111550022 施采緹
Lab1 report

<How the result being obtain?>

- **Task 1 - Calculate AoD (Angle of Departure)**
  - The actual AoD for each Rx is computed using the atan2d function, which finds the angle between the Tx and each Rx. Also, the angles are constrained to a range of 0–180 degrees. The closest predefined beam direction (from tx beam direction) is chosen as the optimal beam direction.
  - $AoD = atan2d(rx\_y - tx\_y, rx\_x - tx\_x)$
  - $AoD = mod(AoD, 180)$
  - $optimal_{beam_{direction}} = argmin(\quad | tx_{beam_{direction}} - AoD | \quad)$

- **Task 2 - Calculate Rx Power and SNR**
  - The wavelength (lambda) is calculated from the speed of light and the transmission frequency. And beamforming gain (G_tx) for each Rx is determined using the uniform and dtft functions. The distance between the Tx and each Rx is then computed. Using the Friis equation, the received power is calculated based on transmission power, gain, and distance. The SNR (Signal-to-Noise Ratio) is then obtained by subtracting the noise power (N0_dBm) from the received power.
  - $\lambda = freq/c$
  - $\psi = 2\pi d\cos(\phi)$
  - **uniform function:**
    `[weights, hpdw] = uniform(d, optimal_beam_direction(i), tx_antenna_number);`
  - **dtft function:**
    `Gain = abs(dtft(weights, -psi)).^2 / tx_antenna_number; di=‖Rxi-Tx‖`
  - **Euclidean distance (Rx, Tx):** $di = \| Rxi - Tx \|$
  - **Friis free-space equation for received power:**

$$P_{Rx} = P_{Tx} + G_{Tx} + G_{Rx} - 20\log_{10}(d) - 20\log_{10}(f) + 20\log_{10}\left(\frac{c}{4\pi}\right)$$

  - $SNR_{dB} = P_{Rx} - N_0$

- **Task 3 - Calculate Interference power and SINR**
  - The interference power from the second beam at the first Rx is calculated using the Friis equation. And the SINR is computed by dividing the received power of beam 1 by the sum of interference power (from beam 2) and noise power.
  - $SINR = S/(I + N)$ – unit watt.
  - **Rx1's interference power:**
    `P_tx_dBm + friis_equation(freq, G_interference(2), 1, rx_Dists(1))`
  - **Rx1's SINR:**

```
(P_tx_dBm + friis_equation(freq, G_tx(1), 1, rx_Dists(1))) -
Wat2dBm(dBm2Wat(P_tx_dBm + friis_equation(freq, G_interference(2),
1, rx_Dists(1))) + dBm2Wat(N0_dBm))
```

<Explanation of Commit history>

For the 1st commit (ignore the initial commit from TA), I add the code for task 1, including finding the actual angle θ1 and θ2 for user 1 and user 2, which is the requirement output of task1. Also, I calculate the tx beam direction closest to user 1 and user 2. (will be used in task2)

And the 2nd commit, I add a function file `printf()` for debug use. It's the function just like `fprintf()` in MATLAB but with tab in front of the display string.

In the 3rd commit, it's the task2 generate from GPT, and originally, I didn't really understand the method(formula) of G_{tx} calculation. The only reference `ewa_function` I gave to GPT was `uniform.m` and the subfunction it calls (steer, scan). And according to the response, I calculate G_{tx} by *img1*. Also I compute the Rx power and SNR with the wrong understanding about what the `friis_equation()`, I fed the wrong input format by `dBm2Wat` of G_{tx} when calculating the Rx power!

Task 2 計算步驟

1. **計算 Tx 陣列增益**
   - 使用 `uniform.m` 來獲取天線權重 (steered weights)。
   - 天線增益可以用**天線方向性增益公式**來計算：

$$G_{tx} = \frac{|\mathbf{w}^H \mathbf{a}(\theta)|^2}{N}$$

   其中：
   - `w` 是方向性權重 (使用 `steer()` 計算)。
   - `a(θ)` 是波束對應的方向向量。

```
% Hint: Calculate the Tx Gain
G_tx_max = zeros(rx_node_number, 1);
G_tx = zeros(rx_node_number, 1);
for i = 1:rx_node_number
    [weights, hpdw] = uniform(d, optimal_beam_direction(i), tx_antenna_number); % 計算權重
    array_factor = sum(weights); % 陣列因子
    G_tx_max(i) = 10 * log10(abs(array_factor)^2 / tx_antenna_number); % 轉成 dB
    dprintf(sprintf('User %d Tx Gain max: %.2f dB\n', i, G_tx_max(i)));

    G_tx(i) = G_tx_max(i) - 12 * ((optimal_beam_direction(i) - optimal_beam_resolutions(i)) / hpdw)^2; % 根據角度差距計算 Tx Gain
    printf(sprintf('User %d Tx Gain: %.2f dB\n', i, G_tx(i)));
end
```

```
% Hint: Calculate the Rx power and SNR
P_rx_dBm = zeros(rx_node_number, 1);
SNR_dB = zeros(rx_node_number, 1);

for i = 1:rx_node_number
    distance = norm(rx_location(i, :) - tx_location); % 計算 Tx 到 Rx 的距離
    P_rx_dBm(i) = P_tx_dBm + friis_equation(freq, 10^(G_tx(i)/10), 1, distance); % 計算 Rx Power
    % P_rx_dBm(i) = P_tx_dBm + friis_equation(freq, G_tx(i), 1, distance); % 計算 Rx Power
    SNR_dB(i) = P_rx_dBm(i) - N0_dBm; % 計算 SNR
    printf(sprintf('User %d Rx Power: %.2f dBm, SNR: %.2f dB\n', i, P_rx_dBm(i), SNR_dB(i)));
end
```

About the 4th commit, I add task3. In order to get the SINR, I (interfere power) is needed. In this version, I calculated I by $I_{(dBm)}(rx1) = P_{tx(dBm)} + friis(freq, G_{tx(Watt)}(rx2), G_{rx} = 1, distance(tx,\ rx1))$ and the SINR by $SINR_{(dB)}(rx1) = P_{(dBm)}(rx1) - I_{(dBm)}(rx1) - N_{0(dBm)}$.

Here we go to the 5th commit! In this version, I modified task 2 (Rx power and SNR) after referencing to CWN's Lab2. And I found out that G_{tx} should be calculated in the following way!

```
a(tx_beam_idx, :) = uniform(d, ph0, tx_antenna_number);
A(tx_beam_idx, :) = dtft(a(tx_beam_idx, :), -psi);
Gain(tx_beam_idx, :) = abs(A(tx_beam_idx, :)).^2;
```

, where psi =

```
phi = (1 : resolution) * pi / resolution;
phi_d = phi * 180 / pi;
psi = 2 * pi * d * cos(phi);
```

By this way, `dtft()` is applied for computing the array factor! Furthermore, I calculate the rx angle by and redirect the angle of $AoD(rx\_idx) >= 179.5° \,||\, AoD(rx\_idx) <= -90°$ to 180° and $(AoD(rx\_idx) < 0 \,\&\&\, AoD(rx\_idx) > -90)$ to 0°. Later on, I computed the G_{tx} by getting the best(max) gain of that specific sector(direction) from gain table. And then, I can therefore calculate the SNR by $rx_{SNR_{dB(1)}} = P_{tx_{dBm}} - friis\_equation(freq, tx\_Gain, 1, rx\_dist) - N0\_dBm;$

In the 6th commit, I found that $SNR = P_{tx_{dBm}} + friis(freq, G_{tx}, 1, Dist_{tx,rx}) - N_{0_{dBm}}$. And also I add $INR = P_{tx_{dBm}} + friis(freq, G_{tx}, 1, Dist_{tx,rx}) - N_{0_{dBm}}$, which's exactly the same way as calculating SNR? However, I didn't notice it at that moment…

In the 7th commit, I think that the calculation of task2 and 3 were wrong. So I just simply comment task 2 and remove task3 for getting back original version and finding more resources online at first. And in the 8th commit, I only modify the print out format, others remain unchanged.

About the 9th commit, I remove the debug version of task2 to shorten the code. Also I modified the print out format to match the requirement in spec. Furthermore, I added task3 again by calculating Rx1's interference power by $P\_tx\_dBm - friis\_equation(freq, G\_tx(2), 1, rx\_Dists(2))$ and Rx1's SINR by $SINR_{rx1} = P\_tx\_dBm + friis\_equation(freq, G\_tx(2), 1, rx\_Dists(2)) - N0\_dBm)$. NOTE: I carefulness modified tx_beam_direction = 0:10:180 in TA section, and I've reverted it back in the future commit!

About the 10th commit, I focus on supporting the report requirements for plotting figures. Function `plot_tx_gain(Gain, phi_d, user_id)` is for plotting Cartesian figures for tx_gain. And function `plot_tx_gain_polar(Gain, phi, user_id)` is for plotting Polar figures for tx_gain. Furthermore, function `plot_optimal_beam(rx_location)` is for plotting the comparison figures for multiple Tx antenna setting. And since we also need to print out many information about the comparison. I just grabbed all the code from task1 to task3 into the function and make little modifications for storing information for plotting graph.

In the 11th commit, (after going to class) the resolution is set to 0.5:0.5:180, and by the announcement of TA: *"user1 的 SINR 是 beam1 到 user1 的 Prx 除以 (beam2 到 user1 的 interference power + noise power)"*, I recalculate the SINR by the (maybe)right formula. And since there're some more information needed as input for computing SINR and the interference power, I added some arrays like G_interference, INR_dB, interference_Beam_idxs. So, in this version, the Rx1 interference power is calculated based on $P\_tx\_dBm + friis\_equation(freq, G\_interference(1), 1, rx\_Dists(1))$, but the G_interference(1) here, means the not wanted (interference) power of rx2 while shotting beam toward rx1. And the $SINR = P_{rx_{dBm}} - I_{dBm} - N_{0_{dBm}}$, which is still incorrect…

In the 12th commit, I noticed that I modified the TA section, so I reverted it back. And about the Gain calculation, even though the TA said the output array of `uniform()` should $/= tx\_antennas\_num$, it made the value of gain unsensible. So I further go the find more resources and I apply the $/=$ operation after `dtft()`. Also I found that the previous calculation of Rx1 interference power and SINR had mistake (in 11th commit). That's I misunderstand the meaning of "interference"; therefore, I changed $G\_interference(1)$ to $G\_interference(2)$ for showing the not wanted power of rx1! And I should convert the unit from dBm to Watt and then back to dBm again while computing SINR! Upon here, these are the overall modifications about the task1~3. And in this commit, I also add the 20_random_run to support the requirement of the report.

For the commit after the 12th commit, it's mainly about modifying the output format.

<Update in 3/10 commitment>: Due to the latest announcements from teacher, the output array from `uniform`() don't need to apply the normalization! So I remove the $/= tx\_antennas\_num$ operation.