Adaptive Gesture-Based Control of a Mobile Robot for Different Control Styles

Francesco Grella, Fabio Torello

01 March 2018

Abstract

What follows is a description of the development of our assignment for the course 'Software Architectures for Robotics': our objective was to provide an improvement of the already implemented gesture-based controller of the mobile robot 'Husqvarna Automower'. In particular we have provided a part of the control architecture with the purpose to filter the incoming data from the sensors of the Android-based smartwatch, with the goal of improve and adjust the response of the robot itself in spite of different control behaviors shown by its drivers.

1 Introduction

Our project was entirely focused on the usage of the mobile robot 'Husqvarna Automower' available in the EmaroLab and of the Gesture-Based control architecture already developed, which features a Smartwatch and an Android Smartphone used for the transmission of data from the wrist of the user to the robot, through a laptop PC connected to the Husqvarna via USB cable. The project development can be divided into two parts: -The first one was related to the analysis of several datasets, collected during a testing session with some volunteers, by the sensors of the smartwatch and the Motion Capture system of the laboratory, voted to a better comprehension of the problem and of the different control styles of the participants. -The second one consisted in the implementation of a control scheme in Matlab/Simulink whose particularity is the usage of a Fuzzy Logic filter built up by the results of the data analysis introducted above. The filter is used to adjust the command values regarding the Linear Velocity of the robot, given from the user by raising/lowering the arm.

Figure 1: An Husqvarna Automower similar to the one we have used for the project.

Before the description of the development of the project it's our duty to inform the reader that the assignment was initially intended to be carried on by using the Java programming language, but when we were given the possibility to use Matlab we have chosen it because of its Fuzzy Logic Toolbox, which provides a GUI for implementing Fuzzy Systems, and for the presence of Simulink, which seemed to us the best way to implement a block-scheme for control purposes.

2 Development of the Project

2.1 Data Analysis Phase

The datasets we had to deal with were recorded during a testing session performed in the lab, which involved the usage of an already developed 'gesture-based' control architecture: an Android Smartwatch uses an application to record the acceleration values of the wrist, then sends it to a smartphone which has a complementary app for reception and re-transmission of data to a PC with ROS(Robot Operating System). The task of the volunteers chosen for the test was to perform a simple close circuit-like track by following its trajectory with the smartwatch-controlled robot. The data registered from the Smartwatch and from the Motion Capture of the room were stored into appropriate data structures: 'dataCommand' for the time values and accelerations of the wrist and 'dataMoCap' for the data recorded by the eight cameras which kept track of position and orientation of the robot along time.

We initially worked on a Matlab script with the purpose of a clear and intuitive data visualization for a better extraction of informations: in particular this allowed us to graphically represent 'dataMocap' and show all the trajectories followed by the robot during the tests. Starting from these representations, first of all we have selected among all the datasets the nine which were associated to the most regular trajectories; then for each track we have isolated one straight and one curve. By manually performing a mapping between the two data structures, in particular we have extracted from 'dataMoCap' the time values in which were performed the straight and we have done the same for the curve, for each of the nine subjects. Then by finding the corresponding time lapses in the 'dataCommand' structure we could get to the goal of the analysis: a representation in time of the accelerations of the wrist associated with a straight or a curve. From the plots of these data, appropriately treated with a median filter to regularize them for a better visual comprehension, we could perform a qualitative analysis of the driving style of each subject, and we have been able to provide a simple categorization in two groups:

- The 'Aggressive' drivers, which have demonstrated an impulsive command behavior, with sudden movements of the wrist and so characterized by a slight difficulty to follow the proposed path, with a tendency to the trajectory adjustment.
- The 'Reliable' drivers, which are characterized by a smooth control style, with the nearly absence of sudden transitions or rough changes of velocity or orientation. They have demonstrated a certain regulation in velocity reaching uniform and low values and so they have followed more correctly the trajectory of the track.

Before proceeding with the description of the work is useful to precise that, like introduced above, the resultant data from the extraction were needed to be treated with an appropriate filter because of the presence of many 'disturbances' registered from gyroscopes and accelerometers of the smartwatch. Anyway after the filtering operations they were good for an appropriate graphical representation. Now is shown the analysis of the behaviors with respect to the graphs which represent the filtered data, along with the 'dataMoCap' graph for each subject, keeping into account that we have chosen to extract straights and curves from the left part of the plots due to their regularity.

2.2 Behavioral Data Analysis

In this section we provide some examples, taken from a set of 9 analyses, of the categorization of driving styles, and for each one we specify the interval of significant values and some deductive comments on the behavior of the subject. The plots have the acceleration values on the y-axis and the time variable on the x-axis.

- Lap 1, Group 1, Volunteer 1
  -Angular Acceleration in straight, defined into [-0-05, -0.45] rad/sec^2: There are two high spikes, but can be possible adjustments of the trajectory.
  -Linear Acceleration in straight defined into [0.14, 0.22] m/sec^2: Mild variations, overall smooth control.
  -Angular Acceleration in curve, defined into [0.3, -0.4] rad/sec^2: A sudden descent, probably due to trajectory correction.
  -Linear Acceleration in curve defined into [0.12, 0.22] m/sec^2: Not smooth overall, but the range of variation is very small.

Figure 2: Trajectory performed during the test.

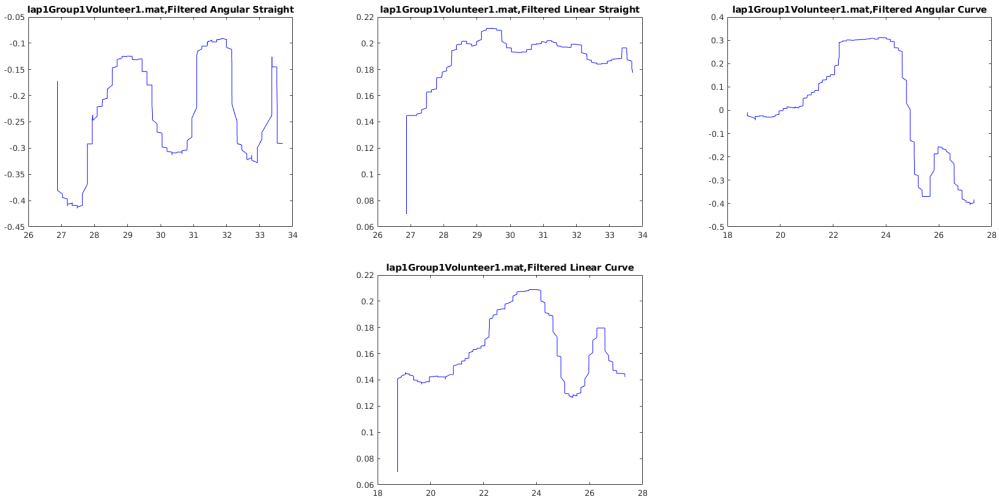

Figure 3: Plots of wrist accelerations during straight and curve.

- Lap 1, Group 3, Volunteer 11
  -Angular Acceleration in straight, defined into [-0.45 , -0.2] rad/sec^2: It can be seen that wrist is not very steady, due to some peaks. Anyway it holds its overall trajectory.
  -Linear Acceleration in straight defined into [0.08 , 0.24] m/sec^2: Even with some irregularity, the range of variation is quite small.
  -Angular Acceleration in curve, defined into [-0.5 , 0.3] rad/sec^2: One continuous peak and then a sudden decrease, it could mean an adjustment or the end of the curve.
  -Linear Acceleration in curve defined into [0.1 , 0.4] m/sec^2: Small range but irregular profile, could be sign of a slightly aggressive control style.

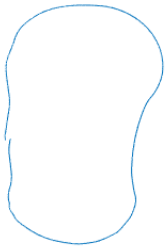

Figure 4: Trajectory performed during the test.

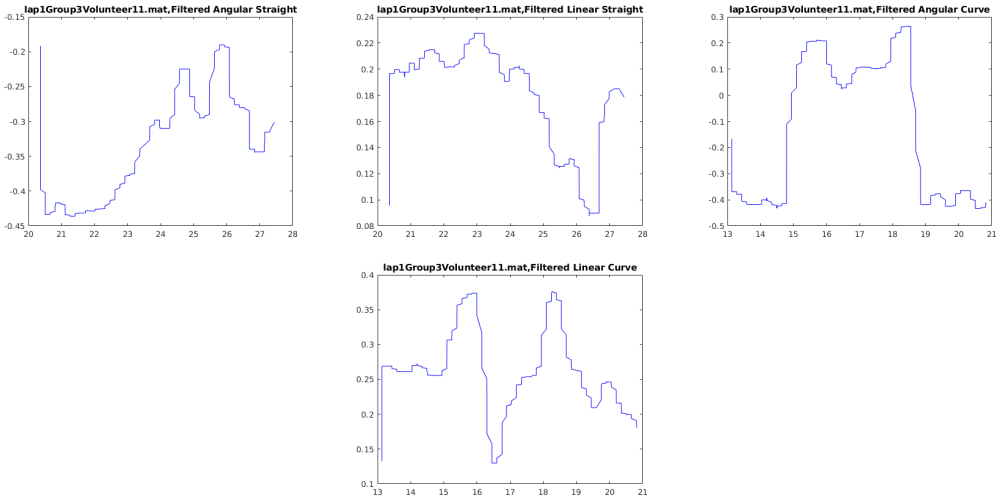

Figure 5: Plots of wrist accelerations during straight and curve.

- Lap 1, Group 3, Volunteer 27
  -Angular Acceleration in straight, defined into [-0.4 , 0.3] rad/sec^2: No peaks but two descents, progressive decreasing of angular in straight line, maybe due to approximations or to the approaching of a curve.
  -Linear Acceleration in straight defined into [0.04 , 0.2] m/sec^2: Very small range, on straight can be a sign of a very calm driving style.
  -Angular Acceleration in curve, defined into [-0.4 , 0.4] rad/sec^2: Wide range but no peaks, very smooth progression in time, calm driving even during the curve.
  -Linear Acceleration in curve defined into [0.06 , 0.2] m/sec^2: Small range and smooth graph, so in curve keeps a good control of the linear speed.

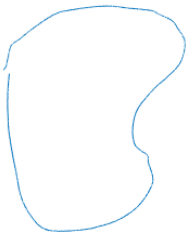

Figure 6: Trajectory performed during the test.

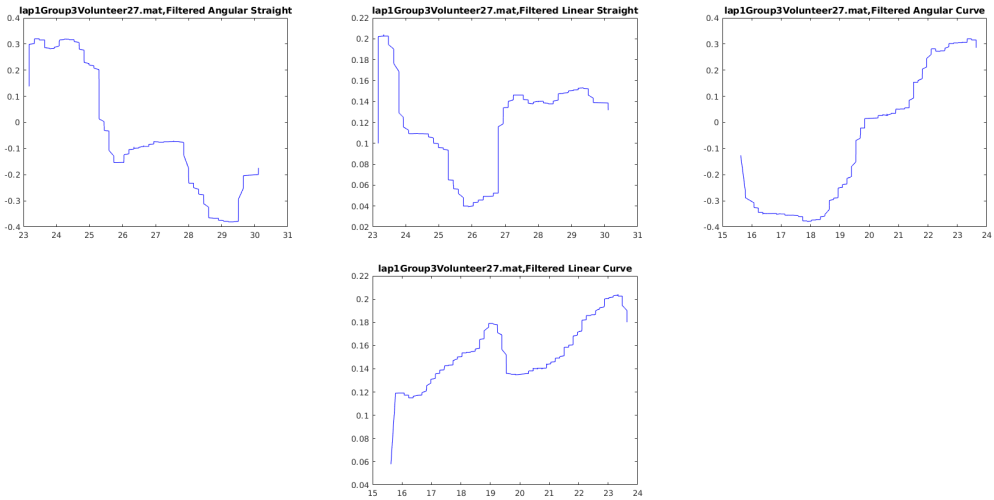

Figure 7: Plots of wrist accelerations during straight and curve.

- Lap 2, Group 3, Volunteer 7
  -Angular Acceleration in straight, defined into [-0.5 , 0.1] rad/sec^2: Sudden jump in less than one second, so it's a bit aggressive adjustment of trajectory during straight .
  -Linear Acceleration in straight defined into [0.1 , 0.35] m/sec^2: Not so uniform, even here the subject shows a bit aggressive style of driving, with sudden variations.
  -Angular Acceleration in curve, defined into [-0.5 , 0.25] rad/sec^2: A sudden fall from 0.1 to -0.4 which can mean a bit aggressive control.
  -Linear Acceleration in curve defined into [0.2 , 0.35] m/sec^2: Linear velocity in the interval evaluated doesn't change too much, even if it's a bit irregular.

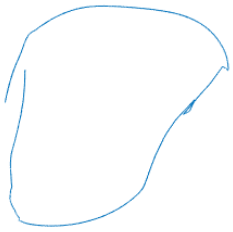

Figure 8: Trajectory performed during the test.

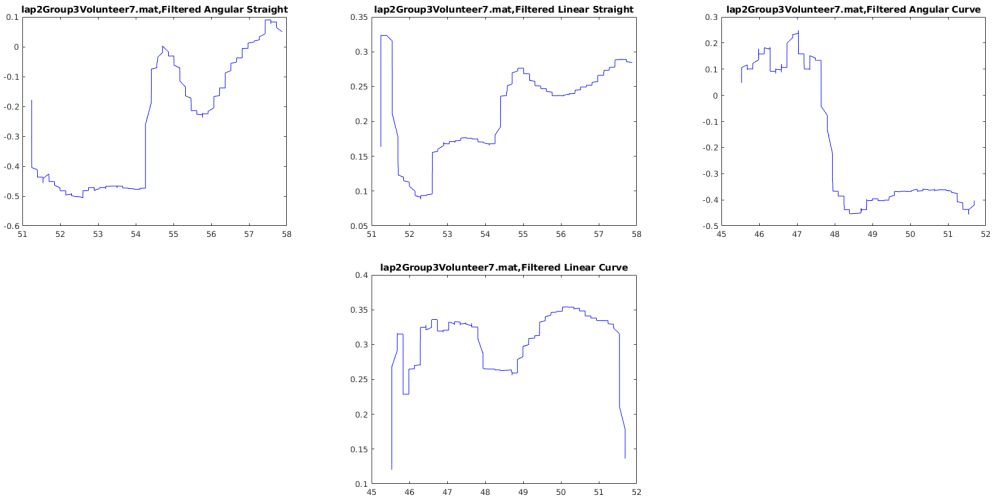

Figure 9: Plots of wrist accelerations during straight and curve.

2.3 Fuzzy Logic Filtering

From the data analysis described along the previous pages we have extracted some significant intervals for the values of linear and angular accelerations, and we have tried to implement two Fuzzy Logic Systems based on Gaussian membership functions,which were intended to be used for filtering both of the accelerations sent from the smartwatch.

Unfortunately we realized that the overall values regarding angular acceleration were too inhomogeneous to be a solid basis for a statistical gaussian representation, so we focused our work on a fuzzy filter to be applied only to linear acceleration values, in order to smooth and regularize the linear velocity of the robot even in case of rough arm movements.

As input we have chosen three gaussian membership functions: one based on the behavior along straight path and two specular relative to left and right curving. The output membership functions, instead, aren't based on collected data but we modeled them to be as much regular as possible for a correct velocity control. In the next page are shown the components of the implemented fuzzy system.

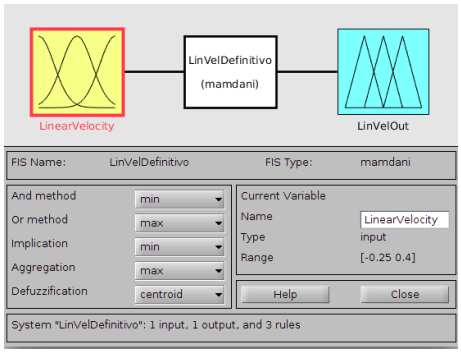

Figure 10: Overall window of the fuzzy logic SISO system.

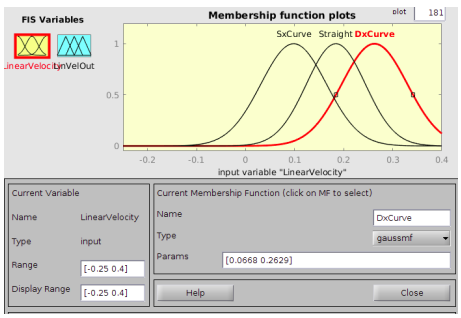

Figure 11: Input gaussian membership functions.

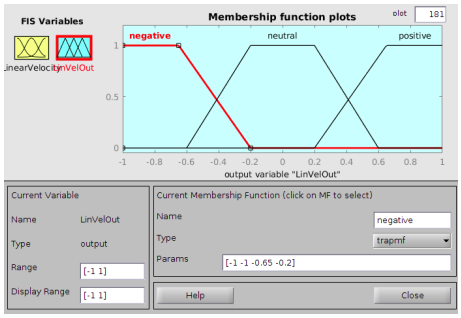

Figure 12: Output trapezoid-shaped membership functions.

3 Simulink implementation of the Control Architecture

Our work in Simulink environment can be described along two sections: A first section was mostly voted to the definition of the overall scheme and the proper testing of the block diagram in different situations and with different types of input data.
During this phase we have defined block control schemes based on the Publish/Subscribe architecture commonly used for ROS communication: for the initial tests we built a Publisher-only scheme and we used it to import arrays of values as time signals from the Matlab workspace, then the Publisher block's task was to send these signals directly so the robot via the '/cmd_vel' topic.
Initially we implemented a script for simple constant signal generation, so the Simulink model was expected to transmit commands to make the robot move along a straight trajectory or along a constant curve.
Then we implemented a second script which was, like the first one, voted to the creaction of command signals, but the arrays were built by the data structures considered for the analysis phase; in particular we have tried to provide a sort of emulation of the gesture-based control by using the 'dataCommand' values of linear and angular acceleration.
The task of the control scheme this time was to receive the signals from the workspace and then, like if they were sent from the sensors, to publish them on the topic '/inertial'; in this way we have provided a merging between our control architecture and the one already implemented by Emaro-Lab staff, because the previously mentioned topic was used to pass the values to the python script 'gb-controller.py' which was intended to operate appropriate computations for data conversion and then publish the output values on '/cmd_vel' topic by itself.
By this way, making proper adjustments to gain values directly into the scheme, we have managed to partially recreate the tests by sending the collected data to the Husqvarna; we could also have a first behavioral test of the Fuzzy Logic system, which gave us the expected result of 'fixing' the linear velocity of the robot to a very limited range of values even if the input data were rough and discontinuous.

The final step of our testing activity led us to build a more complex control scheme, which we have used both for data collection from the smartwatch and for transmission of velocity commands to the robot.
Like it can be seen in Figure 7, this block scheme is intended to provide the functionalities of Subscriber and Publisher in order to make a bridge between the two fundamental ROS topics of the architecture: the first part of the scheme implements a Subscriber which receives messages from '/inertial' topic, then the data are sent through some gain blocks which are used to recreate the value conversion performed by the python script and finally, after the filtering through the Fuzzy Logic of the linear velocity commands, the second part Publishes the value on the '/cmd_vel' topic.

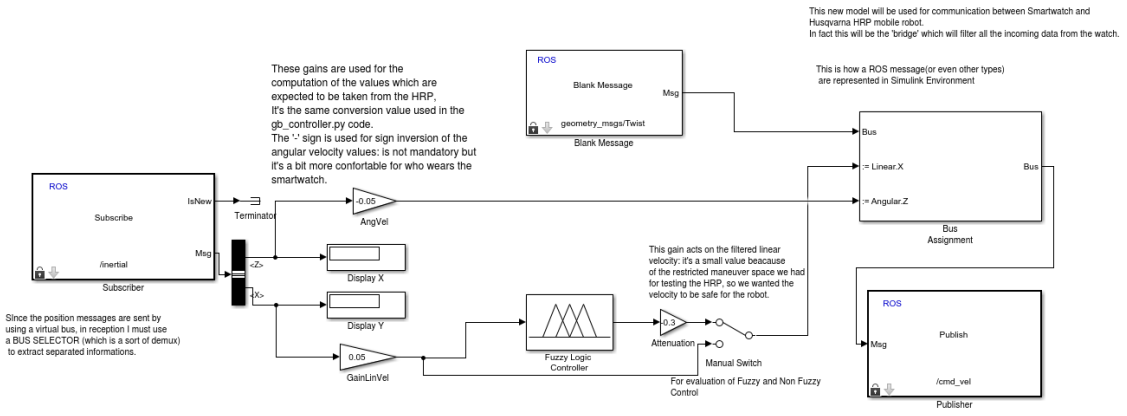

Figure 13: Simulink Block Scheme used as bridge between the smartwatch and the robot.

4 Conclusions and Future Work

In conclusion, it can be said that for which concerns the Fuzzy Logic control our goal has been reached regarding the linear velocity of the robot, which results almost fixed and opportunely scaled even in response to abrupt variations given from the wrist.
It can be also done a final consideration regarding the angular velocity values, which could be helpful for future developments: there is a lack of precision probably due to the sensors, in particular we have noticed during the tests that is difficult to keep a good orientation of the robot without twisting the wrist, and sometimes the curving commands were too unbalanced towards left or right.
We think that these problems could be solved with the implementation of another Fuzzy Logic filter for the angular velocity, maybe statistically based on more accurate and homogeneous data, or maybe even with an approach different from the gaussian distribution that we have used for our filter.