31385 Autonomous Robot Systems
Calibration

(rev 1.1)

1 Objective

The objective of this exercise is to teach how to calibrate the line sensor and the ir-sensors

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

- run the robot with smr-cl scripts
- Calibrate the odometry

2 Calibration

2.1 Calibration of linesensor.

To calibrate the linesensor the sensors response to black paper and white paper is measured. Based on these measurements a linear transformation is found that maps black to 0 and white to one.

Write a smr-cl script that logs at least 100 linesensor measurements at the starting position and the moves 0.3 meters forward and takes another 100 measurements.

Use a variable that is also logged to indicate the state of the program.
Make a matlab script that loads the logfile and extracts the white and black measurements from the file and calculate the linear calibration transformation from raw values to values between 0 and 1 for each sensor based on the black and white measurements.

Run the script in the simulator and calculate the calibration transformations.

    simserver simconfig.calib.xml

2.2 Calibration of IR-distance sensors

To calibrate the front IR-sensors the robot is placed approximately 75 centi-meters from a wall and then forwarding in steps of 10 cm until a distance of 15 cm to the wall is obtained stopping and making 100 ir-measurements after each forward step.

Make a smr-cl script that implements this algorithm and stores the measurements in the log file.

The wall in simconfig.calib.xml has the x-coordinate 1.00 m and the ir-sensors are places 0.235 meters in front of the robot origo.

Run the script in the simulator

simserver simconfig.calib.xml

save the log file.

The output of the sensor, ir_{out} is given as:

$$ir_{out} = K1 * \arctan\left(\frac{K2}{dist}\right) + K3 \tag{1}$$

where *dist* is the distance to the object, *K1* is a scaling factor that limits the maximum output to 255, *K2* is the distance between the source and the PSD, and *K3* is an offset.

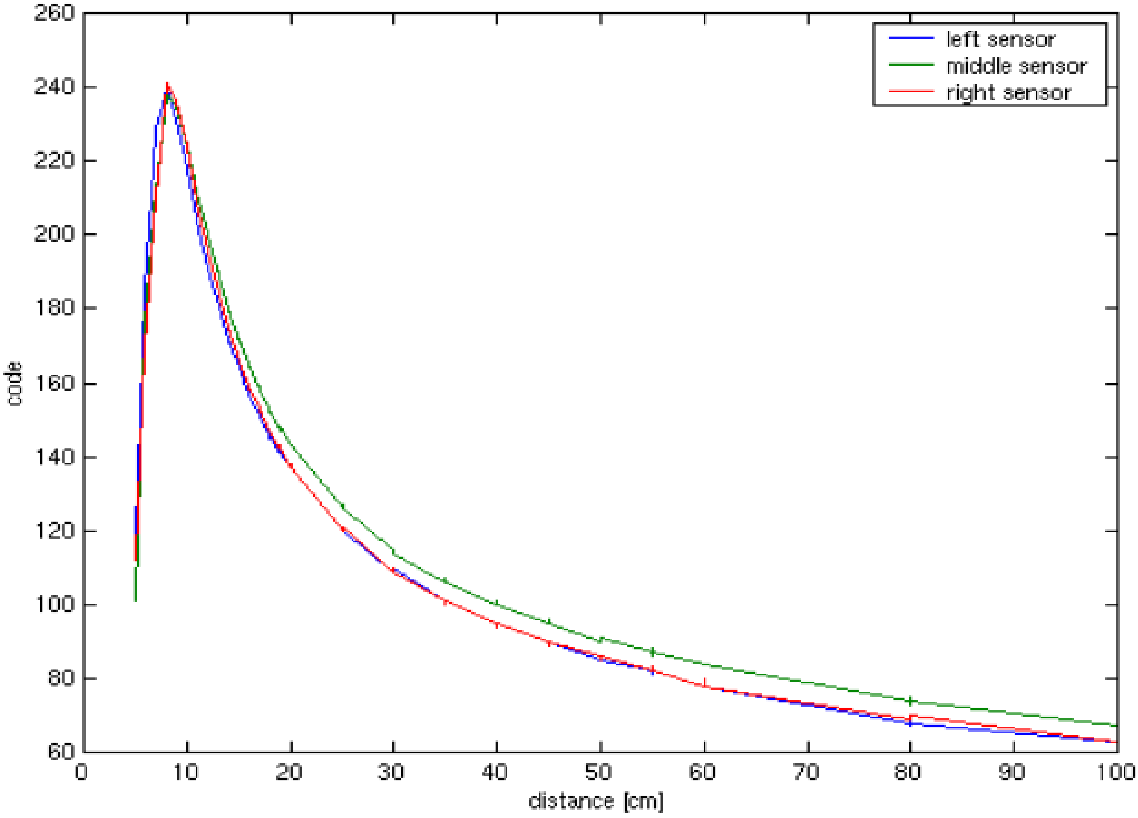
This can approximately be simplified to

$$ir_{out} = \frac{K_a}{dist} + K_b \tag{2}$$

To find the distance *dist*

$$dist = \frac{K_a}{ir_{out} - K_b} \tag{3}$$

In the Figure the output of 3 different sensors has been measured and it can be seen that there is



a small difference between the individual sensors. The constants K_a, K_b will vary from sensor to sensor and has to be calibrated to make the distance measurements reliable. The constants in equation (3) can be fitted using MATLAB based on experimental measurements or a interpolation table can be constructed using linear interpolation.

Estimate K_a and K_b using *lsqcurvefit* in MATLAB and the data from above taking the following steps:

A function irdist that implements equation (2) must be written. The function should take a vector containing K_a and K_b as first parameter and a vector with distances in meter as second parameter. It should output the IR-values corresponding with the distances in the input. The function is written in a text file called *irdist.m* and the first line of the function is:

```
function irout=irdist(k,d)
```

k(1) represents K_a and k(2) represents K_b.

Test the function with a number of distances.

When irdist works optimal parameters, the $[K_a, K_b]$ may be found using *lsqcurvefit*. *Lsqcurvefit* needs the function name *irdist*, a start guess for $[K_a, K_b]$ (use [16, 10]), a vector containing measured ir-outputs and a vector containing corresponding distances(in meters).

Make plots that show the estimated function with full line and measured values with crosses. One plot for each frontsensor.