TUM Institute for Cognitive Systems (ICS)

OpenWalker

Module Description: Real Robot (RRM)

Florian Bergner, Emmanuel Dean, Rogelio Guadarrama-Olvera,
Simon Armleder, and Gordon Cheng

February 14, 2020

1 Module Description

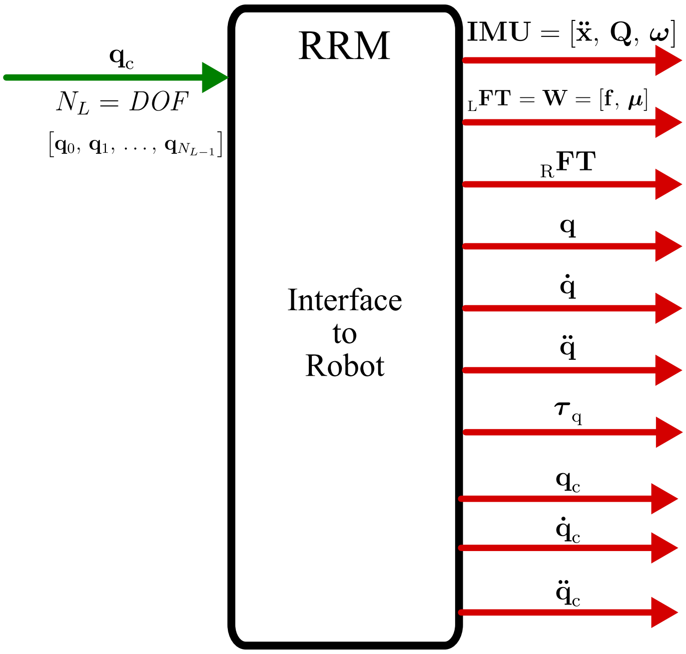

Figure 1.1: Real Robot module: This module implements the interface to the real robot.

The *Real Robot* module (RRM) implements the interface to the real robot. This module acquires all the available information provided by the sensors of the real robot and sends commands to the actuators of the robot. The RRM abstracts robot specific procedures for reading its

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732287.

sensors and write commands to its actuators. Commands and sensor information are provided in the data types of the OpenWalker project and interfaces structure and control the access to the information. The module manages the access to the robot's sensors and actuators and homogenizes the exchanged information. As a result, all modules of the OpenWalker project can access different robots in the same way using the same data types. The common access eases development and maintenance.

2 Module Connections

2.1 Inputs

Symbol	Name	Type	Description
$\mathbf{q}_c \in \mathbb{R}^{DOF}$	Commanded Joint Position	`JointPosition`	This vector contains the next commanded joint positions for all the joints of the robot. The OpenWalker framework uses this module input to send position commands to the position controlled real robot.

2.2 Outputs

Symbol	Name	Type	Description
$\ddot{\mathbf{x}}_{imu} \in \mathbb{R}^3$	IMU Linear Acceleration	`LinearAcceleration`	This vector contains the linear acceleration measured by the Inertial Measurement Unit (IMU) sensor of the robot. The OpenWalker framework can use this module output for model based state estimations.
$\mathbf{Q}_{imu} \in \mathbb{R}^4$	IMU Angular Position	`AngularPosition`	This vector contains the angular position in quaternion measured by the Inertial Measurement Unit (IMU) sensor of the robot. The OpenWalker framework can use this module output for model based state estimations.
$\boldsymbol{\omega}_{imu} \in \mathbb{R}^3$	IMU Angular Velocity	`AngularVelocity`	This vector contains the angular velocity in quaternion measured by the Inertial Measurement Unit (IMU) sensor of the robot. The OpenWalker framework can use this module output for model based state estimations.
${}_L\mathbf{FT} = {}_L\mathbf{W} \in \mathbb{R}^6$	FT Left Foot Wrench	`ForceTorqueSensor`	This vector contains the wrench measured by the force torque sensor in the left foot of the robot. The OpenWalker framework can use this module output for the Zero-Moment-Point estimation.
${}_R\mathbf{FT} = {}_R\mathbf{W} \in \mathbb{R}^6$	FT Right Foot Wrench	`ForceTorqueSensor`	This vector contains the wrench measured by the force torque sensor in the right foot of the robot. The OpenWalker framework can use this module output for the Zero-Moment-Point estimation.
$\mathbf{q} \in \mathbb{R}^{DOF}$	Real Robot Joint Positions	`JointPosition`	This vector contains the real joint positions of the robot. The OpenWalker framework uses this module output to compute the forward kinematics of the real robot.
$\dot{\mathbf{q}} \in \mathbb{R}^{DOF}$	Real Robot Joint Velocities	`JointVelocity`	This vector contains the real joint velocities of the robot. The OpenWalker framework uses this module output to compute the forward kinematics of the real robot.
$\ddot{\mathbf{q}} \in \mathbb{R}^{DOF}$	Real Robot Joint Accelerations	`JointAcceleration`	This vector contains the real joint accelerations of the robot. The OpenWalker framework uses this module output to compute the forward kinematics of the real robot.
$\boldsymbol{\tau}_q \in \mathbb{R}^{DOF}$	Real Robot Joint Torques	`JointEffort`	This vector contains the real joint torques of the robot. The OpenWalker framework can use this module output to monitor the joint torques of the real robot.
$\mathbf{q}_c \in \mathbb{R}^{DOF}$	Commanded Robot Joint Positions	`JointPosition`	This vector contains the currently commanded joint positions of the robot. The OpenWalker framework uses this module output to compute the forward kinematics of the commanded robot.
$\dot{\mathbf{q}}_c \in \mathbb{R}^{DOF}$	Commanded Robot Joint Velocities	`JointVelocity`	This vector contains the currently commanded joint velocities of the robot. The OpenWalker framework uses this module output to compute the forward kinematics of the commanded robot.
$\ddot{\mathbf{q}}_c \in \mathbb{R}^{DOF}$	Commanded Robot Joint Accelerations	`JointAcceleration`	This vector contains the currently commanded joint accelerations of the robot. These are numerically computed by deriving $\mathbf{q}_c$. The OpenWalker framework uses this module output to compute the forward kinematics of the commanded robot.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732287.

2.3 Inter-Connections

The RRM is connected to the two forward kinematics modules (FKMs) and provides the real and commanded joint positions, velocities, and acceleration to these modules. The real and commanded FKM need this information to compute/update the forward kinematics. The RRM also provides the IMU measurements to the Center-of-Mass module (CoMM), the Zero-Moment-Point module (ZMPM), and the Balancer module (BM) to provide additional information for state estimators. Additionally the RRM provides the FT sensor measurements to the ZMPM which filters the measurements. These filtered FT sensor measurements are then used by the ZMPM itself and provided to other modules.

2.4 Common Methods

This module is a pure interface module and thus does not need any mathematical, physical, or robotical models to compute its outputs. The main tasks of the RRM module are access management, type conversions, and structuring of information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732287.