# SENSOR FUSION LAB

## DEIS Course

**This material is designed by Anita Sant'Anna, modified by Hassan Nemati**

The laboratory session of this week is about sensor fusion. By the end of this lab assignment you will:

- Understand the relationship between reality and sensor readings
- Understand the strengths and weaknesses of different sources of information and how best to combine them in order to make accurate measurements and take good decisions

### Before you arrive at the lab:

You should read through the entire document and look up anything you don't quite remember from the lectures. If you have never used MATLAB before, you should familiarize yourself with MATLAB.

### After the lab:

Three weeks after this session you must hand in a **group (of three students)** report and your **source code**. Your report should explain your results, **how** you achieved them, and **what their relevance is**. Make sure you include any plots and figures relevant to your explanation. Your code, developed in order to solve the exercises proposed here, should be well commented and self-explanatory.

Send your report as a **pdf** document and your code as one or several m-files to **naveed.muhammad@hh.se**. The name of your files should include your last names (of each group member separated by underscores), for example *Name1_Name2_Name3_Report.pdf, Name1_Name2_Name3_Code.m*.

The title of your email should be "SENSOR FUSION LAB" and your Group number.

### Grading criteria

You will be judged by your ability to interpret; reason on; and solve the exercises and tasks presented here.

- Pass: Complete Exercises 1,2,3,4 and 5A (within the 3 weeks deadline i.e. 23:59 on October 15th, 2019).
- Fail

## THE DATA

This lab is built upon a synthetic dataset created for the purpose of this lab. The basic setup considers a robot similar to the PIEs used in the course. The robot moves along a predefined path, as shown in Figure 1. The grey area is the space where the robot can move, the red lines are walls which block the robot's passage.

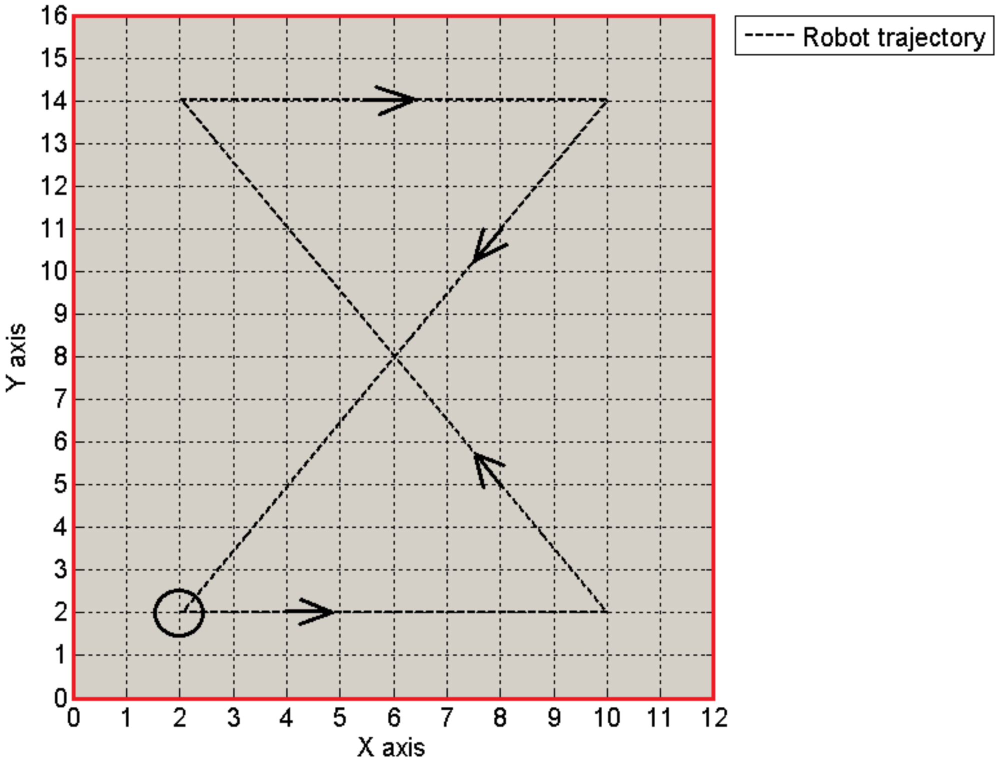


**Figure 1**. The robot space and the reference trajectory. The red lines indicate walls and the grey space is where the robot can move freely.

**Your task** is to estimate as accurately as possible the position of the robot using the acquired sensor signals. The sensors available to us are: incremental optical rotary encoders attached to each wheel, a wide angle camera mounted at the ceiling of the robot environment, and a "magic sensor" that estimates the position of the robot based on its distance from the wall.

For simplicity, we will assume that the sensor data has been pre-processed to give us the following information:

- **From the camera**: A global coordinate position estimate in an Nx3 matrix called **CamPos** containing $[timestamp\ ,\ \hat{x}_{Cam}(kT_{Cam})\ ,\ \hat{y}_{Cam}(kT_{Cam})]$ for each time instant $kT_{Cam}$, where $k = 1,2, \dots, N$ and $T_{Cam}$ is the sampling period of the camera.
- **From the magic sensor**: A global coordinate position estimate in an Mx3 matrix called **MagicPos** containing $[timestamp, \hat{x}_{Magic}(kT_{Magic})\ ,\ \hat{y}_{Magic}(kT_{Magic})]$ for each time instant $kT_{Magic}$, where $k = 1,2, \dots, M$ and $T_{Magic}$ is the sampling period of the magic sensor.
- **From the encoders**: The angular position of each wheel in an Px3 matrix called **Enc** containing $\left[timestamp\ ,\ \hat{\varphi}_{Right}(kT_{Enc})\ ,\hat{\varphi}_{Left}(kT_{Enc})\right]$ for each time instant $kT_{Enc}$, where $k = 1,2, \dots, P$ and $T_{Enc}$ is the sampling period of the encoder. The angular position data ranges between 0 and 2π radians.

- You are also provided with the reference robot position in the matrix **RefPos** containing $[timestamp\ ,\ x(kT_{Ref})\ ,\ y(kT_{Ref})]$.

Positions and distances are expressed as relative units, where the radius of the PIE robot is one unit. The resolution of all sources of information is 0.01 units. For simplicity we will assume that: $T_{Cam} = T_{Magic} = 10\ T_{Enc} = 10\ T_{Ref}$ and that sampling times are synchronized. Bear in mind that this is hardly ever true in reality.

**EXERCISE 1**

Plot the position estimates from the camera and from the magic sensor against the reference data. 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? Explain in words how you could combine the data from these sensors in order to obtain a more accurate estimate of the robot position. For simplicity, we will consider that error in the $x$ and $y$ directions are independent.

**EXERCISE 2**

One way to fuse signals is to use weighted averaging, that is,

$signal_{fused} = A * signal_1 + B * signal_2$ where $A + B = 1$

The weights $A$ and $B$ should be proportional to how much you “trust” each signal. Use weighted averaging to fuse the camera and the magic sensor estimates. 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? (Pick an appropriate error measure)

**EXERCISE 3**

Another way of fusing sensor signals is to utilize prior knowledge about the problem, e.g. known limitations of the sensors or by knowing signal-to-noise ratios. One such approach is the Fuzzy approach, a concept developed by Lotfi A. Zadeh [1] in the mid-sixties. Let us ask ourselves the question: *Which of the sensors should we choose to use at a given location in the robot environment?* By considering a single sensor our answer is binary, either one or zero, i.e. $\{0,1\}$. We say our answer, also denoted *element*, either belongs or does not belong to a crisp set. By using classical sets this is formulated as $\mathcal{X}_A(x): X \to \{0,1\}$ which says: there *exists a function* $\mathcal{X}_A$ *mapping every element of the set* $X$ *to the set* $\{0,1\}$ [2]. The set $X$ is denoted here as the universe of discourse. However, what if we ask the question: *At a given location in the robot environment how much trust do we put in a particular sensor?* When we discuss the word *trust* the concept of fuzziness is introduced and we favour fuzzy sets rather than crisp sets. When using fuzzy sets the *membership* of an element to a set is not considered binary, instead different degrees of membership are allowed between one and zero, i.e. $[0,1]$. The mapping is now formulated $\mu_A(x): X \to [0,1]$ where $\mu_A$ is called the membership function which describes, in this case, the degree of trust to put in a particular sensor.

Create crisp sets that represent whether you should use the camera information or the magic sensor information to estimate the position of the robot. That is, find binary functions $f, g, p, q$ such that $\hat{x}_{fusion} = f(\hat{x}_{Cam}) + g(\hat{x}_{Magic})$ , and $\hat{y}_{fusion} = p(\hat{y}_{Cam}) + q(\hat{y}_{Magic})$, where $f + g = 1$ and $p + q = 1$ (constant function of value 1). How can these same functions be written as *if-then* rules? Plot the result of your fusion of estimates.

EXTRA POINTS for suggesting $f, g, p, q$ as smooth (not binary) fuzzy membership functions. Plot the resulting position estimate.

## EXERCISE 4

The position of the robot can also be estimated from the wheel encoder, given that we know the starting point of the robot. The position, $[\hat{x}, \hat{y}]$ , and direction, $\hat{\theta}$ , of the robot at time $k$ can be computed recursively:

$$\begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{\theta}(k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k-1) \\ \hat{y}(k-1) \\ \hat{\theta}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta x(k) \\ \Delta y(k) \\ \Delta \theta(k) \end{bmatrix} + error$$

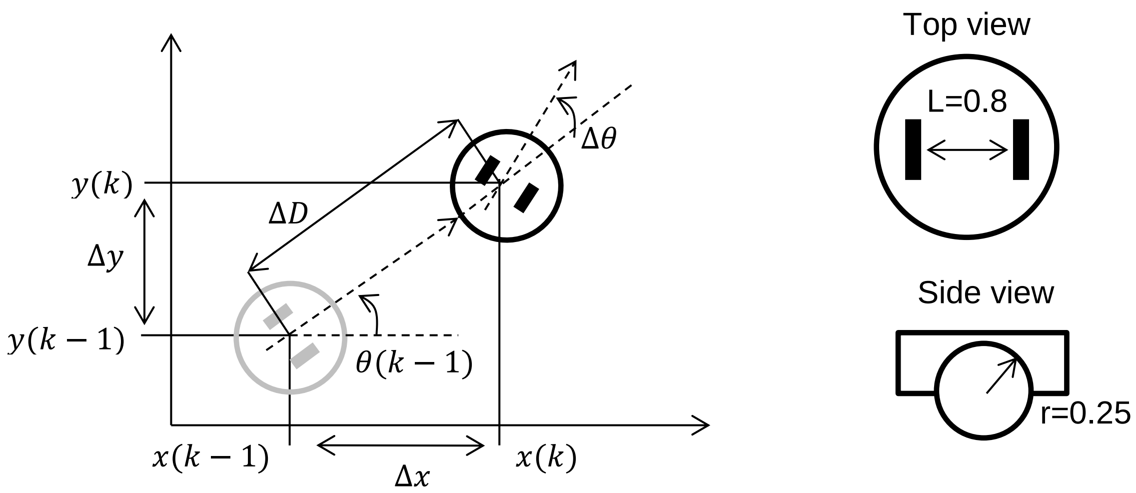


**Figure 2**. A model for estimating the position of the robot from the encoder data.

We will consider the following simplified model:

$\Delta\theta = \left(\Delta\hat{\varphi}_{Right} - \Delta\hat{\varphi}_{Left}\right) * r/L$

$\Delta D = \left(\Delta\hat{\varphi}_{Right} + \Delta\hat{\varphi}_{Left}\right) * r/2$

$\Delta x = \Delta D \cos(\theta)$

$\Delta y = \Delta D \sin(\theta)$

Implement the model above and estimate the position of the robot based on the encoder data. Note that the encoder data ranges from 0 to 2π then loops again to 0 (and vice-versa if it turns the opposite way). Plot your estimate of the position against the reference. What does the error look like? What could explain this behaviour? How could you overcome this problem?

## EXERCISE 5A

At every 10th sample of the encoder, a more accurate position is estimated from the camera and magic sensors. Ideally, this update would be done while the robot is on the move, but let's first assume that the robot moves for a while estimating its position with the encoders, then it stops and checks its estimate against the other sensor readings. The robot then updates its position estimate, and moves again. This process is repeated several times.

Implement this process and improve the encoder position estimation with the fused camera and magic sensor data. Plot your new estimate against the reference.

## EXERCISE 5B

Now assume that the robot receives data from the camera and magic sensors while it continues to move. Consider also that there is a small delay from the time the camera and the magic sensor make their readings to the time the robot receives this information. This means that, when the robot receives the information from the camera and magic sensors, this information is about a moment in the past, not the current position of the robot.

Assuming a delay equal to $4\,T_{Enc}$, explain how this changes the update of the robot's position estimate in Exercise 5A. Implement a position estimate that takes into account this delay. Plot your new estimate against the reference, and against the estimate from Exercise 5A. How are they different?

## REFERENCES

[1] L.A. Zadeh "Fuzzy Sets," Information and Control, vol. 8 pp. 338-353, 1965.

[2] L.H. Tsoukalas, R.E. Uhrig, "Fuzzy and Neural Approaches in Engineering," 1st edition, 1997, published by Wiley Interscience.