

# Lab 5 - Wireless Networking Technologies

## Analyzing a LEO Satellite Pass

by Timo Niemann

### Reproducibility

'-' means use the last non '-' left value

<b>Reproducibility factor</b>	<b>Task 1 value</b>	<b>Task 2 value</b>	<b>Task 3 value</b>	<b>Task 4.1 - 4.3 value</b>	<b>Task 4.4 value</b>	<b>Task 4.5 - 4.6 value</b>
MATLAB Version	26.1.0.3251617 (R2026a) Update 2	-	-	-	-	-
5G Toolbox Version	26.1	-	-	-	-	-
WLAN Toolbox Version	26.1	-	-	-	-	-
Communications Toolbox Version	26.1	-	-	-	-	-
Parallel Computing Toolbox Version	26.1	-	-	-	-	-
Satellite scenario start time	2026-06-17 08:00:00	-	-	-	-	-
Satellite scenario duration	2 h	-	-	10 min	-	-
Satellite scenario sample time	30 s	-	-	-	-	-
Ground station	52.515744683039436 deg lat, 13.325825073341306 deg lon, 37 m altitude	-	-	-	-	-
Orbit propagator	two-body-keplerian	-	-	-	-	-
LEO altitude	600 km	-	-	-	-	-
Eccentricity	0	-	-	-	-	-
Inclination	90 deg	-	-	-	-	-

Reproducibility factor	Task 1 value	Task 2 value	Task 3 value	Task 4.1 - 4.3 value	Task 4.4 value	Task 4.5 - 4.6 value
RAAN	42 deg	-	-	-	-	-
Argument of periapsis	32 deg	-	-	-	-	-
True anomaly	0 deg	-	-	-	-	-
Random stream	mt19937ar with seed	-	-	-	-	-
Random seed	666	-	-	-	-	-
NTN profile	not used	NTN-TDL-C	-	-	-	-
Delay spread	not used	30 ns	-	-	-	-
Carrier frequency	2.4 GHz	-	-	-	[2.4 5 6 10 20 40 45 50 55 60] GHz	-
Channel bandwidth	not used	20 MHz	-	-	-	-
MCS	not used	3	-	-	-	-
APEP length	not used	1000 B	-	-	-	-
TX/RX antennas	not used	1-by-1 SISO	-	-	-	-
Transmit power	not used	20 dBm	-	-	-	-
Free-space path loss	not used	enabled	-	disabled	-	-
WiFi sensitivity threshold	not used	-85 dBm	-	not used	-	-
Reliable receive threshold	not used	-75 dBm	-	not used	-	-
Amplifier gain assumption	not used	30 dB	-	not used	-	-
SNR vector	not used	not used	not used	0 : 1 : 40 dB	-	-

Reproducibility factor	Task 1 value	Task 2 value	Task 3 value	Task 4.1 - 4.3 value	Task 4.4 value	Task 4.5 - 4.6 value
Packet budget	one access-analysis sample every 30 s	power-only pass over all access-analysis samples	reused from Task 2	1000 packets per SNR point	-	no additional simulation
Packet timing	full 2 h scenario grid	-	-	1000 uniformly spaced packets over 10 min	-	-
Pre-compensation cases	not used	not used	not used	none, ideal, 10 Hz, 100 Hz, 1000 Hz, 5000 Hz residual CFO	none, ideal	uses Task 4.1 - 4.4 results
Pilot tracking	not used	default receiver behavior	-	disabled	-	-
Receiver CFO correction	not used	coarse and fine preamble-based CFO correction active	-	-	-	-

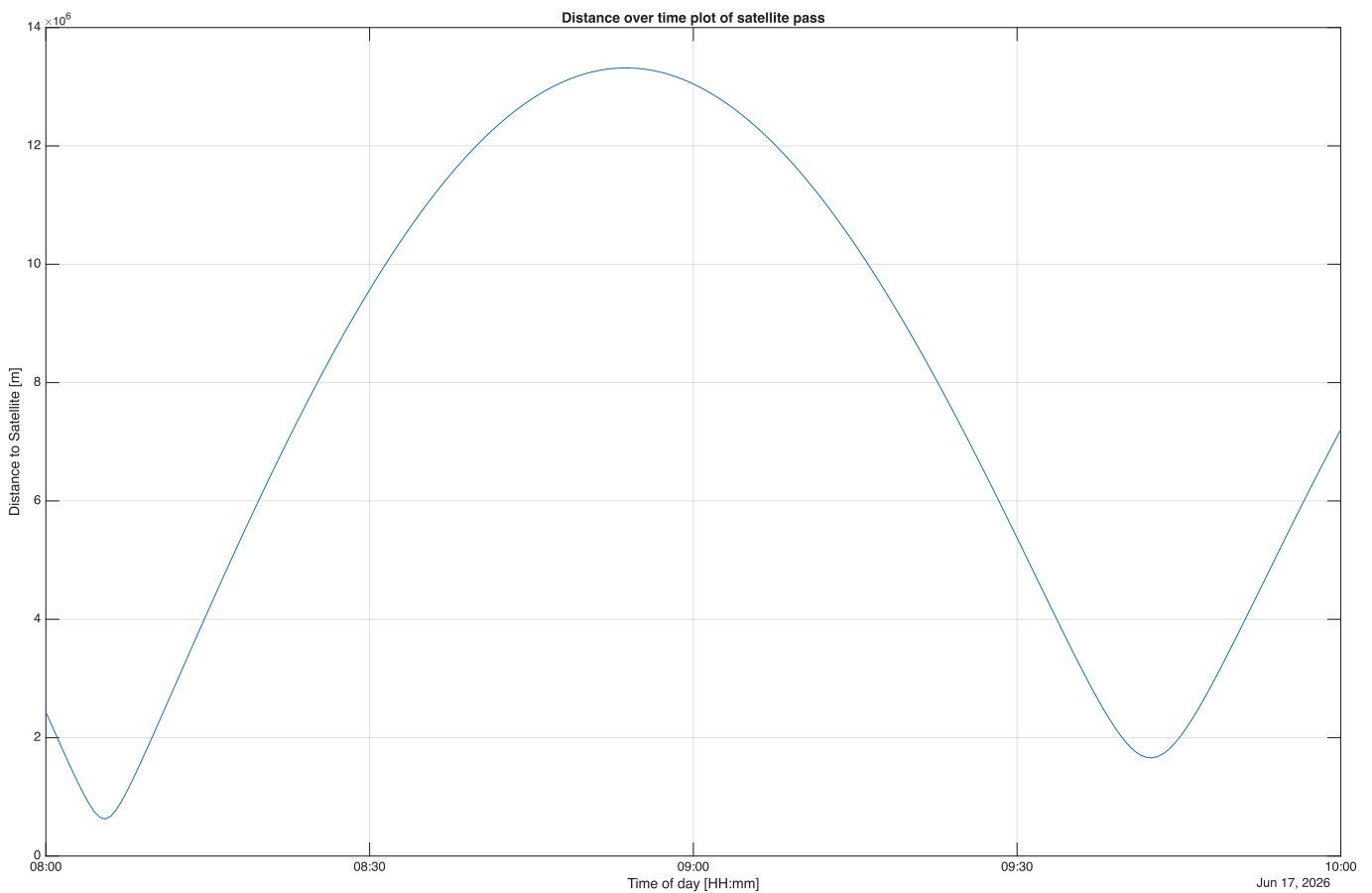
# Task1

## 1.1

See code `lab_5.m` and `create_satellite_sim.m`, where the first file defines all parameters for the simulation and the `create_satellite_sim` function creates and returns the satellite and ground station.

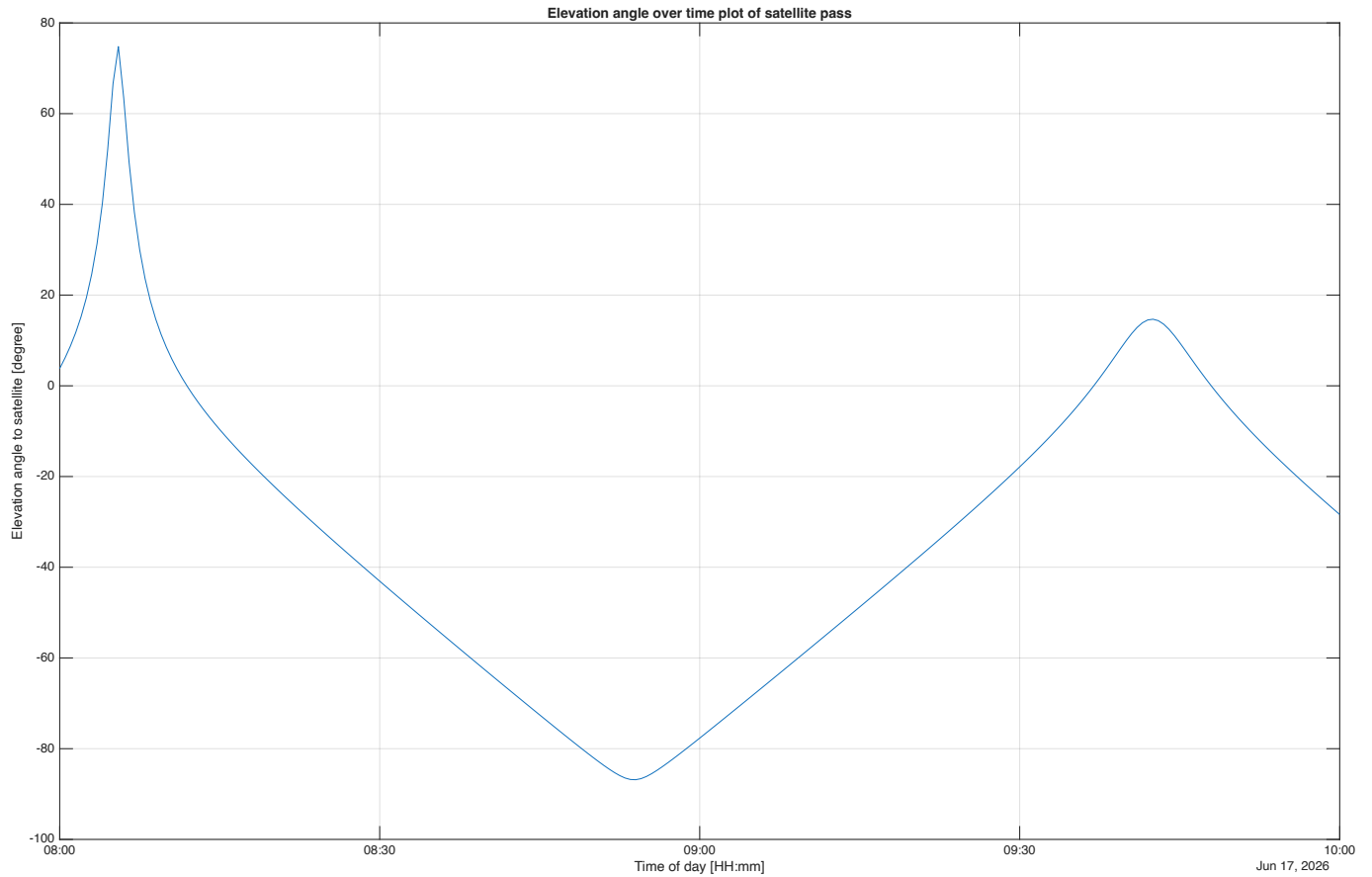
## 1.2

To calculate the distance to the satellite, the MATLAB access analysis example was used as a basis and modified. The distance is calculated with the `aer` function, which returns azimuth, elevation, and slant range between the ground station and the satellite. To avoid invalid channel simulations, the signal-based estimation step is skipped when the Doppler shift `fshift` is non-finite.



The first satellite flyby occurs at around 08:05:30. During the pass, the distance first decreases as the satellite approaches the ground station and then increases again after the closest approach. Around 08:53:30 to 08:54:00, the distance reaches a maximum because the satellite is on the far side of its orbit relative to the ground station. The second pass occurs at around 09:42. Its minimum distance is larger than during the first pass because Earth's rotation changes the relative position of the ground station below the satellite orbit.

### 1.3



As in the distance plot, the elevation angle reaches its highest value during the first pass. The second pass has a lower maximum elevation because Earth's rotation shifts the ground station relative to the satellite ground track.

## Task2

### 2.1

The normalization calculation is done by firstly extracting the **Short Training Field** from the generated WLAN Waveform:

```
tx = wlanWaveformGenerator(txPSDU, cfgEHT);  
stf = tx(ind.LSTF(1):ind.LSTF(2), :);
```

On the extracted STF the mean of the per value absolute potentiation by 2 is calculated to get the power of the STF:

```
stfPower = mean(abs(stf(:)).^2);
```

Following [wikipedia](#) to convert Watt (P) into dBm (x) following formula is used:

$$x = 10 \cdot \log_{10} \frac{P}{1\text{mW}}$$
$$= 10 \cdot \log_{10}(P \cdot 1000)$$

To simplify that:

$$= 10 \cdot (\log_{10}(P) + \log_{10}(1000))$$
$$= 10 \cdot \log_{10}(P) + 10 \cdot \log_{10}(1000)$$
$$= 10 \cdot \log_{10}(P) + 10 \cdot 3$$
$$= 10 \cdot \log_{10}(P) + 30$$

After this the formula is solved for P, because the MATLAB signal has to be scaled with a linear power value in Watt:

$$x = 10 \cdot \log_{10}(P) + 30$$
$$x - 30 = 10 \cdot \log_{10}(P)$$
$$\frac{x-30}{10} = \log_{10}(P)$$
$$10^{\frac{x-30}{10}} = P$$

The  $10^{\wedge}$  is used because it is the inverse function of **log10**. The division by 10 is needed because the logarithm was multiplied by 10 in the dBm formula. For example, 20 dBm results in:

$$P = 10^{\frac{20-30}{10}} = 10^{-1} = 0.1\text{W}$$

The waveform itself is scaled by amplitude and not directly by power. Since power is proportional to the squared amplitude, the scaling factor has to use the square root. Therefore the target power in Watt is divided by the measured STF power and the square root is applied:

```
tx = tx * sqrt(10^((txPower - 30) / 10)) / sqrt(stfPower);
```

With this calculation the generated waveform is normalized by the STF power and then set to the desired transmit power in dBm.

## 2.2

To include free space path loss the corresponding field in `create_baseline_configuration.m:37` is set.

Following [wikipedia](#)'s free space path loss in decibels, the formula is:

$$FSPL(dB) = 20 \cdot \log_{10}\left(\frac{4\pi d}{\lambda}\right), \text{ where } \lambda = \frac{c}{f}$$

Therefore the wavelength is calculated from the carrier frequency:

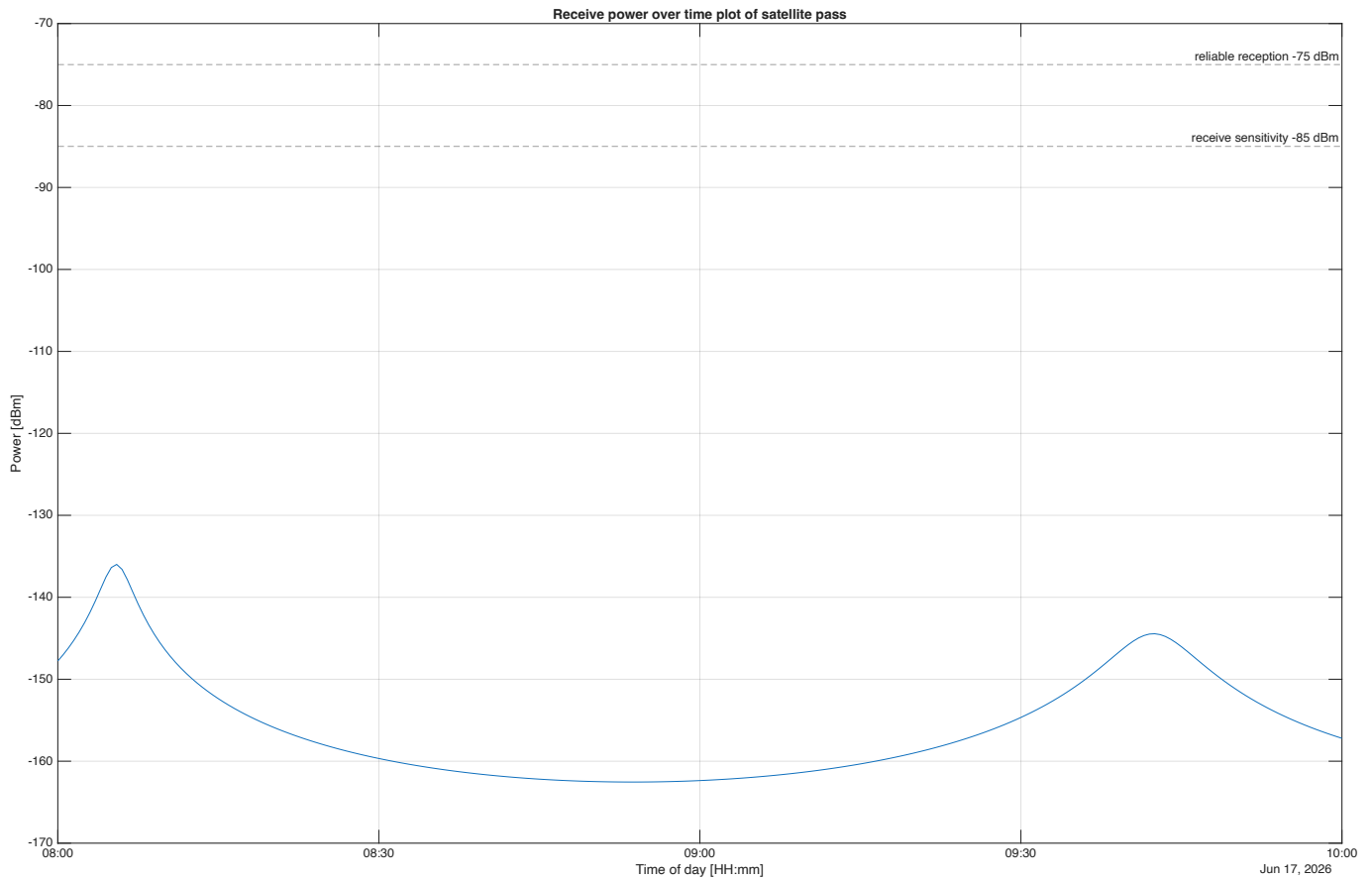
```
lambda = physconst("lightspeed") / carrier_frequency;
```

The received power is then the transmit power minus the free space path loss:

$$P_{rx} = P_{tx} - FSPL$$

This is possible because `P_tx` and `P_rx` are dBm values, while `FSPL` is a dB loss. In the simulation this is implemented as:

```
pathLoss(numPacketsSimulated) = 20 * log10(4 * pi * packetDistance /  
lambda);  
rxPower(numPacketsSimulated) = txPower - pathLoss(numPacketsSimulated);
```



### 2.3

[calculatorshub.net](https://calculatorshub.net) - this site provided helpful formulas and verification possibilities.

A signal is detectable when the receive power has at least a sensitivity of -85 dBm:

$$P_{rx} \geq -85dBm$$

The simulated receive power is below this threshold during the whole satellite pass. With the normal 20dBm WiFi transmit power, the received signal is therefore not detectable by a typical WiFi node.

To calculate the maximum reachable distance for detectability, the free space path loss formula is used in the opposite direction. With a transmit power of 20 dBm, the maximum allowed path loss is:

$$FSPL_{max} = P_{tx} - P_{rx}$$

$$FSPL_{detectable} = 20dBm - (-85dBm) = 105dB$$

The free space path loss formula is then solved for the distance  $d$ :

$$FSPL = 20 \cdot \log_{10}\left(\frac{4\pi d}{\lambda}\right)$$

$$\frac{FSPL}{20} = \log_{10}\left(\frac{4\pi d}{\lambda}\right)$$

$$10^{\frac{FSPL}{20}} = \frac{4\pi d}{\lambda}$$

$$d = \frac{\lambda}{4\pi} \cdot 10^{\frac{FSPL}{20}}$$

This is why the code calculates:

```
maxPathLossDetectable = txPower - wifiSensitivity;
maxDistanceDetectable = lambda / (4 * pi) * 10^(maxPathLossDetectable / 20);
```

The calculation results in a maximum detectable distance of 1.77km with 20dBm transmit power. This is much smaller than the satellite distance during the pass, so a normal WiFi transmit power is not sufficient for this link.

## 2.4

To calculate the required transmit power for reliable reception, the receive power formula is solved for  $P_{tx}$  instead:

$$P_{rx} = P_{tx} - FSPL$$

$$P_{tx} = P_{rx} + FSPL$$

For reliable WiFi reception the target receive power is -75 dBm. Therefore the required transmit power is:

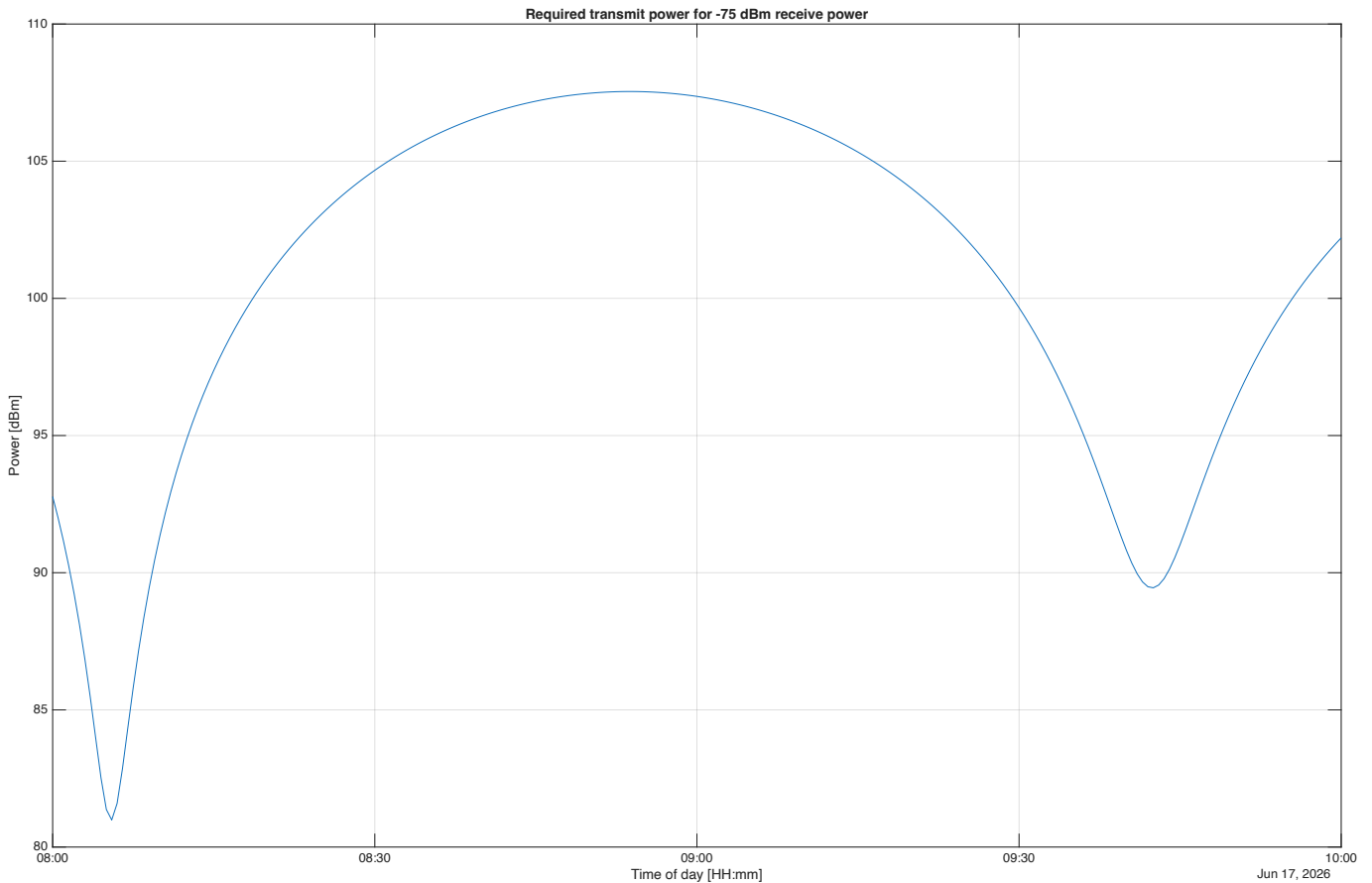
$$P_{tx,required} = -75dBm + FSPL$$

With normal 20dBm transmit power, this reliable reception threshold would only allow a maximum distance of 0.56km:

```
maxPathLossReliable = txPower - reliableRxPower;
maxDistanceReliable = lambda / (4 * pi) * 10^(maxPathLossReliable / 20);
```

In MATLAB this is:

```
requiredTxPower = reliableRxPower + pathLoss;
```



For the simulated satellite pass, the required transmit power for -75dBm receive power is between 80.99dBm and 107.54dBm. This is far above the normal 20dBm WiFi transmit power.

## 2.5

The transmit amplifier can add 30 dB gain. Together with the original 20 dBm transmit power, the amplified transmit power is:

$$P_{tx,amplified} = 20dBm + 30dB = 50dBm$$

If the required transmit power is still higher than this value, the missing difference has to be provided as antenna gain:

$$G_{antenna} = P_{tx,required} - P_{tx,amplified}$$

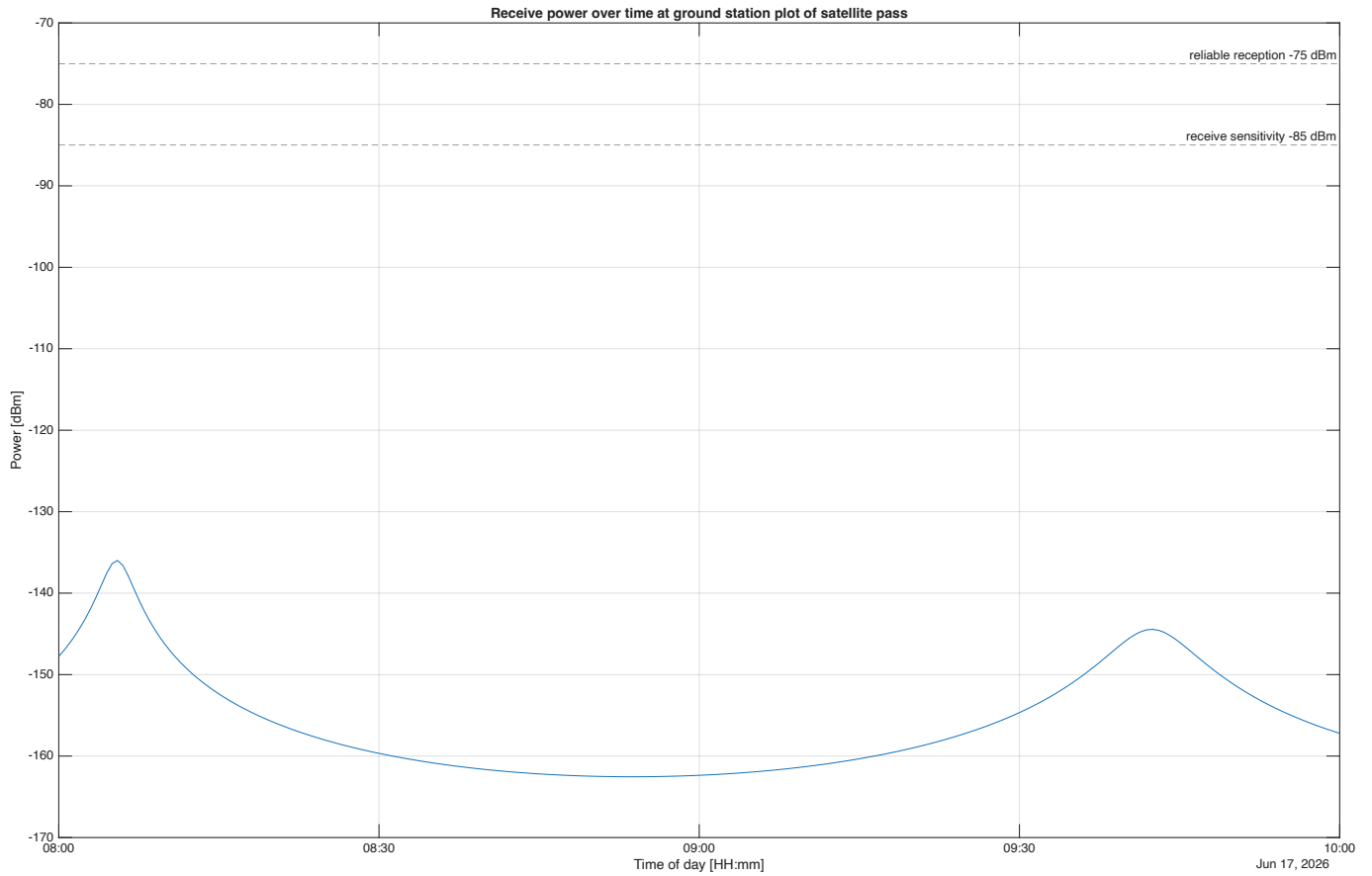
Negative antenna gain is not needed, therefore the value is limited to zero:

```
amplifiedTxPower = txPower + amplifierGain;
requiredAntennaGain = max(requiredTxPower - amplifiedTxPower, 0);
```

With the 30dB amplifier, the remaining required antenna gain is between 30.99dB and 57.54dB.

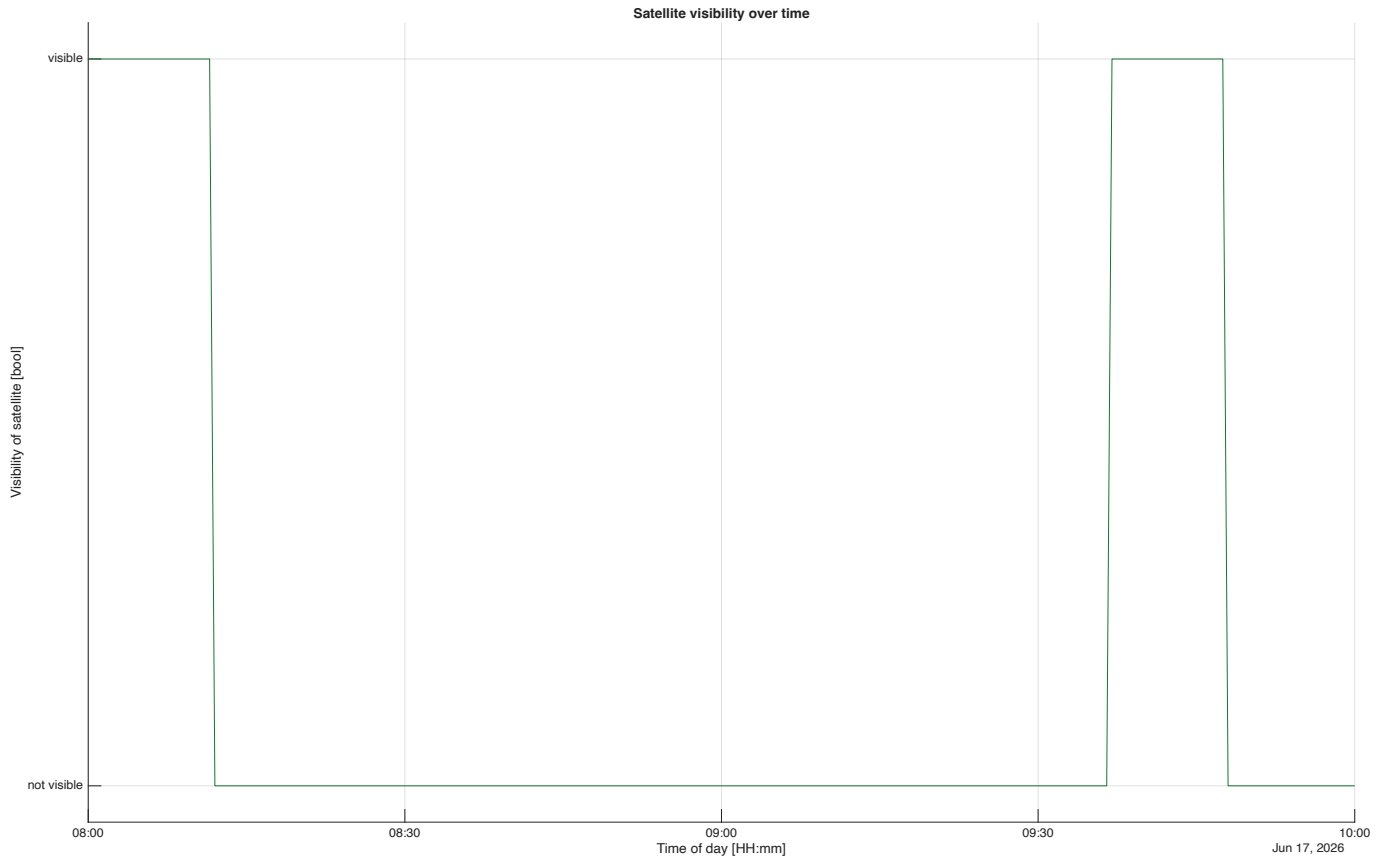
# Task3

## 3.1



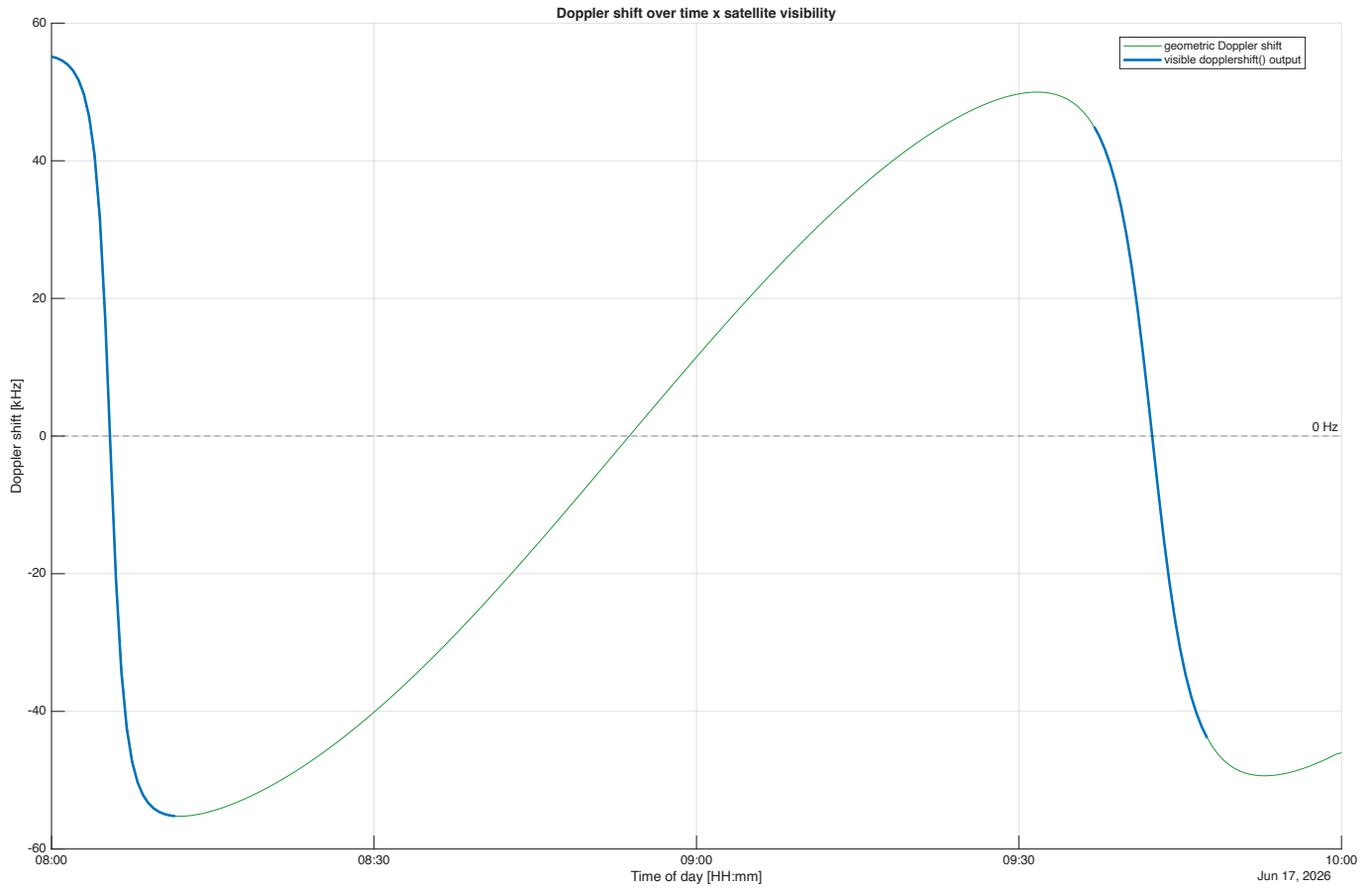
Using the previously set transmit power of 20 dBm, the graph is identical to the receive power plot from task 2.2, because the receiver is the ground station. The received power remains below both the -85 dBm detection threshold and the -75 dBm reliable reception threshold. Therefore, WiFi packet reception is not possible during the simulated pass with normal WiFi transmit power.

## 3.2



The visibility was extracted from the access analysis. MATLAB's `dopplershift` function returns no finite value when the satellite is not visible to the ground station. The plot shows that the satellite is visible from about 08:00 to 08:12 and again from about 09:37 to 09:48. The ground station therefore has a total satellite visibility of about 23 minutes during the two-hour simulation.

### 3.3

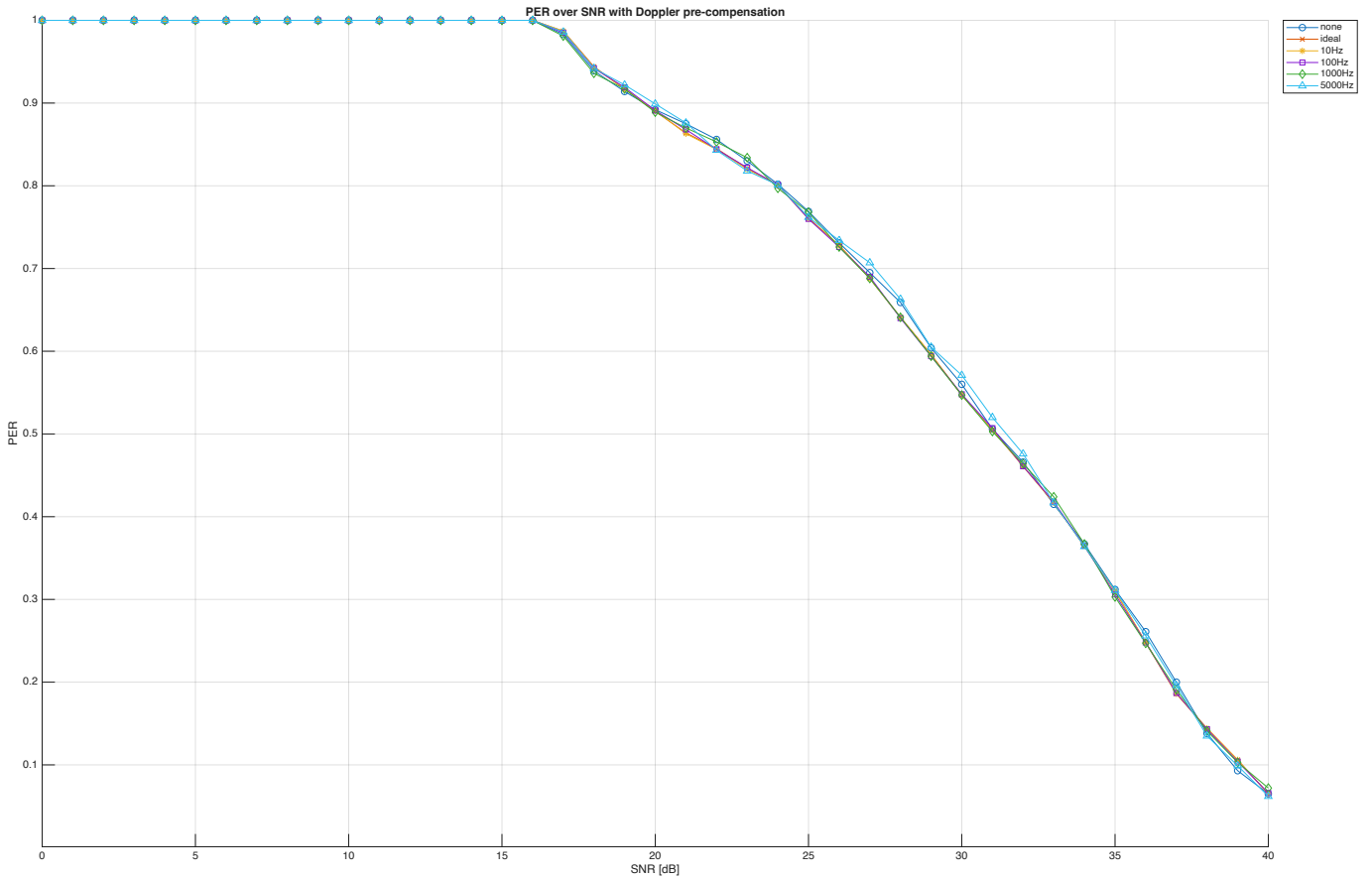


The blue line shows the Doppler shift returned by MATLAB's `dopplershift` function during visible satellite periods. The green line shows a geometric Doppler estimate calculated from the change rate of the satellite distance, the carrier frequency, and the speed of light. The Doppler shift changes sign around the closest approach, because the satellite changes from approaching the ground station to moving away from it.

In this scenario, Doppler shift can support localization because it constrains the relative radial motion between satellite and ground station. However, Doppler alone is not sufficient for unique localization: the satellite orbit, time, and additional measurements such as range or elevation are still required.

# Task4

## 4.1 - 4.3

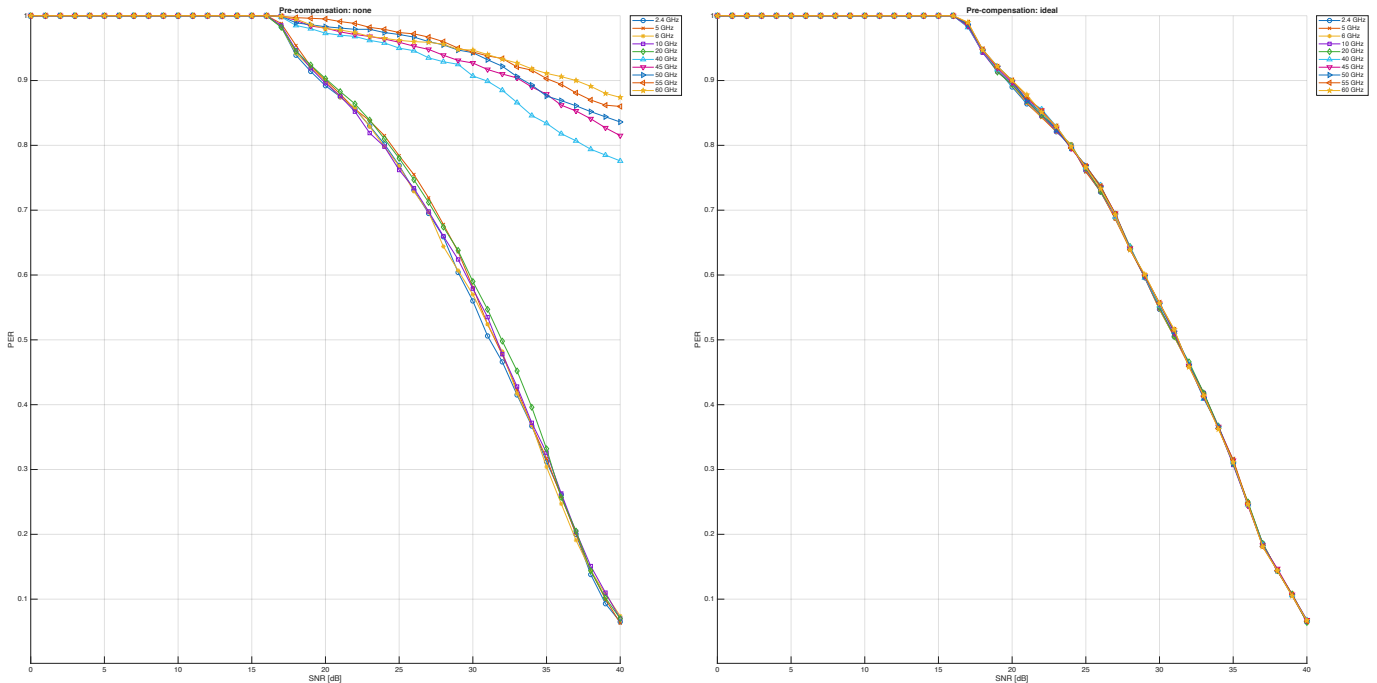


The plot compares the PER over SNR without Doppler pre-compensation, with ideal pre-compensation, and with imperfect pre-compensation errors of 10 Hz, 100 Hz, 1000 Hz, and 5000 Hz.

Only small differences between the curves are visible. The maximum PER difference between no pre-compensation and ideal pre-compensation is 0.018, which corresponds to about 18 packets out of 1000. Therefore, Doppler pre-compensation has only a small visible effect in this simulation setup.

A likely reason is that the 802.11 receiver still performs coarse and fine CFO correction using the WiFi preamble. Pilot tracking is disabled as required by the task, but the preamble-based CFO correction is still active. Because of this, even the case without transmitter-side pre-compensation can still decode many packets successfully. The additional benefit of ideal pre-compensation is therefore limited at the simulated 2.4 GHz carrier frequency.

#### 4.4



For this compare only the test with ideal pre-compensation and without pre-compensation is run. The simulated frequencies are 2.4GHz, 5GHz, 6GHz, 10GHz, 20GHz, 40GHz, 45GHz, 50GHz, 55GHz and 60GHz. For the simulation without pre-compensation the maximum carrier frequency that does not degrade in quality in contrast to the optimal is 20GHz, the next tested 40GHz frequency resulted in only reaching below 0.8 PER in the simulated SNR range instead of well below 0.1 PER. With pre-compensation the simulation did not reach cutoff, all testet frequencies performt well enough to be used as carrier frequencies for WiFi.

#### 4.5

The PER curves from 4.1 - 4.3 show that residual CFOs up to 5kHz do not cause a strong degradation. The simulated 802.11be receiver can tolerate at least about 5 kHz residual CFO after Doppler pre-compensation. This corresponds to 6.4% of the 78.125kHz subcarrier spacing of a 20MHz EHT waveform. The reason is that pilot tracking is disabled, but the receiver still performs coarse CFO estimation using the preamble. Therefore, transmitter-side pre-compensation does not need to be perfect. Errors in the low-kHz range are still corrected well enough in this simulation.

#### 4.6

Following the mentions in this [paper]: Required information: carrier frequency, relative radial velocity along the line-of-sight, positions and velocities of transmitter and receiver, timing, and separation of Doppler from oscillator frequency offset.

How obtained: The required information can be obtained from satellite ephemeris (data collection of calculated positions of a celestial object at regular intervals in this case the satellite) or TLE data, known ground-station coordinates or receiver GNSS feedback, and a synchronized time reference. Using these values together with the carrier frequency, the transmitter can estimate the relative range rate and calculate the Doppler shift before transmission.

## **Special Thanks**

This lab was solved with contribution of GPT 5.5.