Opracowanie modelu obliczeniowego metody Har-Xia-Bertoni

Karol Slomczyński 272223

4 kwietnia 2024

1 Wstęp

Do wykonania projektu użyłem Python'a w wersji 3.10.12 oraz biblioteki math.

2 Opracowanie kodu

2.1 Zabudowa niska

Jak możemy zauważyć w kodzie, metoda Har-Xia-Bertoni jest zaimplementowana w postaci funkcji zależnie od specyfikacji zabudowy i typu trasy Poniżej znajduje się kod funkcji dla zabudowy niskiej i tras schodkowych oraz poprzecznych:

```
    def low(f_G: float , R_k: float, r_h: float, h_t: float, h_bd:float) -> float:
    d_h:float = h_t - h_bd
    path_loss = (139.01 + 42.59*math.log10(f_G))\
                -(14.97 + 4.99*math.log10(f_G))*abs(d_h)*math.log10(1+abs(d_h))\
                + (40.67 - 4.57*abs(d_h)*math.log10(1+abs(d_h)))*math.log10(R_k)
    return path_loss
```

Po analize udostępnionego dokumentu zauważyłem, że w przypadku zabudowy niskiej typ trasy w przypadku trasy schodkowej i poprzecznej jest taka sama, dlatego zdecydowałem się na zaimplementowanie jednej funkcji dla obu przypadków.
Poniżej znajduje się kod funkcji dla zabudowy niskiej i trasy bocznej:

```
    def low_sideways(f_G: float , R_k: float, r_h: float, h_t: float, h_bd:float) -> float:
    d_h:float = h_t - h_bd
    path_loss = (127.39+31.63*math.log10(f_G))\
                -(14.97+4.99*math.log10(f_G))*abs(d_h)*math.log10(1+abs(d_h))\
                +(40.67-4.57*abs(d_h)*math.log10(1+abs(d_h)))*math.log10(R_k)
    return path_loss
```

Wedle dokumentu podane formuły dają poprawne wyniki, gdzie f_G jest w przedziale $(0.9; 2)$ GHz, Δh jest w przedziale $(-8; 6)$m, R_k w przedziale $(0.05; 3)$km
Aby zniwelować błędy związane z obliczeniami zastosowałem współczynnik korekcji dla wysokości budynków, który wynosi:

$$(\Delta P_L)_{\Delta h_m} = 20 \log (\Delta h_m / 7.8)$$

W przypadku rozpatrywanym w dostarczonym dokumencie średnia geometryczna wysokości budynków wynosi 7.8m.
Dokument zakłada że odległości od frontu budynku do środka ulicy wynosi 20 metrów. Przez to współczynnik korekcji wynosi

$$(\Delta P_L)_{r_h} = 10 \log(20/r_h)$$

Funkcja dla wszystkich tras po zastosowaniu korekcji wygląda następująco:

```
    def low(f_G: float , R_k: float, r_h: float, h_t: float, h_bd:float) -> float:
    d_h:float = h_t - h_bd
    path_loss = (139.01 + 42.59*math.log10(f_G))\
                -(14.97 + 4.99*math.log10(f_G))*abs(d_h)*math.log10(1+abs(d_h))\
```

```
                + (40.67 - 4.57*abs(d_h)*math.log10(1+abs(d_h)))*math.log10(R_k)\
                + 20*math.log10(d_h/7.8)+10*math.log10(20/r_h)
    return path_loss
```

2.2 Zabudowa wysoka

W przypadku zabudowy wysokiej analogicznie do zabudowy niskiej trasy schodkowe i poprzeczne dają zbliżone wyniki więc zastosowałem jedną funkcję dla obu przypadków.

```
    def high(f_G: float , R_k: float,  h_t: float) -> float:
    path_loss = 143.21+29.74*math.log10(f_G)\
                - 0.99*math.log10(h_t)+ (47.23+3.72*math.log10(h_t))*math.log10(R_k)
    return path_loss
```

Jednakże powoduje to zwiększe błędu oszacowania
W przypadku trasy bocznej dla zabudowy wysokiej zastosowałem funkcję:

```
    def high_sideways(f_G: float , R_k: float, h_t: float) -> float:
    path_loss = 135.41+12.49*math.log10(f_G)\
        - 4.99*math.log10(h_t)+ (46.84-2.34*math.log10(h_t))*math.log10(R_k)
    return path_loss
```

2.3 Bezpośrednia widoczność

W tym przypadku zastosowałem jedną funkcję z zastosowaniem poleceń 'if'

```
    def direct(f_G: float, R_k:float , r_h:float , h_t:float, l:float) -> float:
    R_bk = (4*h_t*r_h)/1000*l
    if R_k < R_bk:
        path_loss = 81.14+39.40*math.log10(f_G)\
        - 0.09*math.log10(h_t)+(15.80-5.73*math.log10(h_t))*math.log10(R_k)
        return path_loss
    if R_k > R_bk:
        path_loss = (48.38-32.10*math.log10(R_bk))+45.70*math.log10(f_G)\
        + (25.32-13.90*math.log10(R_bk))*math.log10(h_t)\
        + (32.10+13.90*math.log10(h_t))*math.log10(R_k)\
        +20*math.log10(1.6/r_h)
        return path_loss
```

W przypadku odcinka dalekiego należy zastosować współczynnik korekcji

$$(\Delta P_L)_{h_m} = 20 \log (1.6/h_m)$$

w przypadku uwzględnionym w dokumencie wysokość odniesienia wnosi 1.6 metra