UNIVERSITY OF OSLO

CONTROL OF MOBILE ROBOTS

UNIK4490

# An Unnecessarily Extra Long Convoluted Academic Title That Makes Little Sense


*Authors*
Daniel SANDER ISAKSEN, Eirik KVALHEIM and Torgrim R. NÆSS

*Supervisors*
Dr. Kim MATHIASSEN and Magnus BAKSAAS


November 21, 2017

# 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 do a desired pose. Another important part of the project was to learn about ROS to make it possible to implement ROS nodes for the tasks above.
A significant portion of the project time was spent on reverse engineering the robot to better understand the system before we could begin implementing our own solutions. Most of the work was done in collaboration with the other group, as we were working on the same robot while trying to get the system up and running.

# 2 The System

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

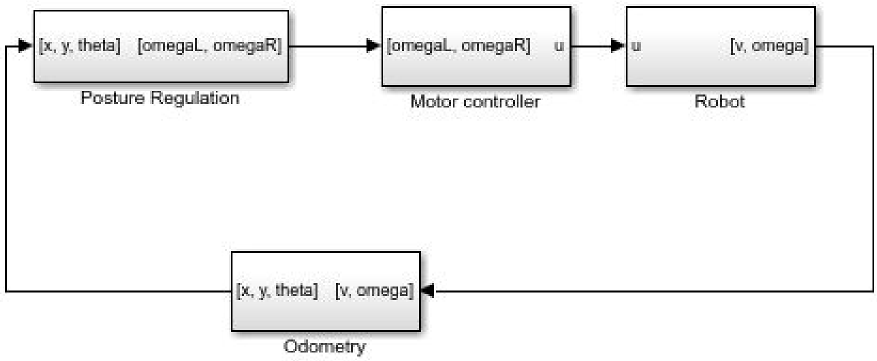


Figure 1: Block diagram of the system.

## 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 $\omega$ is the steering velocity.

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

## 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 avaiable in the robots source.

## 2.3 Odometric Localization

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

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

$$\theta_{k+1} = \theta_k + \omega_k T_s \tag{4}$$

### 2.4 Posture Regulation

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

$$\rho = \sqrt{x^2 + y^2} \tag{5}$$

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

$$\delta = \gamma + \theta \tag{7}$$

where $\rho = ||\vec{e_p}||$ is the distance between current point $(x, y)$ and desired point $(0, 0)$, $\gamma$ is the angle between $\vec{e_p}$ and the sagittal axis of the vehicle and $\delta$ is the axis between $\vec{e_p}$ and the x-axis. v and $\omega$ are found by:

$$v = k_1 \rho cos(\gamma) \tag{8}$$

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

In our implementation of the controller we get $\vec{q}$ from the odometric module and output $\omega_R$ and $\omega_L$ to the motor controller. Equations for $\omega_R$ and $\omega_L$ expressed by error variables $\rho$, $\gamma$ and $\delta$, by setting the following equations (3) and (4) equal to equations (1) and (2) respectively,

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

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

and then solve for $\omega_R$ and $\omega_L$ by the inserting method. This yields:

$$\omega_R = \frac{2k_1\rho cos(\gamma)}{2r} + \frac{dk2\gamma}{2r} + \frac{dsin(\gamma)cos(\gamma)(\gamma + k_3\delta)}{2r\gamma} \tag{12}$$

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

## 3 Testing and Results