

# Übung 1, Teil 2 | bepohl

Im zweiten Teil der Übung sollen die a priori Empfängeruhrfehler berechnet, ausgeglichen und zu Stationierung des Empfängers verwendet werden.

## Berechnung des a-priori Empfängeruhrfehlers

Es werden mit den Beobachtungen die Linearkombination P3k für jeden Satelliten gebildet:

```
f_0 = 10.23 * 10**6
f_1 = 154 * f_0
f_2 = 120 * f_0
return (f_1**2 * p_1 - f_2**2 * p_2) / (f_1**2 - f_2**2)
```

P3k ist die Distanz zwischen Station und Satellit ohne den Einfluss der Ionosphäre. Um den gekürzten Beobachtungsvektor l\_0 zu berechnen, werden die Korrekturen aus dem ersten Teil der Übung angebracht mit dem Uhrfehler aus dem Clockfile:

```
dist = math.dist(np.array(stat_coord_earth_fixed),
earth_fixed_coords_at_send)

return dist + tropo_delay + relativistic_delay - clock_bias_i * c
```

Um den Uhrfehler möglichst gut zu schätzen, werden verschiedenen a-priori Uhrfehler mit der Mapping-Funktion gewichtet (P-Matrix):

```
np.diag(np.cos(zenit_angle)**2)
```

Mit folgender Ausgleichung wird das Delta\_T\_k für eine Epoche geschätzt:

```
np.linalg.inv(A.T @ P @ A) @ A.T @ P @ (I - I_0)
```

```
with A = np.full((n,1), c), c = speed_of_light
```

DeltaTk\_bias for each epoch:

```
-2.60725722515549e-05
```

```
-2.607239480478024e-05
```

```
-2.6070772278066725e-05
```

```
-2.60707140000212e-05
```

```
-2.6069728000436512e-05
```

## Berechnung der Stationskoordinaten mit neuem Uhrfehler und anschließender Ausgleichung

Mit dem berechneten Uhrfehler können nochmals die Satellitenpositionen mit neuen Signallaufzeiten berechnet werden. Die neu gerechneten Koordinaten für G25 lauten wie folgt:

G25 Corrected Sat Positions:

```
[18619309.34432 -15884993.17540 10299699.57366]
```

```
[18131149.06977 -14641084.79629 12725495.90149]
```

```
[17588720.65841 -13139931.16253 14932097.42395]
```

```
[17031860.24737 -11396433.59762 16881478.97330]
```

```
[16498097.89783 -9433722.69675 18540047.10195]
```

Mit allen Satelliten kann nun nochmals eine Ausgleichung für die Stationskoordinaten mit der neuen Designmatrix:

```
np.array([(X_sat - X_s) / dist, (Y_sat - Y_s) / dist, (Z_sat -  
Z_s) / dist, -1
```

für jede Beobachtung (eg. Satellitenkoordinaten).

Daraus sind die neuen Stationskoordinaten berechnet werde:

```
Corrected station coordinates for each epoch:
array([ 3370657.62334,  711876.25165,  5349785.95224])
array([ 3370658.79049,  711875.77809,  5349787.88527])
array([ 3370658.34442,  711875.87220,  5349787.49884])
array([ 3370660.74713,  711876.03286,  5349787.16751])
array([ 3370657.20009,  711876.10735,  5349785.63052])
```

## Vergleich zwischen berechneten und gegebenen Stationskoordinaten in topozentrischen Koordinaten

Um die Verschiebungen bezüglich den gegebenen Koordinaten darzustellen transformieren wir die berechneten Koordinaten mit der Funktion calculateTopoCoords() in das topozentrische System der gegebenen Station. So können die Verschiebungen einfach vorgestellt werden:

```
Differences for each epoch in North, East, Up coordinates to
given coordinates:
[(0.2782328855576145, -0.782017519303059, -1.4433419595222399),
 (0.44696264559170573, -1.486535156952947, 0.7471603322803547),
 (0.5883360336008537, -1.3022725309985053, 0.19649593971454132),
 (-1.5954571758978595, -1.6415805303571782, 1.2086786301766779),
 (0.4772982662828332, -0.8357374743498829, -1.9541805485606965)]
```

## Interpretation

Die Größenordnung der Uhrfehler und der daraus folgenden Korrektur der Stationskoordinaten sind plausibel. Ich denke aber, dass einige Indexfehler die Berechnung verfälschen und deswegen mit Vorsicht zu geniessen sind.

Für die Interpretation der Ergebnisse ist es zu empfehlen, das Jupyter Notebook zu öffnen (oder das exportierte PDF), und Code und Ergebnisse direkt vergleichen zu können.