EduMIP SLAM and UR5 Transporting

March 31, 2017

1. Introduction

EduMIP is a self-balancing robot built around the BeagleBone Blue embeded microprocessor board developed by James Starwson and Professor Thomas Bewley of the Coordinated Robotics Laboratory at UCSD and their collaborators [1].

UR5 is a industrial robot arm from Universal Robts, which is ideal to optimize low-weight collaborative processes, such as picking, placing, and testing [2].

The goal of this project consists of two parts:

The first is, to use a RPLidar A1M8 Laser Scanner which is fixed to an EduMIP robot for implementing Simultaneous Localization and Mapping (SLAM) and use the generated map to move EduMIP autonomously to a set goal location, which is within the reachable range of a fixed UR5 robot arm.

The second is, to use the UR5 robot arm to pick up the moving EduMIP and transport it to a different location, drop it, then let it run autonomously to the destination. Instead of pre-storing fixed transporting locations, one more natural workaround is to attach an AR tag to the EduMIP, then use a camera to catch its locations and positions, UR5 will be capable of autonomously transporting.

2. Hardware and infrastructure

Existing available hardware:

- UR5: a 6 degree-of-freedom industrial robot arm capable of picking and placing
- EduMIP: a low-cost self-balancing two wheels robot

Additional hardware:

- Logitech C920 webcam: a high-resolution USB webcam [3]
- RPLidar A1M8 Laser Scanner: a 360° omnidirectional laser scanner with 5-10 Hz adaptive scan frequency, 0.2 ~ 6 meters range, 2000 times of ranging per second [4]

Custom Hardware:

- Sensor Bracket: use it to keep sensors horizontal when EduMIP is in upright position.

3. Software

Existing packages:

- Gmapping: This package contains a ROS wrapper for OpenSlam's Gmapping.
  http://wiki.ros.org/gmapping
- Navigation: Navigation Stack is a 2D navigation stack that takes in information from odometry, sensor streams, and a goal pose and outputs safe velocity commands that are sent to a mobile base.
  http://wiki.ros.org/navigation
- Serial: Serial is a cross-platform, simple to use library for using serial ports on computers.
  http://wiki.ros.org/serial
- Usb_cam: Usb_cam is a ROS driver for V4L USB cameras. It publishes images as sensor_msgs/Image.
  http://wiki.ros.org/usb_cam
- Ar_track_alvar: This package is a ROS wrapper for Alvar, an open source AR tag tracking library. It provides a transform from the camera frame to AR tag frame.
  http://wiki.ros.org/ar_track_alvar
- Ur_driver: This package is the driver for the UR5/10 arm based on the polyscope control scheme.
  http://wiki.ros.org/ur_driver

Modified packages:

- Edumip_balance_ros & edumip msgs: Edumip_balance ros is the main ROS node for balancing and driving the EduMIP based upon James Strawson's "rc_balance.c" reference. Edumip_msgs is associated message package.
  https://git.lcsr.jhu.edu/lwhitco1/edumip_balance_ros

Custom packages:

- Edumip_SLAM: This package does the SLAM. It subscribes state information from EduMIP and publish tf trees. It reads the message from the Laser Scanner and publishes the transformations to where the UR5 should move.
- Joy_twist: We use this package to manually control EduMIP.
- UR5_transport: This package moves the UR5 to the configurations determined by Edumip_state node. It also publishes a message gripper/cmd to control the gripper.
- Gripper_control: This node subscribes to the message gripper/cmd and actuates the gripper by setting the appropriate digital output.

4. Safety Plan

Number	Step	Possible Hazard(s)	Control(s)
1	EduMIP SLAM	Breaking barrier	Test SLAM algorithm on the ground
2	UR5 picking up EduMIP	Breaking gripper	Test UR5 catching motion with slow velocity
3	UR5 transporting	UR5 losing control	One person must keep holding the emergency stop button of UR5 during the test
4	UR5 transporting	UR5 dropping EduMIP at a certain height that breaks EduMIP	Pave drop-proofing material to reduce the possible damage, and do more tests to reduce the risk

5. References

[1] https://dscl.lcsr.jhu.edu/ME530707_2017_EduMIP_ROS/

[2] https://www.universal-robots.com/products/ur5-robot/

[3] http://support.logitech.com/en_us/product/hd-pro-webcam-c920/specs

[4] https://www.robotshop.com/en/rplidar-a1m8-360-degree-laser-scanner-development-kit.html

6. Project Timeline

Week1: use a UR5 robot arm and an EduMIP robot to implement UR5 transporting task. If time permits, try using AR tags for autonomously transporting.

Week2: design a 3D-printed sensor bracket for holding the RPLidar A1M8 Laser Scanner, start EduMIP SLAM.

Week3: finish EduMIP SLAM, build a simple environment with obstacles for exploring, start testing EduMIP SLAM.

Week4: finish testing EduMIP SLAM, continue using AR tags for autonomously transporting.

Week5: combine EduMIP SLAM and UR5 transporting together, more testing and debugging, finish the project, make a poster, and hand in the final report.