University of Illinois at Urbana-Champaign
Computer Science

CS 498 DV, Fall 2021 **Problem Set 1** Sep 4, 2021

Deadline: This problem set is due by **11:59 pm on September 23, 2021**. You have 5 late days combined for both the problem sets. For example, if you submit this problem set late by 2 days, you can get an automatic extension of 3 days for the second problem set.

Solution: Please submit a pdf file containing the answers to the text-based questions and a jupyter notebook for the coding questions. You can also program in Matlab and submit a Matlab script instead of the jupyter notebook.

Gradescope: Please submit the solution on Gradescope. You can sign up using entry code: BPGJG5.

3-credit Version: If you are taking the 3-credit version, you can skip one of the problem sets. If you attempt both problem sets, you can use the best score across the two problem sets.

1 Wireless Communication

1.1 Modulation and Wireless Channel [5 points]

Modulation: Recall from lectures that we convert bits into complex numbers using the process of modulation. For this problem, we used a simple 2-bit modulation where: 00 maps to 1, 01 maps to 1j, 11 maps to -1, and 10 maps to -1j, where $j = \sqrt{-1}$.

Channel: The wireless channel is represented by a complex number that captures the effect of the medium that the wireless signal travels through. If a device transmits, x, the receiver receives, $y = hx$, where h is the channel.

Preamble: The wireless signal begins with a preamble to help the receiver infer the wireless channel. For this problem, we set the preamble to a 32 bit sequence:
$[1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1]$.

We used the configuration above to transmit data between two devices. The data contains the preamble followed by a message. The received signal is logged in the attached file *rx_signal.npy*. Use a Jupyter iPython notebook to write the following code (you can use the notebook base.ipynb as a starting point):

i Encode the preamble using the modulation defined above into a sequence of complex numbers. List the 16 complex numbers into your answer document.

ii Use the preamble to estimate the wireless channel, h. Write the estimated channel value in the answer document.

iii Use the estimated channel value to decode the message bits. The message contains 128 bits. You do not need to report the decoded bits.

iv Use the function get_ascii in base.ipynb to convert the message into ASCII text. Write that text down in your answer document. (Hint: It should form a complete sentence.)

1.2 MIMO-I [10 points]

For this problem, we retain the same preamble and modulation defined above. However, instead of a single transmission, let us try decoding two streams in a 2×2 MIMO system. In a 2×2 MIMO system, at the transmitter, the first antenna sends a preamble, then the second antenna sends a preamble, and then both the antennas simultaneously send their message. The receiver receives a combination of both messages at its two antennas. We want to decode the two message. We have logged the received signal in two separate files: *rx_mimo_0.npy*, and *rx_mimo_1.npy*. Your goal is to decode the two messages transmitted by the transmitter.

i Identify the 2×2 channel matrix and write it down in your answer document. Remember, we can use the preamble to estimate the channel values.

ii Use the estimated channel values to decode the two messages. Each message contains 128 bits. You do not need to report the decoded bits.

iv Use the function get_ascii in base.ipynb to convert the message into ASCII text. Write that text down in your answer document.

1.3 MIMO-II [5 points]

After solving the above problem, Ben was convinced that wireless nodes with multiple antennas perform better than those with single antennas. Hence, they went ahead and purchased two multi-antenna nodes containing two antennas each. After setting them up, they called over Alex to perform channel measurements and found that the channel matrix H they measured for their 2×2 MIMO system was the following:

$$H = \begin{pmatrix} \frac{1+j}{\sqrt{2}} & \frac{-3+3j}{\sqrt{2}} \\ 2j & -6 \end{pmatrix}$$

where $j = \sqrt{-1}$. After looking at H, Alex remarked that their 2×2 system is at most as good as a single antenna transmitterreceiver pair. But Ben insisted that since each of their nodes has 2 antennas, they should be able to achieve twice the bit-rate of signal antenna system. Who is right: Alex or Ben? Why?

Hint: Try sending two symbols over this channel and see if you can decode them independently. If not, why?

2 Wireless Localization

2.1 Wireless Channel and Location[5 points]

Consider, a transmitter, tx, that transmits symbol x to the receiver, rx. As the signal goes over the medium, it suffers attenuation and a phase rotation, denoted in class by the channel h. Recall that h is a complex number, with its magnitude representing the attenuation. The receiver receives signal y, where $y = hx$. If the channel, h, is just an effect of the propagation delay, it can be modelled as $h \propto \frac{1}{d} e^{\frac{-j2\pi d}{\lambda}}$, where λ is the signal wavelength. However, the channel is corrupted by several other offsets due to hardware differences, as we have discussed in class.

For this problem, we took a single antenna transmitter and a two-antenna receiver and ran a simple measurement for you (1). The transmitter transmits 100 WiFi packets and the receiver measures the channels for each of the packets. As you may recall, WiFi uses OFDM and thus, transmits data on multiple sub-carriers, each with slightly different frequency. However, we handled that for you and for this question, we will consider channel measurements on a narrowband on frequency 5.5 GHz. The channel measurements are in the file, *h_static.npy*. This file contains a 100×2 matrix representing the channel measurement on the two antennas (first column is antenna 1). *t_static.npy* contains a 100×1 matrix containing the timestamps of the measured channels in milliseconds.

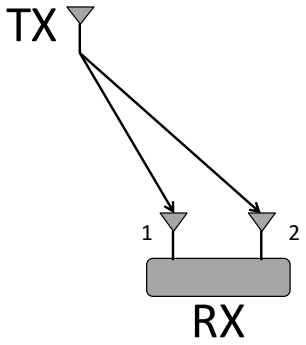

Figure 1: This figure demonstrates the setup for section 2.1

i Plot the phase of the channel measured on antenna 1 with respect to time. Include the plot in your submission. (Hint: use function angle() in numpy to get the phase of a complex number.) You will observe that the channel phase does not remain constant. This is because the transmitter and receiver do not share the same hardware clock and therefore end up using slightly different frequencies for transmission. This effect is called the Carrier Frequency Offset (CFO).

iii Plot the phase of the ratio of the channel measured on antenna 1 to the channel measured on antenna 2 for each packet, with time on the x-axis. Include the plot in your submission.

iv Does the phase remain constant (almost) for the ratio of the channels? If yes, why? If no, why not?

2.2 Let's move the antenna! [5 points]

For this part, we decided to spice things up a bit. We put the transmitter on a Roomba robot and moved the Roomba at a constant speed, v, from left to right and right to left. The setup for this experiment is shown in figure 2

As we moved the Roomba, the transmitter was transmitting packets at regular intervals and we measured the channels at the receiver. The channel measurements are in two files *h1_move.npy* and *h2_move.npy*. Each file contains one 200×2 matrix corresponding to the channel measurements on two receive antennas (same notation as the previous question). The corresponding timestamps are in matrices *t1_move.npy* and *t2_move.npy* respectively, representing the timestamps (in ms) for the respective channel measurements.

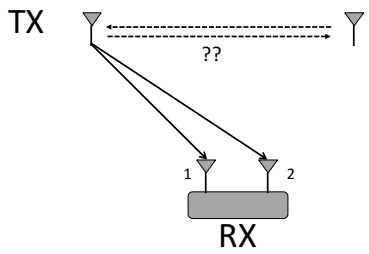

Figure 2: This figure demonstrates the setup for section 2.2

Use your understanding of the channel phase to figure out the direction of the motion of the roomba based on the channel measurements (left to right or right to left) for the two measurements. Describe your approach in the solution. (Hint: You might find the unwrap() function in Numpy useful.)

2.3 Antenna Arrays [10 points]

ArrayTrack introduced you to linear antenna arrays. In this section, we will introduce the circular antenna arrays.

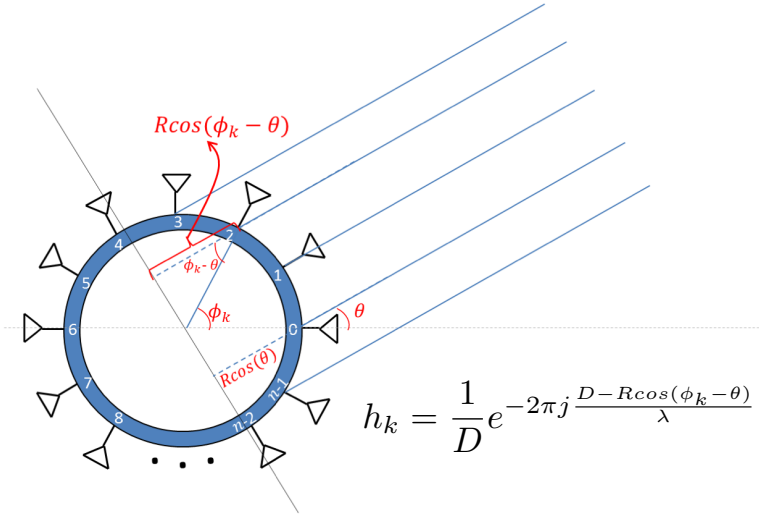

Figure 3: Circular antenna array.

Consider the circular antenna array shown in Figure 3. Each of the antennas receive a signal from a far-off transmitter (located at distance D) and measure the channel. The goal is to find the direction of the transmitter, θ, with respect to the line joining the centre and antenna 0. The power of the signal, P, coming from the direction θ' can be given by the following formula:

$$P(\theta') = |\sum_{k=0}^{n-1} h_k e^{-j2\pi \frac{R\cos(\phi_k - \theta')}{\lambda}}|^2 \tag{1}$$

Here, R is the radius of the antenna array and ϕ_k is the angular position of the k^{th} antenna. Intuitively, this formula just compensates for the additional phase accumulated by channel measurements as the signal travels to different antennas. Again, this is very similar to the linear antenna

array discussed in ArrayTrack. The value of P is high for values of θ' which correspond to the signal from the transmitter or signal reflections due to the environment.

In practice, building such antenna arrays is a tough job. You need to calibrate for hardware offsets in each antenna receive chain, which can be very painful. In order to avoid this, people use rotations of a single antenna system to emulate a rotating antenna array. We did this for you. We strapped the two antenna receiver from figure 1 to a Roomba and rotated it at one rotation per 12.25 seconds. The channel measurements, h, as well as the time stamps, t, for each of the channels are given in file *circular_h.npy* and *circular_t.npy*. R is 20.8 cm for this experiment. The frequency of the signal is 5.5 GHz.

Your goal is to obtain the multipath profile (P at different values of θ' varying from -180 degrees to 180 degrees at intervals of 1 degree) for the signal from the transmitter. Plot the multipath profile and add it to your solution. You should keep the following in mind:

- To compensate for hardware offsets between the transmitter and the receiver, you should apply equation 1 to the ratio of channel measured on antenna 2 divided by the channel measured on antenna 1.
- The antenna sampling is not necessarily uniform. So, you should calculate ϕ by using the time stamps and the angular velocity.
- You can check the correctness of your solution by plotting the values of P in the file *MultipathProfile.npy*. In this file, the value of P is obtained at $\theta' = -180, -177, -174, \ldots, 180°$.