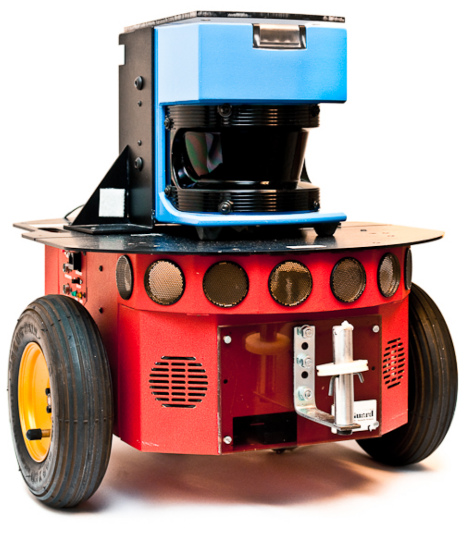
System Description
ABB Mail Robot

Version 1.0

Author: Martin Melin
Date: September 24, 2010



Course name:	Control Project	E-mail:	danba185@student.liu.se
Project group:	ABB MailMen	Document responsible:	Martin Melin
Course code:	TSRT10	Author's E-mail:	marme287@student.liu.se
Project:	ABB Mail Robot	Document name:	SystemDescription_v1.pdf

Status

Reviewed		
Approved		

Project Identity

Group E-mail:	danba185@student.liu.se
Homepage:	under construction
Orderer:	Karl Granström, Linköping University **Phone:** +46 13 281333, **E-mail:** karl@isy.liu.se
Customer:	Marcus Pettersson, ABB Västerås **Phone:** +46 21 345194, **E-mail:** marcus.pettersson@se.abb.com
Course Responsible:	David Törnqvist, Linköping University **Phone:** +46 13 281882, **E-mail:** tornqvist@isy.liu.se
Project Manager:	Daniel Barac
Advisors:	André Carvalho Bittencourt, Linköping University **Phone:** +46 13 282622 , **E-mail:** andrecb@isy.liu.se

Group Members

Name	Responsibility	Phone	E-mail (@student.liu.se)
Daniel Barac (DB)	Project Leader	+46 702641857	danba185@student.liu.se
Martin Melin (MM)	Document responsible	+46 702887793	marme287@student.liu.se
Erik Erkstam (EE)	Software responsible	+46 737266812	erier982@student.liu.se
Hugo Kinner (HK)	Test responsible	+46 733688378	hugki634@student.liu.se
Manfred Hallström (MH)		+46 703679569	manha932@student.liu.se
Simon Hugosson (SH)	Control responsible	+46 708778471	simhu610@student.liu.se
Niklas Carlsson (NC)	Information responsible	+46 709601672	nikca291@student.liu.se
Nicklas Forslöw (NF)	SLAM responsible	+46 739516430	nicfo307@student.liu.se

Document History

Version	Date	Changes made	Sign	Reviewer
0.1	2010-09-16	First draft	SH	SH, MH
0.2	2010-09-20	Second draft	SH, NF	SH, MH, NF
0.3	2010-09-22	Third draft	MM SH	MM EE
1.0	2010-09-24	First Version	SH MM	MM EE

Contents

1 Introduction **1**

2 Overview of the System **1**

2.1 Subsytems . . . 1

2.1.1 Robot . . . 1

2.1.2 Laptop . . . 1

3 The Robot **2**

3.1 Odometers . . . 2

3.2 Laser Sensor . . . 2

3.3 Sonar Sensors . . . 3

4 The Software **3**

4.1 SLAM . . . 3

4.2 Trajectory planning . . . 4

4.3 Control . . . 4

1 Introduction

This document gives an overview of how the autonomous mail robot is going to be designed. It specifies what hardware is used and the basic structure of the software.

2 Overview of the System

The system consists of a MobileRobots P2-DX robot with a HP ProBook 6450b laptop mounted on it. The laptop runs the application that controls the robot and could also be used as an interface if a user wants to manually control the robot or view the robot's map. The robot and the laptop communicate via a RS232 serial port.

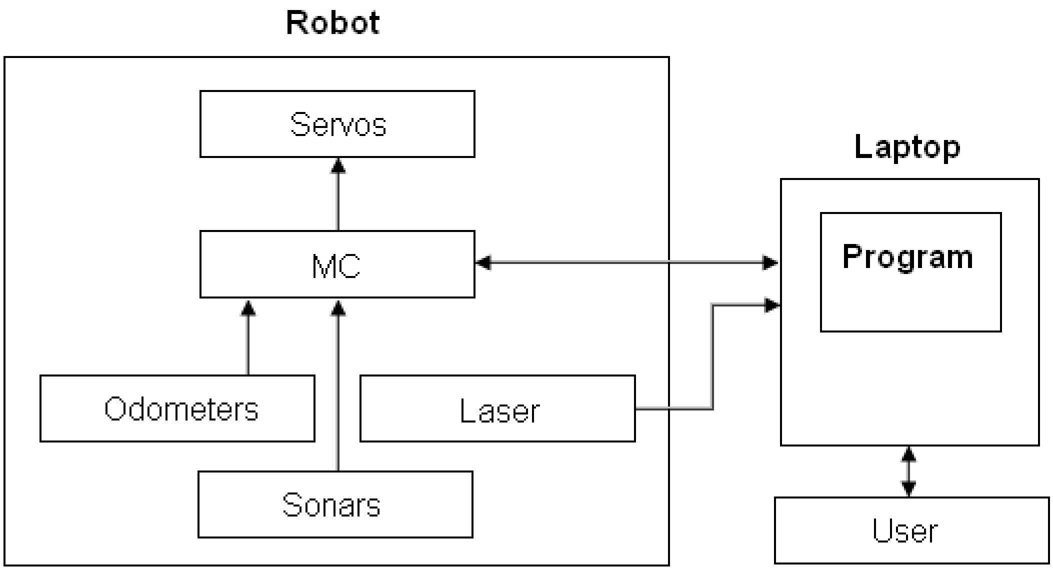


Figure 1: Overview of the system.

2.1 Subsytems

This section includes a short description of the subsystems.

2.1.1 Robot

The robot is a wheeled MobileRobots P2-DX robot equipped with odometers, a laser, sonars and a microcontroller (MC). The microcontroller receives data from the sonars and odometers. It also sends control signals, calculated by the software on the laptop, to the servos.

2.1.2 Laptop

The laptop receives sensor data from the laser and the robot's microcontroller. It is also responsible for the software which handles SLAM, trajectory planning and calculation of the control signals.

3 The Robot

The robot has two drive wheels with encoders for measuring their angular position/speed. It also has two range sensors, a sonar array and a Sick LMS200 laser range finder, both facing forward and with a field of view of 180°. The laptop is connected to a microcontroller in the robot that is responsible for reading sonar and odometry data as well as controlling the motors. The range data from the laser is sent to the laptop through a RS232 connection.

Figure 2: MobileRobots P2-DX robot.

3.1 Odometers

The robot wheels have 800 tick quadrature shaft encoders that is used to provide wheel position and speed estimates.

3.2 Laser Sensor

The SICK LMS200 has a field of vision of 180° and an angular resolution that can be set to 0.25°, 0.5° or 1°. The resolution in the range measurements is 10 mm and the accuracy typically ±15 mm. The range is 10 m for 10 reflectivity, which corresponds to cardboard, and can be up to 32 m for more reflective materials. The sampling rate is around 4 Hz.

3.3 Sonar Sensors

The robot has an array of eight sonars in directions ±10 °, ±30 °, ±50 ° and ±90 ° (where 0° is straight forward). Both the acquisition rate and the sonar gain are adjustable. The acquisition rate is 25 Hz by default and the sensitivity ranges from 10 cm to over 4 m, depending on the gain. One important feature of the sonar is that it can detect transparent objects, which the laser can not.

4 The Software

The application is written in C++ and uses the ActivMedia Robotics Interface for Application (ARIA) library for all communication with the robot. ARIA has utilities for both receiving data from the sensors and controlling the motors.

The program can be divided into three subsystems with specific tasks according to the following figure.

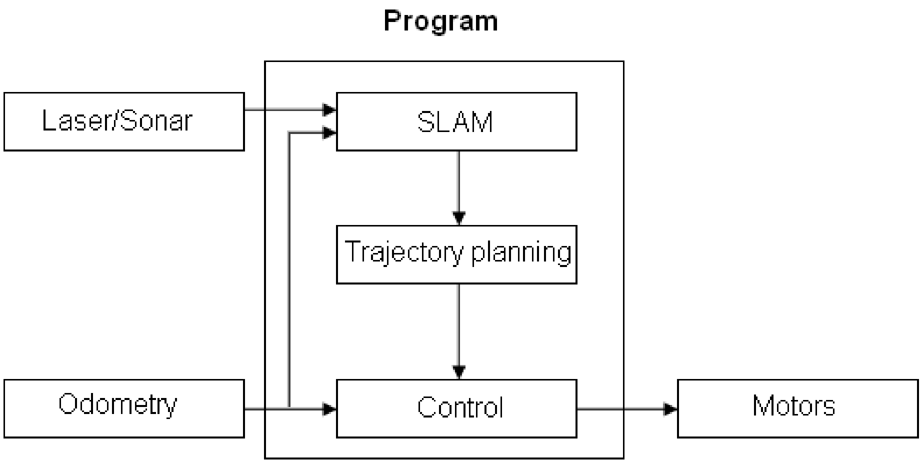


Figure 3: Program structure.

4.1 SLAM

This subsystem is responsible for the simultaneous localization and mapping (SLAM). Sensor fusion is used to estimate the robots position and heading in a map and to update the map as changes are observed. Here, sensor models and a motion model of the robot together with data from all the sensors are used.

Since the laser is the most accurate of the two sensors its measurements will have a greater impact on the estimated robot pose (position and heading) and map. A problem with the laser is the inability to detect certain objects, e.g. glass. The detection of these objects will be the main task for the sonars.

The map that the program uses can either be an existing one that has been translated to the appropriate representation or be created while a user drives the robot around. The most likley map representation to be used is an occupancy grid map. Here, the map is divided into a number of cells of equal size where each cell has a probability of being occupied or not.

The SLAM system receives odometry data and range data from the laser and sonar. It then produces a map that is sent to the trajectory planning unit. The estimated position and heading of the robot is also sent to the trajectory planning unit.

4.2 Trajectory planning

The purpose of this subsystem is to find a route from the current position to a given point in the map. If a user has defined forbidden zones in the map the program should consider this when planning the route. The trajectory planner gets a map and robot pose from the SLAM system and sends a route to the control system.

4.3 Control

When the route is planned the program will use a control algorithm to make the robot follow the route. To avoid collisions the robot will take appropriate actions (slowing down, stopping, perhaps replanning the route) when detecting unexpected new objects, such as humans.

The control system receives a route from the trajectory planner and sends control signals to the motors. It also uses feedback from the odometers to make sure it keeps on track.