A Designed Multi-sensor System for SLAM on Multiple Robots

Xiaodong Wu, Dr. Miguel Abrahantes

Hope College, Department of Engineering

Abstract

A research group at Hope College is designing a multi-ultrasonic-sensor system for the open source robot Kobuki Turtlebot. The operating system used in this research project, ROS (Robot Operating System), is also an open source project initially released in 2007. This system has higher accuracy, reduced response time and lower cost compared to traditional laser sensors. It consists of 8 HY-SRF05 rangers that can work sequentially on an embedded system such as Raspberry Pi. The communication between the embedded system and the robot is achieved via the designed Python package through local TCP/IP (Transmission Control Protocol/Internet Protocol).

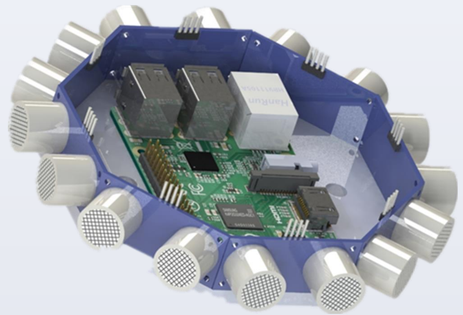
A multi-robot system with one master and two slaves.

Objectives

- Develop a reliable multi-sensor system for SLAM
- Implement a multi-agent robot system on ROS
- Perform echolocation and SLAM experiments with these two built systems

Multi-sensor System

We designed a multi-sensor system based on eight HY-SRF05 transducers and an embedded system (Raspberry Pi) which is capable of signal processing, data storing and transmitting, and wireless communications on local network. Eight sensors can scan the plane and return the localization information in milliseconds.

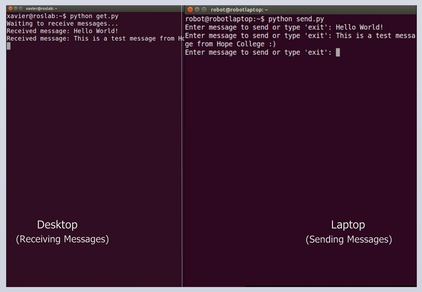


A challenge of this multi-sensor structure is that those sensors may not work simultaneously due to acoustic interference and lag in the embedded system or in the signal transmission process. By making them work sequentially, the response time needs to be reduced as much as possible to minimize the errors in the localization process. This structure can be also applied to our multi-agent system, composed of 3 robots of the same model. By implementing this improved design on a multi-agent system, the SLAM (Simultaneous Localization and Mapping) process can be conducted more efficiently and accurately.

python™

```
import os from socket import *
host = ""
port = 13000
buf = 1024
addr = (host, port)
UDPSock = socket(AF_INET, SOCK_DGRAM)
UDPSock.bind(addr)
print "Waiting to receive messages..."
while True:
    (data, addr) = UDPSock.recvfrom(buf)
    print "Received message: " + data if data == "exit":
    break UDPSock.close()
os._exit(0)
```

Python is essential to this project. It is a simple but powerful programming language that we used in ROS, our developed software packages, and APIs. Left is a sample code for receiving messages from local TCP/IP communications.



A demonstration for local communication over TCP/IP, a essential communication language/protocol of the Internet.

SLAM & Multi-agent Robot System

The development of the multi-agent robot system is based on ROS and Ubuntu. ROS (Robot Operating System) is a collection of software frameworks for robot software development. It provides operating-system-like functionality on a heterogeneous computer cluster.

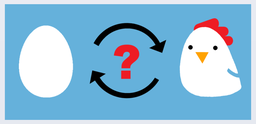
Services Provided by ROS:

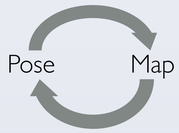
- Hardware abstraction
- Low-level device control
- Message passing service
- Name and Parameter service
- Package management

ROS

ubuntu

Why is SLAM a Hard Problem?





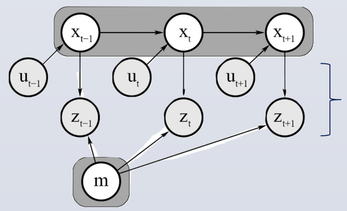
- A map is needed for localization
- A pose estimate is needed for mapping

The robot's path and the map can be estimated by:

Given

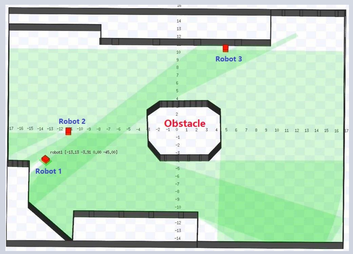
$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

distribution path map observations controls



$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

SLAM can be usually processed through approximation. There are many different algorithms developed for SLAM. ROS also comes with several packages for localization and automatic path optimization.

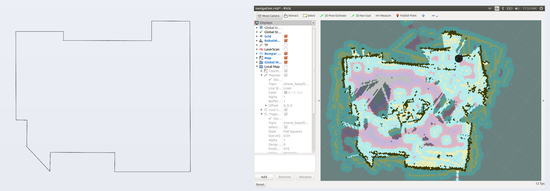


Simulation of Three Robots in the Robotics Lab at Hope College

Future Work

The development of ROS for multi-agent systems is still in a very early stage, which is another challenge for our project.

We also need to keep testing and revising the design of our multi-sensor system to make it work seamlessly with ROS. Currently we managed to scan the environment using a Kinect sensor, and we will modify the software package to replace the Kinect sensor with our ultrasonic sensor system.



Comparison of the scaled lab environment and the SLAM result using Kinect sensor.

Robotics and System Control Lab at Hope College

References

1. https://en.wikipedia.org/wiki/Robot_Operating_System
2. Stachniss, http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/
3. ROS by Example (Indigo), R. Patrick Goebel, 2015
4. http://www.causality.inf.ethz.ch/cause-effect.php

Acknowledgement

Hope COLLEGE
Department of Engineering

Dr. Miguel Abrahantes
Prof. Mark Edgington
Mr. Andrew Bunnell
Mr. Dave Daugherty