31385 Autonomous Robots Systems
Kinematics and Odometry

(rev 1.1)

1 Objective

The objective of this exercise is to give an introduction to kinematics and odometry for mobile robots

When you have finished this exercise you will be able to:

- Understand kinematics and odometry for a differential drive robot
- Find the odometry parameters using UMB-mark

2 Material and preparation

The material for this exercise is found in 'Where am I' section 1 and 5 and UMB-mark chapter 6.

3 Calculation of vehicle position.

The aim is to calculate and control where we are and where we are going. Kinematics is the study of motion without regard to the forces which cause it. Within kinematics one studies the position, velocity and acceleration etc. First we calculate where we are based on encoder measurements. This is called the odometry model.

The incremental displacement of each wheel

$$\Delta U_{L/R}(i)=c_m N_{L/R}(i) \tag{1}$$

where $c_m=\pi D/nC$ and

D is the diameter of the wheels (0.065m)

n is the gear ratio between encoder and wheel(1) and,

C is the number of pulses per revolution(2000).

This displacement of the center point of the robot

$$\Delta U(i)=(\Delta U_R+\Delta U_L)/2 \tag{2}$$

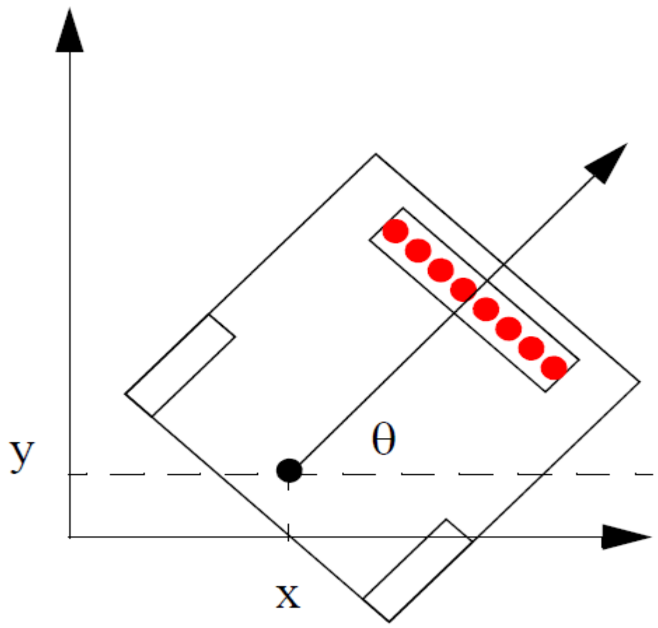

Figure 3.1:Update of the AG pose

This displacement of the center point of the robot

$$\Delta U(i) = (\Delta U_R + \Delta U_L)/2 \tag{3}$$

Incremental change in orientation

$$\Delta\theta(i) = (\Delta U_R - \Delta U_L)/b \tag{4}$$

Where b is the wheel distance (0.26m)
The new pose of the robot is

$$x(i) = x(i-1) + \Delta U(i)\cos(\theta(i)) \tag{5}$$

$$y(i) = y(i-1) + \Delta U(i)\sin(\theta(i)) \tag{6}$$

$$\theta(i) = \theta(i-1) + \Delta\theta(i) \tag{7}$$

The equations show that we based on measurement of the wheel movements are able to calculate the vehicles position at a given time knowing the start position and a number of vehicle parameters. The accuracy of the calculated position depends among other things on the accuracy of these parameters.

A method to find these parameters (UMB-mark) is described in 'where am I' Chapter 5 and UMB-mark chapter 6.

UMB-mark finds the ratio Eb between the real width and the nominal width between the wheels and the ratio Ed between the diameter of the right wheel and the diameter of the left wheel Dr/Dl.

3.1 Matlab script to calculate Eb and Ed

Make a matlab script that takes the errors from a clockwise experiment Xcw and Ycw and the errors from a counter clockwise experiment Xccw and Yccw

and the side length L and the nominal width as inputs and calculate Ed and Eb as output.

The calculations should be based on the formulas 6.22 to 6.29. α and β should be calculated as the mean of the values based on the x-coordinates and the y-coordinates.

Test the script with

(Xcw, Ycw, Xccw, Yccw, L, b0)=(0.20, 0.20, 0.12, -0.12, 1 ,0.26)

The result should be

(Ed, Eb)=(0.9948, 0.9515)

3.2 UMB-mark in the simulator.

In this problem Ed and Eb should be found for a simulated SMR. Go to the sim directory and start the simulator:

simserver simconfig.odo.xml

Go to the mrc directory and run a clocwise and a counter clockwise square with side length 3 m. If you click on the robot in the simulator the world coordinates of the smr wil be shown. The square runs are started by running

mrc -s0

Restart the simulator between the to runs.

Use the found x and y errors and your matlab script to find Ed and Eb.

To use the found values we need to create a calibration file. Start the simulator and run mrc in the mrc directory:

mrc -s0 -c

chose update odometry and enter 0 for the coordinates and 1 for the side length. This will generate a calibration file in the calib directory with the name:

localhost_demo_odo_calib.dat

this is a text file containing three numbers:

the width between the wheels

Ed

the average travelled distance per encoder tick

use your value of Eb and the value in the file to calculate the correct width.

Change the numbers in the file to your values.

Run the squares again and see if the accuracy is improved.

Delete the calibration file and try to do the odometry update with your X and Y errors and see what you get in the calibration file.

Run a UMB-mark experiment with the improved values above and see if this improves the final result.