

A project report on

Indoor Navigation and Localization Mobile Robot

(Final year project)

Submitted in partial fulfillment of the requirement for the award of the
Degree of Bachelor in
COMPUTER AND SYSTEMS ENGINEERING



Submitted by

Ahmed Abdelbadee Elsayed
Ahmed Abdelbasit Mohamed
Aya Ibrahim Elsayed
Nourhan Mansour Mohamed
Omar Raafat Abdullatif
Yasmin Ahmed Abdelbasit

Supervised by:

Dr.Ahmed Mohamed Helmi

July 14, 2017
**Faculty of Engineering
Zagazig University**

Abstract

This project introduces a small-size mobile robot to be used for indoor navigation. It can be operated either autonomously or controlled by man remotely over the network. Its size make it able to navigate in small places and narrow paths man can not go through. Man can discover the environment around it via a video stream transmitted by a camera free to rotate right, left, up and down. A user-friendly controller box is provided for manned control to guide robot's motion, set camera orientation and switch on/off flash light.

The autonomous navigation is based on graph theory and artificial intelligence search algorithms in a predefined map for the environment around robot. With the help of Landmarks detection by computer vision, the robot can identify its location periodically on his path from source to destination. This provides a great help to avoid accumulated errors caused by hardware or sensors in accuracy.

A computer graphical user interface (GUI) application is developed to easily reach the functionality of robot. By this application one can select either autonomous or manual mode and deal with each mode utilities.

The project is based on Robots Operating System (ROS) which makes its functionality reusable in other projects. Modularity and readable codes are considered in the design and implementation of software nodes. Also an optimized communication protocol is developed among project's parts.

About future work there is a wide field of updates like object detection, On-line Mapping of new Environments and installation of manipulator (i.e. robot arm) for a variety of tasks.

Key abbreviations

ROS	Robots Operating System
GUI	Graphical User Interface
RPi	Raspberry Pi
PID	Proportion Integral Derivative
PWM	Pulse Width Modulation
I^2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LiPo	Lithium Polymer
RPM	Revolution Per Minute
ADC	Analog-Digital Converter
LPF	Low Pass Filter
FPS	Frame Per Seconds
GND	Ground

Contents

Abstract	i
Key abbreviations	ii
1 Introduction	1
1.1 History of Mobile Robots	1
1.2 Project Objectives	2
1.3 Limitations	3
1.4 Overview on the project parts	3
2 User Manual and GUI	4
3 System Structure and Proposed Algorithm	4
3.1 Robots Operating System (ROS)	4
3.2 Project Nodes and Communication Process	5
3.3 Proposed Algorithm for Autonomous Navigation	6
4 GUI Implementation	7
5 Mapping of Environment	7
6 Computer Vision	7
7 Hardware Node and Low-Level Control	7
A Solid Design Parts	8

List of Figures

1	A 3-D model for robot frame	1
2	Different applications for mobile robots; industrial, military and transport	2
3	Clustered view of main system structure	3
4	ROS logo.	4
5	The implemented nodes and advertised topics between them.	5
6	Communication between hardware and mapping node over hw_map topic.	6

1 Introduction

In daily life there are a lot of situations in which robots are needed to perform some tasks man can not deal with. Some of these situations may be risky, difficult or such impossible for man to do. Think about a risky place we want to discover like disasters area, places of extreme environmental conditions or military purposes. In these situations introducing a robot is important for saving human life.

Robots also can help people of special needs with what they can not do like carrying heavy things, holding and placing parts or even home cleaning.

Robots have many configurations, styles and mechanisms for motion. Some are legged, others are wheeled and the rest can fly, swim or dive. Each configuration has its functionality that others can not do and also has limitations. In our project we introduce a robot with good navigation and localization technique to solve a lot of problems mentioned above in this section.

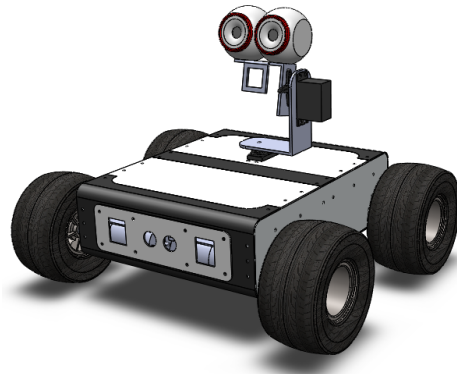


Figure 1: A 3-D model for robot frame

1.1 History of Mobile Robots

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. Mobile robots can be "autonomous" (AMR - autonomous mobile robot) which means they are capable of navigating an uncontrolled environment without the need for physical or electro-mechanical guidance devices. Alternatively, mobile robots can rely on guidance devices that allow them to travel a pre-defined navigation route in relatively controlled space (AGV - autonomous guided vehicle). By contrast, industrial robots are usually more-or-less stationary, consisting of a jointed arm (multi-linked manipulator) and gripper assembly (or end effector), attached to a fixed surface.

Mobile robots have become more commonplace in commercial and industrial settings. Hospitals have been using autonomous mobile robots to move materials for many years.

Warehouses have installed mobile robotic systems to efficiently move materials from stocking shelves to order fulfillment zones. Mobile robots are also a major focus of current research and almost every major university has one or more labs that focus on mobile robot research. Mobile robots are also found in industrial, military and security settings. Domestic robots are consumer products, including entertainment robots and those that perform certain household tasks such as vacuuming or gardening. [2]



Figure 2: Different applications for mobile robots; industrial, military and transport

1.2 Project Objectives

The main objective of this project is to demonstrate a robust mobile robot in a small scale to perform an autonomous navigation from point to another. Also it is required for the robot to localize itself when asked to do. Both navigation and localization depends on the ability of robot to detect and recognize texts on landmarks that uniquely identifies specific nodes in the map. For new environments whose map is not known, the robot can be guided remotely over the network to explore that location and a camera is provided for both streaming live video for the site around robot and to perform the computer vision task. In the hardware level a robust controller is required to perform motion instructions with acceptable precision.

1.3 Limitations

During our work we faced a lot of problems associated mostly with sensors. As we the main part in any control system is the feedback. This is because if we got a wrong indication for current state of system, we will perform a wrong reaction and the error increases more and more. In our project we need sensors to get information about the robot like position, velocity and orientation. Any missing part of them leads to both wrong estimation of state and wrong controller action. In next sections we will talk about our trials, results and algorithms implemented to get the advantage of each sensor and avoid its misleading data. But lets start from a high-level point of view and gradually take important topics with some details.

1.4 Overview on the project parts

The main system structure of our project as shown in figure 3 consists of a GUI and controller box at user side and the master unit (RPi) at robot side. The operation starts from GUI to select the function needed and then a flow of communication commands are passed to master unit over the network to perform the required task. Controller box is used in manned mode of operation to guide robot motion and camera orientation. In next chapter we talk about the GUI; how you can use, how it is implemented and a quick over view on ROS (Robots Operating System).

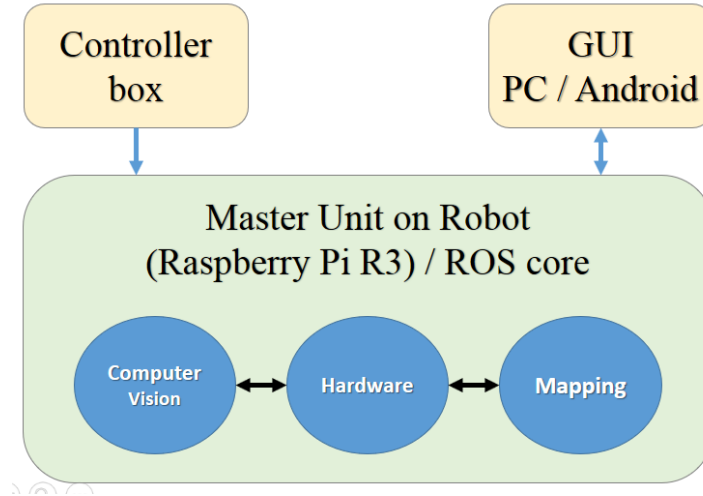


Figure 3: Clustered view of main system structure

2 User Manual and GUI

talk about how to use the GUI to run the project on different modes.

3 System Structure and Proposed Algorithm

In the design stage of project, three main parts arises to be implemented; mapping, computer vision and hardware. Each part represents an executable program that can interact with other parts in a manner to fulfill the required task. So, lets first talk briefly about ROS and how it provided a great help in connecting project parts and how communication between processes becomes easy.

3.1 Robots Operating System (ROS)

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. [1]

In ROS, Node is a common word that represents the executable file. So, in our project we have three main nodes; mapping, computer vision and hardware nodes. communication between nodes in ROS can be performed in many ways. The method we worked with is the message communication. Message represents the ROS data type. We developed a special type that can handle all communication needs between nodes. It is called Instruction and consists of a string variable that holds the command name and two float arguments.

Nodes can deal with messages in two manners; as a publisher, subscriber or both. Another word commonly used in ROS is the topic. It is considered as an intermediate program that holds any published message and forward it to all nodes that subscribed for it. This feature is very useful as a node can perform just one publishing command and any number of nodes can receive it. Now we know a bit about how ROS works and for more details and tutorials you can visit ROS tutorials site: <http://wiki.ros.org/ROS/Tutorials>.

In next section we are to talk about our project nodes and how communication performed among them.



Figure 4: ROS logo.

3.2 Project Nodes and Communication Process

As mentioned before, we have three main nodes; mapping, computer vision and hardware. the communication between them were created as shown in figure 6.

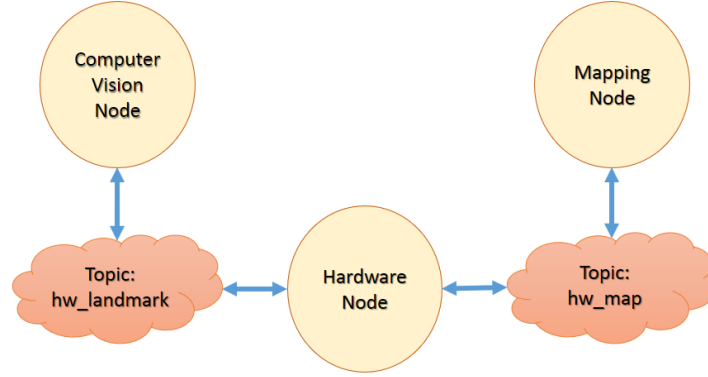


Figure 5: The implemented nodes and advertised topics between them.

There are two advertised topics; `hw_landmark` that connects hardware with computer vision and `hw_map` that connects hardware to mapping node. Over the `hw_map` topic the mapping node publishes the instructions that guide the robot from a landmark to another. When the robot reaches a new landmark, a tuning loop starts between hardware and computer vision node to put the robot exactly in front of the landmark. In this way, errors caused by inaccuracy of sensors or hardware motion performance are eliminated periodically resulting in a good autonomous navigation of the robot. In next section we are to talk about the navigation algorithm and the sequence of instructions with some details.

3.3 Proposed Algorithm for Autonomous Navigation

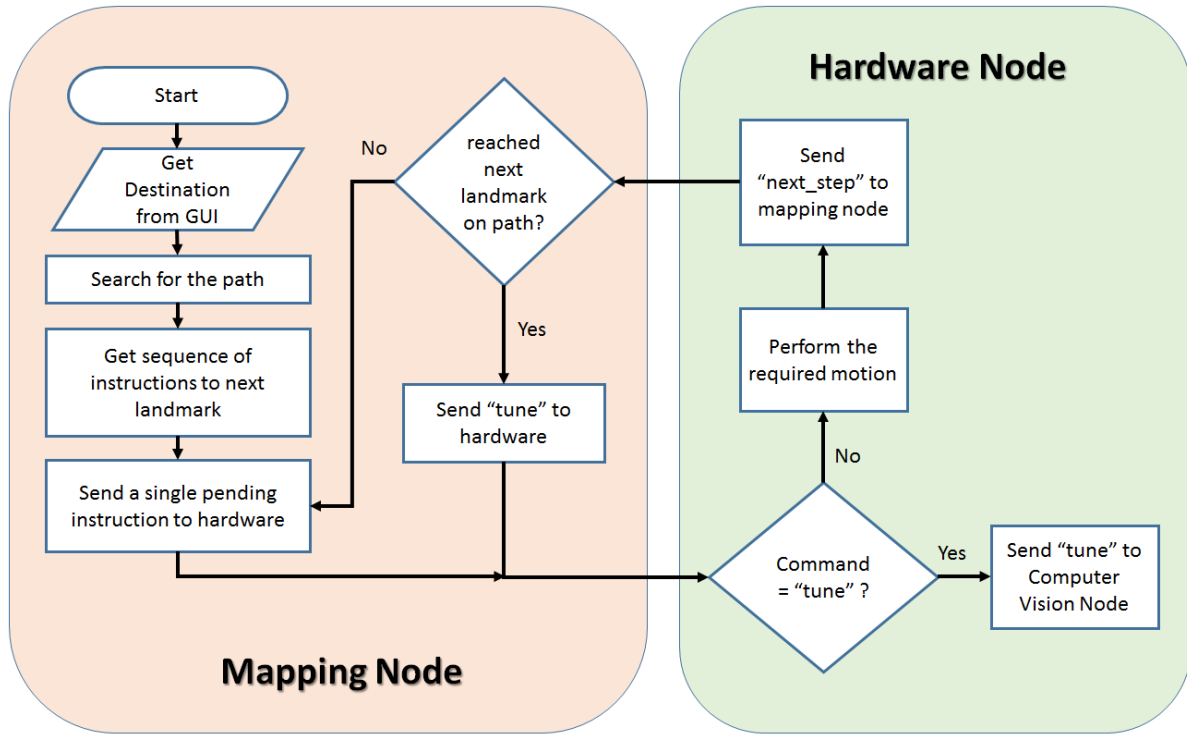


Figure 6: Communication between hardware and mapping node over hw_map topic.

4 GUI Implementation

talk about the Java application and how it is implemented, how it communicates with master unit to send navigation instruction. also how video stream is received

5 Mapping of Environment

talk about graph theory and how map is represented. how landmarks are included in the map. Search algorithms. instructions created to be sent to hardware.

6 Computer Vision

talk about how detection, recognition and tuning are processed.

7 Hardware Node and Low-Level Control

mechanical analysis, solid design, final assembly, Motors choice. Electric components, circuits, structure. Sensors Control system for straight line and rotation motion.

A Solid Design Parts

References

- [1] ROS. About ros. <http://www.ros.org/about-ros/>, Mar 2017.
- [2] Wikipedia. Mobile robots. https://en.wikipedia.org/wiki/Mobile_robot, Jan 2015.