School of Engineering, Discipline of Mechanical Engineering and Mechatronics, pp. 1–14

Lab 5: Square-root extended Kalman filter

MCHA4400

Semester 2 2023

Introduction

A SpaceY Starship is re-entering earth's atmosphere and an extended Kalman filter is needed to estimate its motion. SpaceY engineers were not expecting the Starship to fall in the orientation depicted in Figure 1, as such, the drag coefficient needs to be estimated online along with the altitude and velocity of the rocket.

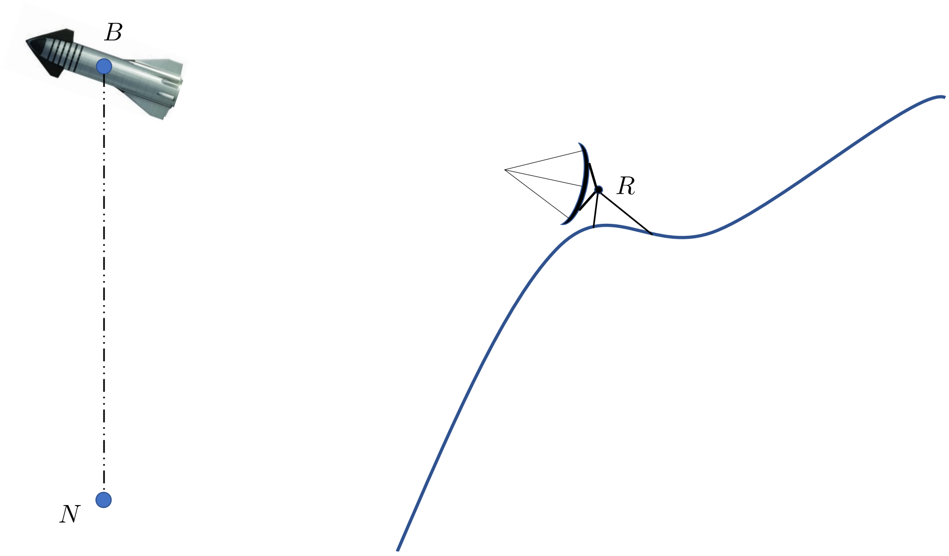

Figure 1: Vertical re-entry configuration of the SpaceY Starship

In this lab, you will do the following:

1. Implement the process dynamics and measurement model.
2. Complete the implementation of a square-root extended Kalman filter.
3. Plot results using VTK.

The lab should be completed within 4 hours. The assessment will be done in the lab at the end of your enrolled lab session(s). Once you complete the tasks, call the lab demonstrator to start your assessment.

The lab is worth 5% of your course grade and is graded from 0–5 marks.

LLM Tip

You are strongly encouraged to explore the use of Large Language Models (LLMs), such as GPT, PaLM or LLaMa, to assist in completing this activity. Bots with some of these LLMs are available to use via the UoN Mechatronics Slack team, which you can find a link to from Canvas. If you are are unsure how to make the best use of these tools, or are not getting good results, please ask your lab demonstrator for advice.

1 Ballistic process model (1 mark)

The following state vector has been proposed:

$$\mathbf{x} = \begin{bmatrix} h \\ v \\ c \end{bmatrix}, \tag{1}$$

where h is the altitude of the rocket above the reference point N, v is the velocity of the rocket and c is the ballistic drag coefficient to be estimated online.

The mean dynamics of the state vector can be expressed in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{2}$$

where the dynamics of the free-falling Starship are given by the following expression:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} v \\ d - g \\ 0 \end{bmatrix}. \tag{3}$$

The acceleration d due to drag is given by

$$d = \frac{1}{2}\rho v^2 c \tag{4}$$

and the air density is given by

$$\rho = \frac{pM}{RT}, \tag{5}$$

where M is the molar mass of air, R is the ideal gas constant and T is the air temperature.

If the air is in hydrostatic equilibrium, then an infinitesimal change in pressure $\mathrm{d}p$ can be related to a infinitesimal change in altitude $\mathrm{d}h$ as follows:

$$\mathrm{d}p = -\rho g \, \mathrm{d}h. \tag{6}$$

Dividing both sides by p and substituting (5) yields

$$\frac{\mathrm{d}p}{p} = -\frac{\rho g}{p} \, \mathrm{d}h = -\frac{gM}{RT} \, \mathrm{d}h. \tag{7}$$

In the lower atmosphere, the temperature can be assumed to vary with altitude according to

$$T = T_0 - Lh, \tag{8}$$

where T_0 is the standard temperature and L is the temperature lapse rate. Substituting (8) into (7) and integrating yields

$$
\begin{aligned}
\frac{\mathrm{d}p}{p} &= -\frac{gM}{R(T_0 - Lh)}\,\mathrm{d}h \\
\int_{p_0}^{p} \frac{1}{p'}\,\mathrm{d}p' &= -\frac{gM}{R}\int_0^h \frac{1}{T_0 - Lh'}\,\mathrm{d}h' \\
\big[\log p'\big]_{p'=p_0}^{p'=p} &= -\frac{gM}{R}\left[-\frac{1}{L}\log(T_0 - Lh')\right]_{h'=h_0}^{h'=h} \\
\log p - \log p_0 &= \frac{gM}{RL}(\log(T_0 - Lh) - \log T_0) \\
p &= p_0 \exp\left(\frac{gM}{RL}\log\left(1 - \frac{Lh}{T_0}\right)\right) \\
p &= p_0\left(1 - \frac{Lh}{T_0}\right)^{\frac{gM}{RL}},
\end{aligned}
\tag{9}
$$

which is known as the barometric formula.

Substituting (9), (8) and (5) into (4) yields the drag acceleration as a function of states,

$$
\begin{aligned}
d &= \frac{1}{2}\rho v^2 c \\
d &= \frac{1}{2}\frac{pM}{RT}v^2 c \\
d &= \frac{1}{2}\frac{Mp_0}{R}\left(\frac{1}{T_0 - Lh}\right)\left(1 - \frac{Lh}{T_0}\right)^{\frac{gM}{RL}} v^2 c.
\end{aligned}
\tag{10}
$$

where $p_0 = 101\,325\,\mathrm{Pa}$ is the standard pressure at sea level, $T_0 = 288.15\,\mathrm{K}$ is the standard temperature at sea level, $g = 9.81\,\mathrm{m/s^2}$ is the acceleration due to gravity, $M = 0.028\,964\,4\,\mathrm{kg/mol}$ is the molar mass of dry air, $R = 8.314\,47\,\mathrm{J/(mol\,K)}$ is the ideal gas constant and $L = 0.0065\,\mathrm{K/m}$ is the temperature lapse rate in the lower atmosphere.

The process model for the rocket dynamics is then given by

$$
\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \dot{\mathbf{w}}(t), \qquad \dot{\mathbf{w}}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}\delta(t - t')),
\tag{11}
$$

where the power spectral density of the process noise is

$$
\mathbf{Q} = \operatorname{diag}\left(\begin{bmatrix} 0 & 1 \times 10^{-20} & 25 \times 10^{-12} \end{bmatrix}\right).
\tag{12}
$$

Tasks

a) Complete the implementation of the `StateBallistic::dynamics` functions in `src/StateBallistic.cpp`.

b) Ensure that the unit tests provided in `test/src/StateBallistic.cpp` pass.

Tip

You can prepare the build directory and run the unit tests as follows:

Terminal

```
nerd@basement:~/MCHA4400/lab5$ cmake -G Ninja -B build -DCMAKE_BUILD_TYPE=Debug && cd build
nerd@basement:~/MCHA4400/lab5/build$ ninja
[Lots of failing unit tests, pay attention to the relevant ones only]
```

If the non-relevant unit tests are too distracting, you can temporarily move them out of the `test/src` directory and move them back in as needed for the following parts.

2 Range measurement model (1 mark)

A range-only RADAR (ROR) unit provides range measurements of the rocket from position $(r_1, r_2) = (5000\,\mathrm{m}, 5000\,\mathrm{m})$. Therefore, the measurement model is given by

$$h_{\mathrm{rng}}(\mathbf{x}) = \sqrt{r_1^2 + (h - r_2)^2}, \tag{13}$$

with Gaussian measurement noise of $\sigma_{\mathrm{rng}} = 50\,\mathrm{m}$.

Tasks

a) Complete the implementation of the `MeasurementRADAR::predict` functions in `src/MeasurementRADAR.cpp`

b) Ensure that the unit tests provided in `test/src/MeasurementRADAR.cpp` pass.

3 Marginal distributions of a Gaussian (1 mark)

Let $\mathbf{x} \in \mathbb{R}^n$ and let

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P}) \tag{14}$$

be a multivariate Gaussian distribution, where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the mean vector and $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the symmetric positive definite-covariance matrix. If we partition

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}, \tag{15}$$

where the head $\mathbf{x}_a \in \mathbb{R}^{n_a}$ and the tail $\mathbf{x}_b \in \mathbb{R}^{n_b}$ with $n_a + n_b = n$, then we can write (14) as

$$p\left(\begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{aa} & \mathbf{P}_{ab} \\ \mathbf{P}_{ba} & \mathbf{P}_{bb} \end{bmatrix}\right), \tag{16}$$

where $\boldsymbol{\mu}_a \in \mathbb{R}^{n_a}$, $\boldsymbol{\mu}_b \in \mathbb{R}^{n_b}$, $\mathbf{P}_{aa} \in \mathbb{R}^{n_a \times n_a}$, $\mathbf{P}_{ab} \in \mathbb{R}^{n_b \times n_a}$, $\mathbf{P}_{ba} \in \mathbb{R}^{n_b \times n_a}$ and $\mathbf{P}_{bb} \in \mathbb{R}^{n_b \times n_b}$.

The marginal distributions of the head $\mathbf{x}_a$ and the tail $\mathbf{x}_b$ are then given as follows:

$$\begin{aligned} p(\mathbf{x}_a) &= \mathcal{N}(\mathbf{x}_a; \boldsymbol{\mu}_a, \mathbf{P}_{aa}), \\ p(\mathbf{x}_b) &= \mathcal{N}(\mathbf{x}_b; \boldsymbol{\mu}_b, \mathbf{P}_{bb}). \end{aligned}$$

We can also write (14) using a upper-triangular square-root parameterisation of the covariance matrix, denoted as

$$p(\mathbf{x}) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{S}), \tag{17}$$

where $\mathbf{S} \in \mathbb{R}^{n \times n}$ is an upper triangular matrix such that $\mathbf{S}^{\mathsf{T}}\mathbf{S} = \mathbf{P}$.

Then, under the partition (15), we have the following relations between the block elements of $\mathbf{P}$ and $\mathbf{S}$:

$$\begin{aligned} \mathbf{P} &= \mathbf{S}^{\mathsf{T}}\mathbf{S} \\ \begin{bmatrix} \mathbf{P}_{aa} & \mathbf{P}_{ab} \\ \mathbf{P}_{ba} & \mathbf{P}_{bb} \end{bmatrix} &= \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{0} & \mathbf{S}_3 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{0} & \mathbf{S}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_1^{\mathsf{T}} & \mathbf{0} \\ \mathbf{S}_2^{\mathsf{T}} & \mathbf{S}_3^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{0} & \mathbf{S}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_1^{\mathsf{T}}\mathbf{S}_1 & \mathbf{S}_1^{\mathsf{T}}\mathbf{S}_2 \\ \mathbf{S}_2^{\mathsf{T}}\mathbf{S}_1 & \mathbf{S}_2^{\mathsf{T}}\mathbf{S}_2 + \mathbf{S}_3^{\mathsf{T}}\mathbf{S}_3 \end{bmatrix}, \end{aligned} \tag{18}$$

where $\mathbf{S}_1 \in \mathbb{R}^{n_a \times n_a}$, $\mathbf{S}_2 \in \mathbb{R}^{n_a \times n_b}$, $\mathbf{S}_3 \in \mathbb{R}^{n_b \times n_b}$ and $\mathbf{S}_1$ and $\mathbf{S}_3$ are upper triangular.

The marginal distribution of $\mathbf{x}_a$ is given by

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a; \boldsymbol{\mu}_a, \mathbf{S}_1^{\mathsf{T}}\mathbf{S}_1) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}_a; \boldsymbol{\mu}_a, \mathbf{S}_a), \tag{19}$$

where $\mathbf{S}_a = \mathbf{S}_1$, since $\mathbf{S}_1$ is already upper triangular. The marginal distribution of $\mathbf{x}_b$ is given by

$$p(\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_b; \boldsymbol{\mu}_b, \mathbf{S}_2^{\mathsf{T}}\mathbf{S}_2 + \mathbf{S}_3^{\mathsf{T}}\mathbf{S}_3) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}_b; \boldsymbol{\mu}_b, \mathbf{S}_b), \tag{20}$$

for some upper triangular matrix $\mathbf{S}_b \in \mathbb{R}^{n_b \times n_b}$ such that $\mathbf{S}_b^{\mathsf{T}}\mathbf{S}_b = \mathbf{S}_2^{\mathsf{T}}\mathbf{S}_2 + \mathbf{S}_3^{\mathsf{T}}\mathbf{S}_3$. To find such a matrix, we first factorise the sum as follows:

$$\begin{aligned} \mathbf{S}_b^{\mathsf{T}}\mathbf{S}_b &= \mathbf{S}_2^{\mathsf{T}}\mathbf{S}_2 + \mathbf{S}_3^{\mathsf{T}}\mathbf{S}_3 \\ &= \begin{bmatrix} \mathbf{S}_2^{\mathsf{T}} & \mathbf{S}_3^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{S}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{S}_3 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{S}_3 \end{bmatrix}. \end{aligned}$$

Then, let

$$\underbrace{\begin{bmatrix} \mathbf{S}_2 \\ \mathbf{S}_3 \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} \mathbf{Y} & \mathbf{Z} \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}}_{\mathbf{R}}$$

be a QR factorisation, where $\mathbf{A} \in \mathbb{R}^{n \times n_b}$, $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{R}^{n \times n_b}$ is an upper triangular matrix. Then,

$$\begin{aligned} \mathbf{A}^\mathsf{T}\mathbf{A} &= \mathbf{R}^\mathsf{T}\mathbf{Q}^\mathsf{T}\mathbf{Q}\mathbf{R} = \mathbf{R}^\mathsf{T}\mathbf{R} && \text{(since } \mathbf{Q} \text{ is orthogonal)} \\ \begin{bmatrix}\mathbf{S}_2 \\ \mathbf{S}_3\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{S}_2 \\ \mathbf{S}_3\end{bmatrix} &= \begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix} \\ &= \mathbf{R}_1^\mathsf{T}\mathbf{R}_1. \end{aligned}$$

Therefore, $\mathbf{S}_b = \mathbf{R}_1$, since $\mathbf{R}_1$ is upper triangular.

As we have seen, when working with an upper-triangular square-root factorisation of a joint covariance matrix, it is much easier to extract the marginal distribution of the head (19) than the tail (20) of a partition. The former requires only extracting the top left block of $\mathbf{S}$ to find $\mathbf{S}_a$, whereas the latter requires extra computation to upper-triangularise the rightmost columns of $\mathbf{S}$ via a Q-less QR decomposition and then extract the top rows of the result to find $\mathbf{S}_b$.

Foreshadowing

In landmark SLAM, extracting the marginal distributions of a non-contiguous set of states is a common operation. At each time step, the marginal distribution of all landmarks that are predicted to be visible in the current frame is needed for correct data association with the measured image features in that frame. Since the order of the landmarks in the state is arbitrary, the states corresponding to visible landmarks do not form simple partitions of the state.

So far, we have only considered the marginal distributions of simple partitions of $\mathbf{x}$. To find the marginal distributions of a non-contiguous subset of $\mathbf{x}$, let us introduce an index vector $\mathcal{I} \in \mathbb{N}^{n_\mathcal{I}}$ to select the desired elements and denote this as $\mathbf{x}_\mathcal{I}$. Each element of $\mathbf{x}_\mathcal{I}$ is given by

$$(\mathbf{x}_\mathcal{I})_i = x_{\mathcal{I}_i}, \quad \text{for } i = 1, 2, \ldots, n_\mathcal{I}.$$

Example

Let $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^\mathsf{T}$ and let $\mathcal{I} = \begin{bmatrix} 3 & 5 & 1 \end{bmatrix}$. Then $\mathbf{x}_\mathcal{I} = \begin{bmatrix} x_3 & x_5 & x_1 \end{bmatrix}^\mathsf{T}$.

The marginal distribution of $\mathbf{x}_\mathcal{I}$ is given by

$$p(\mathbf{x}_\mathcal{I}) = \mathcal{N}(\mathbf{x}_\mathcal{I}; \boldsymbol{\mu}_\mathcal{I}, \mathbf{P}_{\mathcal{I}\mathcal{I}}), \tag{21}$$

where $\boldsymbol{\mu}_\mathcal{I} \in \mathbb{R}^{n_\mathcal{I}}$ with elements given by $(\boldsymbol{\mu}_\mathcal{I})_i = \mu_{\mathcal{I}_i}$ for $i = 1, 2, \ldots, n_\mathcal{I}$ and $\mathbf{P}_{\mathcal{I}\mathcal{I}} \in \mathbb{R}^{n_\mathcal{I} \times n_\mathcal{I}}$ with elements given by $(\mathbf{P}_{\mathcal{I}\mathcal{I}})_{i,j} = P_{\mathcal{I}_i, \mathcal{I}_j}$ for $i, j = 1, 2, \ldots, n_\mathcal{I}$.

Example

Let

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} \\ P_{31} & P_{32} & P_{33} & P_{34} & P_{35} \\ P_{41} & P_{42} & P_{43} & P_{44} & P_{45} \\ P_{51} & P_{52} & P_{53} & P_{54} & P_{55} \end{bmatrix},$$

and let $\mathcal{I} = \begin{bmatrix} 3 & 5 & 1 \end{bmatrix}$. Then,

$$\boldsymbol{\mu}_{\mathcal{I}} = \begin{bmatrix} \mu_3 \\ \mu_5 \\ \mu_1 \end{bmatrix}, \quad \mathbf{P}_{\mathcal{I}\mathcal{I}} = \begin{bmatrix} P_{33} & P_{35} & P_{31} \\ P_{53} & P_{55} & P_{51} \\ P_{13} & P_{15} & P_{11} \end{bmatrix}.$$

Selecting $\mathcal{I}$-rows and $\mathcal{I}$-columns of $\mathbf{P}$ to form $\mathbf{P}_{\mathcal{I}\mathcal{I}}$ is equivalent to selecting *all*-rows and $\mathcal{I}$-columns of $\mathbf{S}$, i.e.,

$$\mathbf{P}_{\mathcal{I}\mathcal{I}} = \mathbf{S}_{:,\mathcal{I}}^{\mathsf{T}} \mathbf{S}_{:,\mathcal{I}},$$

where $\mathbf{S}_{:,\mathcal{I}} \in \mathbb{R}^{n \times n_{\mathcal{I}}}$.

The marginal distribution of $\mathbf{x}_{\mathcal{I}}$ is given by

$$p(\mathbf{x}_{\mathcal{I}}) = \mathcal{N}(\mathbf{x}_{\mathcal{I}}; \boldsymbol{\mu}_{\mathcal{I}}, \mathbf{S}_{:,\mathcal{I}}^{\mathsf{T}} \mathbf{S}_{:,\mathcal{I}}) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}_{\mathcal{I}}; \boldsymbol{\mu}_{\mathcal{I}}, \mathbf{S}_{\mathcal{I}}), \tag{22}$$

for some upper triangular matrix $\mathbf{S}_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}} \times n_{\mathcal{I}}}$ such that $\mathbf{S}_{\mathcal{I}}^{\mathsf{T}} \mathbf{S}_{\mathcal{I}} = \mathbf{S}_{:,\mathcal{I}}^{\mathsf{T}} \mathbf{S}_{:,\mathcal{I}}$. Since $\mathbf{S}_{:,\mathcal{I}}$ is a tall matrix that is not necessarily upper triangular, we can let

$$\mathbf{S}_{:,\mathcal{I}} = \underbrace{\begin{bmatrix} \mathbf{Y} & \mathbf{Z} \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}}_{\mathbf{R}}$$

be a QR factorisation, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{R}^{n \times n_{\mathcal{I}}}$ is an upper triangular matrix. Then,

$$\begin{aligned} \mathbf{S}_{:,\mathcal{I}}^{\mathsf{T}} \mathbf{S}_{:,\mathcal{I}} &= \mathbf{R}^{\mathsf{T}} \mathbf{Q}^{\mathsf{T}} \mathbf{Q} \mathbf{R} = \mathbf{R}^{\mathsf{T}} \mathbf{R} && \text{(since } \mathbf{Q} \text{ is orthogonal)} \\ &= \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \\ &= \mathbf{R}_1^{\mathsf{T}} \mathbf{R}_1. \end{aligned}$$

Therefore, $\mathbf{S}_{\mathcal{I}} = \mathbf{R}_1$, since $\mathbf{R}_1$ is already upper triangular.

Info

Note that the closer $\mathcal{I}$ is to indexing a head partition of $\mathbf{x}$, the cheaper the QR decomposition becomes, becoming essentially free in the case where $\mathcal{I} = \begin{bmatrix} 1 & 2 & 3 & \ldots \end{bmatrix}$ exactly indexes a head partition.

Tasks

a) Complete the implementation of the `Gaussian::marginal` template function in `Gaussian.h`[1]

b) Ensure that the unit tests provided in `test/src/GaussianMarginal.cpp` pass.

4 Conditional distributions of a Gaussian (1 mark)

If we consider the same head and tail partitions as given in (15) and joint distribution (16), then the conditional distributions of the head given the tail and the tail given the head, respectively, are given as follows:

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a; \underbrace{\boldsymbol{\mu}_a + \mathbf{P}_{ab}\mathbf{P}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b)}_{\boldsymbol{\mu}_{a|b}}, \underbrace{\mathbf{P}_{aa} - \mathbf{P}_{ab}\mathbf{P}_{bb}^{-1}\mathbf{P}_{ba}}_{\mathbf{P}_{a|b}}), \tag{23a}$$

$$p(\mathbf{x}_b|\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_b; \underbrace{\boldsymbol{\mu}_b + \mathbf{P}_{ba}\mathbf{P}_{aa}^{-1}(\mathbf{x}_a - \boldsymbol{\mu}_a)}_{\boldsymbol{\mu}_{b|a}}, \underbrace{\mathbf{P}_{bb} - \mathbf{P}_{ba}\mathbf{P}_{aa}^{-1}\mathbf{P}_{ab}}_{\mathbf{P}_{b|a}}). \tag{23b}$$

From the block relationships previously established (18), we have

$$\mathbf{P}_{aa} = \mathbf{S}_1^\mathsf{T}\mathbf{S}_1, \tag{24a}$$
$$\mathbf{P}_{ab} = \mathbf{S}_1^\mathsf{T}\mathbf{S}_2, \tag{24b}$$
$$\mathbf{P}_{ba} = \mathbf{S}_2^\mathsf{T}\mathbf{S}_1, \tag{24c}$$
$$\mathbf{P}_{bb} = \mathbf{S}_2^\mathsf{T}\mathbf{S}_2 + \mathbf{S}_3^\mathsf{T}\mathbf{S}_3. \tag{24d}$$

Substituting (24c) and (24a) into (23b) yields

$$\mathbf{P}_{ba}\mathbf{P}_{aa}^{-1} = \mathbf{S}_2^\mathsf{T}\mathbf{S}_1(\mathbf{S}_1^\mathsf{T}\mathbf{S}_1)^{-1} = \mathbf{S}_2^\mathsf{T}\mathbf{S}_1\mathbf{S}_1^{-1}\mathbf{S}_1^{-\mathsf{T}} = \mathbf{S}_2^\mathsf{T}\mathbf{S}_1^{-\mathsf{T}},$$

and therefore

$$\begin{aligned}\boldsymbol{\mu}_{b|a} &= \boldsymbol{\mu}_b + \mathbf{P}_{ba}\mathbf{P}_{aa}^{-1}(\mathbf{x}_a - \boldsymbol{\mu}_a)\\ &= \boldsymbol{\mu}_b + \mathbf{S}_2^\mathsf{T}\mathbf{S}_1^{-\mathsf{T}}(\mathbf{x}_a - \boldsymbol{\mu}_a).\end{aligned}$$

Similarly, substituting (24) into (23b) yields

$$\begin{aligned}\mathbf{P}_{b|a} &= \mathbf{P}_{bb} - \mathbf{P}_{ba}\mathbf{P}_{aa}^{-1}\mathbf{P}_{ab}\\ &= \mathbf{S}_2^\mathsf{T}\mathbf{S}_2 + \mathbf{S}_3^\mathsf{T}\mathbf{S}_3 - \mathbf{S}_2^\mathsf{T}\mathbf{S}_1(\mathbf{S}_1^\mathsf{T}\mathbf{S}_1)^{-1}\mathbf{S}_1^\mathsf{T}\mathbf{S}_2\\ &= \mathbf{S}_2^\mathsf{T}\mathbf{S}_2 + \mathbf{S}_3^\mathsf{T}\mathbf{S}_3 - \mathbf{S}_2^\mathsf{T}\mathbf{S}_1\mathbf{S}_1^{-1}\mathbf{S}_1^{-\mathsf{T}}\mathbf{S}_1^\mathsf{T}\mathbf{S}_2\\ &= \mathbf{S}_2^\mathsf{T}\mathbf{S}_2 + \mathbf{S}_3^\mathsf{T}\mathbf{S}_3 - \mathbf{S}_2^\mathsf{T}\mathbf{S}_2\\ &= \mathbf{S}_3^\mathsf{T}\mathbf{S}_3.\end{aligned}$$

The conditional distribution of $\mathbf{x}_b$ given $\mathbf{x}_a$ is given by

$$p(\mathbf{x}_b|\mathbf{x}_a) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}_b; \underbrace{\boldsymbol{\mu}_b + \mathbf{S}_2^\mathsf{T}\mathbf{S}_1^{-\mathsf{T}}(\mathbf{x}_a - \boldsymbol{\mu}_a)}_{\boldsymbol{\mu}_{b|a}}, \underbrace{\mathbf{S}_3}_{\mathbf{S}_{b|a}}). \tag{25}$$

[1]Template functions are implemented in a header, since they are recipes for building functions that are needed by all compilation units that include this header.

Info

The computation of the conditional mean $\boldsymbol{\mu}_{b|a}$ and upper triangular square-root covariance $\mathbf{S}_{b|a}$ that appears in (25) involves a trivial triangular solve and extraction of the 3 non-zero blocks of $\mathbf{S}$. This is significantly cheaper than the computation involved in (23b).

To find the conditional distribution of $\mathbf{x}_a$ given $\mathbf{x}_b$, we swap the head and tail partitions as follows:

$$p\left(\begin{bmatrix}\mathbf{x}_b\\ \mathbf{x}_a\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_b\\ \mathbf{x}_a\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_b\\ \boldsymbol{\mu}_a\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{bb} & \mathbf{P}_{ba}\\ \mathbf{P}_{ab} & \mathbf{P}_{aa}\end{bmatrix}\right). \tag{26}$$

Swapping the row and column blocks in $\mathbf{P}$ is equivalent to swapping the column blocks of $\mathbf{S}$, since

$$\begin{bmatrix}\mathbf{P}_{bb} & \mathbf{P}_{ba}\\ \mathbf{P}_{ab} & \mathbf{P}_{aa}\end{bmatrix} = \begin{bmatrix}\mathbf{S}_2 & \mathbf{S}_1\\ \mathbf{S}_3 & \mathbf{0}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{S}_2 & \mathbf{S}_1\\ \mathbf{S}_3 & \mathbf{0}\end{bmatrix}.$$

Since swapping the block columns of $\mathbf{S}$ leads to a non-upper-triangular matrix, let

$$\begin{bmatrix}\mathbf{S}_2 & \mathbf{S}_1\\ \mathbf{S}_3 & \mathbf{0}\end{bmatrix} = \mathbf{Q}\underbrace{\begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}}_{\mathbf{R}}$$

be a QR factorisation, where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{R}^{n\times n}$ is an upper triangular matrix with blocks $\mathbf{R}_1 \in \mathbb{R}^{n_b\times n_b}$, $\mathbf{R}_2 \in \mathbb{R}^{n_b\times n_a}$ and $\mathbf{R}_3 \in \mathbb{R}^{n_a\times n_a}$, where $\mathbf{R}_1$ and $\mathbf{R}_3$ are upper triangular. Then,

$$\begin{aligned}\begin{bmatrix}\mathbf{S}_2 & \mathbf{S}_1\\ \mathbf{S}_3 & \mathbf{0}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{S}_2 & \mathbf{S}_1\\ \mathbf{S}_3 & \mathbf{0}\end{bmatrix} &= \mathbf{R}^\mathsf{T}\mathbf{Q}^\mathsf{T}\mathbf{Q}\mathbf{R} = \mathbf{R}^\mathsf{T}\mathbf{R} && \text{(since } \mathbf{Q} \text{ is orthogonal)}\\ &= \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}.\end{aligned}$$

Therefore, the partition swapped distribution can be parameterised with an upper triangular square-root covariance as follows:

$$p\left(\begin{bmatrix}\mathbf{x}_b\\ \mathbf{x}_a\end{bmatrix}\right) = \mathcal{N}^{\frac{1}{2}}\left(\begin{bmatrix}\mathbf{x}_b\\ \mathbf{x}_a\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_b\\ \boldsymbol{\mu}_a\end{bmatrix}, \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}\right). \tag{27}$$

Finally, the conditional distribution of $\mathbf{x}_a$ given $\mathbf{x}_b$ is given by the following:

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}_a; \underbrace{\boldsymbol{\mu}_a + \mathbf{R}_2^\mathsf{T}\mathbf{R}_1^{-\mathsf{T}}(\mathbf{x}_b - \boldsymbol{\mu}_b)}_{\boldsymbol{\mu}_{a|b}}, \underbrace{\mathbf{R}_3}_{\mathbf{S}_{a|b}}). \tag{28}$$

As we have seen, when working with an upper-triangular square-root factorisation of a joint covariance matrix, it is much easier to compute the conditional distribution of the tail given the head (25) rather than than the conditional distribution of the head given the tail (28). The former requires only a simple triangular solve using the upper left block of $\mathbf{S}$ and extraction of the lower right block of $\mathbf{S}$, whereas the latter requires a extra computation to upper triangularise the block column swapped $\mathbf{S}$ via a Q-less QR decomposition.

So far, we have only considered the conditional distributions of simple partitions of $\mathbf{x}$. To find the conditional distributions of an arbitrary subset of $\mathbf{x}$ given another complementary subset of $\mathbf{x}$, let us introduce the index vectors $\mathcal{A} \in \mathbb{N}^{n_\mathcal{A}}$ and $\mathcal{B} \in \mathbb{N}^{n_\mathcal{B}}$ to select the desired elements of each set and denote these as $\mathbf{x}_\mathcal{A}$ and $\mathbf{x}_\mathcal{B}$, respectively.

The conditional distribution of $\mathbf{x}_\mathcal{A}$ given $\mathbf{x}_\mathcal{B}$ is given by

$$p(\mathbf{x}_\mathcal{A}|\mathbf{x}_\mathcal{B}) = \mathcal{N}(\mathbf{x}_\mathcal{A}; \boldsymbol{\mu}_{\mathcal{A}|\mathcal{B}}, \mathbf{P}_{\mathcal{A}|\mathcal{B}}),$$

where

$$\begin{aligned}\boldsymbol{\mu}_{\mathcal{A}|\mathcal{B}} &= \boldsymbol{\mu}_\mathcal{A} + \mathbf{P}_{\mathcal{A}\mathcal{B}}\mathbf{P}_{\mathcal{B}\mathcal{B}}^{-1}(\mathbf{x}_\mathcal{B} - \boldsymbol{\mu}_\mathcal{B}), \\ \mathbf{P}_{\mathcal{A}|\mathcal{B}} &= \mathbf{P}_{\mathcal{A}\mathcal{A}} - \mathbf{P}_{\mathcal{A}\mathcal{B}}\mathbf{P}_{\mathcal{B}\mathcal{B}}^{-1}\mathbf{P}_{\mathcal{B}\mathcal{A}}.\end{aligned}$$

To obtain this conditional distribution in square-root form, we permute the elements as follows:

$$p\left(\begin{bmatrix}\mathbf{x}_\mathcal{B}\\ \mathbf{x}_\mathcal{A}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_\mathcal{B}\\ \mathbf{x}_\mathcal{A}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_\mathcal{B}\\ \boldsymbol{\mu}_\mathcal{A}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{\mathcal{B}\mathcal{B}} & \mathbf{P}_{\mathcal{B}\mathcal{A}}\\ \mathbf{P}_{\mathcal{A}\mathcal{B}} & \mathbf{P}_{\mathcal{A}\mathcal{A}}\end{bmatrix}\right). \tag{29}$$

Selecting $\mathcal{B}$ and $\mathcal{A}$ rows and columns of $\mathbf{P}$ is equivalent to selecting the $\mathcal{B}$ and $\mathcal{A}$ *columns* of $\mathbf{S}$, i.e.,

$$\begin{bmatrix}\mathbf{P}_{\mathcal{B}\mathcal{B}} & \mathbf{P}_{\mathcal{B}\mathcal{A}}\\ \mathbf{P}_{\mathcal{A}\mathcal{B}} & \mathbf{P}_{\mathcal{A}\mathcal{A}}\end{bmatrix} = \begin{bmatrix}\mathbf{S}_{:,\mathcal{B}} & \mathbf{S}_{:,\mathcal{A}}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{S}_{:,\mathcal{B}} & \mathbf{S}_{:,\mathcal{A}}\end{bmatrix},$$

where $\mathbf{S}_{:,\mathcal{A}} \in \mathbb{R}^{n\times n_\mathcal{A}}$ and $\mathbf{S}_{:,\mathcal{B}} \in \mathbb{R}^{n\times n_\mathcal{B}}$.

Since $\begin{bmatrix}\mathbf{S}_{:,\mathcal{B}} & \mathbf{S}_{:,\mathcal{A}}\end{bmatrix}$ is not necessarily upper triangular, we can let

$$\begin{bmatrix}\mathbf{S}_{:,\mathcal{B}} & \mathbf{S}_{:,\mathcal{A}}\end{bmatrix} = \mathbf{Q}\underbrace{\begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}}_{\mathbf{R}}$$

be a QR factorisation, where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{R}^{n\times n}$ is an upper triangular matrix with blocks $\mathbf{R}_1 \in \mathbb{R}^{n_\mathcal{B}\times n_\mathcal{B}}$, $\mathbf{R}_2 \in \mathbb{R}^{n_\mathcal{B}\times n_\mathcal{A}}$ and $\mathbf{R}_3 \in \mathbb{R}^{n_\mathcal{A}\times n_\mathcal{A}}$, where $\mathbf{R}_1$ and $\mathbf{R}_3$ are upper triangular. Then,

$$\begin{aligned}\begin{bmatrix}\mathbf{S}_{:,\mathcal{B}} & \mathbf{S}_{:,\mathcal{A}}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{S}_{:,\mathcal{B}} & \mathbf{S}_{:,\mathcal{A}}\end{bmatrix} &= \mathbf{R}^\mathsf{T}\mathbf{Q}^\mathsf{T}\mathbf{Q}\mathbf{R} = \mathbf{R}^\mathsf{T}\mathbf{R} && \text{(since } \mathbf{Q} \text{ is orthogonal)}\\ &= \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}^\mathsf{T}\begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}.\end{aligned}$$

$$p\left(\begin{bmatrix}\mathbf{x}_\mathcal{B}\\ \mathbf{x}_\mathcal{A}\end{bmatrix}\right) = \mathcal{N}^{\frac{1}{2}}\left(\begin{bmatrix}\mathbf{x}_\mathcal{B}\\ \mathbf{x}_\mathcal{A}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_\mathcal{B}\\ \boldsymbol{\mu}_\mathcal{A}\end{bmatrix}, \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0} & \mathbf{R}_3\end{bmatrix}\right). \tag{30}$$

Finally, the conditional distribution of $\mathbf{x}_\mathcal{A}$ given $\mathbf{x}_\mathcal{B}$ is given by the following:

$$p(\mathbf{x}_\mathcal{A}|\mathbf{x}_\mathcal{B}) = \mathcal{N}^{\frac{1}{2}}(\mathbf{x}_\mathcal{A}; \underbrace{\boldsymbol{\mu}_\mathcal{A} + \mathbf{R}_2^\mathsf{T}\mathbf{R}_1^{-\mathsf{T}}(\mathbf{x}_\mathcal{B} - \boldsymbol{\mu}_\mathcal{B})}_{\boldsymbol{\mu}_{\mathcal{A}|\mathcal{B}}}, \underbrace{\mathbf{R}_3}_{\mathbf{S}_{\mathcal{A}|\mathcal{B}}}). \tag{31}$$

Info

Note that the closer $\mathcal{A}$ is to indexing a tail partition of $\mathbf{x}$ and the closer $\mathcal{B}$ is to indexing a head partition, the cheaper the QR decomposition becomes, becoming essentially free in the case where $\mathcal{A}$ and $\mathcal{B}$ exactly index tail and head partitions, respectively.

Tasks

a) Complete the implementation of the `Gaussian::conditional` template function in `Gaussian.h`

b) Ensure that the unit tests provided in `test/src/GaussianConditional.cpp` pass.

5 State estimation and visualisation (1 mark)

The framework for an event-based square-root extended Kalman filter has been implemented in the `Gaussian`, `State`, `Event` and `Measurement` classes (see the `.h` and `.cpp` files for each). The `State` and `Measurement` classes are extended to `StateBallistic` and `MeasurementRADAR`, respectively, to provide the problem-specific implementations of the process model dynamics and measurement model. More formally, the `StateBallistic` class inherits from the `State` class, while the `MeasurementRADAR` class inherits from the `Measurement` class, which in turn inherits from `Event` class.

The main loop for each time step of the state estimator is implemented in `src/main.cpp`, which creates each measurement event and then processes it. Measurement events are processed by performing a time update on the state to the time stamp of the measurement followed by a measurement update. This can be seen in the `Event::process`, `State::predict` and `Measurement::update` functions.

The results are plotted in the `plot_simulation` function in `src/ballistic_plot.cpp` using Visualisation Toolkit (VTK), which is a powerful 2D and 3D visualisation library.

Tasks

a) Run the `lab5` application and ensure that you get the following output:

Terminal

```
nerd@basement:~/MCHA4400/lab5/build$ ninja && ./lab5
[Ninja-ing intensifies]
Reading data from ../data/estimationdata.csv
Found 501 rows within ../data/estimationdata.csv

Initial state estimate
mu[0] =
 14000
  -450
0.0005
P[0] =
4.84e+06        0        0
       0    10000        0
       0        0    1e-06
Run filter with 500 steps.

Final state estimate
mu[end] =
    2185.37
   -152.739
0.000982013
P[end] =
    215.265     10.7372 0.000163342
    10.7372     1.56802 2.54573e-05
0.000163342 2.54573e-05   4.791e-10
[A VTK plot window will display estimation results with a missing subplot]
```

b) Complete the implementation of the `plot_simulation` function, to produce a figure similar to Figure 2. The plot should include:

i) the true state of the ballistic coefficient,

ii) the estimated state of the ballistic coefficient,

iii) and the estimated 99.7% confidence region of the ballistic coefficient.

Tip

The VTK C++ examples are a useful resource for getting started with VTK. Some of the following API documentation may also be useful:

- vtkChartXY, for drawing 2-dimensional charts
- vtkPlot, for plotting lines within a chart
- vtkPlotArea, for plotting areas between two lines
- vtkChartLegend, for modifying legends of chart objects
- setLabel, for modifying labels of vtkPlot and vtkPlotArea objects.

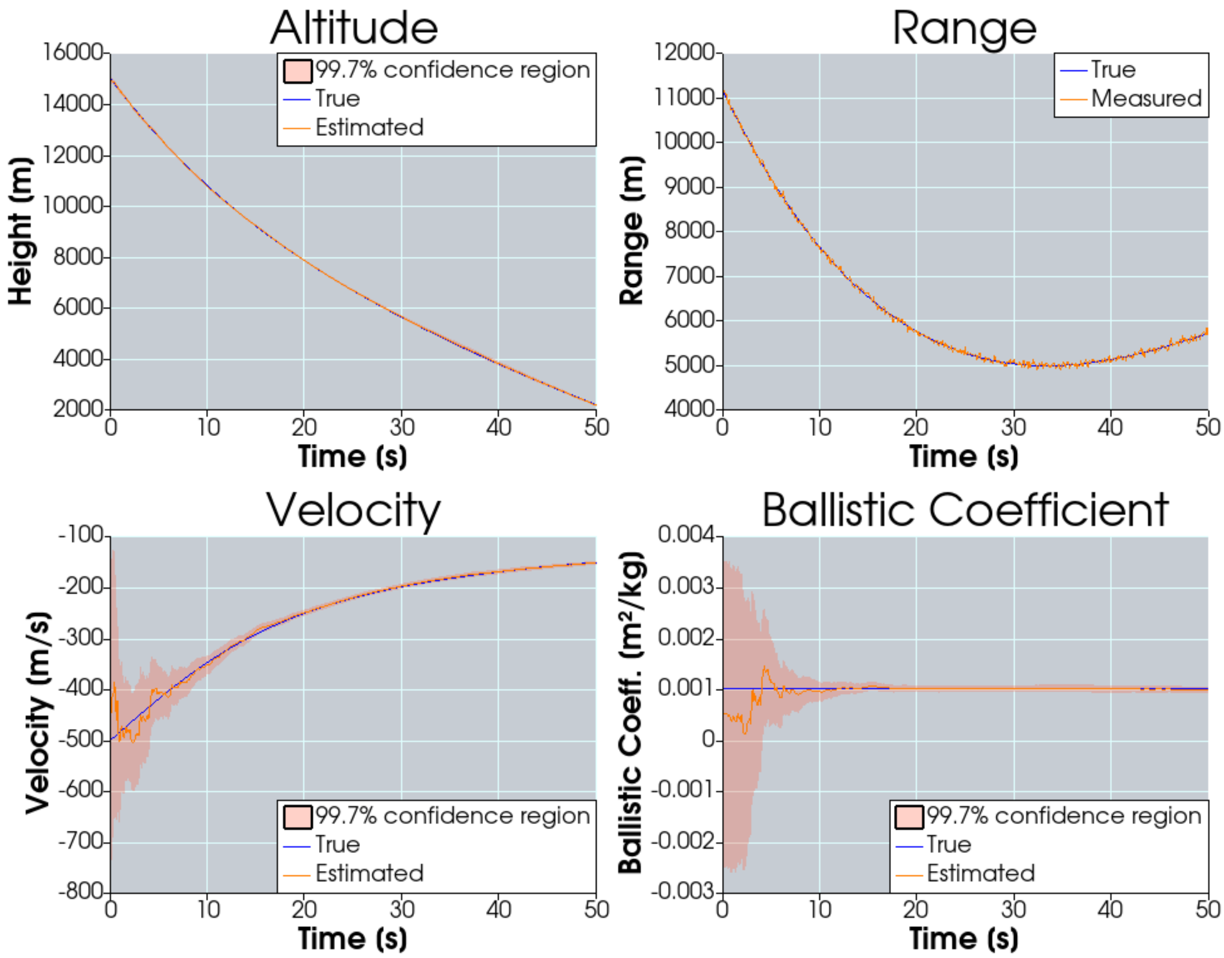

Figure 2: Expected plot.