Universiteit van Amsterdam

Autonomous Mobile Robots

First assignment

OpenLoop Steering Assignment

Ruben Janssen, 10252657
David van Erkelens, 10264019
Laurens Verspeek, 10184465

Department of Computer Science
University of Amsterdam

November 24, 2013

Contents

1 Exercise 1: Motion of Differential-Drive and Omni-Drive Robot

1.1 Kinematic model

The following formula is given:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_1, \dot{\varphi}_2, l, r, l_c, d, r_c, \ldots) \tag{1}$$

When applying this to the differential drive robot, r denotes the diameter of the wheels and l denotes the distance between the origin of the robot and the wheels. l_c and r_c denote these distances for the castor wheel. The d denotes the distance between the wheel contact point and the center of rotation of the castor wheel. $\dot{\varphi}_1$ and $\dot{\varphi}_2$ denote the rotation speed of the wheels. In figure **??**, the allocation of $\dot{\varphi}_1$ and $\dot{\varphi}_2$ to the wheels is shown. In case of the omni-drive robot, $\dot{\varphi}_n$ denotes the movement of the n^{th} wheel and l and r have the same meaning as with the differential-drive robot.

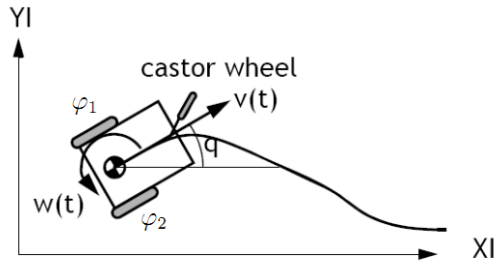

Figure 1: The allocation of $\dot{\varphi}_1$ and $\dot{\varphi}_2$

Moving onwards with formula **??**, we can establish the kinematics of the differential-drive robot (shown in formula **??**) and the omni-drive robot (shown in formula **??**).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \dot{\varphi}_1, \dot{\varphi}_2) \tag{2}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \dot{\varphi}_1, \dot{\varphi}_2, \dot{\varphi}_3) \tag{3}$$

In the formula for the differential drive robot, l_c is left out since the influence of the castor wheel on the movement is negligible.

1.1.1 Odometry of differential drive robot

a The kinematic model for the differential-drive robot is:

$$\xi_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2l} & \frac{-r}{2l} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} \tag{4}$$

This equation states te relation between the movement of the wheels ($\dot{\varphi}_1$, $\dot{\varphi}_2$) and the developement of the position ($\dot{x}$,$\dot{y}$) and orientation $\dot{\theta}$ of the robot in the plane.

b To find out how the speeds of the two wheels ($\dot{\varphi}_1$, $\dot{\varphi}_2$) are related to the rotation w and heading speed v of the robot, it is necessary to rewrite the kinematic model for the differential-drive robot:

$$\xi_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} & 0 \\ 0 & 0 & 1 \\ \frac{r}{2l} & \frac{-r}{2l} & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \end{bmatrix} \tag{5}$$

This yields the same result, but makes certain matrices inversible. Then, the required speeds can be calculated as following:

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} & 0 \\ 0 & 0 & 1 \\ \frac{r}{2l} & \frac{-r}{2l} & 0 \end{bmatrix}^{-1} R(\theta) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{6}$$

This allows us to determine the speed of both wheels. Filling in this equation, it comes clear that when the rotation speeds of the wheels differ, the orientation of the robot ($\dot{\theta}$) will change. In other words, the robot will rotate. The larger the difference in speeds, the bigger the rotation of the robot is.

Similarly, a higher heading speed v will result into higher speeds of te wheels. larger speeds of the wheel will result into a higher heading speed v. Filling in higher speeds ($\dot{x}$,$\dot{y}$) into the equation, heading speeds ($\dot{\varphi_1}$,$\dot{\varphi_2}$) will increase lineairly.

The precision of the position 'estimate' will increase when Δt decreases. If a robot moves, an estimate made is always wrong by a maximum of Δt. Decreasing Δt will reslut into a more real-time estimation of the position of the robot.

2 Preparation for a real experiment

Unfortunately, due to a lack of time. Our implementation does not use the kinematic model provided earlier. It is based on a different approach but the results are very good.

a First, the circle to be followed has to be determined, using a radius r. Then, the number of degrees to follow the circle should be determined, as g. To move the robot as desired, the distance to travel with each wheel should be calculated.

$$\begin{aligned} D_1(r) &= 2\pi \times (r - l) \\ D_2(r) &= 2\pi \times (r + l) \end{aligned} \tag{7}$$

In which D_1 notes the wheel closest to the center of the circle, and D_2 the wheel farthest from the center of the circle. Now the number of rotations to travel this distance should be calculated. This is done by dividing by the diameter d of the wheels:

$$\begin{aligned} Rotations_1(r) &= \frac{D_1(r)}{d_{wheel}} \\ Rotations_2(r) &= \frac{D_2(r)}{d_{wheel}} \end{aligned} \tag{8}$$

The speed of the wheel closest to the center of the circle will be lowered, so $\frac{\varphi_1}{\varphi_2} = \frac{Rotations_1(r)}{Rotations_2(r)}$. Then, φ_1 will become $\varphi_1 \times \frac{Rotations_1(r)}{Rotations_2(r)}$. After this, the `Tacho`-limit of the of the wheels will be set to the values of $Rotations_1(r)$ and $Rotations_2(r)$. Since $\frac{\varphi_1}{\varphi_2} = \frac{Rotations_1(r)}{Rotations_2(r)}$, both wheels will finish their rotations on the same moment.

To make the robot drive in a straight line, only the `Tacho`-limit is of importance. The `Tacho`-limit can be defined as a function f in a distance to travel t:

$$f_n(t) = 360 \times Rotations_n(r) \tag{9}$$

b In order to implement a feedback-loop, the `Tacho`-limit is removed. Instead, every 0.1 the value of both wheels is read to check if they already meet the value which would have been the `Tacho`-limit. If the values are incorrect, movement is reversed or continued. If the read value equals the desired regarding a certain threshold, movement of the wheel is stopped.

c For the Matlab implementation, see `circle.m`, `straightLine.m`, `drive.m` and `henk.m` (named after the robot).

3 Experiments

In order to test the code and formulas, the robot was tested in the *Robolab*. In the *Robolab* lies a football field used for Naos, but that field was perfectly suitable for testing our NXT. The NXT was programmed to follow the circle in the center of the field, which has a diameter of 60cm. In figure **??**, the path of the NXT across this line is shown. On the pictures the NXT has a small offset from the line, but this can be explained by a faulty starting position.

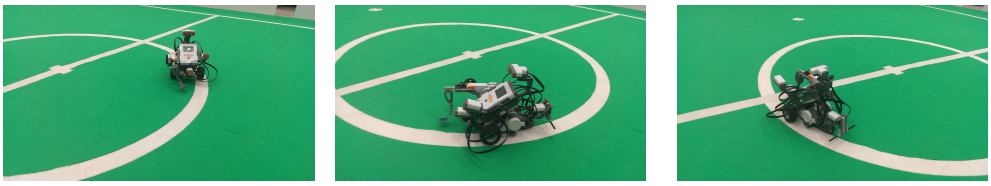

Figure 2: The NXT driving along the line

4 Conclusions

Looking at the experiments, it can be concluded that the formulas and code function, because the NXT was able to follow the line on the field pretty closely.