

Lab Assignment 2

RFID Localization

TA: Suraj Jog

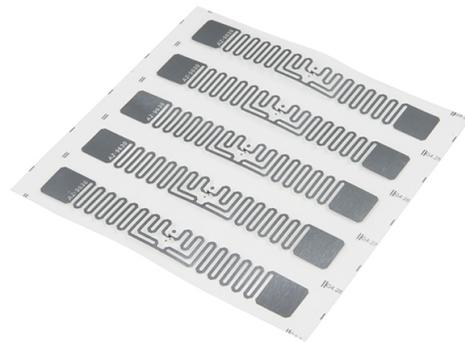
In this lab assignment, we will implement an RFID based localization system. For the theory behind RFID and wireless localization, students are encouraged to refer [Lecture 10](#) and [Lecture 11](#) from the course. We will first describe the hardware we will work with, and then enumerate the tasks in the lab.

1 RFID Hardware

Fig. 1 shows the hardware we will use in this lab. Fig. 1(a) shows the RFID reader. Fig. 1(b) depicts a passive RFID tag. The RFID reader is controlled using a API that can be found at the link <http://www.thingmagic.com/manuals-firmware>. We will use Java for this lab assignment, and the API has been set up on the machine in the lab. The API provides the data read from the tags along with all the associated metadata, which includes the RSSI of reflected signal, the phase of the channel, the frequency and so on. For a detailed description of the functionality of the API, the reader is referred to http://www.thingmagic.com/images/Downloads/Docs/MercuryAPI_ProgrammerGuide_for_v1.27.3.pdf. In this lab, our goal is to observe the phase changes of the channel as the RFID moves, in order to track the trajectory of the RFID tags.



(a) RFID Reader



(b) RFID Tag

Figure 1: RFID Hardware

2 Lab 2 Objectives

This lab consists of 4 tasks. Each task requires collecting some data during the lab session, and processing the data offline to generate output results for evaluation. The students are provided with some skeleton code for each task along with some test data for debugging purposes.

The folder provided to the students comprises of 4 subfolders -

- **SRC_CODE:** This folder contains the code for each task in the lab. The students need to fill in the blank portions of these codes and submit it for evaluation.
- **Test_Scripts:** This folder contains the code to test the implementations and generate results for evaluation.
- **Data_Files:** This folder contains the test signals for evaluation. It has two subfolders **Test** and **Lab_Data**. The files in **Test** is simulated data to test the correctness of your code, whereas **Lab_Data** will contain the data collected in the lab. Note that the **Lab_Data** folder is initially empty, and you will collect the data during your respective lab sessions.
- **Results:** This folder is initially empty. You must save the results to be submitted in this folder.

Finally, you must also submit a short report for the lab where you include the figures and answer the questions in each task.

The data collection process in the lab is the following. We will run a javascript that collects readings from RFID tags for some interval of time that we specify in the code. During this time interval, the RFID reader will collect multiple measurements continuously from all tags in its range. Each measurement is saved as a separate entry in a text file (separated by the character `\r\n`). Each measurement consists of the following fields -

- **EPC:** This field contains the unique ID for the RFID tag.
- **ant:** This field denote the Antenna ID through which the reading was captured. It takes values from 1 to 4 in our hardware setup.
- **time:** Denotes the time at which the RFID reading was measured
- **Frequency:** The frequency channel of the reading.
- **Phase:** The corresponding channel phase for the measurement.

We now describe each task of the lab in detail.

2.1 Task 1

In this task, you will work on data collected during the lab session. We will move the RFID tag in a straight line away from the RFID reader and collect phase measurements. This

process is depicted in Fig. 2. As shown in the figure, we move the RFID tag away from the RFID reader by a distance d and collect phase measurements at each position. As results you must include the following in your report.

1. From observing the phase change across the different positions, can you estimate the distance d ? If yes, what is the estimate for d from your measurements?
2. In order to estimate the distance d , are there any constraints on d with respect to the wavelength λ of the RF signal? If yes, state the condition and explain it.
3. Include a figure in the report, showing the phase variation of the tag across time.

The code for processing the data in this part should be written in the file ‘SRC_CODE/task1.m’. The path for the data file should be appropriately passed into the script.

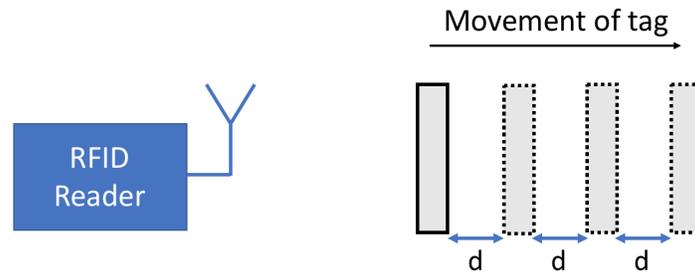


Figure 2: Task 1

2.2 Task 2

In this task, our goal is to estimate the AoA of the reflected signal from the RFID tag. Consider the antenna array setup shown in Fig. 3. Assuming that the channels measured at antenna element Rx_i is given by h_i , where $1 \leq i \leq N$, the AoA of the signal can be given by

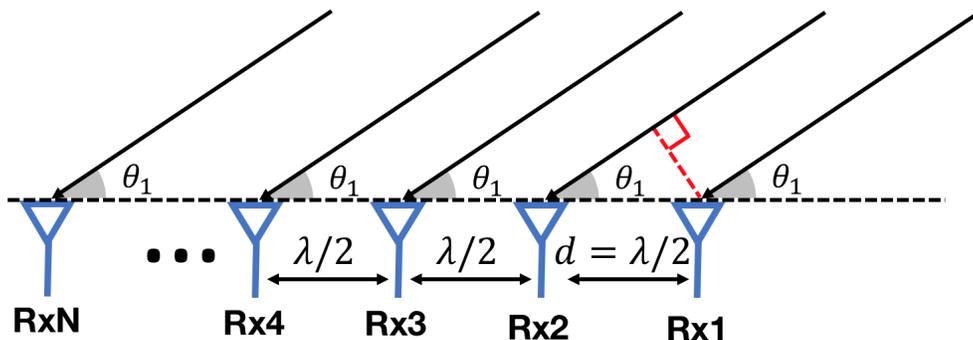


Figure 3: Task 2

$$\text{AoA} = \arg \max_{\theta \in [0, \pi]} \left| \sum_{i=1}^N e^{jh_i} e^{-j \frac{2\pi}{\lambda} d \cos \theta} \right| \quad (1)$$

However, since RFID tags reflect signals, the above formula needs to be modified in order to estimate the AoA. You should write down the modified formula in your report, and explain it.

Next, you must implement the function `estimate_aoa.m` to estimate the AoA given the channel measurements h_i , based on the modified formula. This function will be evaluated on two data sets.

2.2.1 Simulated Data

You will first evaluate your code over simulated data. To this end, you must run the test script `test_estimate_aoa.m`. It will save the output in the 'Results' folder, and this should be submitted for evaluation.

2.2.2 Data from RFID Hardware

Next, you will evaluate your code on data collected in the lab from the RFID hardware. The setup in the lab uses 3 antennas. The RFID tag will be placed along a chosen AoA, but at two different distances from the reader. You must collect phase measurements at each location, and based on these measurements you must estimate the AoA. During the lab, you must also measure the true angle of arrival to obtain the ground truth value. Your report should comment on the following -

- What is the true AoA value and what is the estimated AoA value as per your code? How off are they? Address these questions for both locations of the RFID tag.
- Is the value at one position significantly closer to the ground truth as opposed to the estimated value at the other position? If so, mention which location gave a better estimate, and give a reason why.

The code for processing the data in this subpart should be written in the file '`SRC_CODE/task2.m`'. This script should call the function `estimate_aoa.m` that you implemented in the previous subpart.

2.3 Task 3

In this subtask, you must plot the spatial beam pattern given the channel measurements on the antenna elements. Let the coordinate of the antenna elements i be defined by (x_i, y_i) , and the corresponding channel measurement at antenna element i be h_i . Therefore, the spatial gain (power) at some point (x, y) on the 2D plane can be given by

$$\text{Power}(x,y) = \left| \sum_{i=1}^N e^{\frac{j2\pi\sqrt{(x-x_i)^2+(y-y_i)^2}}{\lambda}} * e^{-jh_i} \right|^2$$

where N is the number of antenna elements, and λ is the wavelength of the RF signal.

However, due to the reflective nature of RFID transmissions, the above formula will need to be modified. You should write down the modified formula in your report, and explain it. Further, you should complete the code based on the above modified formula, in the function ‘`spatial_beam_pattern.m`’ to return the Power value at some point (x,y) in 2D space. This function should takes the following input

- The coordinate (x,y)
- The matrix `antenna_pos`, where the i^{th} row represents the tuple (x_i,y_i)
- The channel vector h , where h_i represents the channel observed at the i^{th} antenna element.
- λ which is the wavelength of the RF signal.

2.3.1 Simulated Data

To evaluate this function, you should run the test script ‘`Test_spatial_beam_pattern.m`’. This will save the output to the Results folder. An illustrative spatial beam pattern is shown in Fig. 4

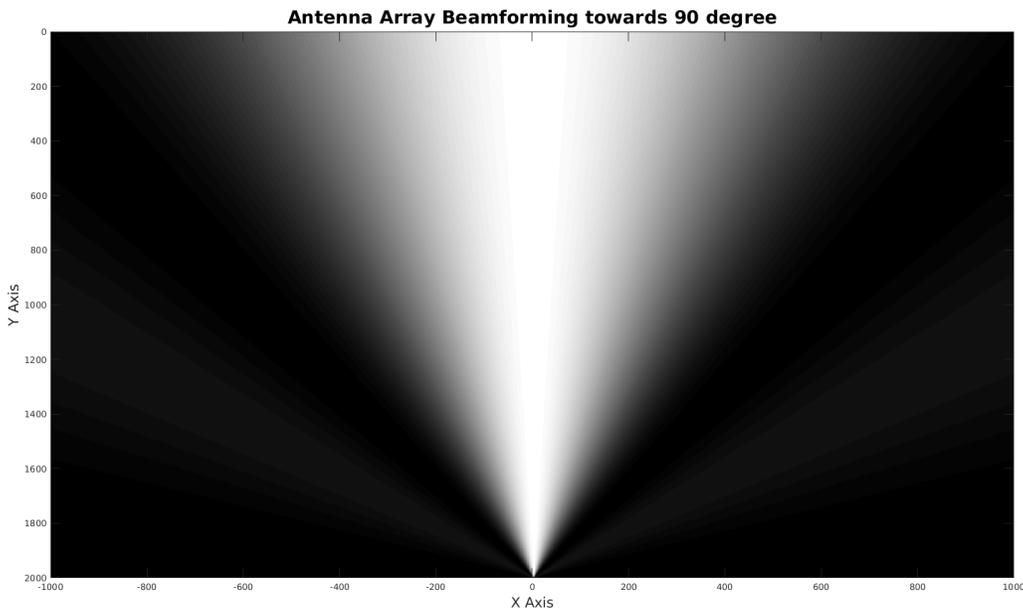


Figure 4: Illustrative Spatial Beam Pattern

2.3.2 Data from RFID Hardware

In this section, you must plot the spatial beam patterns using the data observed from Task 2. You must use the data corresponding to the farther away location of the RFID tag from the reader. From the data, you must obtain the channel at each of the three antenna elements from the setup in Task 2. Let us refer to the channel at antenna element 1 as h_1 . Similarly we define h_2 and h_3 .

You must now plot two figures — 1) The spatial beam pattern considering only antenna 1 and antenna 2 and their corresponding channels, and 2) The spatial beam pattern considering antenna 1 and antenna 3 along with their corresponding channels. The skeleton code has been provided in ‘`Spatial_beam_pattern_hardware.m`’. In this code, you must fill in the channel phases (in degrees) observed at the three antenna elements, along with the coordinates of each antenna. This code will call your function ‘`spatial_beam_pattern.m`’ that you implemented. You can assume antenna 1 to have the coordinates (0,0), and define the coordinates of antenna 2 and antenna 3 accordingly.

Make sure to include the above two plots in your report. Additionally, you must address the following aspects in your report.

- From the two figures, what are the differences that you observe?
- Comment on the tradeoff between resolution and ambiguity here, and explain why it occurs.

2.4 Task 4

The localization technique described in Task 2 and 3 required multiple antennas. However, multiple antennas could prove expensive. Therefore, in this task, we try to localize a moving object using just 1 RFID reader antenna, but with multiple RFID tags. This is a more scalable solution, since RFID tags are extremely cheap and we can fix multiple tags to the object being localized.

The setup is as follows — an object (shown in Fig. 5) is affixed with multiple RFID tags. The relative positioning of the RFID tags on the object is measured a priori. Our goal here is to track the trajectory of the object as it moves. To this end, we will move the object on some trajectory (marked by the black line in Fig. 5 (U shaped)). We collect phase measurements from the two tags as the object moves on the trajectory, and based on these phase measurements we estimate the trajectory of motion of the object.

The core idea is as follows. When an RFID tag moves by some distance Δd , the associated phase change observed at the RFID reader will be $4\pi \frac{\Delta d}{\lambda}$ (assuming no phase wrap takes place). If we knew the initial position of the RFID reader with respect to the tag, then based on the phase change we can measure the new distance of the tag from the RFID reader ($d + \Delta d$). However, this information from a single tag is not sufficient to localize the reader since we do not have information about the relative direction. Therefore, we leverage this information from two tags, and estimate the relative position of the RFID reader using

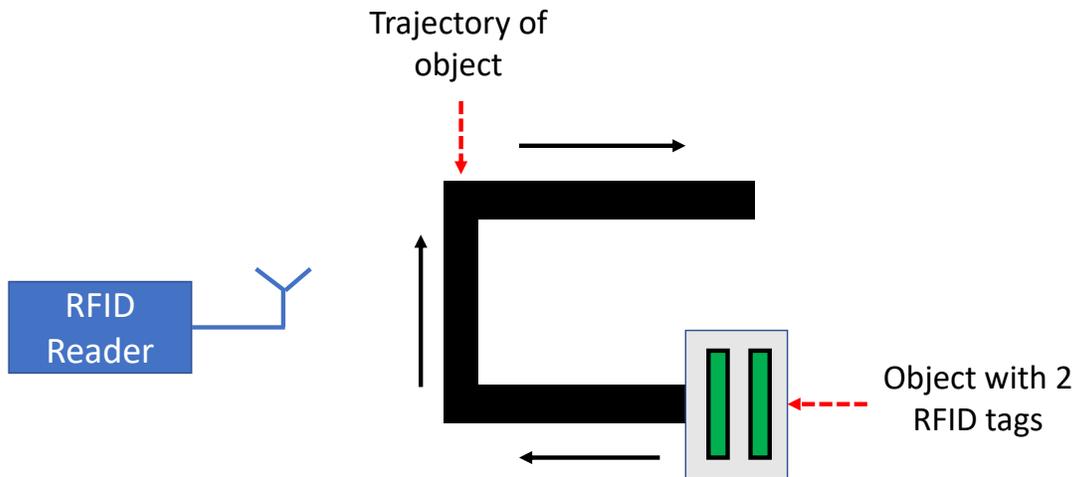


Figure 5: Task 4

trilateration. If we perform this estimation across all successive points along the object's trajectory, we can obtain the entire trajectory of the reader from the frame of reference of the object. Finally, we must rotate this trajectory using matrix multiplications, in order to retrieve the true trajectory of the object.

Your task in this lab is to implement the above system. We further break down this task into the following subtasks.

2.4.1 Estimate change in distance

You must complete the code in 'variation_distance.m'. This function should take as input the previous distance estimate, the observed phase change and the wavelength of the RF frequency. Based on these, it should output the new distance estimate after accounting for the phase change. In order to test this function, you must run the script 'Test_variation_distance.m'. This will save an output file to the 'Results' folder.

2.4.2 Trilateration

You must fill in the code in the file 'trilateration.m'. This code will take as input the coordinates of the two anchor points (RFID tag 1 and tag 2 in our case) in the 2D plane, and the corresponding distance of the RFID reader from the two anchor points. Based on these measurements, we will localize the RFID reader in the 2D plane. In order to test this function, you must run the script 'Test_trilateration.m'. This will save an output file to the 'Results' folder.

Note: One should notice that in the presence of just 2 anchor points, the object cannot be localized uniquely. This is because the two circles will intersect at 2 points, leading to

ambiguity. However, for the sake of this assignment, you must assume that the object lies in the top half of the 2D plane. That is, if we assume the anchor points to lie on the x-axis, then the object will always have a positive y-coordinate. Therefore, among the two intersection points of the circles, your function must return the point with the larger y-coordinate.

2.4.3 Trajectory Rotation

In the above two blocks, we are estimating the relative trajectory of the RFID reader from the frame of reference of the object. Therefore the above blocks will output a sequence of relative positions of the reader with respect to the object. However, in order to obtain the true trajectory of the object, we must invert this trajectory back to a stationary frame of reference.

This can be modelled as a simple matrix multiplication. That is, if we have a matrix v where the i^{th} row of v denotes the relative (x,y) coordinate of the RFID reader at time i , then the true trajectory of the object can be obtained as $v \times R$ where R is a 2×2 matrix.

In your report, you must derive the matrix R . You should use this matrix R , to complete the code in `invert_traj.m`. In order to evaluate the function, you should run the script 'Test_invert_traj.m'. This will save an output file to the 'Results' folder.

2.4.4 Estimate Trajectory

Now that we have all the blocks ready, we can use them to track trajectory of the object. During the lab session, you will collect phase measurements from the RFID tags as the object moves on a trajectory. You will process this data to estimate the relative trajectory of the reader, and further obtain the true trajectory of the object.

Please note the following -

- During the lab, you should measure the initial position of the RFID reader relative to the tags, and the relative positioning of the RFID tags on the object.
- You should also measure the length and shape of the trajectory so you have a measurement of the ground truth trajectory.
- The forward facing direction of the object will be assumed to be the direction of the positive y axis, so you must choose the coordinates of your anchor points accordingly.

Finally, in your report, you must include the following

- A figure showing the estimated trajectory superimposed with the ground truth. Please note that we are interested in the shape of the trajectory as opposed to the absolute trajectory.
- In order to unambiguously estimate the trajectory, are there any restrictions on how fast the object should move? If yes, could you estimate the limit and explain it?

The code for processing the data in this subpart should be written in the file ‘SRC_CODE/task4.m’. This script should call the functions that you implemented in the previous subparts.

2.4.5 BONUS - Optimal Filtering

Finally, the estimated trajectory above will look very noisy. There are various optimal filtering techniques that can help to get better estimates of the trajectory from the noisy ones obtained in the previous subpart. Can you implement some optimal filtering technique to improve the estimated trajectory? You must mention the technique used in the report, and justify your intuition as to why it should work. Some examples that you can try are given below, along with links for tutorials. Also feel free to use ideas from other sources, but please cite whatever material you refer to.

- Kalman Filters - <http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>
- Extended Kalman Filters - https://home.wlu.edu/~levys/kalman_tutorial/
- Particle Filters - <https://www.lancaster.ac.uk/pg/turnerl/PartileFiltering.pdf>

The code for this portion is to be written in the file ‘SRC_CODE/optimal_filter.m’. Note that for this subpart, no skeleton code has been provided, and you need to design the structure of this code yourself.

2.5 Provided Code

In order to help you, we provide a function ‘read_data.m’ in the folder ‘SRC_CODE/functions’. This function takes as input the path of the data file and returns a cell data structure `channel_log`. Each row in `channel_log` corresponds to one RFID measurement.

3 Submission

You should submit the following for evaluation as a single .zip file to Compass.

1. The folder `SRC_CODE` with all functions that you implement.
2. The `Results` folder which includes the .mat files written by the test scripts.
3. The report answering all questions in this document, and providing figures where required.