Position Interface

V1 – May 28th, 2024
CPRog Version: V902-14-003
RobotControl Version: V980-14-003

Change Log

- V1 / May 28th, 2024: Initial documentation

1. Summary

The Position Interface allows fast position streaming from and to the robot using a human readable message format. Once connected the robot sends the current position once each main loop cycle. An application can send target positions at a user-defined interval. If enabled the robot will follow this position stream.

To use the position interface two TCP/IP connections are needed. One connection is to a control interface (currently only CRI is supported) to enable the position interface, to observe the robot's error state and for less timing critical commands like enabling and disabling the motors. The second connection is to the position interface.

2. Example and Test Software

An example client software is available via Github:

https://github.com/CommonplaceRobotics/SW_PositionInterfaceClient

3. Interface States

The Position Interface can be in one of the following states:

- Disabled – The interface is not running, no connection possible
- Enabled – Connection possible, the current position can be received but target positions are ignored
- In Use – The robot follows the target position. A client must be connected, otherwise the state will change back to enabled.

The state can be changed via CRI as explained below.

4. Connecting

Connecting and enabling the position interface generally works as follows:

1. Connect to the CRI
2. Query the position interface state and configuration using "*CMD GetPositionInterface*". This message contains the port number and whether the interface is enabled and in use for positioning.

3. If not enabled yet enable it by sending “*CONFIG SetPositionInterface true*”. This starts the interface; you can now connect to the port and receive the current position stream.
4. Once you want to send target positions make sure your position client is connected, no program is running and that any error is reset. Send “*CMD UsePositionInterface true*”. The robot will now follow the target position.

Important: Send your positions as steadily as possible. The position interface calculates an average interval and uses this for interpolation.

Warning! The robot will try to move to the given position within the calculated timing interval. If the interval is short and the target far this may lead to a very fast motion or position lag / encoder error. The robot checks the joint velocity and stops with a kinematic error if exceeded (observe the CRI STATUS message).

5. Multi-Client support

Unlike the CRI the Position Interface only supports one client. Further incoming connection requests are closed.

6. Definition of the CRI control commands

The CRI provides the following commands to enable the Position Interface. Please refer to the CRI documentation for further information or changes.

`CRISTART 1234 CMD GetPositionInterface CRIEND`
`Gets the state of the position interface. The following response is sent: CRISTART 1234 CMD PositionInterface port running inUse CRIEND`
`The first parameter is the port number used by the position interface. <running> is a Boolean indicating whether the server is running and can be connected to. <inUse> is a Boolean indicating whether the position interface is currently being used as position source.`

`CRISTART 1234 CMD UsePositionInterface true CRIEND`
`Selects or deselects the position interface as position source. The interface must be enabled via CONFIG SetPositionInterface first and a client must be connected. When the client loses connection the robot stops using the interface.`

`CRISTART 1234 CONFIG SetPositionInterface true CRIEND`
`Sets the position interface configuration. The first Boolean defines whether the interface should be running (this value is saved, the interface will start up again after a restart of the software). Use CMD GetPositionInterface to request the state and configuration. Its response is also sent as response to CONFIG SetPositionInterface.`

7. Definition of the Position Interface

7.1 Description

- The robot control sets up a TCP server. The port generally is the CRI port + 20 (3940-3950) but it should be queried via CRI (GetPositionInterface) to be sure.

- There is no keepalive message. The server closes the connection if it fails to send to the client. You may send QUIT to tell the server to close the connection for fast reconnects. This may change in future.
- Messages to and from the server follow the scheme:
  “MSGSTART message MSGEND”
- All values are sent in US style (with points as delimitator). Position and angle units are mm and degrees.

7.2 Error Responses

If you send an incorrect message one of the following error messages are sent:

```
MSGSTART ERROR UNKNOWN MSGEND
Invalid message type.

MSGSTART ERROR FORMAT MSGEND
Valid message type but incorrect format.
```

7.3 Current Position Message

The current position is sent whenever it is updated (i.e. at the robot’s kinematic main loop cycle time, usually 10-20ms). It is sent via the following message:

MSGSTART Pos J *a1 a2 a3 a4 a5 a6* **E** *e1 e2 e3* **C** *X Y Z A B C* **MSGEND**
Note that J is followed by 6 robot axes, E by 3 external axes and C by cartesian X, Y, Z and A, B, C.

7.4 Sending Target Positions

Send the following message to send a target position.

```
MSGSTART Pos J a1 a2 a3 a4 a5 a6 E e1 e2 e3 C X Y Z A B MSGEND

Selecting the motion mode:
Each category (J, E, C) including its entries may be missing – in this case the
last position is used. To do a joint motion do not send C. To do a cartesian
motion do not send J.

Omitting axes:
Within each category entries may be omitted; in this case their value will be
0. E.g. you may only send 3 axes in case of a gantry or if the other axes
should move to 0. In general, it is okay to always send all entries.
```

7.5 Disconnecting

It is fine to disconnect the client by simply closing the TCP socket. The server detects the closed connection when sending a message fails.

```
MSGSTART QUIT MSGEND
Requests to immediately close the connection and get ready for a new
```

```
connection.
```