Batch State Estimation
— Using All Available Data for Estimation —

Charles C. Cossette and Prof. James Richard Forbes

McGill University, Department of Mechanical Engineering

November 7, 2022

Problem Statement

- Consider the following process and measurement models,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \qquad k = 1, \ldots, K \tag{1}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k, \qquad k = 0, \ldots, K. \tag{2}$$

Problem Statement

- Consider the following process and measurement models,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \qquad k = 1, \ldots, K \tag{1}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k, \qquad k = 0, \ldots, K. \tag{2}$$

- Suppose we have access to **all** the inputs $\mathbf{u}_k$ and outputs $\mathbf{y}_k$, as well as an uncertain estimate of the initial state,

$$\mathbf{x}_0 \sim \mathcal{N}(\check{\mathbf{x}}_0, \mathbf{P}_0). \tag{3}$$

- How do we find the “best” estimate of the **all of states**, all at once, using all the data available?

Problem Statement

- Consider the following process and measurement models,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \qquad k = 1, \dots, K \tag{1}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k, \qquad k = 0, \dots, K. \tag{2}$$

- Suppose we have access to **all** the inputs $\mathbf{u}_k$ and outputs $\mathbf{y}_k$, as well as an uncertain estimate of the initial state,

$$\mathbf{x}_0 \sim \mathcal{N}(\check{\mathbf{x}}_0, \mathbf{P}_0). \tag{3}$$

- How do we find the "best" estimate of the **all of states**, all at once, using all the data available?
- This is the *batch state estimation* problem.
- The notation

$$\mathbf{x} = \mathbf{x}_{0:K} = \{\mathbf{x}_0, \dots, \mathbf{x}_K\},$$
$$\mathbf{u} = \mathbf{u}_{0:K} = \{\mathbf{u}_0, \dots, \mathbf{u}_K\},$$
$$\mathbf{y} = \mathbf{y}_{0:K} = \{\mathbf{y}_0, \dots, \mathbf{y}_K\},$$

will be used.

Batch Estimation

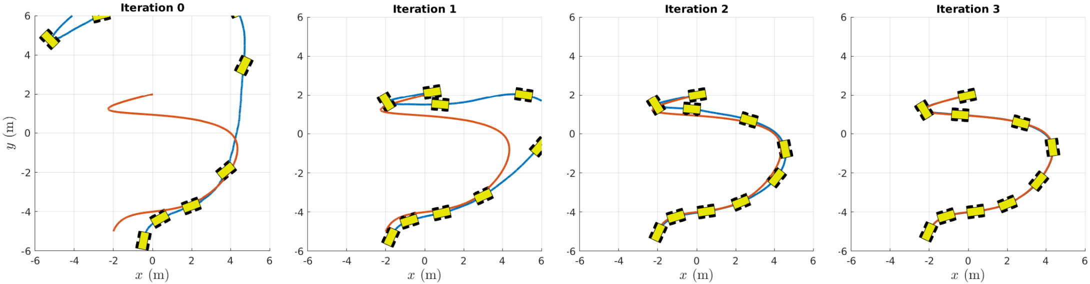
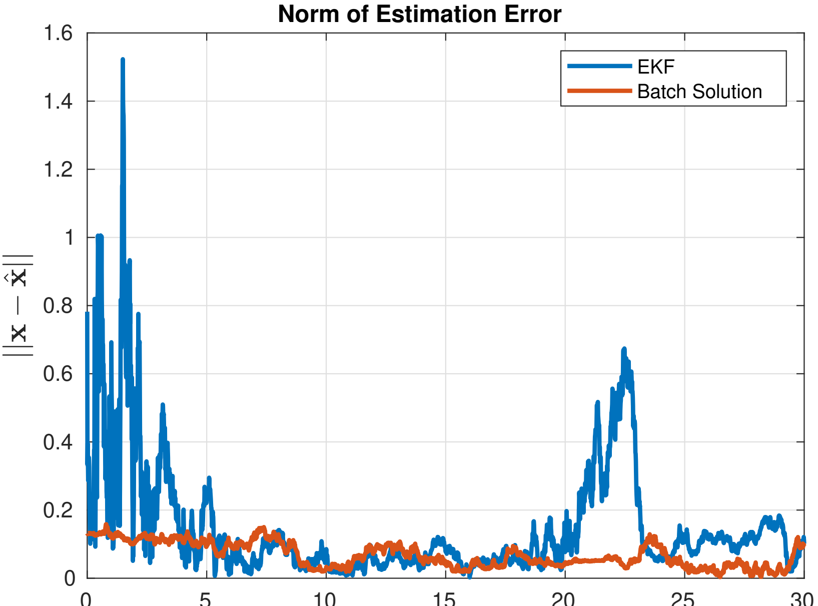


Figure 1: (red) Ground truth trajectory. (blue) Estimated trajectory.
Simulation of the estimation of a ground robot's trajectory using batch estimation.

Batch Estimation vs. Extended Kalman Filter



Maximum A Posteriori

- One strategy is to find the *maximum a posteriori* estimate, which is the solution to

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}). \tag{4}$$

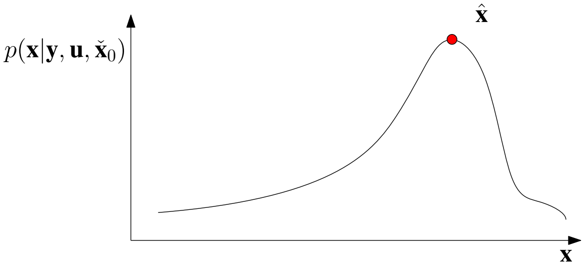


Figure 2: The MAP estimate finds the largest overall value of $p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y})$.

Maximum A Posteriori

- One strategy is to find the *maximum a posteriori* estimate, which is the solution to

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}). \tag{4}$$

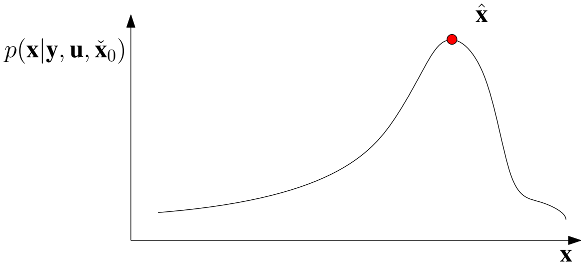


Figure 2: The MAP estimate finds the largest overall value of $p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y})$.

- Note that this is the **mode** of the distribution, as opposed to the mean.

Maximum A Posteriori

- One strategy is to find the *maximum a posteriori* estimate, which is the solution to

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}). \tag{4}$$

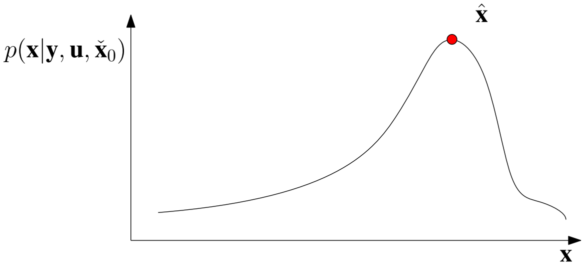


Figure 2: The MAP estimate finds the largest overall value of $p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y})$.

- Note that this is the **mode** of the distribution, as opposed to the mean.
- The next few steps consist of manipulating $p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y})$ into a form so that a gradient-based optimization algorithm (i.e., Gauss-Newton) can be used.

Maximum A Posteriori

- Bayes' rule can be used to reformat the problem,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \frac{p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u})}{p(\mathbf{y}|\check{\mathbf{x}}_0, \mathbf{u})} \quad (5)$$

$$= \arg\max_{\mathbf{x}} \alpha p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}), \quad (6)$$

where the denominator has been lumped into a constant α, which does not depend on $\mathbf{x}$.

Maximum A Posteriori

- Bayes' rule can be used to reformat the problem,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \frac{p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u})}{p(\mathbf{y}|\check{\mathbf{x}}_0, \mathbf{u})} \tag{5}$$

$$= \arg\max_{\mathbf{x}} \alpha p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}), \tag{6}$$

  where the denominator has been lumped into a constant α, which does not depend on $\mathbf{x}$.
- Two simplifying assumptions will be made.

Maximum A Posteriori

- Bayes' rule can be used to reformat the problem,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \frac{p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u})}{p(\mathbf{y}|\check{\mathbf{x}}_0, \mathbf{u})} \tag{5}$$

$$= \arg\max_{\mathbf{x}} \alpha p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}), \tag{6}$$

  where the denominator has been lumped into a constant α, which does not depend on $\mathbf{x}$.

- Two simplifying assumptions will be made.
  1. From our measurement model $\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k$, we can write

$$p(\mathbf{y}_k|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u}) = p(\mathbf{y}_k|\mathbf{x}_k) \tag{7}$$

     since $\mathbf{y}_k$ is only conditioned on $\mathbf{x}_k$.

Maximum A Posteriori

- Bayes' rule can be used to reformat the problem,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \frac{p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u})}{p(\mathbf{y}|\check{\mathbf{x}}_0, \mathbf{u})} \tag{5}$$

$$= \arg\max_{\mathbf{x}} \alpha p(\mathbf{y}|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u})p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}), \tag{6}$$

  where the denominator has been lumped into a constant α, which does not depend on $\mathbf{x}$.
- Two simplifying assumptions will be made.
  1. From our measurement model $\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k$, we can write

     $$p(\mathbf{y}_k|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u}) = p(\mathbf{y}_k|\mathbf{x}_k) \tag{7}$$

     since $\mathbf{y}_k$ is only conditioned on $\mathbf{x}_k$.
  2. From our process model, we can write

     $$p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{u}_{0:K}, \check{\mathbf{x}}_0) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \tag{8}$$

     since $\mathbf{x}_k$ is only conditioned on $\mathbf{x}_{k-1}, \mathbf{u}_{k-1}$ (the *Markov* assumption).

Factored Joint Likelihood

- These assumptions allow us to "split" the PDFs into their *factored joint likehoods*

$$p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) = p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K})$$

Factored Joint Likelihood

- These assumptions allow us to "split" the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K})
\end{aligned}
$$

Factored Joint Likelihood

- These assumptions allow us to "split" the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K})
\end{aligned}
$$

Factored Joint Likelihood

- These assumptions allow us to "split" the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k).
\end{aligned}
$$

Factored Joint Likelihood

- These assumptions allow us to “split” the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k).
\end{aligned}
$$

Factored Joint Likelihood

- These assumptions allow us to “split” the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k).
\end{aligned}
$$

- Similarly,

$$
p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}) = p(\mathbf{x}_0|\mathbf{x}_{1:K}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K})
$$

.

Factored Joint Likelihood

- These assumptions allow us to “split” the PDFs into their *factored joint likehoods*

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\ &= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\ &= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\ &= \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k). \end{aligned}$$

- Similarly,

$$\begin{aligned} p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}) &= p(\mathbf{x}_0|\mathbf{x}_{1:K}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \\ &= p(\mathbf{x}_0|\check{\mathbf{x}}_0)p(\mathbf{x}_K|\mathbf{x}_{1:K-1}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K-1}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \end{aligned}$$

.

Factored Joint Likelihood

- These assumptions allow us to "split" the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k).
\end{aligned}
$$

- Similarly,

$$
\begin{aligned}
p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}) &= p(\mathbf{x}_0|\mathbf{x}_{1:K}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \\
&= p(\mathbf{x}_0|\check{\mathbf{x}}_0)p(\mathbf{x}_K|\mathbf{x}_{1:K-1}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K-1}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \\
&= p(\mathbf{x}_0|\check{\mathbf{x}}_0)p(\mathbf{x}_K|\mathbf{x}_{K-1}, \mathbf{u}_{K-1})p(\mathbf{x}_{1:K-1}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K})
\end{aligned}
$$

.

Factored Joint Likelihood

- These assumptions allow us to “split” the PDFs into their *factored joint likehoods*

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K}) &= p(\mathbf{y}_0, \mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{y}_{1:K}, \mathbf{x}_{0:K})p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}) \\
&= \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k).
\end{aligned}
$$

- Similarly,

$$
\begin{aligned}
p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}) &= p(\mathbf{x}_0|\mathbf{x}_{1:K}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \\
&= p(\mathbf{x}_0|\check{\mathbf{x}}_0)p(\mathbf{x}_K|\mathbf{x}_{1:K-1}, \check{\mathbf{x}}_0, \mathbf{u}_{0:K})p(\mathbf{x}_{1:K-1}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \\
&= p(\mathbf{x}_0|\check{\mathbf{x}}_0)p(\mathbf{x}_K|\mathbf{x}_{K-1}, \mathbf{u}_{K-1})p(\mathbf{x}_{1:K-1}|\check{\mathbf{x}}_0, \mathbf{u}_{0:K}) \\
&= p(\mathbf{x}_0|\check{\mathbf{x}}_0)\prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}).
\end{aligned}
$$

Maximum A Posteriori

- Returning to the optimization problem, it can now be written as

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \alpha p(\mathbf{x}_0 | \check{\mathbf{x}}_0) \left(\prod_{k=1}^{K} p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \right) \left(\prod_{k=0}^{K} p(\mathbf{y}_k | \mathbf{x}_k) \right). \tag{9}$$

Maximum A Posteriori

- Returning to the optimization problem, it can now be written as

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \alpha p(\mathbf{x}_0 | \check{\mathbf{x}}_0) \left(\prod_{k=1}^{K} p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \right) \left(\prod_{k=0}^{K} p(\mathbf{y}_k | \mathbf{x}_k) \right). \tag{9}$$

- Minimizing the negative logarithm of (9) does not change the solution to the optimization problem, as it is a monotonically increasing function.
- Hence,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} -\ln \left(\alpha p(\mathbf{x}_0 | \check{\mathbf{x}}_0) \left(\prod_{k=1}^{K} p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \right) \left(\prod_{k=0}^{K} p(\mathbf{y}_k | \mathbf{x}_k) \right) \right) \tag{10}$$

$$= \arg\min_{\mathbf{x}} -\ln \alpha - \ln p(\mathbf{x}_0 | \check{\mathbf{x}}_0) - \sum_{k=1}^{K} \ln p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) - \sum_{k=0}^{K} \ln p(\mathbf{y}_k | \mathbf{x}_k). \tag{11}$$

Minimizing the Negative Logarithm

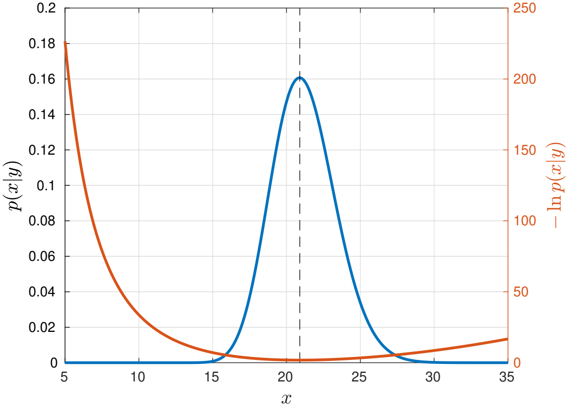


Figure 3: The maximum of $p(x|y)$ is at the same x value as the minimum of $-\ln p(x|y)$. [1]

Using Gaussian Error Distributions

- The problem simplifies further if the probability density functions in (11) are assumed to be Gaussian distributions,

$$
\begin{aligned}
p(\mathbf{x}_0|\check{\mathbf{x}}_0) &= \frac{1}{\sqrt{(2\pi)^n \det \mathbf{P}_0}} \exp\left(-\frac{1}{2}(\mathbf{x}_0 - \check{\mathbf{x}}_0)^\mathsf{T}\mathbf{P}_0^{-1}(\mathbf{x}_0 - \check{\mathbf{x}}_0)\right), \\
p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) &= \frac{1}{\sqrt{(2\pi)^n \det \mathbf{Q}_k}} \\
&\quad \times \exp\left(-\frac{1}{2}(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})^\mathsf{T}\mathbf{Q}_k^{-1}(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}))\right), \\
p(\mathbf{y}_k|\mathbf{x}_k) &= \frac{1}{\sqrt{(2\pi)^p \det \mathbf{R}_k}} \exp\left(-\frac{1}{2}(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k))^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k))\right).
\end{aligned}
$$

Using Gaussian Error Distributions

- The cost function then becomes

$$
\begin{aligned}
\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} & - \ln \alpha - \ln \frac{1}{\sqrt{(2\pi)^n \det \mathbf{P}_0}} - \ln \frac{1}{\sqrt{(2\pi)^n \det \mathbf{Q}_k}} - \ln \frac{1}{\sqrt{(2\pi)^n \det \mathbf{R}_k}} \\
& + \left(\frac{1}{2} (\mathbf{x}_0 - \check{\mathbf{x}}_0)^\mathsf{T} \mathbf{P}_0^{-1} (\mathbf{x}_0 - \check{\mathbf{x}}_0) \right) \\
& + \sum_{k=1}^{K} \left(\frac{1}{2} (\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}))^\mathsf{T} \mathbf{Q}_k^{-1} (\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})) \right) \\
& + \sum_{k=0}^{K} \left(\frac{1}{2} (\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k))^\mathsf{T} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k)) \right).
\end{aligned}
$$

- The first four terms are independent of $\mathbf{x}$, and can be lumped into a single constant α.

Using Gaussian Error Distributions

- Finally, by defining

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_0(\mathbf{x}) \\ \mathbf{e}_{u,1}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{u,K}(\mathbf{x}) \\ \mathbf{e}_{y,0}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{y,K}(\mathbf{x}) \end{bmatrix}, \quad \text{where} \quad \begin{aligned} \mathbf{e}_0(\mathbf{x}) &= \mathbf{x}_0 - \check{\mathbf{x}}_0, \\ \mathbf{e}_{u,k}(\mathbf{x}) &= \mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}), \\ \mathbf{e}_{y,k}(\mathbf{x}) &= \mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{0}), \end{aligned}$$

$$\mathbf{W} = \mathrm{diag}(\mathbf{P}_0^{-1}, \mathbf{Q}_1^{-1}, \ldots, \mathbf{Q}_K^{-1}, \mathbf{R}_0^{-1}, \ldots, \mathbf{R}_K^{-1}),$$

  the optimization problem becomes

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \mathbf{e}(\mathbf{x})^{\mathsf{T}} \mathbf{W} \mathbf{e}(\mathbf{x}) + \alpha, \tag{12}$$

  which is a weighted nonlinear least squares problem!

- Drop the α term.

Summary of MAP Estimation

Maximum A Posteriori

In summary, the optimization problem

$$\hat{\mathbf{x}} = \arg\max p(\mathbf{x}|\mathbf{x}_0, \mathbf{u}, \mathbf{y}) \tag{13}$$

is completely equivalent to

$$\hat{\mathbf{x}} = \arg\min \frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathsf{T}}\mathbf{W}\mathbf{e}(\mathbf{x}), \tag{14}$$

where $\mathbf{e}(\mathbf{x}), \mathbf{W}$ have been defined previously,

1. we assume that $p(\mathbf{y}_k|\mathbf{x}, \check{\mathbf{x}}_0, \mathbf{u}) = p(\mathbf{y}_k|\mathbf{x}_k)$,
2. we assume that $p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{u}_{1:K}, \check{\mathbf{x}}_0) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$, and
3. we assume that the PDFs $p(\mathbf{x}_0|\check{\mathbf{x}}_0)$, $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$, $p(\mathbf{y}_k|\mathbf{x}_k)$ are Gaussian.

An Aside on Matrix Construction

- When constructing the batch matrices, **the order of states and errors is arbitrary**.
- It is equally mathematically valid to choose any ordering, so long as the construction of the matrices is consistent with the ordering.
- Some orderings provide computational benefits (sparsity in matrices).

$$\mathbf{x} = \{\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1}\}$$

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{u,k-1}(\mathbf{x}) \\ \mathbf{e}_{u,k}(\mathbf{x}) \\ \mathbf{e}_{y,k-1}(\mathbf{x}) \\ \mathbf{e}_{y,k}(\mathbf{x}) \\ \mathbf{e}_{y,k+1}(\mathbf{x}) \end{bmatrix}$$

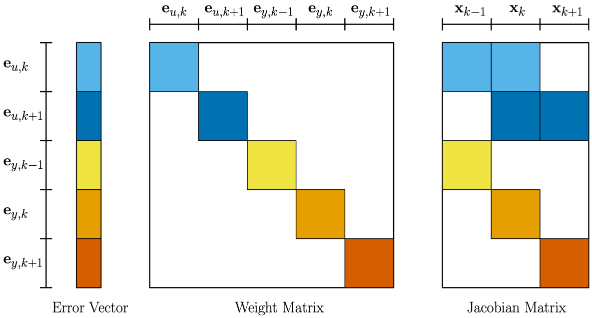


Figure 4: One choice of state and error ordering.

An Aside on Matrix Construction

- When constructing the batch matrices, **the order of states and errors is arbitrary**.
- It is equally mathematically valid to choose any ordering, so long as the construction of the matrices is consistent with the ordering.
- Some orderings provide computational benefits (sparsity in matrices).

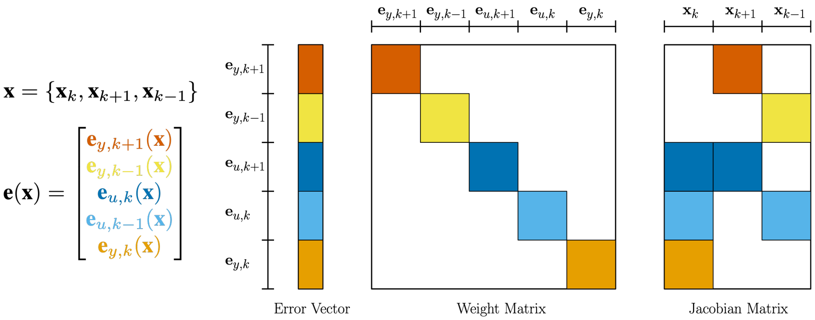


Figure 5: Another choice of state and error ordering.

Solving the Nonlinear Case

- We must use iterative nonlinear least-squares algorithms, such as the *Gauss-Newton* algorithm.

  1. Start with an initial guess for $\mathbf{x}^{(0)}$,

Solving the Nonlinear Case

- We must use iterative nonlinear least-squares algorithms, such as the *Gauss-Newton* algorithm.
  1. Start with an initial guess for $\mathbf{x}^{(0)}$,
  2. compute the Jacobian of the error

$$\mathbf{H} = \left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(i)}}, \tag{15}$$

Solving the Nonlinear Case

- We must use iterative nonlinear least-squares algorithms, such as the *Gauss-Newton* algorithm.

1. Start with an initial guess for $\mathbf{x}^{(0)}$,
2. compute the Jacobian of the error

$$\mathbf{H} = \left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(i)}}, \tag{15}$$

3. compute the Gauss-Newton step

$$\delta \mathbf{x}^{(i)} = -(\mathbf{H}^{\mathsf{T}} \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^{\mathsf{T}} \mathbf{W} \mathbf{e}(\mathbf{x}), \tag{16}$$

4. update the estimate

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha \delta \mathbf{x}^{(i)}, \tag{17}$$

5. and repeat until convergence.

Solving the Nonlinear Case

- We must use iterative nonlinear least-squares algorithms, such as the *Gauss-Newton* algorithm.
  1. Start with an initial guess for $\mathbf{x}^{(0)}$,
  2. compute the Jacobian of the error
  $$\mathbf{H} = \left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(i)}}, \tag{15}$$
  3. compute the Gauss-Newton step
  $$\delta\mathbf{x}^{(i)} = -(\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{e}(\mathbf{x}), \tag{16}$$
  4. update the estimate
  $$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha\delta\mathbf{x}^{(i)}, \tag{17}$$
  5. and repeat until convergence.
- α is a step size (can be chosen with line search).
- Could also use Levenberg–Marquardt.

The Linear Case

- Given linear process and measurement models,

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{18}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{19}$$

it follows that the error matrix $\mathbf{e}(\mathbf{x})$ can be written as

$$\mathbf{e}(\mathbf{x}) = \mathbf{H}\mathbf{x} - \mathbf{z} \tag{20}$$

where $\mathbf{x} = [\mathbf{x}_0^\mathsf{T} \ \ldots \ \mathbf{x}_K^\mathsf{T}]^\mathsf{T}$,

$$\mathbf{z} = \begin{bmatrix} \check{\mathbf{x}}_0 \\ \mathbf{B}_0\mathbf{u}_0 \\ \vdots \\ \mathbf{B}_{K-1}\mathbf{u}_{K-1} \\ \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} \mathbf{1} & & & \\ -\mathbf{A}_0 & \mathbf{1} & & \\ & \ddots & \ddots & \\ & & -\mathbf{A}_{K-1} & \mathbf{1} \\ -\mathbf{C}_0 & & & \\ & -\mathbf{C}_1 & & \\ & & \ddots & \\ & & & \mathbf{C}_K \end{bmatrix}. \tag{21}$$

The Linear Case

- The Gauss-Newton step becomes

$$\delta\mathbf{x}^{(i)} = -(\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{e}(\mathbf{x}^{(i)}), \tag{22}$$

$$= -(\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}(\mathbf{H}\mathbf{x}^{(i)} - \mathbf{z}), \tag{23}$$

$$= -\mathbf{x}^{(i)} + (\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{z} \tag{24}$$

- The iterations $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \delta\mathbf{x}^{(i)}$ then reduce to a single solution for the optimal estimate

$$\hat{\mathbf{x}} = (\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{z}. \tag{25}$$

Starting with a Continuous-Time Model

- Suppose that we instead have a continuous time process model $\boldsymbol{f}(\cdot)$ where

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \qquad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}(t)), \tag{26}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k). \tag{27}$$

- We can linearize about some trajectory $\mathbf{x}(t) = \bar{\mathbf{x}}(t) + \delta\mathbf{x}(t)$, $\mathbf{w}(t) = \mathbf{0} + \delta\mathbf{w}(t)$, $\mathbf{y}_k = \mathbf{g}(\bar{\mathbf{x}}_k) + \delta\mathbf{y}_k$ to create a linear approximation for the perturbation dynamics

$$\delta\dot{\mathbf{x}}(t) = \mathbf{A}(t)\delta\mathbf{x}(t) + \mathbf{L}(t)\delta\mathbf{w}(t) \qquad \delta\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}(t)), \tag{28}$$

$$\delta\mathbf{y}_k = \mathbf{C}_k\delta\mathbf{x}_k + \mathbf{v}_k \qquad \delta\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k). \tag{29}$$

- Using a discretization scheme (zero-order-hold), we can create a discrete time equivalent model

$$\delta\mathbf{x}_k = \mathbf{A}_{k-1}\delta\mathbf{x}_{k-1} + \delta\mathbf{w}_{k-1} \qquad \delta\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}), \tag{30}$$

$$\delta\mathbf{y}_k = \mathbf{C}_k\delta\mathbf{x}_k + \delta\mathbf{v}_k \qquad \delta\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{31}$$

where $\delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{f}(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0})$ and $\delta\mathbf{y}_k = \mathbf{y}_k - \mathbf{g}(\bar{\mathbf{x}}_k)$.

Starting with a Continuous-Time Model

- To proceed with the batch MAP framework, we set the linearization points to simply be our current best state estimate $\bar{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}_{k-1}^{(i)}$ at iteration i.
- The state is given by

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \delta\mathbf{x}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{A}_{k-1}\delta\mathbf{x}_{k-1} + \delta\mathbf{w}_{k-1} \quad (32)$$

$$= \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{A}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^{(i)}) + \delta\mathbf{w}_{k-1} \quad (33)$$

$$= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \underbrace{\mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}) - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1}^{(i)}}_{\triangleq \boldsymbol{u}_{k-1}} + \mathbf{w}_{k-1} \quad (34)$$

and so it follows that $\mathbf{x}_k \sim \mathcal{N}(\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \boldsymbol{u}_{k-1}, \mathbf{Q}_{k-1})$.

- This produces a linear batch problem with the error written as $\mathbf{e}(\mathbf{x}) = \mathbf{H}\mathbf{x} - \mathbf{z}$, and as usual,

$$\delta\hat{\mathbf{x}}^{(i)} = -(\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{e}(\hat{\mathbf{x}}^{(i)}) \quad (35)$$

$$\hat{\mathbf{x}}^{(i+1)} = \hat{\mathbf{x}}^{(i)} + \alpha\delta\hat{\mathbf{x}}^{(i)} \quad (36)$$

- We then recompute (35) at the new state estimate, and this is repeated until convergence.

Estimate Mean and Covariance

- The solution to our optimization problem gave us the *mode* of our state distribution, $p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y})$.
- It is useful to also know its *mean* and *covariance*.
- For this, it is more convenient to use the *information form* of a Gaussian PDF.

Information Form of a Gaussian Distribution

- Recall that a Gaussian PDF is given by

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right). \tag{37}$$

Information Form of a Gaussian Distribution

- Recall that a Gaussian PDF is given by

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{37}$$

- We can expand and manipulate the inside of the $\exp(\cdot)$ to give

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x} - 2\boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \tag{38}$$

(39)

(40)

Information Form of a Gaussian Distribution

- Recall that a Gaussian PDF is given by

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{37}$$

- We can expand and manipulate the inside of the $\exp(\cdot)$ to give

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x} - 2\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \tag{38}$$

$$= \frac{\exp(-\frac{1}{2}(\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}))}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x}\right) \tag{39}$$

(40)

Information Form of a Gaussian Distribution

- Recall that a Gaussian PDF is given by

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right). \tag{37}$$

- We can expand and manipulate the inside of the $\exp(\cdot)$ to give

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \mathbf{x} - 2\boldsymbol{\mu}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu})\right) \tag{38}$$

$$= \frac{\exp(-\frac{1}{2}(\boldsymbol{\mu}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}))}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \mathbf{x}\right) \tag{39}$$

$$= \frac{\exp(-\frac{1}{2}(\boldsymbol{\eta}^\mathsf{T} \boldsymbol{\Lambda}^{-1} \boldsymbol{\eta}))}{\sqrt{(2\pi)^n \det \boldsymbol{\Lambda}^{-1}}} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T} \boldsymbol{\Lambda} \mathbf{x} + \boldsymbol{\eta}^\mathsf{T} \mathbf{x}\right) \tag{40}$$

Information Form of a Gaussian Distribution

- Recall that a Gaussian PDF is given by

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{37}$$

- We can expand and manipulate the inside of the $\exp(\cdot)$ to give

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x} - 2\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \tag{38}$$

$$= \frac{\exp(-\frac{1}{2}(\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}))}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x}\right) \tag{39}$$

$$= \frac{\exp(-\frac{1}{2}(\boldsymbol{\eta}^\mathsf{T}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}))}{\sqrt{(2\pi)^n \det \boldsymbol{\Lambda}^{-1}}} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Lambda}\mathbf{x} + \boldsymbol{\eta}^\mathsf{T}\mathbf{x}\right) \tag{40}$$

where we have defined $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ as the *information matrix* and $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$ as the *information vector*.

Information Form of a Gaussian Distribution

Information Form of a Gaussian Distribution

In summary, a Gaussian PDF can equivalently be expressed in information form, denoted $\mathcal{N}^{-1}(\boldsymbol{\eta}, \boldsymbol{\Lambda}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\mathcal{N}^{-1}(\boldsymbol{\eta}, \boldsymbol{\Lambda}) = \beta \exp\left(-\frac{1}{2}\mathbf{x}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{x} + \boldsymbol{\eta}^{\mathsf{T}}\mathbf{x}\right), \tag{41}$$

where

- β is a normalization constant given in (40),
- $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ is called the *information matrix*, and
- $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$ is called the *information vector*.

Estimate Mean and Covariance - The Linear Case

- In the linear case, the PDF of $\mathbf{x}$ is

$$p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}) = \underbrace{\frac{1}{\sqrt{(2\pi)^{K(n+p)} \det \mathbf{W}^{-1}}}}_{\text{some constant } \beta} \exp\left(-\frac{1}{2}(\mathbf{H}\mathbf{x} - \mathbf{z})^\mathsf{T}\mathbf{W}(\mathbf{H}\mathbf{x} - \mathbf{z})\right). \tag{42}$$

- We can manipulate the inside of the exponential to get

$$p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}) = \beta \exp\left(-\frac{1}{2}(\mathbf{x}^\mathsf{T}\mathbf{H}^\mathsf{T} - \mathbf{z}^\mathsf{T})\mathbf{W}(\mathbf{H}\mathbf{x} - \mathbf{z})\right), \tag{43}$$

$$= \beta \exp\left(-\frac{1}{2}(\mathbf{x}^\mathsf{T}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H}\mathbf{x} - 2\mathbf{z}^\mathsf{T}\mathbf{W}\mathbf{H}\mathbf{z} + \mathbf{z}^\mathsf{T}\mathbf{W}\mathbf{z})\right), \tag{44}$$

$$= \underbrace{\beta \exp(-\frac{1}{2}\mathbf{z}^\mathsf{T}\mathbf{W}\mathbf{z})}_{\text{new constant } \kappa} \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\underbrace{\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H}}_{\boldsymbol{\Sigma}^{-1}=\boldsymbol{\Lambda}}\mathbf{x} + \underbrace{(\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{z})^\mathsf{T}}_{\boldsymbol{\eta}^\mathsf{T}}\mathbf{x}\right) \tag{45}$$

which is exactly in the *information form* of a Gaussian PDF.

Estimate Mean and Covariance - The Linear Case

- Hence, from

$$p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}) = \kappa \exp\left(-\frac{1}{2}\mathbf{x}^\mathsf{T} \underbrace{\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H}}_{\boldsymbol{\Sigma}^{-1}=\boldsymbol{\Lambda}} \mathbf{x} + \underbrace{(\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{z})^\mathsf{T}}_{\boldsymbol{\eta}^\mathsf{T}} \mathbf{x} \right) \tag{46}$$

  we see that $\boldsymbol{\Lambda} = \mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H}$ is the *information matrix*, and $\boldsymbol{\eta} = \mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{z}$ is the *information vector*.
- Given the information matrix and information vector, it is easy to extract the covariance and mean with

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1} \tag{47}$$

$$= (\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1} \tag{48}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\eta} \tag{49}$$

$$= (\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{z} = \hat{\mathbf{x}} \tag{50}$$

- In the linear case, the mean of the distribution is also the mode.

Estimate Mean and Covariance - The Nonlinear Case

- In the nonlinear case, the PDF of $\mathbf{x}$ is

$$p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}) = \beta \exp\left(-\frac{1}{2}\mathbf{e}(\mathbf{x})^\top \mathbf{W}\mathbf{e}(\mathbf{x})\right). \tag{51}$$

  which is **not** Gaussian.

- However, we can approximate it as a Gaussian using a the first-order approximation evaluated at our estimate $\hat{\mathbf{x}}$

$$\mathbf{e}(\mathbf{x}) \approx \underbrace{\mathbf{e}(\hat{\mathbf{x}})}_{\bar{\mathbf{e}}} + \mathbf{H}(\mathbf{x} - \hat{\mathbf{x}}). \tag{52}$$

- This leads to,

$$p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}) \approx \beta \exp\left(-\frac{1}{2}(\bar{\mathbf{e}}^\top + (\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{H}^\top)\mathbf{W}(\bar{\mathbf{e}} + \mathbf{H}(\mathbf{x} - \hat{\mathbf{x}}))\right), \tag{53}$$

$$\ldots$$

$$= \kappa \exp\left(-\frac{1}{2}\mathbf{x}^\top \underbrace{\mathbf{H}^\top \mathbf{W}\mathbf{H}}_{\boldsymbol{\Lambda}} \mathbf{x} + \underbrace{(\mathbf{H}^\top \mathbf{W}\mathbf{H}\hat{\mathbf{x}} - \mathbf{H}^\top \mathbf{W}\bar{\mathbf{e}})^\top}_{\boldsymbol{\eta}^\top} \mathbf{x}\right). \tag{54}$$

Estimate Mean and Covariance - The Nonlinear Case

- Hence, from

$$p(\mathbf{x}|\check{\mathbf{x}}_0, \mathbf{u}, \mathbf{y}) \approx \kappa \exp\left(-\frac{1}{2}\mathbf{x}^\top \underbrace{\mathbf{H}^\top\mathbf{W}\mathbf{H}}_{\boldsymbol{\Lambda}} \mathbf{x} + \underbrace{(\mathbf{H}^\top\mathbf{W}\mathbf{H}\hat{\mathbf{x}} - \mathbf{H}^\top\mathbf{W}\bar{\mathbf{e}})^\top}_{\boldsymbol{\eta}^\top} \mathbf{x} \right) \tag{55}$$

  we see that $\boldsymbol{\Lambda} = \mathbf{H}^\top\mathbf{W}\mathbf{H}$ is the *information matrix*, and $\boldsymbol{\eta} = \mathbf{H}^\top\mathbf{W}\mathbf{H}\hat{\mathbf{x}} - \mathbf{H}^\top\mathbf{W}\bar{\mathbf{e}}$ is the *information vector*.

- Given the information matrix and information vector, it is easy to extract the covariance and mean with

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1} \tag{56}$$

$$= (\mathbf{H}^\top\mathbf{W}\mathbf{H})^{-1} \tag{57}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\eta} \tag{58}$$

$$= \hat{\mathbf{x}} - \underbrace{(\mathbf{H}^\top\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{W}\bar{\mathbf{e}}}_{\text{should converge to } \mathbf{0}} \tag{59}$$

Batch Estimation on Matrix Lie Groups

- The invariant framework can be leveraged for batch estimation problems where the state is an element of a matrix Lie group.
- Let the state be represented by an element of a matrix Lie group, $\mathbf{X} \in G$, with process and measurement models given by

$$\dot{\mathbf{X}}(t) = \boldsymbol{F}(\mathbf{X}(t), \mathbf{u}(t), \mathbf{w}(t)), \tag{60}$$

$$\mathbf{y}_k = \mathbf{g}_k(\mathbf{X}_k) + \mathbf{v}_k. \tag{61}$$

- Linearization using any perturbation definition will lead to

$$\delta\dot{\boldsymbol{\xi}}(t) = \mathbf{A}(t)\delta\boldsymbol{\xi}(t) + \mathbf{L}(t)\delta\mathbf{w}(t), \tag{62}$$

$$\delta\mathbf{y}_k = \mathbf{C}_k\delta\boldsymbol{\xi}_k + \mathbf{v}_k, \tag{63}$$

- Discretization using any scheme (zero-order-hold, euler) will lead to

$$\delta\boldsymbol{\xi}_k = \mathbf{A}_{k-1}\delta\boldsymbol{\xi}_{k-1} + \delta\mathbf{w}_k \tag{64}$$

$$\delta\mathbf{y}_k = \mathbf{C}_k\delta\boldsymbol{\xi}_k + \mathbf{v}_k, \tag{65}$$

Batch Estimation on Matrix Lie Groups

- After the discretization, a corresponding nonlinear discrete time process model will have the form

$$\mathbf{X}_k = \mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \tag{66}$$

- Starting from an initial guess $\check{\mathbf{X}}_0$, the error in the initial guess can be defined using the left-or right invariant error definition, with the left-invariant initial error is given by

$$\exp\left(\mathbf{e}_0^{\mathrm{L}\wedge}\right) = \mathbf{E}_0^{\mathrm{L}} = \mathbf{X}_0^{-1}\check{\mathbf{X}}_0, \tag{67}$$

  and the right-invariant initial error is given by

$$\exp\left(\mathbf{e}_0^{\mathrm{R}\wedge}\right) = \mathbf{E}_0^{\mathrm{R}} = \check{\mathbf{X}}_0\mathbf{X}_0^{-1}. \tag{68}$$

Batch Estimation on Matrix Lie Groups

- The error due to the input is denoted $\mathbf{E}_{u,k} \in G$, and can also be defined in a left- or right-invariant sense.
- The left-invariant error due to the input is given by

$$\exp\left(\mathbf{e}^{\mathrm{L}\,\wedge}_{u,k}\right) = \mathbf{E}^{\mathrm{L}}_{u,k} = \mathbf{X}_k^{-1}\mathbf{F}_{k-1}\left(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right), \tag{69}$$

whereas the right-invariant error due to the input is given by

$$\exp\left(\mathbf{e}^{\mathrm{R}\,\wedge}_{u,k}\right) = \mathbf{E}^{\mathrm{R}}_{u,k} = \mathbf{F}_{k-1}\left(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right)\mathbf{X}_k^{-1}. \tag{70}$$

Batch Estimation on Matrix Lie Groups

- Leveraging the invariant framework, recall that left- and right-invariant measurements are of the form

$$\mathbf{y}_k^{\mathrm{L}} = \mathbf{X}_k \mathbf{b}_k + \mathbf{v}_k, \tag{71}$$

$$\mathbf{y}_k^{\mathrm{R}} = \mathbf{X}_k^{-1} \mathbf{b}_k + \mathbf{v}_k, \tag{72}$$

  where $\mathbf{b}_k$ is some known column matrix.

- For left-invariant measurements, the measurement error is defined by

$$\mathbf{e}_{y,k}^{\mathrm{L}} = \mathbf{X}_k \left(\mathbf{y}_k^{\mathrm{L}} - \mathbf{g}_k \left(\mathbf{X}_k, \mathbf{0} \right) \right), \tag{73}$$

  as is done in the IEKF.

- For right-invariant measurements, the measurement error is defined by

$$\mathbf{e}_{y,k}^{\mathrm{R}} = \mathbf{X}_k^{-1} \left(\mathbf{y}_k^{\mathrm{R}} - \mathbf{g}_k \left(\mathbf{X}_k, \mathbf{0} \right) \right). \tag{74}$$

Batch Estimation on Matrix Lie Groups

- Using the invariant error definitions, the errors in the initial guess, the errors due to input, and the measurement errors can then be stacked as

$$\mathbf{e}(\mathbf{X}) = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_{u,1} \\ \vdots \\ \mathbf{e}_{u,K} \\ \mathbf{e}_{y,0} \\ \vdots \\ \mathbf{e}_{y,K} \end{bmatrix}. \tag{75}$$

- For a group affine process model and a left-or right-invariant measurement model, the Jacobian of the error, written

$$\mathbf{H} = \left. \frac{\partial \mathbf{e}(\mathbf{X})}{\partial \delta \boldsymbol{\xi}} \right|_{\mathbf{X}=\mathbf{X}^{(i)}}, \tag{76}$$

is state-estimate independent.

Batch Estimation on Matrix Lie Groups

- The Gauss-Newton algorithm becomes the following.

1. Start with an initial guess for $\check{\mathbf{X}}_0$,
2. compute the Jacobian of the error
$$\mathbf{H} = \left. \frac{\partial \mathbf{e}(\mathbf{X})}{\partial \delta \boldsymbol{\xi}} \right|_{\mathbf{X}=\mathbf{X}^{(i)}} . \tag{77}$$
3. compute the Gauss-Newton step
$$\delta \boldsymbol{\xi}^{(i)} = -(\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{e}(\mathbf{X}^{(i)}), \tag{78}$$
where $\delta \boldsymbol{\xi}^{(i)} = [\delta \boldsymbol{\xi}_0^{(i)\mathsf{T}}, \ldots, \delta \boldsymbol{\xi}_K^{(i)\mathsf{T}}]^{\mathsf{T}}$,
4. update the estimate for all $k = 0, \ldots, K$, using the appropriate left-invariant or right-invariant correction step given respectively by
$$\mathbf{X}_k^{(i+1)} = \mathbf{X}_k^{(i)} \exp\left(-\alpha \delta \boldsymbol{\xi}_k^{(i)\wedge}\right), \tag{79}$$
$$\mathbf{X}_k^{(i+1)} = \exp\left(-\alpha \delta \boldsymbol{\xi}_k^{(i)\wedge}\right) \mathbf{X}_k^{(i)}, \tag{80}$$
where α is a line search parameter,
5. and repeat until convergence.

Closing Remarks

- MAP/batch estimation is one of the most accurate and robust state estimation techniques we have today.
- However, in this form, it is not appropriate for real-time state estimation, because the complexity continues to grow as the state history gets larger and larger.
- There are many alternatives to the Gauss-Newton algorithm, such as the *Levenberg-Marquart* algorithm, which may have better performance.
- The matrix $(\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})$ is block tri-diagonal and sparse, which allows for efficient techniques to solve the Gauss-Newton step.

References

For more details, see [1]

[1] T. Barfoot, *State Estimation for Robotics*. Toronto, ON: Cambridge University Press, 2019.