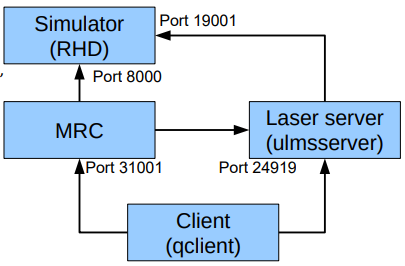
**Ex5:**

Exercise has focus on laser scanner server (ulmserver) with and without auzoneobst plugin, which is used for managing the SMR laser scanner. auzoneobst plugin is used for object detection and it limits the scanfield to 180 degrees so the SMR doesn’t detect itself.

Information about laser-scanner:

The laser scanner has an angular resolution of 0.36 degrees and a scanfield of 180 degrees separated in 9 zones with 20 degrees each. Furthermore the laser scanner has a range of 4 meters

Communication between servers:



Laser scanner simulator:

Another simulator is introduced in this exercise, which shows the laser simulator ranges of the SMR. Typing the zoneobst command will return the distances of the 9 sensors. SMR-CL commands can be entered in the client, which will trigger movements in both the original SMR simulator and the laser scanner simulator.

Furthermore various commands can change the display in the laser simulator and save images of simulations.

The laser scanner simulator can be started up with the other SMR simulator by using the following commands:

start simulator: simserver1 (in offline/sim directory)

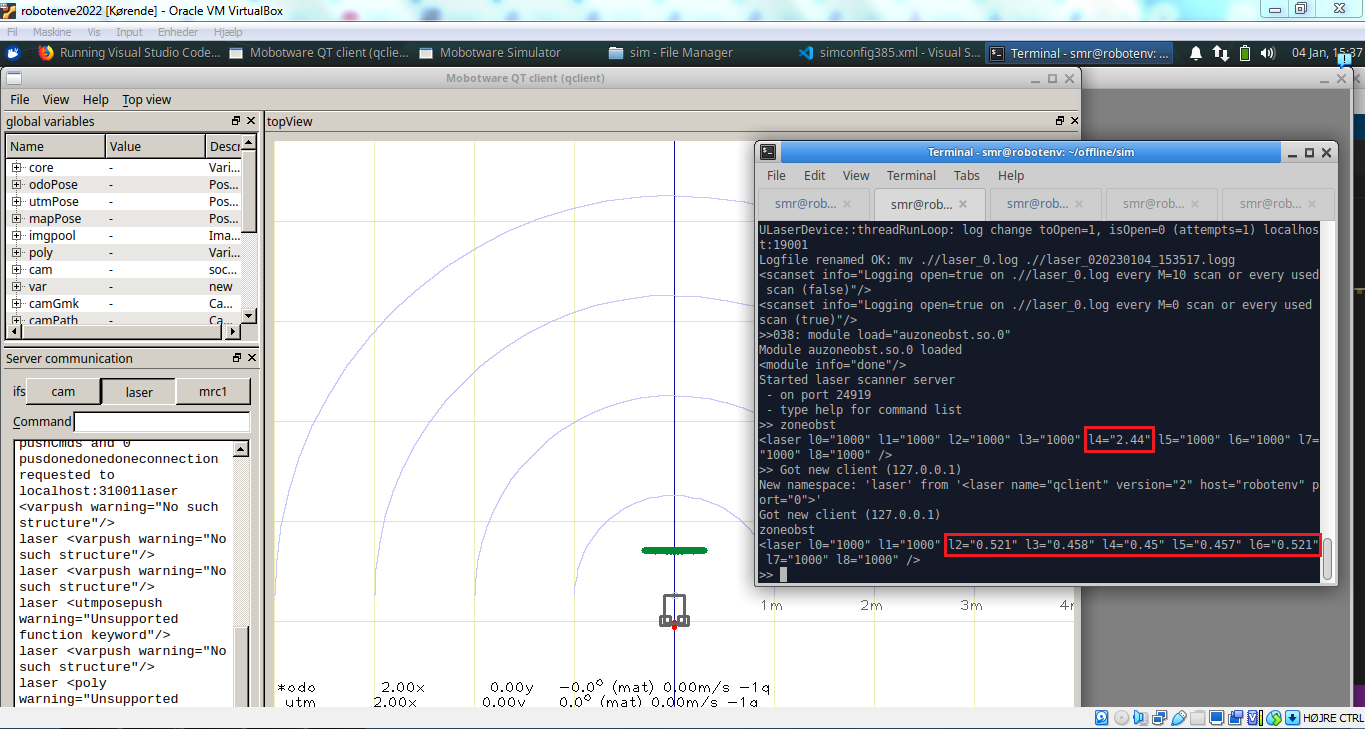
start laser scanner server: ulmsserver (in offline/sim directory)

start MRC: mrc -s8000 -t1 (in offline/sim directory)

start monitoring client: qclient -style a (in offline/sim directory)

Wall positions and floor lines can also be configured for simulation in simconfig385.xml.

Command 1: SMR looking towards a wall with a distance of 45 cm from the laser to the wall:



This was done by using command: mrc1 fwd 2

It can be seen above that only sensor l4 (in the middle) registers an object, which is the distance 2.44m. Thus the command drives 2 meters forward and more of the sensors (l2,l3,l4,l5,l6) detect the object. l4 is changed to 0.45 m from the object after driving 2 meters forward as expected.

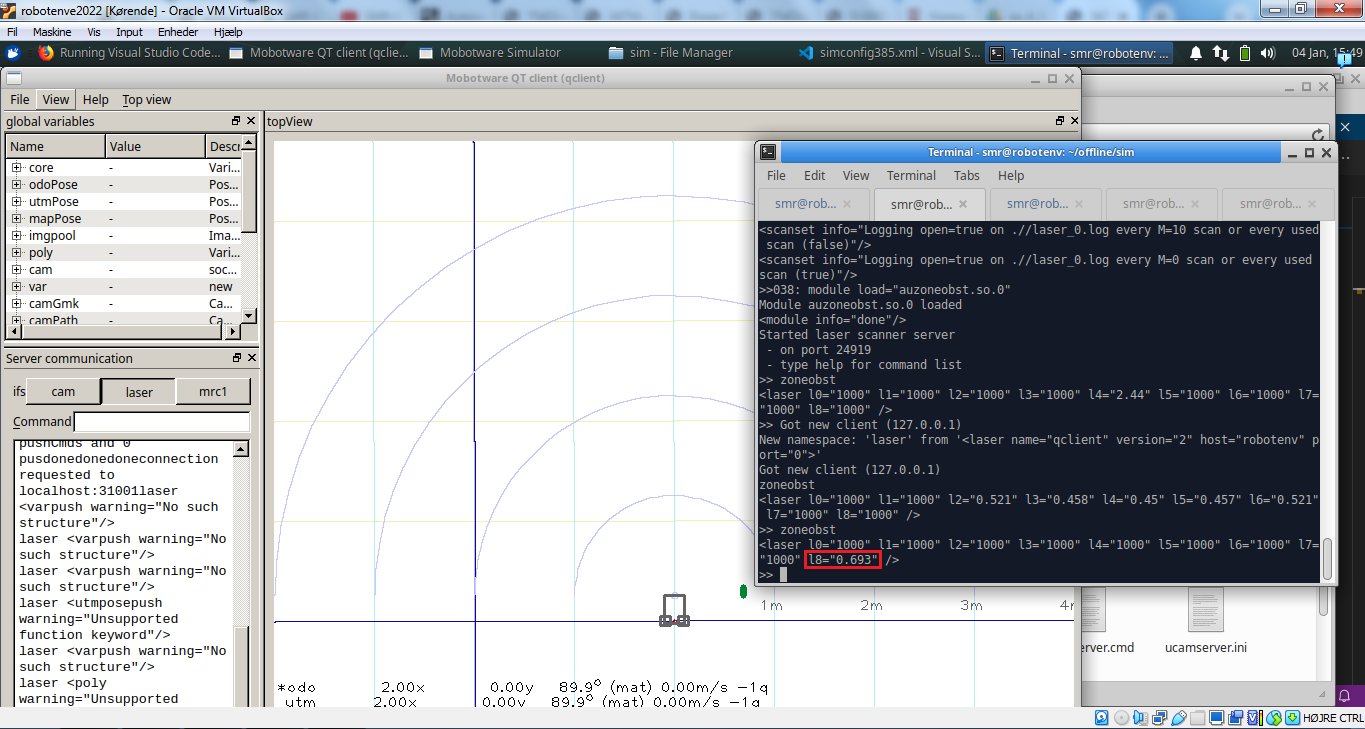
Command 2 and 3: SMR looking parallel to the wall with the wall to the left and a distance of 45 cm from the laser to the wall:

After driving towards the wall with the distance 45 cm, the SMR was rotated 90 degrees in each direction by using the commands:

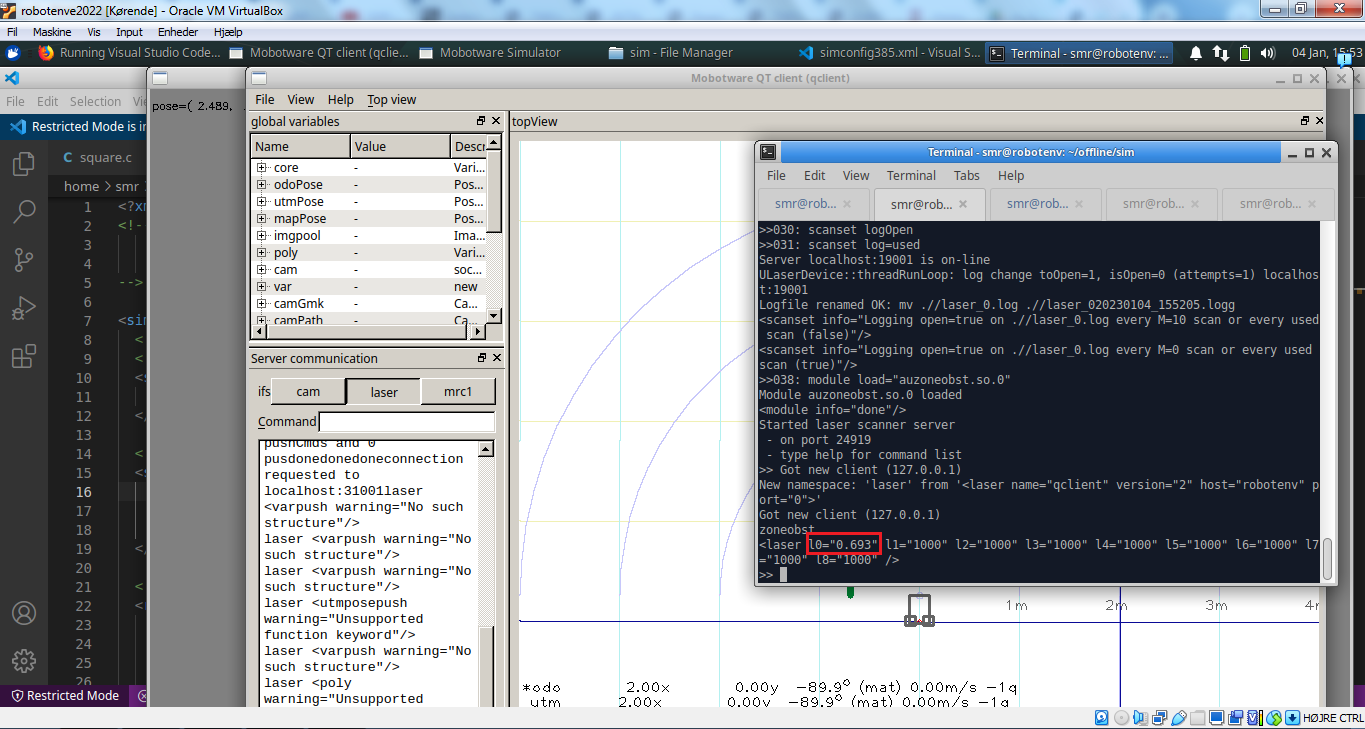
mrc1 turn 90 (turn left)

mrc1 turn -90 (turn right)

The results of the rotations can be seen below:



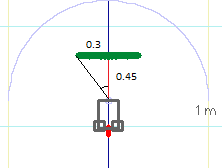
When the SMR turned 90 degrees to the left, only the closest sensor l8 could register the obstacle.



When the SMR turned 90 degrees to the right, only the closest sensor l0 could register the obstacle.

Theoretical calculations:

Knowing the distance of 45 cm between the SMR and obstacle, the height of the obstacle is 60 cm and that the SMR is perpendicular on the obstacle, then a right triangle can be constructed with the lengths 30 cm and 45 cm. Trigonometry can be used to find the maximum sensor angle as shown below:

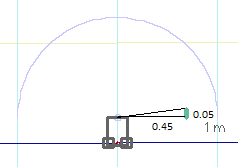


Theta = tan^(-1)(0.3/0.45) \* (180/pi) = 33.69 degrees

Since there is an unequal number of sensors (9), the middle sensor l4 must have a detection range of -10 degrees to 10 degrees, and the adjacent sensors must have detection ranges of 10 to 30 degrees and 30 to 50 degrees in each direction, which all include the 33.69 degrees maximum. That makes a total of 5 sensors in the middle, which would be l2, l3, l4, l5, and l6, which is the same as the results of the simulation.

Knowing that the SMR spins around its own center between the back wheels, and that the sensors are in the front of the SMR, the sensors are moved 25 cm in the y-direction when the SMR is turned 90 degrees in left or right direction. Subtracting the movement from the 30 cm a new right angle can be constructed and a maximum angle can be calculated as before.

Theta = tan^(-1)(0.05/0.45) \* (180/pi) = 6.34 degrees



Subtracting this angle from 90 degrees due to having 0 degrees in the middle:   
90-6.34 = 83.66 degrees

Since the sensors most to the sides have ranges of 70-90 degrees and 50-70 degrees, only the sensors furthest to the sides contain this angle, thus only these sensors should be able to detect the obstacle. This also fits the results of the simulations, since only one sensor detects the obstacle when the SMR is parallel to the obstacle.

Logging of laser-sensor values with MRC:

**code:**

laser "scanpush cmd='zoneobst' " % run zoneobst plugin command for every scan

log "$l0" "$l1" "$l2" "$l3" "$l4" "$l5" "$l6" "$l7" "$l8" % logging laser-sensor values to log-file

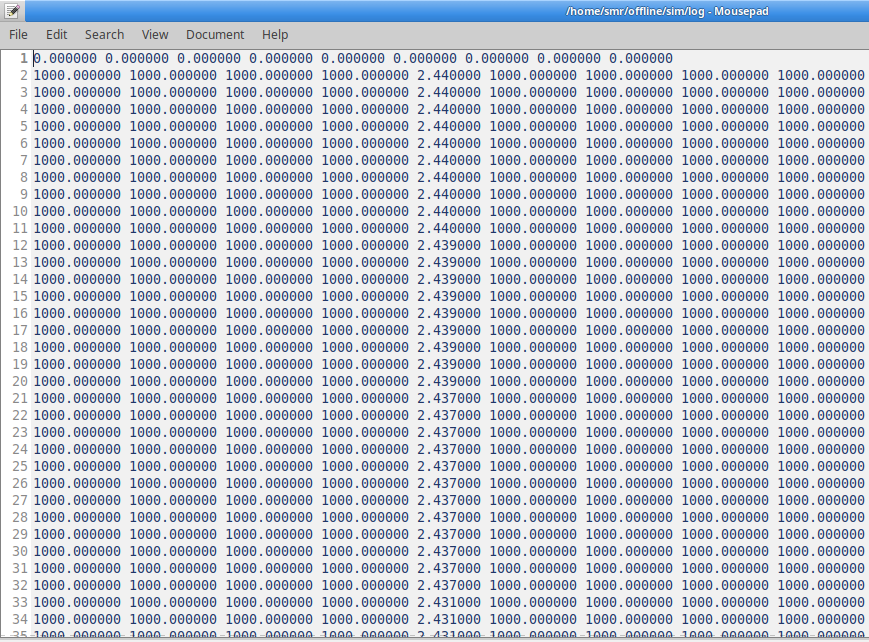
fwd 2 % move SMR forward

wait 1 % wait 1 second

turn 90 % turn SMR 90 degrees (left)

The data for previous commands have been logged by moving the SMR 2 meter forward followed by turning it 90 degrees to the left.

The code at the beginning executes the zoneobs plugin command for every scan first, which makes the laser-sensor values for logging. Afterwards the laser-sensor values $l0-$l8 are logged in a log-file in offline/sim/log. A wait command has been added between the fwd and turn commands to better distinguish the transition between the commands in the log-file. The content of the log-file can be seen below, where l4 starts at 2.44 as before slowly decreases as the SMR moves closer to the obstacle.



Logging of laser-sensor values with square programme (in C):

???

skal laves i C ... men min kode virker ikke lige nu