Lab Assignment 3

RFID Localization

In this lab assignment, we will implement an RFID based localization system. Before starting the lab, please review the class lectures on RFID. We provide the data for students to work on, and in Appendix A we give more details on the RFID hardware used to capture data.

1 Objectives

This lab consists of several tasks. The students are provided with some python skeleton code for each task along with some test data and testing code for debugging purposes.

The folder provided to the students comprises of:

- The code for each task in the lab in jupyter notebooks. The students need to fill in the blank portions of these codes and submit it for evaluation. The notebooks also contain code cells for testing the implemented functions and generate evaluation results.
- **Lab_Data:** This folder contains the data collected in the lab using the RFID hardware for evaluation.
- **Test:** This folder contains simulated data to test the correctness of individual portions of your code.
- **Results:** This folder is initially empty. You must save the results to be submitted in this folder.

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

The format of the RFID data is as follows. We 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 denotes 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. Note that the measurements are in kHz and need to be appropriately converted for processing.
- Phase: The corresponding channel phase for the measurement.

In order to help you, we provide a function 'read_data' in the file 'utils.py'. This function takes as input the path of the .txt data file an returns a dictionary data structure channel_log. Each row in channel_log corresponds to one RFID measurement. We now describe each task of the lab in detail.

2 Distance Estimation from Phase Measurements

In this task, you will work on the data in 'Lab_Data/lab3_task1.txt'. The setting of the data collection is as follows. We move the RFID tag in a straight line away from the RFID reader and collect phase measurements. This process is depicted in Fig. 1.

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 the measurements in the txt file?
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.

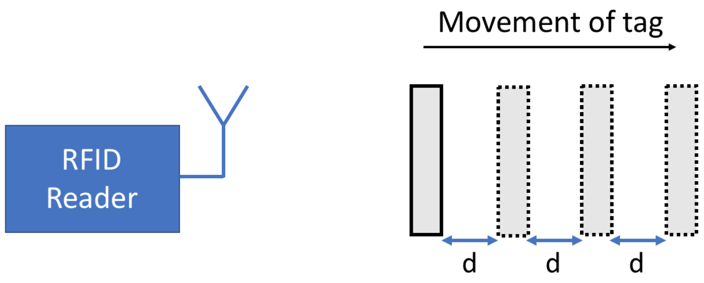

Figure 1: Data Collection Setting

You will find the skeleton code for this task in 'task1.ipynb'. Make sure the input data path in the code is correct.

3 AoA Estimation

3.1 Linear Antenna Arrays

In this task, our goal is to estimate the AoA of the reflected signal from the RFID tag. Consider the linear antenna array setup shown in Fig. 2. Assuming that the channels measured at antenna element Rxi is given by h_i, where $1 \le i \le N$, the AoA of the signal can be given by:

$$AoA = argmax_{\theta \in [0,\pi]} \left| \sum_{i=1}^{N} e^{jh_i} e^{-ji\frac{2\pi}{\lambda} d cos\theta} \right| \tag{1}$$

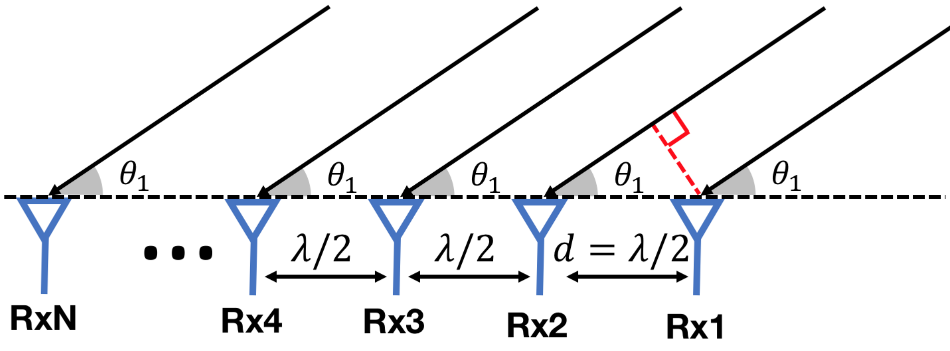

Figure 2: AoA estimation with Linear Array

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` to estimate the AoA given the channel measurements hi, based on the modified formula. The `estimate_aoa` function should return AoA values in the resolution of 1 degree. This function will be evaluated on two data sets.

3.1.1 Simulated Data

You will first evaluate your code over simulated data. To this end, you must run the test cell inside the `task_1.ipynb` file. It will save the output in the `Results` folder, and this should be submitted for evaluation.

3.1.2 Data from RFID Hardware

Next, you will evaluate your code on data collected in the lab from the RFID hard- ware. The setup in the lab uses 4 antennas. The RFID tag is placed along a chosen ground truth AoA, and we collect phase measurements, provided in the file `Lab_Data/lab3_task2.txt`. Based on these phase measurements you must estimate the AoA of the RFID tag. Your report should comment on the following:

- What is your estimated AoA value?
- The ground truth value of the AoA is 60 deg. If your estimated value is not close, what could be the most probably reason and explain why?

The code for processing the data in this subtask should be written in `task_2.ipynb`. This script should call the function `estimate_aoa` that you implemented in the previous subtask.

3.2 Circular Antenna Arrays

Finally, in this subtask you will estimate the AoA from a circular array of RFIDs as shown in Fig. 3. The circular array has n number of RFID tags and the radius of the array is R. Fig. 3 also shows the channels b_k measured at antenna k for a signal arriving from θ direction. Based on this, write down the formula similar to Equation 1 for calculating AoA for such a circular array. Assume the channels measured from RFID element i is given by h_i. Note that your formula should account for the fact that RFID tags reflect signals, similar to the previous subpart.

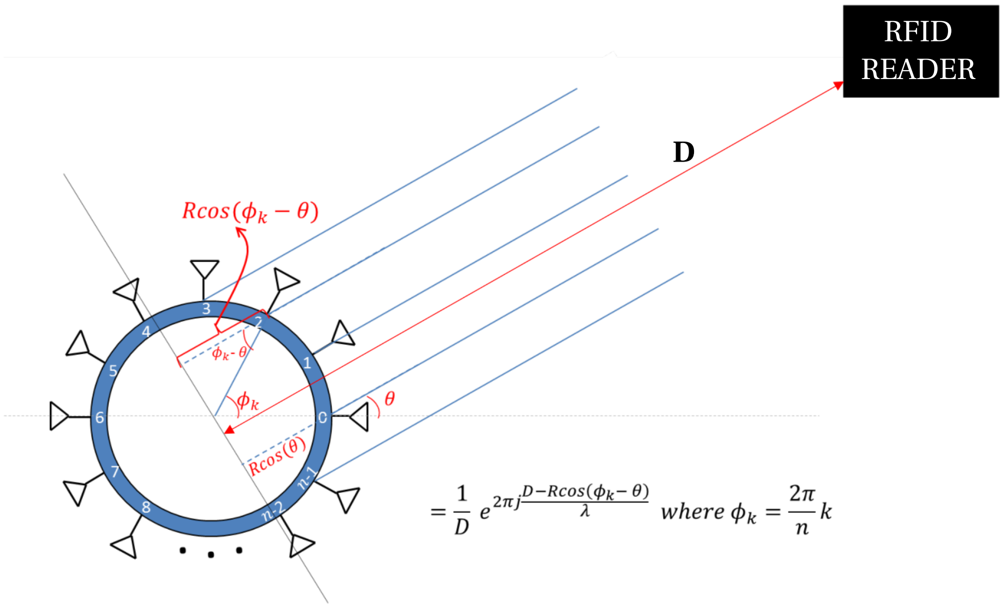
RFID READER
D
$Rcos(\phi_k - \theta)$
$\phi_k - \theta$
ϕ_k
θ
$Rcos(\theta)$
0
1
2
3
4
5
6
7
8
n-1
n-2
$= \frac{1}{D} e^{2\pi j \frac{D - Rcos(\phi_k - \theta)}{\lambda}}$ $where\ \phi_k = \frac{2\pi}{n} k$

Figure 3: AoA Estimation with Circular Array

Now you must implement the function estimate_aoa_circular.m to estimate AoA given the channel measurements hi's from a circular RFID array. The function also takes as input the radius of the array R and the λ. The `estimate_aoa_circular` function should return AoA values in the resolution of 1 degree. To evaluate your function, you must run the test script inside the task notebook which will save the output in the `Results` folder.

4 Spatial Beam Patterns

In this task, 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:

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

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` to return the Power value at some point (x, y) in 2D space. This function should take the following inputs

- 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.

4.1 Simulated Data

You will first evaluate your code over simulated data. To this end, you must run the test cell inside the `task_3.ipynb` file. It will save the output in the `Results` folder, and this should be submitted for evaluation. An illustrative spatial beam pattern is shown in Fig. 4.

4.2 Data from RFID Hardware

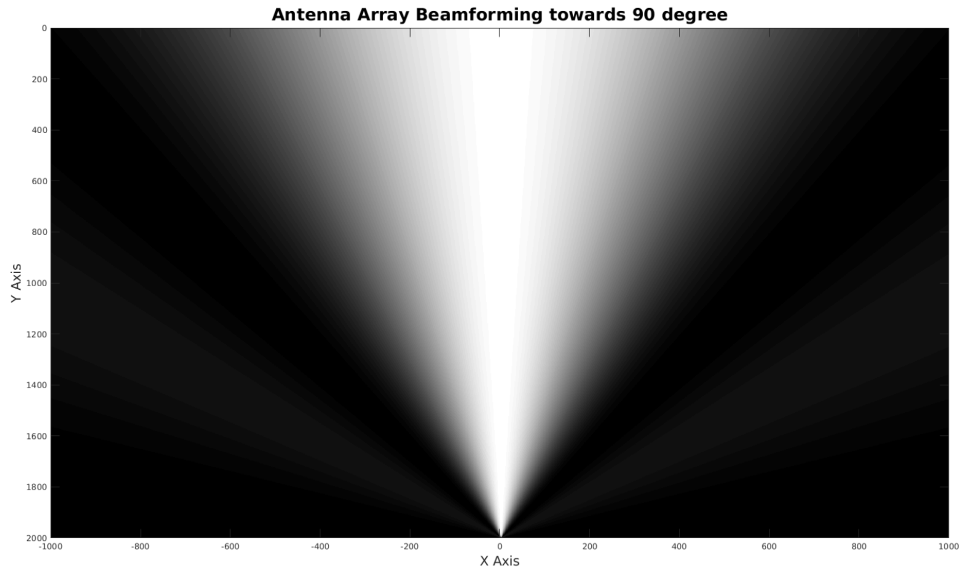

Figure 4: Illustrative Spatial Beam Pattern

In this section, you must plot the spatial beam patterns using the data captured from RFID hardware. The skeleton code has been provided in `task_3` notebook. This code will call the function `spatial_beam_pattern` that you implemented. For this task, we only look at the channel measurements on the first three antennas, and we refer to their channel measurements as $h1$, $h2$ and $h3$ respectively. The channel values as well as the position coordinates of all antenna elements is provided in the skeleton code.

You must now plot two figures:

- The spatial beam pattern considering only antenna 1 and antenna 2 and their corresponding channels
- The spatial beam pattern considering antenna 1 and antenna 3 along with their corresponding channels.

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 trade-off between resolution and ambiguity here, and explain why it occurs.

5 Localization Using Multiple RFID Tags

The localization technique described in previous sections 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 apriori, and provided to you in the skeleton code where needed. Our goal here is to track the trajectory of the object as it moves. To this end, we 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.

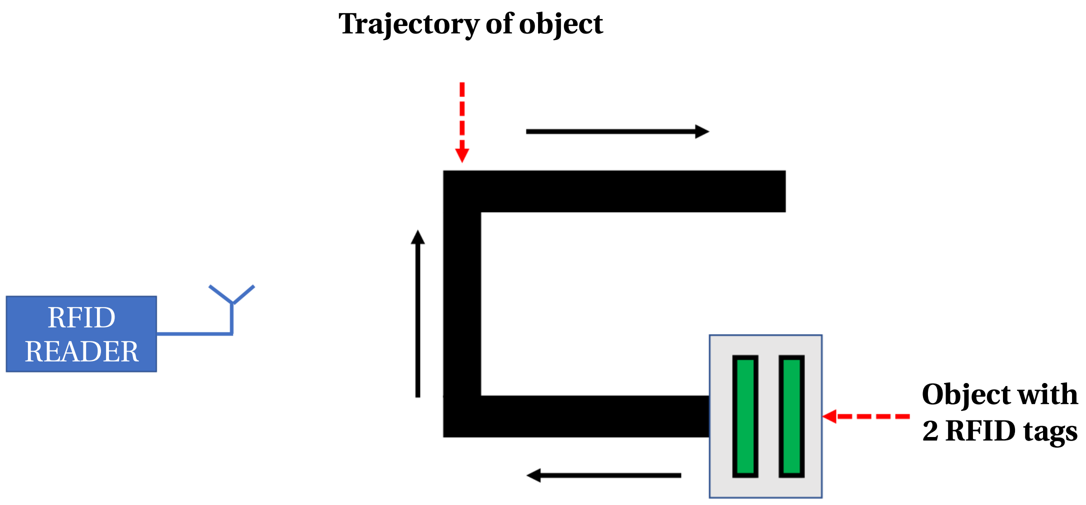

Figure 5: Trajectory Localization with Multiple RFIDs

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 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 sub-tasks.

5.1 Estimate change in distance

You must write the function `variation_distance` inside `task_4.ipynb`. 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 test cell inside `task_4.ipynb`. This will save an output file to the `Results` folder.

5.2 Trilateration

You must fill in the function `trilateration` inside `task_4.ipynb`. 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 trilateration test cell inside `task_4.ipynb`. 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.

5.3 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 modeled as a simple matrix multiplication. That is, if we have a matrix v where the $i_t h$ 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 function `invert_traj`. In order to evaluate the function, you should run the script inverted trajectory test cell inside `task_4.ipynb`. This will save an output file to the `Results` folder.

5.4 Estimate Trajectory

Now that we have all the blocks ready, we can use them to track trajectory of the object. We collect phase measurements from the RFID tags as the object moves on the trajectory, and provide them in the data file `Lab_Data/lab3_task4.txt`. You will process this data to estimate the relative trajectory of the reader, and further obtain the true trajectory of the object. The code for this subpart should be written in the file `task4.ipynb`. The script should call the functions that you implemented in the previous sub-parts. The skeleton code provides you with all the measurements you need, such as the initial position of the RFID reader relative to the tags, the relative positioning of the RFID tags on the object, and the shape and dimensions of the ground truth trajectory of the object.

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 (i.e. the relative 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?

6 Submission

1. The completed notebook for each task.
2. a `functions.py` file containing all functions that you implemented for the four tasks.
3. The Results folder which includes the .mat files generated by the test scripts.
4. The report answering all questions in this document, and providing figures where required.