Design of Embedded and Intelligent Systems, Lab 3

Group 1, Simon Brunauer, Harald Lilja, Anton Olsson

October 2019

1 Introduction

This is the report for the third lab in the course Design of Embedded and Intelligent Systems, conducted in autumn 2019 at Halmstad university. The lab consists of 5 parts that have to be completed in order to pass the assignment. To complete the tasks a data-set was given, which has to be used for the calculations afterwards. Those calculations were performed with Matlab.

2 Exercise 1

The first exercise consists of plotting the camera data and the data from the magic sensor against the reference data. The following questions is answered:

- What can you say about the accuracy of the information provided by each sensor?
- Can you describe how this accuracy changes according to the position of the robot?

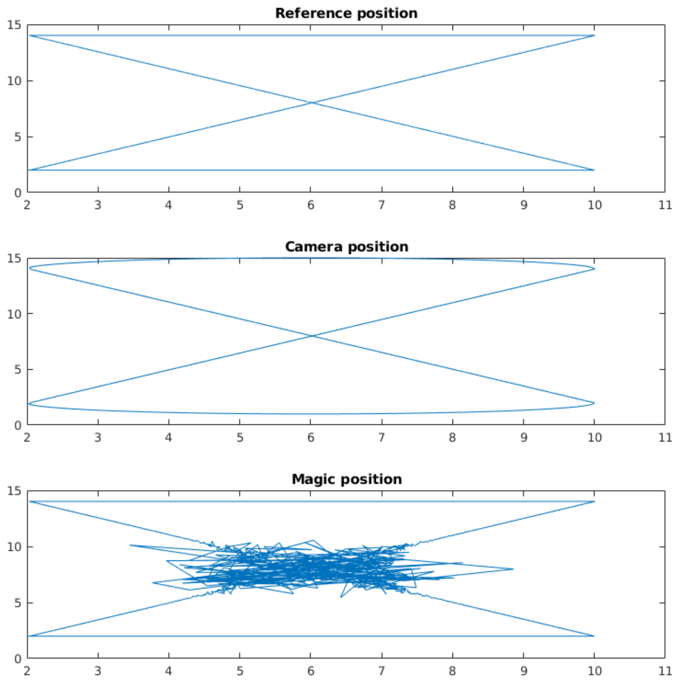

Figure 1: The raw data from each sensor and the reference position

To answer the first question about the accuracy for each sensor the project members came to the conclusion that the magic sensor has good data far from the center. The edges and the lines is closest to the reference. As for the camera the data is good in the center but worse at the lines. This we think is due to the camera being mounted above the robot with a wide-angle lens giving the data this fish-eye characteristic. This ties in to the second question, so when the robot moves close to the walls the accuracy of the magic sensor is the best but when the robot moves closer to the center the camera sensor has the best accuracy and the magic sensor accuracy is worsened.

3 Exercise 2

For the second exercise we were supposed to combine the sensor data using weighted averaging. It should look like this:

$$Signal_{fused} = A * Signal_{camera} + B * Signal_{magic} \tag{1}$$

Where $A + B = 1$ and each scalar should correspond to how much one can trust each sensor. If a sensor is reliable then the scalar value should be closer to 1. Exercise 2 posed the following questions:

- How good is your fused estimate of the robot position?
- How does it compare to the estimates from the camera only or from the magic sensor only?

First of the project members started of by calculating the error measure for each sensor to the reference data. The residual in each of the plots correspond to the difference between each sensor to the reference data. The value for the scalars were chosen as 0.5 each as they seemed to give the best results after tests. To answer the first question of how good the fused sensor is the plot shows that it has the lowest residual(difference in position to reference). One can see from the plot that the noise in the middle is reduced for the fused sensor in comparison to the magic sensor. It also shows a small improvement in regards to the wide-angle properties of the camera sensor.

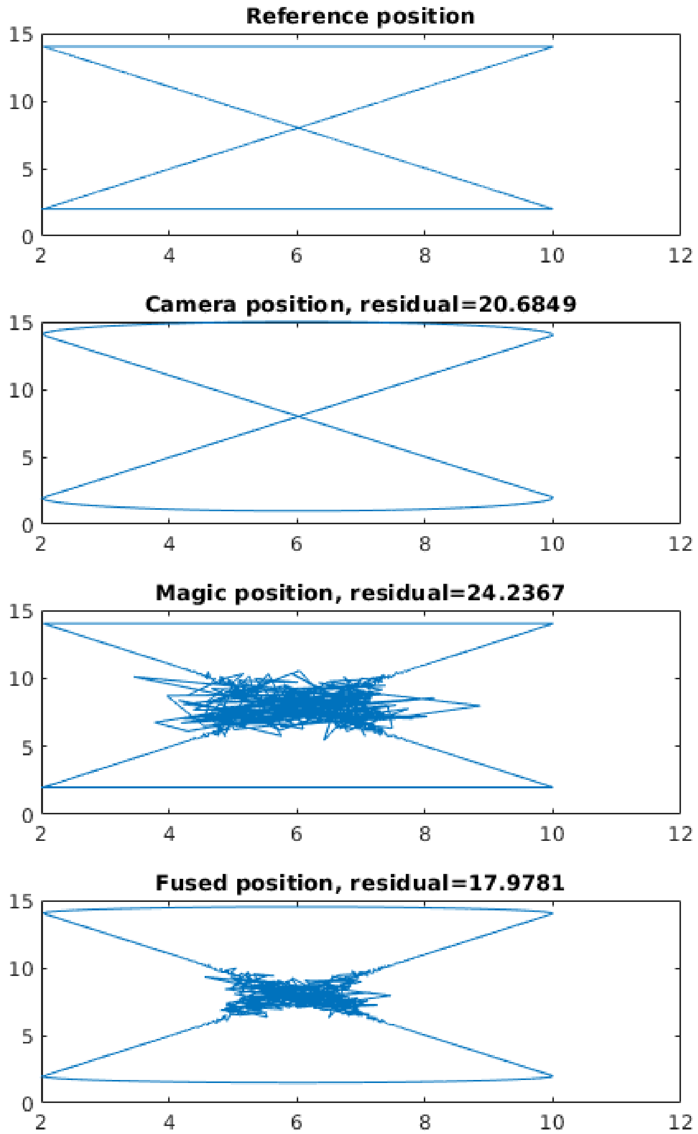

Figure 2: Data for each sensor and the fused sensor to the reference.

4 Exercise 3

In this exercise fuzzy sets should be created, which serve for the decision, which sensor should be trusted. Therefore, a crisp set with the functions f, g, p and q was defined. With the function g it was designed, that the magic position sensor should be trusted from 1 to 4 and from 8 to 10 in x direction, whereas the function f defined, that the camera should be trusted in between. The function q defined, that the magic position sensor should be trusted from 1 to 5 and 10 to 15 in y-direction, whereas p is the opposite again. The resulting plot of the fusion estimates can be seen in figure 3.

Afterwards a fuzzy set was created. The functions f1, g1, p1 and q1 represent those sets. The function f1 ranges from 1 to 2 and is then smoothing until 3 and is increasing from 8 to 9 again and ranges until 10. The function p1 also ranges from 1 to 2, but is then smoothing until 5 and increasing again from 10 to 12 and ends at 15. As before g1 and q1 are just the mirrored functions. The numbers for the function were decided, after looking onto the graph of the 2 sensors and evaluating the range, where those are trusted. The resulting plot can be seen in figure 3.

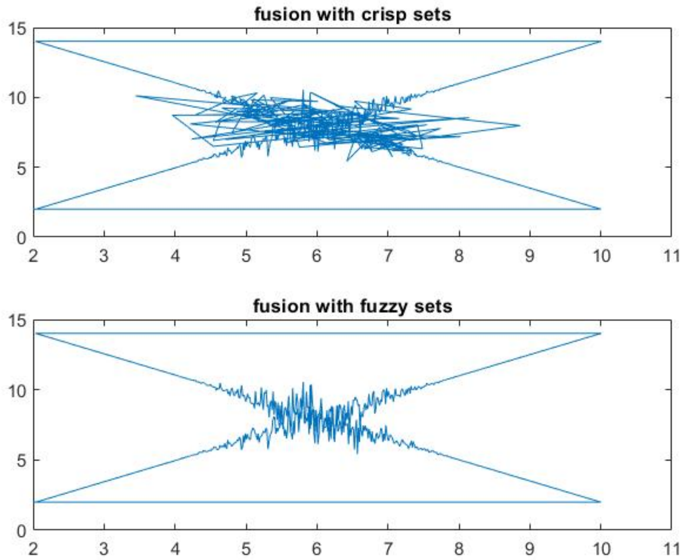

Figure 3: fused sensor signals with fuzzy logic

5 Exercise 4

In this exercise the position of the robot should be estimated, based on the data from the wheel-encoders. Those always show the current angle of each wheel. To calculate the position, the formulas, given at the assignment, were implemented in Matlab. It was important to consider, that the encoders only range from 0 to 2π. This means that there will be a jump once the wheels did one whole turn. This jump of the angle has to be subtracted then and the correct change of the angle can be taken into account again. The resulting estimated position were plotted against the reference position. This can be seen in figure 4. This figure also shows, that there is an error occurring in the estimated position. This can be explained, as the wheels dont always do the same way during a straight movement and therefore create a continuous error, which adds up the further the robot moves. A possible solution for this problem, would be to update the position from time to time with the position, which is given by a sensor.

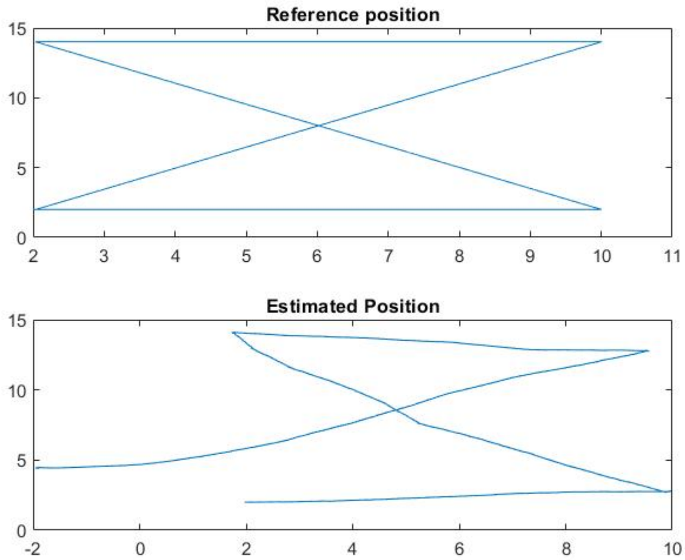

Figure 4: estimated robot position based on wheel encoder value

6 Exercise 5a

For this exercise it was said, that the fused sensor signal gives a better position at every 10^{th} sample. Therefore the position is updated at every 10^{th} sample with the corresponding sample from the fused sensor signal. This overcomes the problem of the the continuous error in the estimated signal. The result of this can be seen in figure 5. There it is compared to the reference signal.

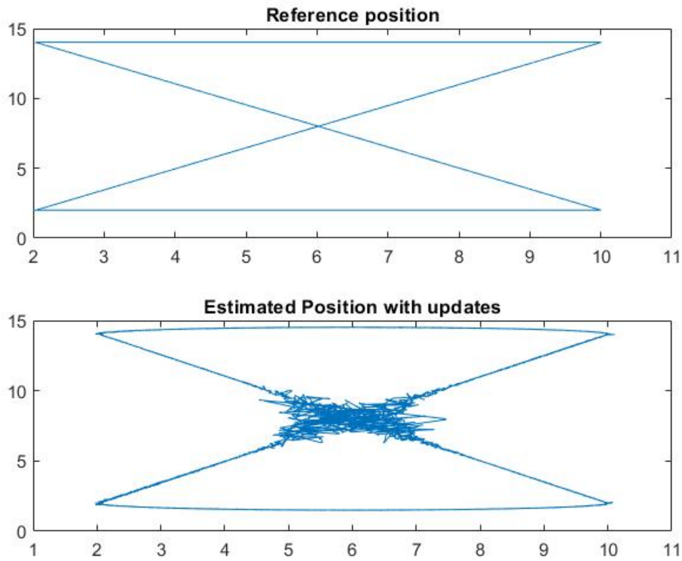

Figure 5: estimated robot position with updates from sensors

7 Exercise 5b

In this exercise it was assumed, that the positions from the fused signal are delayed by samples of the encoders. Therefore, the update has to be done at 4 samples prior to the current one. To ensure that this update is also taken for the newer calculations, the following 4 samples have to be updated too. The result can be seen in figure 6, where it is also compared to the reference position.

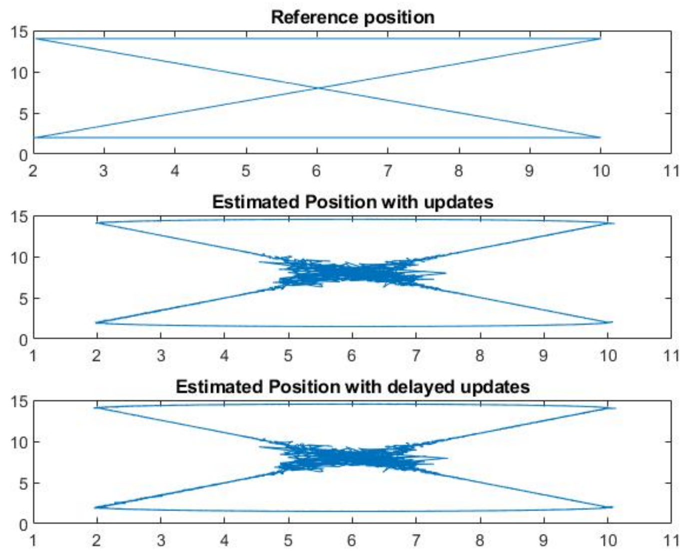

Figure 6: estimated robot position with delayed updates from sensors