Laboratory Work #7: Implementation and test, in a real omnidirectional robot, of an Extended Kalman Filter to estimate the robot pose, and some basic trajectory control

Step 1 – Install VNC Viewer client:

https://www.realvnc.com/en/connect/download/viewer/

on your computer.

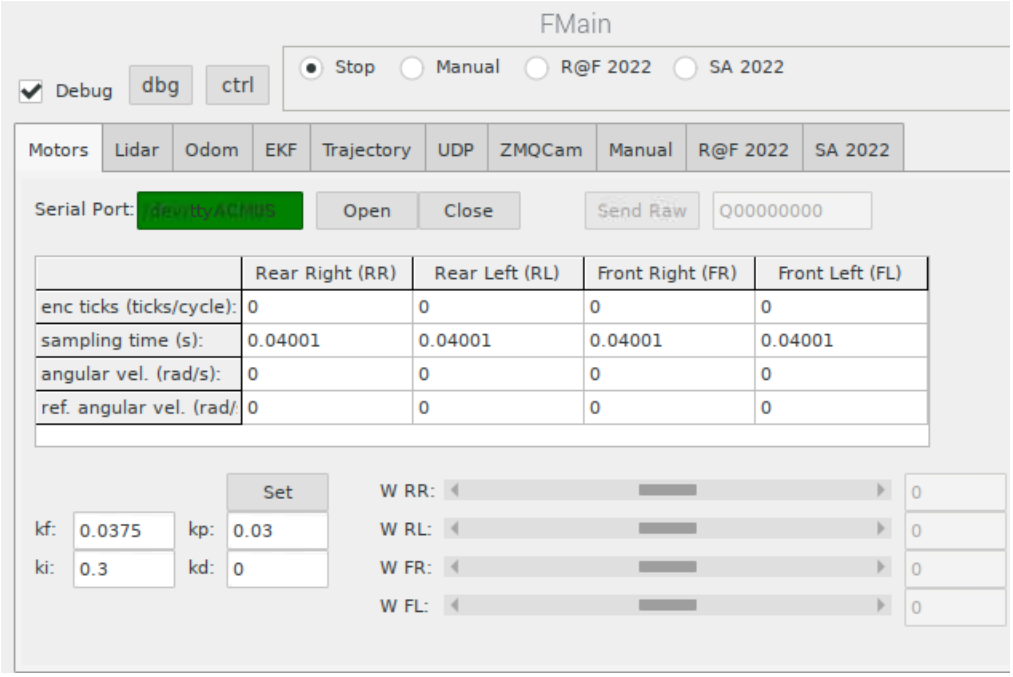
Step 2 – Connect to the wireless lan with ssid TP-Link_28CD (password = 49871005)

Step 3 – Goto “Control Panel -> Network and Internet -> Network and Sharing Center –> Wi-Fi (TP-Link_28CD -> Properties -> Internet Protocol Version 4 (TCP/IPv4) - > Proprieties” and change to “Use the following IP Address = 192.168.3.100” and “Subnet mask = 255.255.255.0”

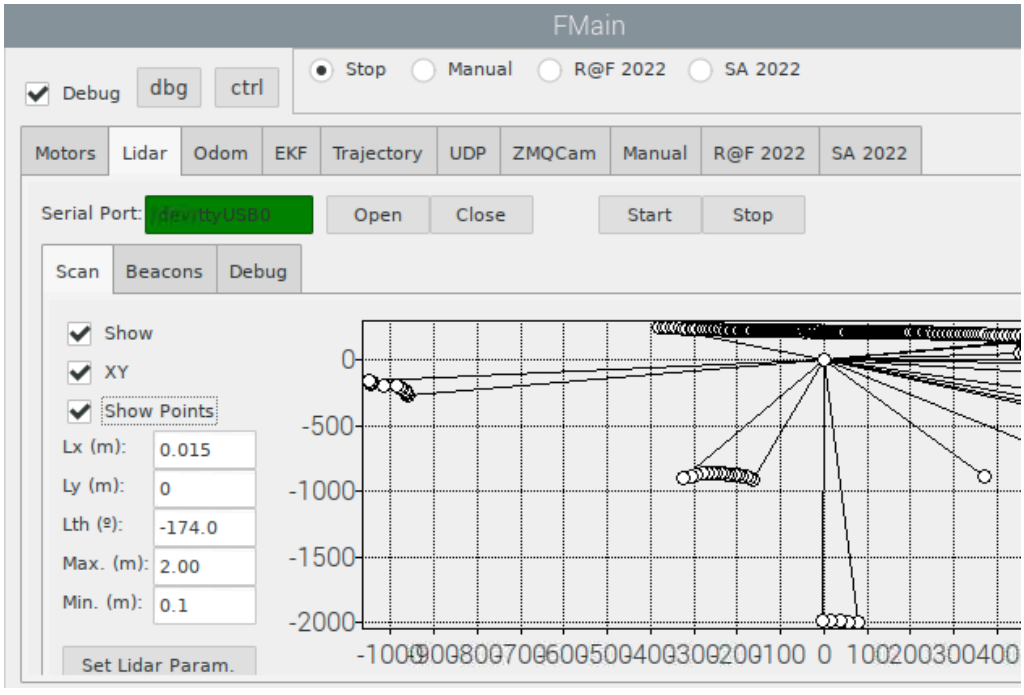
Step 4 – Start VNC Viewer and enter ip address 192.168.3.XXX, where XXX is the robot number. If the system ask for a user name and password use, user: pi and pass: 5dpo

Test the Extended Kalman Filter to estimate the robot pose:

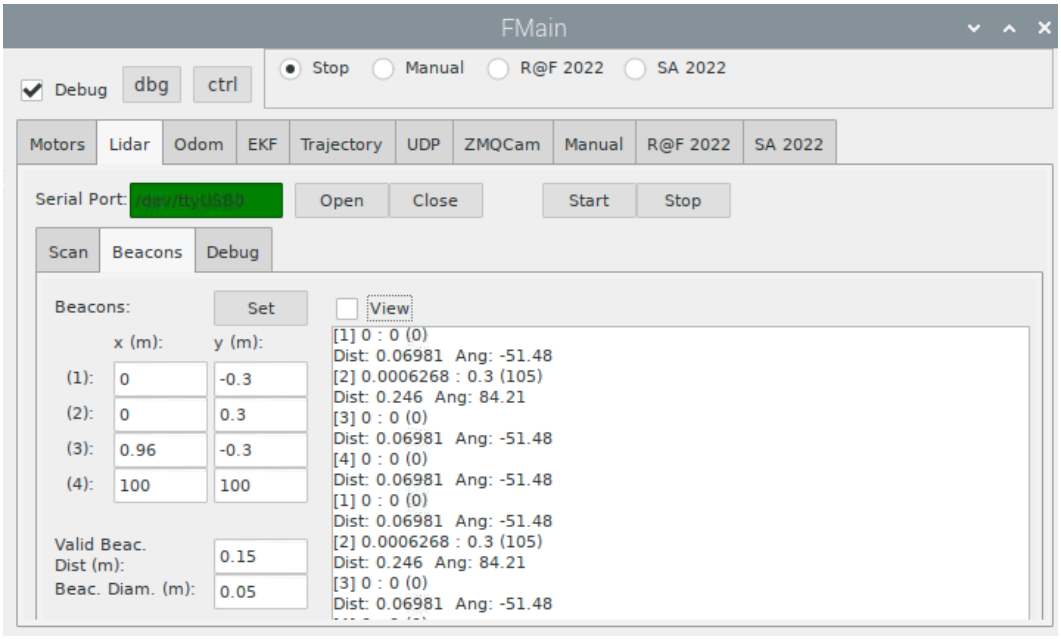
Step 5 – Start the script “Robot Omni” present at the Desktop. Go to tab “Motors” and Open the serial port. The communication between the high-level controller (at Raspberry Pi) and the low lever control system (at Arduino) will start.



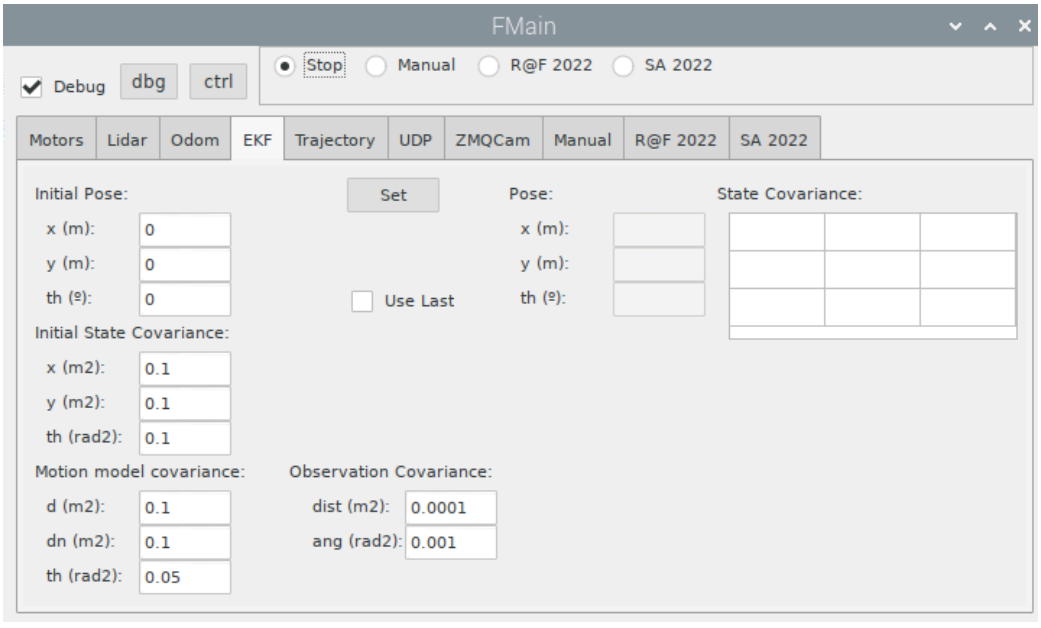
Step 6 – Go to tab “Lidar” and Open the serial port. The communication between the Raspberry and the Lidar will start.



Step 7 – Go to sub-tab “Beacons” at tab “Lidar”. Chose a position for your beacons and insert it. If you use only 3 beacons insert un impossible value in beacon number (4). Check “View” to observe the detected beacons. If a deacon is detected, you should have a non-zero value between brackets.



Check at tab “EKF” that you “Initial Pose” is correct and that you estimated “Pose” is near the true pose.



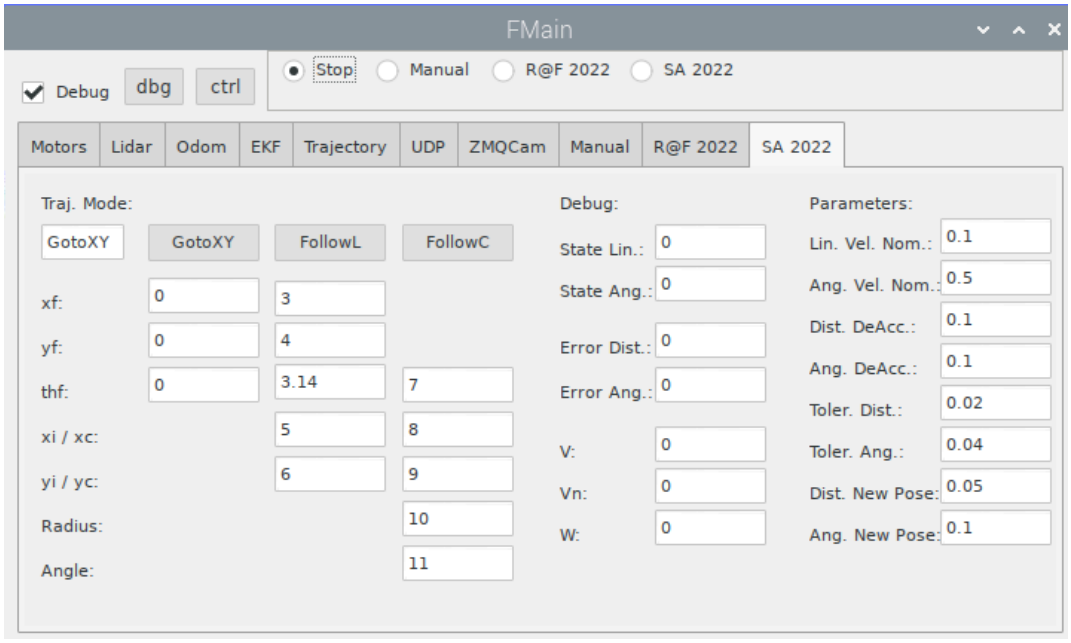
Basic trajectory control:

Step 8 – Start “Lazarus_fppcup deluxe” double clinking in the icon at the desktop.

If you click in the code editor, you can find any text using keys “Ctrl + Shift + f”. Use it to find procedure “SAUT_GotoXY”, “Control”, “ControlSA”, etc. The main procedure that controls de robot is “TFMain.Control” at file “main.pas”. As illustrated, it is triggered by the serial odometry data sent by the Arduino at a fixed rate of 25 frames per second (control cycle equal to 40 ms). The Laser data is asynchronous and with a different cycle period (near 125 ms), so we don’t have observations and the update phase of the EKF in every control cycle. The connection between the laser loop and the control loop is made using the flag “do_update”.

Step 9 – Use menu option “Run ->Run” (F9)

Step 10 – Go to tab “SA 2022”. Observe and adjust the different parameters, then select the “SA 2022” on the top of the form.



Step 11 – Close the program and go to the code editor. You can switch between editing the code or the form pressing F12. Adapting as example the procedure “SAUT_GotoXY”, create new procedures “SAUT_FollowLine” and “SAUT_FollowCircle” to implement the similar procedures you developed in the simulator. You can copy the simulator file with the code (using a PEN), open it in the code editor and perform copy/paste to the Lazarus procedures with some adaptations namely in the name of the variables, number of parameters, etc, as you can see in SAUT_GotoXY.

To create and use a new procedure for trajectory control, you must declare it in the public section (starts at line 515), write it (can be next to SAUT_GotoXY), and then call it form the “procedure TFMain.ControlSA;”.