ENAE 441 - 0101

Final Project: Extended Kalman Filter

Due on December 16th, 2025 at 12:30 PM

Dr. Martin, 09:30 AM

Vai Srivastava

December 17th, 2025

Overview

You are working at NASA GSFC on the navigation team for an Earth orbiting science satellite launched into orbit on a Falcon-9. SpaceX designed its launch to deliver the payload into low Earth orbit with reference coordinates of

$$\boldsymbol{X}_{\text{œ}} = \begin{bmatrix} a \\ e \\ i \\ \omega \\ \Omega \\ \theta \end{bmatrix} = \begin{bmatrix} 7 \times 10^3\,\text{km} \\ 0.2 \\ 45^\circ \\ 0^\circ \\ 270^\circ \\ 78.75^\circ \end{bmatrix}$$

however it's the navigation's team responsibility to verify this orbit and/or determine any errors in its delivery. To accomplish this, the navigation team has access to range and range-rate measurements from the spacecraft provided by the Deep Space Network (DSN). The latitude and longitude (ϕ, λ) of the three DSN ground stations are provided below:

- DSN # 0: Goldstone, USA: $(35.297^\circ, -116.914^\circ)$
- DSN # 1: Madrid, Spain: $(40.4311^\circ, -4.248^\circ)$
- DSN # 2: Canberra, Australia: $(-35.4023^\circ, 148.9813^\circ)$

These ground stations are positioned on a spherical Earth of radius 6378.137 km , with a rotation rate of $\omega_{\mathcal{E}/\mathcal{N}} = 7.292\,115 \times 10^{-5}\,\frac{\text{rad}}{\text{s}}$ and a Local Sidereal Time of $\gamma_0 = 0^\circ$.
`Project-Measurements-Easy.npy` contains the range ρ and range-rate $\dot{\rho}$ measurements alongside additional information. Specifically, each row in the datafile is formatted as:

$$[t, i, \rho, \dot{\rho}]$$

where t corresponds with the time the measurement was received and i corresponds to the ground station index (as labeled above).
As best as you are aware, the spacecraft's motion is governed by the following differential equation:

$$\ddot{\boldsymbol{r}} = -\frac{\boldsymbol{\mu}}{r^3} \cdot \boldsymbol{r}$$

where $\boldsymbol{r}$ is the position of the spacecraft in the inertial frame. Similarly, the measurements provided can be computed using the following equations:

$$\begin{aligned} \boldsymbol{\rho} &= \boldsymbol{r} - \boldsymbol{R}_{\text{site},i} \\ \rho &= \|\boldsymbol{r} - \boldsymbol{R}_{\text{site},i}\| \\ \dot{\rho} &= \frac{\boldsymbol{\rho}\left(\dot{\boldsymbol{r}} - \dot{\boldsymbol{R}}_{\text{site},i}\right)}{\rho} \\ \dot{\boldsymbol{R}}_{\text{site},i} &= \boldsymbol{\omega}_{\mathcal{E}/\mathcal{N}} \times \boldsymbol{R}_{\text{site},i} \end{aligned}$$

where $\boldsymbol{R}_{\text{site},i}$ is the location of the DSN station also in the inertial frame. Note that the measurements provided by the DSN have some intrinsic noise which can be modeled as Gaussian white noise. Explicitly, the noise in the range is characterized by a variance of $1\,\text{m}^2$, and by $1\,\frac{\text{cm}^2}{\text{s}^2}$ variance in the range-rate.
Using your knowledge of the spacecraft's dynamics and the measurements provided by the DSN, generate a report which culminates in an estimate of the spacecraft's state over time. To help facilitate your progress, please generate your report in the following order, answering the following intermediate questions:

Problem 1: Problem Setup

a. Express the non-linear system in continuous time state-space form, clearly defining the vectors $\boldsymbol{f}\left(X\left(t\right)\right)$ and $\boldsymbol{h}\left(X\left(t\right)\right)$

b. Define the linearized dynamics and measurement matrices $A\left(t\right)$ and $C\left(t\right)$.

c. Show how these matrices are converted to their discrete time forms F_k and H_k. Recall F_k is the state transition matrix $\boldsymbol{\Phi}\left(t_j, t_i\right)$ which requires integration.

d. Define your noise matrices Q_k and R_k, and discuss their relationship to the aforementioned system of equations.

e. Plot the measurements as a function of time.

Solution

Part A

Answer

Part B

Answer

Part C

Answer

Part D

Answer

Part E

Answer

Problem 2: Plan Filter Implementation

Provide pseudocode from which you will base your extended Kalman filter implementation. Highlight the major steps in your algorithm and any noteworthy modifications or subtle details required for this problem that you want the grades to be aware of. Be comprehensive, as this is what the grading team will primarily reference if the results/plots don't quite look right.

Solution

Answer

Problem 3: Pure Prediction

a.

b.

c.

Problem 3 continued on next page...

Solution

Code

See the Python code for this assignment.

Problem 4: Measurement Updates

a.

b.

c.

Solution

Code

See the Python code for this assignment.

Problem 5: Filter Solutions

a.

b.

c.

d.

Solution

Code

See the Python code for this assignment.

Problem 6: Debugging Efforts *(Optional)*

Use this section to outline any of your debugging efforts for if things aren't going your way. This is a good place to earn some partial credit. This should be a "research" log of what experiments you performed and why. A list of guiding questions if your stuck include:

1. Consider how process noise matrix is used in the filter. What happens if there are large gaps between measurements?
2. Consider if the values used in your measurement noise matrix are appropriate. Should these values only reflect the uncertainty in the sensor?
3. If your filter is diverging, does the divergence start at the beginning or mid-way through? What possible reasons exist for either outcome?
4. If you had to define a single scalar metric to evaluate your filter's quality, what would it be, and can you use this to help you determine optimal tuning values?

Problem 6 continued on next page...

5. Does plotting your best estimate in a different reference frame or element description help?

Solution

Answer

Problem 7: Challenge Orbit *(Bonus)*

Perform a second analysis with the more difficult dataset `Project-Measurements-Hard.npy`. Use

$$\begin{bmatrix} a \\ e \\ i \\ \omega \\ \Omega \\ \theta \end{bmatrix} = \begin{bmatrix} 7000\,\text{km} \\ 0.6 \\ 45^\circ \\ 180^\circ \\ 0^\circ \\ 45^\circ \end{bmatrix}$$

as your initial guess. Your analysis can amount to your debugging process, and points will be awarded based on how thoughtful your experimentation is and the quality of your solution.

Solution

Answer

Code

See the Python code for this assignment.

Code

```python
import numpy as np
import matplotlib.pyplot as plt
import os


def main():
    print("Hello World!")

    #######
    # p02 #
    #######

    x = np.linspace(0, 10, 100)
    y = np.sin(x)

    with open("./outputs/text/s02.txt", "w", encoding="utf-8") as f:
        f.write((f"x:\n{x}\n\ny:\n{y}"))

    plt.figure()
    plt.plot(x, y)

    save_path = os.path.join(os.path.dirname(__file__), "./outputs/figures/s02.png")
    plt.savefig(save_path)
    plt.show()


if __name__ == "__main__":
    main()
```