ENAE 441 - 0101

HW05: State Estimation

Due on December 4^{th}, 2025 at 09:30 AM

Dr. Martin, 09:30 AM

Vai Srivastava

December 3^{rd}, 2025

Problem 1: Batch Least Squares Estimation *(30 pts.)*

Imagine there is a spacecraft flying in a circular geosynchronous orbit around the Earth. To an observer on the rotating Earth, the spacecraft location appears static. You do not know the exact semi-major axis, but you can estimate it using a range measurement taken from a radio on the ground.

$$\rho = \sqrt{x^2 + y^2 + z^2}$$

Assume you are positioned directly beneath the satellite such that range measurement can be reduced to the following form:

$$\rho = \sqrt{z^2} = z$$

For **50** consecutive nights, you go outside and point your radio antenna towards the spacecraft and collect 200 noisy range measurements. The radio antenna is not particularly precise, so it has continuous measurement noise properties $\tilde{\boldsymbol{v}}(t) \sim \mathcal{N}(0, V)$ where $V = 100\,\mathrm{m}^2$. Use this information and the measurements provided in `HW5Measurements.npy`[1] to answer the following questions:

a. Express the system in continuous time state-space form assuming a state of

$$\boldsymbol{X} = \left[x, y, z, \dot{x}, \dot{y}, \dot{z}\right]^T$$

and convert to a discrete time state-space model assuming $\Delta t = 10\,\mathrm{s}$

b. Use batch least squares algorithm[2] to estimate the spacecraft state from the first trial in `HW5Measurements.npy`. Plot the estimated position as a function of $k = \{1, \ldots, 200\}$ measurements.

c. Repeat the batch estimation process for all **50** trials included in `HW5Measurements.npy`, and plot the resulting state estimates as a function of k along side their $\pm 3\sigma$ error bounds[3][4].

d. Using all 50 trials, plot histograms for each state estimate in $\hat{\mathbf{x}}$ after $k \in \{10, 50, 200\}$ measurements. Include the sample mean and error covariance for each distribution as annotations[5]. Do these values make sense? Why?

e. Measure the average amount of time it takes to compute an estimate as a function of k. Use Python's `time` library.

Solution

Part A

In matrix form:

$$\dot{\boldsymbol{X}}(t) = A_c \boldsymbol{X}(t),$$

with

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

[1] The measurements provided are formatted as a $N \times M$ array where N corresponds to the trial and M corresponds to the number of measurements per trial and the units are **meters**.

[2] Hint: When you need to take a matrix inverse for BLLS and subsequent algorithms, use `np.linalg.pinv` rather than `np.linalg.inv` to avoid singularities.

[3] Recall that the diagonal elements of the state error covariance matrix yield σ_i^2.

[4] The `matplotlib` function `plt.fill_between` can be useful for the $\pm 3\sigma$ bounds that center on the mean.

[5] Use `np.nanmean` and `np.nanstd` over all values for a particular k.

Problem 1 continued on next page...

Measurement model:

$$\rho(t) = H\,\boldsymbol{X}(t) + \tilde{\boldsymbol{v}}(t), \quad \tilde{\boldsymbol{v}}(t) \sim \mathcal{N}(0, V),$$

with

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad V = 100\,\mathrm{m}^2$$

The discrete state transition is:

$$\boldsymbol{X}_{k+1} = \boldsymbol{\Phi}\boldsymbol{X}_k, \quad \boldsymbol{\Phi} = e^{A_c \Delta t}$$

A_c has the structure of a constant-velocity model:

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathcal{I}_{3\times3} & \Delta t \mathcal{I}_{3\times3} \\ 0_{3\times3} & \mathcal{I}_{3\times3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, the measurement equation in discrete time is:

$$\boxed{\rho_k = H\boldsymbol{X}_k + v_k, \quad v_k \sim \mathcal{N}(0, V), \quad V = 100\,\mathrm{m}^2}$$

Part B

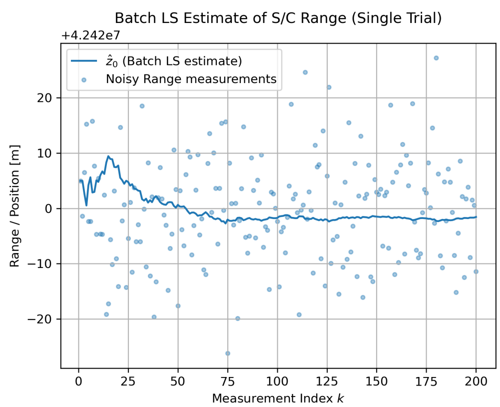


Figure 1: blah

Part C

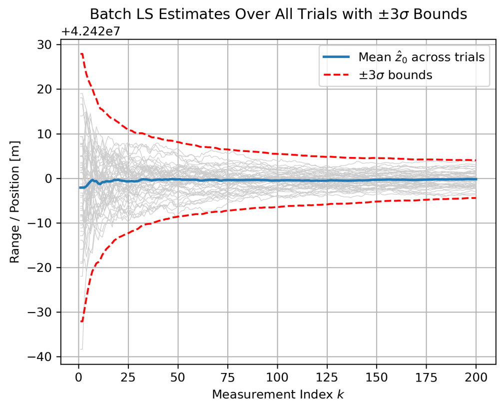
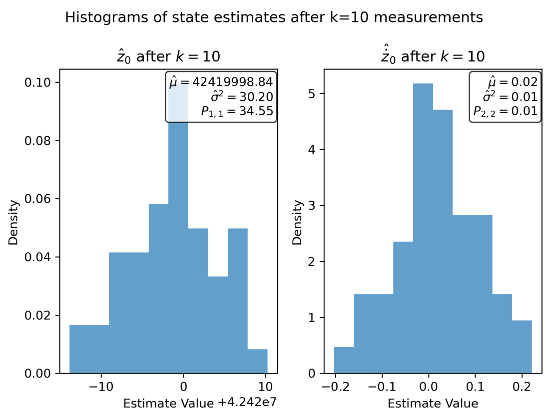
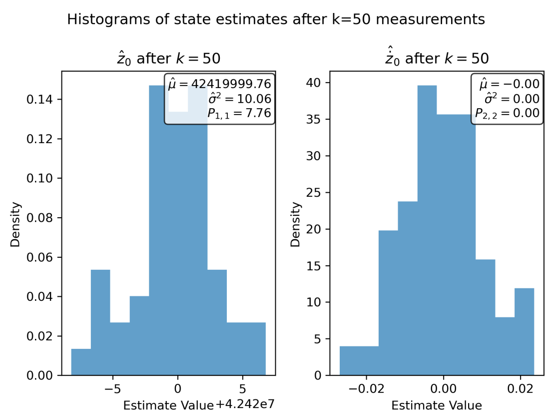


Figure 2: blah

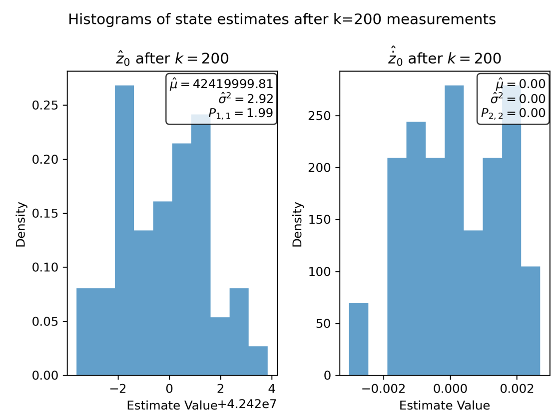
Part D



(a) blah



(b) blah



(c) blah

```
For each k ∈ {10, 50, 200}, the histograms show the distribution of the
batch least-squares state estimates across the 50 trials.
```

As k increases, the histograms for $\hat{z}_0$ become narrower and more tightly clustered around a common mean. The sample variances of $\hat{z}_0$ decrease and approach the corresponding diagonal entries of the analytical covariance $P_k = (\Gamma_k^T R^{-1} \Gamma_k)^{-1}$, which is expected for a linear unbiased least-squares estimator driven by independent Gaussian noise.

The estimates of $\dot{\hat{z}}_0$ are centered close to zero (since the spacecraft is effectively stationary in the chosen frame), and their spread also shrinks with increasing k, reflecting the fact that longer time spans give more leverage to estimate the slope in the constant-velocity model.

Part E

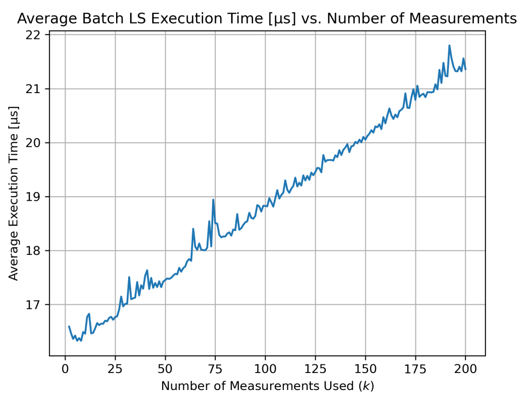


Figure 4: blah

Code

See the Python code for this problem.

Problem 2: Recursive Least Squares Estimation *(30 pts.)*

a. Using the same dataset, implement the recursive least squares algorithm to estimate $\hat{\mathbf{x}}$. Use an initial estimate of:

$$\hat{\mathbf{x}}_0 = \left[0\,\text{km}, 0\,\text{km}, 42 \times 10^3\,\text{km}, 0\,\frac{\text{km}}{\text{s}}, 0\,\frac{\text{km}}{\text{s}}, 0\,\frac{\text{km}}{\text{s}}\right]$$

and a covariance of:

$$P_0 = \begin{bmatrix} 50 \cdot \mathbb{I}_{3\times3}\text{km}^2 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & 1 \cdot \mathbb{I}_{3\times3}\frac{\text{km}^2}{\text{s}^2} \end{bmatrix}$$

Plot the state estimate for the 50 trials as a function of k.

b. Plot the sample mean state estimate μ_k across all trials and the $\pm 3\sigma$ error bounds around the estimate. Does this mirror the same plot in Problem 1, Part D? Why or why not?

c. Measure/plot the average amount of time it takes to perform each update from $k = \{1, \ldots, 200\}$ and compare against the batch least squares estimator. Explain any differences.

Solution

Part A

Figure 5: blah

Part B

Problem 2 continued on next page...

Figure 6: blah

```
Write your answer here.
Write your answer here.
```

Part C

Figure 7: blah

```
Write your answer here.
Write your answer here.
```

Code

See the Python code for this problem.

Problem 3: Kalman Filtering *(40 pts.)*

Solution

Part A

Answer

Part B

```
x_final = 0
P_final = 2
```

Part C

Figure 8: blah

Part D

Figure 9: blah

Problem 3 continued on next page...

Part E

```
Write your answer here.
Write your answer here.
```

Part F

Figure 10: blah

```
Write your answer here.
Write your answer here.
```

Code

See the Python code for this problem.