

UNIVERSITY OF OSLO

CONTROL OF MOBILE ROBOTS

UNIK4490

Odometry and Posture Regulation for a 4 by 4 mobile robot

Autumn 2017

Authors

Eirik KVALHEIM, Daniel SANDER
ISAKSEN and Torgrim R. NÆSS

Supervisors

Dr. Kim MATHIASSEN and
Magnus BAKSAAS





1 Introduction

The main goal of this project was to implement motor control, posture regulation and odometric localization in order to get the robot to move to a desired pose. We were working on an existing software stack running ROS (Robot Operating System), so getting to know the previous software as well as ROS as a system, was also a goal in this project.

A significant portion of the project time was spent on reverse engineering the robot to better understand the system in order to implement our own solutions. In the end we had also spent a lot of time on tuning the regulator parameters, as well as compensating for odometric faults. Most of the work was done in collaboration with another group that was working on an identical robot with the same software stack.

2 The System

Figure 1 shows a block diagram of the system. It consists of posture regulation, motor controller, kinematic model, and odometry. The following subchapters describes will describe this in more detail.

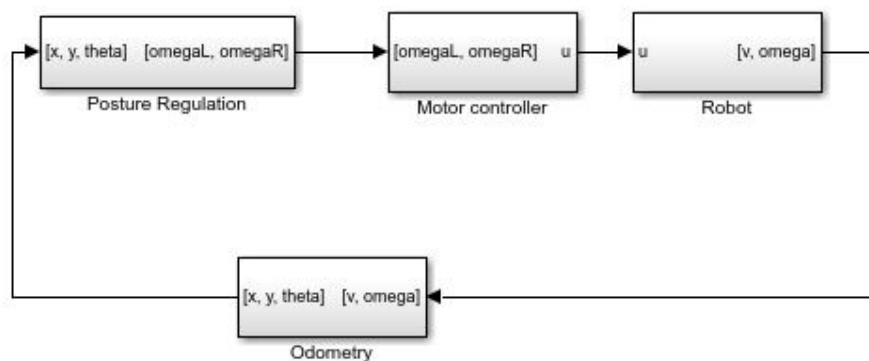


Figure 1: Block diagram of the system.

2.1 Kinematic Model

We used the kinematic model of a unicycle from page 478 in the book, where v is the driving velocity and ω is the steering velocity.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (1)$$

2.2 Motor Control

We started our project by running and reverse engineering the mobile robot together with the other group. As we were not familiar with the system and since there were no README or comments in the code, the challenges for reverse engineering the motor controller was: to login, to run robot, understand the controller and deduce the reason behind the constants in the controller. The implementation of motor control for each wheel was already available in the robots source.

2.3 Odometric Localization

$$x_{k+1} = x_k + \frac{v_k}{\omega_k} (\sin\theta_{k+1} - \sin\theta_k) \quad (2)$$

$$y_{k+1} = y_k + \frac{v_k}{\omega_k} (\cos\theta_{k+1} - \cos\theta_k) \quad (3)$$

$$\theta_{k+1} = \theta_k + \omega_k T_s \quad (4)$$

2.4 Posture Regulation

In general the posture regulation controller takes in the configuration variables, $q = [x, y, \theta]^T$, and outputs v and ω . It is assumed that the desired variables are $q_d = [0, 0, 0]^T$ and the error from q_d is represented by

$$\rho = \sqrt{x^2 + y^2} \quad (5)$$

$$\gamma = \text{Atan2}(y, x) - \theta + \pi \quad (6)$$

$$\delta = \gamma + \theta \quad (7)$$

where $\rho = \|\vec{e}_p\|$ is the distance between current point (x, y) and desired point $(0, 0)$, γ is the angle between \vec{e}_p and the sagittal axis of the vehicle and δ is the axis between \vec{e}_p and the x-axis. v and ω are found by:

$$v = k_1 \rho \cos(\gamma) \quad (8)$$

$$\omega = k_2 \gamma + k_1 \frac{\sin(\gamma) \cos(\gamma)}{\gamma} (\gamma + k_3 \delta) \quad (9)$$

In our implementation of the controller we get \vec{q} from the odometric module and output ω_R and ω_L to the motor controller. Equations for ω_R and ω_L expressed by error variables ρ , γ and δ , by setting the following equations (3) and (4) equal to equations (1) and (2) respectively,

$$v = \frac{r(\omega_R + \omega_L)}{2} \quad (10)$$

$$\omega = \frac{r(\omega_R - \omega_L)}{d} \quad (11)$$

and then solve for ω_R and ω_L by the inserting method. This yields:

$$\omega_R = \frac{2k_1 \rho \cos(\gamma)}{2r} + \frac{dk_2 \gamma}{2r} + \frac{d \sin(\gamma) \cos(\gamma) (\gamma + k_3 \delta)}{2r \gamma} \quad (12)$$

$$\omega_L = \frac{2k_1 \rho \cos(\gamma)}{2r} - \frac{dk_2 \gamma}{2r} - \frac{d \sin(\gamma) \cos(\gamma) (\gamma + k_3 \delta)}{2r \gamma} \quad (13)$$

3 Testing and Results