it is possible to ask the Compute Box for information about the sensor.

To find out the IP addresses of all Compute Boxes on the subnet that the Robot Controller is attached to, send out a UDP broadcast message with the following attributes:

Port number: 55660

Contents of the message: “OFD\0\n”

The Compute Box replies with a message. The IP address of the Compute Box can be extracted from the UDP message answer. The information contained in the message is described in the table below:

Byte Number	**Included information**
0-15	The hostname “COMPUTEBOX”
16-17	“\r\n”
18-34	MAC address of the Compute Box. Example: “B8-27-EB-DB-E6-5F”)
35-36	“\r\n”
37-52	Serial Number of the sensor. Example: “HEXEA123” (or “NAN” if no sensor is attached to the Compute Box)
53-54	“\r\n”
55-70	Reserved
71-72	“\r\n”
73-98	Description of the sensor “Single-channel 6 axis DAQ” (or “DAQ not found (timeout)” if no sensor is attached to the Compute Box)
99-100	“\r\n”

5 Using the Interface

The interface is divided into two parts, each with a different function.

5.1 Real-Time Interface

The Real-Time interface serves the purpose of transferring robot data, sensor data, and robot control instructions between the robot system and the Compute Box.

All numerical data on the Real-Time interface is expected and sent in network byte order.

The TCP port number for the Real-Time interface is 32000.

Only one TCP connection is allowed at the same time.

The following sections describe the functionality of the Real-Time Interface.

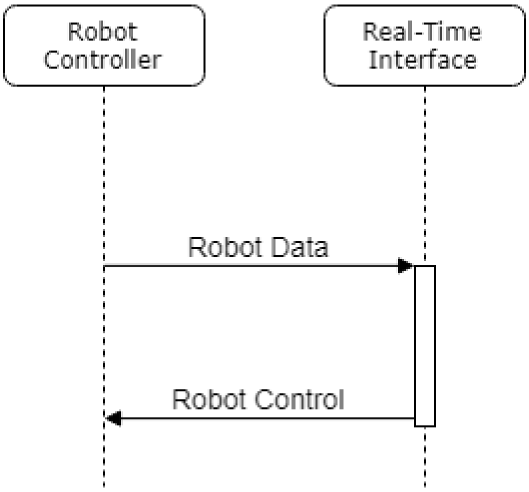
5.1.1 Establishing and Closing Connection

To allow data transfer to take place on the Real-Time interface, the user must establish a TCP connection with the interface first. If the robot program finishes, the connection must be closed.

5.1.2 Robot Data, Robot Control

The Compute Box calculates the needed speeds based on the current position of the robot. The Robot Data message is used to transfer the robot position data to the Compute Box, and the Robot Control message is used to transfer the speeds back to the robot controller, as an answer message.

If a second Robot Data message is sent to the Compute Box, before receiving the Robot Control message answer for the first Robot Data message, the messages are buffered, and the answer Robot Control messages will be sent in order, until all Robot Data messages have been answered. To avoid buffering, it is recommended to send Robot Data messages no more often than every 4 ms, that is, robot cycle time is recommended to be equal to, or higher than 4 ms.



The format of the Robot Data message is described in the table below.

Robot Data	Data Type	Number of Values	Size in bytes	Comment
Response window	byte	1	1	Not used
Robot cycle time	byte	1	1	Time between 2 consecutive Robot Data messages in ms
Robot state	byte	1	1	0: OFF 1: INIT 2: ON
Program state	byte	1	1	0: STOP 1: IDLE 2: RUNNING
Tool Center Point position	Float32	6	24	Position: Vector 3D Rotation: Rotational Vector
Tool Center Point offset	Float32	6	24	Position: Vector 3D Rotation: Rotational Vector
Sensor offset	Float32	6	24	Position: Vector 3D Rotation: Rotational Vector

The length of the Robot Data message is 76 bytes.

The format of the Robot Control message is described in the table below.

Control Data	Data Type	Number of Values	Size in bytes	Comment
Terminate flag	byte	1	1	Reserved
Command id	integer	1	4	Reserved
Command Result	byte	1	1	Reserved
Sensor status	integer	1	4	Sensor status
Speeds	Float32	6	24	mm/s, rad/s
Sensor values	Float32	6	24	N, Nm

The length of the Robot Control message is 58 bytes.

The Speeds and Sensor values data are sent only if both the Robot State and Program state data are sent as 2. In all other cases both the Speeds and the Sensor values sent are set to 0.

The meaning of the Sensor status bits is described in the table below, 0 is the least significant bit, and 17 is the most significant bit.

Bit	Function
First 16 bits are all 1 (Status word is 65535)	No sensor is attached
0-3	Reserved
4	OVERLOAD – in Fx
5	OVERLOAD – in Fy
6	OVERLOAD – in Fz
7	OVERLOAD – in Tx
8	OVERLOAD – in Ty
9	OVERLOAD – in Tz
10-11	Sensor Failure
12	Reserved
13	Sensor power or EEPROM error
14	Communication error between the sensor and the Compute Box
15	Reserved
16	Communication timeout error between the sensor and the Compute Box, during operation
17	Communication timeout error between the sensor and the Compute Box, after start

5.2 Command Interface

The Command Interface serves the purpose of sending commands to the Compute Box. All commands are non-blocking, that is, while one command is executed, other commands are still processed and executed by the Compute Box.

If there is no connection on the Real-Time interface, the Compute Box only process the following commands on the Command Interface:

Configuration

Start Recording

Stop Recording

Store Cached Path

Get Path List

Delete Path

The TCP port number for the Command Interface is 32002.

More than one client can establish connection at the same time, the messages will be answered to the appropriate client.

The messages are JSON formatted, with two new lines at the message end.

Syntax: {JSON DATA}\r\n\r\n

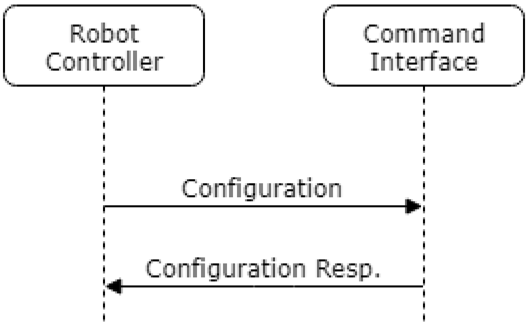
The following sections describe the messages on the Command Interface.

5.2.1 Establishing and Closing Connection

To allow data transfer to take place on the Command Interface, the user must establish a TCP connection with the interface first. If the robot program finishes, the connection must be closed.

5.2.2 Configuration

The Robot Config message is used to transfer general data to the Compute Box, about the robot, and configure the sensor. If the Robot Config message is not sent, no data transfer can take place on any of the two parts of the CBRCI, only connection setup.



The format of the Configuration message is described below:

```
{"message_id":<string>,
"command":
{"id":"configuration",
"robot_cycle":<integer>,
"sensor_cycle":<integer>,
"max_translational_speed":<floating point>,
"max_rotational_speed":<floating point>,
"max_translational_acceleration":<floating point>,
"max_rotational_acceleration":<floating point>}}
```

The properties of the data included in the Configuration message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory

id	Identity of the command, value: “configuration”	N/A	N/A	Mandatory
robot_cycle	The robot cycle time	ms	>0	Mandatory
sensor_cycle	the interval at which the sensor sends data	ms	2-254	Mandatory
max_translational_speed	the maximum translational speed of the robot tool	mm/2	>0	Mandatory
max_rotational_speed	the maximum rotational speed of the robot tool	rad/s	>0	Mandatory
max_translational_acceleration	The maximum translational acceleration of the robot tool	mm/s^2	>0	Mandatory
max_rotational_acceleration	The maximum rotational acceleration of the robot tool	rad/s^2	>0	Mandatory

It is recommended to set the sensor read-out interval to a lower value than the robot cycle time.

5.2.2.1 CONFIGURATION RESPONSE

The format of the Configuration Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"configuration"}}
```

The properties of the data included in the Configuration Response message is described in the table below.

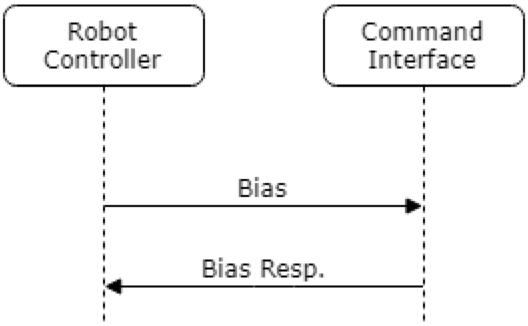
Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “configuration”	N/A	N/A	Mandatory

The possible result codes of the Configuration Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.3 Bias

The Bias message is used to bias (zero) the force and torque reading.



The format of the Bias message is described below:

{"message_id":<string>,
"command":
{"id":"bias”}}

The properties of the data included in the Bias message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “bias”	N/A	N/A	Mandatory

Do not bias the sensor during robot movement, or when the tool is in contact with a solid object. It is recommended to bias the sensor when the tool has picked up or put down an object, or it has changed direction with a significant load.

5.2.3.1 BIAS RESPONSE

The format of the Bias Response message is described below:

{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"bias"}}

The properties of the data included in the Bias Response message is described in the table below.

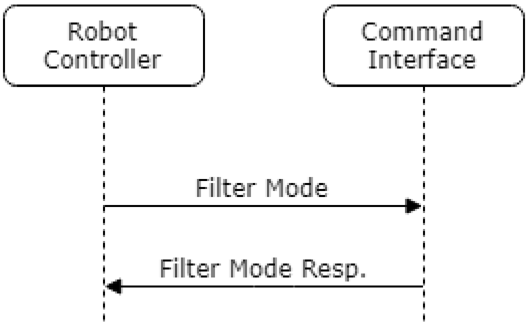
Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “bias”	N/A	N/A	Mandatory

The possible result codes of the Bias Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.4 Filter Mode

The Filter Mode message is used to set the sensor filter cutoff frequency.



The format of the Filter Mode message is described below:

{"message_id":<string>,
"command":
{"id":"filter_mode”,
"filter_mode":<integer>}}

The properties of the data included in the Filter Mode message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “filter_mode”	N/A	N/A	Mandatory
filter_mode	The setting for the sensor filter mode	N/A	0-6	Mandatory

The possible sensor filter mode values are described in the table below.

Value	Explanation
0	No filter
1	500 Hz
2	150 Hz
3	50 Hz
4	15 Hz
5	5 Hz
6	1.5 Hz

5.2.4.1 FILTER MODE RESPONSE

The format of the Filter Mode Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"filter_mode"}}
```

The properties of the data included in the Filter Mode Response message is described in the table below.

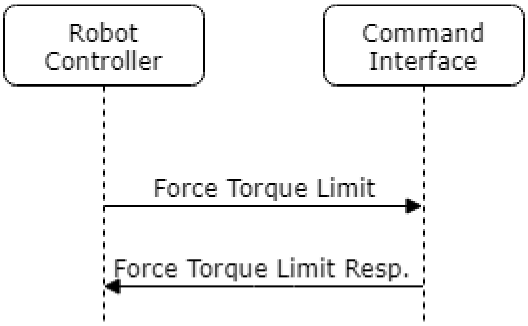
Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “filter_mode”	N/A	N/A	Mandatory

The possible result codes of the Filter Mode Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.5 Force Torque Limit

The Force Torque Limit message is used to stop the robot, if the measured force/torque values have exceeded a set limit.



5.2.5.1 ENABLE FORCE TORQUE LIMIT

The format of the Force Torque Limit message for enabling force/torque limiting is described below:

```
{"message_id":<string>,
"command":
```

```
{"id":"force_torque_limit",
"force_torque_limit":[<float>,<float>,<float>,<float>,<float>,<float>],
"force_magnitude_limit":<float>,
"torque_magnitude_limit":<float>,
"use_absolute_values":[<boolean>,<boolean>,<boolean>,<boolean>,<boolea
n>,<boolean>],
"frame_type":<integer >,
"frame_modifier":[<float>,<float>,<float>,<float>,<float>,<float>]}}
```

The properties of the data included in the Force Torque Limit message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “force_torque_limit”	N/A	N/A	Mandatory
force_torque_limit	The force and torque component values above which the robot will stop, sending 0 for any component turns off limiting force or torque for the respective axis	N, Nm	N/A	Mandatory
force_magnitude_limit	The overall force limit above which the robot will stop	N	N/A	Optional
torque_magnitude_limit	The overall torque limit above which the robot will stop	Nm	N/A	Optional
use_absolute_values	The settings for which force components are regarded as absolute values	N/A	N/A	Mandatory
frame_type	The coordinate system in which the forces will be measured	N/A	0-1	Mandatory
frame_modifier	Coordinates of a translational vector and a rotational vector by which the coordinate system is modified	mm, rad	N/A	Optional

The possible frame type values are described in the table below.

Value	Explanation
0	Base
1	Tool

5.2.5.2 DISABLE FORCE TORQUE LIMIT

The format of the Force Torque Limit message for disabling force/torque limiting is described below:

```
{"message_id":<string>,
"command":
{"id":"force_torque_limit"}}
```

The properties of the data included in the Configuration message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional

command		N/A	N/A	Mandatory
id	Identity of the command, value: “force_torque_limit”	N/A	N/A	Mandatory

5.2.5.3 FORCE TORQUE LIMIT RESPONSE

The format of the Force Torque Limit Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"force_torque_limit"}}
```

The properties of the data included in the Force Torque Limit Response message is described in the table below.

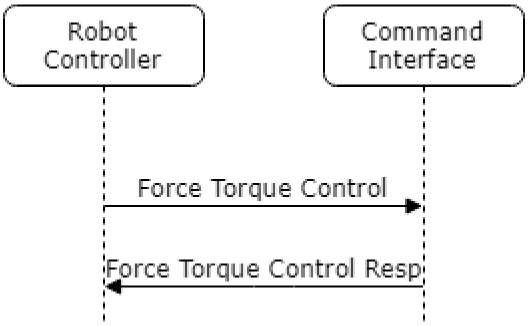
Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message,	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “force_torque_limit”	N/A	N/A	Mandatory

The possible result codes of the Force Torque Limit Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.6 Force Torque Control

The Force Torque Control message is used to keep the set force/torque values constant along/about the axes set to be compliant.



5.2.6.1 ENABLE FORCE TORQUE CONTROL

The format of the Force Torque Control message for enabling force/torque controlling is described below:

```
{"message_id":<string>,
"command":
{"id":"force_torque_control",
"force_torque":[<float>,<float>,<float>,<float>,<float>,<float>],
"frame_type":<integer>,
"frame_modifier":[<float>,<float>,<float>,<float>,<float>,<float>],
"proportional_gain":[<float>,<float>,<float>,<float>,<float>,<float>],
"integral_gain":[<float>,<float>,<float>,<float>,<float>,<float>],
" derivative_gain":[<float>,<float>,<float>,<float>,<float>,<float>],
"enabled_axes":[<boolean>,<boolean>,<boolean>,<boolean>,<boolean>,<boolean>],
"force_square_threshold":<float>,
"torque_square_threshold":<float>,
"max_translational_speed_by_control":<float>,
"max_rotational_speed_by_control":<float>}}
```

The properties of the data included in the Enable Force Torque Control message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “force_torque_control”	N/A	N/A	Mandatory
force_torque		N/A	N/A	Optional, default value: 0
frame_type	The coordinate system in which the forces will be measured	N/A	0-1	Optional, default value: 0
frame_modifier	Coordinates of a translational vector and a rotational vector by which the coordinate system is modified	mm, rad	N/A	Optional, default value: 0
proportional_gain	the proportional gains of the force and torque controller, one for each axis	N/A	N/A	Optional, default value: 1
integral_gain	the integral gains of the force and torque controller, one for each axis	N/A	N/A	Optional, default value: 0
derivative_gain	the derivative gains of the force and torque controller, one for each axis	N/A	N/A	Optional, default value: 0
enabled_axes	set which axes are force and torque controlled	N/A	N/A	Optional, default value: false
force_square_threshold	the force value below which the measured force values are replaced with the square of the value for the force controller, that is, the force	N	N/A	Optional, default value: 0

	controller behavior will change; if the value is set to 0, square power blending will not be used			
torque_square_threshold	the torque value below which the measured torque values are replaced with the square of the value for the torque controller, that is, the torque controller behavior will change; if the value is set to 0, square power blending will not be used	Nm	N/A	Optional, default value: 0
max_translational_speed_by_control	the maximum translational speed that the force controller can move the robot, if this value is set to 0, the translational speed is not limited	mm/s	N/A	Optional, default value: 0
max_rotational_speed_by_control	the maximum rotational speed that the torque controller can move the robot, if this value is set to 0, the rotational speed is not limited	rad/sec	N/A	Optional, default value: 0

The possible frame type values are described in the table below.

Value	Explanation
0	Base
1	Tool

It is recommended to set the thresholds for square power blending to a low value.

5.2.6.2 DISABLE FORCE TORQUE CONTROL

The format of the Force Torque Control message for disabling force/torque controlling is described below:

```
{"message_id":<string>,
"command":
{"id":"force_torque_control"}}
```

The properties of the data included in the Force Torque Control message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “force_torque_control”	N/A	N/A	Mandatory

5.2.6.3 FORCE TORQUE CONTROL RESPONSE

The format of the Force Torque Control Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"force_torque_control"}}
```

The properties of the data included in the Force Torque Control Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: "force_torque_control"	N/A	N/A	Mandatory

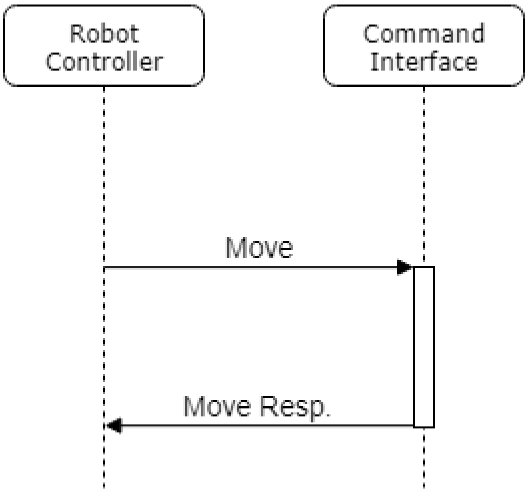
The possible result codes of the Force Torque Control Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.7 Move

The Move message is used to specify the path that the Compute Box will command the robot to move along.

The Move Result message is used to answer the Move command, either when the robot has reached the final position, or if it exited due to exceeding any set limit.



5.2.7.1 MOVE WITH POINTS

The format of the Move message with Points is described below:

```
{"message_id":<string>,
"command":
{"id":" move",
"speed":<float>,
"acceleration":<float>,
"points":[<float>,<float>,<float>,<float>,<float>,<float>],[<float>,<float>,<float>,<float>,<float>,<float>],[<float>,<float>,<float>,<float>,<float>,<float>]]}
```

The properties of the data included in the Move message with Points is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “move”	N/A	N/A	Mandatory
speed	the translational speed of the robot	mm/s	>0	Mandatory
acceleration	the translational acceleration of the robot	mm/s^2	>0	Mandatory
points	Coordinates of the point by a translational vector and a rotational vector	mm, rad	N/A	Mandatory

The number of points can be 1-250. Do not send 2 consecutive points with the same x,y,z coordinates. The first point of the Force Move can be in the same x,y,z coordinates as the current tool position.

The position controller is only used in case there is only one point included in the move command.

5.2.7.2 MOVE WITH PATH

The format of the Move message with Path is described below:

```
{"message_id":<string>,
"command":
{"id":" move",
"speed":<float>,
"acceleration":<float>,
"path_id":<integer>}}
```

The properties of the data included in the Move message with Path is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “move”	N/A	N/A	Mandatory
speed	the translational speed of the robot	mm/s	>0	Mandatory
acceleration	the translational acceleration of the robot	mm/s^2	>0	Mandatory
path_id	The identity of a Path, -1 plays the path from the memory.	N/A	1000-9999	Mandatory

Replaying a path moves the Tool through all points stored in the Path file (or memory) in a curved path. The maximum error of the replayed Path compared to the original recorded movement is 1 mm.

5.2.7.3 MOVE RESPONSE

The format of the Move Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":" move"}}
```

The properties of the data included in the Move Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	-14 - 12	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “move”	N/A	N/A	Mandatory

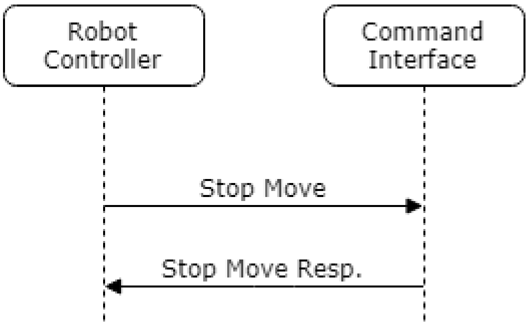
The possible result codes of the Move Response message are described in the table below.

Result Value	Description
4	Move stopped by the Stop command
9	Move finished
12	Move stopped by Force Torque Limit
-1	Error
-11	Path file not found (not applicable for Move with Points)
-12	Empty Path (not applicable for Move with Points)
-13	Empty Path file (not applicable for Move with Points)
-14	Path file cannot be opened (not applicable for Move with Points)

5.2.8 Stop Move

The Stop Move message is used to stop an ongoing Move command.

The Stop Move Result message is used to answer the Stop Move command.



The format of the Stop Move message is described below:

{"message_id":<string>,
"command":
{"id":"stop_move"}}

The properties of the data included in the Stop Move message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: "stop_move"	N/A	N/A	Mandatory

5.2.8.1 STOP MOVE RESPONSE

The format of the Stop Move Response message is described below:

{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"stop_move"}}

The properties of the data included in the Stop Move Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: "stop_move"	N/A	N/A	Mandatory

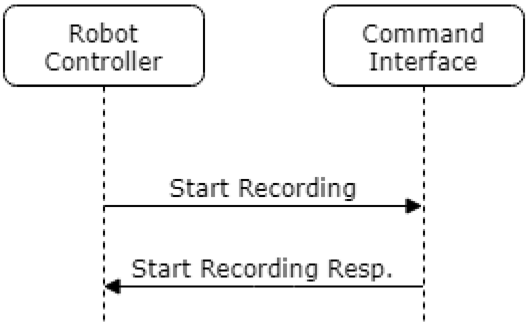
The possible result codes of the Stop Move Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.9 Start Recording

The Start Recording message is used to start recording a Path in the memory of the Compute Box. The recording stores the position of the robot into the memory of the Compute Box as received inside the Robot Data messages. If the Real-Time Interface connection is not established, no points are recorded. If the Real-Time Interface connection is established, and Robot Data messages are sent, the recording of points will start.

The Start Recording Result message is used to answer the Start Recording command.



The format of the Start Recording message is described below:

{"message_id":<string>,
"command":
{"id":"start_recording"}}

The properties of the data included in the Start Recording message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “start_recording”	N/A	N/A	Mandatory

5.2.9.1 START RECORDING RESPONSE

The format of the Start Recording Response message is described below:

{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"start_recording"}}

The properties of the data included in the Start Recording Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: "start_recording"	N/A	N/A	Mandatory

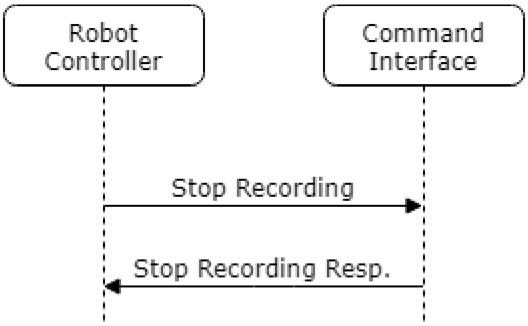
The possible result codes of the Start Recording Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.10 Stop Recording

The Stop Recording message is used to stop recording a Path. After finishing the recording, the points recorded throughout the movement are filtered by an algorithm to remove stops within the Path. These points are still only stored in the memory of the Compute Box.

The Stop Recording Result message is used to answer the Stop Recording command.



The format of the Stop message is described below:

{"message_id":<string>,
"command":
{"id":"stop_recording"}}

The properties of the data included in the Stop Recording message is described in the table below.

Name	Description	Dimension	Range	Type

message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “stop_reccording”	N/A	N/A	Mandatory

5.2.10.1STOP RECORDING RESPONSE

The format of the Stop Recording Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"stop_recording",

"start":[<float>,<float>,<float>,<float>,<float>,<float>]}}
```

The properties of the data included in the Stop Recording Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “stop_recording”	N/A	N/A	Mandatory
start	First point of the Paths saved in the Compute Box. If the Path has no points in it, this attribute is not sent	mm, rad	N/A	Optional

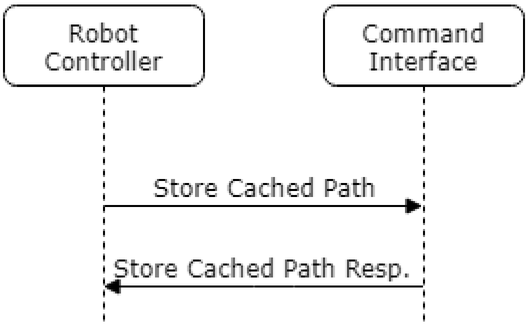
The possible result codes of the Stop Recording Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.11 Store Cached Path

The Store Cached Path message is used to save a Path from the memory of the Compute Box to permanent storage.

The Store Cached Path Result message is used to answer the Store Cached Path command.



The format of the Store Cached Path message is described below:

{"message_id":<string>,
"command":
{"id":"store_cached_path”}}

The properties of the data included in the Store Cached Path message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “store_cached_path”	N/A	N/A	Mandatory

5.2.11.1STORE CACHED PATH RESPONSE

The format of the Store Cached Path Response message is described below:

{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"store_cached_path",
"path_id":<integer>}}

The properties of the data included in the Store Cached Path Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “store_cached_path”	N/A	N/A	Mandatory
path_id	The identity of the path stored	N/A	1000-9999	Mandatory

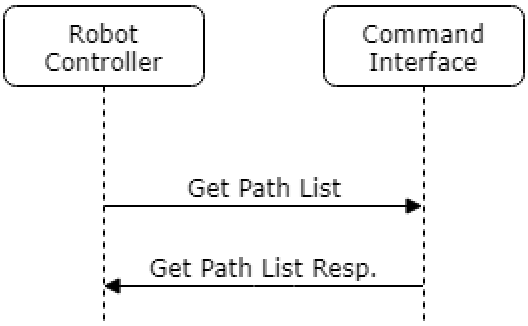
The possible result codes of the Store Cached Path Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.12 Get Path List

The Get Path List message is used to request the Compute Box to send the list of stored Paths along with the first point of each Path.

The Get Path List Result message is used to answer the Stop command.



The format of the Get Path List message is described below:

```
{"message_id":<string>,
"command":
{"id":"get_path_list"}}
```

The properties of the data included in the Get Path List message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: "get_path_list"	N/A	N/A	Mandatory

5.2.12.1 GET PATH LIST RESPONSE

The format of the Get Path List Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":" get_path_list ",
"path_list":[<integer>,<integer>,<integer>],
"start_list":[<float>,<float>,<float>,<float>,<float>,<float>],[<float>,
```

<float>,<float>,<float>,<float>,<float>],[<float>,<float>,<float>,<float>,<float>,<float>]}}

The properties of the data included in the Get Path List Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “get_path_list”	N/A	N/A	Mandatory
path_list	List of the Paths stored in the Compute Box	N/A	1000-9999	Mandatory
start_list	List of the first points of the Paths saved in the Compute Box.	mm, rad	N/A	Mandatory

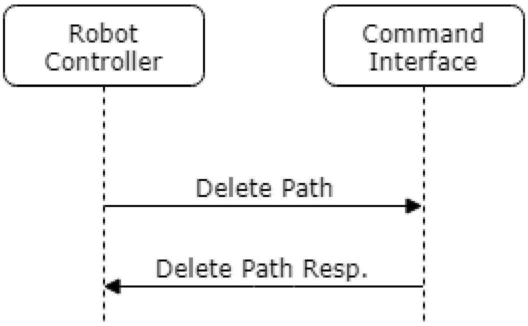
The possible result codes of the Get Path List Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.2.13 Delete Path

The Delete Path message is used to delete one path from the Compute Box.

The Delete Path Result message is used to answer the Delete Path command.



The format of the Delete Path message is described below:

{"message_id":<string>,
"command":
{"id":"delete_path”}}

The properties of the data included in the Delete Path message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	The identity of the specific message and its response	N/A	N/A	Optional
command		N/A	N/A	Mandatory
id	Identity of the command, value: “delete_path”	N/A	N/A	Mandatory

5.2.13.1DELETE PATH RESPONSE

The format of the Delete Path Response message is described below:

```
{"message_id":<string>,
"result":<integer>,
"error":<string>,
"response":
{"id":"delete_path"}}
```

The properties of the data included in the Delete Path Response message is described in the table below.

Name	Description	Dimension	Range	Type
message_id	Contains the value of the message_id of the Configuration message, if there was no such data sent, it is an empty string	N/A	N/A	Conditional,
result	The description of the outcome of the command execution	N/A	0, -1	Mandatory
error	Explanation of the error, can be JSON syntax error, or command execution error	N/A	N/A	Mandatory
response		N/A	N/A	Mandatory
id	Identity of the command, value: “delete_path”	N/A	N/A	Mandatory

The possible result codes of the Delete Path Response message are described in the table below.

Result Value	Description
0	OK
-1	Error

5.3 Recommendations and Command Interactions

It is recommended to set up the Interfaces according to the following procedure:

1. Power OnRobot Controller, Compute Box
2. Connect to Command Interface
3. Send Configuration message
4. Robot Program start
5. Connect to Real-Time Interface

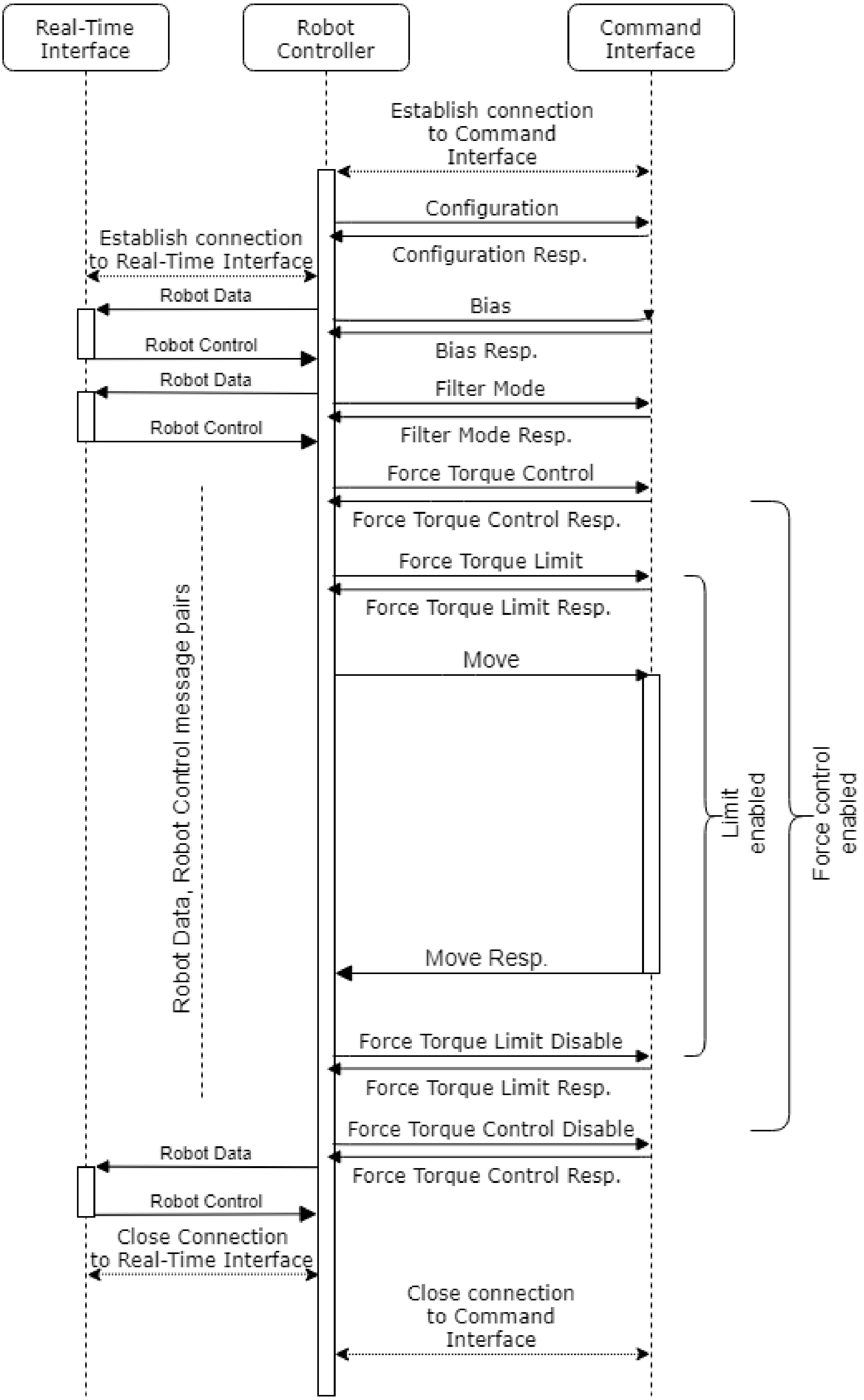
6. Issue commands according to robot program, with possible loop
7. Robot Program end
8. Disconnect from Real-Time Interface
9. Disconnect from Command Interface
10. Power off Robot Controller, Compute Box

If a movement is to be force controlled or limited, it is recommended to send the Force Torque Control and Limit commands before the Move command, and disable both only after the Move Response is received.

5.4 Examples

5.4.1 Controlled, Limited Movement

The following image shows a message flow example, that implements a force-, torque-limited, force-, torque-controlled movement.



This example includes the following flow:

Connection establishment for both parts of the CBRCI and robot configuration information transfer.

Configuring the sensor behavior.

Commanding the robot to perform a force-controlled, limited movement.

Closing connection for both parts of the CBRCI.

Below are examples for messages within the flow described above. Connection establishment, Connection Closing, Robot Data and Robot Control messages are omitted.

{"command":{"id":"configuration","robot_cycle":8,"sensor_cycle":4,"max_translational_speed":1500,"max_rotational_speed":1.75,"max_translation_alacceleration":3000,"max_rotational_acceleration":1.2}}

{"result":0,"error":"","response":{"id":“configuration”}}

{"command":{"id":"bias”}}

{"result":0,"error":"","response":{"id":“bias”}}

{"command":{"id":"filter_mode”,"filter_mode":0}}

{"result":0,"error":"","response":{"id":“filter_mode”}}

{"command":{"id":"force_torque_control","ft":[10,0,0,0,0,0],"frame_type":1,"frame_modifier":[100,0,0,0,0,0],"proportional_gain":[1.2,1.2,1.2,1.5,1.5,1.5],"integrative_gain":[0.2,0.2,0.2,0.3,0.3,0.3],"enabled_axes":[true,true,true,false,false,false],"force_square_threshold":0,"torque_square_threshold":0,"max_translational_speed_by_control":1000,"max_rotational_speed_by_control":0}}

{"result":0,"error":"","response":{"id":“force_torque_control”}}

{"command":{"id":"force_torque_limit","force_torque_limit":[50,50,50,0,0,0],"force_magnitude_limit":60,"torque_magniitude_limit":0,"use_absolute_values":[true,true,false,false,false,false],"enabled_axes":[true,true,true,false,false,false],"frame_type":1,"frame_modifier":[100,0,0,0,0,0]}}

{"result":0,"error":"","response":{"id":“ force_torque_limit”}}

{"command":{"id":"move","speed":50," acceleration":500,"points":[100,-120,42.524,-3.91E-4,0.999843,0.012478,0.012537],[-79.925,-606.86896,230.789,-3.91E-4,0.884561,0.123256,0.011143]}}}

{"result":0,"error":"","response":{"id":“move”}}

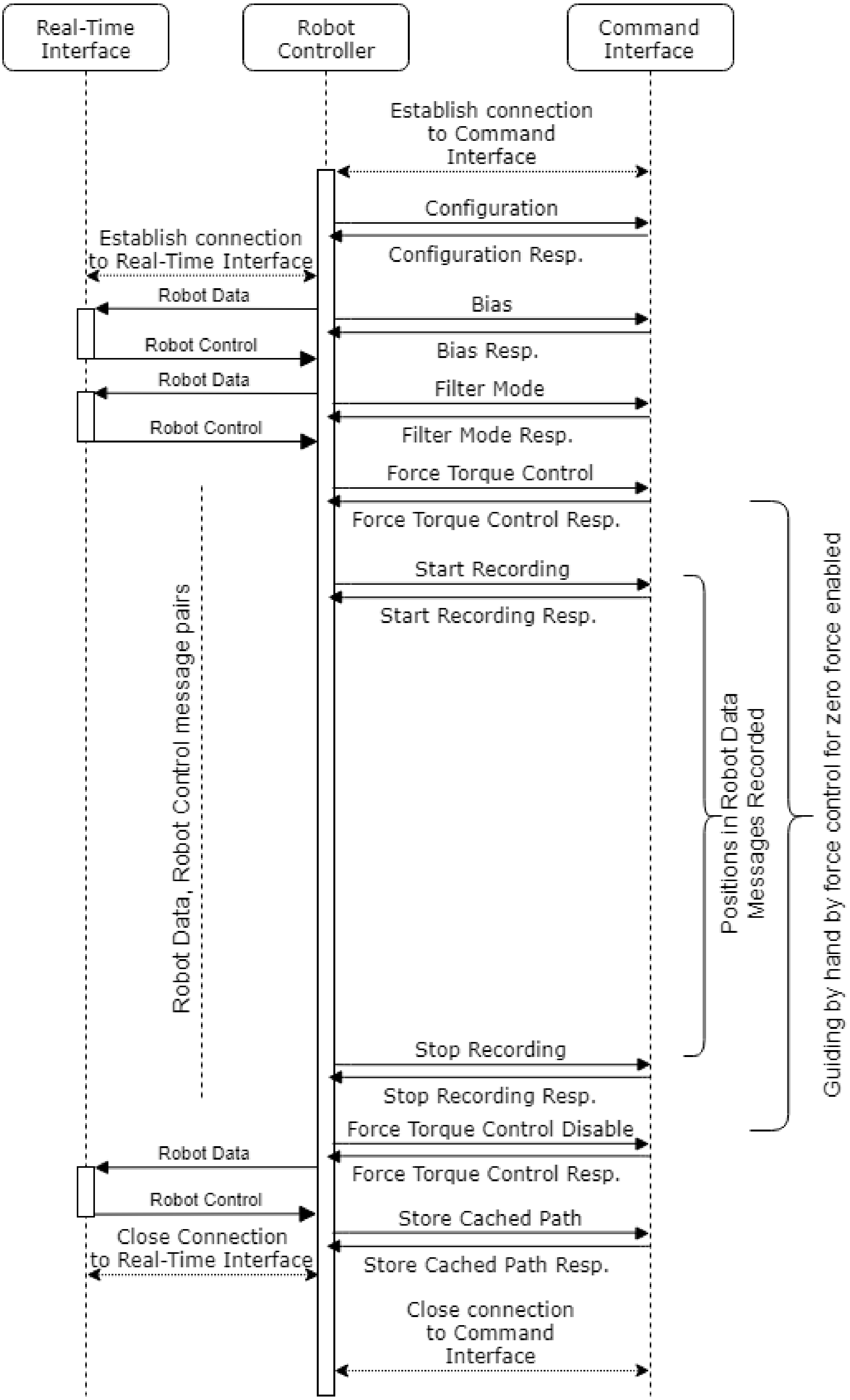
{"command":{"id":"force_torque_limit"}}

{"result":0,"error":"","response":{"id":“ force_torque_limit”}}

{"command":{"id":"force_torque_control"}} {"result":0,"error":"","response":{"id":“ force_torque_control”}}

5.4.2 Path Recording and Saving

The following image shows a message flow example, that implements a Path Recording example.


Real-Time Interface
Robot Controller
Command Interface
Establish connection to Command Interface
Configuration
Configuration Resp.
Establish connection to Real-Time Interface
Robot Data
Bias
Robot Control
Bias Resp.
Robot Data
Filter Mode
Robot Control
Filter Mode Resp.
Force Torque Control
Force Torque Control Resp.
Start Recording
Start Recording Resp.
Robot Data, Robot Control message pairs
Positions in Robot Data Messages Recorded
Guiding by hand by force control for zero force enabled
Stop Recording
Stop Recording Resp.
Force Torque Control Disable
Robot Data
Force Torque Control Resp.
Robot Control
Store Cached Path
Close Connection to Real-Time Interface
Store Cached Path Resp.
Close connection to Command Interface

This example includes the following flow:

Connection establishment for both parts of the CBRCI and robot configuration information transfer.

Configuring the sensor behavior.

Commanding the robot to be guided by hand by Controlling for 0 force.

Recording and storing a Path.

Closing connection for both parts of the CBRCI.

Below are examples for messages within the flow described above. Connection establishment, Connection Closing, Robot Data and Robot Control messages are omitted.

```
{"command":{"id":"configuration","robot_cycle":8,"sensor_cycle":4,"max_translational_speed":1500,"max_rotational_speed":1.75,"max_translation_alacceleration":3000,"max_rotational_acceleration":1.2}}

{"result":0,"error":"","response":{"id":“configuration”}}

{"command":{"id":"bias”}}

{"result":0,"error":"","response":{"id":“bias”}}

{"command":{"id":"filter_mode”,"filter_mode":0}}

{"result":0,"error":"","response":{"id":“filter_mode”}}

{"command":{"id":"force_torque_control","ft":[0,0,0,0,0,0],"frame_type":1,"proportional_gain":[1.2,1.2,1.2,1.5,1.5,1.5],"integrative_gain":[0.2,0.2,0.2,0.3,0.3,0.3],"enabled_axes":[true,true,true,true,true,true],"force_square_threshold":0,"torque_square_threshold":0,"max_translational_speed_by_control":1000,"max_rotational_speed_by_control":1}}

{"result":0,"error":"","response":{"id":“force_torque_control”}}

{"command":{"id":"start_recording”}}

{"result":0,"error":"","response":{"id":“ start_recording”}}

{"command":{"id":"stop_recording”}}

{"result":0,"error":"","response":{"id":“ stop_recording”}}

{"command":{"id":"force_torque_control"}}

{"result":0,"error":"","response":{"id":“ force_torque_control”}}

{"command":{"id":"store_cached_path”}}

{"result":0,"error":"","response":{"id":"store_cached_path","path_id":1357}}
```

6 Glossary of Terms

Term	Description
Compute Box	A unit provided by OnRobot along with the sensor. It performs the calculations needed to use the commands and applications implemented by OnRobot. It needs to be connected to the sensor and the robot controller.
OnRobot Data Visualization	Data visualization software created by OnRobot, to visualize the data provided by the sensor. Can be installed on Windows operating system.

7 List of Acronyms

Acronym	Expansion
CBRCI	Compute Box Robot Controller Interface
DHCP	Dynamic Host Configuration Protocol
DIP	dual in-line package
F/T	Force/Torque
ID	Identifier
IP	Internet Protocol
IT	Information technology
MAC	media access control
PC	Personal Computer
RPY	Roll-Pitch-Yaw
SW	software
TCP	Transmission Control Protocol
USB	Universal Serial Bus
UTP	unshielded twisted pair

8 Appendix

8.1 Editions

Edition	Comment
Edition 1	This is the first edition of this document.
Edition 2	Editorial error correction.
Edition 3	Asynchronous Interface removed, all commands use the Command Interface. Robot Data message format updated (rotation changed from quaternion to rotational vector) All command syntaxes updated. Responses added for all commands. Method of disabling Force Torque limit and Force Torque Control changed. Stop message added. Path handling messages added. Section Compute Box Discovery added.
Edition 4	Editorial changes.
Edition 5	Error correction: move response result value 0 removed.
Edition 6	New look & feel. Editorial changes.