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

Some Heading

Figure 1: 4 by 4 Rover

The main task of the project was to make the robot move to a desired pose, and this was divided into four subtasks/stages as listed below the following summary:
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. We decided to keep the original PID motor control code and write code for odometric localization and posture regulation. 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.
(Note! One important subtask that is not mentioned in the list was to create a system of ROS nodes, for which we had to learn about ROS.)

1. Implement motor control for each wheel
   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. Implement odometric localization

3. Implement posture regulation motion control
   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 following variables:

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

$$\gamma = Atan2(y, x) - \theta + \pi$$

$$\delta = \gamma + \theta$$

   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:

$$v = k_1 \rho cos(\gamma) \tag{1}$$

$$\omega = k_2 \gamma + k_1 \frac{sin(\gamma)cos(\gamma)}{\gamma}(\gamma + k_3 \delta) \tag{2}$$

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,

$$v = \frac{r(\omega_R + \omega_L)}{2} \tag{3}$$

$$\omega = \frac{r(\omega_R - \omega_L)}{d} \tag{4}$$

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

$$\omega_R = \frac{2k_1\rho cos(\gamma)}{2r} + \frac{dk2\gamma}{2r} + \frac{dsin(\gamma)cos(\gamma)(\gamma + k_3\delta)}{2r\gamma} \tag{5}$$

$$\omega_L = \frac{2k_1\rho cos(\gamma)}{2r} - \frac{dk_2\gamma}{2r} - \frac{dsin(\gamma)cos(\gamma)(\gamma + k_3\delta)}{2r\gamma} \tag{6}$$

4. Move the robot to a pose