

Demonstração da DopplerLib

Bibliotecas e Funções

1. **LibDoppler** é a pasta com as funções usadas para cálculo da velocidade a partir da velocidade doppler, entre elas: `calculate_vel`, `getObservation`, `getObservationTDCP`.
2. **LibGnss** é a pasta com funções associadas a leitura e processamento de dados GNSS.
3. **ExampleCalculatingVel** tem funções auxiliares para carregar os dados de forma direta (`readData` e `selectGpsData`)

```
clear
addpath("LibDoppler\")
addpath("LibGnss\")
addpath("ExampleCalculatingVel\")
```

Dados

Os dados são indexados pela data e estão presentes no caminho `Data/dd_mm_yyyy`

`readData`: É uma função auxiliar responsável por carregar os dados

- `rover.obs` e `rover.nav`
- `ground.obs` e `ground.nav`
- `rover.pos`

`selectGpsData`: Seleciona dados da constelação GPS e agrupa em uma struct chamada de `gpsData`

```
dataPath = "Data/30_07_2021/";
readData;
selectGpsData;

wavelength = 2.99792458e8/1.57542e9;
```

Processar dados

Durante o processamento de dados as velocidades são computadas para cada leitura do GNSS.

Alocando espaço...

```
times = unique(gpsData.roverObs.time);
n = length(times);
V_TDCP = zeros(3, n);
T = zeros(1, n);
res = zeros(1, n);
```

Obtém uma observação específica para o modo **TDCP (Time-Differenced Carrier-Phase)**.

São levados em conta os seguintes critérios:

- Leitura de fase e doppler devem ser diferentes de 0
- Os dados do mesmo satélite devem estar disponíveis em t_{k-1} , t_k e t_{k+1}

Além disso, o desvio doppler no instante k , de cada satélite, é calculado a partir da fórmula

$$d_k = -\frac{\Phi_{k+1} - \Phi_{k-1}}{2\Delta t}$$

sendo Δt o período de medição.

```
for k = 2:n-1
    if k == 2149
        pause
    end
    obs = getObservationTDCP(k, posData, gpsData);
    [V_TDCP(:, k), res(k)] = calculate_vel(obs.recPos, obs.satsPos, obs.satsVel, ...
                                         obs.dopplers', wavelength);

    T(k) = obs.time;

    if mod(k, round(.1*n)) == 0
        fprintf("\b\b\b%.0f%%", k/n*100);
    end
end
```

100%

Plotando Resultados

Para fins de comparação, cálculo as variações das posições xyz do arquivo .pos

```
Vpos = [diff(posData.xecef), diff(posData.yecef), diff(posData.zecef)];
```

Plotando ambos resultados juntos

```
figure
for j = 1:3
    subplot(3, 1, j)
    hold on
    %plot(T(2:end-1), V_RD(2:end-1, j), DisplayName="Raw Doppler", LineWidth=1.5);
    plot(T(2:end-1), V_TDCP(j, 2:end-1), DisplayName="TDCP", LineWidth=1.5);
    plot(posData.time(1:end-1), Vpos(:, j), DisplayName="Derivada da posição (.pos)", LineWidth=1.5);
    legend()
    grid()
end
```

