Imperial College London

Human-centered manipulation and navigation with Robot DE NIRO

Fabian Falck [1], Sagar Doshi [1], Nico Smuts [1], John Lingi [1], Kim Rants [1], Petar Kormushev [2]

[1] Department of Computing, Robot Intelligence Lab, Imperial College London

[2] Dyson School of Design Engineering, Robot Intelligence Lab, Imperial College London

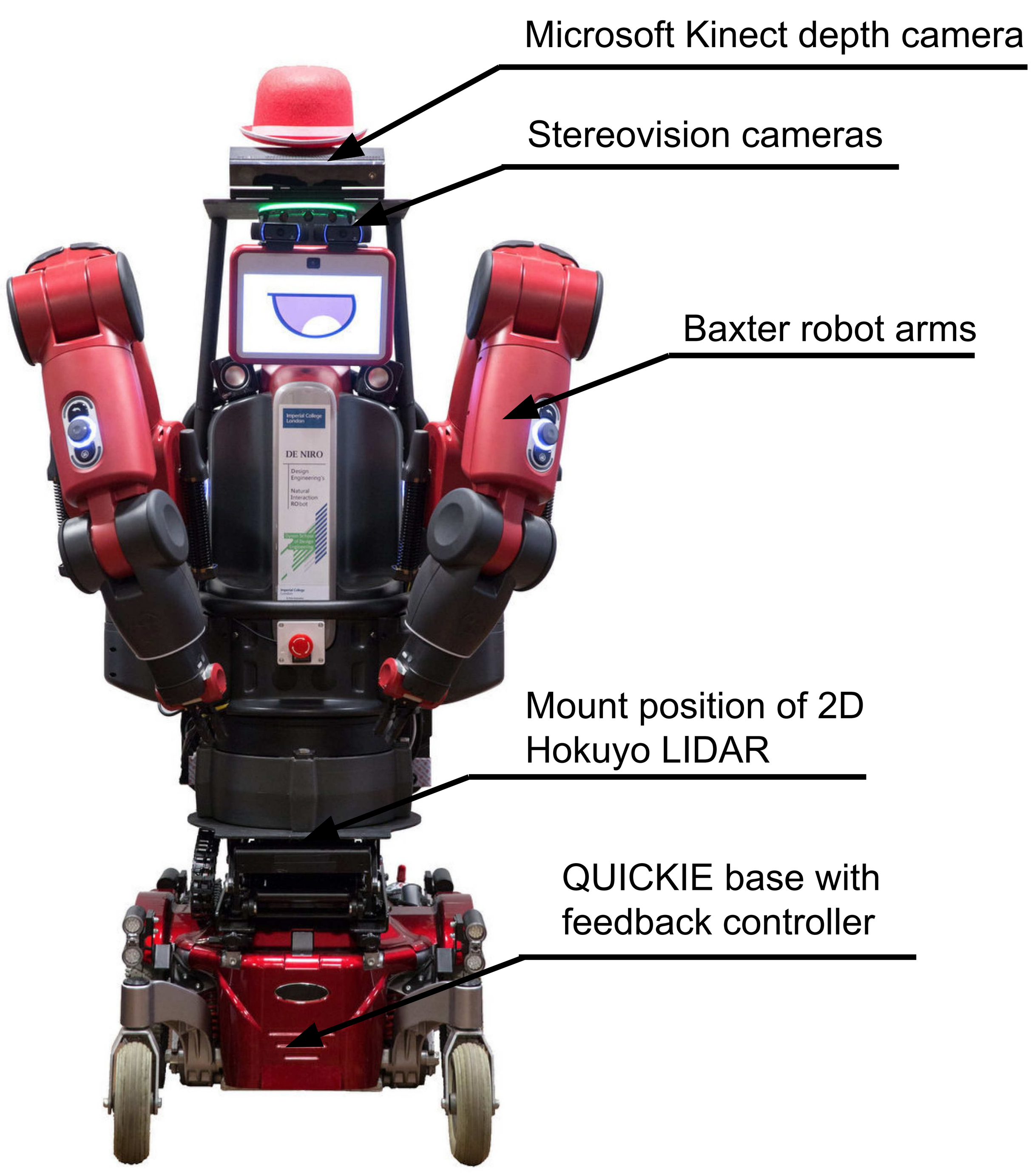
Summary

We present **Robot DE NIRO** (**D**esign **E**ngineering's **N**atural **I**nteraction **Ro**bot), a collaborative research platform for mobile manipulation. Given the macrosocial trends of aging and long-lived populations, robotics-based care research mainly focused on helping the elderly live independently [1] [2]. In contrast, DE NIRO aims to support the supporter (the caregiver) and also offers direct human-robot interaction for the care recipient.

Hardware Design

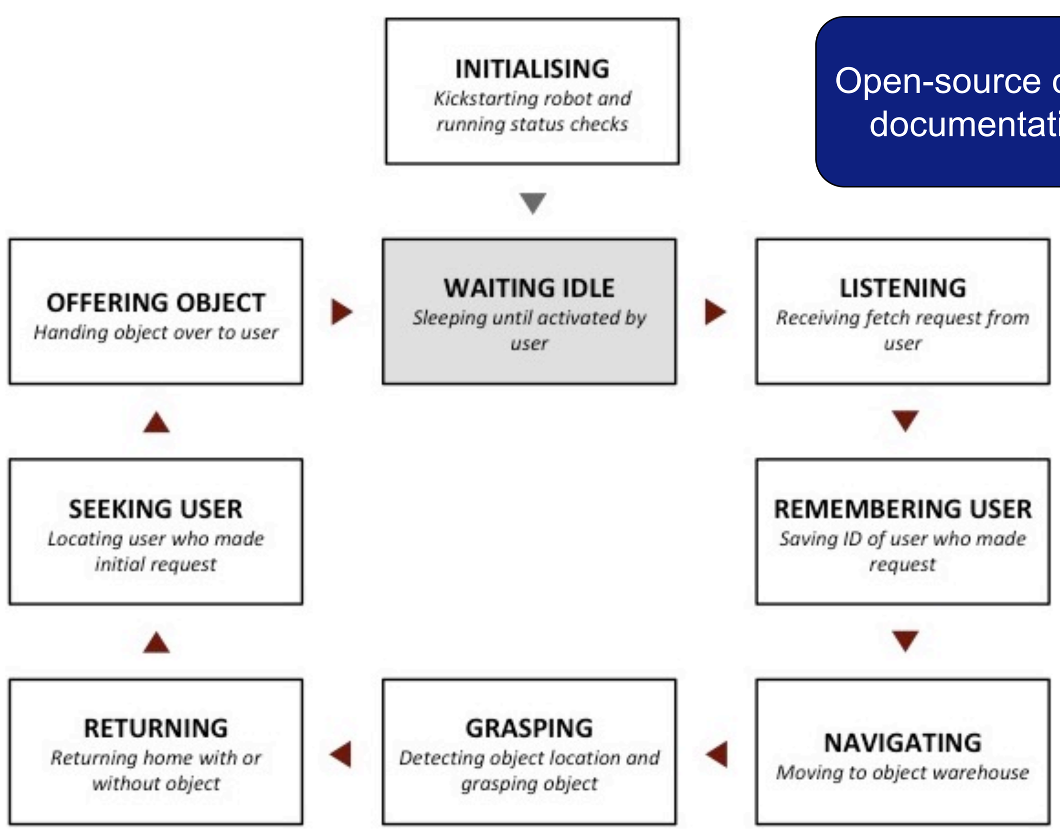
DE NIRO's core design idea is to combine the industrial Baxter dual robot arms with autonomous navigation into a mobile manipulation research platform. It is equipped with a variety of actuators and sensors depicted below, of which shall be highlighted:

- **Baxter dual robot arms**: Passive compliance through series elastic actuators are a particular safety feature, allowing the robot to interact with humans in close proximity, since in the case of a contact, most of the physical impact is absorbed.
- **QUICKIE base**: Differential drive operated with a custom PID angular position and velocity controller, allowing primitive motion commands for navigation [3].
- **Microsoft Kinect RGB-D camera**: High-resolution depth camera.
- **2D Hokuyo LIDAR**: Industrial standard laser scanner capable of mapping and localization.



Software Implementation

In designing DE NIRO, we have put an emphasis on natural and safe human-robot interaction procedures across multiple components, including speech and face recognition and collision avoidance. To handle concurrent execution and both synchronous and asynchronous communication between components, we use **Robot Operating System (ROS)** as a middleware. We define distinct functionalities of the robot with a **finite-state machine** depicted below, such as *listening* (for command input) or *grasping* (to physically pick up an object). The state machine handles the control flow among these states.



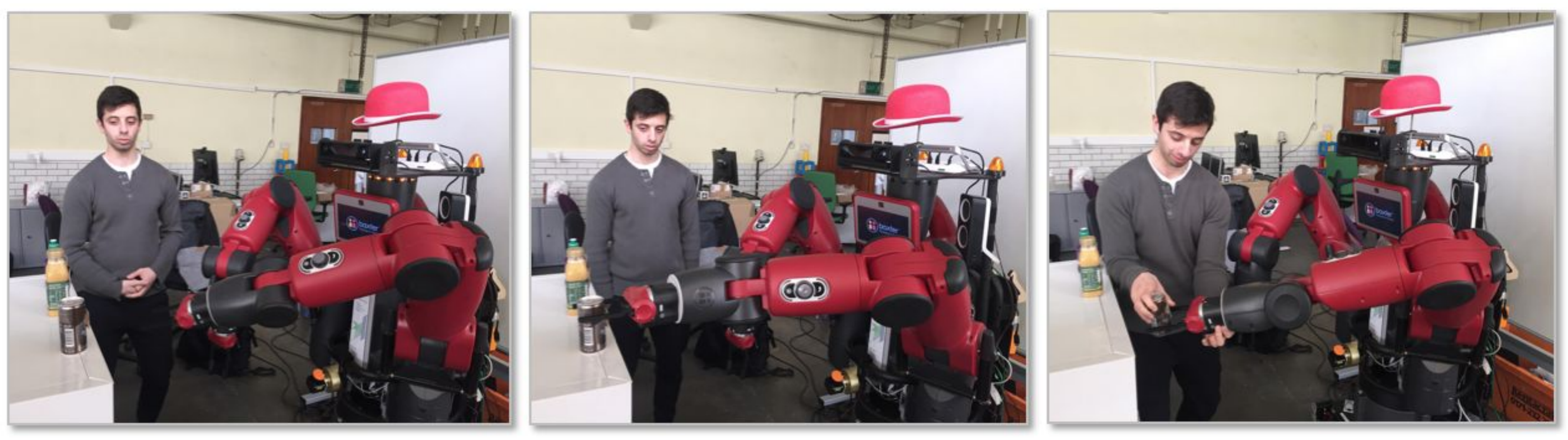
Open-source code, documentation

Scan me

Scan me

Object Recognition and Manipulation

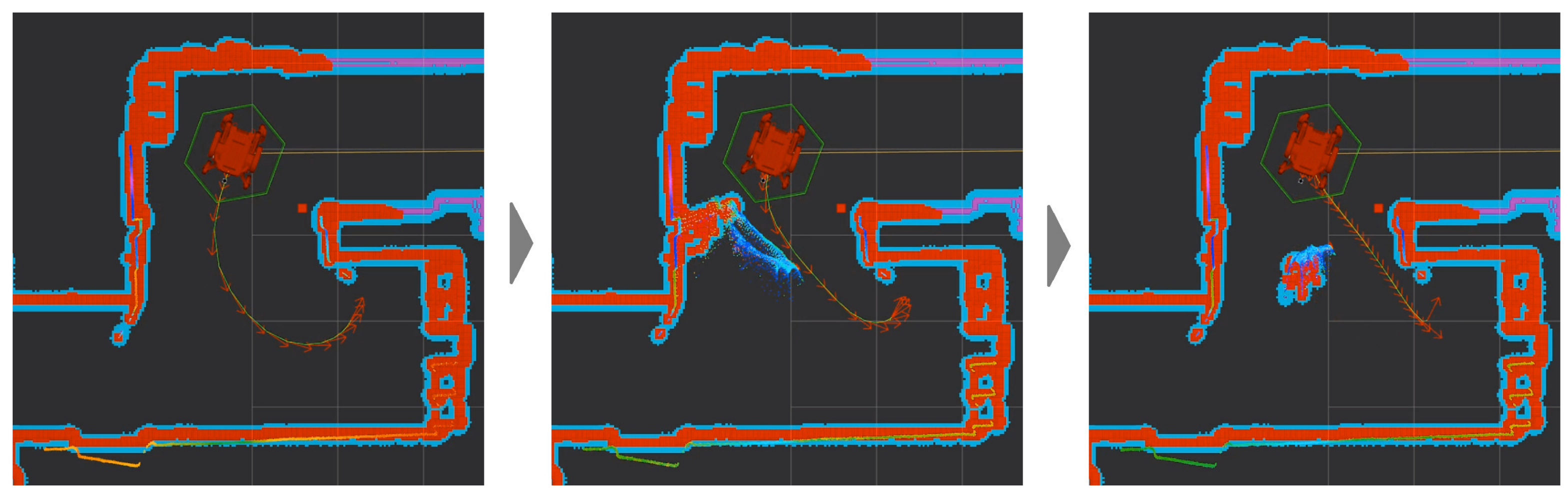
- Target objects are localized using **2D fiducial markers** [5].
- To control the Baxter arms, we employ an **inverse kinematics solver** to compute each of the seven joint angle trajectories needed to reach an object [6].
- A **dynamic awareness procedure** reacts to changes of the location of the target object during grasping and actively avoids collisions.



Navigation, Mapping and Planning

A navigation stack consisting of mapping, localization and trajectory planning was implemented (illustrated below).

- **Static mapping**: We apply a SLAM-based approach to the LIDAR sensor in order to detect any spatial boundaries and 2D artifacts in a predefined space [4].
- **Localization**: Overlaying a dynamic map onto a static map to detect obstacles for collision avoidance. A costmap is imposed as a virtual cushion for additional safety.
- **Trajectory planning**: "Timed elastic band" approach, conceiving of trajectory planning as a multi-objective optimization problem [7] [8].



Perception and user interaction

For a natural interaction with the user, we implemented face and speech recognition.

- **Face recognition**: A *ResNet* model pre-trained on faces was applied to video frames retrieved by the Kinect camera, reaching an accuracy of 99.38% on a standard benchmark [9]. The model compares the output vector encodings of known faces with those extracted from processed frames by computing a distance metric.
- **Speech recognition**: The offline library *CMU Sphinx* was implemented [10]. We defined a *Jspeech Grammar* to allow reliable voice commands in a specific format and automatically calibrate to background noise levels.
- **Speech output**: *eSpeak* [11] yielding a high reliability, rapid response time and an offline implementation.

Conclusion

- **Limitations**: nonholonomic design; maximum payload of 2.2 kg per arm; currently limited to forward motion only due to limited sensor capabilities
- **Future work**: increased awareness and safety through 360-degree camera rig; 3D LIDAR; more robust localization without predefined mapping; teleoperation through virtual reality headset and body tracking markers

REFERENCES

[1] B. Graf, C. Parlitz, and M. Hägele, "Robotic home assistant care-o-bot - product vision and innovation platform," in *International Conference on Human-Computer Interaction*. Springer, 2009, pp. 312–320.

[2] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, "Humanoid robot HRP-3," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 2471–2478.

[3] E. S. Aveiga, "State estimation and feedback controller design for autonomous navigation of a high-performance mobile robot," Imperial College London, 2017.

[4] S. Kohlbrecher, "Hector mapping ROS package," http://wiki.ros.org/ hector mapping, 2018.

[5] S. Lemaignan,"ROSMarkersChilitags,"https://github.com/chili-epfl/ros_markers.

[6] R. Robotics, "Inverse kinematics solver service," http://sdk.rethinkrobotics.com/wiki/IK_Service_-_Code_Walkthrough.

[7] C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," European Conference on Mobile Robots Barcelona pp 138-143, 2013.

[8] C. Rosmann, "Timed elastic band algorithm implementation," http://wiki.ros.org/teb local planner, 2018.

[9] "dlib face recognition documentation," https://web.archive.org/web/20170302162536/http://dlib.net/dnn face recognition ex.cpp.html, 2018.

[10] C. M. University, "CMU Sphinx documentation," https://cmusphinx.github.io/wiki/, 2018.

[11] eSpeak, "eSpeak text to speech," http://espeak.sourceforge.net/index.html, 2018.