Benjamin David Evans
bdevans@sun.ac.za

Recursive State Estimation

Reference notes

2023-10-06

State estimation: *a group of methods that combine available sources of data to estimate (calculate as accurately as possible) the value of variables that cannot be directly observed.*

1 Probability Theory

We define the following terminology as the basic building blocks of probability:

- **Experiment**: a physical process with an uncertain result
- **Outcome**: one possible result of the experiment
- **Sample space**: the collection of all possible outcomes of the experiment
- **Event**: a collection of outcomes, or put differently, a subset of the sample space
- **Probability** (of an event): the likelihood that the outcome of the experiment is in the event
- **Intersection** (of two events): the collection of outcomes that are in both events
- **Union** (of two events): the collection of outcomes that are in either event

Joint probability: *the probability of events A and B is the intersection of A and B.* This equation can be written for the intersection (and) or the union (or).

$$P(A \cap B) = P(A) + P(B) - P(A \cup B) \tag{1}$$

Conditional Probability: *how the probability of an event changes if we get additional information about the experiment.* Given an event B with a nonzero probability $P(B) > 0$, the conditional probability of event A is defined as,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2}$$

Bayes theorem: *how conditional properties relate to each other.* Equations 1 and 2 can be combined to write the conditional probability of A in terms of the conditional probability of B.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad \text{and} \qquad P(B|A) = \frac{P(B \cap A)}{P(A)}. \tag{3}$$

The above equations from the definition of conditional probabilities can be rewritten into Bayes' theorem,

$$\overbrace{P(A|B)}^{\text{posterior}} = \frac{\overbrace{P(B|A)}^{\text{likelihood}}\overbrace{P(A)}^{\text{prior}}}{\underbrace{P(B)}_{\text{evidence}}} \qquad \text{(Bayes' theorem)} \tag{4}$$

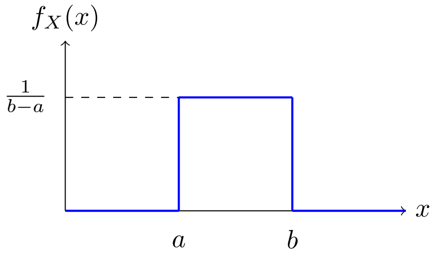
Total Probability: *If you integrate over one variable in the joint distribution, you obtain the marginal distribution.* The second equation combines this with Equation 3 that describes a joint probability $P(A, B)$ as a conditional and marginal distribution.

$$P(A) = \int p(A, B)dB = P(A) = \int p(A|B)p(B)dB \tag{5}$$

2 Random Variables

Random variable: *a real-valued, scalar variable of which the value is unknown.* Random variables are usually denoted by a capital letter, i.e. X, Y. For finite (discrete spaces), a probability can simply be assigned to each outcome. However, for infinite (continuous) spaces, a probability density function is used to describe the probability as a function of the outcome space.

Probability density functions (PDFs) are denoted by $f_X(x)$, $f(x)$ or $p(x)$. They are always positive, and the sum of the area under the graph equals 1. Integrating between two x values gives the probability of the random variable lying between those two values.



Properties:

- $\forall x, f_X(x) \geq 0$
- $\int_{-\infty}^{\infty} f_X(x)\,\mathrm{d}x = 1$
- $\int_{x_1}^{x_2} f_X(x)\,\mathrm{d}x = P(x_1 < X \leq x_2)$
- $f_X(x) \triangleq \frac{\mathrm{d}F_X(x)}{\mathrm{d}x}$

Cumulative distribution functions (CDF): *the integrals of PDF's*. The CDF at an x value is the probability of the random variable being smaller than that x value. They always start at 0 and end at 1 (since the total area under a PDF is always 1). CDFs are monotonically increasing, meaning that the y value increases or remains constant as the x value increases.

Expectation: and variance: From the PDF of a random variable X, we can calculate the expected value (mean) , variance and standard deviation. The expected value can be calculated as,

$$\overline{X} = \mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x)\mathrm{d}x. \tag{6}$$

The variance is the expectation of deviation from the mean. The standard deviation is the square root of the variance. The variance and standard deviation can be calculated as,

$$\sigma_X^2 = var[X] = \mathbb{E}[(X - \overline{X})^2] \tag{7}$$

$$\sigma_X = \sigma(X) = \sqrt{var[X]} \tag{8}$$

If you are given many samples $x_1, x_2, ..., x_N$, you can calculate the approximate mean and variance as,

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{9}$$

$$\sigma_X^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \overline{X})^2 \tag{10}$$

Multiple dimensions: when there are multiple unknown variables that influence each other, they are modelled together as a multi-dimensional PDF. For a number of random variables $\mathbf{X} = [X_1, X_2, ..., X_N]$, we define,

$$\text{Joint PDF:} \quad f_{\mathbf{X}}(\mathbf{x}) = p(\mathbf{x}) = p(x_1, x_2, ..., x_N) \triangleq \frac{\partial F_{\mathbf{X}}(\mathbf{x})}{\partial \mathbf{x}} \tag{11}$$

$$\text{Joint CDF:} \quad F_{\mathbf{X}}(\mathbf{x}) \triangleq P(\mathbf{X} \leq \mathbf{x}) \tag{12}$$

$$\text{Marginal PDF:} \quad p(x_1) = \int_{-\infty}^{\infty} p(\mathbf{x})\,\mathrm{d}x_2 \ldots \mathrm{d}x_N \tag{13}$$

$$\text{Conditional PDF:} \quad p(x_1 | x_2 \ldots x_N) = \frac{p(\mathbf{x})}{p(x_2 \ldots x_N)} \tag{14}$$

For multiple dimensions, the mean vector is straightforwardly a vector of the means of each dimension $\overline{\mathbf{X}} = [\overline{X}_1, \overline{X}_2, ..., \overline{X}_N]$. The **covariance matrix** is used to explain the impact of the variance of each variable relative to each other variable.

$$C_{\mathbf{X}} = \begin{bmatrix} C_{11} & C_{12} & \ldots & C_{1N} \\ C_{21} & C_{22} & \ldots & C_{2N} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \ldots & C_{NN} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \ldots & \rho_{1N}\sigma_1\sigma_N \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \ldots & \rho_{2N}\sigma_2\sigma_N \\ \vdots & \vdots & & \vdots \\ \rho_{1N}\sigma_1\sigma_N & \rho_{2N}\sigma_2\sigma_N & \ldots & \sigma_N^2 \end{bmatrix} \tag{15}$$

- $\rho_{ij} \in [0, 1]$ is the correlation coefficient
- $C_{\mathbf{X}}$ is symmetric: $C_{\mathbf{X}}^T = C_{\mathbf{X}}$ and diag $(C_{\mathbf{X}})$ contains the marginal variances: $C_{ii} = \sigma_{X_i}^2 = \sigma_i^2$

Each value in the matrix is the variance of the one dimension multiplied by the variance in the other direction,

$$C_{ij} \triangleq \mathbb{E}[(X_i - \overline{X_i})(X_j - \overline{X_j})] \tag{16}$$

3 Gaussian Distributions

Since it is difficult (or impossible) to determine $p(\mathbf{x})$ accurately, a Gaussian (or normal) distribution is often used.

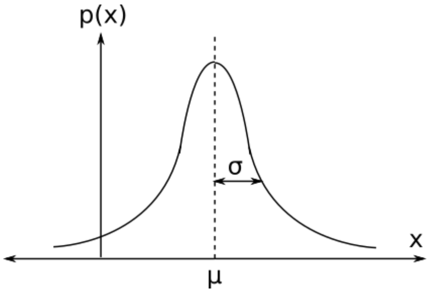
$$p(\mathbf{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \tag{17}$$

A Gaussian is fully described by mean $\overline{\mathbf{X}} = \mu$ and covariance $C_{\mathbf{X}} = \Sigma$. It provides a good model for uncertainty in many physical systems.

Univariate Gaussian distribution

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \mathcal{N}(x; \mu, \sigma^2) \tag{18}$$

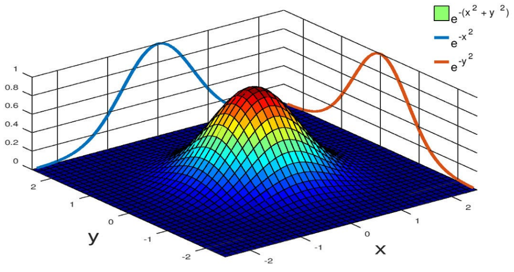
- Symmetric around the peak at the mean
- Read probability $P(X < x)$ from table
- Probability of X within 1, 2, 3 standard deviations 68%, 95%, 99%.

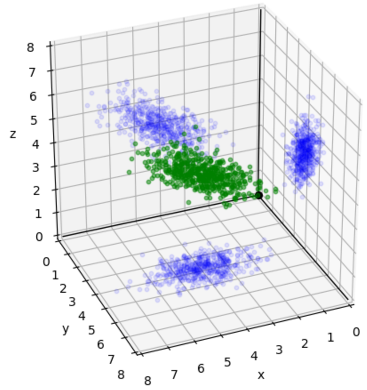


Statistical Operations: for a 2D (joint) Gaussian distribution $p(\mathbf{x}, \mathbf{y})$, the previous operations of calculating the marginal and conditional distributions. The **marginal** distribution can be calculated as,

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{xx}}). \tag{19}$$

It helps to understand the marginal distribution as a projection.





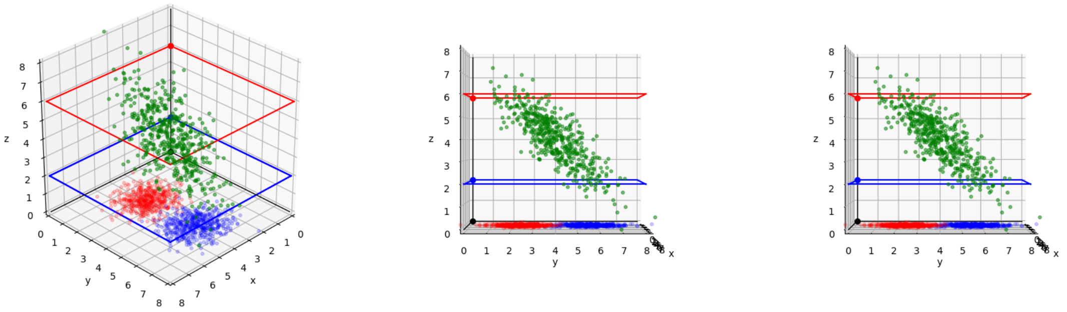
For a 2D Gaussian distribution surface, the marginal projections on each side are shown as lines.

For the 3D distribution (green points), the blue points are the projections (marginal distributions) on the 2D planes.

The **conditional** is *the probability of one random variable if information about the others is known*. It can be calculated as,

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \tag{20}$$

If the joint distribution is Gaussian, then so is the conditional. Conditioning can be interpreted as a slicing operation. In the images, the greed dots represent a 3D distribution. The blue and red slices are the conditionals for the z values of 2 and 6. The blue and red dots are the respective slices on those planes.



Multiplying Gaussian distributions is calculating the intersection of the distributions. i.e. the space represented by both distributions.

Linear transformation of Gaussians: linear systems of equations can use distributions.

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{21}$$

$$\mathbf{y} = A\mathbf{x} + \mathbf{b} \tag{22}$$

The linear transformation of a Gaussian results in another Gaussian as,

$$p(\mathbf{y}) = \mathcal{N}(A\boldsymbol{\mu} + \mathbf{b}, A\Sigma A^T). \tag{23}$$

Gaussian distributions can also be added.

The **normalisation constant** η can be calculated when the covariance is known and is often excluded from equations.

$$\eta = \frac{1}{\sqrt{|2\pi\Sigma|}} \tag{24}$$

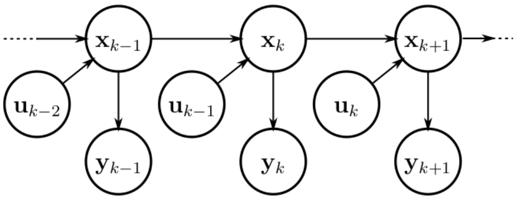
4 Bayes Filter

Sate estimation is *the problem of calculating the most likely states given a model of the system, the control inputs and the measurements.* State estimation is difficult due to noisy sensors and measurements. The Bayes filter is the general form of all estimation algorithms. It is defined for discrete time steps $k \in 1, 2, ..., k$.

Markov assumption: *the next state is only dependent on the current state and current inputs.* The future in independent of the past, given the present. Mathematically,

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{u}_{0:k-1}, \mathbf{y}_{1:k-1}) &= p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \\ p(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{u}_{0:k-1}, \mathbf{y}_{1:k-1}) &= p(\mathbf{y}_k|\mathbf{x}_k) \end{aligned} \tag{25}$$

This image shows the flow as time progresses. State $\mathbf{x}_k$ is the result from a previous state $\mathbf{x}_{k-1}$ and control $\mathbf{u}_{k-1}$. Each state $\mathbf{x}_k$ produces an output $\mathbf{y}_k$.



The task of state estimation is, given the previous state $\mathbf{x}_{k-1}$ control $\mathbf{u}_{k-1}$ and the output $\mathbf{y}_k$, calculate the current (latent or non-visible) state $\mathbf{x}_k$.
Recursive state estimation adjusts the problem from having the true state $\mathbf{x}_{k-1}$, to having a belief of the previous state (probability distribution of what the state is), written as bel$(\mathbf{x}_k)$.

Models: Filtering requires several models of the system. We can use kinematic equations to construct state updates for the system in the form,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_k), \tag{26}$$

where $\mathbf{w}_t$ is a process noise vector that describes the noise in the system. Since the noise is non-deterministic, this can be rewritten as a conditional distribution, called the **motion model**,

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_k). \tag{27}$$

For the measurement, we can represent the measurement $\mathbf{z}_k$ as a function of the state and some measurement noise as $f(\mathbf{x}_k, \mathbf{v}_k)$, where $\mathbf{v}_k$ is the measurement noise. Similarly, we can transform this equation to a distribution called the **measurement model**,

$$p(\mathbf{z}_k|\mathbf{x}_k) \tag{28}$$

Objective: use the motion and measurement models to compute an estimate of the belief at the current time. The **belief** is the probability distribution of the current state, given all the previous controls and observations. The belief is written as

$$\text{bel}(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{z}_{0:k}, \mathbf{u}_{0:k-1}) \tag{29}$$

Bayes filter derivation: these steps are how the Bayes filter is derived from the Bayes rule. In the derivation, the normal (not bold) symbols are used for simplicity.

$$\text{bel}(x_k) = p(x_k|z_{0:k}, u_{0:k-1}) \tag{30}$$

Applying the Bayes rule (Equation 4) with,

A = x_k, B = z_k, and u is added into every distribution.

$$= \frac{p(z_k|x_k, z_{0:k-1}, u_{0:k-1})p(x_k|z_{0:k-1}, u_{0:k-1})}{(z_k|z_{0:k-1}, u_{0:k-1})} \tag{31}$$

$$= \eta p(z_k|x_k, z_{0:k-1}, u_{0:k-1})p(x_k|z_{0:k-1}, u_{0:k-1}) \qquad \because p(z_k) \text{ is constant} \tag{32}$$

The Markov assumption says that the state is all that is needed for the measurement

$$= \eta p(z_k|x_k)p(x_k|z_{0:k-1}, u_{0:k-1}) \tag{33}$$

Use the theorem of total probability (Equation 5) to replace the 2nd term

$$= \eta p(z_k|x_k) \int p(x_k|x_{k-1}, z_{0:k-1}, u_{0:k-1})p(x_{k-1}|z_{0:k-1}, u_{0:k-2})dx_{k-1} \tag{34}$$

Use Markov assumption to remove redundant terms

$$= \eta p(z_k|x_k) \int p(x_k|x_{k-1}, u_k)p(x_{k-1}|z_{0:k-1}, u_{0:k-2})dx_{k-1} \tag{35}$$

The result of this derivation is the Bayes filter, which comprises the following parts,

$$\text{bel}(\mathbf{x}_k) = \eta \underbrace{p(\mathbf{z}_k|\mathbf{x}_k)}_{\substack{\text{Measurement}\\ \text{model}}} \int \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)}_{\substack{\text{Process}\\ \text{model}}} \underbrace{p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1}, \mathbf{u}_{0:k-2})}_{\text{bel}(\mathbf{x}_{k-1})} d\mathbf{x}_{k-1} \tag{36}$$

$$\text{bel}(\mathbf{x}_k) = \eta p(\mathbf{z}_k|\mathbf{x}_k) \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)\text{bel}(\mathbf{x}_{k-1}) \tag{37}$$

$$\text{bel}(\mathbf{x}_k) = \eta p(\mathbf{z}_k|\mathbf{x}_k)\overline{\text{bel}}(\mathbf{x}_k) \tag{38}$$

Bayes filter updates: The filter works recursively using the control and measurement update steps.

1. The **control update** uses the previous belief $\text{bel}(\mathbf{x}_{k-1})$ to calculate the **prior belief** $\overline{\text{bel}}(\mathbf{x}_{k-1})$.

$$\overline{\text{bel}}(\mathbf{x}_k) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})\, \text{bel}(\mathbf{x}_{k-1})\, \text{d}\mathbf{x}_{k-1} \tag{39}$$

2. The **measurement update** uses the prior and the likelihood of the measurement to calculate the posterior (updated belief).

$$\text{bel}(\mathbf{x}_k) = \eta\, p(\mathbf{y}_k|\mathbf{x}_k)\, \overline{\text{bel}}(\mathbf{x}_k) \tag{40}$$

5 Linear Kalman Filter

The linear Kalman Filter (LKF) is the application of the Bayes filter to linear dynamics systems with zero-mean, Gaussian-distributed noise. These assumptions allow us to simplify the integrals and distributions into matrix equations.

1. Linear systems:

$$\mathbf{x}_k = A_k\mathbf{x}_{k-1} + B_k\mathbf{u}_{k-1} + \mathbf{w}_k \tag{41}$$

$$\mathbf{y}_k = C_k\mathbf{x}_k + \mathbf{v}_k \tag{42}$$

$\mathbf{x}_k \rightarrow N$-dimensional state vector, $\mathbf{y}_k \rightarrow M$-dimensional output vector

2. Zero-mean Gaussian-distributed white noise:

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k) \tag{43}$$
$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, R_k) \tag{44}$$

   White noise → $\mathbf{w}_k$ and $\mathbf{v}_k$ independent of all other noise, state and input values

3. The initial belief distribution is Gaussian:

$$\text{bel}(\mathbf{x}_0) = p(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0) \tag{45}$$

   Kalman filter is recursive and called at every sampling instant by calculating the motion and measurement updates. If previous belief distribution is Gaussian, the next belief will also be Gaussian:

$$\overline{\text{bel}}(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{u}_{0:k-1}, \mathbf{y}_{1:k-1}) = \mathcal{N}(\overline{\boldsymbol{\mu}}_k, \overline{\Sigma}_k) \tag{46}$$
$$\text{bel}(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{u}_{0:k-1}, \mathbf{y}_{1:k}) = \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k) \tag{47}$$

The time-varying LKF algorithm is given by:

```
Input: $\boldsymbol{\mu}_{k-1}$, $\Sigma_{k-1}$, $\mathbf{u}_{k-1}$, $\mathbf{y}_k$
Output: $\boldsymbol{\mu}_k$, $\Sigma_k$
$\overline{\boldsymbol{\mu}}_k = A_k\boldsymbol{\mu}_{k-1} + B_k\mathbf{u}_{k-1}$    ▷ Calculate prior mean
$\overline{\Sigma}_k = A_k\Sigma_{k-1}A_k^T + Q_k$    ▷ Calculate prior standard deviation
$L_k = \overline{\Sigma}_k C_k^T \left(C_k\overline{\Sigma}_k C_k^T + R_k\right)^{-1}$    ▷ Calculate Kalman gains
$\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + L_k(\mathbf{y}_k - C_k\overline{\boldsymbol{\mu}}_k)$    ▷ Calculate posterior mean
$\Sigma_k = (I - L_kC_k)\,\overline{\Sigma}_k$    ▷ Calculate posterior deviation
```

Control systems application (steady-state KF): Constant noise covariance matrices Q and R result in the gains converging quickly. Therefore, constant gains L_c are typically used and found using the Ricatti equation offline. This assumption results in the reduction of the above equations to,

$$\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + L_c(\mathbf{y}_k - C\overline{\boldsymbol{\mu}}_k) \tag{48}$$
$$\widehat{\mathbf{x}}_k = A\widehat{\mathbf{x}}_{k-1} + B\mathbf{u}_{k-1} + L_c(\mathbf{y}_k - C(A\widehat{\mathbf{x}}_{k-1} + B\mathbf{u}_{k-1})) \tag{49}$$

6 Extended Kalman Filter

Extended Kalman Filter (EKF): The adaptation of the LKF to non-linear systems by using Taylor approximations. The EKF is an approximation but tractable solution to non-linear problems. The assumptions for the EKF are a nonlinear model with additive Gaussian noise.

Taylor Series Expansion: *an infinite sum of terms that are expressed in terms of the function's derivatives at a single point.*

$$f(s) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots. \tag{50}$$

The Taylor expansion can be used to obtain a first-order approximation around the point $x = a$:

$$\begin{aligned} f(x) &\approx f(a) + f'(a)(x-a) && (51)\\ &= f'(a)\,x + (f(a) - f'(a)\,a) && (52)\\ &= mx + c && (53) \end{aligned}$$

For a multivariate vector function $\mathbf{f}(\mathbf{x})$:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{a}) + F(\mathbf{a})(\mathbf{x} - \mathbf{a}) \tag{54}$$

The Jacobian $F(\mathbf{x})$ is an $M \times N$ matrix of partial derivatives:

$$F(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_M}{\partial x_1} & \frac{\partial f_M}{\partial x_2} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} \tag{55}$$

Rearranging into standard linear form:

$$\mathbf{f}(\mathbf{x}) \approx F(\mathbf{a})\,\mathbf{x} + (\mathbf{f}(\mathbf{a}) - F(\mathbf{a})\,\mathbf{a}) = A\mathbf{x} + \mathbf{b} \tag{56}$$

The EKF changes from assuming linear updates to using probability density functions to represent the updates.

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_k \qquad (57) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \qquad (58) \end{aligned}$$

The EKF algorithm works by linearising around the best estimate (mean) and then recursively updating the means and covariance. The motion and measurement Jacobians are the first-order derivatives at the respective means:

$$\begin{aligned} F_k &= \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}_{k-1})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\boldsymbol{\mu}_{k-1}} \qquad (59) \\ H_k &= \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\overline{\boldsymbol{\mu}}_k} \qquad (60) \end{aligned}$$

The EKF algorithm is:

```
Input: $\boldsymbol{\mu}_{k-1}$, $\Sigma_{k-1}$, $\mathbf{u}_{k-1}$, $\mathbf{y}_k$
Output: $\boldsymbol{\mu}_k$, $\Sigma_k$
$\overline{\boldsymbol{\mu}}_k = \mathbf{f}(\boldsymbol{\mu}_{k-1}, \mathbf{u}_{k-1})$    ▷ Use the linearised motion model
$\overline{\Sigma}_k = F_k \Sigma_{k-1} F_k^T + Q_k$    ▷ Use the motion Jacobians $F_k$
$L_k = \overline{\Sigma}_k H_k^T \left( H_k \overline{\Sigma}_k H_k^T + R_k \right)^{-1}$    ▷ Use the measurement Jacobians $H_k$
$\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + L_k(\mathbf{y}_k - \mathbf{h}(\overline{\boldsymbol{\mu}}_k))$    ▷ Use the linearised measurement model
$\Sigma_k = (I - L_k H_k)\, \overline{\Sigma}_k$    ▷ Use the measurement Jacobians $H_k$
```

EKF Implementation Notes: While the LKF provides the covariance matrix Σ as in indication of the accuracy and convergence, the EKF provides no such guarantees. A small sigma matrix does not necessarily indicate good performance. Operation, stability and accuracy are often coupled to the sampling period.

7 Unscented Kalman Filter

Unscented Kalman Filter (UKF): Alternative state estimation for nonlinear systems that approximates belief distributions with Gaussians. The UKF samples belief, passes samples through nonlinear transforms and calculates approximate Gaussian distributions from the resulting samples.

Unscented Transform:

UT approximates a nonlinear transform of a Gaussian distribution with a Gaussian:

$$\begin{aligned} \mathbf{y} &= \mathbf{f}(\mathbf{x}) \qquad (61) \\ p(\mathbf{x}) &= \mathcal{N}(\boldsymbol{\mu}, \Sigma) \qquad (62) \\ p(\mathbf{y}) &\approx \mathcal{N}(\boldsymbol{\mu}', \Sigma') \qquad (63) \end{aligned}$$

- Draw $2N + 1$ deterministic sample points (called **sigma points**) from $p(\mathbf{x})$:

  $$\mathcal{X} = \left[\mathbf{x}^{[0]} \quad \dots \quad \mathbf{x}^{[2N]} \right] = \left[\mathbf{0}, \quad \gamma\sqrt{\Sigma}, \quad -\gamma\sqrt{\Sigma} \right] + \boldsymbol{\mu}\mathbf{1}^T \tag{64}$$

  The square root matrix can be found using a Cholesky decomposition, and the scaling parameter γ determines how far sigma points are from the mean.

- Propagate each sigma point $\mathbf{x}^{[i]}$ through nonlinear function $\mathbf{f}(\mathbf{x})$:

  $$\mathbf{y}^{[i]} = \mathbf{f}\left(\mathbf{x}^{[i]}\right) \tag{65}$$

- Calculate mean and covariance from transformed sigma points:

  $$\begin{aligned} \boldsymbol{\mu}' &= \sum_{i=0}^{2N} w_m^{[i]} \mathbf{y}^{[i]} \qquad (66) \\ \Sigma' &= \sum_{i=0}^{2N} w_c^{[i]} \left(\mathbf{y}^{[i]} - \boldsymbol{\mu}' \right) \left(\mathbf{y}^{[i]} - \boldsymbol{\mu}' \right)^T \qquad (67) \end{aligned}$$

Relative weights $w_m^{[i]}$ and $w_c^{[i]}$ different for each sigma point and determined by γ.

Unscented Kalman Filter - Motion Update: The UKF has more steps than we have previously seen; therefore, the motion and measurement updates are separated.

```
Input: $\boldsymbol{\mu}_{k-1}$, $\Sigma_{k-1}$, $\mathbf{u}_{k-1}$
Output: $\overline{\boldsymbol{\mu}}_k$, $\overline{\Sigma}_k$
1: $\mathcal{X}_{k-1} = \left[\mathbf{0}, \quad \gamma\sqrt{\Sigma_{k-1}}, \quad -\gamma\sqrt{\Sigma_{k-1}}\right] + \boldsymbol{\mu}_{k-1}\mathbf{1}^T$    ▷ Calculate sigma points
2: $\overline{\mathcal{X}}_k = \mathbf{f}\left(\mathcal{X}_{k-1}, \mathbf{u}_{k-1}\right)$    ▷ Propagate points through the function
3: $\overline{\boldsymbol{\mu}}_k = \sum_{i=0}^{2N} w_m^{[i]} \overline{\mathcal{X}}_k^{[i]}$    ▷ Calculate the new mean
4: $\overline{\Sigma}_k = \sum_{i=0}^{2N} w_c^{[i]} \left(\overline{\mathcal{X}}_k^{[i]} - \overline{\boldsymbol{\mu}}_k\right) \left(\overline{\mathcal{X}}_k^{[i]} - \overline{\boldsymbol{\mu}}_k\right)^T + Q_k$    ▷ Calculate new standard deviation
```

Unscented Kalman Filter - Measurement Update:

```
Input: $\overline{\boldsymbol{\mu}}_k$, $\overline{\Sigma}_k$, $\mathbf{y}_k$
Output: $\boldsymbol{\mu}_k$, $\Sigma_k$
1: $\overline{\mathcal{X}}_k = \left[\mathbf{0}, \quad \gamma\sqrt{\overline{\Sigma}_k}, \quad -\gamma\sqrt{\overline{\Sigma}_k}\right] + \overline{\boldsymbol{\mu}}_k\mathbf{1}^T$    ▷ Calculate sigma points
2: $\overline{\mathcal{Y}}_k = \mathbf{h}\left(\overline{\mathcal{X}}_k\right)$    ▷ Propagate sigma points
3: $\overline{\boldsymbol{\mu}}_k^{\mathbf{y}} = \sum_{i=0}^{2N} w_m^{[i]} \overline{\mathcal{Y}}_k^{[i]}$    ▷ Calculate mean of expected measurement
4: $\overline{\Sigma}_k^{\mathbf{y},\mathbf{y}} = \sum_{i=0}^{2N} w_c^{[i]} \left(\overline{\mathcal{Y}}_k^{[i]} - \overline{\boldsymbol{\mu}}_k^{\mathbf{y}}\right) \left(\overline{\mathcal{Y}}_k^{[i]} - \overline{\boldsymbol{\mu}}_k^{\mathbf{y}}\right)^T + R_k$    ▷ Calculate std. dev. of measurement expectation
5: $\overline{\Sigma}_k^{\mathbf{x},\mathbf{y}} = \sum_{i=0}^{2N} w_c^{[i]} \left(\overline{\mathcal{X}}_k^{[i]} - \overline{\boldsymbol{\mu}}_k\right) \left(\overline{\mathcal{Y}}_k^{[i]} - \overline{\boldsymbol{\mu}}_k^{\mathbf{y}}\right)^T$    ▷ Calculate the std. dev. of the measurement based on the state
6: $L_k = \overline{\Sigma}_k^{\mathbf{x},\mathbf{y}} \left(\overline{\Sigma}_k^{\mathbf{y},\mathbf{y}}\right)^{-1}$    ▷ Calculate the Kalman update
7: $\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + L_k(\mathbf{y}_k - \overline{\boldsymbol{\mu}}_k^{\mathbf{y}})$    ▷ Calculate the posterior mean
8: $\Sigma_k = \overline{\Sigma}_k - L_k \overline{\Sigma}_k^{\mathbf{y},\mathbf{y}} L_k^T$    ▷ Calculate the posterior std.
```

Lines 3, 4 calculate the distribution of the sigma points after they have passed through the model. This is the probability that the measurement is true. Line 5 is the probability of the measurement based on the state. Previously, the Kalman gain L was the ratio of uncertainty in the measurement itself vs uncertainty in the chance of the measurement. This general form is kept, and the probability of the measurement based on the state is divided by the probability of the measurement based on the measurement model. Lines 6-8, then apply the Kalman update using these measurements.

Application of UKF: While the EKF linearises around a single point, the UKF linearises around multiple (sigma) points. If the model is not differentiable, then UKF is the only option. The UKF has a resemblance to the particle filter.

8 Particle Filter

The **particle filter** (PF) is designed for non-parametric distributions, beliefs represented by a set of samples (particles) that do not form a parameterised distribution. It is more expressive than a Gaussian since you can use multiple particles instead of only a mean and covariance. While the UKF uses deterministic samples, the PF uses random samples.

The PF algorithm uses a state transition distribution, $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$. A set of particles $\overline{\mathcal{X}}$ is kept for the prior and resampled to form the posterior $\overline{\mathcal{X}}$.

```
Input: $\mathcal{X}_{k-1}$, $\mathbf{u}_{k-1}$, $\mathbf{y}_k$
Output: $\mathcal{X}_k$
1: $\overline{\mathcal{X}}_k = \mathcal{W}_k = \mathcal{X}_k = \emptyset$
2: for $i = 1, P$ do
3:     sample $\mathbf{x}_k^{[i]} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{[i]}, \mathbf{u}_{k-1})$   ▷ Sample a particle from the state transition distribution
4:     $w_k^{[i]} = p(\mathbf{y}_k|\mathbf{x}_k^{[i]})$   ▷ Calculate the weight of the particle as the probability of the measurement
5:     add $\mathbf{x}_k^{[i]}$ to $\overline{\mathcal{X}}_k$, $w_k^{[i]}$ to $\mathcal{W}_k$   ▷ Add the particle to the set of prior particles
6: end for
7: for $i = 1, P$ do
8:     add $\mathbf{x}_k^{[j]}$ to $\mathcal{X}_k$ with probability $\propto w_k^{[j]}$   ▷ Resample the particles based on their weights
9: end for
```

In the algorithm, line 3 is the control update of selecting a next state based on the previous state and control. The belief $\overline{\text{bel}}(\mathbf{x})$ is represented as a set of particles. The weight in line 4 is normally calculated based on the difference between what the measurement of the state would be and the actual measurement received $\mathbf{y}_k$. Once the total number of particles (a tuneable parameter) has been sampled, they are resampled based on the weights calculated for each particle. The resampling ensures that particles with higher importance weights occur more frequently and lower weights will disappear.