Wheeled Humanoid Hubo ROS API

Moonyoung Lee, Yujin Heo, Saihim Cho
1. Motivation

**Strengths** of DARPA Robotics Challenge Winning Platform (Hardware)

- **Locomotion (Strong Actuators)**
- **Manipulation (Precise Control)**
1. Motivation

Strengths of DARPA Robotics Challenge Winning Platform (Software PODO)
1. Motivation

1. Strengths of DARPA Robotics Challenge Winning Platform

![Diagram of RTOS architecture]
1. **Motivation**

1. **Strengths** of DARPA Robotics Challenge Winning Platform
1. Motivation

**Strengths** of DARPA Robotics Challenge Winning Platform

Limitations?
1. Motivation

**Limitations** of DARPA Robotics Challenge Winning Platform

SW for tele-operation & precise control → limited framework for autonomy
1. Motivation

**Limitations** of DARPA Robotics Challenge Winning Platform

SW for tele-operation & precise control → **limited framework for autonomy**
1. Motivation

**Limitations** of DARPA Robotics Challenge Winning Platform

SW for tele-operation & precise control → **limited framework for autonomy**
2. Solution: API

**Interface** for Hubo platform in ROS

- **Vision**: RGB-D Camera, 3D Lidar & IMU
- **Arm**: Joint Space (Reference(angular), time), Task Space (Position(xy,z), orientation(quaternion), time), Joint trajectory (Reference array)
- **Gripper**: Open / Close (Side, command)
- **Base**: Move goal (Position(xy), orientation(yaw), Velocity(xy), orientation(yaw))
2. Solution: API

1 Software Architecture

- Task Planner
- Action Client
- Motion Planner (MoveIT!)
- Action Server
- Base
- Gripper
- Arm
- LAN COMMUNICATION
- Arms, Gripper, Base Control @ 200Hz
- Shared Memory
- Encoder
- Motor Controller @ 1 kHz

Goal → Feedback, Result

CMD, Action Goal, Trajectory

θ_{current}[n] → θ_{ref}[n]
2. Solution : API

1 Software Architecture

- **Task Planner**
  - Action Client
  - Motion Planner (MoveIT!)

- **Vision PC**
  - ROS
  - Action Server
    - Base
    - Gripper
    - Arm
  - Goal / Feedback, Result

- **Motion PC**
  - PODO
  - LAN COMMUNICATION
    - Arms, Gripper, Base Control @ 200Hz
    - CMD, Action Goal, Trajectory
  - Shared Memory
    - Encoder
    - Motor Controller @ 1 kHz
    - \( \theta_{\text{current}[n]} \)
    - \( \theta_{\text{ref}[n]} \)

- **LAN Communication**
  - MoveX = 1.0m
  - Action Goal
2. Solution: API

Software Architecture

- **Motion PC**: Encoder, Motor Controller @ 1 kHz
- **Vision PC**: Motion Planner (MoveIT!), Task Planner
- **LAN COMMUNICATION**: Arms, Gripper, Base Control @ 200 Hz
- **Action Server**: Action Client
- **Joint q[n]**, At 200 Hz?
- **θ_current[n]**, **θ_ref[n]**
- **Goal, Feedback, Result**: Motion Planner, Action Client

At 200 Hz?
3. Problem: Delay

1 Communication Delay between ROS and PODO

1. Total communication delay: 22 ms (on average)
2. Delay inconsistency from NRT & RT communication
3. Problem

**3 Transmit time** of the packet from ROS to PODO

![Diagram showing the communication between ROS and PODO](image)

- **Action Client**
- **Action Server**
- **Motion Planner** (MoveIT)
- **LAN COMMUNICATION**
- **Arms, Gripper, Base Control**
- **Shared Memory**

200Hz sendGoal
3. Problem

4 The method chosen to minimize the impact of delays

High-frequency-request

Single-request

Interpolation
4. Result

Desired joint state trajectory input using ROS MoveIt

Collision-free path is generated for the robot
4. Result

**Trajectory Planning:**
Whole Body Collision Checking

**Machine Learning:**
Learn from Human Imitation

**Task Manager:**
Service Robot Application
5. Future work

1. Robot independent API with autonomy
Acknowledgements

Robot built by KAIST Hubo Lab & Rainbow Robotics © Funded by Ministry of Trade, industry & Energy(MI, Korea)