Assignment 02 - Navigation

Spacecraft Guidance and Navigation

AY 2021-2022

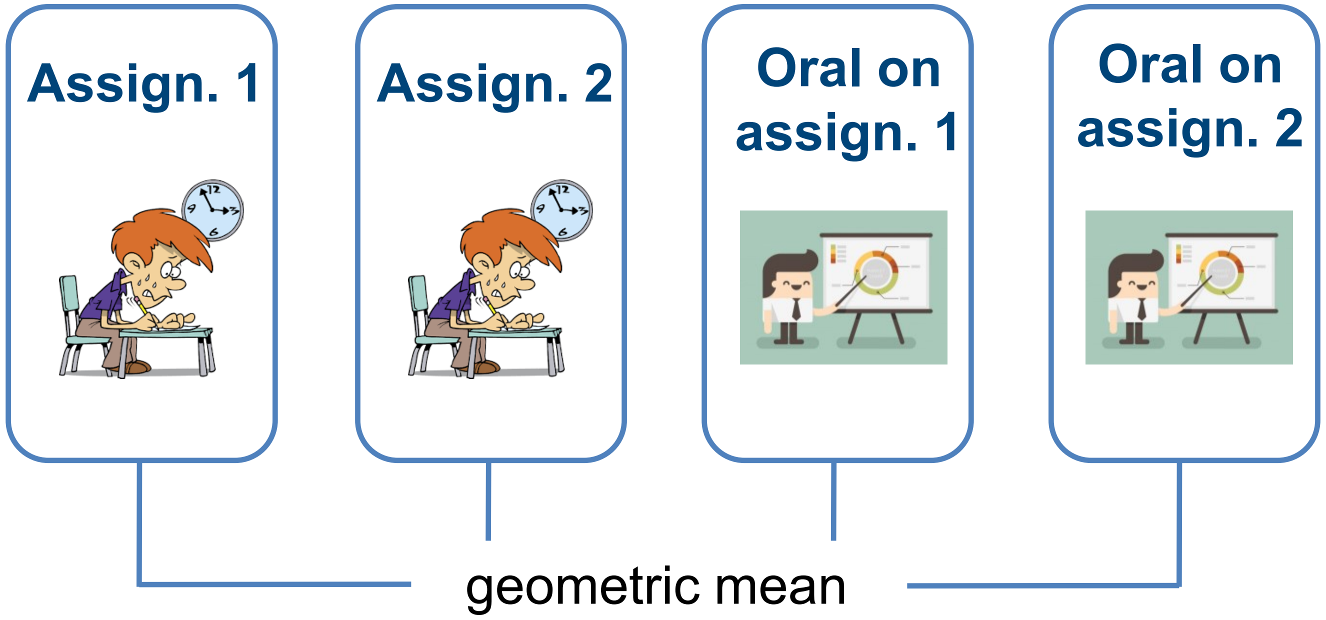
POLITECNICO
MILANO 1863

Exam

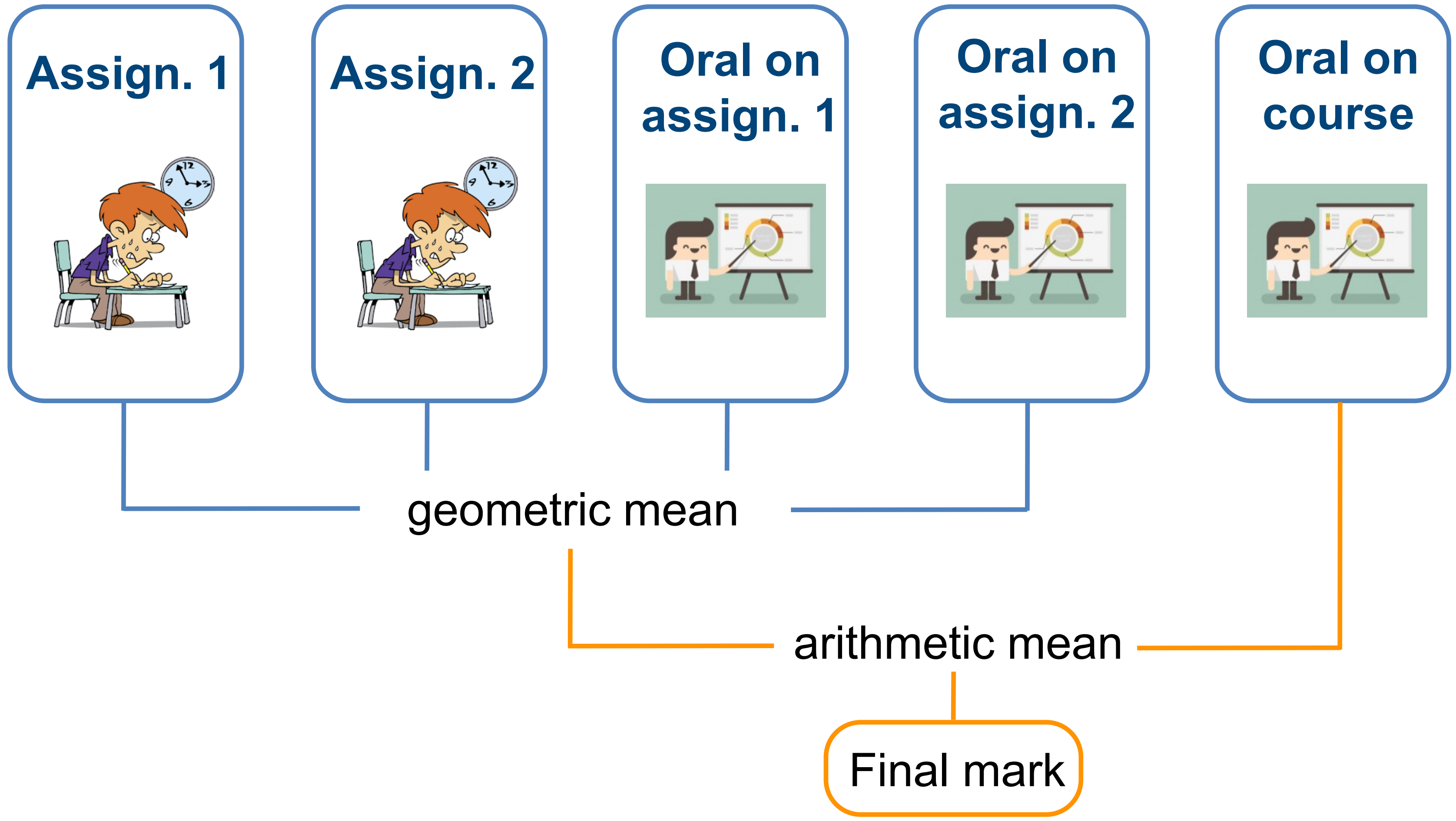
Assign. 1

Assign. 2

Exam


Assign. 1
Assign. 2
Oral on assign. 1
Oral on assign. 2
geometric mean

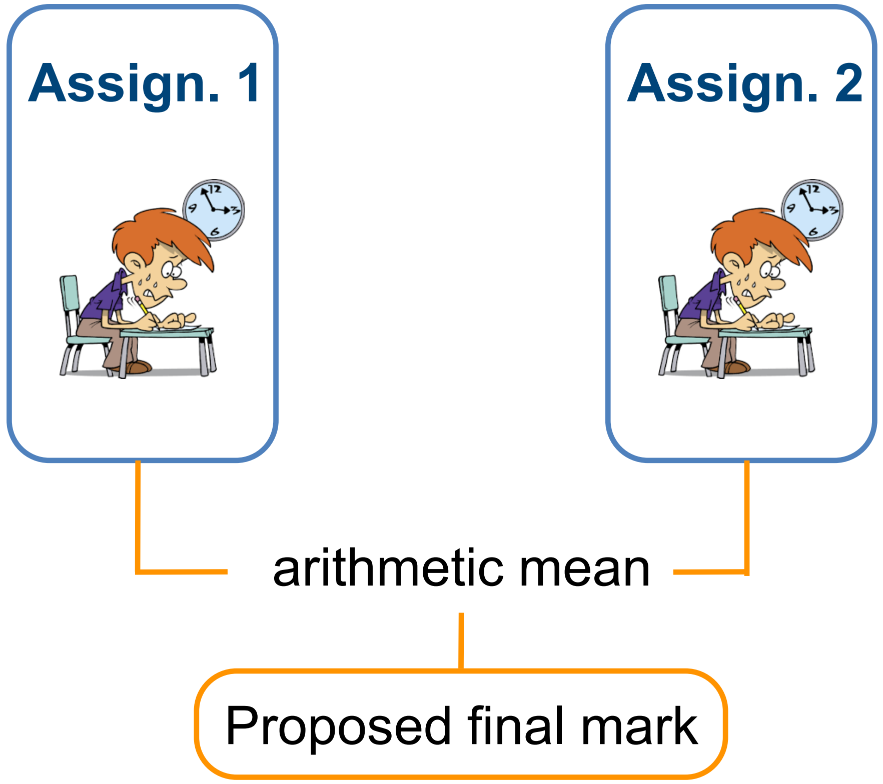
Exam


Assign. 1
Assign. 2
Oral on assign. 1
Oral on assign. 2
Oral on course
geometric mean
arithmetic mean
Final mark

Exam

JUST JOKING!

Exam



- All of you will be asked to come in person during the official exam sessions to check our evaluation and discuss any doubts you/we might have

Exam

Assign. 1

Assign. 2

Oral on course

arithmetic mean

optional

Proposed final mark

- All of you will be asked to come in person during the official exam sessions to check our evaluation and discuss any doubts you/we might have
- In case you are not satisfied with the proposed final mark, you can take an optional oral exam (which will take place during the official exam sessions)

Exam

➢ Remarks:

- For those who will not submit the report by the deadlines, you’ll be asked to do it before the official exam sessions
- Once you receive the evaluation of the assigmments, please wait for the in-person meeting during the exam sessions to discuss about the evaluation

Laboratory sessions

- Laboratory sessions **will not be recorded**
  - we are here to give you answers while you are working

- When writing emails:
  - Use the following object structure:
    [SGN – Assignment2] Your email object here
  - Do **not** attach/send code (or screenshot of the code) asking for debugging

Timetable		Room
24 Nov	16.15-18.15	BL28.12 (+ Morselli virtual room)
26 Nov	10.15-13.15	L11 (+ Morselli virtual room)
30 Nov	14.15-16.15	BL27.14 (+ Morselli virtual room)
01 Dec	16.15-18.15	BL28.12 (+ Morselli virtual room)
03 Dec	10.15-13.15	L11 (+ Morselli virtual room)
16 Dec	**23.30 CET**	**Assignment 2 deadline**

Delivery of assignments

Assigment will be delivered through **WeBeep**:

DEADLINE ⇨ Dec 16, 2021, 23:30:00 CET

1) Click on the link to **load Assignment 2** in your Overleaf

   https://bit.ly/SGN_21_Assignment2

   ⛔ **Not active yet**

2) **Fill the report** and be sure it is compiled properly
3) **Download the PDF** and merge it in a **zipped file** with MATLAB code.
   Rename it `lastname123456_Assign2.zip`

   Do **NOT** put spaces in file names
4) **Submit the compressed file** by uploading it on WeBeep

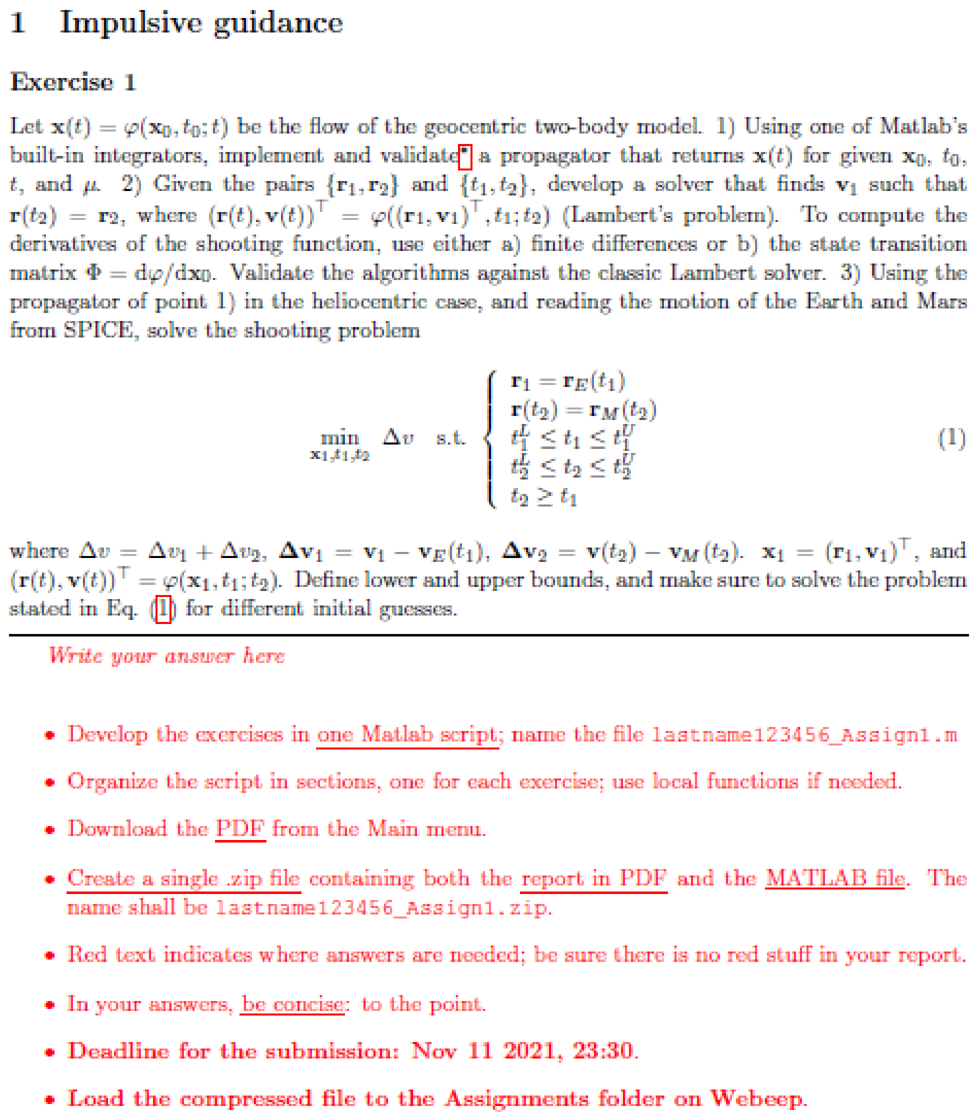
Assignment 02 - Topics

3 Exercises

- Uncertainty propagation
- Batch filters
- Sequential filters

Suggested MATLAB Version: R2021b

Report



1 Impulsive guidance

Exercise 1

Let $\mathbf{x}(t) = \varphi(\mathbf{x}_0, t_0; t)$ be the flow of the geocentric two-body model. 1) Using one of Matlab's built-in integrators, implement and validate* a propagator that returns $\mathbf{x}(t)$ for given $\mathbf{x}_0$, t_0, t, and μ. 2) Given the pairs $\{\mathbf{r}_1, \mathbf{r}_2\}$ and $\{t_1, t_2\}$, develop a solver that finds $\mathbf{v}_1$ such that $\mathbf{r}(t_2) = \mathbf{r}_2$, where $(\mathbf{r}(t), \mathbf{v}(t))^\top = \varphi((\mathbf{r}_1, \mathbf{v}_1)^\top, t_1; t_2)$ (Lambert's problem). To compute the derivatives of the shooting function, use either a) finite differences or b) the state transition matrix $\Phi = \mathrm{d}\varphi/\mathrm{d}\mathbf{x}_0$. Validate the algorithms against the classic Lambert solver. 3) Using the propagator of point 1) in the heliocentric case, and reading the motion of the Earth and Mars from SPICE, solve the shooting problem

$$\min_{\mathbf{x}_1, t_1, t_2} \Delta v \quad \text{s.t.} \begin{cases} \mathbf{r}_1 = \mathbf{r}_E(t_1) \\ \mathbf{r}(t_2) = \mathbf{r}_M(t_2) \\ t_1^L \le t_1 \le t_1^U \\ t_2^L \le t_2 \le t_2^U \\ t_2 \ge t_1 \end{cases} \tag{1}$$

where $\Delta v = \Delta v_1 + \Delta v_2$, $\mathbf{\Delta v}_1 = \mathbf{v}_1 - \mathbf{v}_E(t_1)$, $\mathbf{\Delta v}_2 = \mathbf{v}(t_2) - \mathbf{v}_M(t_2)$. $\mathbf{x}_1 = (\mathbf{r}_1, \mathbf{v}_1)^\top$, and $(\mathbf{r}(t), \mathbf{v}(t))^\top = \varphi(\mathbf{x}_1, t_1; t_2)$. Define lower and upper bounds, and make sure to solve the problem stated in Eq. (1) for different initial guesses.

Write your answer here

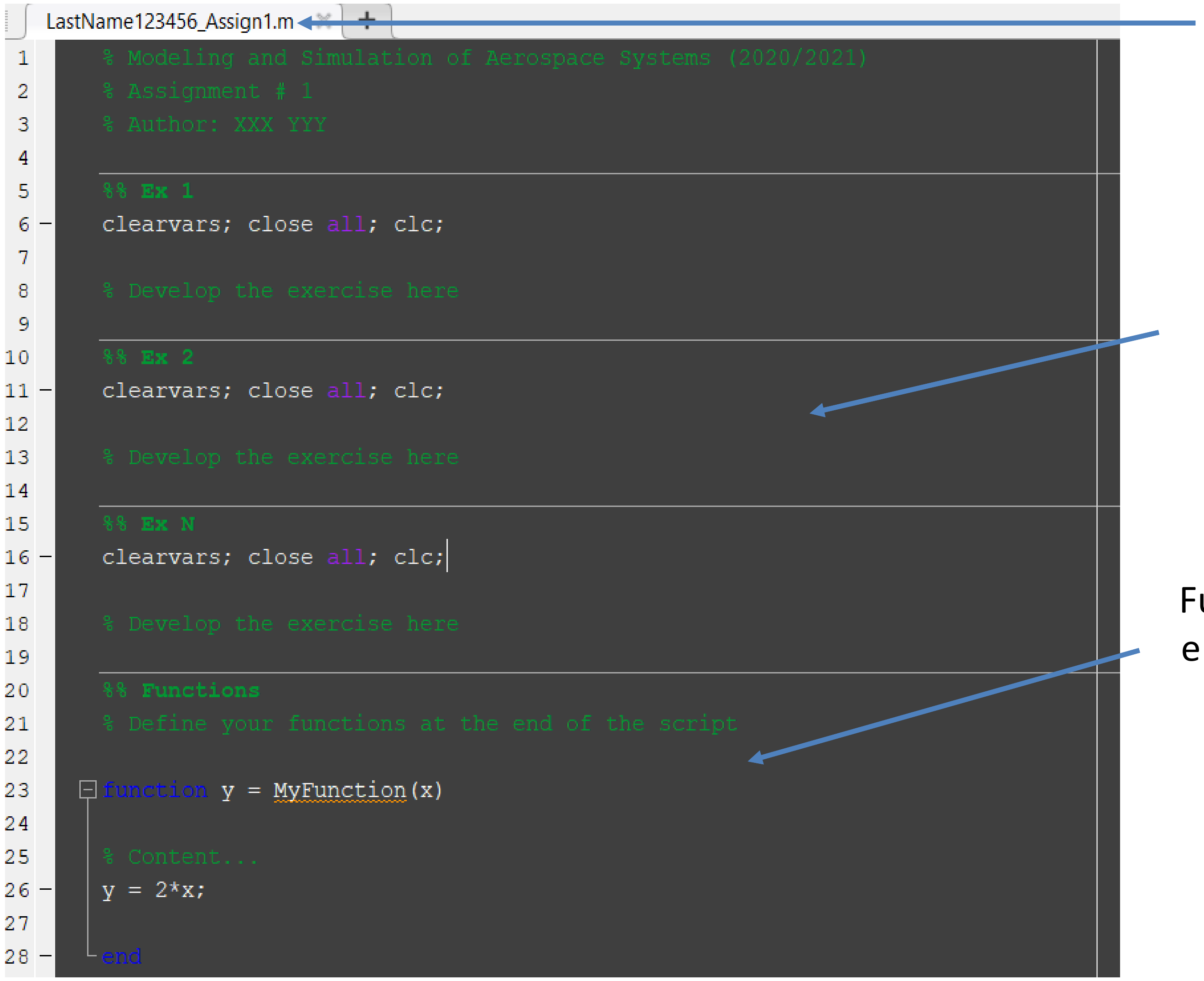
- Develop the exercises in one Matlab script; name the file `lastname123456_Assign1.m`
- Organize the script in sections, one for each exercise; use local functions if needed.
- Download the PDF from the Main menu.
- Create a single .zip file containing both the report in PDF and the MATLAB file. The name shall be `lastname123456_Assign1.zip`.
- Red text indicates where answers are needed; be sure there is no red stuff in your report.
- In your answers, be concise: to the point.
- **Deadline for the submission: Nov 11 2021, 23:30.**
- **Load the compressed file to the Assignments folder on Webeep.**

Exercise

Answer here

POLITECNICO MILANO 1863

Script



```
% Modeling and Simulation of Aerospace Systems (2020/2021)
% Assignment # 1
% Author: XXX YYY

%% Ex 1
clearvars; close all; clc;

% Develop the exercise here

%% Ex 2
clearvars; close all; clc;

% Develop the exercise here

%% Ex N
clearvars; close all; clc;

% Develop the exercise here

%% Functions
% Define your functions at the end of the script

function y = MyFunction(x)

% Content...
y = 2*x;

end
```

Name of the script

One section per exercise

Functions at the end of the script

Exercise 1 – Uncertainty propagation

Exercise 1: Uncertainty propagation

You are the operator of the sensor network composed by the stations reported in Table 1. You have been assigned the task of tracking the spacecraft GIOVE-A (NORAD ID: 28922, INT.DES.: 2005-051A) and you have been provided with an estimate of the spacecraft state at time t_0 in terms of its mean and covariance, as reported in Table 2. Assume Keplerian motion can be used to model the spacecraft dynamics.

1. By using the mean state to predict the reference trajectory over a uniform time grid (one point per minute), compute all the visibility windows from the available stations in the time interval from t_0 to t_f = `2021-11-23T14:00:00` (UTC).

2. Propagate the initial mean and covariance to the last epoch of each visibility window (as computed with the uniform time grid in the previous point) using both a linearized approach and the unscented transform. Elaborate on the results of the uncertainty propagation, including possible differences between the two approaches.

3. Perform a Monte Carlo simulation (using at least 100 samples drawn from the initial covariance) to predict the spacecraft state distribution at the very last visibility epoch (the closest to t_f, as resulting from the uniform time grid used at point 1). Compute the sample mean and sample covariance, and compare them with the results obtained at the previous point. Then, for each sample, compute the angle between the topocentric direction of the satellite and the topocentric direction of the center of the field-of-view (FoV) of the sensor. Provide the percentage of the samples that lie inside the sensor FoV.

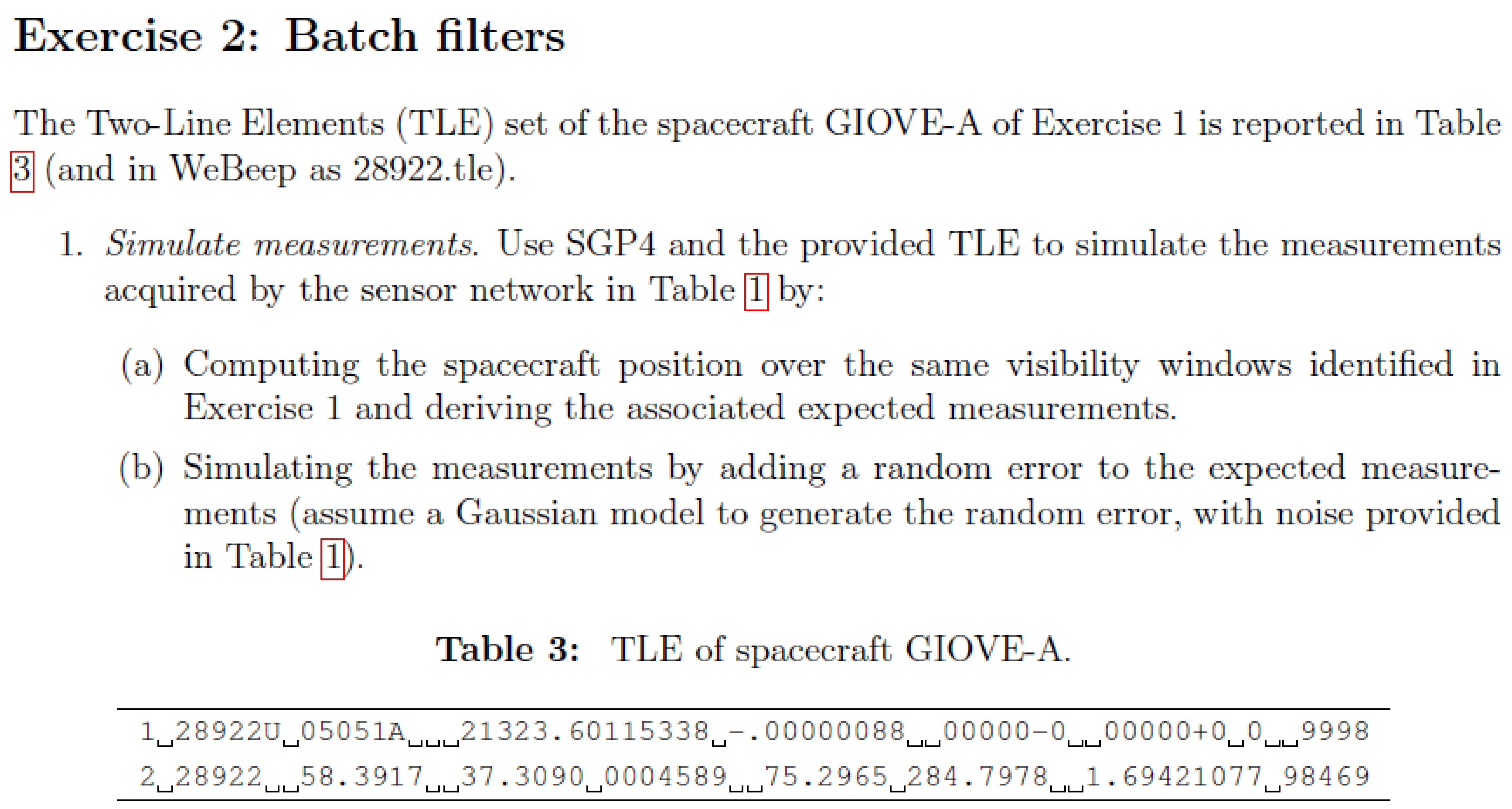
Table 1: Sensor network: list of stations, including their features.

Station name	Milano	Wellington	La Silla
Coordinates	LAT = 45.50122° LON = 9.15461° ALT = 20 m	LAT = -41.28437° LON = 174.76697° ALT = 117 m	LAT = -29.26117° LON = -70.73133° ALT = 2400 m
Type	Radar (monostatic)	Optical	Optical
Provided measurements	Az, El [deg] Range (one-way) [km]	Ra, Dec [deg]	Ra, Dec [deg]
Measurements noise (diagonal noise matrix R)	$\sigma_{Az,El} = 0.1$ deg $\sigma_{range} = 0.01$ km	$\sigma_{Ra,Dec} = 0.0005$	$\sigma_{Ra,Dec} = 0.001$
FoV (circular)	6 deg	2 deg	1 deg
Minimum elevation	30 deg	20 deg	10 deg

Table 2: Estimate of the spacecraft state at t_0 provided in ECI J2000.

Parameter	Value
Ref. epoch t_0 [UTC]	2021-11-19T14:25:39.652
Mean state $\hat{x}_0$ [km, km/s]	$\hat{r}_0 = [23721.610, 17903.673, -49.918]$ $\hat{v}_0 = [-1.150987, 1.529718, 3.122389]$
Covariance P_0 [km^2, km^2/s, km^2/s^2]	$\begin{bmatrix} +2.6e-2 & +1.4e-2 & -1.8e-3 & 0 & 0 & 0 \\ +1.4e-2 & +1.8e-2 & +2.3e-3 & 0 & 0 & 0 \\ -1.8e-3 & +2.3e-3 & +1.0e-2 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1.6e-7 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1.6e-7 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1.6e-7 \end{bmatrix}$

Exercise 2 - Batch filters



Exercise 2: Batch filters

The Two-Line Elements (TLE) set of the spacecraft GIOVE-A of Exercise 1 is reported in Table 3 (and in WeBeep as 28922.tle).

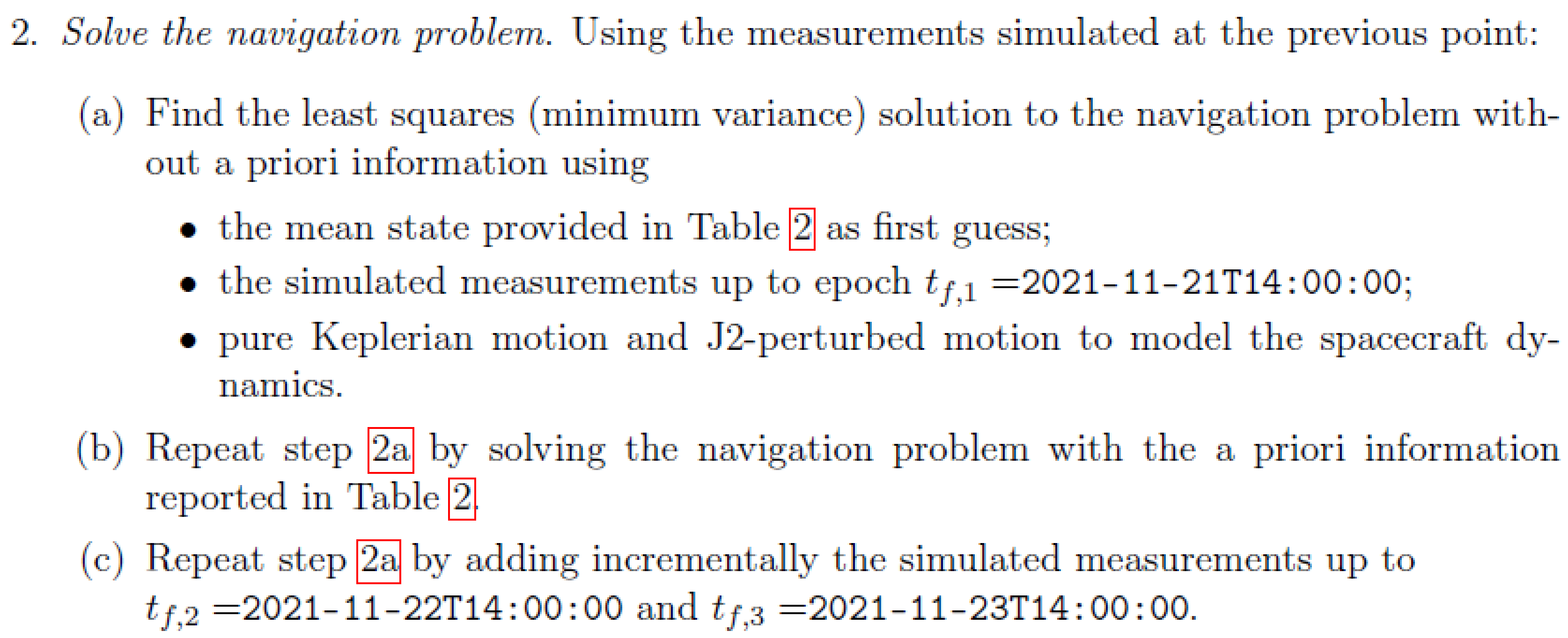
1. *Simulate measurements.* Use SGP4 and the provided TLE to simulate the measurements acquired by the sensor network in Table 1 by:
   (a) Computing the spacecraft position over the same visibility windows identified in Exercise 1 and deriving the associated expected measurements.
   (b) Simulating the measurements by adding a random error to the expected measurements (assume a Gaussian model to generate the random error, with noise provided in Table 1).

Table 3: TLE of spacecraft GIOVE-A.

```
1␣28922U␣05051A␣␣␣21323.60115338␣-.00000088␣␣00000-0␣␣00000+0␣0␣␣9998
2␣28922␣␣58.3917␣␣37.3090␣0004589␣␣75.2965␣284.7978␣␣1.69421077␣98469
```

! Do not copy-paste from PDF

Exercise 2 - Batch filters



2. *Solve the navigation problem.* Using the measurements simulated at the previous point:

   (a) Find the least squares (minimum variance) solution to the navigation problem without a priori information using

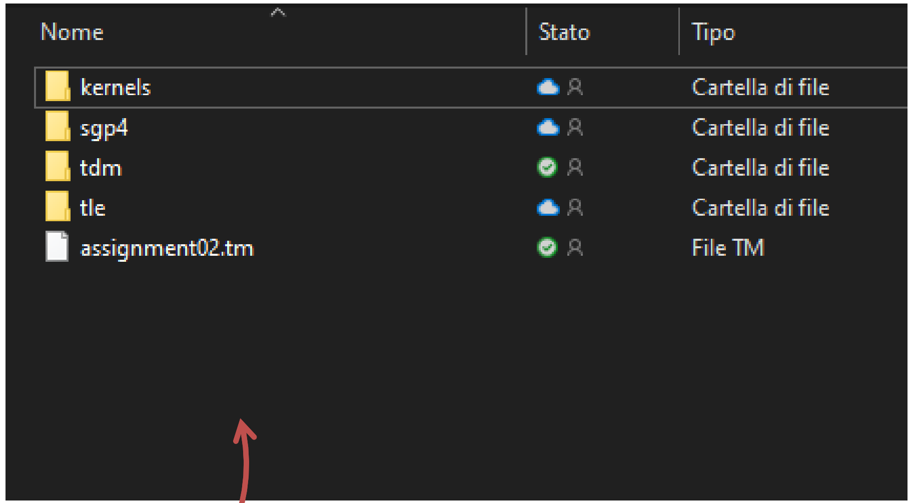
   - the mean state provided in Table 2 as first guess;
   - the simulated measurements up to epoch $t_{f,1}$ =2021-11-21T14:00:00;
   - pure Keplerian motion and J2-perturbed motion to model the spacecraft dynamics.

   (b) Repeat step 2a by solving the navigation problem with the a priori information reported in Table 2.

   (c) Repeat step 2a by adding incrementally the simulated measurements up to $t_{f,2}$ =2021-11-22T14:00:00 and $t_{f,3}$ =2021-11-23T14:00:00.

Supporting material

- Available on **WeBeep**, inside the folder Laboratories
- Compressed folder **Assignment02.zip** contains:
    - Subfolder **`kernels`** with all required SPICE kernels
    - Subfolder **`sgp4`** with all sgp4 functions (Lab03)
    - Subfolder **`tle`** with two-line elements
    - Subfolder **`tdm`** with Tracking Data Message files
    - Meta-kernel **`assignment02.tm`**



Nome	Stato	Tipo
kernels		Cartella di file
sgp4		Cartella di file
tdm		Cartella di file
tle		Cartella di file
assignment02.tm		File TM

lastname123456_Assign2.m

How to prepare your script

- Unzip the folder and work on your Matlab script inside it
- Remember to **load the meta-kernel** and to **add the path to sgp4 subfolder** in your script
    - Suggestion: use relative paths
- **Obs:** no need to upload the supporting material when preparing your delivery on WeBeep

POLITECNICO MILANO 1863

A quick recap

Usage of SGP4

a. Initialize the satellite record with
   - `twoline2rv` (requires tle strings)
   - `read_TLE` (requires TLE file name or sat id)

b. Call SGP4
   - Provide time since reference epoch (minutes)

c. Convert from/to TEME (if needed)
   - Compute `ttt`
     - Centuries from TDT 2000 January 1 00:00:00
   - TEME acceleration can be set to [0,0,0]
   - Get offsets dPsi and dEpsilon
     - Convert them in rad if provided in arcsec
   - Call functions `teme2eci` or `eci2teme`

```
typerun    = 'u';  % user-provided inputs to SGP4 Matlab function
opsmode    = 'a';  % afspc approach ('air force space command')
whichconst =  72;  % WGS72 constants (radius, gravitational parameter)
```

```
% initialize the satrec structure, using the function twoline2rv
satrec = twoline2rv(longstr1, longstr2, typerun,'e', opsmode, whichconst)
```

```
satrec = read_TLE(123456, whichconst);
```

```
% Evaluate the TLE
[satrec,rteme,vteme] = sgp4(satrec,  0.0);
```

```
% Centuries from TDT 2000 January 1 00:00:00.000
et0 = cspice_str2et('2004-04-06T07:51:28.386009');
ttt = cspice_unitim(et0, 'ET', 'TDT')/cspice_jyear()/100;
```

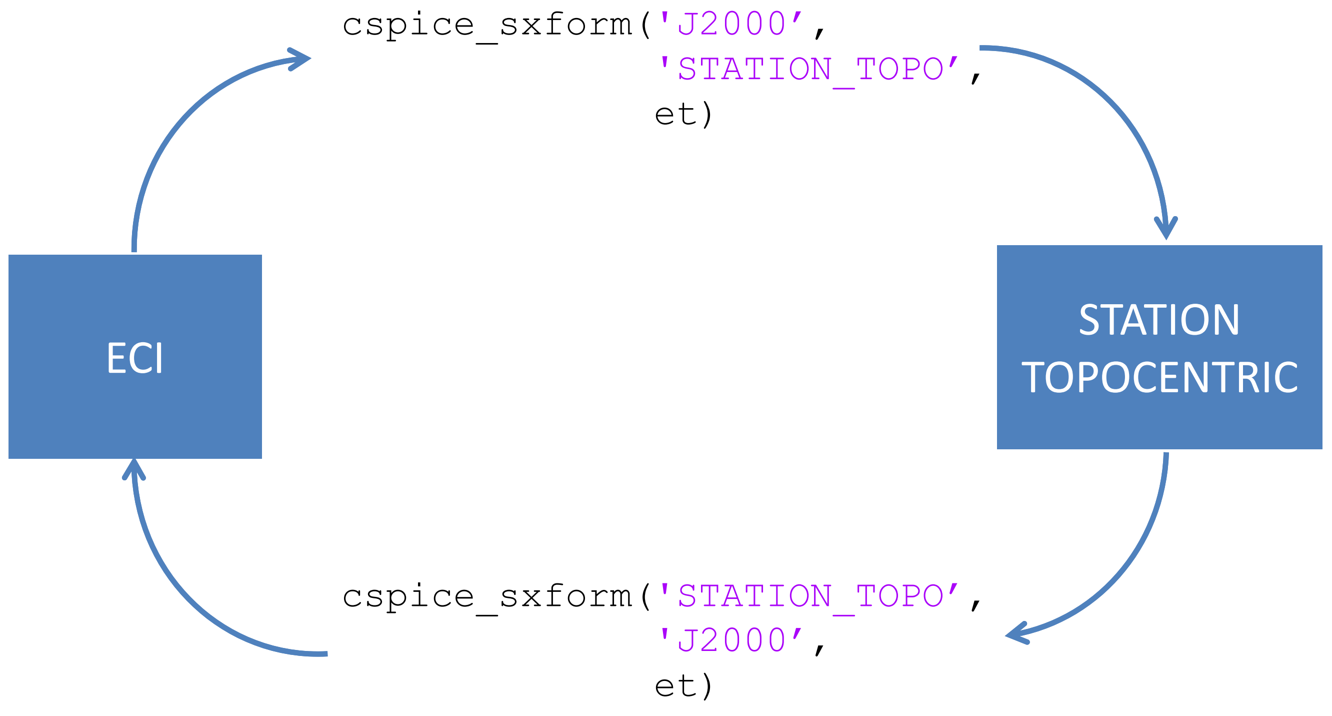
```
% -------- teme2eci    - transform teme to eci vectors
ateme = [0;0;0];
[reci, veci, aeci] = teme2eci(rteme, vteme, ateme, ttt, ddpsi, ddeps);
```

```
% -------- eci2teme    - transform eci to teme vectors
[rteme, vteme] = eci2teme(reci, veci, aeci, ttt, ddpsi, ddeps);
```

A quick recap (2)

Reference frame conversions

With station kernel generated with `pinpoint`



```
cspice_sxform('J2000',
              'STATION_TOPO',
              et)
```

```
cspice_sxform('STATION_TOPO',
              'J2000',
              et)
```

Without station kernel

```
% Define station coordinates
lat = 45.50122; % deg
lon = 9.15461; % deg
alt = 20; % m

% Get Earth radii (equatorial and polar)
% (requires pck00010.tpc kernel)
radii = cspice_bodvrd('EARTH', 'RADII', 3);
re = radii(1); rp = radii(3);

% Compute flattening
flat = (re - rp) / re;

% Convert to radians and km
lat_rad = deg2rad(lat); lon_rad = deg2rad(lon);
alt_km = alt / 1000;

% Compute station pos wrt Earth center (ECEF)
pos_ECEF = cspice_pgrrec(...
    'EARTH', lon_rad, lat_rad, alt_km, re, flat);
% Compute ECEF2TOPO rotation matrix
ECEF2TOPO = cspice_eul2m(...
    lat_rad - pi, pi - lon_rad, pi / 2, 2, 1, 2);
```

A quick recap (3)

Use `mvnrnd` to generate multivariate normal random numbers

- **Generate sample points for Monte Carlo simulations**
  Example: generate `100` vectors with given mean `[1 3]` and covariance `diag([0.01 0.02])`:
  ```
  n = 100; mu = [1 3]; Sigma = diag([0.01 0.02]);
  R = mvnrnd(mu, Sigma, n);  % R size: 100x2
  ```

- **Add random noise to measurements**
  Example: add noise with covariance `diag([0.01 0.02])` to a set of angular measurements :
  ```
  mu = [1 3; 1.5 2; 2 1]; Sigma = diag([0.01 0.02]);
  R = mvnrnd(mu, Sigma);  % R size: 3x2
  ```

Notes:

- When Sigma is diagonal, you can directly pass the diagonal (e.g. `Sigma = [0.01 0.02];`)
- It is possible to define a different Sigma for each measurement (using the 3rd dimension)

A quick recap (4a)

Solve nonlinear least-squares problems in MATLAB

```
[x, resnorm, residual, exitflag, ~, ~, jac] = lsqnonlin(fun, x0, lb, ub, opt);
```

Inputs:

- `fun`: handle of the cost function to be minimized
    - `fun` must take as input only the x and must return the residual
    - the residual can be a vector, `lsqnonlin` implicitly computes the norm
- `x0`: initial point
- `lb, ub`: lower and upper bounds
- `opt`: to be defined using the `optimoptions` function

Outputs:

- `x`: solution
- `resnorm`: squared norm of the residual
- `residual`: value of `fun(x)`
- `exitflag`: reason the solver stopped (convergence if `exitflag` > 0)
- `jac`: jacobian matrix of `fun` with respect to `x`

A quick recap (4b)

Solve nonlinear least-squares problems in MATLAB - example

```
% Variables of the problem
x_0 = ...; var_1 = ...; var_2 = ...;

% Encapsuate the variables in the function handle
fun = @(x) costfunction(x, var_1, var_2, ...);

% Optimization options
opt = optimoptions('lsqnonlin', 'Algorithm', 'levenberg-marquardt', 'Display', 'iter');

% Execute least squares
[x, resnorm, residual, exitflag, ~, ~, jac] = lsqnonlin(fun, x_0, [], [], opt);

% Compute resulting covariance
Jac = full(jac);
P_ls = resnorm / (length(residual)-length(x_0)) .* inv(Jac'*Jac);

function residual = costfunction(x, var_1, var_2, ... )
    residual = []; % Initialize output variable
    % If least squares with "a priori" append it to the residual
    diff_apriori_weighted = W_ap * (x - x0);
    residual = [residual; diff_apriori_weighted(:)];
    % Propagate x to the epochs of the measurements
    x_prop = ...;
    % Compute predicted measurements
    meas_pred = ...;
    % Compute the residual of the measurements and append it to the output
    diff_meas_weighted = W_m * (meas_pred - meas_real);
    residual = [residual; diff_meas_weighted(:)];
end
```

Weight for «a priori»

```
W_ap = inv(sqrtm(P_0));
```

Weight for measurements:

```
W_m = diag(1 ./ sigma_meas.~~^2~~);
```

General hints

LaTeX

- Never put multiplication symbols, especially as asterisks
- Pay attention to distinguish between vectors and scalar: write vector quantities using underline or in bold
  - Latin alphabet: `\mathbf{x}` or `\mathbf{r}`
  - Greek symbols : `\boldsymbol{\lambda}`
  - **Obs:** No need to use norm with this notation $\|\boldsymbol{\lambda}_x\| \leftrightarrow \lambda_x$
- Write dot product using `\cdot`
- Prefer the use of `\dfrac{}{}`over`\frac{}{}` when writing equations
- When inserting text in an equation wrap it in `\mathrm{}`

Plots

- Make sure that axis labels and titles are clearly readable, even without zooming
- Remember to put the (correct!) units