Approaching Autonomous Open World Transportation

Brayan S. Zapata-Impata[1], Vikrant Shah[2], Hanumant Singh[2] and Robert Platt[3]

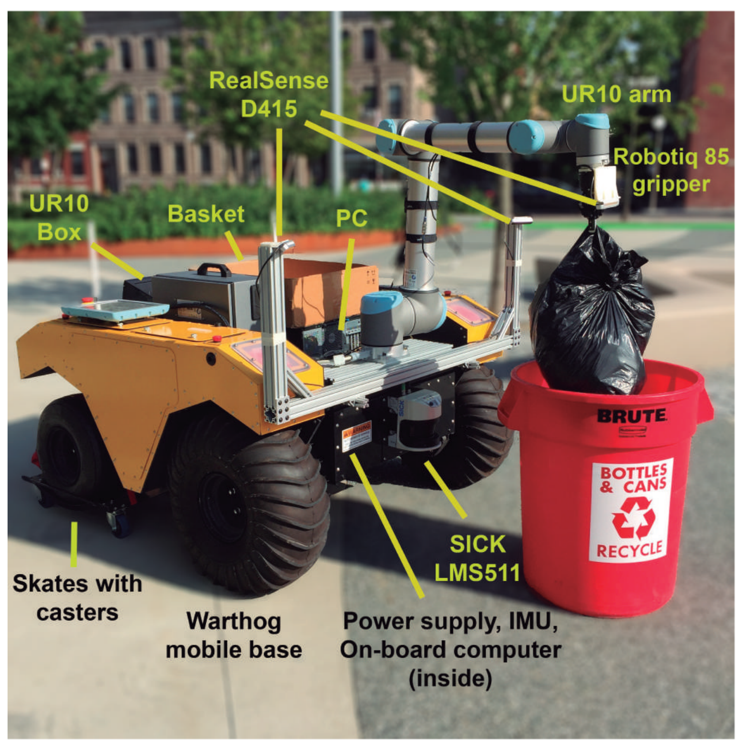
Universitat d'Alacant
Universidad de Alicante

[1] *Dept. of Physics, System Engineering and Signal Theory, University of Alicante, Spain*
[2] *Dept. of Electrical and Computer Engineering, Northeastern University, Boston, USA*
[3] *College of Computer and Information Science, Northeastern University, Boston, USA*

`brayan.impata@ua.es, shah.vi@husky.neu.edu, ha.singh@northeastern.edu, rplatt@ccs.neu.edu`

Northeastern University
College of Computer and Information Science

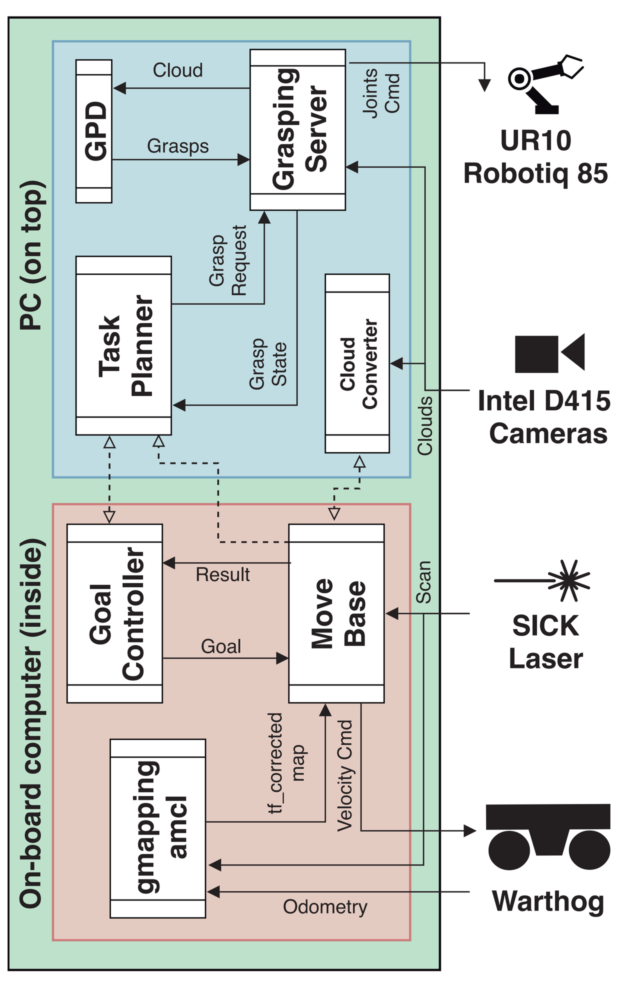
Problem Statement



We approach the task of **object transportation** in open world scenarios outdoors using a mobile manipulator. We call *open world* those **unstructured environments** that are not controlled nor adapted to ease the task for the robot. In addition, they have **dynamic elements** that can change from trial to trial (i.e. obstacles, uneven ground, light conditions).

Systems aiming to solve tasks in such scenarios have to be robust to work with little prior knowledge about the environment. In this work, a system is said to solve this problem if it can **transport all of the items from a given pick point to a drop point**, having only a map of the world for navigation. With this goal in mind, we use the **custom mobile manipulator** shown in the figure.

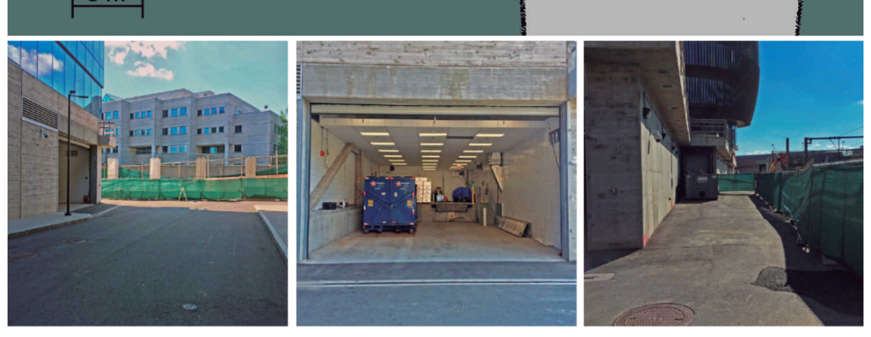
Approach



We propose an architecture composed of three components, developed completely in **ROS**:

- **Navigation Stack:** Delivers the robot base to a position such that the target objects are within the manipulator workspace. We use **GMapping SLAM** for creating maps, **AMCL** for localization and the *move_base* package for route planning and control of the robot.
- **Grasping Stack:** We calculate grasps using the **Grasp Pose Detection package (GPD)**, which samples grasps hypotheses over a 3D point cloud and ranks their potential success. Before selecting the best grasp, we prune kinematically infeasible grasps by checking inverse kinematics (IK) solutions against the environment constraints in **OpenRAVE**.
- **Task Planner:** It sends goals to the navigation and requests to the grasping stack to **provide the mobile manipulation functionality**. It requires **three inputs**: the type of task (*collect all* or *collect one by one*), the pick position and the drop position. Afterwards, the autonomous task can start: the robot **moves to the pick location**, **collects the items**, moves to the drop position in order to **deliver them** and repeats this logic in case of a *collect one by one* task.

Experimentation



Extense experiments were performed using four types of objects: **trash bags, general garbage, gardening tools and fruits**. These tests were carried out on **city streets** in the vecinity of a loading dock: on each trial the robot started from a random position, objects were in another random location and a bin was placed in a third random location.

Set - Trials	**P-Nav**	**P-Grasp**	**D-Nav**	**D-Grasp**	**Task**
Bags - 5	20/23 (87%)	15/19 (79%)	15/15 (100%)	15/15 (100%)	5/5 (100%)
Garbage - 6	6/6 (100%)	50/56 (89%)	6/8 (75%)	50/60 (83%)	5/6 (83%)
Tools - 5	5/5 (100%)	17/28 (61%)	5/5 (100%)	18/27 (67%)	3/5 (60%)
Fruits - 5	5/5 (100%)	30/33 (91%)	5/5 (100%)	30/39 (77%)	5/5 (100%)
Total - 21	96.57%	78.98%	93.06%	80.85%	**84.09%**

Sub-process	**Pick**	**Drop**
Register Point Cloud	3.78s ± 0.21s	33.20s ± 9.00s
Calculate Grasp	10.14s ± 5.41s	17.72s ± 8.05s
Execute Grasp	48.12s ± 7.09s	50.86s ± 13.72s
Navigate to Point	98.18s ± 25.73s	137.86s ± 39.49s

Challenges:

- **Object Size:** Small objects like general garbage can be **difficult to distinguish from the ground** in the 3D point clouds, making it harder to calculate precise grasps. Some tools were left behind undetected in the ground.
- **Ground:** The 2D laser scanner was not enough for localizing the robot in **uneven terrain**. A 3D laser scanner would be a more suitable solution.
- **Other Factors:** The **wind** can move the objects before the robot grasps them, resulting in grasp failures. The sun extremely affects the quality of the cloud, being midday **sunlight** the worst since it creates artifacts and noise.
- **Future Work:** Our main next goal is to **reduce the time gap** from registering a cloud to actually perform a grasp. This will **speed up** the system throughput and reduce failures caused by external factors (i.e.wind).

Watch video

Work supported in part by the National Science Foundation through IIS-1427081, IIS-1724191, and IIS-1724257, NASA through NNX16AC48A and NNX13AQ85G, ONR through N000141410047, Amazon through an ARA to Platt, and Google through a FRA to Platt, and partially funded by the Spanish Government through BES-2016-078290 to Zapata-Impata.